

**El Uso de Estrategias Colaborativas Mediadas por Tecnología.
La Enseñanza de Programación en el Primer Año de la
Licenciatura En Sistemas de la UNRN**



Tesista: Edith Noemí Lovos

Director : Mgter. Alejandro González Príncipe

Co-Directora: Dra. Inés Fernández Mouján

Tesis presentada para obtener el grado de Magister en
Tecnología Informática Aplicada en Educación

FACULTAD DE INFORMÁTICA
UNIVERSIDAD NACIONAL DE LA PLATA

Diciembre, 2013

Dedicatorias

A mi madre y a Camila.

Agradecimientos

A Dios que me acompaña cada día.

A Tommy , Martín y Nico por esperarme siempre con una sonrisa .

A los directores de esta tesis, Alejandro González e Inés Mouján, no solo por la formación académica, que ha sido muy importante y valiosa, sino por confiar en la propuesta y por ayudarme a transitar este camino, en el que compartimos más que trabajo.

A mi director Alejandro, le agradezco especialmente, la formación mientras compartimos la docencia, y el haberme animado a empezar con la maestría y a seguir adelante.

A Inés, le agradezco enormemente la paciencia y la guía en los primeros pasos de esta investigación.

A los alumnos y docentes que participaron de la experiencia, gracias por su colaboración.

A Sole Vercellino y a Tatiana Gibelli, por sus sugerencias y observaciones y por su amistad.

A Rosario Molfino, de la Biblioteca de la Facultad de Informática de la Universidad Nacional de La Plata, por facilitarme materiales de lectura y asesoramiento.

A la Secretaría de Investigación, Desarrollo y Transferencia Tecnológica (SIDTT) de la Universidad de Río Negro (UNRN), por la beca que me permitió llevar adelante este trabajo. En particular al director de mi beca, Rodolfo Bertone por su confianza y asesoramiento.

Por último al Dr. Daniel Barrios y a Sofia Pocai de la SIDTT de la Sede Atlántica de la UNRN, por la atención y la gestión.

Indice General

| | |
|---|-----|
| 1. CAPÍTULO I. INTRODUCCIÓN AL PROBLEMA DE INVESTIGACIÓN..... | 6 |
| MARCO TEORICO..... | 9 |
| 2. CAPÍTULO II. ENSEÑANZA DE LA PROGRAMACIÓN. ESTRATEGIAS Y HERRAMIENTAS..... | 10 |
| 2.1 Estrategias Utilizadas..... | 13 |
| 2.1.1 Clases Magistrales y de Laboratorio..... | 14 |
| 2.1.2 Software Visualization..... | 15 |
| 2.1.3 Robots..... | 19 |
| 2.1.4 Aprendizaje Basado en Problemas (ABP)..... | 21 |
| 2.1.5 Aprendizaje Cognitivo..... | 23 |
| 2.2 RESUMEN DEL CAPÍTULO..... | 25 |
| 3. CAPÍTULO III. TRABAJO COLABORATIVO, TIC Y LA ENSEÑANZA Y APRENDIZAJE DE LA PROGRAMACIÓN..... | 26 |
| 3.1 Introducción..... | 26 |
| 3.2 Enseñar y Aprender en colaboración..... | 28 |
| 3.2.1 De los Agrupamientos en el trabajo colaborativo | 32 |
| 3.2.2 Beneficios pedagógicos..... | 33 |
| 3.2.3 Aprendizaje colaborativo Mediado por Tecnología..... | 34 |
| 3.2.3.1 Beneficios del uso de CSCL..... | 37 |
| 3.3 La Enseñanza Colaborativa de la Programación..... | 38 |
| 3.3.1 Herramientas Colaborativas para la enseñanza y el aprendizaje de la Programación..... | 39 |
| 3.3.1.1 COLLEGE (COLLaborative Edition, Compiling and Execution of Programs)..... | 40 |
| 3.3.1.2 EclipseGavab..... | 43 |
| 3.3.1.3 Virtual Programming Lab (VPL)..... | 45 |
| 3.4 RESUMEN DEL CAPÍTULO..... | 48 |
| DESARROLLO DE LA INVESTIGACION..... | 50 |
| 4. CAPÍTULO IV METODOLOGÍA..... | 51 |
| 4.1 TÉCNICAS DE RECOLECCIÓN DE DATOS..... | 52 |
| 4.1.1 Observación..... | 52 |
| 4.1.2 Encuestas..... | 52 |
| 4.1.3 Entrevistas | 53 |
| 4.2 Recursos TIC disponibles dentro del aula virtual de la materia Programación I | 54 |
| 5. CAPÍTULO V. DISEÑO DE LA PROPUESTA DE ENSEÑANZA Y DE APRENDIZAJE | 55 |
| 5.1 Contexto..... | 55 |
| 5.2 Destinatarios..... | 56 |
| 5.3 Contenidos Curriculares e Inserción en la Carrera | 57 |
| 5.4 Metodología de Trabajo en la Asignatura..... | 59 |
| 5.4.1 Acreditación de la Asignatura..... | 60 |
| 5.4.2 Fases del Desarrollo de las APE..... | 63 |
| 6. CAPITULO VI. ANÁLISIS DE RESULTADOS DE LA IMPLEMENTACIÓN..... | 69 |
| 6.1 La Evaluación de los Alumnos Respecto de sus Propios Procesos de Aprendizaje..... | 77 |
| 6.2 La Valoración de los Alumnos Respecto al Trabajo Colaborativo..... | 85 |
| 6.3 La Evaluación de los Alumnos Respecto a la Enseñanza | 91 |
| 6.4 Análisis de las Habilidades Desarrolladas en el Marco de las APE..... | 95 |
| 7. CAPÍTULO VII – CONCLUSIONES..... | 98 |
| 7.1 Conclusión I..... | 98 |
| 7.2 Conclusión II..... | 99 |
| 7.3 Conclusión III..... | 100 |
| 7.4 Dificultades Encontradas..... | 100 |
| 7.5 TRABAJO FUTURO..... | 101 |
| BIBLIOGRAFÍA..... | 103 |
| ANEXOS..... | 110 |
| ANEXO I - CONSIGNAS APE..... | 111 |
| APE Nro 1 – Grupos Pares - La Tapadita..... | 111 |
| APE Nro 1 – Grupos Impares -Teclado Telefónico..... | 115 |
| APE Nro 2 – Grupos Impares - Cuadrado Magico..... | 118 |

| | |
|---|-----|
| APE Nro 2 – Grupos Pares – Tres en Linea..... | 121 |
| APE Nro 3 – Grupos Impares - Pilas..... | 124 |
| APE Nro 3 – Grupos Pares -Colas..... | 127 |
| ANEXO II - ENCUESTAS | 130 |
| Encuestas APE 1 y 2..... | 130 |
| Encuesta APE 3..... | 133 |
| ANEXO III - OBSERVACIÓN DE CLASE..... | 134 |
| ANEXO IV - PROGRAMA DE LA ASIGNATURA..... | 139 |

Indice de Imagenes

| | |
|---|----|
| Imagen 1 - COLLEGE interface..... | 42 |
| Imagen 2 - EclipseGavab interface..... | 44 |
| Imagen 3 - VPL Modo Docente..... | 47 |
| Imagen 4 - UNRN - Distribución geográfica..... | 56 |
| Imagen 5 - Plataforma Virtual Lic. en Sistemas UNRN – Sede Atlántica..... | 60 |
| Imagen 6 - Aula Virtual Estructura Unidad 5 | 62 |
| Imagen 7 - Fases de Desarrollo de las APE | 63 |
| Imagen 8 - Listado de temas en el Foro Grupos tema Pilas..... | 64 |
| Imagen 9 - Foro Debate APE2 – Tema Tablero Mágico..... | 65 |
| Imagen 10 - VPL – Entregas Tema Cuadrado Mágico..... | 66 |
| Imagen 11 - VPL – Evaluación Grupo 2-4. Tema Cuadrado Mágico..... | 67 |
| Imagen 12 - Grupos APE1..... | 73 |
| Imagen 13 - Grupos APE2..... | 74 |
| Imagen 14 - Grupos APE3..... | 74 |
| Imagen 15 - Presentación Power Point. APE1 – Grupo 3..... | 95 |
| Imagen 16 - Presentación Power Point. APE1 – Grupo 1..... | 96 |
| Imagen 17 - Presentación Power Point. APE3 – Grupo G2-4..... | 97 |

Indice de Figuras

| | |
|--|----|
| Figura 1: Sistema de Información | 11 |
| Figura 2: Trabajo Colaborativo. Niveles de intervención..... | 36 |

Indice de Tablas

| | |
|---|----|
| Tabla 1. Cronograma de Actividades & Recursos APE2..... | 61 |
| Tabla 2. Programación de las APE | 70 |
| Tabla 3. Programación de las asignaturas introductorias a la UNRN | 72 |
| Tabla 4. Cantidad de grupos por etapas de las APE..... | 75 |
| Tabla 5. Encuesta - Auto-evaluación..... | 77 |
| Tabla 6. Correlación entre Interes y Comprensión de la materia..... | 82 |
| Tabla 7. Encuesta - Evaluación del Grupo de Trabajo..... | 85 |
| Tabla 8. Encuesta – Evaluación del Tutor..... | 91 |

1. CAPÍTULO I. INTRODUCCIÓN AL PROBLEMA DE INVESTIGACIÓN

La presente investigación trabaja en torno a tres aspectos dentro de la enseñanza y aprendizaje de la carrera de Sistemas, abordando:

- la complejidad que implica el proceso de resolución de problemas a través de la algoritmia.
- el contexto educativo actual, donde el esquema convencional que posiciona al docente en un rol de enseñante y al alumno en un rol de aprendiz, ha migrado a otro. Dónde, el aprendizaje se considera un proceso social, construido a través de la interacción de todos sus miembros (docentes – alumnos, alumnos - alumnos) y donde el contexto y el significado que cada uno le asigna a lo que aprende adquiere suma importancia. (Zañartu Correa (2003), Maldonado Pérez (2007)).
- la formación del futuro profesional como parte de una sociedad, que se encuentra atravesada por las denominadas Tecnologías de la Información y la Comunicación (TIC).

Desde esta perspectiva constructivista del aprendizaje, lo más importante es que el estudiante pueda crear y comprender nueva información y conocimientos a través del proceso en el que los nuevos conocimientos se van adaptando a los conocimientos previos, es decir a través del andamiaje (Vázquez Mariño, 2011).

López Segrera en Zuñiga (2012) afirma que la metodología de trabajo colaborativo está en sintonía con el cambio de paradigma que se plantea en el ámbito académico de nivel universitario, donde se le otorga al estudiante una mayor responsabilidad en el aprendizaje y se persigue una formación orientada al “saber hacer”, al “aprender a aprender” y al desarrollo de competencias profesionales. Se puede encuadrar como antecedente de estos desarrollos a la perspectiva teórica de Lev Vygotsky (Maldonado Pérez, 2007); que pone el énfasis en la *interacción social* como factor clave para el aprendizaje y la transmisión de cultura. Según Johnson et al.(1999), Vygotsky sostenía el carácter social del conocimiento y su construcción a partir de los esfuerzos cooperativos por aprender, entender y resolver problemas. Un concepto clave, definido por Vygostky(1978) , es el de la zona de desarrollo próximo, entendiéndola como aquella zona situada entre lo que un estudiante puede hacer solo y lo que puede lograr si trabaja guiado por un instructor o en colaboración con otros pares más avanzados. Así, la enseñanza y como consecuencia el aprendizaje, sólo tiene lugar en la zona en la que el sujeto puede desarrollar una actividad en colaboración con otro (Vygostky,1978). En este sentido, Johnson et al. (1999), sostiene que a menos que los alumnos trabajen de manera cooperativa, no crecerán

intelectualmente; por lo tanto, debe reducirse al mínimo el tiempo que los alumnos pasan trabajando solos en las actividades académicas. En este nuevo enfoque de trabajo resulta fundamental la participación activa tanto del docente como del alumnos. El docente es dinamizador, orientador y asesor de todo el proceso de enseñanza y aprendizaje el alumno es un sujeto activo que propone, pregunta y debate (Filippi; et al. 2010).

¿Cómo establecemos el nexo entre estas teorías, el trabajo colaborativo y el desarrollo de la programación? Como afirman las investigaciones, el trabajo colaborativo en el ámbito académico resulta beneficioso para el desarrollo de las actividades grupales, pues en estas los alumnos necesitan comunicarse, discutir y emitir opiniones a otros miembros del grupo. En particular en las carreras vinculadas a la enseñanza de las ciencias informáticas, a medida que los estudiantes avanzan en la carrera, profundizan los conceptos vinculados al desarrollo de un producto de software (análisis de requerimientos, diseño, implementación, testing) . Esta actividad en el ámbito profesional rara vez se desarrolla en soledad, debido principalmente a la complejidad que presentan los problemas a tratar. Así, el desarrollo del software resulta una actividad que requiere del trabajo en equipo y en colaboración. En este sentido, los ambientes colaborativos pueden ofrecer un importante soporte a los alumnos durante las actividades aprendizaje de la programación; pues la resolución de problemas a través de la colaboración alienta la reflexión, un mecanismo que estimula el proceso de aprendizaje. Surge así, la necesidad de utilizar estrategias colaborativas en el ámbito de la enseñanza y del aprendizaje de la programación desde los inicios de la formación del estudiante de sistemas.

Siguiendo estos lineamientos en este trabajo de investigación, se propone diseñar e implementar una estrategia de trabajo colaborativo para la enseñanza y aprendizaje de la programación en las asignaturas vinculadas a la misma desde la etapa inicial de la formación. Para lo cual como objetivo principal se pretende:

- Investigar y determinar cómo la metodología de trabajo colaborativo aplicada a la enseñanza de la programación potencia el aprendizaje.

Con la intención de alcanzar esta finalidad se plantean los siguientes objetivos específicos:

- Realizar una caracterización del trabajo colaborativo, su aplicación mediada por la tecnología, en el ámbito educativo de nivel universitario.
- Indagar sobre los procesos de resolución de problemas, en particular aquellos que pueden expresarse a través de un algoritmo.
- Evaluar y seleccionar los productos de software que posibiliten el desarrollo colaborativo de las actividades de programación y que mejor se adapten al ámbito educativo de nivel superior, en particular al contexto de la Licenciatura en Sistemas de la Universidad Nacional de Río Negro (UNRN) – Sede Atlántica.

- Analizar cómo el uso de los recursos TIC promueven un ambiente colaborativo, que permite potenciar las habilidades necesarias para el desarrollo de software, en las prácticas de las materias de programación de los primeros años de la Lic. en Sistemas de la UNRN.

A través de este análisis, se pretende conocer las valoraciones de los alumnos sobre sus propios procesos de aprendizaje, el trabajo colaborativo y la enseñanza en el marco de una propuesta de enseñanza y aprendizaje de la Programación para alumnos ingresantes, de carácter colaborativa y mediada por las TIC, denominada APE (Actividades Prácticas Entregables). De esta forma se introduce en el aula, una estrategia de trabajo colaborativo, que es supervisada y evaluada en cada una de sus instancias y cuyos resultados se presentan como conclusiones de la investigación.

En el Capítulo II se presenta una revisión teórica respecto a las estrategias y herramientas utilizadas en la enseñanza y aprendizaje de la programación. A continuación en el Capítulo III, se reflexiona sobre las implicancias del uso de estrategias de trabajo colaborativo mediadas por TIC en la enseñanza y aprendizaje de la programación, en el ámbito educativo de nivel universitario.

En el capítulo IV se plantea el marco metodológico que se utilizó para llevar adelante la investigación. A continuación en el capítulo V, se presenta el diseño de la propuesta de enseñanza y aprendizaje para un curso de Programación del primer año de la Lic. en Sistemas de la UNRN, Sede Atlántica, basada en actividades prácticas entregables (APE) cuya resolución se propone que se realice combinando el trabajo colaborativo y las herramientas TIC compatibles con el entorno EVEA que utiliza en la Lic. en Sistemas, como soporte a las clases presenciales.

En el capítulo VI se presentan los resultados de la implementación de la propuesta. En el capítulo VII, se exponen las conclusiones y las nuevas inquietudes que surgen del mismo. Finalmente se incluye la bibliografía consultada y los anexos del trabajo.

MARCO TEORICO

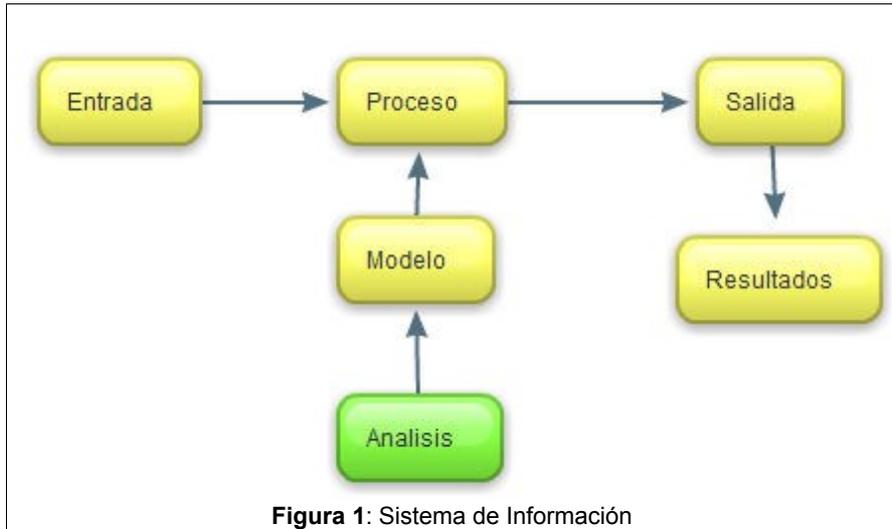
2. CAPÍTULO II. ENSEÑANZA DE LA PROGRAMACIÓN. ESTRATEGIAS Y HERRAMIENTAS

En carreras vinculadas a las Ciencias de la Computación, la enseñanza de la programación ha sido siempre un pilar fundamental y uno de los primeros cursos que deben tomar los alumnos ingresantes a estas carreras universitarias (Matthíasdóttir, Á., 2006). Así como también tema de investigación a nivel global, para el cual no existe aún una respuesta cerrada. Eileen Costelloe (2001), Lahtinen & Ala-Mutka (2005), Matthíasdóttir (2006), DiGiusti et al.(2003); Madoz et al. (2005) entre otros sostienen que la enseñanza y aprendizaje de programación en estos cursos, es una actividad intelectual compleja y dificultosa, tanto para los alumnos como para quienes llevan adelante la enseñanza; más aún cuando su impacto es muy importante en la mayoría de las asignaturas sucesivas y en el campo profesional del futuro egresado.

La tarea de programar en la formación de un profesional de sistemas, no puede limitarse solo a conocer la sintaxis y semántica de un lenguaje de programación, sino como señala Meyer (2003), su propósito principal es lograr que los alumnos desarrollen a lo largo de la carrera un conjunto de habilidades que le permitan la resolución de problemas computacionales complejos. La solución a un problema computacional, es decir un problema cuya solución pueda ser expresada en un programa entendible por una computadora, implica varios pasos: obtener un modelo del problema, diseñar un algoritmo, la implementación del mismo en un lenguaje de programación, prueba y documentación y evaluación de la solución. (Aho, et al. 1988). Para Costelloe (2001), la escritura de un programa, puede dividirse en dos etapas:

- ◆ resolución del problema
- ◆ implementación

En la primera etapa, se diseña la solución al problema. En la segunda etapa, la solución propuesta se traduce en instrucciones del lenguaje de programación elegido. El diseño de la solución a un problema consiste en determinar como hace el programa la tarea solicitada. Para esto es necesario primero interpretar el problema, modelarlo y buscar una solución algorítmica. Un algoritmo es el conjunto finito, y no ambiguo de pasos expresados en un determinado orden que, para unas condiciones iniciales, permiten resolver el problema en un tiempo finito (Joyanes Aguilar,1993). Aquí es muy importante la elección de la representación adecuada que tendrán los datos de la solución algorítmica. En resumen, un algoritmo debe ser planteado como un sistema de información (Vásquez; 2006), cuya representación se muestra en la figura 1.



Uno de los métodos que se utiliza para el proceso de diseño se basa en la estrategia “Divide y Conquista”, así la solución a un problema complejo se realiza dividiendo el problema en subproblemas y a su vez la resolución de estos subproblemas seguirá la misma estrategia. Esta forma de resolución de problemas se conoce como diseño descendente o Top-Down. De esta forma, el problema original es más fácil de comprender al dividirse en partes más simples y la solución a cada una de esas partes se denominan módulos (subprogramas). Un programa modularizado consta de un módulo principal (el módulo del nivel más alto) que invoca, a los módulos de nivel más bajo, que a su vez pueden llamar a otros módulos. Esta descomposición, permite que los diferentes módulos, puedan ser diseñados, codificados, testeados y depurados en forma independiente y luego combinarlos entre si. El diseño del algoritmo debe ser independiente del lenguaje de programación en el que se vaya a implementar posteriormente, así como también de la máquina sobre la que se ejecute. De esta forma el lenguaje de programación como la máquina son solo medios que permiten que el algoritmo se ejecute para llevar adelante un proceso (Joyanes et al. 2005).

La segunda etapa, de implementación, implica convertir el algoritmo de la etapa de diseño, en un programa, esta actividad como señala Vásquez (2006) requiere de varios pasos de refinamiento progresivo e involucra diversas áreas de conocimiento, que los alumnos deben dominar:

- ✓ Sintaxis del lenguaje de programación elegido: los alumnos deben realizar un gran esfuerzo para lograr manipular los símbolos del lenguaje de acuerdo a su sintaxis y asociarlos a una determinada semántica (Chesñevar, 2000). Como señala, Costelloe (2001), es un trabajo complejo, donde los alumnos pierden mucho tiempo en tratar de solucionar errores de sintaxis, más que el que invierten en la elaboración de la solución. Se alegran cuando obtienen un producto sintácticamente correcto, más allá de que la respuesta al problema no sea la correcta.

- ✓ Elementos de Programa: hasta las construcciones de programas más simples (secuencia, iteración, condicional) pueden resultar problemáticas para los alumnos que recién se inician en la resolución de problemas a través de la algoritmia. Es común, que los alumnos comprendan los ejemplos que se trabajan en el aula, pero al intentar resolver un nuevo problema, no logran realizar la conexión entre las situaciones problemáticas. De la misma forma, cuando se avanza en los cursos pasando por estructuras de datos simples a otras más complejas como listas vinculadas, arboles etc; los alumnos se sienten agobiados y aflora esa sensación de no estar en el curso correcto. (Costelloe, 2001)
- ✓ Ambiente de Desarrollo: para la edición, compilación y ejecución de un programa se necesita de un ambiente integrado de desarrollo (IDE). Estos ambientes, no son en general adecuados para los programadores que recién se inician, sobretodo cuando la mayoría de ellos no tienen soporte para la fase de resolución de problemas y diseño. Por otra parte, estos IDE comerciales, presentan una amplia cantidad de opciones y de información que los alumnos que recién se inician en la práctica de la programación, no pueden comprender tan fácilmente porque aún no tienen los conceptos necesarios para manipularlas (Pérez Pérez et al. 2006)
- ✓ Testing y Debbuging: Según Costelloe (2001), la falta de feedback de algunos entornos de desarrollo deja a los alumnos en un estado de confusión, cuando sus programas no funcionan como se esperaba. En este sentido, los mensajes de error que se muestran, son complejos y confunden aún más al alumno. Por otra parte, la generación de datos y planes de prueba para la solución implementada; generan una complejidad extra para los alumnos novatos.

El arte de programar involucra conocimientos sobre lenguajes y herramientas de programación, habilidades para la resolución de problemas y estrategias efectivas para el diseño e implementación (Ala-Mutka, 2006). En este sentido, en el área disciplinar de la programación, existen diferentes paradigmas que se seguirán para el diseño e implementación de la solución. Entendiéndose por paradigma al conjunto de reglas, patrones y estilos de programación que son usados por un grupo de lenguajes de programación. Siguiendo a Dasso y Funes (2012), estos se clasifican en: lógico, funcional, procedural o imperativo y el de Programación Orientada a Objetos (POO). El paradigma imperativo describe la programación como una secuencia instrucciones o comandos que cambian el estado de un programa. Dentro de este paradigma, nos encontramos con el paradigma procedural, donde los programas se dividen en módulos (funciones y procedimientos) funcionalmente independientes y que pueden o no comunicarse entre sí. Por otra parte, el flujo de ejecución del programa se controla por tres estructuras de control: secuencia, selección e iteración. Este es el paradigma más ampliamente usado en la enseñanza aprendizaje

de la programación de los primeros cursos, y el lenguaje más popular y desarrollado específicamente para la enseñanza es Pascal (Ferreira & Rojo, 2006). El paradigma funcional hace uso de funciones para el desarrollo de la solución al problema. Algunos lenguajes de programación que se apoyan en este paradigma son Lisp, Haskell y Scheme, entre otros. A través del paradigma lógico, se utiliza la lógica para describir las relaciones lógicas que existen entre los objetos y los datos, que forman parte del problema. Prolog es el lenguaje de programación más difundido en este estilo de programación. Por último el paradigma de Orientación a Objetos o POO, se basa en la idea de encapsular en un todo - objeto- estado y comportamiento; permitiendo la manipulación de los objetos sólo a través de las acciones que se han definido para los mismos. Así la solución al problema, se resuelve a través de la comunicación - pasaje de mensajes entre los dichos objetos. Se habla entonces de programación orientada a mensajes. Un lenguaje que sigue este paradigma es Smalltalk.

2.1 ESTRATEGIAS UTILIZADAS

En los cursos de programación de los primeros años de las carreras vinculadas a las ciencias informáticas, se espera que los alumnos al finalizarlos, conozcan y manejen conceptos básicos de programación: estructuras de control (secuencia, selección, iteración), estructuras de datos estáticas y dinámicas (registros, arreglos, listas), y alguna técnica de programación. Una forma común de enseñanza en estos cursos, consiste en primero enseñar las bases del lenguaje de programación elegido y a partir de allí guiar a los alumnos a través de distintas estrategias para alcanzar el proceso completo que implica la programación. (Ala-Mutka, 2006)

Alania Vera et al.(2012) en una investigación sobre estrategias de enseñanza y estilos de aprendizaje de algoritmia, sostienen que la complejidad implicada en los problemas computacionales a los que se enfrentan los profesionales de sistemas en la actualidad, hace necesario definir estrategias de enseñanza que lleven a un aprendizaje significativo, es decir, la instrucción debe ayudar a los alumnos a organizar y relacionar la información nueva con el conocimiento previo que existe en su memoria y para que la instrucción sea efectiva, debe tomar como base las estructuras mentales existentes en los alumnos.

Y agregan que las investigaciones en esta materia pueden clasificarse en :

- estudios sobre inteligencia indican que ésta no es única sino múltiple y modificable
- estudios que consideran al alumno como sujeto activo y constructor de sus propios conocimientos, y
- estudios que tienen en cuenta el carácter cultural e interpersonal de la actividad de aprendizaje

Costelloe (2001) en su investigación "Teaching Programming. The State of the Art" ha realizado

una clasificación de las estrategias y herramientas que se utilizan en la enseñanza de la programación, a saber:

- Clases Magistrales y de laboratorio
- Software Visualization
- Robots
- Aprendizaje Basado en Problemas
- Aprendizaje Cognitivo

A partir del siguiente apartado, se desarrollarán cada una de las categorías propuestas por Costelloe (2001), recurriendo a otros autores que han investigado la temática para ahondar en el análisis de las mismas.

2.1.1 CLASES MAGISTRALES Y DE LABORATORIO

Para Costelloe (2001) las clases teóricas de un curso de programación tienen por objetivo, tratar temas que pueden ser reforzados con el material bibliográfico recomendado por la cátedra. Dejando la resolución de ejercicios que ponen en práctica los conceptos vistos en las teorías, para las actividades de laboratorio. En este ambiente, los alumnos se comportan, en general como receptores pasivos de la información, con mínima interacción, en particular en los casos donde las clases son numerosas. Respecto de este método tradicional de enseñanza, Boyle y sus colaboradores sostienen que “las estrategias de enseñanza tradicionales restringen las habilidades naturales que tienen los estudiantes, para la resolución de problemas” (Boyle, et. Al; 2000). Sin embargo este tipo de enseñanza presenta beneficios como:

- Proveer una experiencia de aprendizaje guiada para los alumnos novatos
- Las clases de laboratorio proveen del espacio para que los alumnos experimenten los nuevos conocimientos adquiridos.
- Las clases de teoría proveen una introducción a un área que se desarrolla a través de la ejercitación en el laboratorio. Y son un vehículo para despertar la atención en el tema.

En cuanto al enfoque pedagógico, Costelloe (2001), afirma que la combinación de clases magistrales y de laboratorio intentan moverse hacia un enfoque más constructivista, donde los alumnos en las clases de laboratorio tratarán de resolver ejercicios, haciendo uso del conocimiento adquirido en las clases de teoría, construyendo su propio conocimiento. Así la pedagogía subyacente en estos escenarios depende mucho de como el docente usa las clases de teoría y de laboratorio. El docente puede adoptar un modelo rígido en las clases de teorías con ejercitaciones poco flexibles en las prácticas de laboratorio. O por el contrario pueden adoptar un modelo más discursivo, usando una estrategia de resolución de problemas en forma colaborativa

para las clases teóricas con el soporte de clases de laboratorio auto-guiadas.

2.1.2 SOFTWARE VISUALIZATION

Consiste en representar las ideas abstractas presentes en el código de un programa en representaciones visuales que asistan al programador o usuario a comprender el artefacto observado (algoritmo, pasos de la ejecución de un programa). Los ambientes de visualización de software son desarrollados como herramientas de programación para profesionales o herramientas instruccionales para demostraciones o estudio interactivo. Según Yeheshel (2002) “las visualizaciones tiene un potencial cognitivo, que si se explota efectivamente en los ambientes de aprendizaje permiten crear una interacción fructífera entre el alumno y la visualización”. En cuanto a la motivación, que lleva a incorporar la visualización como una estrategia de enseñanza en los cursos de programación, se pueden mencionar factores que van desde el captar la atención de los alumnos a las construcción de ejercicios más sofisticados. Tom Naps (Naps et al. 1996) presentó en un reporte del Working Group of Visualization, los factores que motivan el uso de la visualización como recurso instruccional, entre los cuales se citan:

- Clarifica conceptos complejos a través del uso de imágenes.
- Provee modos alternativos de presentación
- Las herramientas de visualización pueden usarse como un enganche para captar la atención del alumno
- La visualización permite que los alumnos investiguen el problema a estudiar, modificando las entradas, prediciendo próximos pasos y de esta forma motivándolos en la experimentación activa.
- Las buenas visualizaciones pueden incrementar la comprensión en los alumnos
- Las visualizaciones ahorran tiempo al docente, evitando ilustrar manualmente las estructuras en la clase e incrementando la claridad de las presentaciones.

En la enseñanza de programación se utilizan principalmente tres tipos de visualizaciones:

- ◆ Visualización de Programas (VP). Este tipo de visualización se focaliza en el uso de representaciones gráficas estáticas o animadas y objetos textuales que permiten visualizar un programa en ejecución y sus datos. De esta forma se busca reforzar la comprensión de las diferentes partes de un programa en ejecución por parte del alumno. Las técnicas típicas de VP incluyen el resaltado del código, visualización de la pila de ejecución y la presentación de la información contenida en las variables. El nivel de abstracción presente en este tipo de visualización es muy bajo, ya que existe una representación directa del código y los datos del programa. El usuario puede realizar interacciones predefinidas

como cambiar la velocidad de ejecución o los datos de entrada del programa. La VP puede ser estática o dinámica. La VP estática visualiza las estructuras del programa y las relaciones entre los objetos del programa. La VP dinámica visualiza la ejecución de los programas. En general muestran como progresa la ejecución de un programa resaltando partes del código y visualizando los cambios de estado de las variables. A continuación se citan algunas herramientas que usan este tipo de visualización:

- BlueJ (<http://www.bluej.org>) la herramienta tiene como objetivo proveer un ambiente de instrucción fácil de usar para la enseñanza del lenguaje Java especialmente a los alumnos novatos. Por otra parte, se hace énfasis en las técnicas de interacción, dando como resultado un entorno altamente interactivo que promueve la exploración y la experimentación (Kölling et al 2003). En el año 2007 se publicó una versión del ambiente que incluye soporte para trabajo en grupo a través de un repositorio compartido. Una vez instalado el sistema en la máquina local, el alumno dispone de un ambiente dotado de todas las herramientas que necesita para el desarrollo de una aplicación en un lenguaje de POO. El entorno se focaliza en la relación existente entre el diagrama de clases y el código con la intención de facilitar la adquisición de conceptos del paradigma de orientación a objetos. De esta forma la interface se ha reducido a un nivel que reduzca al mínimo el nivel de distracción respecto a los conceptos de programación. Para Kölling (2008) el manejo de un IDE es una actividad compleja que excede a un alumno ingresante, por eso con BlueJ han tratado de mantener el sistema simple y fácil de usar, excluyendo todas las funcionalidades no esenciales, y poniendo el foco en las herramientas necesarias para las tareas básicas.
- Alice (www.alice.org), es un programa que tiene como objetivo facilitar a los programadores novatos, la construcción de animaciones gráficas en 3D. La herramienta permite que el alumno comprenda la estrategia de resolución de problemas y brinda los conceptos necesarios para desarrollar un programa. Alice está desarrollado usando el lenguaje de programación Python (www.python.org).
- Visual Da Vinci: es un lenguaje de programación que permite que los usuarios puedan programar los movimientos de un robot abstracto en una ciudad también abstracta. La ciudad es un cuadrado que contiene 100 calles horizontales y 100 avenidas verticales, como puede verse en la figura1. El robot camina de una esquina a la otra ejecutando las instrucciones especificadas en el programa. Durante la ejecución, el robot puede recoger y/o depositar dos tipos de objetos: flores y papeles. Para esto dispone con dos bolsas, una para cada tipo de objeto. Cuenta además con la posibilidad de incorporar obstáculos en las esquinas de manera de bloquear un camino, contar objetos y mostrar

resultados. El lenguaje define las instrucciones primitivas que el robot ejecuta y las construcciones necesarias que permiten una programación modular y estructurada. En Visual Da Vinci, un programa puede ser desarrollado en modo texto o visualmente. Cualquiera sea el modo seleccionado, todo lo que se hace en una representación simultánea y automáticamente se reproduce en el otro modo. La verificación sintáctica y la ejecución de los programas desarrollados se lleva a cabo sobre el código en modo texto, de manera independizar la eficiencia del lenguaje del formato visual del algoritmo (Champredonde et al, 2000). Para Madoz et al. (2005), el uso de Visual DaVinci en los cursos vinculados a la resolución de problemas y programación impartidos en la facultad de Informática de la UNLP , les ha permitido presentar problemas de muy fácil interpretación y resolución, que logran situar al alumno novato frente a las estructuras de control de una forma natural, al mismo tiempo que le permiten comprender los conceptos de dato y variable que serán necesarios en las materias de primer año.

- ◆ Animación de Programas. Jean Greyling (2009) investigador de la Universidad Nelson Mandela , de Sud Africa, entiende a la animación de programas, como una herramienta de soporte tecnológico que estimula la comprensión de los algoritmos a través de la visualización de la ejecución de los mismos. En esta categoría se encuentran por ejemplo, los programas que permiten visualizar el comportamiento de los algoritmos de ordenación (quicksort, heapsort, etc) que se utilizan comúnmente en el aprendizaje de estructuras de datos. Estos programas disponen de controles que el usuario (alumno) puede utilizar en cualquier momento de la ejecución (cantidad de elementos a ordenar, pausar la animación y/o modificar la velocidad de la misma, etc). La animación de algoritmos de ordenamiento se remonta a los años 80 con la aparición del video "Sorting Out Sorting". El mismo mostraba vistas de los datos ordenados por diferentes algoritmos con la idea de ayudar a los alumnos a comprender como trabajan los algoritmos y como pueden compararse. Diversos estudios sugieren que los mejores resultados de aprendizaje se obtienen cuando se permite que los usuarios (alumnos) participen interactivamente con la animación, más que cuando actúan como observadores pasivos de la ejecución de un algoritmo (Hundhausen, Naps, Fleischer, McNally en Greyling 2009). Greyling (2009) como parte de su investigación concluyó que no existía un modelo unificado ni comúnmente aceptado para evaluar la efectividad de los sistemas de animación de algoritmos y propuso una lista con los requerimientos que debería cumplir la animación de programas para hacerla pedagógicamente eficaz en base en los seis niveles de motivación propuestos por Naps (Naps et al. 2003) cuando definió la taxonomía de interacción de los alumnos con las visualizaciones de algoritmos. La lista incluye los siguientes requerimientos:

- ◆ R1: Permitir el control de la velocidad de la animación
- ◆ R2: Permitir retroceder la animación
- ◆ R3: Permitir que el usuario puede ingresar datos al algoritmo
- ◆ R4: Proveer consultas al alumno que le permitan predecir el comportamiento del algoritmo
- ◆ R5: Permitir control sobre los pasos de la animación
- ◆ R6: Proveer soporte para la construcción de animaciones por parte de los alumnos
- ◆ R7: Soporte para el trabajo liso (smooth motion). Por ejemplo, cuando se están visualizando grandes conjuntos de datos, los alumnos deberían ser capaces de deshabilitar la animación y visualizar los pasos discretos del algoritmo (Rößling and Naps 2002 en Greyling 2009).
- ◆ R8: Incluir características que permitan el análisis comparativo de algoritmos
- ◆ R9: Proveer múltiples vistas de un algoritmo, permitiendo ver la ejecución de los algoritmos a diferentes niveles de abstracción.
- ◆ R10: Proveer material instruccional extra, pseudo-código, códigos fuentes, material multimedia como audio o vídeo con explicaciones de los algoritmos visualizados.

Así la eficacia de un sistema de animación de algoritmos esta determinada por la capacidad que tiene el sistema de involucrar a los alumnos en un proceso de aprendizaje activo (requerimientos R1-R6), y las características del sistema que proporcionan información extra que realza la comprensión de la animación o aumentan su utilidad en el ámbito educativo (R7 - R10).

A continuación se presentan dos de los nueve sistemas de animación de algoritmos evaluados por Greyling en su investigación, identificando los requerimientos que cumplen según la lista antes mencionada:

- Brown University Algorithm Simulator and Animator II (BALSA).

Es un sistema de animación que soporta entradas dinámicas para generar la animación (R3), pero según Greyling (2009) en la documentación no se hace ninguna referencia a si el sistema permite que esta característica este directamente accesible a los alumnos. El sistema permite que se controle la velocidad de las animaciones (R1) y alternar entre las mismas (R5). Por otra parte, permite realizar comparaciones entre algoritmos (R8) y soporta múltiples vistas en simultaneo de las estructuras de datos del algoritmo (R9).

- Java And Web-based Algorithm Animation (JAWAA). Es un sistema de animación de algoritmos que utiliza lenguaje de scripting. Los objetos visuales y los comandos disponibles en JAWAA están diseñados para la animación de las operaciones de los algoritmos con soporte específico para distintas estructuras de datos como arreglos, punteros, listas enlazadas, colas y pilas. (Pierson and Rodger 1998; Akingbade, Finley,

Jackson et al. 2003). Los alumnos pueden controlar las animaciones (R1). El sistema esta implementado usando applets de Java con lo cual es posible integrarlo a la página web del cliente (R10). Los comandos provisto con el lenguaje de script pueden ejecutarse individualmente o en bloques, permitiendo que los alumnos construyan animaciones (R6).

- Programación Visual. Se define como el uso de componentes visuales (gráficos, animaciones o iconos) para la construcción del programa. El lenguaje SIVIL (Simple Visual Language) desarrollado por Masterson & Mayer (2001) en el Canisius College Buffalo, Ny; entra en esta clasificación y tiene por objetivo facilitar el aprendizaje de la programación a los alumnos novatos. El lenguaje solo utiliza imágenes e iconos para representar todas las estructuras necesarias en un programa. Sus creadores concluyen que aunque el lenguaje solo se usa a modo de preparación para la enseñanza de un lenguaje textual, los alumnos que usan SIVIL pueden comprender mejor las estructuras de control (un bucle por ejemplo) que aquellos que comienzan el aprendizaje de la programación con un lenguaje textual (ej. Java).

La visualización se utiliza entonces, para clarificar conceptos complejos y permite que los alumnos desarrollen modelos mentales de esos conceptos. Para Ben Ari (2001), “el constructivismo afirma que durante el aprendizaje, cada individuo crea estructuras cognitivas (modelos), los datos sensoriales se combinan con el conocimiento existente para crear nuevas estructuras cognitivas, que servirán de base para futuras construcciones”. En esta misma línea, Costelloe (2001), compara la visualización del software con el constructivismo, en el sentido que la primera, asiste en la construcción del conocimiento creando modelos de conceptos complejos: el alumno puede controlar la actividad y recibe respuesta en forma inmediata. En ambos casos, existe una construcción activa del conocimiento.

2.1. 3 ROBOTS

El uso de la robótica en el ámbito educativo se desarrolla con base en las teorías de desarrollo cognitivo de Piaget, que luego fueron revisadas por Seymour Papert. Un matemático y psicólogo, creador del lenguaje Logo, quien desarrollo dentro de la teoría del constructivismo una corriente llamada construccionismo, orientando su metodología a la creación de contextos de aprendizajes donde la máquina juega un rol relevante con la intención de que los niños pudieran comprender de manera natural cualquier materia de la enseñanza formal. En esta nueva corriente de aprendizaje, el alumno tiene un papel central, con un rol totalmente activo donde el conocimiento se amplía a través de la manipulación y construcción de objetos (Miglino et al, 1999; Sanchez, 2004 en

Monsalves, 2011).

La utilización de robots en la enseñanza de programación, no solo facilitan el aprendizaje de forma constructiva sino que además proveen de la posibilidad de experimentar con máquinas reales y no solo simuladas. Esto genera la posibilidad para los alumnos, de poder trabajar en un ambiente complejo haciéndolo más interesante (Linder, et al, 2001). El uso y manipulación de los robots se convierten en una estrategia más atractiva que la codificación de un programa. Los alumnos aprenden conceptos básicos de programación en forma intuitiva y lúdica, explorando instrucciones y sentencias del lenguaje para su manipulación, experimentando sus resultados en forma interactiva y mediante la observación directa del robot.

Byron Becker (2007), en su libro Java: Learning to Program with Robots, expone las ventajas que él ha encontrado a lo largo de su experiencia docente en la Universidad de Waterloo de Ontario Canada, al uso de robots para la enseñanza y aprendizaje de la programación orientada a objetos (POO). Entre ellas cita:

La visualización: en este sentido sostiene que las cualidades visuales que presentan los robots facilitan la especificación de un problema a través de imágenes y unas pocas líneas de texto. De la misma forma, simplifican la tarea de corrección (debugging) del programa, a través de la observación de los movimientos del robot (feedback visual).

La programación: Becker (2007), afirma que los programas que siguen el enfoque orientado a objetos (POO) son más fáciles de escribir cuando uno puede imaginar que harían esos objetos en el programa. En este sentido los objetos robots facilitan la tarea, ya que los comportamientos asociados a los mismos -mover, girar, levantar objetos del suelo, y ponerlos de nuevo son actividades de la vida cotidiana y resulta entonces fácil dar instrucciones a uno u otro objeto -robot.

El inicio rápido (Quick Startup): el robot de micro-mundos permite que los alumnos comiencen la programación OO en forma inmediata usando objetos reales en un ambiente real. En sentido, Becker(2007) sostiene que el uso de robots es similar a las estrategias que usan visualización, solo que los robots son más intuitivos y tienen aspectos algorítmicos más interesantes.

Desde el punto de vista pedagógico, para Becker (2007) el uso de Robots, es una forma sutil pero profunda, de comenzar con la introducción de la Programación Orientada a Objetos, haciendo uso de la pedagogía Usa, luego Codifica (Use, then Write). Para el autor, esta forma de trabajo, ayuda a los alumnos a codificar programas OO de forma correcta. Y coincide con Linder (2001), quien afirma que para que los alumnos asimilen los conceptos enseñados, la estrategia de “hacer y discutir” es muy eficiente. Teniendo en cuenta que el trabajo con robots implica la colaboración con otros alumnos, es necesario que estos puedan representar de forma más elaboradas sus ideas abstractas, a través del uso de sentencias concretas que puedan ser discutidas con sus

compañeros de clase, generando y testeando hipótesis y asimilando más información. En cuanto a los beneficios pedagógicos que ofrece el uso de robots para la enseñanza de la programación, Costelloe (2001) cita:

- Promueven el aprendizaje activo.
- Promueven la colaboración y los robots son parte de este proceso colaborativo.
- Proveen la posibilidad de experimentar con máquinas reales.
- Los alumnos pueden generar hipótesis y probarlas, obteniendo una respuesta en forma inmediata
- Promueven la creatividad y los buenos diseños.

2.1.4 APRENDIZAJE BASADO EN PROBLEMAS (ABP)

Para Barell(1999), el aprendizaje basado en problemas (ABP) es un proceso de indagación que permite resolver preguntas dudas o incertidumbres a cerca de los fenómenos complejos que se presentan en la vida. Entendiendo como problema, cualquier incertidumbre o dificultad que debe resolverse de alguna forma. Restrepo Gómez (2005) lo define como un método didáctico que se apoya en la enseñanza, denominada aprendizaje por descubrimiento y construcción. Donde, el protagonista es el propio estudiante, quien debe apropiarse del proceso de aprendizaje: buscar información, seleccionarla, organizarla e intentar resolver los problemas que se le plantean. Los alumnos trabajan en grupos pequeños, con la asistencia del docente (tutor) quien es el responsable de facilitar los recursos para resolver el problema, pero la resolución en sí depende de las elecciones que realice el grupo respecto a que necesitan aprender de las áreas relevantes. A entender de Guevara (2010), ABP es una metodología que asume el conocimiento no como algo absoluto, sino como una construcción que realiza el alumno en base a sus conocimientos previos y en las visiones globales del mundo. Respecto a sus orígenes, señala que el ABP tuvo sus primeras aplicaciones y desarrollo en la escuela de medicina en la Universidad de Case Western Reserve en los Estados Unidos, a principios de la década de 1950. Luego a principios de los años 80, la Mercer University, de los Estados Unidos adoptó un currículum con ABP y a finales de esa década, lo hace también la escuela de medicina de la Universidad de Harvard. El objetivo de esta metodología esta puesto en mejorar la calidad de la educación médica, cambiando la orientación de un currículum que se basaba en una colección de temas y exposiciones del maestro, a uno más integrado y organizado en problemas de la vida real y donde confluyen las diferentes áreas del conocimiento que se ponen en juego para dar solución al problema. En cuanto a los roles que asumen docentes y alumnos, utilizando esta metodología, puede decirse que el docente actúa como tutor dejando de lado la postura de expositor y transmisor del conocimiento. Así, su

participación debe ser activa orientando el proceso de aprendizaje, asegurándose que los alumnos progresen de manera adecuada para alcanzar los objetivos de aprendizaje. Para esto, deberá realizar consultas que promuevan el análisis y la síntesis de la información además de la reflexión crítica de cada tema. Por otra parte, deberá apoyar el desarrollo de la habilidad en los alumnos para buscar información y recursos de aprendizaje que les sirvan en su desarrollo personal y grupal (Guevara, 2010).

Restrepo Gómez (2005), en relación al rol del tutor en el grupo, sostiene que éste debe comportarse como facilitador del aprendizaje y experto en el dominio del problema. El énfasis está puesto en el trabajo del grupo, el aprendizaje auto-dirigido, la reflexión y la motivación del aprendizaje en una forma no autoritaria. Respecto a las diferencias con otras estrategias, Costelloe (2001), afirma que el trabajo en grupo es una característica que no es tenida en cuenta en los métodos tradicionales de enseñanza. En este sentido, la aplicación de este método de enseñanza en las ciencias de la computación, se ve favorecido por factores como el hecho de que la computación está dirigida por problemas y el aprendizaje debe ser continuo debido a la naturaleza cambiante de la industria informática, por otra parte el desarrollo de proyectos en grupos es predominante en el ámbito profesional.

Morales y Landa (2004) señalan que el cambio de roles que implica el uso de ABP, no solo afecta al docente; el estudiante también debe cambiar su forma de actuar, debe ser un estudiante activo, que trabaja en forma cooperativa y asumiendo la responsabilidad de su proceso de aprendizaje. Guevara (2010) sostiene que sería deseable que un alumno que participa en un proceso de enseñanza aprendizaje usando ABP presentará o estuviera dispuesto a mejorar algunas de las siguientes características:

- Motivación profunda y clara sobre la necesidad de aprendizaje.
- Disposición para trabajar en grupo.
- Tolerancia para enfrentarse a situaciones ambiguas.
- Habilidades para la interacción personal tanto intelectual como emocional.
- Desarrollo de los poderes imaginativo e intelectual.
- Habilidades para la solución de problemas.
- Habilidades de comunicación.
- Ver su campo de estudio desde una perspectiva más amplia.
- Habilidades de pensamiento crítico, reflexivo, imaginativo y sensitivo.

Respecto al uso de Aprendizaje Basado en Problemas (ABP) en la enseñanza-aprendizaje de programación, es posible mencionar algunos estudios, como el caso de Cataldi y Cabero(2006), quienes realizaron una experiencia con un grupo específico de estudiantes de un curso de

Algoritmos y Programación I de la Facultad de Ingeniería de la Universidad de Buenos Aires, aplicando ABP conjuntamente con los aprendizajes cooperativos y colaborativos, como estrategias de aprendizaje centradas en el alumno, usando tecnología informática (chat, foros). Los investigadores concluyen, que esta forma de resolución de problemas les resultó a los alumnos, el método de trabajo más efectivo, la propuesta de trabajo les pareció buena, destacando la importancia de contar con instancias de auto evaluación. También hacen referencia a que a través del diálogo de los estudiantes se pueden evidenciar algunas de las competencias genéricas, que son de gran importancia para el ámbito laboral en el que se desempeñaran los futuros egresados. Guevara (2010), presenta una experiencia de uso de ABP para la enseñanza-aprendizaje de un tema específico de programación de los primeros curso, como es la recursividad, y resume los beneficios de la aplicación de ABP en la enseñanza-aprendizaje de la temática en:

- el estudiante se hace responsable de su propio aprendizaje.
- Posibilita la ruptura de la presencia directa y constante del docente.
- Independiza al estudiante del docente. No con la intención de eliminar el contacto presencial con el docente, sino de construirlo y asumirlo de otra manera.
- Prepara al alumno para el aprendizaje adulto, necesario para el estudio permanente y la actualización constante en el ámbito profesional del futuro egresado.
- Posibilita el proceso de aprender haciendo e investigando.

En cuanto a las desventajas encontradas, menciona:

- Requiere de una adecuada planificación de las actividades, ya que la aplicación de la técnica requiere de tiempo suficiente, para el desarrollo completo del tema.
- El trabajo en grupo requiere que se establezcan claramente los roles, con la intención de igualar el trabajo de todos los integrantes
- Se hace necesario, previo al inicio de la actividad, motivar e incentivar al estudiante a que se responsabilice de su propio aprendizaje, dejando de lado la enseñanza-aprendizaje centrada en el docente.

2.1.5 APRENDIZAJE COGNITIVO

Es un modelo de aprendizaje basado en la teoría de cognición situada, que sostiene que el aprendizaje esta ligado naturalmente a actividades reales, contexto y cultura (Brown et al, 1989). Donde la atención se centra en el proceso y no sólo en los productos finales. Poniendo énfasis en la optimización del entrenamiento y la orientación a disposición de los estudiantes. Está asociado a la educación para el aprendizaje de oficios, donde el mismo se adquiere mientras se trabaja bajo la dirección de un maestro de alto nivel.(Vihavainen et. al; 2011).

El Aprendizaje Cognitivo divide el proceso de instrucción en tres etapas: modelado, andamiaje, y transición. En la etapa de modelado, el docente le presenta a los alumnos un modelo conceptual del proceso, con el que un experto realizaría la tarea en estudio. Un ejemplo de esto sería el desarrollo de una tarea de programación desde el principio al fin, donde el docente piensa en voz alta todo el tiempo, permitiendo que los alumnos conozcan las decisiones tomadas durante el proceso. Luego se pasa a la etapa de andamiaje, donde los alumnos se enfrentaran con la resolución de situaciones problemáticas (ejercicios) bajo la dirección de un instructor experimentado. Quien les dará pistas y no respuestas directas, que los orienten en la resolución del problema. Esta forma de trabajo se basa en la idea de Vigostky (1978), quien entendía que el aprendizaje es más eficiente cuando el alumno recibe la información suficiente, que le permita aumentar su capacidad para terminar la tarea. Como explica Arto Vihavainen (2011), cuando el alumno comienza a dominar por si mismo la tarea, la etapa de andamiaje se desvanece y aparece la etapa de transición.

Para Chalk (2000), el advenimiento de la web y Java, produjeron una proliferación de applets interactivos y bibliotecas de software que pueden ser usadas con fines educativos, alentando de esta forma la estrategia de enseñanza experimental. El mismo autor ha realizado estudios experimentales aplicando los recursos disponibles en la web, en la enseñanza de materias de las ciencias de la computación, que incluyen el uso de ambientes interactivos de aprendizaje basados en Web: webworlds, como por ejemplo herramientas para la realización de diagramas en la etapas de diseño y testing de la ingeniera del software. Sobre estas experiencias, Chalk (2000) concluye que “la web, el soporte colaborativo y las herramientas de modelado en conjunto, proveen el significado para que pueda desarrollarse un aprendizaje virtual en la práctica de ingeniera del software”. Esta forma de aprendizaje, según Costelloe (2001) presenta los siguientes beneficios :

- promueve en los alumnos el desarrollo de las habilidades meta-cognitivas, a través de la propia reflexión y el auto-análisis.
- permite que los alumnos con más experiencia puedan brindar asistencia cuando es necesario.
- promueve el aprendizaje y la colaboración en un contexto social.
- promueve el desarrollo de otras habilidades como la verbalización y la comunicación.

En el ámbito profesional, este enfoque es adoptado por las Metodologías Agiles para el desarrollo del software. Estas metodologías se focalizan en la generación temprana de valor y su acento en el aspecto humano del desarrollo de software. Desde sus inicios, surgió el interés de poder incorporarlas en la formación académica, con la intención de enriquecer la formación de los futuros ingenieros de software.

2.2 RESUMEN DEL CAPÍTULO

El aprendizaje de la programación es un problema con el que se enfrentan no solo los alumnos que se inician en una carrera vinculada a las ciencias informáticas, sino también los docentes. Con la intención de mejorar la enseñanza y el aprendizaje de la disciplina, se vienen llevando a adelante diversas investigaciones, que como se expone en este capítulo han dado como resultado estrategias bien diversas que van desde aquellas manejadas por tecnología como la visualización del software, el uso de robots o las herramientas de diseño hasta aquellas que involucran paradigmas de enseñanza como el aprendizaje basado en problemas, aprendizaje cognitivo o aprendizaje basado en patrones o casos. Sin embargo, todas las estrategias siguen siendo complementarias de las clases tradicionales de teoría y práctica.

En cuanto al uso de robots como estrategia para la enseñanza y el aprendizaje de la programación Monsalves (2011) referenciando a Miglino et al.(1999) y a Sanchez (2004), sostiene que en esta nueva corriente de aprendizaje, el alumno tiene un papel central, con un rol totalmente activo donde el conocimiento se amplía a través de la manipulación y construcción de objetos. La estrategia promueve el trabajo activo y la posibilidad de experimentar con máquinas reales (Costelloe, 2001). Las hipótesis se pueden probar, obteniendo una respuesta en forma inmediata (Costelloe, 2001), facilitando a través del feedback visual, la tarea de corrección del programa (Becker, 2007).

Los enfoques pedagógicos, ABP, el aprendizaje cognitivo y el aprendizaje basado en casos ponen atención en la construcción del conocimiento (Enkenberg, 2001 en Costelloe, 2001), por otra parte los desarrollos para la web, como por ejemplo WebWorlds (Chalk, 2001), facilitan el trabajo colaborativo mediado por TIC.

Es posible concluir que de las estrategias antes mencionadas, exceptuando la primera – Clases teóricas y de laboratorio - las demás se apoyan en la aplicación del constructivismo, donde el conocimiento lo construye el propio individuo, interactuando con su entorno. Para esto construye, prueba y refina representaciones cognitivas que le permiten darle sentido al contexto. Así el constructivismo tiene una gran influencia en el aprendizaje de las ciencias de la información y la computación (Jiménez Builes, et al, 2008).

3. CAPÍTULO III. TRABAJO COLABORATIVO, TIC Y LA ENSEÑANZA Y APRENDIZAJE DE LA PROGRAMACIÓN

3.1 INTRODUCCIÓN

Para poder reflexionar sobre las implicancias del uso de estrategias de trabajo colaborativo mediadas por TIC en la enseñanza y aprendizaje de la programación, en el ámbito educativo de nivel universitario, resulta necesario definir el término trabajo colaborativo. Para autores como Maldonado Pérez (2007), Vega et al.(1999), Vázquez Mariño(2011) en otros, el modelo de aprendizaje colaborativo va más allá de la idea de recoger información procesarla y adquirir nuevos conocimientos. El modelo colaborativo se apoya en el constructivismo en el sentido, que los alumnos en plena interacción con su entorno construyen su propio conocimiento. El aprendizaje se vuelve un proceso social que se construye en la interacción no solo con el docente y sus pares, sino también con el contexto y con el significado que el alumno le otorga a lo aprendido (Maldonado Pérez, 2007).

Dependiendo del ámbito de aplicación, se distinguen diferentes definiciones para trabajo colaborativo. Es posible encontrar términos como grupos de aprendizaje (learning groups), comunidades de aprendizaje (learning communities), enseñanza entre pares (peer teaching), aprendizaje cooperativo (cooperative learning), y aprendizaje colaborativo (collaborative learning) (Dillenbourg, Gros, Salinas en Zañartu Correa (2003)).

En el ámbito educativo, Pierre Dillenbourg expresa que no hay una única definición, sin embargo propone una, a la que él mismo considera “insatisfactoria” pero más abarcadora del término aprendizaje colaborativo, considerándolo como la situación en la cual una o más personas aprenden o intentan aprender algo en forma conjunta. Dejando claro que esta, es una definición parcial en el sentido que es difícil delimitar a qué se hace referencia con un o más personas (grupo). ¿Se refiere a una pareja, un pequeño grupo, una clase, una comunidad?. El autor también se cuestiona acerca del significado del término aprendizaje, significa: ¿seguir un curso, estudiar un material de un curso, resolver un problema en forma conjunta?.(1999:1).

Beltrán y Morales(2011), por su parte, consideran que el trabajo colaborativo, es una metodología de enseñanza y de realización de la actividad laboral basada en la creencia de que el aprendizaje y la actividad laboral se amplifican cuando se desarrollan habilidades de cooperación para

aprender y solucionar los problemas y situaciones que se nos presentan en el ámbito educativo o laboral en cual estemos inmersos. Y agregan: "el trabajo colaborativo surge como metodología para abordar los problemas que son cada vez más complejos, que requieren de la intervención de más personas, más especializadas que suman ideas para solucionarlos" (Beltrán y Morales, 2011:7). En cuanto al éxito de esta metodología, estos mismos autores señalan, que no depende sólo de las habilidades de cada uno de los miembros de un grupo, sino del nivel de colaboración, creatividad individual y la participación activa de cada uno de sus miembros. Así mismo consideran fundamental, la toma de decisiones apropiadas, la autonomía de acción, así como la responsabilidad establecida para alcanzar los objetivos preestablecidos.

López Segrera, 2011 en Zuñiga, 2012, afirma que la metodología de trabajo colaborativo está en sintonía con el cambio de paradigma que se plantea en el ámbito académico de nivel universitario, donde se le otorga al estudiante una mayor responsabilidad en el aprendizaje y se persigue una formación orientada al "saber hacer", al "aprender a aprender" y al desarrollo de competencias profesionales.

Para Zañartu Correa (2003), la importancia que actualmente, tiene el aprendizaje colaborativo en los ámbitos educativos y laborales, esta dado por el hecho que nos encontramos frente a un nuevo contexto – Internet - donde como indica la autora, no existen barreras ni de cultura ni de idiomas y cuyas características de instantaneidad e interactividad la hacen muy atractiva. Esta situación, posiciona a docentes y alumnos en un nuevo entorno de aprendizaje que permite conectar simultáneamente a millones de personas, independientemente de su ubicación física. Para la autora, el aprendizaje colaborativo, nace y responde a un nuevo contexto socio cultural donde se define el "cómo aprendemos" (socialmente) y "dónde aprendemos" (en red). En este mismo sentido, Maldonado Pérez (2007), manifiesta que el advenimiento de las TIC, genera que la sociedad le demande a la educación formal, que en la formación de sus estudiantes se contemple este nuevo contexto, de manera que el estudiante se forme para compartir e interactuar con otros y con la mejor calidad. Para la autora, teniendo en cuenta que la educación formal prepara a las personas para la vida en democracia, debe necesariamente aportar elementos formativos que permitan lograr que los procesos de interacción humana sean de alta calidad, poniendo énfasis en la enseñanza que recupera el valor del individuo como ser social y otorgando reconocimiento al esfuerzo comunitario (colectivo). Beltrán y Morales (2011), por su parte, afirman que en el diseño curricular de futuras generaciones debe considerar a este nuevo estudiante – definido como hombre autónomo, es decir aquél que aprende del grupo social al que pertenece pero al mismo tiempo decide, con sentido crítico y responsabilidad, tomar distancia de las convenciones sociales del entorno y de la influencia de las personas que le rodean - y que en su

mayoría están inmersos en el uso de las TIC y destinados a ser parte de grupos colaborativos por supervivencia.

3.2 ENSEÑAR Y APRENDER EN COLABORACIÓN

Maldonado Pérez (2007) expresa la importancia de reconocer el carácter social que implica el enseñar y aprender, donde el esquema convencional que posiciona al docente con el rol de enseñante y al alumno en su rol de aprendiz en forma exclusiva, está fuera de lugar. Para la autora, el aprendizaje es un proceso social, construido a través de la interacción no solo del docente con los alumnos, sino entre alumnos y teniendo en cuenta el contexto y el significado que cada uno le asigna a lo que aprende. Esta forma de aprendizaje, consiste en la generación de estructuras cognoscitivas que se crean a través de la modificación de los reflejos iniciales del recién nacido y que se van enriqueciendo a través de la interacción del individuo con el medio. A través de estas estructuras, el individuo adquiere información, usando los procesos de asimilación y acomodación de la misma. De aquí, que el proceso de aprendizaje no se basa en la memorización de la información, sino en asimilar o incorporar información a esquemas que poseen una información previa. Desde la perspectiva teórica de Vygotsky(1979), la interacción social es un factor clave para el aprendizaje y la transmisión de cultura (Maldonado Pérez, 2007: pp.265). A entender de Johnson et al.(1999), Vygotsky (1979) sostenía el carácter social del conocimiento y su construcción a partir de los esfuerzos cooperativos por aprender, entender y resolver problemas. Un concepto clave, definido por Vygostky, es el de la zona de desarrollo próximo, entendiéndola como aquella zona situada entre lo que un estudiante puede hacer solo y lo que puede lograr si trabaja guiado por un instructor o en colaboración con otros pares más avanzados. Así, la enseñanza y como consecuencia el aprendizaje, sólo tienen lugar en la zona en la que el sujeto puede desarrollar una actividad en colaboración con otro. En este sentido, Johnson et al.(1999), sostiene que a menos que los alumnos trabajen de manera cooperativa, no crecerán intelectualmente; por esto proponía, reducir al mínimo el tiempo que los alumnos pasan trabajando solos en las actividades académicas. Maldonado Pérez (2007), basándose en la teoría de Vygotsky afirma que los procesos que desarrolla un grupo en interacción serán internalizados por cada uno de sus miembros, formando de esta forma parte de su propio aparato cognoscitivo. Y destaca el espacio fundamental que ocupan el lenguajes y los procesos de comunicación en esta interacción. En cuanto al docente, la misma autora señala que es el docente el responsable de alentar, promover y crear el espacio adecuado que permita la construcción del conocimiento. Esto implica que se organice la enseñanza y el uso de de estrategias y metodologías apropiadas, que permitan la creación de nuevos espacios de interacción humana y tecnológica.

En el ámbito educativo, el trabajo colaborativo es visto por autores como Maldonado Pérez (2007) como un modelo de aprendizaje interactivo, que propicia la construcción de conocimiento en forma conjunta, de manera que los alumnos se ven en la necesidad de conjugar esfuerzos, talentos y competencias a través de transacciones que les permitan lograr los objetivos que se han establecido en forma consensuada .

Barkley et al.(2007) en su libro *Técnicas de Aprendizaje Colaborativo*, definen al trabajo colaborativo aplicado en el ámbito educativo, como aquellas actividades de aprendizaje que han sido diseñadas para que se lleven adelante en parejas o pequeños grupos interactivos. Y resaltan tres características:

- El diseño intencional de la actividad propuesta
- La colaboración: debe existir un compromiso entre los miembros del grupo para llevar adelante la tarea propuesta.
- Enseñanza significativa: los estudiantes deben incrementar o profundizar sus conocimientos a través del desarrollo de la actividad colaborativa. De esta forma los miembros del equipo deben trabajar juntos, compartiendo en forma equitativa la carga de trabajo a medida que avanzan hacia los resultados de aprendizaje previstos en la actividad colaborativa que están desarrollando.

Para Guitert y Siménez (2000) en Maldonado Pérez, (2007), el trabajo colaborativo tiene lugar cuando existe reciprocidad entre un grupo de individuos que saben diferenciar y contrastar sus puntos de vista de tal manera que llegan a generar un proceso de construcción de conocimiento. En este proceso cada individuo aprende más de lo que aprendería por sí solo, producto esto de la interacción con el grupo.

Panitz et al. (1998), sostienen que el aprendizaje colaborativo no puede limitarse a una técnica, sino que debe ser visto como una filosofía personal. Y señalan, que en todas las situaciones donde las personas se agrupan, hay una sugerencia, una forma de tratar, que resalta la habilidades y contribuciones individuales de todos los miembros del grupo. Remarcando que la premisa del aprendizaje colaborativo se basa en la construcción del consenso a través de la cooperación de los miembros del grupo, de forma totalmente opuesta a la competencia. Así quienes ponen en práctica la filosofía de aprendizaje colaborativo no solo lo hacen en el salón de clases, sino en cualquier situación de vida que implique tratar con otras personas. Panitz et al. (1998) incluyen en la definición de colaboración a la cooperación, señalando que ésta, es una estructura de interacción diseñada para facilitar el logro de una meta o producto final específico por un grupo de personas que trabajan juntas. Para estos, aprendizaje cooperativo y colaborativo no son sinónimos, sino que representan los extremos del abanico de posibilidades que se

presentan en el proceso enseñanza-aprendizaje que va de ser altamente estructurado por el profesor (aprendizaje cooperativo) hasta dejar la responsabilidad del aprendizaje principalmente en el estudiante (aprendizaje colaborativo).

Para la pedagoga Begonia Gros (2000), el aprendizaje colaborativo es un proceso en el que todos sus miembros tienen el compromiso de aprender algo juntos. Ese objetivo de aprendizaje sólo se consigue si el trabajo grupal es colaborativo. En una situación de aprendizaje colaborativo, el grupo es el responsable de decidir cómo llevar a cabo la tarea, cuáles son los procedimientos a adoptar y cómo realizar la división del trabajo. Siguiendo con lo señalado por Panitz et al.(1998), en este tipo de aprendizaje los procesos de comunicación y negociación son las claves del mismo. A lo que, Salinas (2000) señala que en los procesos de aprendizaje colaborativo es fundamental considerar la interacción que se produce entre los distintos participantes (docentes - alumnos ,alumnos- alumnos) teniendo en cuenta que el proceso persigue por un lado metas académicas y por otro mejorar las relaciones sociales.

Para Collazos et al.(2001) cuando se habla de aprendizaje colaborativo, se describe una situación en la cual se espera que ocurran formas particulares de interacción, que promuevan mecanismos de aprendizaje, pero no hay una garantía total que éstas condiciones ocurran efectivamente. Respecto a la interacción, el autor señala que el grado de interactividad entre los integrantes no es definido por la frecuencia de las interacciones, sino por la forma en que estas pueden influenciar el proceso cognitivo de aquellos. (Collazos et al. 2006). Y destaca además que en situaciones de enseñanza y aprendizaje colaborativas, el conocimiento es construido, transformado y extendido por los estudiantes con una participación activa del profesor cambiando su rol. Así su esfuerzo estará dirigido a brindar asistencia a los alumnos, de manera que estos puedan desarrollar talentos y competencias utilizando nuevos esquemas de enseñanza, convirtiéndose así, en un guía dentro del proceso de enseñanza-aprendizaje (Collazos et al. 2001). En cuanto a la supervisión del aprendizaje del grupo; Barkley et al.(2007), afirma que no le corresponde al docente realizar esa tarea, sino que su responsabilidad está en poder formar junto a los alumnos, parte activa de una comunidad que busca el saber.

Lucero (2003) define al trabajo colaborativo, como el conjunto de métodos de instrucción y entrenamiento que a través de estrategias, promueven el desarrollo de habilidades como aprendizaje y desarrollo personal y social, donde cada integrante del equipo es responsable no sólo de su aprendizaje sino del de los demás integrantes del equipo.

Por lo aquí expuesto podemos identificar los siguientes factores involucrados en un proceso de trabajo colaborativo :

- **Manejo de la autoridad:** no sigue un modelo jerarquizado, de esta forma el desafío consiste en argumentar diferentes puntos de vistas, justificar y consensuar con los demás miembros del grupo. (Maldonado, 2007)
- **Capacidad de negociación** para alcanzar el consenso, a través de procesos dialógicos (Panitz et al. 1998; Collazos et al. 2006)
- **Reciprocidad entre los integrantes:** cada miembro del equipo debe hacer sus aportes con las debidas argumentaciones, luego los mismos serán tratados al interior del grupo en forma crítica y constructiva. Para esto, es fundamental que todos los integrantes, puedan acceder a la información. (Maldonado, 2007)
- **Sentido de responsabilidad:** los miembros del grupo deben asumir responsabilidad tanto por sus aportes individuales como por los resultados que logran en conjunto. Cada miembro debe ser consciente de que no puede depender del trabajo de los otros. (Johnson et al. 1999 en Maldonado 2007). La responsabilidad grupal es conocida en el contexto del TC como interdependencia positiva, y significa que ningún integrante del grupo podrá alcanzar el éxito si el resto de los miembros no lo alcanzan. Johnson et al.(1999) compara esta situación con la que se produce en un equipo de fútbol con un mediocampista que hace pases y un delantero, si el primero no hace un buen pase, por más que el jugador que recibe la pelota sea un gran goleador le resultará difícil convertir y lo mismo a la inversa. Así los jugadores son positivamente interdependientes. Si uno falla el otro también. Para autores como Collazos et al. (2006), Maldonado (2007) entre otros, este concepto es fundamental en el trabajo colaborativo, en el sentido que promueve la organización y funcionamiento del grupo. Para Johnson et al.(1999), cuando los estudiantes comprenden que significa interdependencia positiva, se dan cuenta que el trabajo de cada integrante es fundamental para que el grupo alcance sus metas, y que cada uno tiene un aporte exclusivo, debido a la información con la que cuenta, al rol que asume y a su responsabilidad en la tarea.
- **Relaciones sociales:** Maldonado (2007), señala que los miembros del grupo para lograr los objetivos deberán procurar relaciones afectivas positivas al interior del mismo. Así el trabajo en grupos colaborativos, debe ser apropiado por sus integrantes, como lo que ella denomina: grupos de encuentros. Y resalta la necesidad de crear conciencia en los docentes acerca de la importancia del vínculo interpersonal entre los integrantes del grupo, para llevar a cabo actividades colaborativas que generen aprendizaje. Sobre este punto Collazos et al. (2001) señala que para que la colaboración sea efectiva, se hace necesario que cambien los roles de los estudiantes y de los profesores. Y define los roles del profesor en este nuevo escenario educativo: mediador cognitivo, entrenador y diseñador

instruccional. Como mediador cognitivo es el encargado de generar habilidades metacognitivas en los alumnos, de manera que los alumnos logren ser independientes y auto-dirigir su propio aprendizaje, es decir capaces de aprender a aprender y de administrar el aprendizaje. Como entrenador es el responsable de realizar actividades de enseñanza (de las unidades temáticas, habilidades sociales y trabajo en equipo) para esto deberá entre otras actividades:

- Verificar que los grupos trabajen juntos
- Verificar que la tarea este encaminada.
- Observar y proveer retroalimentación

Y como diseñador instruccional será quien defina las condiciones esenciales que den lugar al trabajo colaborativo dentro del aula.

3.2.1 DE LOS AGRUPAMIENTOS EN EL TRABAJO COLABORATIVO

Barkley et al.(2007),sostiene siguiendo la línea planteada por Johnson (1991) que los posibles agrupamientos que tienen lugar en una actividad colaborativa son los: grupos informales, formales y básicos. Los grupos informales se constituyen en forma rápida y aleatoria con la intención de que sus integrantes trabajen juntos durante un periodo de tiempo breve. Por ejemplo para responder a una pregunta o actividad que sirva de paréntesis dentro de una actividad de clase más larga. Los grupos formales, se forman para alcanzar un objetivo más complejo, como puede ser el armado de un informe o presentación y donde la duración de la actividad puede demandar varias clases y/o semanas. Los grupos básicos, son los que se mantienen a largo tiempo, por ejemplo un cuatrimestre, y su intención es conseguir un objetivo general de la materia, buscando que los integrantes del grupo trabajen en distintas tareas a lo largo del curso.

Respecto al tamaño de los grupos, Barkley et al. (2007) afirma que en las experiencias de trabajo colaborativo, los grupos suelen ser entre 2 y 6 personas. Y referenciando a Bean (1991), sostiene que el número óptimo sería 3 para los denominados grupos básicos. Sin embargo, señala que el tamaño de un grupo, está condicionado no sólo por el tipo de grupo, sino también por otros factores como el tipo de la actividad, el tiempo que demande la misma y hasta del medio sobre el que se realizarán las tareas. Y finaliza señalando que el tamaño del grupo debe ser lo suficientemente pequeño, de manera que permita la participación plena y genere confianza entre sus miembros, pero al mismo tiempo debe ser lo suficientemente amplio como para asegurar la diversidad y los recursos necesarios para llevar adelante la actividad.

3.2.2 BENEFICIOS PEDAGÓGICOS

Calzadilla (2002), afirma que a través de las estrategias de aprendizaje colaborativo es posible incluir de forma eficiente a la educación dentro del proyecto de vida y vincular el crecimiento personal con el desarrollo de un proyecto colectivo que favorezca la cohesión, promoviendo procesos dialógicos que lleven a la confrontación de diversas perspectivas y a la negociación características propias de la dinámica de todo aprendizaje que conduzca al desarrollo.

A continuación se exponen los beneficios pedagógicos que diversos autores han identificado en esta forma de enseñanza y aprendizaje:

- Desarrollo de habilidades pensamiento de alto nivel. Cuando los estudiantes trabajan en grupo colaborativamente, desarrollan habilidades de resolución de problemas a través de la formulación de sus ideas, discutiendo, retro alimentándose con las respuestas recibidas y respondiendo a las preguntas y comentarios que realizan sus pares. (Webb 1982 en Panitz & Panitz 1998).
- Promueve la interacción y familiaridad entre el estudiante y la organización educativa (escuela, universidad etc). El proceso de colaboración le permite al docente moverse entre la clase y observar cómo actúan los estudiantes. Esta observación de los grupos, le permiten al docente identificar problemas y evaluar si están alcanzando los objetivos de aprendizaje con los que fue diseñada la tarea. El docente puede interactuar con los estudiantes directamente o en pequeños grupos logrando un clima de familiaridad, re-orientandolos o promoviendo el tratamiento más profundo sobre algunos temas. Por otra parte, la observación le permite al docente, recopilar información para el armado de los grupos y/o para la elaboración de futuras actividades (Panitz & Panitz 1998, Barkley et al. 2007) .
- Incrementa la retención de los alumnos. Los alumnos que se sienten involucrados activamente en el proceso de aprendizaje, están más interesados en aprender y hacen un mayor esfuerzo por permanecer en el curso.
- Aumenta la satisfacción de los alumnos con la experiencia de aprendizaje. Los grupos efectivos logran apropiarse del proceso y esto se logra cuando los miembros del equipo están animados a trabajar juntos para lograr una meta común, definida generalmente por el grupo
- Promueve un actitud positiva frente a los problemas de la materia. Si los alumnos están entusiasmados , tienen una mirada positiva frente a los problemas de la materia y logran compartir este entusiasmo con sus pares del grupo, promoviendo de esta forma no sólo la estima individual sino también la grupal

- Desarrolla habilidades de comunicación oral. Clarificar y explicar una respuesta es una parte importante del proceso de colaboración y representa una habilidad de pensamiento de alto nivel
- Desarrolla habilidades de interacción social: a diferencia de los modelos educativos que promueven la competencia.
- Promueve la comprensión de la diversidad de criterios. Comprender la diversidad de estilos aprendizajes y habilidades que existen entre estudiantes es uno de los mayores beneficios del aprendizaje colaborativo. (Panitz & Panitz 1998)
- Promueve la responsabilidad individual. Los estudiantes trabajando colaborativamente se brindan asistencia mutua y juegan diferentes roles. El docente se convierte en un facilitador y el alumno se convierte en un participante activo.

3.2.3 APRENDIZAJE COLABORATIVO MEDIADO POR TECNOLOGÍA

Las siglas CSCL (Aprendizaje Colaborativo Soportado por Computador), hacen referencia al conjunto de métodos de instrucción y entrenamiento soportado por tecnología, así como también a las estrategias que promueven el desarrollo de las denominadas habilidades mixtas, a saber aprendizaje, desarrollo personal y social. De esta forma cada miembro de un equipo de trabajo es responsable no sólo de su propio aprendizaje sino del aprendizaje del resto del equipo (Jimenez Builes et al., 2008). Siguiendo a Lucero (2003), el aprendizaje en ambientes colaborativos, intenta promover espacios para el desarrollo de habilidades a nivel individual y grupal que tienen lugar a partir de la discusión que se genera entre los alumnos de un grupo de trabajo al momento de explorar nuevos conceptos. Y agrega, que los enfoque de Piaget y Vygotsky basados en la interacción social, encuentran aplicación en los ambientes colaborativos.

Stahl et al. (2006) en su investigación sobre los orígenes de CSCL, señala que los ambientes CSCL aparecen en los noventa como contrapartida a los productos de software que promovían el aprendizaje en forma individual. El potencial de Internet para permitir que las personas se conecten en formas innovadoras fueron el estímulo para la investigación en CSCL.

Cabrera Murcia (2004) referenciando a Barros, Mizoguchi & Verdejo (2001), afirma que el CSCL se basa en dos perspectivas del aprendizaje que son complementarias: la constructivista y la denominada teoría de la actividad (TA), que permiten representar las actividades de grupos donde la tecnología actúa como mediador en el aprendizaje. El CSCL entiende al estudiante como un agente activo, responsable de la construcción de su aprendizaje, poseedor y generador de conocimiento, así para Roschelle et al. (2001, p .6 en Cabrera Murcia 2004), CSCL es una herramienta útil para el aprendizaje desde el punto de vista constructivista. Desde la perspectiva

de la TA, Cabrera Murcia (2004) remarca que lo relevante es que la actividad interna en el individuo surge desde la actividad externa que se origina dentro de su contexto y cita los diferentes elementos que son parte activa del proceso de aprendizaje: la comunidad y las normas sociales que gobiernan la situación, la división de la tarea, los roles que tienen lugar al interior del grupo, las herramientas para trabajar, los sujetos de la actividad, el objeto de la actividad y finalmente el resultado producido por el grupo (Cole,1999 en Cabrera Murcia 2004). Begonia Gross (2004) sintetiza el aprendizaje mediado por ordenador como la actividad de aprender a colaborar y colaborar para aprender. Respecto de las herramientas tecnológicas (ordenador, software) indica que deben favorecer los procesos de interacción y de solución conjunta de los problemas. Por otra parte, sostiene que lograr la articulación de los diferentes elementos que contribuyen a la colaboración no es una tarea fácil y, resalta el hecho, que no es suficiente con poner a un grupo a interactuar para que se produzca el aprendizaje. En referencia a Lipponen (2003), Gross (2004) señala que el CSCL ha puesto el foco en analizar cómo el aprendizaje colaborativo mediado puede acentuar la interacción entre pares y el trabajo en equipo, y cómo la tecnología y la colaboración permiten la distribución del conocimiento y el compartir experiencias a través de una comunidad virtual.

Los ambientes CSCL tuvieron su origen en los ambientes colaborativos de trabajo apoyados por computador (CSCW). Por CSCW se entiende a un grupo de personas trabajando conjuntamente en un mismo ambiente de trabajo, propiciando la colaboración y asistidos por la tecnología. CSCW se utiliza a nivel organizacional, donde existe de antemano una división de las tareas que le facilitan al individuo alcanzar aprendizajes relacionados con los objetivos de la organización (Lucero, 2003). Así la diferencia entre ambos sistemas está dada por el hecho, que en el sistema CSCL se propone que los aprendices logren obtener un conocimiento en grupo, en lugar de buscar la eficiencia en el trabajo.

El CSCL está soportado por una tecnología conocida como Groupware, a la que Peter y Trudy Johnson-Lenz (en Salinas 2008) definieron como: "procesos intencionales de un grupo para alcanzar objetivos específicos" más "herramientas de software diseñadas para dar soporte y facilitar el trabajo". Según la Wikipedia, el término groupware hace referencia a los métodos y herramientas de software que facilitan el trabajo en grupo, mejorando su rendimiento, y contribuyen a que personas que están localizadas en puntos geográficos diferentes puedan trabajar a la vez, ya sea directamente o de forma anónima, a través de las redes. Muchos expertos coinciden en que los equipos son la unidad primaria de rendimiento en cualquier organización. Hoy en día existe un nuevo tipo de equipo "virtual", equipo formado por personas que se comunican electrónicamente.

Román Gravan (2004) en un artículo sobre las posibilidades formativas de las herramientas groupware, señala que esta tecnología trabaja entorno a tres niveles de intervención :compartiendo la información, valorando la información y construyendo la información. Cada nivel, constituye una fase distinta, siendo a entender del autor, la fase de construcción de la información, la más difícil de alcanzar. Por otra parte, el mismo autor, sostiene que las herramientas groupware pueden pensarse como un soporte tanto para un grupo que interactúa cara a cara, como para otro que trabaja distribuido físicamente. De igual forma, pueden utilizarse como soporte a un grupo que interactúa en tiempo real o a otro que lo hace en forma asincrónica .

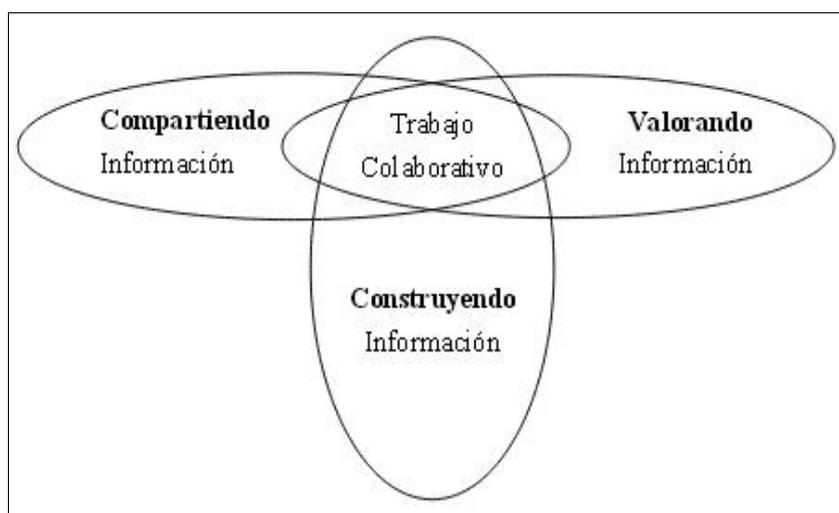


Figura 2: Trabajo Colaborativo. Niveles de intervención. *Imagen tomada de <http://tecnologiaedu.us.es/tecnoedu/images/stories/pedro/2004-articulo1-proman-revista-comunicacionypedagogia.pdf>*

En cuanto al manejo de los procesos de colaboración, Soller et al. (2005) entiende que en los ambientes distribuidos de CSCL, este está soportado por uno o más modelos computacionales de interacción de aprendizaje colaborativo, los cuales proveen representaciones funcionales soportadas por tecnología que le permiten al docente entender, explicar y predecir patrones de comportamiento de los grupos y soporte para los procesos de aprendizaje en grupo. Esto se convierte en una herramienta que permite estructurar el ambiente en el que se lleva a cabo la colaboración, o regular la interacción del alumno durante las actividades de aprendizaje. Para Stahl et al. (2006) estas formas de apoyo pedagógico provistas por los ambientes CSCL, pueden ser implementadas con mecanismos computacionales complejos, como técnicas de inteligencia artificial. Dando la posibilidad de ofrecer diferentes vistas sobre las discusiones que están teniendo los miembros de un grupo y de la información que están compartiendo. A la vez que pueden ofrecer un sistema de retro alimentación, basado posiblemente en un modelo de investigación del grupo. De igual manera pueden soportar la socialización , a través del monitoreo de patrones de interacción y brindando retro alimentación a los estudiantes. A la vez, que remarca que estos productos de software están diseñados para soportar y no reemplazar, los proceso de

colaboración que se dan entre los grupos de humanos.

Rincón et al.(2008) González Carella et al.(2007), sostienen que un entorno colaborativo que provea un soporte eficientemente para el aprendizaje colaborativo, debe estar provisto de herramientas que permitan: percepción o conciencia de grupo (awareness), comunicación, coordinación y memoria de trabajo en grupo. A través de las herramientas de percepción cada usuario puede conocer las actividades que realizan el resto de los usuarios. La comunicación entre los usuarios debe poder realizarse en forma sincrónica como asincrónica. La comunicación entre los usuarios de un entorno de programación colaborativo, debe ser simple y confiable, para garantizar el intercambio de información entre los miembros del grupo. En cuanto a la coordinación son necesarios mecanismos que permitan publicar y distribuir las actividades sin que se generen conflictos por concurrencia. Por otra parte, son necesarios elementos que permitan recuperar la información del proceso de aprendizaje. (Rincón et al.,2008, González Carella et al. 2007.

3.2.3.1 BENEFICIOS DEL USO DE CSCL

Gravan (2004) referenciando a Guadalupe,1999 y Unigarro,2001, manifiesta que en función de cómo se diseñe la formación a través de herramientas CSCL, estos ambientes preparan a los alumnos para:

- Participar en forma activa de la construcción colectiva de la información, a la vez que se genera conocimiento.
- Asumir y cumplir responsabilidades en un equipo de trabajo.
- Asistir a los demás y pedir asistencia cuando se requiera.
- Poner al servicio de los demás sus fortalezas individuales.
- Comprender las necesidades de los demás.
- Descubrir soluciones que beneficien a todos.
- Establecer objetivos, tareas, recursos, roles, etc.
- Escuchar en forma crítica y con respeto las opiniones de los demás
- Exponer sus ideas en forma argumentada.
- Aceptar las críticas razonadas de otros miembros del grupo.
- Familiarizarse con procesos democráticos

3.3 LA ENSEÑANZA COLABORATIVA DE LA PROGRAMACIÓN

La programación es vista en general como una actividad que tiende a desarrollarse en soledad y forma competitiva más que colaborativa. En el ámbito académico de nivel superior, las materias relacionadas a la programación realizan evaluaciones sobre trabajos individuales, reforzando la idea presentada. Sin embargo, como lo indica Jurado (2012), el desarrollo del software es una actividad compleja que requiere de la colaboración de grandes equipos de personas. Y agrega, que en el ámbito profesional una de las habilidades que persiguen las empresas de un desarrollador software sea su capacidad de trabajo en equipo. Sumado a esto, el auge del movimiento de software libre, muestra que es posible el desarrollo colaborativo de grandes productos de software, donde la colaboración tiene lugar entre grupos de personas que se hallan geográficamente distantes.

En este sentido, Newman (1989) señala que el objetivo principal que se debe perseguir al aplicar estrategias de enseñanza colaborativa para el aprendizaje de lenguajes de programación será la experiencia adquirida por los estudiantes y no los resultados en sí.

Para Teague & Roe (2008), la interacción constante entre pares, que tienen los estudiantes universitarios de la llamada “generación Y”, durante su tiempo libre, sea vía chat, sms, etc; se ha convertido en una estructura de apoyo tanto para su educación como en su vida social. Los autores afirman que estos estudiantes no pueden adaptarse a un ámbito universitario que demanda un estudio individual focalizado y nada interactivo. Sumado a esto, la falta de recursos tecnológicos, espacio físico, distribución geográfica entre otros, no es posible que los alumnos puedan realizar las prácticas de programación cara a cara. De allí la importancia que adquiere el uso de estrategias colaborativas mediadas por tecnología, en la enseñanza y aprendizaje de las materias vinculadas a la programación. Para Teague & Ro (2000), su aplicación, promueve un compromiso más fuerte y activo por parte de los estudiantes, alentándolos a pensar en voz alta, a verbalizar cada paso del proceso de resolución de problemas y satisfaciendo de esta forma la necesidad de interacción y soporte que demanda su generación.

Es importante hacer una distinción entre programación en equipo y programación colaborativa. La programación en equipo, siguiendo a Nosek (1998), se define como suma de los esfuerzos coordinados de programadores individuales, quienes se dividen la tarea de programación de un producto de software complejo. A diferencia de ésta, la programación colaborativa hace referencia al hecho que dos o más programadores trabajen en forma conjunta sobre el mismo algoritmo o el mismo programa.

Un tipo de programación colaborativa usada ampliamente en el ámbito laboral, es la denominada

Programación en Parejas (Pair Programming), especialmente en aquellas empresas donde utilizan Programación Extrema (eXtreme Programming, XP). La programación en parejas es un estilo de programación, donde dos programadores trabajan *lado a lado* en una misma computadora colaborando en forma continua sobre el mismo diseño, algoritmo, código o testing. Mientras uno implementa el algoritmo o arma un diseño. El otro, observa el trabajo de su compañero buscando errores. Una relación efectiva en este tipo de programación debe ser activa y con un intercambio de roles.

Este modelo de programación en parejas, es común encontrarlo en las clases prácticas de las materias de programación. Existen estudios cuantitativos y cualitativos (Williams et.al., 2001 en Bravo et. al; 2004 : 351), que indican que el uso de este modelo en las clases programación para alumnos novatos, aumentan su satisfacción, reducen la sensación de frustración compartida entre los alumnos y ayuda a disminuir los defectos de los programas. Proveer a los estudiantes de los cursos de programación de un ambiente colaborativo que imite la programación en parejas, le permite a los alumnos intercambiar roles, reflexionar sobre su propio trabajo y sobre el de su par, y compartir la experiencia de aprendizaje durante el proceso de resolución de problema y de desarrollo del programa

Esteves et al., (2006), sostiene que los ambientes colaborativos pueden ofrecer un importante soporte a los alumnos durante las actividades aprendizaje de la programación. Y agrega que la resolución de problemas a través de la colaboración promueve la reflexión, un mecanismo que estimula el proceso de aprendizaje. Para el desarrollo de una actividad grupal los alumnos necesitan comunicarse, discutir y emitir opiniones a otros miembros del grupo, alentando de esta forma una actitud de reflexión que conduce al aprendizaje. En este mismo sentido, Collazos (2001) afirma que este tipo de herramientas pueden fortalecer aspectos como el razonamiento, el auto-aprendizaje y el aprendizaje colaborativo. Jurado et al.(2012) afirman que el uso de CSCL en la enseñanza aprendizaje de la programación, permite a los estudiantes beneficiarse del conocimiento y habilidades del resto de integrantes del grupo, mejorando así sus propias destrezas.

3.3.1 HERRAMIENTAS COLABORATIVAS PARA LA ENSEÑANZA Y EL APRENDIZAJE DE LA PROGRAMACIÓN

Como se señalaba anteriormente, la enseñanza de la programación, no es una tarea fácil, ni para quienes tienen por delante la tarea de enseñar ni para los alumnos. De allí que diversos autores hayan realizado investigaciones y desarrollado herramientas que puedan utilizarse como apoyo a la enseñanza, facilitando el aprendizaje de conceptos y el desarrollo de las actividades

involucradas en la resolución de problemas usando una computadora.

A nivel profesional, las herramientas comerciales que se utilizan para el desarrollo del software, presentan una amplia cantidad de opciones y de información que los alumnos que recién se inician en la práctica de la programación, no pueden comprender tan fácilmente porque aún no tienen los conceptos necesarios para manipularlas (Pérez Pérez, et. al; 2006). Por otra parte, como afirma Collazos et al.(2001) en el ámbito educativo los entornos de desarrollo que se usan para la enseñanza-aprendizaje de la programación, tienen como objetivo brindar a los estudiantes un ambiente que les facilite las tareas relacionadas con el desarrollo del software.

En cuanto a las características de una herramienta que esté orientada tanto para el aprendizaje como para el desarrollo colaborativo, González de Rivera Fuentes y Paredes Velasco (2008) en un informe del Departamento de Lenguajes y Sistemas Informáticos I de la Universidad Rey Juan Carlos (España), señalan que deben estar incluidas características como: las actividades comunes, el entorno compartido y el espacio/tiempo. Por actividades comunes se entiende a aquellas tareas comunes que los participantes del grupo llevan a cabo; el entorno compartido nos da la posibilidad de tener informado a cada miembro del proyecto sobre el estado de éste, lo que cada miembro está trabajando, etc.; y el espacio/tiempo soporta que la interacción del grupo de trabajo se produzca en el mismo lugar y momento. En cuanto a la interacción es posible encontrar dos tipos: síncrona o asíncrona, que a su vez puede ser distribuida o centralizada.

Grosclaude et al. (2006), de la facultad de Informática de la Universidad Nacional del Comahue, señalan que una aplicación colaborativa mediada por tecnología debe estar caracterizada por:

- Telepresencia: Lograr la sensación de presencia de los participantes remotos a través de audio, video, o cualquier otro medio de representación.
- Espacio de trabajo compartido: Un espacio compartido que permita a los usuarios lograr una vista común y comprensión de los objetos o ideas, involucrados en la colaboración.
- Control de interacción: Es el medio por el cual se logra la colaboración ordenada entre los participantes remotos. Esto se consigue a través de la funcionalidad de administración del ambiente que tiene control total de los elementos que la aplicación necesita utilizar.

A continuación se describirán un conjunto de herramientas que han sido utilizadas en distintos centros educativos de nivel superior para la enseñanza-aprendizaje de la programación y que permiten la construcción colaborativa de un programa utilizando un lenguaje de alto nivel. Una revisión de las mismas se ha presentado en el Primer Congreso Argentino de la Interacción-Persona Computador@, Telecomunicaciones, Informática e Información Científica, IPCTIIC 2012¹.

¹ Lovos, Edith. Revisión de Herramientas Colaborativas para la Enseñanza de la Programación a Alumnos Novatos. En Actas del Primer Congreso Argentino de la Interacción-Persona Computador@, Telecomunicaciones, Informática e Información Científica. IPCTIIC 2012. Córdoba, 30 de noviembre 2012

3.3.1.1 COLLEGE (COLLABORATIVE EDITION, COMPILING AND EXECUTION OF PROGRAMS)

Es una herramienta creada por el grupo de investigación CHICO (Computer Human-Interaction and Collaboration) de la Escuela Superior de Informática de la Universidad de Castilla - La Mancha (España). Sus desarrolladores lo definen como un sistema colaborativo para el desarrollo de las actividades de programación en forma sincrónica, que puede ser usado más allá del aula presencial. Por otra parte, lo proponen como un medio para investigar los comportamientos que tienen lugar cuando las actividades de edición compilación y ejecución se realizan en forma grupal.

El sistema permite dos tipos de actores: profesor y alumno. El profesor es responsable de definir las sesiones de trabajo y gestionar a través de los instrumentos provistos por el sistema a los usuarios que participan de las mismas. Una sesión se define por un nombre, un tipo y un archivo con la especificación del problema a resolver por el grupo y una lista con las sesiones propuestas. Cuando los alumnos acceden al sistema, se abre la herramienta de manejo de sesión, mostrando la lista de sesiones disponibles. Algunas de las cuales, serán públicas, con libre acceso para los usuarios y otras privadas, donde es necesario ser miembro para participar de las mismas. El proceso de implementación de un algoritmo en un lenguaje de programación, podría dividirse en tres etapas: edición y revisión del código, compilado del código fuente y ejecución del programa objeto. Estas tres etapas se corresponden con las tres áreas de trabajo (workspaces) compartidas que ofrece COLLEGE. En el workspace de Edición, un usuario edita mientras que los demás revisan el código en forma asincrónica, haciendo sugerencias al usuario que está programando o intercambiando su rol con otro usuario. En el workspace de Compilación, se compila el programa editado y revisado por los usuarios. Los errores de compilación están visibles a todos los usuarios. En el workspace de Ejecución, los participantes se ponen de acuerdo respecto a los parámetros que se van a usar para la ejecución. Así las tareas de programación, se llevan adelante siguiendo un protocolo explícito de colaboración (Bravo et al., 2007). Los alumnos en forma democrática y haciendo uso de las herramientas de coordinación provistas por el sistema, deciden cuando compilar y/o ejecutar un programa. Además de las tareas propias del dominio de la programación, COLLEGE provee soporte colaborativo, principalmente a través de una herramienta de mensajería instantánea (chat estructurado) y una herramienta para la toma de decisión. El chat estructurado, dispone de un conjunto preestablecido de actos de comunicación.

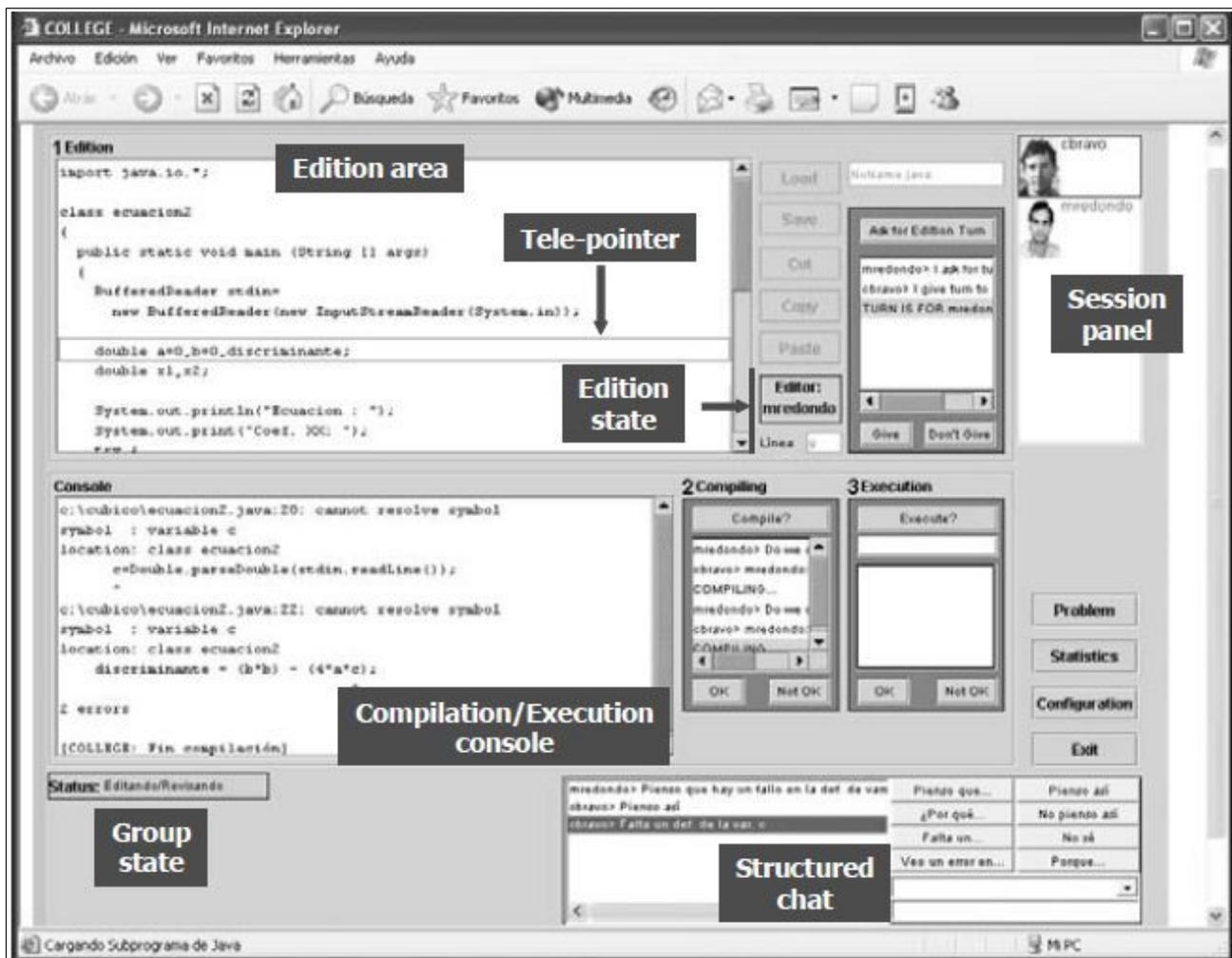


Imagen 1 - COLLEGE interface.

COLLEGE clasifica los mensajes de acuerdo a tres criterios:

- 1) Tipo de mensaje: aserciones, preguntas y respuestas.
- 2) Lugar del diálogo: mensajes iniciales y reactivos.
- 3) Tipo de adaptación: mensajes adaptables (información de las tareas del dominio) y no adaptables. Existe también la posibilidad de hacer uso de mensajes libres y/o reutilizar mensajes enviados, sin embargo lo siempre estarán disponibles con la intención de forzar al usuario a usar la combinaciones preestablecidas.

De esta forma el chat estructurado ofrece un conjunto pre-establecido de mensajes, alentando la participación de los usuarios, reduciendo la carga de escritura y focalizando al usuario en la tarea específica. (Bravo et al., 2007).

En cuanto a los mecanismos de awareness o telepresencia como lo denominan Grosclaude et al. (2006), están provistos principalmente a través de etiquetas de estado, listas de interacciones

pasadas y sonidos, que facilitan la percepción de las actividades que realizan otros usuarios en el espacio de trabajo compartido, y de esta forma la realización del trabajo en grupo. Los telepunteros se usan para mantener un puntero de edición para el usuario de turno, así poder ver que zona se está editando o qué carácter se está manipulando. También posee una lista de interacciones referida a las áreas de coordinación: cambio de turno, propuesta de compilación y propuesta de ejecución, y con ellas los mensajes de propuesta y de posible acuerdo o desacuerdo de cada uno. (Bravo et al., 2005, 2007)

En cuanto a la arquitectura del sistema, sigue el modelo cliente/servidor. El manejo de los datos y los servicios de sincronización que permiten que la colaboración sea sincrónica, están centralizados en el servidor. Mientras que los clientes distribuidos (los usuarios ejecutando el sistema) acceden al sistema a través de una página web (Bravo et al., 2007).

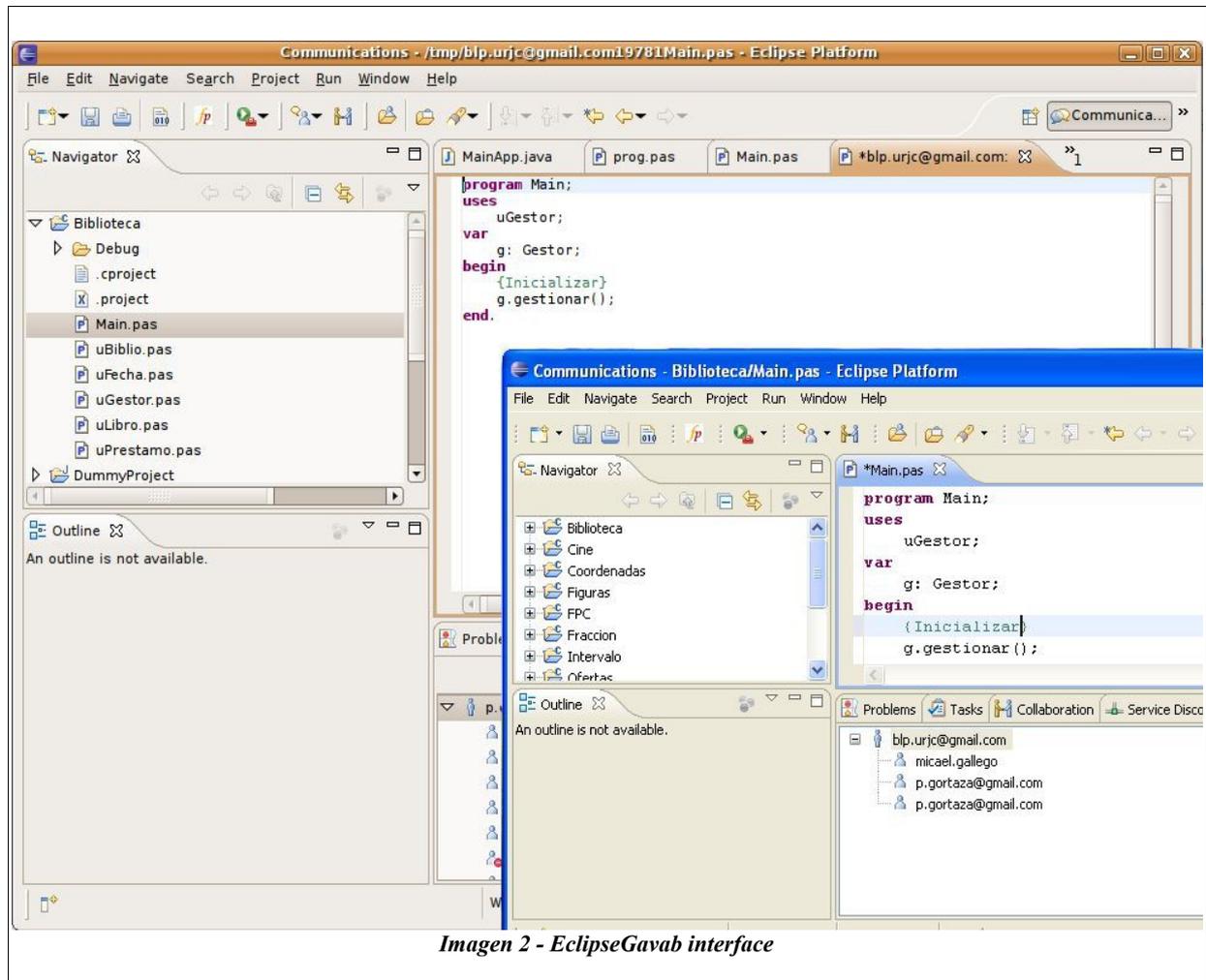
En cuanto a los lenguajes de programación que soporta, al momento de escribir este apartado, sólo se ha obtenido información sobre C y Java. COLLEGE ha sido pensado no sólo para la enseñanza-aprendizaje de la disciplina sino también para ser utilizado en la producción de software a nivel profesional.

3.3.1.2 ECLIPSEGAVAB

Es una versión personalizada del entorno de desarrollo Eclipse², que ha sido especialmente diseñada para la docencia por un grupo de investigadores de la Universidad Rey Juan Carlos dentro de un Proyecto de Innovación Educativa en el curso 2008/2009. El producto surge como necesidad de contar con herramientas que permitieran la enseñanza de la informática en tanto en instancias presenciales como a distancia. EclipseGavab permite la implementación del Aprendizaje Basado en Proyecto (ABP) a través del uso de características colaborativas como son la mensajería instantánea (chat), la edición compartida de código, y un cliente del sistema de control de versiones. Estas características permiten que los docentes puedan supervisar las producciones de los grupos en forma no presencial y que los alumnos puedan trabajar en forma colaborativa virtualmente. Los desarrolladores de EclipseGavab sostienen que la herramienta permite compartir el código usando un repositorio de código remoto, eliminando así la necesidad de enviar el código usando el correo electrónico, y facilitando enormemente el proceso de colaboración. Los cambios producidos por cada integrante (alumno) de un grupo colaborativo, son registrados en el servidor. El docente puede acceder al repositorio, y es quien se lo proporciona a los alumnos, siendo de gran utilidad para tutorizar el desarrollo de las prácticas. El docente en su rol de tutor, puede hacer sugerencias a través del chat, e incluso modificar de manera remota el código del alumno. Las modificaciones pueden incluir la inserción de

² Eclipse <http://www.eclipse.org/>

comentarios sobre aspectos que deberían modificarse o problemas que pueden presentarse debidos a ciertas decisiones tomadas en el código.



EclipseGavab soporta la programación en varios lenguajes de alto nivel, entre ellos Pascal, C y Java; los cuales son ampliamente utilizados en la enseñanza-aprendizaje de la programación. Para cada uno de estos lenguajes, EclipseGavab incluye editores con resaltado de sintaxis y marcación de errores de compilación en el propio código, depuración, y ayuda integrada. La figura 2, muestra un snapshot de la herramienta.

EclipseGavab, es multiplataforma, así puede utilizarse tanto sobre el sistema operativo Windows como en Linux; dotando de libertad al usuario respecto del sistema operativo. La herramienta puede instalarse con un único instalador, sin necesidad de instalar o configurar ninguna otra aplicación o compilador adicional. Incluye los compiladores y máquinas virtuales necesarios para los lenguajes soportados (Gallego Cortazar, 2009).

Por ser EclipseGavab una distribución de Eclipse completa, incluye todo lo necesario (plugins de Eclipse y compiladores) para el desarrollo de aplicaciones. La última versión puede ser descargada desde el sitio <http://www.sidelab.es/projects/eclipsegavab/>

Sus creadores han evaluado el uso de la herramienta en la enseñanza de asignaturas como estructuras de datos, programación funcional, programación orientada a objetos, programación concurrente y en una asignatura de aprendizaje de los lenguajes de programación C y C++ durante dos años y concluyen que los alumnos prefieren EclipseGavab a otros ambientes como TurboPascal (Gallego et al., 2009).

3.3.1.3 VIRTUAL PROGRAMMING LAB (VPL)

Es un producto de software de código abierto creado por el Departamento de Informática y Sistemas, de la Universidad de Las Palmas de Gran Canaria; que permite la gestión de prácticas de programación sobre el entorno virtual de enseñanza-aprendizaje Moodle³, de esta forma se permite la incorporación del ambiente de desarrollo de software al aula virtual de las materias donde se utiliza. Su arquitectura está compuesta de un módulo Moodle, un applet editor de código fuente y un proceso especial Linux, comúnmente llamado demonio, que permite la ejecución remota de programas de forma segura.

VPL tiene como propósitos el ahorro de tiempo y mejorar la gestión general de este tipo de actividades, tanto en los cursos de programación que se dictan en forma online como así también en cursos que usan la modalidad B-Learning, además de permitir la realización de las prácticas utilizando solo un navegador. La intención de la herramienta es facilitar el seguimiento y la orientación personalizada y continua del proceso de aprendizaje del alumno, contribuyendo de esta forma a tratar las dificultades a las que se enfrenta éste en la realización de las actividades de programación. Se busca proveer, en particular a los alumnos ingresantes de los cursos de programación de los primeros años de un entorno de desarrollo que sea simple. Sus características más destacadas son: la posibilidad de editar el código fuente y ejecutar las prácticas de forma interactiva desde el navegador, ejecutar pruebas que revisen las prácticas y analizar la similitud entre prácticas para el control del plagio (Rodríguez del Pino, et al., 2010). En la figura 3, puede visualizarse una sesión en modo profesor.

En julio de 2012, se publicó la versión 2.0 de VPL, que es compatible con la versión 2.0 de Moodle e incorpora características que permiten el trabajo en grupo. Cada grupo dispone de un repositorio compartido de entregas, así cualquier miembro del grupo puede entregar una nueva versión del programa que están realizando y todos los miembros del grupo recibirán el resultado

3 Moodle <http://moodle.com/>

de la evaluación.

A continuación se detallan las características de VPL:

- Control de entregas: permite limitar el periodo de entrega y el acceso a la tarea. Es posible limitar la cantidad de archivos que se adjuntan en las entregas y el tamaño de los mismos. Es posible restringir las operaciones de visualización, edición y subida de archivos, a las redes que se especifiquen o protegerlos a través del uso de contraseñas.
- Gestión de archivos: la edición de los archivos se realiza desde el propio navegador, para lo cual se emplea un editor de código applet java componente del sistema. Es posible establecer contenidos iniciales en los archivos de entregables. De esta forma, cuando el alumno accede al editor éste estará precargado con contenido preestablecido.
- Ejecución y Evaluación de la práctica: es posible definir scripts y programas para evaluar las actividades. Se puede ejecutar el programa con entrada y salida en una consola en el navegador. También permite imponer límites a los recursos empleados en la ejecución: tiempo, memoria, tamaño de los archivos y cantidad de procesos.

El módulo VPL necesita para ejecutar o evaluar una entrega, disponer de al menos un servidor, denominado jail server. Un servidor es *“una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de una computadora y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. Es posible que un ordenador cumpla simultáneamente las funciones de cliente y de servidor”*.⁴ Para permitir la ejecución interactiva, VPL requiere además que en la máquina del servidor Moodle se puedan abrir al menos dos puertos. El jail server requiere de un sistema operativo Linux/GNU con Kernel 2,6,18 o superior. Para la interacción completa con el sistema se necesita un navegador web con las siguientes características:

- JavaScript
- Soporte para applets Java 1,5

En julio de 2013, se anunció la publicación de VPL 2.0.3. Esta nueva versión es compatible con Moodle 2.3, 2.4 y 2.5, y agrega nuevas características entre ellas:

- Resaltado sintáctico para el lenguaje Scala⁵
- Comparadores de similitud para los lenguajes Scala y Python⁶
- Corrige y mejora el resaltado sintáctico de Matlab⁷
- Corrige y mejora el resaltado sintáctico de Scheme⁸

4 <http://es.wikipedia.org/wiki/Servidor>

5 The Scala Programming Language – Official Website, <http://www.scala-lang.org/>

6 Online Python Tutor, <http://www.onlinepythontutor.com/>

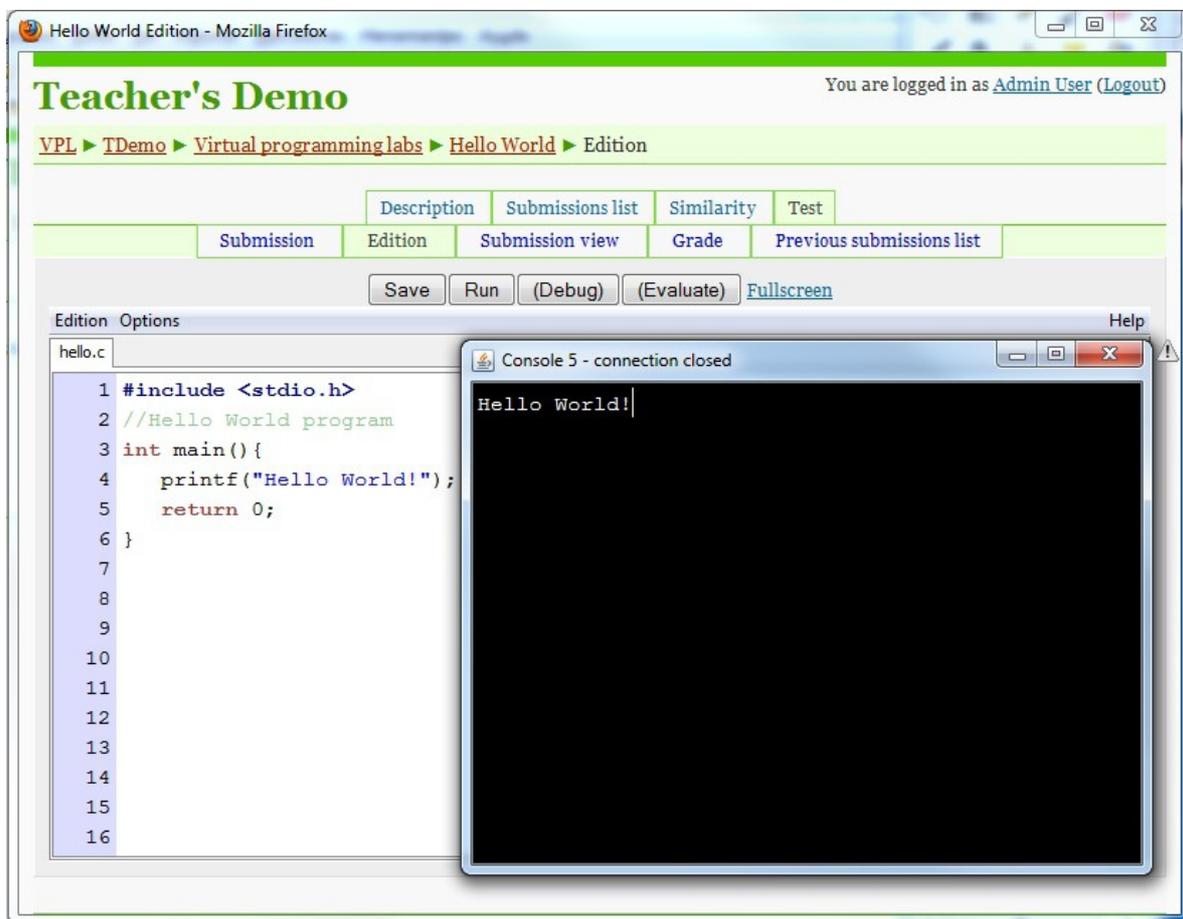
7 Getting Started with Matlab. The Math Works (2001). http://fisica.unav.es/~angel/matlab/getstart_r12.pdf

8 Teach Yourself Scheme in Fixnum Days. Dorai Sitaram (2008). <http://www.ccs.neu.edu/home/dorai/t-y-scheme/t-y-scheme.html>

- Agrega código a los script por defecto para decodificar los ficheros en base 64 (b64). Esta es una forma fácil de enviar archivos binarios a un servidor cárcel.
- Agrega traducción al alemán
- Actualización de la traducción al catalán, al japonés y al estoniano
- Corrige la comprobación de permisos en trabajo en grupo para permitir a los administradores la edición y subida de trabajos en grupo.

Esta última versión de VPL, está acompañada por la publicación de AcodeEditor 1.3.1. El applet que permite editar el código y ejecutar los programas en una consola de texto.

Todos los aplicativos necesarios para la puesta en marcha de VPL, están disponibles para su descarga en el sitio web <http://vpl.dis.ulpgc.es>.



*Imagen 3 - VPL Modo Docente
(Tomada de <http://vpl.dis.ulpgc.es/>)*

Jiménez Builes et al. (2008) presentan SABATO, un ambiente instruccional que permite integrar los paradigmas de aprendizaje basado en problemas (PBL) y el aprendizaje colaborativo apoyado

en computador (CSCL) y que puede ser utilizado en la enseñanza-aprendizaje de algoritmos y programación tanto en forma presencial como a distancia, pero que no incluye herramientas que permitan la construcción colaborativa de la implementación en un lenguaje de alto nivel de un algoritmo.

A nivel local, aquí en la Patagonia Argentina, existen experiencias de nivel universitario, en el uso de laboratorios Remotos Colaborativos para la enseñanza de la programación, como la que presentan Grosclaude et al. (2006) de la Facultad de Informática de la Universidad Nacional del Comahue, con el lenguaje NQC para robots LEGO. Para lo cual han creado un prototipo que implementa una arquitectura para laboratorios remotos con recursos físicos y virtuales. Los autores sostienen que en su experiencia, trabajar con una producción tangible como es el caso de los robots, logra que el grupo se focalice sobre un objetivo común, aportando visiones y enfoques distintos a una construcción específica. Entienden que de esta forma se logra, estimular la creatividad tanto en forma individual como colectiva, se mejora cualitativamente la producción grupal, se incrementan las oportunidades de diálogo constructivo y se reestructuran la percepción previa acerca de cómo abordar un problema.

3.4 RESUMEN DEL CAPÍTULO

López Segrera, 2011 en Zuñiga 2012 afirma que la metodología de trabajo colaborativo está en sintonía con el cambio de paradigma actual que se plantea en el ámbito académico de nivel universitario, donde se le otorga al estudiante una mayor responsabilidad en el aprendizaje y se persigue una formación orientada al “saber hacer”, al “aprender a aprender” y al desarrollo de competencias profesionales. Específicamente en el terreno del desarrollo del software, la misma es una actividad compleja que requiere de la colaboración de grandes equipos de personas. Donde una de las habilidades que persiguen las empresas de un desarrollador software es su capacidad de trabajo en equipo. Sumado a esto, el auge del movimiento de software libre, muestra que es posible el desarrollo colaborativo de grandes productos de software, donde la colaboración tiene lugar entre grupos de personas que se hallan geográficamente distantes. (Jurado; 2012). De aquí la necesidad de preparar a los alumnos en esta forma de trabajo.

La enseñanza de la programación, no es una tarea fácil, ni para quienes tienen la responsabilidad de enseñar ni para los alumnos. De allí que diversos autores hayan realizado investigaciones y desarrollado herramientas que puedan utilizarse como apoyo a la enseñanza, facilitando el aprendizaje de conceptos y el desarrollo de las actividades involucradas en la resolución de problemas usando una computadora. Así las actividades de enseñanza aprendizaje de la programación en forma colaborativa usando ambientes CSCL, le permite a los estudiantes

beneficiarse del conocimiento y habilidades del resto de integrantes del grupo, mejorando así sus propias destrezas.(Jurado; 2012)

Respecto a las herramientas que se presentan en este capítulo, como soporte a la implementación de estrategias de enseñanza-aprendizaje colaborativa; es posible concluir que EclipseGavab a pesar de estar basada en un entorno de desarrollo integrado de código abierto y ampliamente utilizado en el ámbito laboral como es Eclipse; cuenta con amplia cantidad de funcionalidades que facilitan las tareas del programador a nivel profesional, pero que como argumentan Pérez et al., (2006) Kölling (2008), pueden exceder a un estudiante que recién se inicia, por no contar aún con los conceptos necesarios para manipularlas. Por otra parte, la herramienta tiene una arquitectura client/server, de esta forma para su uso, no sólo debe instalarse en un servidor, sino que también el cliente (miembros del equipo) debe realizar una instalación. En cuanto a COLLEGE, una de las limitaciones para su uso en los curso de programación de los primeros años, es el hecho de soportar solo los lenguajes C y Java; teniendo en cuenta el hecho que el uso del lenguaje Pascal en la enseñanza de programación a alumnos novatos, sigue siendo ampliamente utilizado en las carreras de informática y afines.

VPL se presenta como una solución a los inconvenientes de instalación, sencillez de la herramienta , lenguajes de programación soportados e idiomas en los que se presenta la herramienta; especialmente si se piensa en su uso para cursos de alumnos ingresantes. Por otra parte, VPL se integra a la plataforma Moodle, de esta forma, todo el soporte a la clase sea presencial o virtual, puede encontrarse en un mismo espacio (aula virtual). La colaboración se realiza a través de las funcionalidades provistas por Moodle (mensajería, wiki, foros, etc) y del repositorio compartido de entregas.

DESARROLLO DE LA INVESTIGACION

4. CAPÍTULO IV METODOLOGÍA

En el desarrollo de este estudio se ha utilizado una metodología de investigación con un enfoque mayormente cualitativo, que integra aspectos de tipo cuantitativo. Donde, el objetivo principal consiste en *interpretar y entender* como una estrategia de enseñanza y aprendizaje basada en el trabajo colaborativo y mediada por recursos tecnológicos, denominada APE (actividades prácticas entregables), puede potenciar el aprendizaje en los alumnos de la asignatura Programación de Computadoras I, de la Lic. en Sistemas de la UNRN, Sede Atlántica. En este sentido, tal como expresa Ruth Sautu (2005) la metodología, es un conjunto de procedimientos que hacen al modo de operar en la investigación, se construye y se pone en funcionamiento desde el mismo comienzo del proceso. La aproximación cualitativa ha contribuido notablemente a la comprensión de los fenómenos sociales, elucidando los significados que estructuran el mundo social a través de la indagación de las orientaciones de los individuos, de sus comportamientos y sus implicancias para el sistema social. Por otra parte, cabe aclarar, que en la mayoría de los casos, la interpretación de cualquier relación estadística entre variables incluye e integra elementos cualitativos no explicitados en la operacionalización de las variables. Los estudios cualitativos frecuentemente integran teorizaciones y caracterizaciones inferidas de datos recolectados por medios cuantitativos. De modo, que se considera en esta investigación dejar atrás ciertas visiones excluyentes, y hacer hincapié una vez más en la posibilidad y conveniencia de la integración de métodos en la investigación social.

Por ello, se ha optado por una metodología de tipo cualitativa que busca poder captar los significados nodales que se presentan en el espacio de enseñanza y aprendizaje de una asignatura del área de lenguajes y algoritmos, a saber, Programación de Computadoras I.

Como diseño de investigación se optó por el estudio de caso, con la intención de explorar, describir, explicar, evaluar y abrir un camino para posibles transformaciones en la enseñanza y aprendizaje de la disciplina. De este modo, el relevamiento de los datos se dió en su contexto natural: la asignatura Programación de Computadoras I, de la Licenciatura en Sistemas de la Universidad Nacional de Río Negro, Sede Atlántica.

Se realizó una indagación bibliográfica para identificar los procesos de resolución de problemas en particular aquellos que pueden expresarse a través de un algoritmo. También se investigó los marcos teóricos sobre la noción de trabajo colaborativo y su aplicación mediada por TIC. Se evaluaron distintos aplicativos que posibilitan el trabajo colaborativo, y a partir de allí se seleccionó uno que se adaptara al contexto de la propuesta.

4.1 TÉCNICAS DE RECOLECCIÓN DE DATOS

La información con la que se contó para la realización de este trabajo de investigación, proviene fundamentalmente de los diferentes instrumentos utilizados, ya sea para recolectar información específica sobre las percepciones y opiniones de los participantes (alumnos y docentes), como para recolectar el resultado de la experiencia de enseñanza y aprendizaje propuesta.

4.1.1 OBSERVACIÓN

Observar científicamente es percibir activamente la realidad exterior con el propósito de obtener los datos que, previamente, han sido definidos como de interés para la investigación (Sabino, 2000). En esta investigación se han realizado diferentes tipos de observaciones, por un lado, la observación participante, donde el investigador se integro al grupo participante de la experiencia observada, esto implica que ha realizado, como señala Sabino (2000), una doble tarea, en el caso de la investigación que nos convoca, tutor o guía del proceso de enseñanza y aprendizaje y a la vez ha sido quien recogió los datos necesarios para la investigación

Durante la fase de presentación y defensa de la primera APE se llevo a cabo una observación no participante, con la intención de obtener información sobre el comportamiento de los participantes (alumnos y docentes) en esa fase de la experiencia propuesta. Así, el observador estudió al grupo pero en forma separada del mismo; observando las conductas simbólicas (verbales o escritas) de los sujetos observados, en respuesta a los distintos estímulos que recibe. (Cerde, 2007). La observación, estuvo a cargo de una docente de la carrera del área de matemáticas, que pertenece al mismo equipo de investigación que la autora del presente trabajo (en el anexo III, se presentan las notas del observador externo de la clase).

4.1.2 ENCUESTAS

En relación a las encuestas, las mismas se realizaron al total de los participantes de la experiencia y fueron encuestas anónimas a nivel individual.

Los objetivos centrales de la encuesta fueron conocer:

- Las percepciones de los alumnos en relación a sus propios procesos de aprendizaje.
- Las valoraciones de los alumnos en relación al trabajo colaborativo.
- La evaluación de los alumnos respecto a la enseñanza.

La misma encuesta se realizó en todas las etapas de la investigación. Y estuvieron accesibles desde el aula virtual de la materia sobre la que se realizó la experiencia. Como señala Meneses (2007), la encuesta es una técnica que permite obtener información sobre el tema investigado y puede encontrarse en diferentes formatos, desde un número reducido de preguntas a un conjunto

de preguntas de diferentes formatos conocido como cuestionario. En esta investigación, las encuestas utilizadas permitieron a los alumnos hacer una auto-evaluación en relación a sus propios procesos de aprendizaje, una evaluación del trabajo colaborativo y una evaluación del tutor respecto a su participación en las actividades propuestas.

Las encuestas realizadas se basan en un modelo propuesto por el Instituto Tecnológico y de Estudios Superiores de Monterrey⁹ y están disponibles en el anexo II, de esta investigación.

Las encuestas se dividieron en 3 partes a saber: auto-evaluación, evaluación del grupo de trabajo y evaluación del tutor. Y para todas se utilizaron los siguientes tipos de preguntas:

- Cerradas limitadas a un conjunto de opciones, de esta forma se facilita la categorización y codificación de las respuestas y presentan mayor confiabilidad, aunque como sostiene Meneses (2007) ponen al encuestado en situación de seleccionar una opción con la que tal vez no esté de acuerdo.
- Abiertas a modo de texto, donde el encuestado puede desarrollar la respuesta

Respecto del volumen de las encuestas, las dos primeras tienen un total de 22 preguntas y la tercera 26 preguntas. Las mismas se desarrollaron usando los recursos provistos por GoogleDocs (formulario) y luego se insertaron dentro del aula virtual de la materia donde se implementó la propuesta de enseñanza y aprendizaje de esta investigación. Esto permitió la recolección automática de los datos. Las respuestas se tomaron en su totalidad, teniendo en cuenta que la cantidad de alumnos que participaron de la experiencia.¹⁰ Para el análisis cuantitativo de los datos obtenidos con las encuestas se utilizaron las herramientas provistas por el formulario de GoogleDocs, la planilla de cálculo del paquete OpenOffice y el aplicativo InfoStat¹¹.

4.1.3 ENTREVISTAS

En relación a la entrevista como expresa, Ruiz Garzón (2012), es una técnica cualitativa que permite investigar una realidad social. De esta forma es posible obtener información sobre acontecimientos y aspectos subjetivos de las personas: creencias y actitudes, opiniones, valores o conocimiento, que de otra manera no estarían al alcance del investigador.

Existen diferentes tipos de entrevistas de acuerdo a su estructuración, dirección, finalidad y número de participantes.

En el caso particular de esta investigación, las entrevistas han sido respecto a su nivel de estructuración, de tipo no estructurada, donde se busca que las preguntas sean abiertas y sea el

9 Instituto Tecnológico y de Estudios Superiores de Monterrey. "El aprendizaje basado en problemas como técnica didáctica" <http://www.ub.edu/mercanti/abp.pdf>. 2000

10 Cabe aclarar que la población objeto de estudio es acotada pues la UNRN en donde se desarrolla la experiencia, cuenta con una baja matrícula debido a que es una Universidad de reciente creación (Esto se presenta en detalle en el Capítulo V)

11 Grupo InfoStat Di Rienzo J.A., Casanoves F., Balzarini M.G., Gonzalez L., Tablada M., Robledo C.W. InfoStat versión 2013. Grupo InfoStat, FCA, Universidad Nacional de Córdoba, Argentina. URL <http://www.infostat.com.ar>

entrevistado el que construya la respuesta. En cuanto a su dirección, se ha utilizado una entrevista dirigida, en el sentido que existía una lista de cuestiones o aspectos a ser explorados durante el desarrollo de la misma. Respecto al momento de aplicación, las entrevistas se tomaron al finalizar la experiencia, con la intención de contrastar información y concluir aspectos observados con otros instrumentos. Así mismo, algunas entrevistas se han hecho en forma grupal y otras individualmente. En el anexo digital (CD) que acompaña este documento, se puede acceder a las entrevistas realizadas a los alumnos participantes de la experiencia.

4.2 RECURSOS TIC DISPONIBLES DENTRO DEL AULA VIRTUAL DE LA MATERIA PROGRAMACIÓN I

Se utilizaron específicamente los siguientes recursos:

- ✓ Foros
- ✓ WIKI
- ✓ Virtual Programming Lab (VPL)
- ✓ Mensajería

El análisis de la información provista por estos recursos, permitió conocer el nivel de conocimiento y el uso que hacen los alumnos de los mismos. Por otra parte, a través de los mismos se pudo observar el comportamiento y el nivel de participación de todos los actores (alumnos, tutores) involucrados en el desarrollo de las APE.

5. CAPÍTULO V. DISEÑO DE LA PROPUESTA DE ENSEÑANZA Y DE APRENDIZAJE

5.1 CONTEXTO

La propuesta de enseñanza y de aprendizaje que se presenta en este capítulo, estuvo destinada a los alumnos ingresantes a la Licenciatura en Sistemas de la UNRN (Universidad Nacional de Río Negro) que cursaban Programación I. ¹²

La UNRN es una universidad nueva, que fue creada a través de la Ley Nacional 26330 promulgada el 19 de diciembre de 2007 por la Presidenta de la Nación Argentina, Dra. Cristina Fernández de Kirchner, y que inició sus actividades académicas en el año 2009. Su propuesta universitaria se extiende al territorio en el cual está inserta, a saber, la provincia de Río Negro en la región norte de la Patagonia Argentina. Río Negro dispone de un territorio cuyo tamaño es de 200.000 km², superando en relación a esta medida, a países como Holanda, Bélgica, España, Francia, Italia, Gran Bretaña o Alemania. El proyecto institucional de la UNRN, persigue una universidad que combine de forma adecuada las funciones de docencia, investigación y extensión; y que avance hacia la incorporación de enseñanzas mediadas por las tecnologías de la información y la comunicación en sus propuestas formativas.

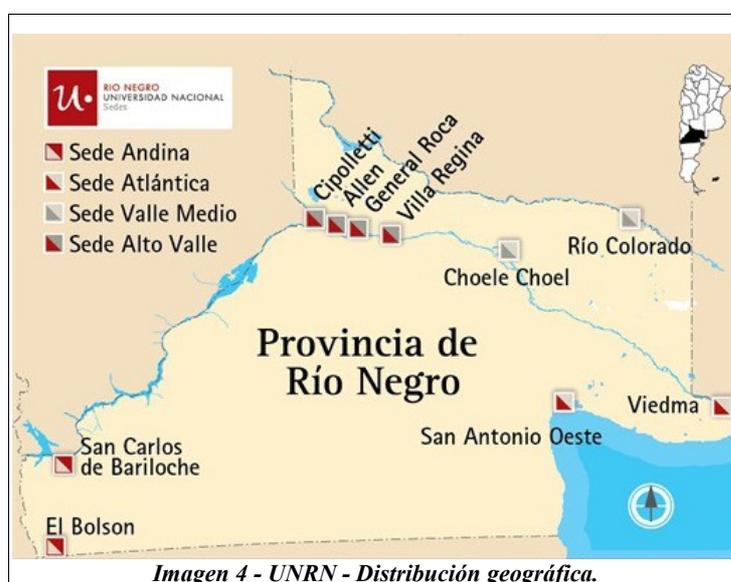
La UNRN esta organizada en tres sedes que permiten abarcar la distribución geográfica del territorio de la provincia de Río Negro, a saber Sede Atlántica abarca el Valle inferior y Costa Atlántica (Viedma, General Conesa, San Antonio Este y Region Sur), Sede Andina (San Carlos de Bariloche y El Bolsón) y Sede Alto Valle y Valle Medio que permite atender a las regiones de los valles medio y alto de la provincia y ciudades de provincias limítrofes como Neuquén y La Pampa. En la actualidad, la universidad cuenta con más de 6000 alumnos distribuidos entre las 60 carreras que coordinan las tres sedes. El rectorado de la UNRN se encuentra en la ciudad Capital de la Provincia , Viedma.

En la Sede Atlántica se dictan las carreras de Ingeniería Agronómica, Licenciatura en Administración, Licenciatura en Seguridad Ciudadana, Licenciatura en Sistemas Informáticos, la Licenciatura en Comunicación Social y Licenciatura en Trabajo Social. También se dicta el ciclo de Licenciatura de articulación en Educación Física, orientado a quienes cuentan con el título de Profesor de Educación Física.

¹² La propuesta se presentó bajo el título “Estrategias de enseñanza colaborativa para un Curso de programación de primer año de la Licenciatura en Sistemas” en el XVIII Congreso Argentino de Ciencias de la Computación, CACIC 2012. UNS – Bahía Blanca, Octubre - 2012 .

La Licenciatura en Sistemas es una carrera que se dicta en forma presencial en la Sede Atlántica específicamente en la ciudad de Viedma. La carrera comenzó a dictar sus cursos en el año 2009 y a través de un acuerdo marco con la UNLP, recibió asistencia, que permitió entre otras cosas que, docentes de la Facultad de Informática de la UNLP se desempeñarán como docentes viajeros asistiendo y formando recursos humanos propios de esta sede. La distancia geográfica que separa a ambas instituciones y las actividades de los docentes viajeros en sus lugar de origen fueron motivos para la búsqueda de estrategias que permitieran el desarrollo normal de las actividades académicas en el el primer año de la carrera. (Lovos y Gonzalez; 2011). Cuestión que en el 2011 a partir del llamado a concurso, el dictado de las materias de Programación de primer año estuvo a cargo de docente locales.

El siguiente mapa (imagen 4) muestra la ubicación geográfica de la UNRN y la distribución a lo largo de la provincia de Río Negro.



5.2 DESTINATARIOS

En las primeras cohortes (2009-2010) los alumnos que se inscribieron para cursar las materias de programación del primer año de la carrera, se dividieron en dos grupos: adultos con más de 25 años, en general con cargas de laborales y/o familiares y con alguna experiencia universitaria previa y luego jóvenes entre 18 y 25 años que habían terminado en los últimos 2 - 5 años el nivel medio.

A partir de la cohorte 2011, los alumnos que se inscriben en estas materias, son en su mayoría ingresantes a la universidad, recientemente egresados del nivel medio, cuyas edades oscilan

entre los 17 y 21 años, y que toman contacto por primera vez con la actividad de programación. En general, son residentes de la ciudad de Viedma, lugar donde se encuentra-ubicada la Sede Atlántica. También-proviene de poblaciones cercanas (San Antonio Oeste, Sierra Grande, Conesa, Carmen de Patagones, San Javier y Línea Sur de la Provincia de Río Negro).

En las últimas dos cohortes (2012, 2013) es posible observar que cada vez son más los alumnos que llegan al curso con netbooks, ya sea del programa del Estado Nacional “Conectar Igualdad” u otras. Además, resulta algo común verlos con sus teléfonos celulares accediendo a Internet, y/o escuchando música o mirando vídeos, no solo dentro del edificio donde se cursa la carrera, sino aún dentro del espacio presencial de las clases. Este acceso a las tecnologías, hace suponer que los alumnos tienen manejo de recursos como navegadores de Internet, redes sociales como facebook entre otros aplicativos de software.

5.3 CONTENIDOS CURRICULARES E INSERCIÓN EN LA CARRERA

El plan de estudio de la Lic. en Sistemas que ofrece la UNRN (Resolución 226 del año 2011) contempla en su propuesta el dictado de las siguientes asignaturas: Matemáticas I, Programación I y Organización de Computadoras en el primer cuatrimestre y Matemática II, Programación II y Arquitectura en el segundo cuatrimestre.

Programación I es una materia perteneciente al área Algoritmos y Lenguajes de Programación; la misma se dicta en forma presencial en el primer cuatrimestre del primer año con un total de 96 horas. Como se expone en el capítulo 2 dedicado a la enseñanza de la programación, la resolución de un problema usando una computadora (a través de algoritmos) requiere un pensamiento analógico pues se debe utilizar una representación de la realidad a través de un modelo.

De esta forma, la solución al problema expresada en un programa implica varios pasos: obtener un modelo del problema, diseñar un algoritmo y la implementación del mismo en un lenguaje de programación. Un programa entonces, puede definirse como el conjunto de instrucciones que la computadora tendrá que seguir para resolver el problema, y donde el responsable de proporcionarlas en el orden correcto, es el programador.

Existen en el área disciplinar de la programación diferentes paradigmas. Entendiéndose por paradigma al conjunto de reglas, patrones y estilos de programación que son usados por un grupo de lenguajes de programación. Y se clasifican en: lógico, funcional, procedural o imperativo y el de Programación Orientada a Objetos (POO).

La asignatura Programación I, tiene como objetivos generales que los alumnos puedan analizar problemas resolubles con computadora, poniendo énfasis en la modelización, la abstracción de funciones y en la descomposición funcional de los mismos, a partir de un paradigma procedural/

imperativo. Se realiza una introducción de las nociones de estructuras de datos, tipos de datos y abstracción de datos.

La abstracción es el proceso a través del cual se identifican las propiedades importantes del problema que se está modelando, ignorando los detalles irrelevantes. La abstracción es un concepto clave en las teorías de la programación y está fuertemente vinculado con los lenguajes de programación. Por un lado, los lenguajes son las herramientas con las que los programadores pueden implementar los modelos abstractos (“programas”) y por otro, los lenguajes de programación, son en sí mismos abstracciones del procesador (máquina) sobre el cual se implementa el modelo.

En un curso de programación que siga el paradigma procedural/imperativo se manejan dos tipos de abstracciones: de datos, aquella que permite modelar los datos manipulados por el programa. Y abstracción procedimental, aquella que permite modelar las distintas funcionalidades del programa. De esta forma, usando una técnica de descomposición funcional un problema complejo puede descomponerse en sub-problemas más simples.

El paradigma procedural o imperativo, permite que el programador se concentre en el procesamiento, en el algoritmo necesario para llevar a cabo el cómputo deseado.

El programa de la materia para la cohorte 2013, puede verse en el capítulo de anexos de esta tesis. Como se mencionaba anteriormente, en los inicios de la actividad académica de la Licenciatura en Sistemas, se debieron buscar estrategias que permitieran el normal desarrollo de las actividades docentes, teniendo en cuenta que los docentes no residían en la ciudad de Viedma y viajaban para el dictado de clases en forma quincenal. Es así que surge el uso de un entorno virtual de enseñanza y aprendizaje (EVEA), un programa informático que permite diseñar, estructurar y realizar diversos procesos formativos a través de la web (Casteñeda & López, 2007). El EVEA que se utiliza en la carrera es la plataforma Moodle¹³, básicamente por ser un producto de software libre y contar con soporte no sólo a través de la web sino también en otras instituciones locales.

Dentro de la plataforma Moodle, la materia Programación I tiene un aula virtual (Imagen 5) a la cual tienen acceso sólo los alumnos matriculados al curso. El aula virtual, actúa como soporte a las clases presenciales, donde los alumnos pueden encontrar el material teórico práctico, cronogramas de actividades, novedades, etc; y es un espacio extra de comunicación para todos los participantes del curso (docentes, alumnos).

El aula virtual se estructura de la siguiente manera: un espacio reservado para Novedades, donde se incluye la bienvenida al curso, el programa y cronograma de actividades de la materia y documentos relacionados con la metodología de trabajo. Luego hay seis unidades que se

¹³ Moodle. <https://moodle.org/>

corresponden con las unidades del programa de la materia. Por cada unidad existen dos carpetas Material Teórico y Material Práctico y en la unidades 2, 5 y 6 se desarrollan las actividades prácticas entregables (APE); así en estas unidades se habilitan por grupos los siguientes recursos provistos por la plataforma Moodle: foros, wiki y un laboratorio virtual (VPL) que se inserta en el EVEA.

5.4 METODOLOGÍA DE TRABAJO EN LA ASIGNATURA

El curso está dividido en clases teóricas y prácticas. En las primeras se desarrollan los conceptos teóricos previstos en el plan de estudio (resolución de problemas, estructuras de control, modularización, estructuras de datos) para esto se utilizan ejemplos prácticos que permitan la aplicación de los conceptos analizados. Se intenta que estas clases a pesar de ser de presentación de contenidos, resulten lo menos expositivas posibles, en este sentido durante el desarrollo de un tema, se plantean problemas simples que permitan la participación de los alumnos en forma cooperativa con el docente en la resolución de los mismos.

Respecto a las clases prácticas, las mismas tienen como objetivo la aplicación de los conceptos trabajados en las clases de teoría, en la resolución de problemas computacionales, a través del diseño algoritmos. En un paso siguiente estas soluciones serán implementadas en un lenguaje de programación de alto nivel tipo Pascal. El énfasis de la asignatura está puesto en la parte práctica, ya que para desarrollar la habilidad de resolver problemas usando algoritmos es fundamental el entrenamiento que se realiza a través de las situaciones que se plantean en los trabajos prácticos, ya que a través de estos los alumnos se enfrentan con situaciones en la que tienen que decidir sobre la naturaleza del problema, seleccionar una representación que ayude a resolverlo (modelo) y, monitorear sus propios pensamientos (metacognición) y estrategias de solución. (Gary Stager, 2003 en el portal Eduteka).

El programa de la materia, consta de seis unidades didácticas, cada una con su correspondiente trabajo práctico y tres Actividades Prácticas Entregables (APE) integradoras, las cuales deben ser entregadas y evaluadas para poder acceder al examen parcial. Las fechas de publicación de la APE, están establecidas en el cronograma de actividades de la materia. Por otra parte, en el programa de la materia se incluye un apartado especial que indica cómo será el desarrollo y evaluación de las mismas.

The screenshot shows the 'Aula Virtual de Programación I - Cursada 2013' interface. The top header includes the UNRN logo and the text 'Plataforma Virtual Sede Atlántica Licenciatura en Sistemas'. The user is identified as 'Edith Lovos (Salir)'. The main content area is titled 'Aula Virtual de Programación I - Cursada 2013' and contains a welcome message, contact information for Edith Lovos, and a list of documents including 'Programa 2013', 'Cronograma', and 'Cómo estudiar en la Universidad'. The left sidebar has sections for 'Navegación', 'Ajustes', 'Mensajes', 'Actividades', and 'Usuarios en línea'. The right sidebar features 'Buscar foros', 'Últimas noticias', and 'Eventos próximos'.

Imagen 5 - Plataforma Virtual Lic. en Sistemas UNRN – Sede Atlántica

5.4.1 ACREDITACIÓN DE LA ASIGNATURA

La parte práctica de la materia, se acredita mediante un examen parcial integrador, con dos instancias recuperatorias, realizados durante el cuatrimestre que dura el curso. Más el desarrollo y entrega de las 3 Actividades Prácticas Entregables (APE) integradoras. Para obtener la aprobación final de la asignatura se debe primero aprobar el examen parcial en cualquiera de sus instancias, en las fechas que publique la cátedra, y haber entregado las 3 APE y luego rendir una evaluación final teórica, en las fechas de exámenes finales fijados por el calendario académico de la UNRN. Actividades Prácticas Entregables (APE)

Las APE consisten en la resolución colaborativa en equipos de trabajo, de problemas de mediana complejidad, cuya solución es un programa computacional que se implementará en el lenguaje de programación elegido por la cátedra. En el caso de Programación I, se utiliza el lenguaje Pascal. Las consignas de trabajo que se proponen a través de la APE, incluyen la definición del problema, formas de entrega, consistente en un cronograma de actividades, fechas de previstas para cada etapa del proceso de resolución y recursos que se proponen para su desarrollo, información sobre pruebas, es decir se definen o se proporcionan los datos con los que serán puestos a prueba los programas y consideraciones especiales. En la tabla 1 se puede observar un modelo de

cronograma de actividades para una APE, en el marco de la implementación de la propuesta de enseñanza y aprendizaje que se desarrollo en la cohorte 2013 de la materia Programación I.

| Etapa | Fecha de Entrega | Recursos |
|--|----------------------------------|---|
| Debate Inicial | 30/04/13 a partir de las 14:30hs | Clase Presencial - Aula Magna |
| Análisis y Diseño | 05/05/2013 hasta las 23:30 hs | Wiki, Foro, mensajería, chat |
| Implementación | 13/05/2013 hasta las 23:30 hs | Wiki, Foro, VPL mensajería, chat |
| Presentación y Defensa | 17/05/2013 14:30 HS | Wiki, Foro, Aula Presencial |
| Evaluación del equipo de trabajo, del tutor y personal | A partir del 14/05/2013 | Página Encuesta dentro del aula virtual |

Tabla 1. Cronograma de Actividades & Recursos APE2

La estrategia de enseñanza y aprendizaje que se propone a través de esta investigación , consiste en el desarrollo de las actividades APE, combinando trabajo colaborativo y recursos TIC, con la intención de promover la participación de los alumnos y el desarrollo de competencias transversales tales como el razonamiento crítico, la capacidad de análisis, el trabajo en equipo, la autorregulación y la comunicación. Como se mencionaba en el apartado Contexto, los alumnos de Programación I, son jóvenes que tienen cierto manejo de la tecnología, así, la intención de esta propuesta de desarrollo de actividades prácticas mediada por tecnología es también que ellos las apropien como un recurso útil para construir y enriquecer su aprendizaje.

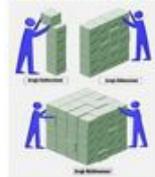
Haciendo uso de las funcionalidades provistas por el entorno Moodle (Foro, Wiki, mensajería) y del laboratorio virtual de programación (VPL¹⁴); se posibilita el desarrollo colaborativo del análisis y diseño de la solución y de la implementación del programa computacional que resuelve el problema propuesto en la APE. Las tres herramientas TIC se configuran de manera tal que cada grupo disponga de una instancia de las misma, de manera que los participantes solo puedan ver y editar las asociadas a su equipo.

En la imagen 6, se puede visualizar la estructura del EVEA Moodle para la Unidad Didáctica IV-correspondiente al tema: Datos Compuestos Indexados, donde se incluye la APE Nro 2 y las herramientas TIC que se utilizarán en cada etapa.

¹⁴ Todos los módulos necesarios para instalar Virtual Programming Lab (VPL) están disponibles en <http://vpl.dis.ulpgc.es/index.php/es>

Unidad 5

Datos Compuestos Indexados - Vectores & Matrices



- Material Teórico
- Material Práctico
- Ejemplos

Actividad Práctica Entregable - Nro 2

Consigna APE2 documento PDF

Desde aquí pueden descargar la consigna de las dos actividades propuestas.

Los grupos Pares, trabajarán con el CUADRADO MAGICO y los impares con el juego del TATETI

En el primer caso el tutor será Gerardo y el segundo Horacio . En ambos casos Edith va a estar colaborando también.

Les recomendamos hacer una lectura crítica de las consignas previa reunión con sus compañeros de grupo y7 o tutores.

El martes 30/04 a las 1430 nos reunimos en el aula magna para debatir sobre los temas de las diferentes consignas.

Éxitos!

Grupos AE2 -

Listado de grupos para la AE2

TATETI

WIKi - TATETi (AE2)

NuevoForoTateti (AE2)

Foro - Tateti (AE2)

En este espacio se propone que se trabaje sobre el análisis y diseño del problema.

Qué nos plantea el problema? C+omo podríamos dividir funcionalmente el problema? etc

TatetiLab (AE2)

Cuadrado Mágico

WIKI - Cuadrado Mágico (AE2)

Foro_Cuadrado Mágico (AE2)

CuadradoMagicoLaboratorio (AE2)

Evaluación de la APE2

Imagen 6 - Aula Virtual Estructura Unidad 5

En cuanto a la organización de los grupos de trabajo, en función de la complejidad que presentan las APE y teniendo en cuenta que desde los inicios de la carrera, en el año 2009, los alumnos inscriptos en el curso no supera los 50 en promedio, se propone que los equipos de trabajo incluyan como máximo 4 alumnos. Respecto a su conformación, en esta primer implementación se propone que para la primer APE, sean los propios alumnos quienes decidan como agruparse para trabajar, luego en las siguientes APE los equipos serán re-armados por el equipo docente de acuerdo al seguimiento realizado, teniendo en cuenta el grado de participación en la actividad y las evaluaciones que cada estudiante hace luego de entregada la actividad, respecto al desempeño de sus compañero, del tutor asignado y de su propia evaluación.

Al momento de presentación de la consigna de trabajo de la APE dentro del aula virtual se indica

por grupo con que tutor trabajará. El tutor es el responsable de hacer el seguimiento del grupo y puede ser un docente de la teoría o de la práctica de la materia.

El desarrollo de las APE involucra cinco etapas a saber:

- Debate inicial,
- Análisis y Diseño de la solución,
- Implementación,
- Presentación y Defensa,
- Evaluación (a nivel personal, del equipo de trabajo y del tutor asignado)

5.4.2 FASES DEL DESARROLLO DE LAS APE

A continuación se describen las fases propuestas para el desarrollo de las actividades prácticas entregables (APE). La imagen 7 presenta un resumen de las mismas y de los recursos usados durante el ciclo de desarrollo.

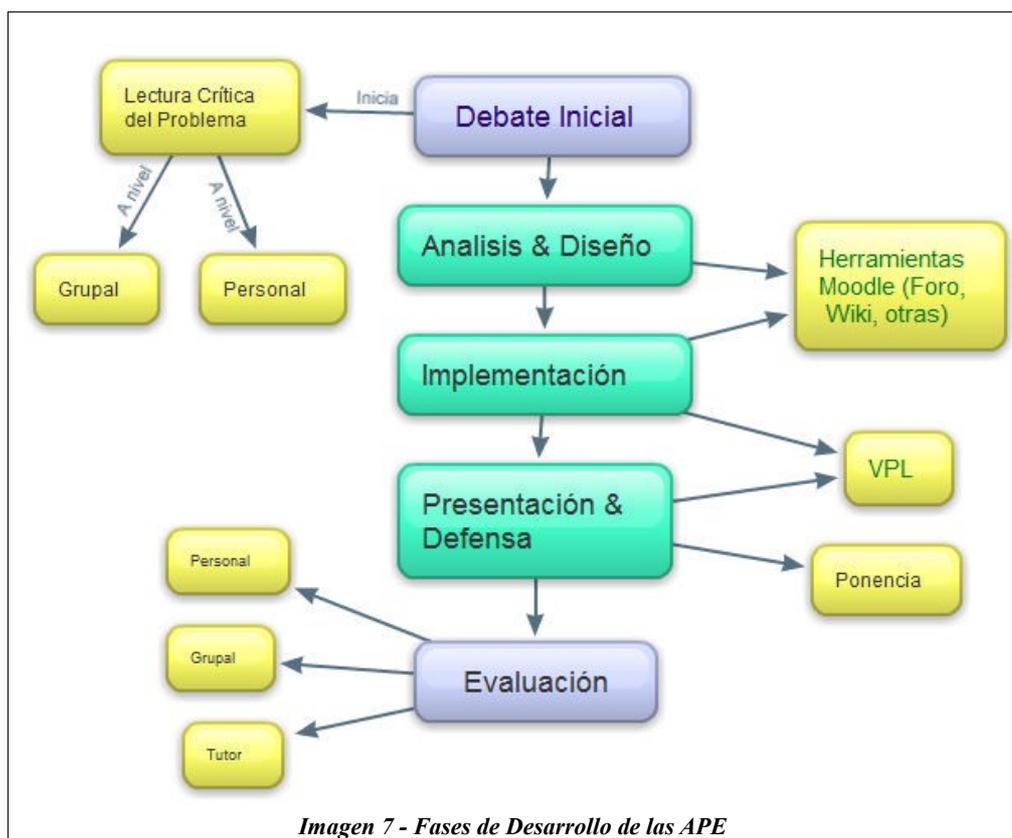


Imagen 7 - Fases de Desarrollo de las APE

Debate inicial: En la clase siguiente a la presentación de la consigna (archivo en formato pdf) en el aula virtual, se reserva una hora de la misma para que los grupos junto a los tutores asignados

puedan debatir acerca del problema. De esta forma se busca atender dudas y consultas sobre la consigna, promoviendo el análisis y discusión de las temáticas planteadas en las consignas. Entre la fecha que se habilita la consigna de la APE sobre la plataforma y la fecha prevista para el debate, los estudiantes disponen de al menos 3 días para realizar una lectura crítica del problema en forma personal y grupal de manera de llevar a la clase de debate inicial dudas y consultas sobre la consigna propuesta. Cuando se presenta la consigna de la APE se indica por grupo con que tutor trabajará.

Análisis y Diseño : en esta etapa se modelará la solución al problema. Se propone utilizar una wiki y un foro ambas herramientas provistas por el entorno Moodle. La imagen 8 permite visualizar el listado de temas abiertos en la APE3, para los grupos que trabajaron con el problema de Colas. En la imagen 9, se muestra una visión parcial del debate generado en un grupo para la APE2, tema Tablero Mágico.

| Tema | Comenzado por | Grupo | Réplicas | No leído ✓ |
|-----------------------------|--|------------------|----------|------------|
| Presentaciones |  Ezequiel Scalesi | grp5_ArceScalesi | 1 | 0 |
| APE3_PILA_PUNTO B |  JAVIER ISAURO ANAZGO | GAE2-7 | 2 | 0 |
| PILA1 |  JAVIER ISAURO ANAZGO | GAE2-7 | 5 | 0 |
| Punto B |  Ezequiel Scalesi | grp5_ArceScalesi | 1 | 0 |
| Prueba ape3 |  Ezequiel Scalesi | grp5_ArceScalesi | 1 | 0 |
| Prueba |  Ezequiel Scalesi | grp5_ArceScalesi | 3 | 0 |
| dudas con el modulo "sacar" |  Rodrigo Arce | grp5_ArceScalesi | 1 | 0 |

Imagen 8 - Listado de temas en el Foro Grupos tema Pilas

The image shows a screenshot of a forum thread with three posts. Each post has a header with a user profile picture, a title, and a timestamp. The first post is by Maila Martínez, asking a question about board placement. The second post is by Edith Lovos, providing a detailed answer with suggestions. The third post is by Sergio Choque, offering a second opinion. Each post includes a set of navigation links at the bottom right.

Pregunta respecto al ae2.
de Maila Martínez - viernes, 3 de mayo de 2013, 19:19

Hola buenas noches. Tenemos una duda, es necesario determinar una opción para la posición donde quiero colocar el nro en el tablero? o cada nro qe ingrese se colocara automaticamente en el 1º casillero del tablero? Saludos

Editar | Borrar | Responder

Re: Pregunta respecto al ae2.
de Edith Lovos - viernes, 3 de mayo de 2013, 23:17

Hola, Maila
Interesante pregunta

Una opción como uds plantean sería ubicar cada número que ingresa el usuario comenzando por el casillero 1,1 ya sea por fila o por columna.
La otra opción es pedirle al usuario que indique posición y valor
Cualquiera sea la opción que elijan, el usuario debe tener claro el criterio para la carga de datos

Cuál es la opción que más les conviene? Que piensan los demás compañeros del equipo?
Seguimos en contacto.

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

Re: Pregunta respecto al ae2.
de Sergio Choque - domingo, 5 de mayo de 2013, 01:22

hola buenas
para mi lo mas conveniente es que el usuario vaya indicando la posicion y despues el valor
sino que pasaria con los numeros ya cargados en el 1er caso ?

Mostrar mensaje anterior | Editar | Partir | Borrar | Responder

Imagen 9 - Foro Debate APE2 – Tema Tablero Mágico

Implementación: en esta fase se propone continuar con la wiki, el foro y el uso del laboratorio virtual de programación (VPL) integrado al entorno Moodle, que dispone de la opción de trabajo en grupo. A través de este laboratorio es posible que los grupos editen, compilen y ejecuten sus programas. Cada grupo tiene un repositorio compartido de entregas, y cualquier integrante del grupo puede entregar una nueva versión y todos los miembros de un grupo recibirán la misma evaluación. VPL es un aplicativo desarrollado por un grupo de investigadores de la La Universidad de Las Palmas de Gran Canaria, que permite la gestión de prácticas de programación integrándose con [Moodle](#), posibilitando de esta forma la entrega, edición y ejecución de prácticas de programación, además de una evaluación continua y automática de éstas. Más detalles sobre el aplicativo se explican en el capítulo 3 de esta tesis. La imagen 10 muestra, el listado de entregas subidas al VPL para la APE2 Tema Cuadrado Mágico.

Prog I ► Unidad 5 ► CuadradoMagicoLaboratorio ► Lista de entregas

Descripción **Lista de entregas** Similaridad Probar actividad

Grupos separados (AE2) Todos los participantes ▼

Selección de entregas Todas las entregas ▼

Evaluar Elegir... ▼

| | Grupo ↕ | Entregada el ↕ | Entregas ↕ | Calificación ↕ | Evaluada por ↕ | Evaluada el ↕ |
|---|---------|--------------------------------------|------------|----------------|----------------|-----------------------------------|
| 1 | GAE2-4 | lunes, 20 de mayo de 2013, 00:19 | 6 | 70 / 100 | Edith Lovos | lunes, 20 de mayo de 2013, 00:23 |
| 2 | GAE2-6 | miércoles, 15 de mayo de 2013, 19:02 | 9 | 80 / 100 | Edith Lovos | lunes, 20 de mayo de 2013, 00:28 |
| 3 | GAE2-10 | lunes, 20 de mayo de 2013, 00:13 | 2 | 40 / 100 | Edith Lovos | lunes, 20 de mayo de 2013, 00:15 |
| 4 | GAE2-14 | miércoles, 15 de mayo de 2013, 23:05 | 6 | 60 / 100 | Edith Lovos | jueves, 16 de mayo de 2013, 14:58 |

| # | Nombre / Apellido | Calificación |
|---|---|---------------|
| 1 |  Edith Lovos | 4/4 (100.00%) |

Imagen 10 - VPL – Entregas Tema Cuadrado Mágico

Presentación y defensa: en esta fase se propone la elaboración de una presentación que resume la solución al problema. Cada grupo puede subirla a su wiki y su exposición se desarrollará en una clase presencial de manera de poder compartir con el resto de los grupos las producciones realizadas.

Evaluación: La evaluación de las APE se desarrolla en tres partes: una evaluación del programa computacional a través del entorno virtual usando VPL, otra evaluación en forma presencial a modo de exposición y defensa de la solución propuesta y una tercera evaluación a modo de encuesta a través del aula virtual, que permite que los alumnos evalúen el proceso de desarrollo de la APE, evaluando su propio desempeño, el de su grupo de trabajo y el del tutor asignado. La imagen 11, muestra el aplicativo VPL durante la evaluación de la APE2 para el grupo GAE2-4. En la imagen 6, con el símbolo asociado a página web dentro de Moodle, se visualiza el vínculo a la encuesta correspondiente a la APE2.

Alicia Camilioni (1998:68), sostiene que no es posible separar la evaluación de los aprendizajes, de los procesos de enseñanza y aprendizaje que los han generado. Y remarca la necesidad de que el diseño de las estrategias de evaluación sea consistente con las propuestas de enseñanza y aprendizaje. En particular sobre los esquemas de evaluación de una propuesta de enseñanza y aprendizaje colaborativa, Collazos et al. (2001) sostiene que los esquemas de evaluación de una propuesta de enseñanza y aprendizaje colaborativa, tienen que enfocarse en los procesos y los resultados del aprendizaje y de la instrucción, y deben llevarse a cabo en los ambientes

apropiados. Para Méndez Pérez (2011) docente del equipo ABP de la facultad de psicología de la Universidad de Murcia, es responsabilidad del tutor observar los procesos que tienen lugar durante el desarrollo de la actividad, para entrenar y evaluar las competencias que se desarrollan en los mismos.

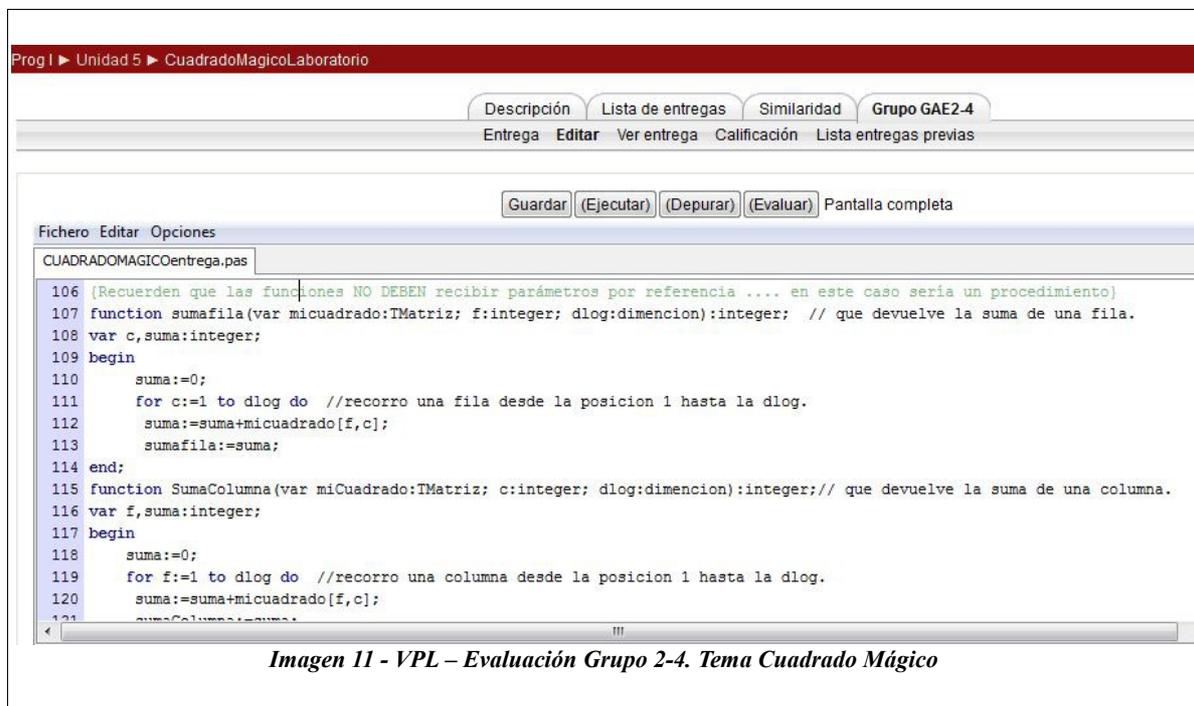


Imagen 11 - VPL – Evaluación Grupo 2-4. Tema Cuadrado Mágico

Siguiendo estos enfoques, se propone evaluar la experiencia tomando en cuenta no sólo el resultado final de las APE - el programa computacional que resuelve el problema-, sino también el proceso de aprendizaje a nivel grupal e individual que dan lugar al mismo. Para esta propuesta de trabajo, este proceso de aprendizaje estará soportado fuera del aula física, por el aula virtual de la materia y a través de diversas herramientas (wiki, foros, VPL). Las evaluaciones de las APE servirán de información para los docentes y de orientación para el alumno.

Como se explicaba al principio, una de las evaluaciones consiste en la presentación de una ponencia, realizada por los grupos de trabajo, a través de la cual explicarán y defenderán la solución propuesta para la APE. Con este tipo de evaluación se pretende propiciar la capacidad de comunicación, teniendo en cuenta que la informática es una ciencia aplicada y como futuros profesionales trabajarán en equipos multidisciplinares.

Para cada APE los alumnos deberán realizar una evaluación de sus compañeros, una auto-evaluación y una evaluación del tutor. Las mismas se realizan en forma anónima desde el aula virtual. Cabe aclarar que este enfoque de evaluación de pares, permite integrar al alumno en el proceso de evaluación del aprendizaje. Así los alumnos pueden evaluar las competencias que desarrollan sus compañeros de equipo durante el desarrollo de la actividad educativa. Los

investigadores del Departamento de Informática y Sistemas Universidad de Las Palmas de Gran Canaria, España,(Díaz Roca et al. 2012), señalan que este tipo de evaluación requiere del alumno una mayor responsabilidad y el desarrollo de habilidades que le permitan valorar el trabajo de sus compañeros de equipo.

Como se señalaba más arriba, el desarrollo de estas actividades colaborativas (APE) conviven con el dictado de las clases de teoría y práctica de la materia, así la aprobación del parcial es una condición necesaria para acreditar la materia, pero donde también se toman en cuenta los procesos que se genera con las APE y la participación de los alumnos en clase.

6. CAPITULO VI. ANÁLISIS DE RESULTADOS DE LA IMPLEMENTACIÓN

En este capítulo se trabaja sobre los resultados de la implementación de la propuesta de enseñanza y aprendizaje presentada en el capítulo V. Un análisis preliminar de los mismos se presentó en el XIX Congreso Argentino de Ciencias de la Computación¹⁵.

La cohorte 2013 del curso Programación I, se inició con un total de 64 inscriptos, de los cuales el 18,75% son mujeres y el resto son varones. Del total de inscriptos, los alumnos ingresantes corresponden al 88% y el 12% son alumnos recursantes de la cohorte 2012.

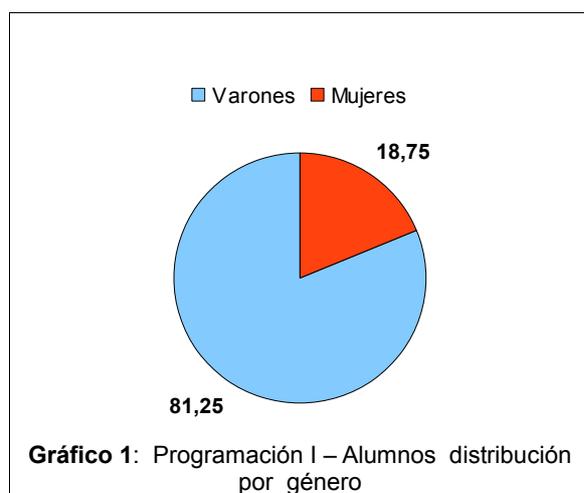


Gráfico 1: Programación I – Alumnos distribución por género

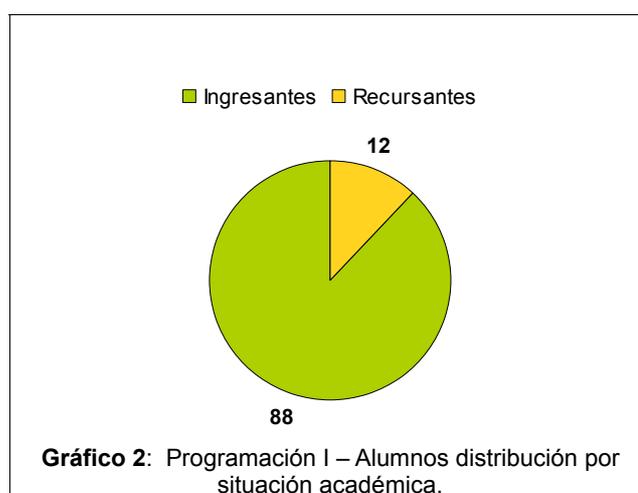


Gráfico 2: Programación I – Alumnos distribución por situación académica.

El gráfico 3 muestra la distribución de los alumnos que iniciaron la experiencia respecto a su lugar de procedencia. Se observa que la mayoría (85%) pertenecen a la ciudad de Viedma, el 13% proviene de zonas de influencia (Carmen de Patagones y San Javier) y el 2% restante corresponde a otras ciudades, cuya distancia respecto a la ubicación de la Sede Atlántica de la UNRN supera los 100km.

¹⁵ *Experiencia de utilización de Herramientas Colaborativas para la enseñanza y el aprendizaje de la Programación de Computadoras.* Edith Lovos, Alejandro González, Rodolfo Bertone. XIX Congreso Argentino de Ciencias de la Computación. CACIC 2013. Universidad CAECE. Mar del Plata. Octubre, 2013

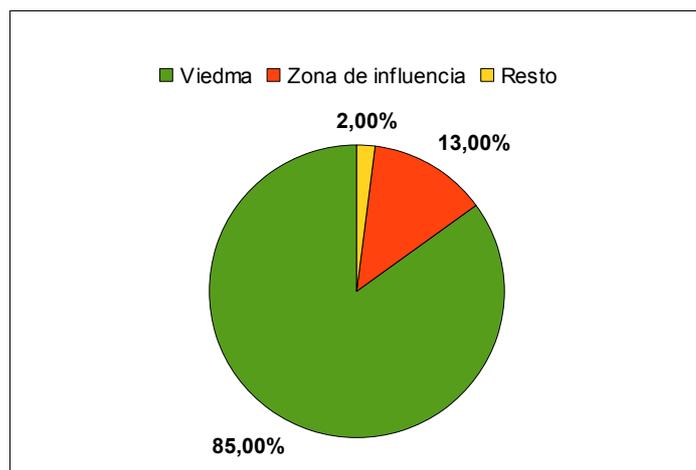


Gráfico 3: Programación I – Alumnos distribución por lugar de procedencia

Respecto a los docentes que participaron de la experiencia, uno estaba designado como profesor adjunto responsable de la parte teórica y otro como Jefe de Trabajos Prácticos (JTP) con una dedicación simple en ambos casos. Para el JTP, esta fue su segunda experiencia en la materia, el otro docente trabaja en la materia desde el año 2009.

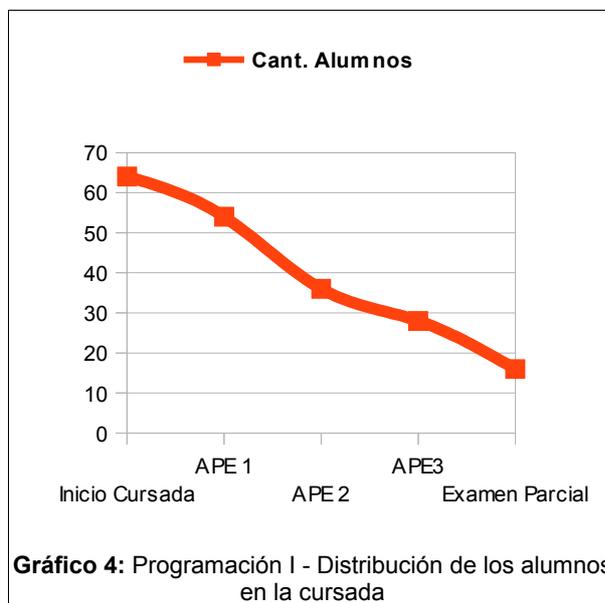
En cuanto al programa de la materia, para la cohorte 2013, el mismo, contempla el desarrollo de tres actividades prácticas entregables (APE) durante la cursada, diseñadas para trabajar los temas correspondientes a las unidades: II (Introducción a Pascal. Estructuras de Control, Datos y Tipos de Datos), V (Tipos de Datos indexados. Vectores y Matrices) y VI (Punteros y Estructuras de Datos Compuestas. Listas Enlazadas). Para cada APE se han definido dos consignas, de esta forma cada tutor/docente trabajará sobre un tema diferente; sin que esto implique una exclusión a atender pedidos de consulta o colaborar con su par y/o grupos de otros temas diferentes al que él tiene asignado. Desde el inicio de la cursa se presenta un cronograma tentativo para cada una de las APE.

| Actividad | Fecha de Inicio | Tema | Consignas | Cant. Alumnos |
|-----------------------|-----------------|---|--|---------------|
| APE 1 | 21-03-2013 | Introducción a Pascal. Estructuras de Control, Datos y Tipos de Datos | Juego "La Tapadita" Simulación "Teclado telefónico" | 54 |
| APE 2 | 25-04-2013 | Tipos de Datos indexados. Vectores y Matrices | Juego "Tateti" Simulación "Cuadrado Mágico" | 36 |
| APE 3 | 27-05-2013 | Punteros y Estructuras de Datos Compuestas. Listas Enlazadas | Simulación Pilas Simulación Colas | 28 |
| Examen Parcial | 07-06-2013 | Integrador del programa | | 17 |

Tabla 2. Programación de las APE

La experiencia se inició con un total de 54 alumnos. Donde, para la APE1 se dividieron en 12

grupos de los cuales completaron todas las etapas de la actividad 11 grupos. Para la APE2 se formaron 14 grupos con un total de 36 alumnos y completaron la actividad 5 grupos. Para la APE3 se formaron 11 grupos con un total de 28 alumnos y finalizaron la actividad 4 grupos. El gráfico 4 muestra el movimiento de alumnos desde el inicio de la cursada hasta el examen parcial de la materia.



Respecto al desgranamiento que se observa no solo entre las APE sino desde el inicio de la cursada, coincide con los estudios sobre este tema presentados entre otros, por García et al. (2011), Cernadas et al. (2009), Maldonado et al. (2009); De Giusti et al. (2003), donde se señala que los alumnos manifiestan arribar a la universidad con una preparación insuficiente, no contar con hábitos de estudios, entre ellos el análisis y comprensión de textos; etc. De Giusti et al. (2003), en referencia a los alumnos de los cursos introductorios a la programación de las carreras vinculadas a informática, señalan algunas características presentes en ellos, que generan dificultades en el aprendizaje y concreción de los objetivos de las materias, a saber:

- falta de entrenamiento en pensar y expresar ideas.
- dificultad para aprender temas básicos que requieran algún grado de abstracción.
- falta de orientación vocacional adecuada.
- gran disparidad de conocimientos y formación previa.
- Poca valoración por el trabajo sistemático.
- cierta metodología de “prueba y error” asociada al uso de las computadoras.

De esta forma, se vuelve necesario buscar estrategias que faciliten la inserción a la vida universitaria y permitan la retención de los alumnos. En el caso particular de la Lic. en Sistemas de la UNRN, otra de las causales de este desgranamiento que se observa en las materias del

primer año, podría estar asociado también con el hecho que muchos alumnos a medida que cursan las materias correspondientes al primer cuatrimestre de la carrera, están también cursando las asignaturas introductorias de “Razonamiento y resolución de problemas” (RRP) e “Introducción a la lectura y escritura académica” (ILEA) que la UNRN ha dispuesto como un recorrido previo de ingreso universitario para todas sus carreras. De esta forma quienes no aprobaron estas materias antes del inicio del primer cuatrimestre, pueden cursarlas durante el mismo y/o rendirlas en forma libre en las fechas establecidas por el calendario académico. El estudiante deberá regularizar a través del examen final o promoción (sólo para la cursada cuatrimestral) estas dos asignaturas extracurriculares de formación básica para poder cursar las materias de segundo año y siguientes de cada carrera. Ambas materias tienen una carga horaria de 64hs y en el caso de matemáticas se cursaba los días sábados de 8-12 hs. El calendario académico para rendir los exámenes parciales correspondientes a estas asignaturas es el que se muestra en la tabla 3.

| | Evaluaciones voluntarias de ILEA y RRP: | Exámenes Cursada ILEA y RPP |
|-------------|---|---|
| 2012 | 22 y 23 de octubre 12 y 13 de noviembre 17 y 18 de diciembre 2012 | |
| 2013 | 13 y 14 de febrero 25 de febrero y 7 de marzo | 20 de Abril 08 de Junio 15 de Junio |

Tabla 3. Programación de las asignaturas introductorias a la UNRN

Respecto a la conformación de los grupos, las imágenes 12,13 y 14, presentan la conformación de los grupos sobre el entorno Moodle. Para la primera APE los alumnos fueron quienes decidieron como conformar un equipo de trabajo; a condición de que el tamaño de los grupos no superara los 4 alumnos. Esta cota máxima al tamaño de los grupos, se estableció en base a lo propuesto por Barkley et al. (2007), para quienes el tamaño del grupo debe ser lo suficientemente pequeño, de manera que permita la participación plena y genere confianza entre los miembros, y a la vez lo suficientemente amplio como asegurar la diversidad . En una entrevista de cierre de la experiencia, se les consulto a un grupo de alumnos respecto a que aspectos habían tenido en cuenta para la conformación de los equipos de trabajo. Algunos indicaron el nivel de cercanía con los compañeros (por amistad, porque provienen de la misma escuela o porque se conocían de otra cursada), otros indicaron que para la primer APE como no se conocían entre sí y no tenían ningún tipo de preferencia por algún compañero, se agruparon por cercanía física en el aula. En particular, los alumnos recursantes, indicaron que preferían agruparse con alumnos que ya se conocían, más allá de los conocimientos que tuvieran sobre la temática de la APE en particular. La primer APE se puso en marcha 6 clases después de comenzada la cursada de la materia, esto

dejó espacio para que los alumnos se conocieran e intercambiaran entre sí, entraran en confianza y pudieran observar la dinámica de las distintas clases tanto de teoría como de prácticas, donde como se señala en el capítulo de diseño de la propuesta de enseñanza y aprendizaje, se plantea un modelo de enseñanza y aprendizaje más cercano al alumno, que busca generar situaciones que lleven a los alumnos a involucrarse y participar en la resolución de problemas. Se pudo observar tal como plantean Perales et al. (2010), la influencia que ejerce el rol de los docentes en el proceso educativo, logrando al finalizar la cursada que los alumnos en su diversidad, comprendan y asimilen la propuesta de la materia. En el sentido, han entendido la importancia de reconocer el lugar de ellos, de sus actividades, de los saberes previos que trajeron al curso, la valoración de los mismos y de forma muy importante la comunicación con sus pares y docentes.

| Grupos (12) | Miembros del grupo | Número de usuarios |
|-------------|--|--------------------|
| Grupo1 | FEDERICO EMANUEL CASTRO, Juan Carlos Nicolas Leuze, Federico Antonio Senra | 3 |
| Grupo10 | Rodrigo Arce, Mariano Jose Cobas, Miranda Lucas Sebastian | 3 |
| Grupo11 | Alejandro Brivi, Enzo Encina, Vanesa Ibañez, Cesar Server | 4 |
| Grupo12 | JAVIER ISAURO ANAZGO, German Rolando Dietz Allende, Marcelo Emmanuel Monserrat, Gonzalo Silva Erripa | 4 |
| Grupo2 | Cecilia Abrameto, Bianca Albariño, Gisela Rodriguez | 3 |
| Grupo3 | Mauro Benitez, Nicolas Elosegui, FEDERICO ZANARDI | 3 |
| Grupo4 | Federico Difabio, Francisco Vicente, Jonathan Zarzosa | 3 |
| Grupo5 | Ezequiel Scalesi, Rocio Sosa Luna, Santiago Sosa Luna | 3 |
| Grupo6 | Sergio Choque, Lourdes Lefinir, Maila Martinez, Jairo Ezequiel Sandalié | 4 |
| Grupo7 | Gustavo Federico Chorolque, Brian Emmanuel Curiqueo, Graff Kevin Emanuel, Ricardo Luis Peña | 4 |
| Grupo8 | Pablo Nicolas Guenomil, Franco Luque, Matias Valicenti | 3 |
| Grupo9 | Gabriela Albornoz, Lucas Ivan Calbucoy, Joaquin Carlos Suarez | 3 |

Imagen 12 - Grupos APE1

| Grupos (14) | Miembros del grupo | Número de usuarios |
|-------------|--|--------------------|
| GAE2-1 | FEDERICO EMANUEL CASTRO, Juan Carlos Nicolas Leuze, Federico Antonio Senra | 3 |
| GAE2-10 | Rodrigo Arce, Mariano Jose Cobas, Miranda Lucas Sebastian | 3 |
| GAE2-11 | Enzo Encina, Vanesa Ibañez, Cesar Server | 3 |
| GAE2-12 | German Rolando Dietz Allende, Marcelo Emmanuel Monserrat, Gonzalo Silva Erripa | 3 |
| GAE2-13 | Alejandro Brivi, Jonathan Alexis Estevanacio | 2 |
| GAE2-14 | Sergio Choque, Maila Martinez | 2 |
| GAE2-2 | Cecilia Abrameto, Gisela Rodriguez | 2 |
| GAE2-3 | Mauro Benitez, Nicolas Elosegui | 2 |
| GAE2-4 | Federico Difabio, Ricardo Luis Peña | 2 |
| GAE2-5 | Ezequiel Scalesi, Rocio Sosa Luna, Santiago Sosa Luna | 3 |
| GAE2-6 | Lourdes Lefinir, Jairo Ezequiel Sandalié | 2 |
| GAE2-7 | JAVIER ISAURO ANAZGO, Gustavo Federico Chorolque, FEDERICO ZANARDI | 3 |
| GAE2-8 | Pablo Nicolas Guenomil, Franco Luque, Matias Valicenti | 3 |
| GAE2-9 | Gabriela Albormoz, Lucas Ivan Calbucoy, Joaquin Carlos Suarez | 3 |

Imagen 13 - Grupos APE2

| Grupos (11) | Miembros del grupo | Número de usuarios |
|---------------------|--|--------------------|
| GAE2-11 | Enzo Encina, Vanesa Ibañez, Cesar Server | 3 |
| GAE2-12 | German Rolando Dietz Allende, Marcelo Emmanuel Monserrat, Gonzalo Silva Erripa | 3 |
| GAE2-13 | Alejandro Brivi, Jonathan Alexis Estevanacio | 2 |
| GAE2-2 | Cecilia Abrameto, Gisela Rodriguez | 2 |
| GAE2-3 | Mauro Benitez, Nicolas Elosegui | 2 |
| GAE2-4 | Federico Difabio, Ricardo Luis Peña | 2 |
| GAE2-7 | JAVIER ISAURO ANAZGO, Gustavo Federico Chorolque, FEDERICO ZANARDI | 3 |
| GAE2-8 | Pablo Nicolas Guenomil, Franco Luque, Matias Valicenti | 3 |
| grp10_CasCalMiranda | Lucas Ivan Calbucoy, FEDERICO EMANUEL CASTRO, Miranda Lucas Sebastian | 3 |
| grp5_ArceScalesi | Rodrigo Arce, Ezequiel Scalesi | 2 |
| grp6_LefMarSandalie | Lourdes Lefinir, Maila Martinez, Jairo Ezequiel Sandalié | 3 |

Imagen 14 - Grupos APE3

A partir de la APE2, los docentes en virtud del trabajo desarrollado sobre el aula virtual, en atención a los pedidos de los alumnos respecto a los grupos en los que estaban, y atendiendo a los objetivos académicos que se perseguían con las APE, sugirieron y realizaron cambios. Si se analizan las imágenes 12, 13 y 14 es posible observar estos movimientos de equipos. Así por ejemplo, el grupo 7 de la APE1, se desarmó, en parte porque algunos de sus integrantes abandonaron la materia, y otra porque los alumnos que quedaron decidieron no continuar

trabajando juntos. En este sentido se observa que uno de los alumnos de este grupo, pudo consolidar un grupo para las APE 2 y 3. Por otra parte, otros equipos como es el caso del grupo 6, se dividieron en dos grupos para la APE2 y luego se volvieron a reunir para la APE3. En este caso particular la división del grupo para la APE2, estuvo motivada por el hecho que la consigna planteaba grupos con máximo de 3 personas. Atendiendo a las migraciones que se produjeron entre los grupos, es posible diferenciar en atención a la categorización de los grupos hecha por Johnson et al. (1999) dos tipos de grupos: formales y de base. Los formales son los grupos que se armaron para una APE específica y luego se disolvieron y los de base son los que lograron consolidarse propiciando relaciones duraderas que permitieron el apoyo mutuo y el respaldo entre los miembros del equipo.

| Actividad | Etapas del Desarrollo de las APE | | | | | |
|--------------|----------------------------------|--------------------------------|---|---------------------------------------|---|---------------------|
| | Debate Inicial | Análisis & diseño Foros / Wiki | | Implementación Programa/ Presentación | | Evaluación Encuesta |
| APE 1 | 12 | 12 | 6 | 12 | 6 | 12 |
| APE 2 | 14 | 8 | 4 | 8 | 5 | 5 |
| APE 3 | 11 | 6 | 0 | 6 | 4 | 6 |

Tabla 4. Cantidad de grupos por etapas de las APE

En la tabla 4 contiene información para cada una de las etapas del desarrollo de las APE referida a la cantidad de grupos que participaron de las actividades propuestas para las mismas, tanto en forma virtual a través de las herramientas TIC : foro, wiki y VPL como de las que se desarrollaron en forma presencial en el espacio del aula.

Se puede observar en esta tabla que la herramienta TIC que menos se utilizó fue la wiki. Por su parte, los alumnos expresaron a través de comentarios en forma anónima y entrevistas de cierre de la experiencia, que no podían visualizar la utilidad de la herramienta para el desarrollo del modelo de la solución y/o de la implementación, por otra parte su uso no les resultó intuitivo y en algunos casos solo pudieron utilizar la funcionalidad que permitía subir y compartir archivos. En este sentido, durante el desarrollo de las actividades, algunos estudiantes, consultaron sobre la posibilidad de utilizar el editor de documentos del aplicativo GoogleDocs. Otra de las dificultades observadas por los docentes con respecto al uso de la wiki sobre Moodle, está asociada a la gestión de los usuarios. Para cada consigna de una APE, se crea una wiki con la intención de desarrollar sobre la misma el análisis y diseño del problema, aquí sucede que las páginas de las wikis quedan accesibles a todos los grupos y eso crea confusión y miedo al plagio en los alumnos. Estas percepciones por parte de los alumnos como de los docentes coinciden con lo analizado por Córdoba, Torrecilla y Cuesta Morales (2009).

A continuación se resume para cada una de las APE, los temas y consignas que abordan las mismas. La APE 1 tiene como objetivos que los alumnos trabajaran sobre la resolución de problemas sencillos, poniendo el énfasis en el búsqueda del modelo y en manejo de las estructuras de control. Así se plantean dos opciones una correspondiente al juego de apuestas conocido como la Tapadita o de las Tres Copas en su versión clásica. Del juego participan dos jugadores: el apostador y el mezclador. El otro problema trabaja sobre un interprete de mensajes escritos con un teclado telefónico, formado por los dígitos del 0 al 9 y las teclas * y # . Así se debe desarrollar un método para que el usuario pueda ingresar texto completo a partir de las teclas disponibles que es significativamente menor que las necesarias para representar las letras y los caracteres de puntuación.

Las siguientes APE requieren de conceptos más complejos como son el manejo de la modularización y el de estructuras de datos. Así se plantean para la APE2, los siguientes problemas: TA-TE-TI o tres en línea, donde cada grupo debe modelar e implementar el juego de manera que se pueda jugar contra la máquina o entre dos jugadores. El otro problema se divide en dos: la primer parte consiste en dado un cuadrado de $N \times N$ (cuyos valores ingresa el usuario pero acotado a $N=5$), si el mismo es un cuadrado mágico o si es un cuadrado latino. En el caso que el cuadrado sea mágico, deberá indicar también la constante mágica. Para la segunda parte, a partir de un número N que ingresa el usuario y tomando como base el cuadrado de origen chino "Lo-Shu", los grupos deben modelar e implementar un algoritmo que retorne un cuadrado mágico de orden 3.

Para la APE3, se plantean dos consignas que trabajan sobre el manejo de Pilas y Colas implementadas usando listas enlazadas. La consigna del problema incluyó una definición introductoria de las mismas, que los grupos debieron ampliar a través de la investigación para poder desarrollar la APE. Las APE se han diseñado con la intención de ser integradoras de los conceptos de cada una de las unidades asociadas a la misma, así como también con el objetivo de articular con los conceptos de actividades previas.

Cómo se indicó en el capítulo de propuesta, al finalizar la fecha de entrega de cada APE los alumnos respondieron a una encuesta anónima, que permitió evaluar su propio desempeño, el de su grupo y el del tutor asignado. En el siguiente apartado se exponen y analizan los resultados de las mismas.

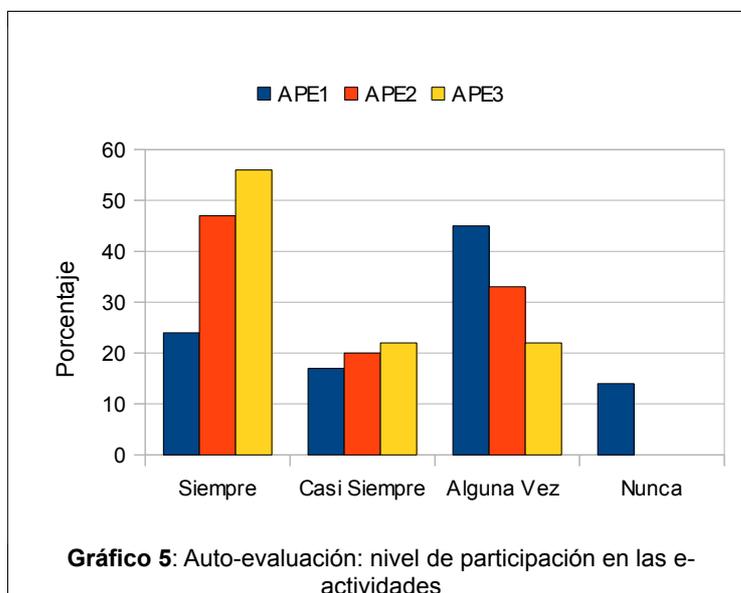
6.1 LA EVALUACIÓN DE LOS ALUMNOS RESPECTO DE SUS PROPIOS PROCESOS DE APRENDIZAJE

La tabla 5 muestra algunas de las preguntas que incluía la encuesta respecto a la auto-evaluación que realizaron los alumnos, en relación a sus propios procesos de aprendizajes en el marco de las APE, y el porcentaje de respuestas a cada una de ellas. Luego se analizan los resultados a través de los gráficos correspondientes.

| Pregunta | APE 1 | APE2 | APE3 |
|--|--|--|---|
| Participé de las e-actividades propuestas (Wiki, foro, VPL) | 24% Siempre 17% Casi Siempre 45% Alguna vez 14% Nunca | 47% Siempre 20% Casi siempre 33% Alguna vez 0 % Nunca | 56% Siempre 22% Casi Siempre 22% Alguna Vez 0% Nunca |
| Dominio (conocimiento, manejo) de la información que se discutió para el desarrollo de la APE | 34% Siempre 41% Casi Siempre 24% Alguna Vez 0% Nunca | 53% Siempre 33% Casi Siempre 13% Alguna Vez 0% Nunca | 56% Siempre 22% Casi Siempre 11% Alguna Vez 11% Nunca |
| Asistí a los encuentros (presenciales o virtuales) establecidos por el grupo | 76% Siempre 24% Casi Siempre 0% Alguna Vez 0% Nunca | 80% Siempre 13% Casi Siempre 7% Alguna Vez 0% Nunca | 89% Siempre 0% Casi Siempre 11% Alguna Vez 0% Nunca |
| Aporte información nueva y relevante en las discusiones que realizó el grupo | 38% Siempre 21% Casi Siempre 31% Alguna Vez 10% Nunca | 47% Siempre 27% Casi Siempre 20% Alguna Vez 7% Nunca | 67% Siempre 22% Casi Siempre 11% Alguna Vez 0% Nunca |
| Respeto por las opiniones de los demás | 83% Siempre 17% Casi Siempre 0% Alguna Vez 0% Nunca | 93% Siempre 7% Casi Siempre 0% Alguna Vez 0% Nunca | 89% Siempre 11% Casi Siempre 0% Alguna Vez 0% Nunca |
| El desarrollo de la APE le permitió una mejor comprensión de los conceptos de las unidades involucradas. | 0% Nada 7% Poco 55% Normal 38% Mucho | 7% Nada 0% Poco 60% Normal 33% Mucho | 11% Nada 11% Poco 0% Normal 78% Mucho |
| Hubiese podido realizar esta misma actividad en forma individual | 28% No 55% Tal Vez 17% Si | 27% No 47% Tal Vez 27% Si | 22% No 22% Tal Vez 56% Si |
| Cómo calificaría su interés en la materia, luego de realizar la APE | 28% Muy Grande 34% Grande 38% Normal 0% Poco 0% Nulo | 20% Muy Grande 33% Grande 47% Normal 0% Poco 0% Nulo | 44% Muy Grande 11% Grande 33% Normal 11% Poco 0% Nulo |

Tabla 5. Encuesta - Auto-evaluación

El gráfico 5 permite observar las apreciaciones de los alumnos, respecto a su propia participación en las e-actividades propuestas, a saber: foros, wiki, y el laboratorio virtual VPL. En este sentido, es posible observar que el nivel de participación aumentó en cada APE. Solo para la primer APE, un poco más del 10% manifestó haber tenido una participación nula.



En una clase práctica de la materia, durante el periodo de tiempo en el que se estaba desarrollando la APE2, se les pidió a los alumnos, que en forma anónima comentaran porque no estaban participando de las e-actividades. Algunos indicaron que por falta de coordinación y de organización. Otros manifestaron que no podían encontrarle utilidad al uso de la wiki o el foro, que preferían reunirse en forma personal.

A continuación se transcriben algunas de las respuestas:

“La herramienta del foro o la wiki no es algo muy útil para nosotros pero entiendo que para los profesores es importante para poder evaluar a los alumnos porque personalmente no pueden ver lo que hace cada uno.”

“El foro no le veo ninguna utilidad, lo mismo la wiki, es mucho mejor personalmente”

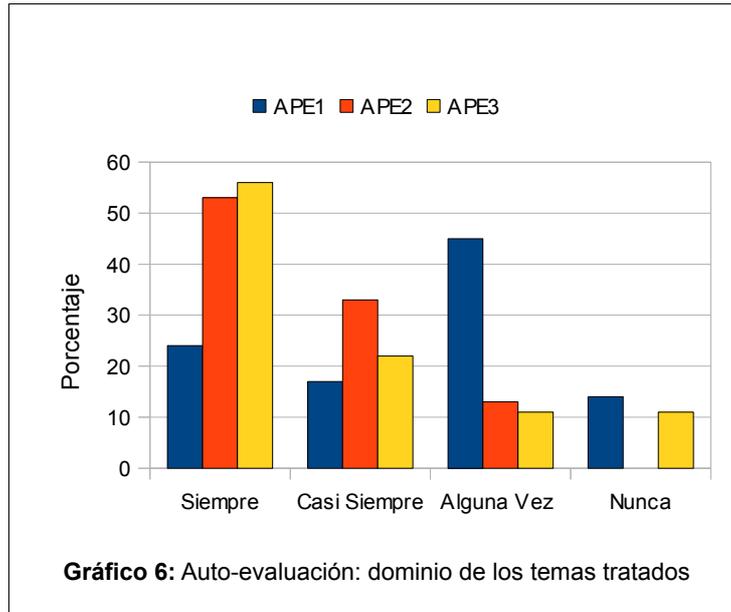
“No hubo tiempo de hacer y entregar el trabajo por el parcial de matemática y el recuperatorio de Organización”

“Preferiría entregar el trabajo en papel”

“No hubo tiempo de juntarnos en alguna casa o lugar, ¿por qué no consideramos realizar esto por separado?”

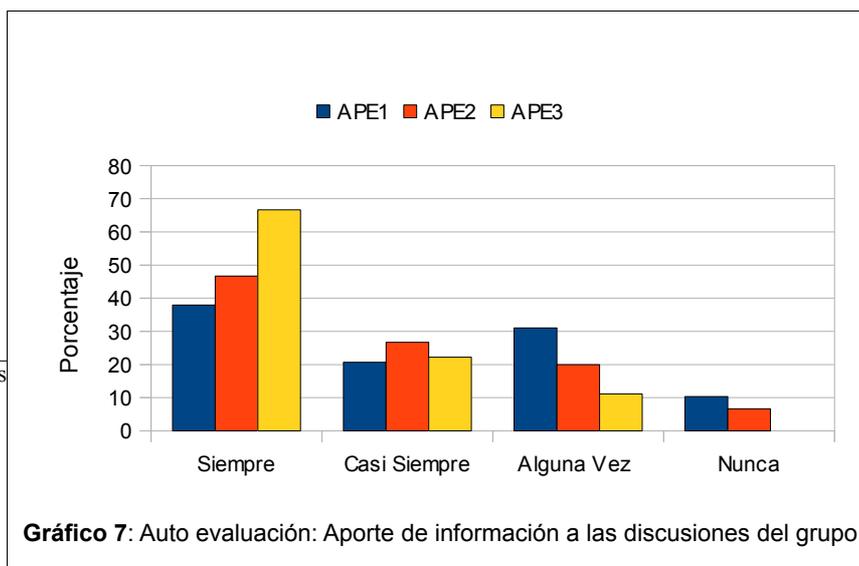
Si se analizan las respuesta de los alumnos, es posible observar que al principio de la experiencia les resultó difícil adoptar la propuesta de trabajo grupal en forma colaborativa y mediada por los

recursos TIC propuestos . Sin embargo, conciben a la wiki y al foro como medios de evaluación. Al finalizar la experiencia, se hicieron entrevistas a un grupo de alumnos que formaron parte de la misma, algunos de ellos están cursando la materia correlativa Programación II y otros el taller remedial de la materia Programación I. En la entrevista se les volvió a consultar a cerca de la utilidad de los recursos TIC propuestos y algunos manifestaron que después de haber experimentado con las mismas, consideran al foro y al laboratorio como recursos que les permiten sacarse dudas, compartir la resolución de problemas con otros compañeros y docentes y poder avanzar con las actividades propuestas por la materia, más allá de las clases presenciales de teoría y práctica.

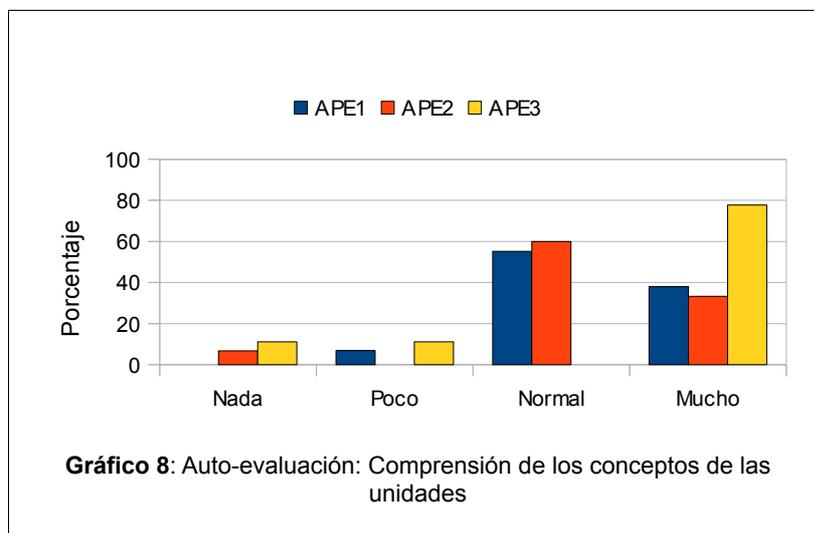


El gráfico 6, presenta el resultado de la auto-evaluación que hicieron los estudiantes respecto a si tenían dominio (conocimiento y manejo) de la información que se discutía en cada APE. Se puede observar que en cada APE se produjo un incremento del mismo. En este sentido vale destacar que cada APE era integradora de los conceptos vistos en las unidades involucradas más los analizados en la APE anterior. Así por ejemplo para las APE 1 y 2, entre el 30 y 50% de los encuestados consideraron tener siempre o casi siempre conocimiento sobre los temas que se abordaron. En la APE3, aunque el porcentaje de alumnos que manifestaron tener siempre dominio de la información para el desarrollo de la APE fue alto, se observa que hubo un porcentaje de alumnos (11%) que respondieron no haber tenido nunca conocimiento ni manejo de la información sobre la que se discutía durante el desarrollo de la APE.

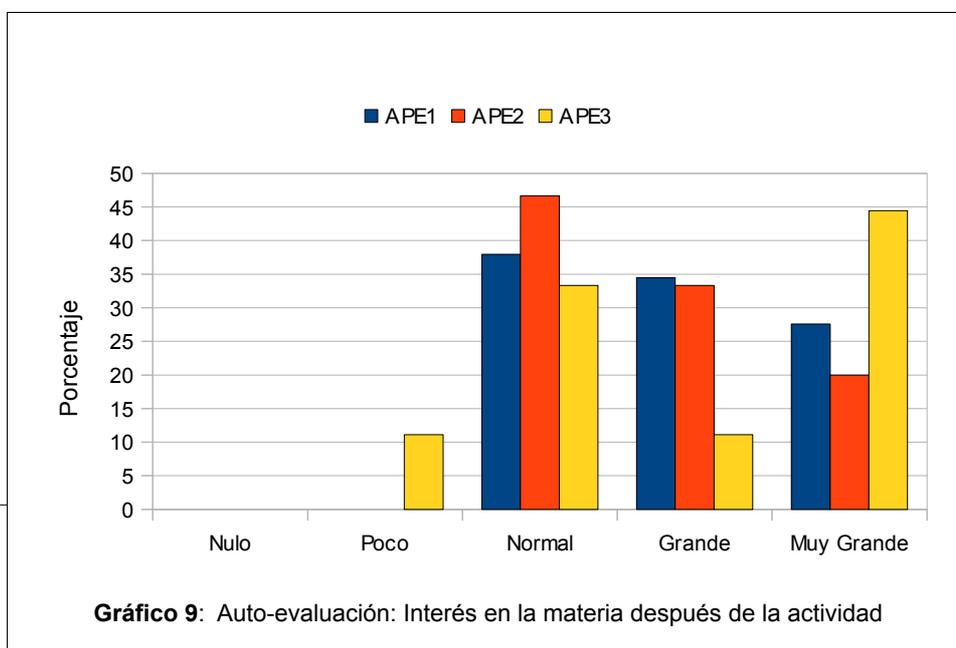
El gráfico 7, muestra las respuestas que indicaron los alumnos respecto a si durante el desarrollo de las discusiones que se generaron para el desarrollo de las APE, pudieron realizar aportes de información nueva y relevante. En su mayoría las respuestas han sido positivas (siempre, casi siempre) para las tres actividades. Solo para las dos primeras APE, se observa que hubieron alumnos que consideran que no haber podido realizar aportes relevantes para el desarrollo de las mismas.



El gráfico 8, permite visualizar las percepciones de los alumnos respecto a si el desarrollo de las APE les permitió una mejor comprensión de los conceptos involucrados. Se observa que más del 50% consideran que las APE contribuyeron en forma normal en las dos primeras actividades y en la última se observa que la contribución superó el 70%. Esta APE resultó una experiencia de investigación para los grupos, ya que la resolución del problema planteado requirió de conceptos no analizados en clase (Pilas y Colas). En este sentido, los grupos han realizado un aprendizaje de las temáticas planteadas a través del descubrimiento y construcción del mismo, ya que han sido los propios alumnos quienes han tenido que buscar información, seleccionarla, organizarla e intentar resolver el problema planteado en la consigna de la APE. Contando con la orientación del tutor, pero donde la resolución del problema ha dependido de las elecciones realizadas por el grupo (Restrepo, 2005).



El gráfico 9 muestra el interés que manifestaron los alumnos en la materia, luego de concluida cada actividad. Se observa que el mismo, se mantuvo en valores que variaban de normal a muy grande, salvo para la última APE.



En la tabla 6, se muestran los resultados de analizar la relación existente entre las respuestas a las preguntas de los gráficos 8 y 9 para cada una de las APE. Para esto, se calculo usando el aplicativo InfoStat¹⁶, el índice de Pearson que indica el grado de relación/ asociación entre las variables medidas.

Para las tres APE se observa un índice de correlación positivo, sin embargo solo es significativo para las APE 1 y APE 2. Este coeficiente indicaría que un aumento en una de las variables (interés materia, comprensión) implicaría un aumento en la otra.

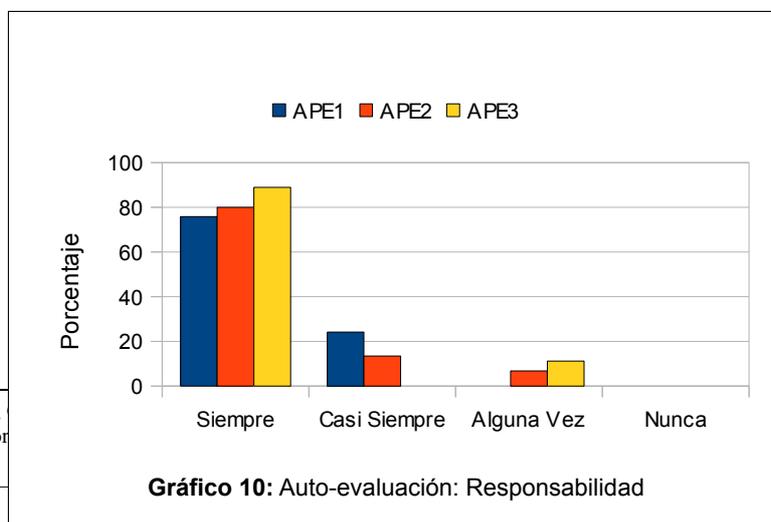
Correlación de Pearson: **coeficientes**\probabilidades

| APE1 | Interés Materia | Comprensión |
|-----------------|-----------------|-------------|
| Interés Materia | 1 | 0,01 |
| Comprensión | 0,5 | 1 |
| APE2 | Interés Materia | Comprensión |
| Interés Materia | 1 | 0,01 |
| Comprensión | 0,68 | 1 |
| APE3 | Interés Materia | Comprensión |
| Interés Materia | 1 | 0,08 |
| Comprensión | 0,61 | 1 |

Tabla 6. Correlación entre Interes y Comprensión de la materia

En el gráfico 10 se puede observar la auto-evaluación que hicieron los alumnos respecto a su responsabilidad en relación a la asistencia a los encuentros sean estos presenciales o virtuales, que se establecieron dentro del grupo. El 88% manifiestan sentirse responsables y consideran que han podido cumplir con los compromisos establecidos con el grupo. En una entrevista de cierre de cursada, algunos alumnos manifestaron que para poder llevar adelante el trabajo, más que el conocimiento respecto al tema a tratar fuera heterogéneo dentro del grupo, valoraron más el sentido de responsabilidad para con las tareas a desarrollar. En decires de un alumno:

“... no siempre el que más sabe es el que más participa... sino el que más ganas le pone...”



16 Di Rienzo J.A.,
Nacional de Cór

13. Grupo InfoStat, FCA, Universidad

Una situación a destacar, es la auto-evaluación que hacen los alumnos respecto a si han sido respetuosos de las opiniones ajenas en la relación a los temas trabajados en cada APE. Así, el gráfico 11, muestra un resultado altamente positivo para las tres APE. Esta es una capacidad muy requerida en ámbitos laborales donde los grupos de trabajo son multidisciplinarios, como es el caso del desarrollo del software a nivel profesional.

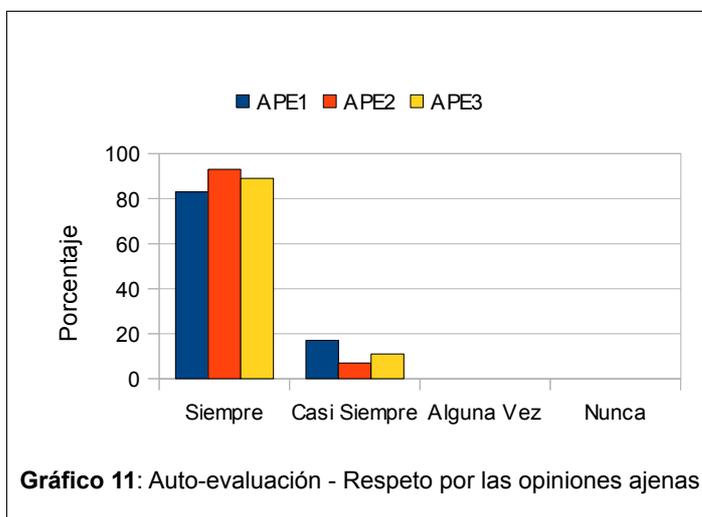
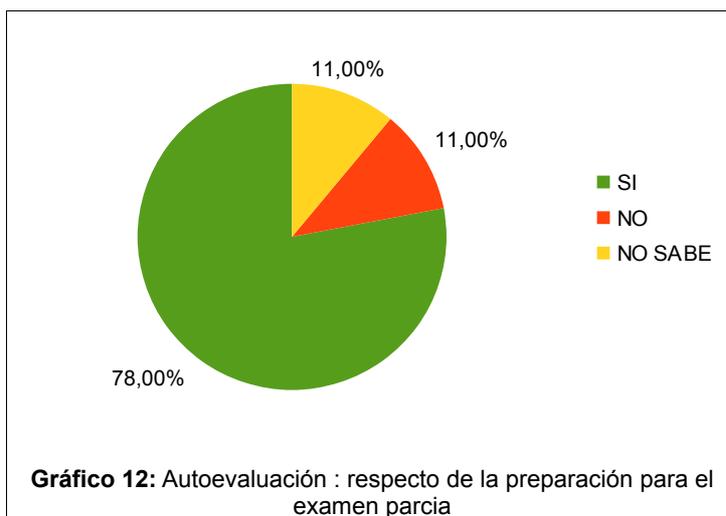


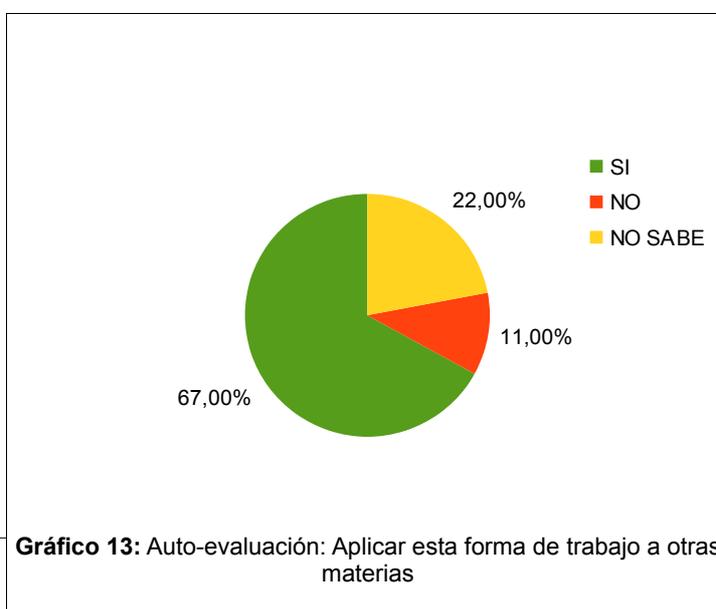
Gráfico 11: Auto-evaluación - Respeto por las opiniones ajenas

Luego de finalizada la entrega de la APE 3 y después de que los alumnos rindieron el examen parcial de la materia, se les consultó acerca de si esta propuesta de trabajo les había permitido prepararse mejor para rendir el examen, aquí casi el 78% de los alumnos respondió positivamente, aún entre quienes indicaron no haber aprobado el examen.



La APE 3, resultó una caso particular, cuyos resultados coinciden con lo expresado por Guevara (2010), en el sentido que se trató de una actividad donde los grupos, como se señalaba anteriormente, debieron hacer un proceso de investigación, atendiendo a que las estructuras de datos necesarias (Pilas y Colas) para la resolución de la actividad no habían sido presentadas en las clases teóricas o prácticas. Así, se pudo confirmar que esta situación genera una situación de independencia entre los docentes y los alumnos, sin que esto haya implicado una falta de comunicación entre ambos. Fueron los propios grupos quienes construyeron los conceptos de Pilas y Colas, en base a un proceso de investigación que ellos mismos desarrollaron y que contó con la asistencia de los tutores, en el sentido de orientar las búsquedas y brindar asistencia antes las consultas. Y en este sentido, coincidimos con Guevara(2010), en cuanto a que esta modalidad de trabajo permite preparar a los alumnos para el estudio permanente y la actualización constante que se le será requerida en el ámbito profesional.

Consultados acerca de si les gustaría que esta forma de trabajo se aplicará a otras materias, el 67% de los alumnos respondió en forma afirmativa.



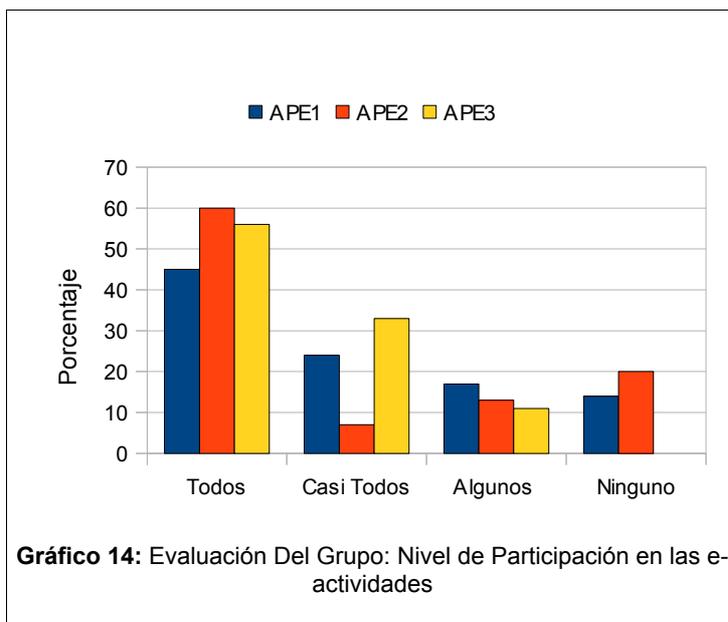
6.2 LA VALORACIÓN DE LOS ALUMNOS RESPECTO AL TRABAJO COLABORATIVO

En este apartado se analizarán de las respuestas que indicaron los estudiantes en relación a la evaluación de su grupo de trabajo en cada una de las APE . En la tabla 7, se muestran las preguntas correspondientes a la evaluación grupal y las respuestas obtenidas en cada APE.

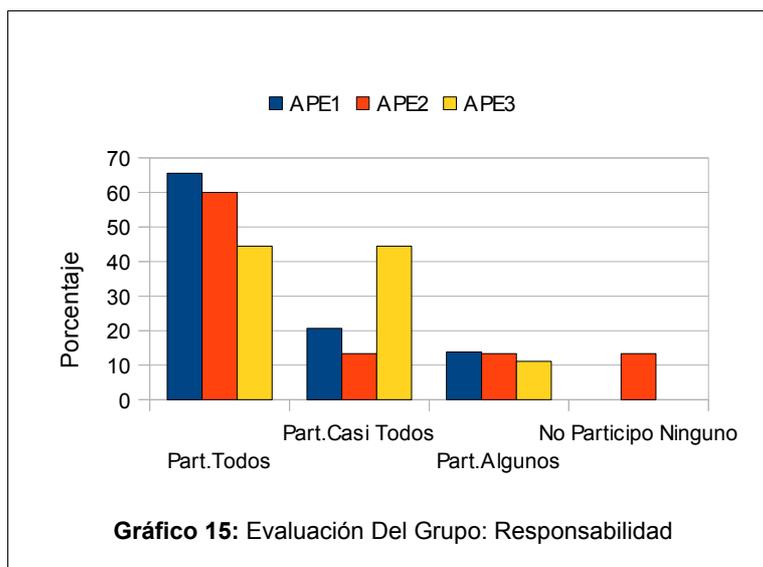
| Pregunta | APE 1 | APE2 | APE3 |
|--|--|---|--|
| Participación de las actividades propuestas (wiki, foro, VPL) | 45% Part. Todos 24% Part. Casi Todos 17% Part. Algunos 14% No Part. Ninguno | 60% Part. Todos 7% Part. Casi Todos 13% Part. Algunos 20% No Part. Ninguno | 56% Part. Todos 33% Part. Casi Todos 11% Part. Algunos 0% No Part. Ninguno |
| Asistencia a los encuentros (presenciales o virtuales) establecidos por el grupo | 66% Part. Todos 21% Part. Casi Todos 14% Part. Algunos 3% No Part. Ninguno | 60% Part. Todos 13% Part. Casi Todos 13% Part. Algunos 13% No Part. Ninguno | 44% Part. Todos 44% Part. Casi Todos 11% Part. Algunos 0% No Part. Ninguno |
| Aportes de información nueva y relevante en las discusiones que realizó el grupo | 34% Todos Aportaron 31% Casi Todos Aportaron 31% Solo Algunos Aportaron 3% Ninguno Aporto | 53% Todos Aportaron 13% Casi Todos Aportaron 7% Solo Algunos Aportaron 27% Ninguno Aporto | 33% Todos Aportaron 44% Casi Todos Aportaron 22% Solo Algunos Aportaron 0% Ninguno Aporto |
| Dominio de la información que se discutió para el desarrollo de la APE | 41% Todos tenían conocimiento 34% Casi Todos tenían Conocimiento 24% Solo algunos tenían conocimiento 3% Ninguno tenía Conocimiento | 60% Todos tenían conocimiento 13% Casi Todos tenían Conocimiento 13% Solo algunos tenían conocimiento 13% Ninguno tenía Conocimiento | 33% Todos tenían conocimiento 33% Casi Todos tenían Conocimiento 33% Solo algunos tenían conocimiento 0% Ninguno tenía Conocimiento |
| Respeto por las opiniones de los demás | 100% Todos los integrantes 11% Casi todos los integrantes 0% Algunos 0% Ninguno de los integrantes | 67% Todos los integrantes 27% Casi todos los integrantes 0% Algunos 7% Ninguno de los integrantes | 100% Todos los integrantes 11% Casi todos los integrantes 0% Algunos 0% Ninguno de los integrantes |
| Le gustaría volver a trabajar con este grupo | 55% Si con todos los integrantes 38% Si, con algunos 7% No | 47% Si con todos los integrantes 13% Si, con algunos 40% No | 44% Si con todos los integrantes 56% Si, con algunos 0% No |

Tabla 7. Encuesta - Evaluación del Grupo de Trabajo

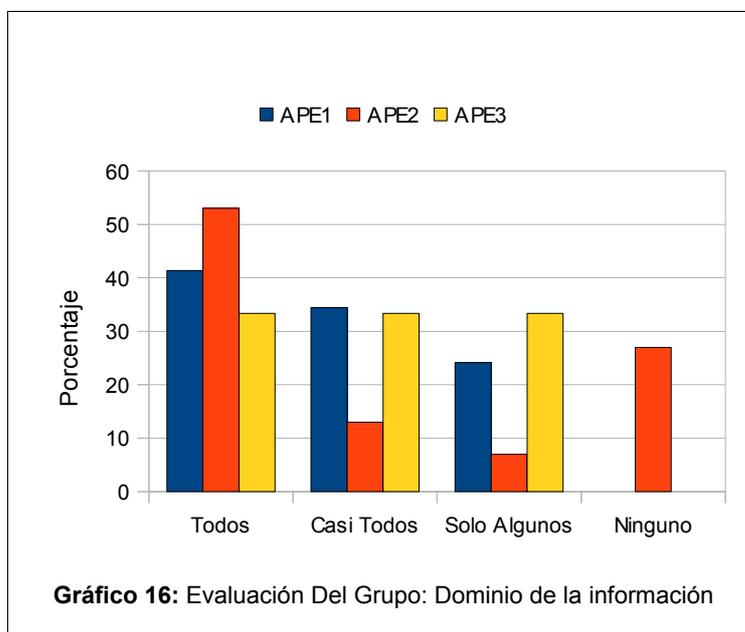
En el gráfico 14, se muestra el nivel de participación que consideran tuvieron sus compañeros en las e-actividades (foros, wiki, VPL). En su mayoría manifiestan una apreciación positiva, considerando que todos participaron, solo manifiestan una falta de participación para las APE1 y APE2, que no supera el 20%. Y esto coincide con los resultados que se presentan en el gráfico 5, donde los alumnos respondieron respecto a su propia participación en la actividades mediadas por TIC propuestas para cada APE.

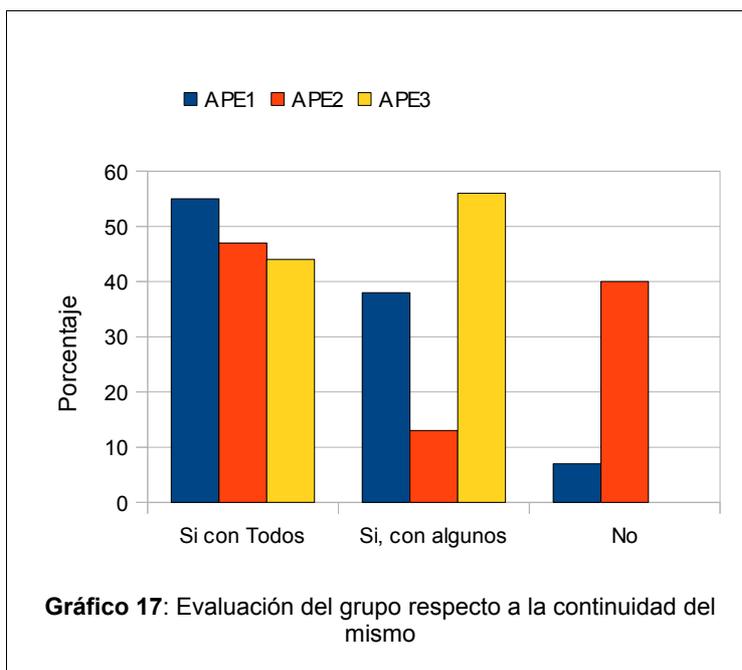


En este sentido si se observa el gráfico 15, que muestra el grado de responsabilidad que consideran tuvo el equipo durante las actividades, es posible observar que aunque las apreciaciones son mayormente positivas, sin embargo, para la APE 2, el 13% de los encuestados manifestó que en el grupo con el que trabajo durante el desarrollo de la misma habían algunos compañeros que solo habían participado algunas veces.



El gráfico 16 ilustra las respuestas que dieron los alumnos respecto al dominio de la información con la que contaban sus compañeros de grupo durante el desarrollo de las actividades. Aquí puede verse que las apreciaciones son mayormente positivas, sin embargo para la APE2 hubieron grupos donde es posible apreciar que solo algunos miembros tenían conocimiento (13%) y otros donde el ninguno tenía conocimiento sobre los temas tratados (13%).





El gráfico 17 visualiza las respuestas que dieron los alumnos al ser consultados acerca de si volverían trabajar con el mismo equipo. Para las tres APE más del 40% respondió afirmativamente. Sin embargo, en aquellos casos donde se observó a través del aula virtual y/o de la encuesta, que sus miembros no deseaban seguir trabajando juntos y/o que los alumnos lo manifestaron en forma personal equipo al docente, se hicieron cambios para la siguiente APE. Este es el caso de la APE2, donde un 40% afirmó no tener intenciones de continuar trabajando con el grupo actual y un 13% solo estaba interesado en continuar solo con algunos de sus integrantes.

Ante la pregunta: *qué fue lo mejor y lo peor de trabajar con este grupo?*, en general las respuestas coinciden y señalan: la falta de dominio sobre los temas tratados, lo cual posicionaba a los integrantes del equipo en distintos niveles de conocimiento, la despreocupación y la negativa de algunos integrantes a utilizar las herramientas TIC propuestas. En cuanto a lo mejor resaltaron el debate, la puesta en común de las posibles soluciones a las situaciones problemáticas planteadas en las APE y el respeto hacia las opiniones de los demás.

A continuación se transcriben algunas de las respuestas de los alumnos para dicha pregunta.

“Fue muy bueno trabajar con los chicos de mi grupo, porque nos llevamos muy bien y todos aportamos puntos imprescindibles para la solución de esta actividad. ” (alumno XX, APE1 Grupo 1)

“Personalmente lo que me gusto del grupo fue la responsabilidad de cada integrante y el interés por realizar la actividad. Por el contrario fue un poco incomodo adaptar las ideas

por la falta de conocimientos técnicos de algunos integrantes.” (alumno XX, APE1 Grupo 1)
“Lo mejor es que se puede coordinar de manera excelente los encuentros de trabajo (Ya sea personalmente o virtualmente).

Lo peor es que lamentablemente no todos tenemos los mismos conocimientos (Cómo debe suceder en todos los grupos), por lo que, a la hora de discutir la implementación del trabajo, no es parejo. ” (alumno x, APE1, Grupo 4)

“No sé si hay un "mejor" o "peor" opinión sobre este grupo.. Tal que somos todos recursantes, tenemos idea de que se trata y de lo que es la materia en si. Por ende tenemos bien presente de lo que se trata los AE... Para mi, trabajar con estos compañeros fue normal, digamos.. nada fuera de lo común. ” (alumno recursante, APE1, Grupo 6)

“Si consideramos este grupo con 3 integrantes los aspectos fueron positivos ya que se realizo la actividad de forma dinámica, respetuosa y responsable.” (alumno XX, APE1, Grupo 5)

“La falta de conocimiento y comprensión de todo el grupo, (me incluyo) hizo que se tardara en resolver el APE2. Personalmente tuve que estudiar mucho ,leer y mirar tutoriales en Internet para poder comprender realmente la manera de trabajar con los nuevos conocimientos. ” (alumno XX, APE2, Grupo 1)

Estos resultados coinciden con lo expuesto por García Válcárcel et al. (2012), respecto a que existe un grado de resistencia a trabajar grupalmente, en parte porque trabajar en grupo y en forma colaborativa exige mayor dedicación (“*No hubo tiempo de juntarnos en alguna casa o lugar, ¿por qué no consideramos realizar esto por separado?*”) y porque la colaboración se entiende como una división del trabajo y puesta en común de los resultados alcanzados.

Volviendo sobre las opiniones que han manifestado los alumnos en cuanto a su comportamiento y al del grupo, con respecto al sentido de responsabilidad para con las actividades propuestas, asistencia a los encuentros y respeto por las opiniones ajenas, se observa que las opiniones son mayormente positivas y refuerzan lo expuesto por los autores García-Valcárcel et al.(2012) quienes señalan la importancia de estos aspectos en el propio aprendizaje de los alumnos; así como la mejor comprensión de los temas tratados gracias a las explicaciones que pueden recibir de sus pares.

El grado de interactividad entre los miembros de un grupo, que se pudo observar a través del aula virtual no fue parejo para todos los grupos sin embargo, esas interacciones no pueden medirse solo en términos de la frecuencia de las misma, sino como señala Collazos et al. (2006), por la forma en que estas pueden influir en el proceso cognitivo del grupo. Y en este sentido, vale traer las percepciones de los alumnos respecto a que cuando los grupos son dispares en cuanto su

conocimiento sobre el tema a tratar, es difícil para los que menos saben realizar aportes en el sentido de la solución y en algunos casos esta situación obliga a los que más conocimientos tienen a realizar un esfuerzo extra para poder explicar su razonamiento a sus compañeros. De esta forma, ambos perfiles se benefician de la situación.

6.3 LA EVALUACIÓN DE LOS ALUMNOS RESPECTO A LA ENSEÑANZA

Como se explicaba en el capítulo correspondiente al diseño de la propuesta de enseñanza y aprendizaje, cada grupo de trabajo cuenta durante todo el desarrollo de las APE con un tutor, que tiene como misión guiar y acompañar al grupo, brindándole asistencia en los contenidos temáticos involucrados en cada actividad, en la resolución de situaciones problemáticas y en el uso de las herramientas TIC. Como se indica al principio del capítulo, en esta primera implementación de la propuesta, se trabajó con dos tutores.

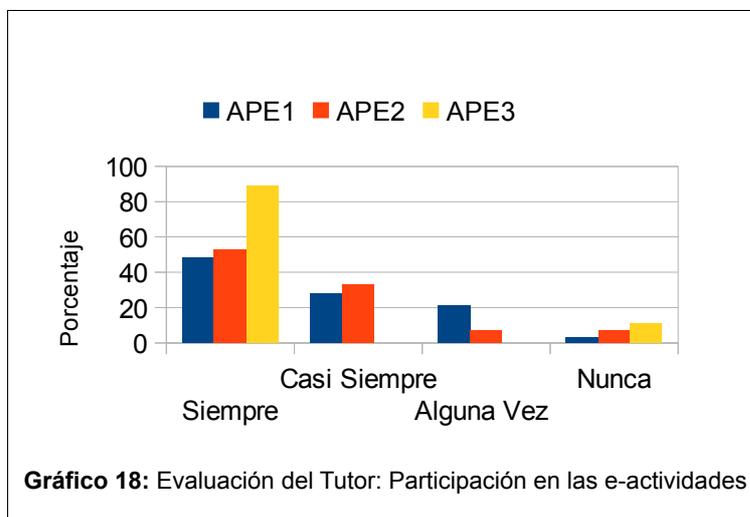
La tabla 8 contiene las preguntas realizadas en cada encuesta de fin de actividad, en relación al comportamiento del tutor durante el desarrollo de las mismas.

| Pregunta | APE 1 | APE2 | APE3 |
|---|---|---|--|
| El tutor ha sido dinámico y activo, fomentando la colaboración (foros, wiki, VPL) | 48% Siempre, 28 % Casi Siempre, 21% Alguna vez, 3% Nunca | 53% Siempre 33% Casi siempre 7% Alguna vez 7% Nunca | 89% Siempre, 0% Casi Siempre, 0% Alguna Vez 11% Nunca |
| El tutor nos ha animado a preguntar y nos ha dado respuestas satisfactorias | 45% Siempre 38% Casi Siempre 14% Alguna Vez 3% Nunca | 46.67% Siempre 46,67% Casi Siempre 6,67% Alguna Vez 0% Nunca | 78% Siempre 0% Casi Siempre 11% Alguna Vez 11% Nunca |
| El tutor tenía dominio de la información que se discutió para el desarrollo de la APE | 38% Siempre 48% Casi Siempre 14% Alguna Vez 0% Nunca | 53% Siempre 40% Casi Siempre 7% Alguna Vez 0% Nunca | 78% Siempre 11% Casi Siempre 0% Alguna Vez 11% Nunca |
| Aporto información nueva y relevante en las discusiones que realizó el grupo | 38% Siempre 38% Casi Siempre 17% Alguna Vez 7% Nunca | 40% Siempre 47% Casi Siempre 13% Alguna Vez 0% Nunca | 44% Siempre 44% Casi Siempre 0% Alguna Vez 11% Nunca |
| Respecto al uso del Laboratorio Virtual (VPL), el tutor les brindo asistencia | 48% SI 14% No 38% No fue Necesario | 60% SI 20% No 20% No fue Necesario | 78% SI 0% No 22% No fue Necesario |

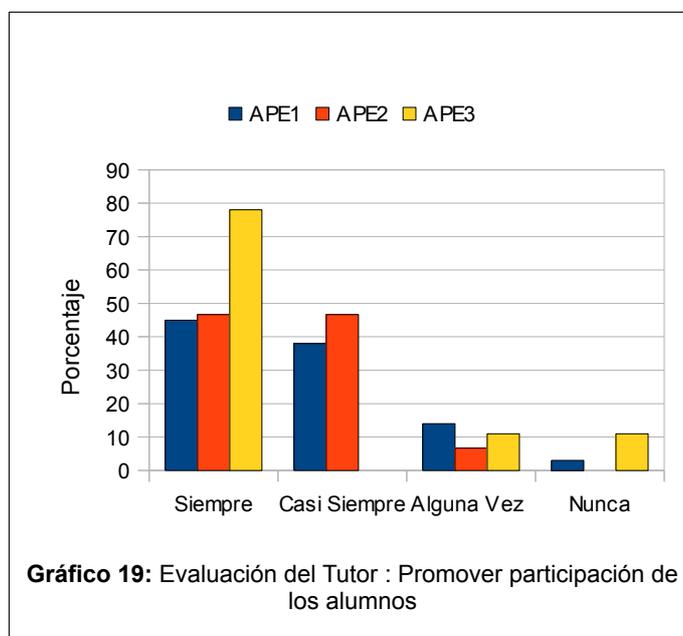
Tabla 8. Encuesta – Evaluación del Tutor

En la encuesta que respondieron los alumnos, consultados acerca de si el tutor asignado par la actividad, ha sido dinámico y activo durante el desarrollo de las APE, fomentando la colaboración a través de las herramientas TIC , foro, Wiki y VPL se obtuvieron los resultados que se muestran

en el gráfico 18. Se puede observar que las apreciaciones de los estudiantes han sido mayormente positivas, sin embargo para la APE 3 los estudiantes manifiestan en un 11 % que hubo una participación nula del tutor a través de las herramientas TIC propuestas.

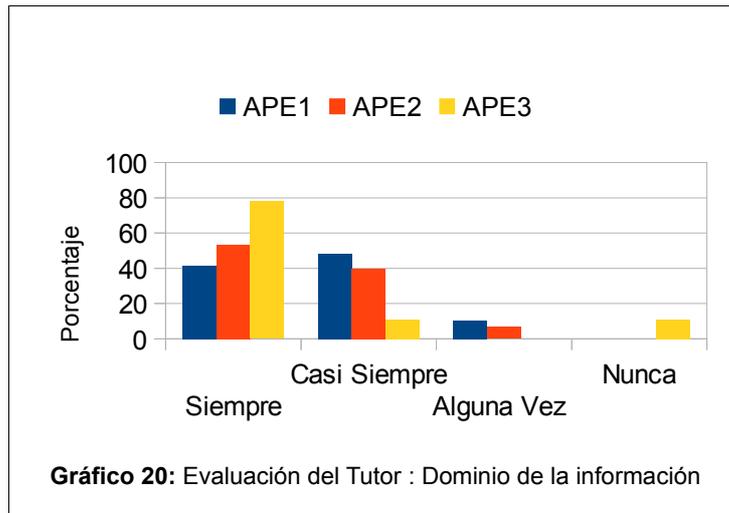


En relación a si el tutor había promovido la participación de los grupos, es decir si los había animado a preguntar y si consideraban que había dado respuestas satisfactorias cuando fue consultado, los alumnos indicaron las respuestas que se muestran en el gráfico 19, donde se puede observar que consideran que el tutor siempre o casi siempre lo había hecho. Solo para la APE 1 un 3% y para la APE 3 un 11% de los alumnos manifestaron, no haberse sentido satisfechos.



En cuanto al dominio de la información que se puso en discusión para el desarrollo de cada APE,

los alumnos consideran (gráfico 20) que los tutores que trabajaron con sus grupos han estado siempre o casi siempre en situaciones de conocimiento y manejo de la información tratada en las APE . Solo en la última APE, se observan resultados negativos, donde un 11% indica que el tutor asignado no tenía dominio de la información, esto en concordancia con los resultados obtenidos para la consulta nivel de participación del tutor (gráficos 18 y 19). En este sentido, como lo indican los autores Perales et al. (2010), un docente que incluya el modelo colaborativo en sus clases deberá prepararse mucho más que aquel que sigue el modelo convencional, conformando un sistema de tareas específico que brinde a los alumnos la oportunidad de investigar por si mismos, y desde su propio acercamiento conceptual, interactuar con sus pares. Los mismos autores remarcan la necesidad de preparar a los alumnos en este nuevo enfoque de trabajo, teniendo confianza en ellos, orientándolos y propiciando en todo momento el intercambio y una comunicación libre a nivel del grupo.



Consultados los alumnos, específicamente acerca de si el tutor les había brindado asistencia en el uso del laboratorio virtual VPL, la mayoría de los alumnos respondieron en la encuesta de las tres APE en forma afirmativa, observándose que para la primera actividad, donde se comenzaba a experimentar con la herramienta, un 38% de los alumnos indico no tener necesidad de recibir asistencia para su utilización. Esto confirma lo que indican los desarrolladores de VPL (Rodríguez del Pino et al. 2010) respecto a su facilidad de uso.

En una entrevista realizada al finalizar la experiencia, se les consulto a algunos alumnos sobre la participación de los tutores durante el desarrollo de la experiencia, y coincidieron en que se evidenciaban dos modelos, a los que hemos nominados como:

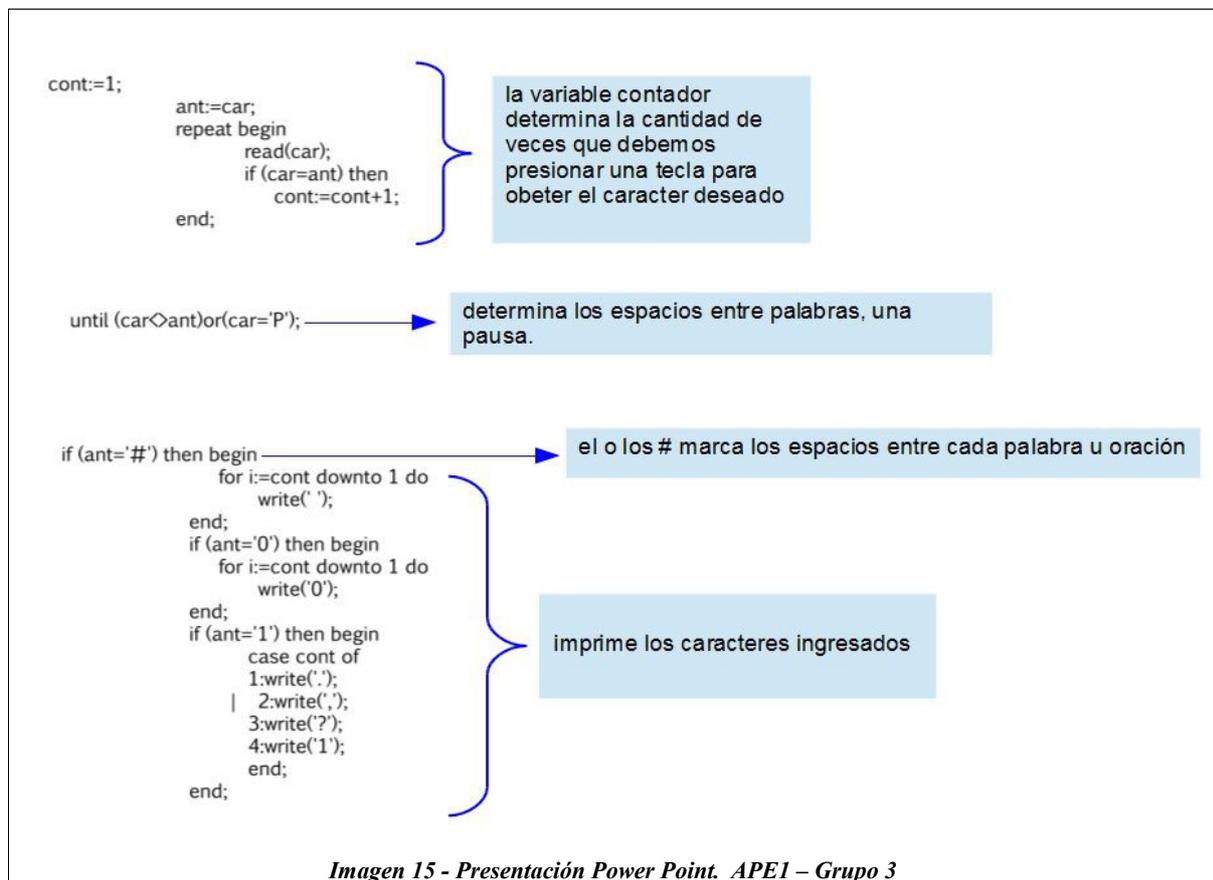
- Pasivo: donde la participación del tutor durante el desarrollo de la APE, consistía en la corrección (a través de VPL) de la etapa de implementación de la APE (programa computacional) y alguien a quien los alumnos podían recurrir en forma virtual o presencial, cuando se encontraban frente a un situación que no les permitía avanzar en alguna de las etapas del desarrollo de la APE.
- Activo: que realizaba un seguimiento ya sea en forma presencial o a través de las e-actividades propuestas del desarrollo de la APE, más allá de ser requerido por los grupo.

En su mayoría coincidieron en que cualquiera fuera la actitud del tutor, ellos podían adaptarse y no sintieron en el desarrollo de las APE la ausencia de los mismos. En este sentido, vale traer los aportes de Vigostky (1978), en relación al concepto por él definido como “*zona de desarrollo próximo*”, entendiéndola como aquella zona situada entre lo que un estudiante puede hacer solo y lo que puede lograr si trabaja guiado por un instructor o en colaboración con otros pares más avanzados. De esta forma, el rol del tutor no siempre está puesto en el docente, en el caso de esta experiencia el profesor y/o el ayudante de la cátedra, sino que ese rol lo puede desarrollar un par más avanzado es decir otro alumno miembro del grupo con más conocimientos sobre la

temática a tratar. Así puede pensarse que cuando el rol de tutor (docente / ayudante de la cátedra) es más pasivo, la tutoría del grupo se traslada al par más avanzado del grupo.

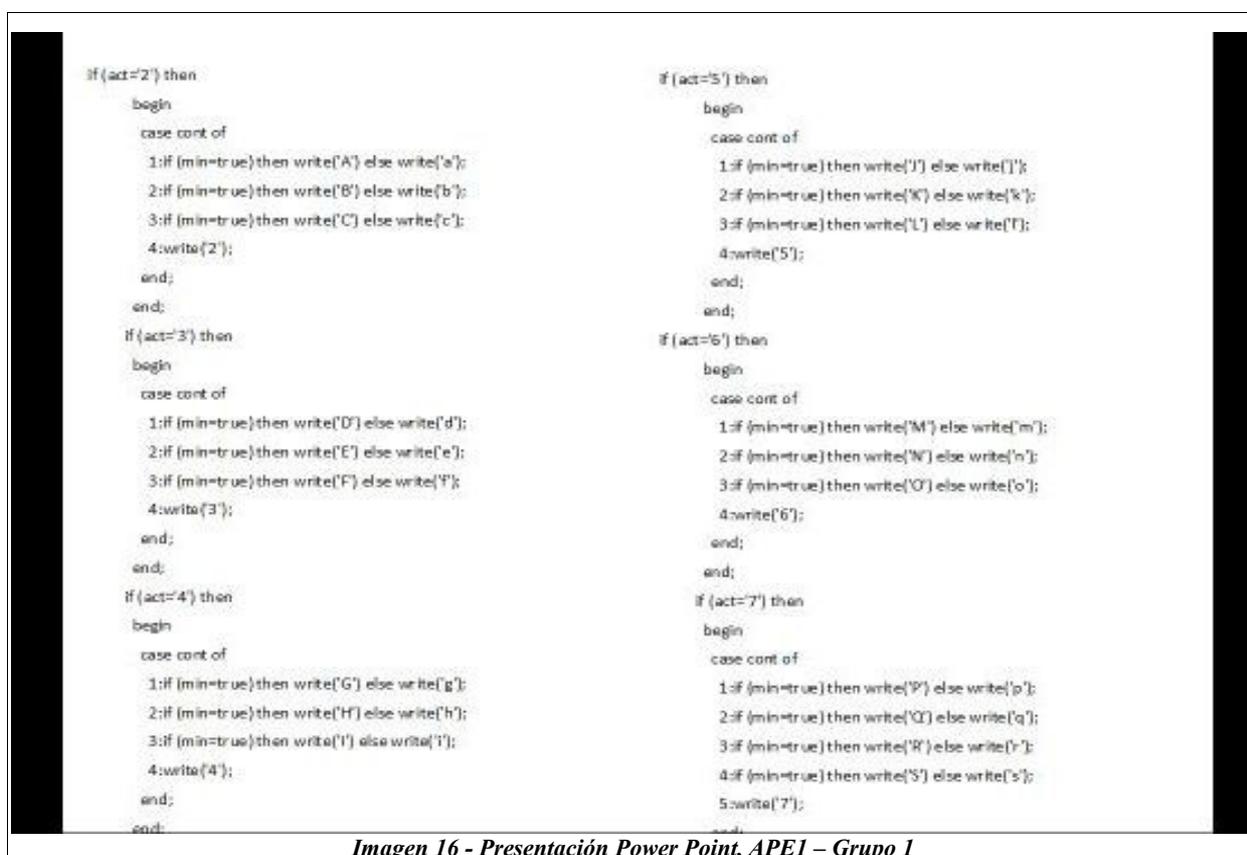
6.4 ANÁLISIS DE LAS HABILIDADES DESARROLLADAS EN EL MARCO DE LAS APE

La fase de presentación y defensa de las soluciones elaboradas por los equipos de trabajo para situaciones problemáticas planteadas en las APE, no se ha considerado solo como una etapa de cierre, sino como otro espacio que le da a los grupos la oportunidad de compartir sus aprendizajes, reforzando de esta forma el aprendizaje no solo del que esta exponiendo sino de todos los que están participando de la actividad. Por otra parte, cuando a través de las ponencias los alumnos articulan sus experiencias y resultados, comienzan a apropiarse del saber desde una perspectiva diferente. En cuanto a los resultados, el escuchar a otros que han encontrado soluciones similares, les permite reforzar ideas respecto de la temática tratada. Por otra parte, las ponencias no solo aportan beneficios para el alumno, sino también para los docentes ya que permiten identificar situaciones de aprendizaje que no están claras (lagunas de aprendizaje) y aportar entre todos a rellenar ese espacio.(Barkley et al. 1997)



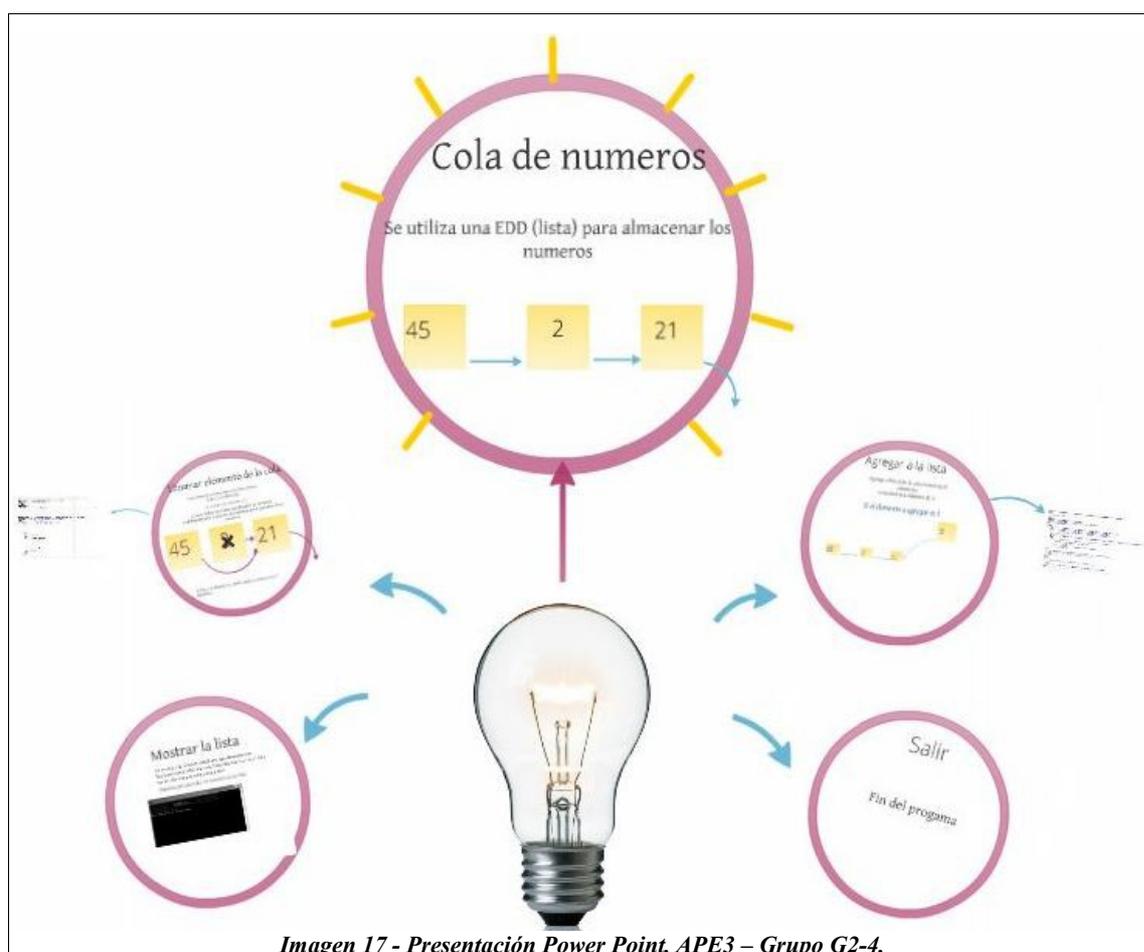
En cuanto a las ponencias que realizaron los diferentes grupos durante la fase de presentación y defensa de las soluciones planteadas, es importante destacar como evolucionaron las presentaciones, tanto en lo que respecta a la presentación digital en sí misma, como a la forma de comunicar y debatir con los docentes y pares. En este sentido, es importante resaltar que las

metas del trabajo colaborativo no solo académicas, sino también promover situaciones que permitan mejorar las relaciones sociales (Salinas; 2000), así resulta importante que los alumnos puedan desarrollar en la etapa de presentación y defensa de las APE, la habilidad de exponer con claridad las ideas y la predisposición al intercambio de opiniones. De esta forma, se desarrollan sentimientos de seguridad en los alumnos, que les permiten superar miedos e inhibiciones. Las habilidades que se busca promover en esta etapa del desarrollo de las APE, tiene en cuenta el hecho que la ciencia informática es una ciencia aplicada y en particular el desarrollo del software es una actividad compleja que necesariamente se desarrolla en grupos de trabajo colaborativos y multidisciplinarios, donde la capacidad de comunicación es un habilidad requerida en el ámbito laboral.



En cuanto a las presentaciones digitales, para las actividades APE1, APE2, se pudo observar que las ponencias incluyeron presentaciones tipo Power Point donde, y donde todas se caracterizaron por ubicar el foco en la implementación (código en el lenguaje de programación Pascal) de la solución al problema propuesto (imagenes 15,16) hasta pasar a la presentación de la última actividad APE3, donde el foco se puso más sobre el modelo de la solución que sobre el código y donde algunos grupos se animaron a innovar en el uso de aplicativos más complejos como es caso de Prezzi¹⁷ y que permiten la colaboración para el desarrollo de presentaciones. (Imagen 17).

17 <http://prezi.com/>



Es importante resaltar, la innovación que manifestaron los estudiantes en cuanto al uso de este recurso, en el sentido de su proyección como profesionales, teniendo en cuenta que la informática es una ciencia en continuo cambio, y la capacidad de adaptarse a nuevas tecnologías y formas de trabajo son habilidades requeridas en el ámbito laboral. Por otra parte, la producción del software requiere del desarrollo de la creatividad tanto en la búsqueda de algoritmos y modelos que resulten eficientes como en su enseñanza (González Hernández et al. 2004) y en este sentido, este grupo ha podido generar y exponer una síntesis de su trabajo a través de esta presentación y del uso de recursos TIC que no fueron vistos en la materia.

Respecto al desarrollo de los materiales que utilizaron en las ponencias, los alumnos sostienen que llegados a esta parte del desarrollo de la APE se incrementaba la participación, y entienden que esto era así porque habían logrado resolver el problema que planteaba la actividad y esto los ponía en una situación más homogénea con respecto a los conocimientos sobre la temática planteada.

7. CAPÍTULO VII – CONCLUSIONES

Al finalizar con el análisis de la implementación de la propuesta de enseñanza y aprendizaje objeto de este estudio y de la lectura crítica de las fuentes y la bibliografía consultada, es posible confirmar la importancia de la temática abordada. Sin pasar por alto, las distintas visiones y enfoques que existen sobre la cuestión, en este capítulo se despliegan las conclusiones y/o confirmaciones alcanzadas. Esta claro que, todo el trabajo hasta aquí desarrollado no es suficiente para cubrir la amplitud de la temática, ni desentrañar la complejidad de la misma.

En este estudio, se propuso abordar la enseñanza y aprendizaje de la programación de algoritmos para resolver problemas usando una computadora, entendiendo que la misma presenta una complejidad que precisa ser abordada desde el inicio de la formación de los futuros licenciados de la carrera de Sistemas. De este modo se observó, que a medida que los alumnos avanzan en la carrera profundizan los conceptos relacionados al desarrollo de un producto de software, que es en sí mismo un trabajo que a nivel laboral no se realiza en soledad sino en equipos de trabajo colaborativos. Por otra parte, la situación actual del desarrollo del software a nivel profesional, se encuentra atravesada no solo por las TIC sino por un nuevo paradigma de trabajo denominado Desarrollo Global del Software (DGS), donde una de sus características es que los miembros de un grupo de trabajo pueden estar distribuidos geográficamente (Monazor et al. 2011). Así resulta necesario desarrollar en los futuros profesionales habilidades y destrezas que les permitan incorporarse a ese nuevo espacio laboral. En este sentido, y atendiendo a los requerimientos de la actividad profesional, se observó la necesidad de utilizar estrategias colaborativas en el ámbito de la enseñanza y del aprendizaje de la programación. En este trabajo se presentó el diseño de una propuesta de enseñanza y aprendizaje. A continuación se detallan las conclusiones obtenidas para la primera experiencia de implementación de la misma.

7.1 CONCLUSIÓN I

Que los alumnos como parte de una comunidad necesitan de la interacción, de esta forma prefieren el trabajo grupal al individual, y que cuando este trabajo se realiza en colaboración estimula la participación de los alumnos y el sentido de responsabilidad para con el grupo. Así lo demuestran los resultados de las encuestas respecto al nivel de participación y asistencia a los encuentros tanto virtuales como presenciales organizados por los grupos, tanto a nivel personal como grupal. Así mismo, los resultados muestran que los alumnos perciben en su mayoría que ha existido a lo largo de la experiencia respeto por las opiniones de los demás. De esta forma, es

posible concluir que el trabajo desarrollado por los grupos en las APE puede verse como un proceso en el que todos los integrantes han tomado el compromiso de construir la solución a un problema en forma conjunta. Como se pudo observar en la experiencia, cuando el nivel de compromiso de los grupos es alto y parejo entre sus miembros, se confirman las afirmaciones de investigadores como Begonia Gross (2000), cuando dice que es posible definir ese proceso de aprendizaje como trabajo colaborativo.

Respecto al nivel de interacción de los grupos, y en sintonía con lo que expresa Collazos et al. (2001) aunque todos los grupos han tenido las mismas posibilidades de interacción a través del soporte TIC, esto no fue condición suficiente para garantizar que se produjeran situaciones que promovieran los mecanismos de aprendizaje en todos los grupos. Aquí resulta importante mencionar los distintos factores que influyeron sobre la propuesta:

- El grado de interés en la materia y en la carrera.
- El conocimiento y manejo que los alumnos tenían sobre los recursos TIC propuestos.
- Las situaciones problemáticas que planteaban las consignas de las APE
- El rol del tutor asignado a los grupos, como guía y orientador de los procesos de aprendizaje que implica el desarrollo de las APE.

En cuanto a los distintos niveles de conocimiento que se presentaron dentro de los grupos, respecto a las temáticas de las APE, los alumnos que indicaron sentirse en un nivel más alto en relación al de sus pares, consideraron que esta situación los obligó a realizar un esfuerzo cognitivo mayor para poder transmitir sus ideas al grupo, en particular en la fase inicial de las APE, donde se trabajó sobre el análisis y diseño de la solución. Así los resultados, coinciden con otras investigaciones que sostienen que la construcción del conocimiento en grupo, permite hacer más heterogéneo el aprendizaje, sin que esto signifique necesariamente perjudicar el aprendizaje de los alumnos “más aventajados”, pero sí beneficiando a los alumnos con menos conocimientos sobre los temas tratados en el desarrollo de la actividad colaborativa propuesta.

7.2 CONCLUSIÓN II

En cuanto a los recursos TIC utilizados en la experiencia, el uso del laboratorio virtual VPL no les resultó en absoluto difícil y pudieron apropiarse de su uso casi sin necesidad de asistencia de los tutores; el uso del foro al principio costó, pero luego pudieron apropiarse del mismo. No sucedió lo mismo con la herramienta wiki del entorno Moodle. Su manejo no les resultó ni fácil ni intuitivo. Así, es posible afirmar que es necesario un tiempo de maduración no solo sobre el uso de los recursos TIC provistos por el entorno de trabajo, en el caso de esta experiencia, Moodle; sino también sobre la forma de trabajo, que para la mayoría de los integrantes resultó novedosa. Este

tiempo de maduración, es necesario para que los alumnos logren apropiarse de las herramientas y adoptarlas como un recurso más para el aprendizaje, en particular para el desarrollo de las actividades didácticas colaborativas específicas del área de programación.

En este sentido se propone a futuro, el desarrollo de un taller pre-ingreso que trabaje sobre la introducción y el uso de las TIC aplicadas al aprendizaje.

7.3 CONCLUSIÓN III

Se ha podido corroborar que la estrategia de desarrollo de las APE en forma colaborativa, ha fomentado la participación de los alumnos, la capacidad de comunicación dentro y fuera del grupo. Así se ha puesto en evidencia en la etapa de presentación y defensa de las soluciones encontradas, donde se observó como mejoro en los alumnos la habilidad de exponer con claridad las ideas, la predisposición al intercambio de opiniones y la sensación de seguridad que les permitió superar miedos e inhibiciones que manifestaron al inicio de la experiencia.

7.4 DIFICULTADES ENCONTRADAS

- La administración de la plataforma. Cuando se comenzó con el diseño de la propuesta de enseñanza y aprendizaje en el marco de las APE, la Lic. en Sistemas contaba con la plataforma Moodle versión 1.9. Luego se migro a la versión 2.3 y la versión del laboratorio virtual (VPL) compatible con Moodle 2.X, recién se publicó en Julio de 2012. Por otra parte, en la nueva plataforma no se habilitaron las opciones que permitían la generación de estadísticas para los cursos, sino hasta el segundo cuatrimestre del año lectivo 2013.
- El tiempo disponible para el diseño y elaboración de los materiales, y la configuración de los ambientes de aprendizaje que constituyen las APE.
- Visiones encontradas respecto al modelo de trabajo a seguir en la cátedra. En cuanto a los objetivos que se planteaban desde la práctica, algunos docentes proponían un modelo más orientado a los exámenes, es decir a “preparar” a los alumnos para la acreditación, teniendo en cuenta que el reglamento de la materia exige para la acreditación de la cursada aprobar un examen parcial integrador, así desde esta perspectiva se planteaba el desarrollo de ejercitaciones que pusieran al alumno en situaciones similares a la de los exámenes. En otro sentido, las actividades de evaluación propuestas para las APE, como se expreso en el capítulo IV, intentaron promover un espacio de comunicación con los alumnos, donde a través de sus aportes (presentación, defensa), los docentes pudieran recolectar información que permitiera hacer ajustes en los procesos de enseñanza y aprendizaje. A la vez que los alumnos obtienen una base sólida de información que les

permite regular su propia actuación, potenciando sus fortalezas y trabajando sobre sus debilidades. De esta forma, uno de los puntos claves de la propuesta de enseñanza y aprendizaje que se presentó en esta investigación, está puesto en la retro-información que los grupos de trabajo obtienen en cada una de las fases del desarrollo y aquí se observa la importancia del docente como tutor y de los alumnos como pares avanzados dentro de los grupos.

7.5 TRABAJO FUTURO

Esta investigación es un primer avance en la implementación de estrategias de trabajo colaborativo mediadas por TIC en la enseñanza y aprendizaje de programación en los primeros años de la Lic. en Sistemas de la UNRN. A continuación se proponen algunas líneas de investigación y desarrollo a partir de los resultados de esta experiencia:

- Trabajar en el ajuste del diseño de la propuesta teniendo en cuenta las observaciones realizadas por los estudiantes y docentes de la cátedra. En particular sobre el tipo de problemas a resolver y su adecuación para el desarrollo de actividades colaborativas de programación y los recursos TIC como foros y wiki.
- Teniendo en cuenta que esta primer experiencia se realizó mayormente con alumnos ingresantes, podría resultar interesante promover la implementación de la propuesta, en otros cursos más avanzados como Programación II y III y /o Estructuras de datos.
- De igual forma, podría articularse esta forma de trabajo con las materias del área de Algoritmos y Lenguajes y de Ingeniería de Software. De manera que los alumnos, como futuros profesionales, puedan tener una visión más acabada del desarrollo colaborativo de un producto de software.
- Continuar con la investigación acerca de las herramientas que permiten la edición colaborativa de un programa computacional y que puedan incluirse en un EVEA.
- En función de los resultados obtenidos, surge la inquietud, acerca de si esta propuesta de enseñanza y aprendizaje, no podría re-pensarse de manera que pueda convertirse en una herramienta que permita sostener a los alumnos a lo largo de la cursada. El desarrollo de actividades colaborativas en programación favorece el futuro desarrollo profesional de los estudiantes, en el sentido que se les presenta situaciones similares a las que van a tener que desarrollar.
- Revisar la propuesta de la cátedra para incorporar las APE como un ítem dentro de la evaluación, realizando una evaluación sumativa, donde el examen parcial sea un

elemento más pero no de carácter definitorio.

- Mejorar los instrumentos que se utilizaron en la investigación para recolectar información, de manera que permitan determinar los procesos que tienen lugar en el desarrollo colaborativo de las prácticas de programación.
- Utilizar herramientas colaborativas para mostrar la producción final de todo el grupo. Se busca generar un espacio de muestra de las producciones de cada grupo donde todos puedan acceder y colaborar en la construcción de este espacio. De esta forma todos los alumnos tendrían acceso a las producciones de sus compañeros de una manera colaborativa y visual, por ejemplo usando Murally¹⁸ o paddlet¹⁹.

18 [Mural.ly](https://mural.ly) - Visual Collaboration for Creative People

19 [Padlet](http://padlet.com/). <http://padlet.com/>

BIBLIOGRAFÍA

Alania Vera, R. C, Diez Arenas C. A., Pinglo Puertas J. C. (2012) . Estrategias de enseñanza y estilos de aprendizaje en los alumnos del curso Introducción a la Algoritmia: caso Cibertec . Informe de Seminario de Grado del Magister en Docencia para la Educación Superior. Universidad Andres Bello. Facultad de Humanidades y Educación. Chile. Disponible: <http://hdl.handle.net/10757/274492>. Junio 2013

Araiza Vazquez, Maria D. Jesus, Dorfer Claudia, Castillo Corpus, Rosalinda. (2011). Una experiencia de desarrollo y aplicación de competencias profesionales en los estudiantes de la licenciatura de tecnologías de la información. Universidad de Nueva León. XIV Encuentro Iberoamericano de Educación Superior a Distancia en la Universidad Técnica Particular de Loja. Disponible: <http://memorias.utpl.edu.ec/sites/default/files/documentacion/aiesad2011/utpl-aiesad2011-una-experiencia.pdf>

Barkley Elizabeth F., Cross Patricia, Howell Major C. (2007) Técnicas de Aprendizaje Colaborativo:Manual para el profesorado. Ediciones Morata S.L.. Madrid. ISBN 978-84-7112-5

Beltran Silva, E.E., Morales Hernández, I. (2011). Autonomía y Trabajo Colaborativo. XII Congreso Internacional de Teoría de la Educación. Universidad de Barcelona. Disponible: www.cite2011.com/Comunicaciones/Escuela/152.pdf Agosto 2012

Ben-Ari, M. (2001). Constructivism in Computer Science Education. Journal of Computers in Mathematics and Science Teaching, 20(1), 45-73. Norfolk, VA: AACE. Disponible: <http://stwww.weizmann.ac.il/department40/publications/BenAri/ben-ari2001.pdf> . Junio 2012.

Boyle, T. (2000) Constructivism: A Suitable Pedagogy for Information and Computing Sciences?. Disponible: <http://www.ics.ltn.ac.uk/pub/conf2000/Papers/tboyle.htm> . Junio 2012.

Bravo C., Duque R., Gallardo J., García J., Garcia P. (2007). Groupware System for Distributed Collaborative Programming: Usability Issues and Lessons Learned. TOMAG+REMIDI 2007 Proceedings. The Seventh International Conference of Computer Ethics: Philosophical Enquiry.

Bravo C., Marcelino, M.J., Gomes A., Esteves, M., Mendes, J. A. (2005). Integrating Educational Tools for Collaborative Computer Programming Learning . In Journal of Universal Computer Science, vol. 11, no. 9 (2005), 1505-1517. Disponible: http://www.jucs.org/jucs_11_9/integrating_educational_tools_for

Bravo C., Redondo M.A., Ortega M. (2004). Aprendizaje en grupo de la programación mediante técnicas de colaboración distribuida en tiempo real. In Proceedings of V Congreso Interacción Persona Ordenador, Lleida, Spain, 2004, 351-357.

Brown John Seely; Collins Allan; Duguid Paul. (1989) . Situated Cognition and the Culture of Learning. Educational Researcher, Vol. 18, No. 1. (Jan. - Feb., 1989), pp. 32-42.

Byron Weber Becker. (2007). Java: Learning to Program with Robots. COPYRIGHT 2007 Thumbbody's Thinking Inc. ISBN 0-619-21724-3.

Cabrera Murcia Elsa P. (2004). Aprendizaje colaborativo soportado por computador (cscl): su estado actual. Revista Iberoamericana de Educación (ISSN: 1681-5653). Disponible: http://campus.usal.es/~ofeees/NUEVAS_METODOLOGIAS/COLABORATIVO/729Cabrera1085B15D.pdf Julio 2012

Calzadilla María Eugenia. (2002). Aprendizaje colaborativo y tecnologías de la información y la comunicación. OEI-Revista Iberoamericana de Educación. ISSN: 1681-5653

Camilloni, Alicia. (1998). Sistemas de calificación y regímenes de promoción. En: Camilloni et al. La evaluación de los aprendizajes en el debate didáctico contemporáneo. Buenos Aires: Ed. Piados

Carro Pastor. (2007). Ventajas del uso de la tecnología en el aprendizaje colaborativo. Revista

Iberoamericana de Educación (ISSN: 1681-5653) n.o 41/4. Febrero de 2007

Castañeda Quintero, I., López Vicent, P. (2007) "Entornos Virtuales de Enseñanza Aprendizaje Libres: MOODLE". En Prendes Espinosa, M. P. Herramientas Telemáticas Para La Enseñanza Universitaria En El Marco Del Espacio Europeo De Educación Superior . Grupo de Investigación de Tecnología Educativa. Universidad de Murcia. CD – ROM. ISBN: 978-84-611-7947-3

Cataldi, Zulma, Cabero Almenara, Julio (2006). Los Aportes de la Tecnología al Aprendizaje Grupal Interactivo: La Resolución de Problemas a través de Foro de Discusión y de Chat. Pixel-Bit: Revista de medios y educación, ISSN 1133-8482, N°. 27, 2006, págs. 115-130

Cernadas, María A. ,Cuenca Pletsch, Liliana R. (2009) . Las TIC como medio para evitar el desgranamiento temprano. En IV Congreso de Tecnología en Educación y Educación en Tecnología. <http://sedici.unlp.edu.ar/handle/10915/19000>

Chalk, P. (2000). Apprenticeship learning of software engineering using Webworlds. ITiCSE '00 Proceedings of the 5th annual SIGCSE/SIGCUE ItiCSE conference on Innovation and technology in computer science education Pages 112-115 . Disponible: <http://ebookbrowse.com/p112-chalk-apprenticeship-learning-of-software-engineering-using-webworlds-pdf-d289928416> Mayo 2012

Champredonde, Raul, Palacios, Ainchil (2000). "Teaching Experiences in Programming Using the Visual Da Vinci Language", publicado en First International Congress on Tools for Teaching Logic", Universidad de Salamanca, Junio 2000.

Chesñear Carlos. (2000). Utilización de Mapas Conceptuales en la Enseñanza de la Programación. Universidad Nacional del Sur. Disponible lidedc.cs.uns.edu.ar/~cic/cs_education/2000_jornadas.pdf Julio 2013

Collazos O, César Alberto, Guerrero Luis, Vergara, Adriana. (2001). Aprendizaje Colaborativo: un cambio en el rol del profesor. <http://www.dcc.uchile.cl/~luguerre/papers/CESC-01.pdf> Mayo 2012

Collazos, C. A. & Mendoza, J. (2006). Cómo aprovechar el Aprendizaje Colaborativo en el aula. En Revista: Educación y Educadores, Dic 2006, vol.9, no.2, p.61-76. ISSN 0123-1294

Cordoba Torrecilla, J., Cuesta Morales P. (2009). Adaptando un sistema de Wikis para su uso educativo. XV Jornadas de Enseñanza Universitaria de la Informática. JENUI 2009. Barcelona, 8-10 de julio de 2009 ISBN: 978-84-692-2758-9. Barcelona: AENUI, pp. 209–216, 2009

Costelloe, E. (2001). Teaching Programming. The State of the Art. Department of Computing, Institute of Technology Tallaght, Dublin 24. CRITE Technical Report, 2004a. Disponible https://www.scss.tcd.ie/disciplines/information_systems/crite/crite_web/publications/sources/programmings1.pdf Abril 2012

Dasso Aristides, Funes Ana (2012). Notas de clase para el curso Programación I. Área de Programación y Metodologías de Desarrollo del Software Departamento de Informática Facultad de Ciencias Físico-Matemáticas y Naturales Universidad Nacional de San Luis Argentina . Disponible <http://proguno.unsl.edu.ar/t1.pdf>

De Giusti A., Madoz, M.C., Bertone, R., Naiouf, M., Lanzarini, L., Gorga, G., Russo, C. (2001). *Algoritmos, Datos y Programas* . Buenos Aires. Prentice Hall. ISBN: 987-9460-64-2

De Giusti, A. E., Madoz, M. C., Gorga, G., Feierherd, G. E., & Depetris, B. O. (2003). Enfoques y herramientas en la enseñanza de un primer curso de computación (CS1). En IX Congreso Argentino de Ciencias de la Computación.

Diaz Roca, M. Rodriguez del Pino, J.C., Hernández Figuero, Z., Vicente, C. M. El Gestor de Coevaluación Orientado Grupos. Una herramienta de apoyo a la participación del alumno en el proceso de evaluación. 7ª Conferencia Ibérica de Sistemas y Tecnologías de Información. CISTI 2012, Madrid. (2012) <http://www2.dis.ulpgc.es/~mdiaz/GestorCoevaluacionOrientadoGrupos>.

Dillenbourg, Pierre. (1999). What do you mean by collaborative learning?. En P. Dillenbourg (Ed) Collaborative – learning: Cognitive and Computational Approaches. 1 – 19. Oxford:

Elsevier.

Esteves M., Morgado L., Martins P., Fonseca B. "The use of Collaborative Virtual Environments to provide student's contextualisation in programming". En: Proceedings of m-ICTE 2006. (2006)

Ferreira Szpiniak, A., y Rojo, G. A. (2006). Enseñanza de la programación. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología. Disponible: <http://sedici.unlp.edu.ar/handle/10915/19182>

Gallego M., Gortázar F. (2009). EclipseGavab, un entorno de desarrollo para la docencia online de la programación. JENUI 2009. Barcelona, 8-10 de julio de 2009 ISBN: 978-84-692-2758-9. <http://jenui2009.fib.upc.edu/>

García, J.C., González, M.L., Zanfrillo, A.I. (2011). Desgranamiento universitario: perspectiva estudiantil en Ingeniería. En XI Coloquio Internacional sobre Gestión Universitaria en América del Sur. Florianópolis. Septiembre 2011. Disponible: <http://repositorio.ufsc.br/xmlui/handle/123456789/25923>

García-Valcárcel Muñoz-Repiso Ana, Hernández Martín Azucena, Recamán Payo Adriana (2012) .La metodología del aprendizaje colaborativo a través de las TIC: una aproximación a las opiniones de profesores y alumnos. Revista Complutense de Educación ISSN 1130-2496. ISSN-e 1988-2793. Vol (23) Nro 1 – 2012. Disponible <http://revistas.ucm.es/index.php/RCED/article/view/39108> Agosto 2012

Gary Stager (2003) "The Case for Computing" . "Snapshots!, Educational insights from the Thornburg Center", 2003, Thornburg Center (<http://www.tcpd.org/Stager/Stager.html>)

Gonzalez Carella Maria I., Zanfrillo Alicia I. (2007). Entornos de aprendizaje colaborativos para el desarrollo de estudios de posgrado. VII Coloquio Internacional sobre Gestión Universitaria en América del Sur. Universidad Nacional de Mar Del Plata. Disponible: http://www.inpeau.ufsc.br/wp/wp-content/Bd_documentos/2145.pdf Octubre 2012

González Hernández, W. , Sentí Estrada, V., Martíne, M. (2004). Contribución al desarrollo de la creatividad a través de la enseñanza de la programación. Revista Pedagogía Universitaria Vol. 9 No. 3 2004. CUBA. ISSN 1609-4808. <http://169.158.24.166/texts/pd/1894/04/3/189404308.pdf>.

González de Rivera Fuentes, M., Paredes Velasco, M. (2008). Aprendizaje con programación Colaborativa. Número 2008-02. Serie de Informes Técnicos DLSI1-URJC. ISSN 1988-8074. Departamento de Lenguajes y Sistemas Informáticos I Universidad Rey Juan Carlos

Guevara Mora, Gabriela (2010). Aprendizaje Basado En Problemas Como Técnica Didáctica Para La Enseñanza Del Tema De La Recursividad. InterSedes: Revista de las Sedes Regionales, vol. XI, núm. 20, 2010, págs. 142-167. Universidad de Costa Rica Ciudad Universitaria Carlos Monge Alfaro, Costa Rica

Gros Begonia (2000). El Aprendizaje Colaborativo a través de la Red: Límites y Posibilidades. En [Aula de innovación educativa](#), ISSN 1131-995X, N° 162, 2007, págs. 44-50

Grosclaude E., Bertogna L., López Luro F., Zanellato C., Sánchez L., Rodríguez J., Del Castillo R. (2006). Experiencia con Laboratorio Remoto Colaborativo. I Congreso de Tecnología en Educación y Educación en Tecnología. Pags. 441-445. Disponible sedici.unlp.edu.ar/handle/10915/19252 Septiembre 2012

Jean Greyling. (2009). Developing a Set of Requirements for Algorithm Animation Systems. International Journal of Computing and ICT Research, Special Issue Vol. 3, No. 1, pp. 83-89. Disponible: <http://www.ijcir.org/Special-Issuevolume3-numbe1/article9.pdf>

Jiménez Builes J.A., Pavony Meneses M., Alvarez Serna, A. F. (2008). Entorno de integración de PBL y CSCL para la enseñanza de algoritmos y programación en ingeniería. En revista Avances en Sistemas e Informática, ISSN 1909-0056, Vol. 5, N°. 3, 2008 , págs. 189-194. Disponible: <http://www.revista.unal.edu.co/index.php/avances/article/view/10112> Junio 2012

Johnson, David W., Johnson, Frank P. (1997). Joining Together: Group Theory and Group Skills. Needham Heights, MA: Allyn & Bacon.

Johnson, David W., Johnson, Frank P. (1999). Learning Together and Alone: Cooperative, Competitive, and Individualistic Learning. Needham Heights, MA: Allyn & Bacon.

Johnson, David W., Johnson, Roger. (1999) Aprender juntos y solos. Editorial Aique. Bs. As.

Johnson, David , Johnson, Roger. , Holubec, Edythe J. (1999). El Aprendizaje cooperativo en el aula. Paidós Educador. Buenos Aires

Joyanes Aguilar Luis, Zahonero Martínez, Ignacio (2005). Programación en C: metodología, algoritmos y estructura de datos. McGraw-Hill Interamericana de España S.L

Kölling, M., Quig, B., Patterson, A. and Rosenberg, J. (2003), The BlueJ system and its pedagogy, Journal of Computer Science Education, Special issue on Learning and Teaching Object Technology, Vol 13, No 4, Dec 2003.

Kölling, M (2008). Using BlueJ to Introduce Programming. In Reflections on the Teaching of Programming. Lecture Notes in Computer Science Volume 4821, 2008, pp 98-115 <http://www.cs.kent.ac.uk/pubs/2008/2697/content.pdf> Abril 2012

Lahtinen E, Ala-Mutka K, et al.(2005) A Study of the Difficulties of Novice Programmers. 10Th annual SIGCSE conference on Innovation an technology in computer science education ItiCSE '05

Lillo Zúñiga, Felix Gabriel (2012). Revista de Psicología Universidad Viña del Mar 2012, Vol. 2, N° 4 109 - 14 2 ISSN 0719-888X <http://sitios.uvm.cl/revistapsicologia/revista/04.05.aprendizaje.pdf>

Linder, et al, (2001) "Facilitating Active Learning With Inexpensive Mobile Robots". Journal of Computing in Small Colleges,16, 4. Disponible <http://delivery.acm.org/10.1145/380000/378656/p21-linder.pdf?key1=378656&key2=7062133701&coll=GUIDE&dl=ACM&CFID=15363060&CFTOKEN=16364368>

Junio 2012

Lovos, Edith, Gibelli, Tatiana. (2012). La Simulación como Recurso de Enseñanza-Aprendizaje en el Análisis de Eficiencia de Algoritmos. Revista Iberoamerica de Tecnología en Educación y Educación en Tecnología (TE&ET) Nro 7 ISSN 1850-9959 RedUNCI-UNLP .Pag. 36-41

Lovos, Edith; González, Alejandro Héctor; Bertone, Rodolfo Alfredo. (2013). Experiencia de utilización de herramientas colaborativas para la enseñanza y el aprendizaje de la programación de computadoras. En Actas del XVIII Congreso Argentino de Ciencias de la Computación. Universidad CAECE (sede Mar del Plata). Octubre 2013. ISBN 978-987-23963-1-2

Lucero, Mariana (2003). Entre el trabajo colaborativo y el aprendizaje colaborativo. Revista Iberoamericana de Educación (ISSN: 1681-5653). Vol. 36, Núm. 1, pág. 23-200. Disponible: http://www.Capmpus.oei.org/revista/de_los_lectores/528/Lucero.pdf

Madoz, M.C., Gorga, G., Russo, C. (2005). Análisis del Impacto de las TIC's en el proceso de aprendizaje de alumnos universitarios de nivel inicial. Congreso de Tecnologías de la Información y la Comunicación en la Enseñanza de las Ciencias. TICEC 05. La Plata, 29 de Septiembre del 2005.

Maldonado C., Odetti M. A., Muñoz, A., Marciszack, M. (2009). Programa de Innovación educativa para captación de aspirantes y contención de los alumnos de la carrera de Ingeniería de Sistemas de Información. En IV Congreso de Tecnología en Educación y Educación en Tecnología. Disponible: http://sedici.unlp.edu.ar/bitstream/handle/10915/19009/Documento_completo.pdf?sequence=1

Maldonado Pérez, Marisel.(2007). El trabajo colaborativo en el aula universitaria. Revista Laurus, vol.13 nro. 23. Universidad Pedagógica Experimental Libertador. Caracas Venezuela. ISSN 1315-883X

Materson T. R, Meyer, M. R. (2001). Journal of Computing Sciences in Colleges Volume 16 Issue 4, 2001. Pages 74-86

Matthíasdóttir, Á. (2006). How to teach programming languages to novice students? Lecturing or not?, Proceedings of the International Conference on Computer Systems and Technologies, June 15-16, University of Veliko Tarnovo, 2006, Bulgaria.

Meneses Benítez, Gerardo. (2007). NTIC, INTERACCIÓN Y APRENDIZAJE EN LA UNIVERSIDAD. UNIVERSITAT ROVIRA I VIRGILI. ISBN:978-84-691-0359-3/DL: T.2183-2007

Meyer Bertrand. (2003). The Outside-In Method of Teaching Introductory Programming, in Manfred Broy and Alexandre V. Zamulin, eds., Ershov Memorial Conference, volume 2890 of Lecture Notes in Computer Science, pages 66-78. Disponible: <http://se.ethz.ch/~meyer/publications/teaching/teaching-psi.pdf>

Monsalves González Sara. (2011). Estudio sobre la utilidad de la robótica educativa desde la perspectiva del docente. Revista de Pedagogía, vol. 32, núm. 90, enero-junio, 2011, pp. 81-117, Universidad Central de Venezuela, Venezuela. Disponible: <http://www.redalyc.org/articulo.oa?id=65920055004>

Mora Miguel Angel, Saiz, Francisco, Moriyón Roberto. (2002) .Aprendizaje colaborativo guiado: Fundamentos y aplicaciones. Revista de Enseñanza y Tecnología – Septiembre-Diciembre 2002

Morales, Patrica, Landa, Victoria. (2004). Aprendizaje basado en problemas, en Revista Theoria, Vol.13. Págs. 145-157. Disponible: http://campus.usal.es/~ofeees/NUEVAS_METODOLOGIAS/ABP/13.pdf

Naharro-Berrocal, F., Velázquez-Iturbide, J.Á., Pareja-Flores, C. and Medina-Sánchez, M.Á. Valoración de la eficacia de las animaciones de algoritmos en el aprendizaje de la programación funcional. Unpublished report, 2002. Disponible <http://vido.escet.urjc.es/winhipe/docs/209-erpmeuiona.pdf>

Naps, T.L., et al. (2003): Evaluating the Educational Impact of Visualization. Proc. ITiCSE 2003, Thessaloniki, Greece. ACM Press

Newman, D., Goldman, S.V., Brienne, D., Jackson, I., Magzamen, S. (1989) Computer mediation of collaborative science investigations. Journal of Educational Computing Research, 5 (2), pp. 151-166. (1989)

Okamoto, T., Kayama, M. and Cristea, A. (2001). Proposal for Collaborative Learning Standardization. En Proceedings of the IEEE International Conference on Advanced Learning Techniques -ICALT'01, 267 - 268

Ovalle Carranza Demetrio Arturo, Jiménez Builes Jovani Alberto, Collazos Ordóñez César Alberto. (2003) Modelo de awareness en el CSCL ALLEGRO utilizando la arquitectura blackboard. Ingeniería e Investigación, Bogotá, v. 26, n.3, dez. 2006. Disponible: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-56092006000300008&lng=pt&nrm=iso

Panitz Theodore, Panitz Patricia. (1998). Encouraging the use of collaborative learning in Higher Education. NY: Garland Publishing

Perales, P. E. C., Valladares, A. G., Palmer, Á. L. C. (2010). El aprendizaje colaborativo: una opción para los docentes de la disciplina (...). Disponible: <http://revistas.mes.edu.cu/greenstone/collect/repo/import/repo/20090820/978959161092853.pdf>.

Pérez Pérez, J. R., Paule Ruiz, J.M., Del Puerto M., Cueva Lovelle J. M. (2006). Capítulo 3. Sistemas orientados a la mejora de la calidad del software. En congreso IV International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE2006)

Restrepo Gomez, B. (2005). Aprendizaje basado en problemas (ABP):una innovación didáctica para la enseñanza universitaria. Educadores, Bogotá, Colombia, v. 8, p. 9-19, 2005. Disponible redalyc.uaemex.mx/redalyc/pdf/834/83400803.pdf. Junio 2012

Rincón C., Acurero A., Bracho D. (2008). Aspectos de diseño de un entorno de programación colaborativo. Enla@ce Revista Venezolana de Información, Tecnología y Conocimiento. 5 (3), 11-23 ISSN: 1690-7515

Rodríguez del Pino, J.C., Royo Rubio E., Hernandez Figueroa. (2010). VPL: Laboratorio virtual de programación para Moodle. En Actas de las XVI Jornadas de Enseñanza Universitaria de Informática, Jenui 2010, pags. 429-435, Santiago de Compostela, Julio 2010. Disponible:

http://www.di.uniovi.es/~juanrp/investigacion/tesis/2%20Tesis_SICODE_Estado_del_arte.pdf Junio 2012

Román Graván P. (2004). Posibilidades formativas de las herramientas groupware. El aprendizaje colaborativo en la educación. II Congreso sobre la Educación en Internet o Internet en la Educación: Madrid, octubre de 2003. ISBN: 84-369-3827-5 (978-84-369-3827-2). Disponible: http://tecnologiaedu.us.es/tecnoedu/images/stories/pedro_roman_groupware.pdf Agosto 2013

Román Graván P. (2004). *Análisis comparativo de diversos entornos de trabajo colaborativo y su aplicación en la enseñanza*. Revista Comunicación y Pedagogía Nro. 194, pp.39-43. ISSN: 1136-7733 Disponible: <http://tecnologiaedu.us.es/tecnoedu/images/stories/pedro/2004-articulo1-proman-revista-comunicacionypedagogia.pdf>

Sacristán Gimeno J., Pérez Gómez, A. I. (1996). "Comprender y transformar la enseñanza." Ediciones Morata. España. Quinta Edición. Pp115-136

Salinas Silvia. (2008). Software para trabajo colaborativo y bibliotecas. E-LIS. E-prints in Library and Information Science; 8 2008. Disponible: http://eprints.rclis.org/14721/1/Software_para_Trabajo_Colaborativo_y_Bibliotecas1.pdf

Sautu R., Boniolo, P., Dalle P y Elbert R. (2005). Manual de metodología. Construcción del marco teórico, formulación de los objetivos y elección de la metodología. Buenos Aires. CLACSO. ISBN 987-1183-32-1

Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer - supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.), Cambridge handbook of the learning sciences (pp. 409 - 426). Cambridge, UK: Cambridge University Press. Disponible http://GerryStahl.net/cscl/CSCL_English.pdf

Teague, D. and Roe, P. (2008). *Collaborative learning - towards a solution for novice programmers*. In Proc. Tenth Australasian Computing Education Conference (ACE 2008), Wollongong, NSW, Australia. CRPIT, 78. Simon and Hamilton, M., Eds. ACS. 147-154.

Tecnológico de Monterrey. Dirección de Investigación e Innovación Educativa. (2000). Las Técnicas Didácticas en el Modelo Educativo del Tecnológico de Monterrey. Septiembre, 2000 Disponible: http://sitios.itesm.mx/va/dide/docs_internos/inf-doc/tecnicas-modelo.PDF

Vásquez Balderrama Carlos. (2006). Tema 2 – Algoritmos. http://moodle2.unid.edu.mx/dts_cursos_md/lic/IC/EA/AM/06/Algoritmos.pdf

Vázquez Mariño, Iria . (2011). Aplicación de teorías constructivistas al uso de actividades cooperativas en la clase de E/LE. RedELE nº 21; 18 de febrero de 2011. http://www.mecd.gob.es/dctm/redele/Material-RedEle/Revista/2011_21/2011_redELE_21_08Vazquez.pdf?documentId=0901e72b80dcdafa

Vega, Gim Ivy, Rojo Muñoz Bernardo (1999). Educación a Distancia y aprendizaje colaborativo. Centro Educación a Distancia. Universidad Católica Del Norte, Chile. En : III Jornadas de Educación a Distancia – Mercosur '99, 1999, Chile. Disponible: <http://estrategiaslxuvm.wikispaces.com/file/view/Aprendizaje+colaborativo.pdf>

Vihavainen, A., Paksula, M., and Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. In Proceedings of the 42nd ACM technical symposium on Computer science education, SIGCSE '11, pages 93–98. ACM. Disponible <http://www.cs.helsinki.fi/u/ikiistal/harjoitus/wad/p93-vihavainen.pdf>

Villalobos, J., Casallas, R. (2006). Teaching/Learning a First Object-Oriented Programming Course outside the CS Curriculum. 10th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts - ECOOP (European Conference on Object-Oriented Programming)

Villena Moya, Agustín Antonio (2008). Un Modelo Empírico de Enseñanza de las Metodologías Ágiles. Tesis de Postgrado. Universidad de Chile. Facultad de Ciencias Físicas y Matemáticas, 2008. Disponible <http://www.scribd.com/doc/39816155/Un-modelo-empirico-de-ensenanza-de-las>

metodologias-agiles

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.

Waldegg, G. (2002). El uso de las nuevas tecnologías para la enseñanza y el aprendizaje de las ciencias. *Revista Electrónica de Investigación Educativa*, v.4 n.1. Disponible : <http://redie.uabc.mx/vol4no1/contenido-waldegg.html>

Williams, I.A., Kessler, R.R. (2000). All I really need to know about pair programming learned in kindergarten. *Communications of the ACM*, Vol. 43, No. 5

Zañartu Correa Luz Maria.(2003). Aprendizaje colaborativo: una nueva forma de Diálogo Interpersonal y en Red. *Revista Digital de Educación y Nuevas Tecnologías*. Año V, Nro 28.

ANEXOS

ANEXO I - CONSIGNAS APE

APE NRO 1 – GRUPOS PARES - LA TAPADITA

Este documento presenta el problema que deberá resolverse para la primer APE de las actividades de laboratorio 2013

FORMA DE ENTREGA

Las entregas se realizarán a través del aula virtual y por los siguientes recursos del aula. Se indican en **negrita** los recursos que deben utilizarse en forma obligatoria.

| ETAPA | CARACTERÍSTICAS | RECURSOS AULA VIRTUAL |
|------------------------|--|--|
| Análisis y Diseño | Los alumnos se reúnen en grupos en forma sincrónica o asincrónica y analizan el problema. Se plantean diferentes modelos. Se diseña el algoritmo. | Foro Mensajería interna chat WIKI |
| Implementación | Los alumnos en forma colaborativa implementan el algoritmo diseñado usando Pascal. | Foro Mensajería interna VPL |
| Presentación y Defensa | Los alumnos elaboran una presentación tipo Power Point en la que se presenta la solución al problema. Luego en forma presencial realizan la presentación y defensa de su trabajo | Foro Mensajería interna WIKI Tarea |
| Evaluación | Los alumnos realizan una evaluación del proceso de aprendizaje a nivel personal, grupal y del tutor | Encuestas |

Fecha de Entrega

Se debe entregar hasta la etapa de implementación hasta el día 06/04/2013 hasta las 23:30 hs.

Problema

La tarea consiste en escribir un programa que simule el juego conocido como *La Tapadita*. En su versión clásica, La Tapadita tiene dos participantes: el *apostador* y el *mezclador*. El mezclador cuenta con tres copas (o cartas o cáscaras de nuez) y esconde en una de ellas una pequeña pelota (o moneda). Luego, el apostador indica la copa en la que piensa que está la pelota, y apuesta una cierta cantidad. Si la pelota se encuentra en la copa seleccionada, el mezclador le pagará al apostador el monto apostado. En caso contrario, el apostador le pagará al mezclador el monto apostado.

En nuestro problema el usuario tendrá actuará como apostador y el programa como mezclador. El usuario comenzará el juego con un saldo de 100 pesos. Al iniciar el programa, se le preguntará al usuario dónde está la moneda.

0 0 0

1 2 3

Copa?: _

El usuario ingresará un número del 1 al 3 que representa la copa en la que piensa que está la moneda. Luego el programa le solicitará el monto que desea apostar. La pantalla debería mostrar:

Apuesta?: _

El usuario ingresará un *número entero positivo* que representa la cantidad que desea apostar. La cantidad ingresada deberá ser menor o igual que el saldo del que dispone al momento de realizar la apuesta.

Si la copa elegida es la que contiene la moneda, el saldo del usuario se incrementará con el valor de la apuesta. Suponiendo que el usuario eligió la copa 2, la moneda se encuentra en la copa 2, el usuario apostó 100 pesos y esta es su primera apuesta, la salida del programa será la siguiente:

O . O

Ha acertado

Saldo: 110

Si la copa elegida no contiene a la moneda, se le restará la cantidad apostada al saldo del usuario. Suponiendo que el usuario eligió la copa 2, la moneda se encuentra en la copa 1, el usuario apostó 10 pesos y esta es su primera apuesta, la salida del programa será la siguiente:

. O O

No ha acertado

Saldo: 90

Es común que los mezcladores engañen a los apostadores en este juego. Por eso nuestro programa, tendrá la opción de simular también ese comportamiento, para esto **no deberá permitir** que el usuario acierte más de dos veces seguidas. Para lo cual será necesario que el programa no muestre la verdad, así la salida en estos casos será diferente. Suponiendo que el usuario eligió la copa 2, la moneda se encuentra en la copa 2, el usuario apostó 10 pesos, su saldo es de 150 pesos y esta es la tercera vez seguida que el usuario acierta, la salida del programa será la siguiente:

O O

No ha acertado

Saldo: 140

Nota: observe que el despliegue de las copas cuando el programa engaña, difiere del despliegue de las copas al no acertar la apuesta.

Las apuestas se repiten hasta que el saldo del usuario llegue a 0. El programa terminará luego de mostrar el siguiente mensaje:

Gracias por jugar

Se pide que diseñe un algoritmo y lo implemente en Pascal que se comporte de acuerdo a la explicación del juego. El programa debe llamarse TapaditaGX, donde X indica el número del grupo. El programa debe permitir que el usuario elija si quiere jugar limpio o con trampa.

Se puede utilizar los siguientes conceptos visto en las clases teóricas y prácticas:

- Tipos de datos:
 - Integer
 - Boolean
 - Char
- Instrucciones:
 - Asignación
 - Entrada y Salida(read, write, WriteLn, ReadLn)
 - Secuencia
 - Selección (if, case)
 - Repetición (for, while, repeat)

En esta parte NO SE PUEDE otros subprogramas salvo el indicado en el archivo sorteo.pas.

DATOS DE ENTRADA

Como pre-condición se asume que los datos de entrada son correctos. No se requiere realizar ningún control sobre los mismos.

DÓNDE UBICAR LA MONEDA

La ubicación de la moneda debe ser aleatoria y no coincidir forzosamente entre ejecución y ejecución. Para ello se proporciona el código de una función (sortearCopa) en el archivo sorteo.pas, que resuelve el problema de dónde esconder la moneda para cada ronda (o apuesta). Es decir, recibe el número de ronda y devuelve el número de copa donde esconder la moneda.

Para incluir el código que está en el archivo sorteo.pas, se debe agregar la línea {\$INCLUDE sorteo.pas} luego de la declaración de variables y antes del begin del programa principal, como en el siguiente ejemplo:

```
Program laTapadita;  
Const ...;  
Var ...;  
{$INCLUDE sorteo.pas}  
Begin  
    ...  
End.
```

El archivo sorteo.pas se puede descargar de la carpeta APE1-Recursos. Luego tienen que guardarlo en la **misma** carpeta donde está el código fuente del programa que deben realizar.

El código fuente que Uds. entreguen debe:

- tener la línea {\$INCLUDE sorteo.pas} y **no** el código copiado y pegado del archivo sorteo.pas.
- usar (obligatoriamente) la función sortearCopa definida en el archivo sorteo.pas, con exactamente el mismo nombre, para determinar la copa donde se va a esconder la moneda en cada ronda. Es decir, en la primera ronda se debe esconder en sortearCopa(1), en la segunda en sortearCopa(2) y así sucesivamente, si el usuario no se quedó sin saldo positivo antes.

Por ejemplo, suponiendo que **copa** y **nro** son variables enteras que indican la copa donde se debería esconder la moneda y número de ronda (o apuesta), la función se usa así:

```
copa := sortearCopa(nro);
```

EJEMPLOS

Juego de la tapadita con trampa

```
0 0 0  
1 2 3  
Copa?: 3  
Apuesta?: 1000  
0 0 .  
Ha acertado  
Saldo: 2000
```

```
0 0 0  
1 2 3  
Copa?: 1
```

Apuesta?: 2000
. 0 0
Ha acertado
Saldo: 4000

0 0 0
1 2 3
Copa?: 2
Apuesta?: 4000
0 0
No ha acertado
Saldo: 0

Gracias por jugar

CONSIDERACIONES

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas prácticas de programación que permitan lograr un código legible, bien documentado y mantenible, para esto es importante tener en cuenta:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

APE Nro 1 – GRUPOS IMPARES -TECLADO TELEFÓNICO

Lectura e interpretación de texto simulando el Teclado Telefónico

Este documento presenta el problema que deberá resolverse para la primer APE de las actividades de laboratorio 2013. Recuerde que durante todo el proceso de resolución del problema cuenta con la asistencia de un tutor.

PROBLEMA

Se pide diseñar e implementar un programa que permita la lectura de texto escrito usando solo los caracteres que se corresponden con un teclado telefónico. Este teclado está típicamente formado por los dígitos del 0 al 9 y las teclas * y # . La dificultad principal reside en que la cantidad de teclas es significativamente menor que las necesarias para representar las letras y los caracteres de puntuación; por lo tanto se debe ofrecer un método para que el usuario pueda ingresar texto completo a partir de las teclas disponibles.

En esta APE vamos a trabajar con la configuración estándar que asocia un conjunto de caracteres a cada letra como ilustra la figura 1.

Se implementará el método para el ingreso de texto conocido como “Multi-tap” donde cada símbolo se obtiene pulsando repetidas veces una tecla.

Este método funciona de la siguiente manera:

Cada tecla del 0 al 9 tiene asociado una secuencia de caracteres. En general se parte de una configuración estándar que es la que se muestra en la figura 2:



Figura 1 – Teclado Telefónico

| Tecla | Caracteres |
|-------|---------------------|
| 0 | 0 |
| 1 | . , ? 1 |
| 2 | a b c 2 |
| 3 | d e f 3 |
| 4 | g h i 4 |
| 5 | j k l 5 |
| 6 | m n o 6 |
| 7 | p q r s 7 |
| 8 | t u v 8 |
| 9 | w x y z 9 |
| # | espacio |
| * | minúscula/mayúscula |

Figura 2 – Teclas & secuencias de caracteres

CÓMO SE REALIZA EL INGRESO DE LETRAS, NÚMEROS, ESPACIOS Y SÍMBOLOS

El método multi-tap puede resumirse en la siguiente regla:

Regla Multi-tap: Si c_1, c_2, \dots, c_n es la secuencia de caracteres asociada con la tecla D, entonces para escribir el carácter c_k hay que pulsar k veces la tecla D.

Las secuencias de caracteres se recorren circularmente. Esto significa que si la tecla D se pulsa $(n+1)$ veces se obtiene el carácter c_1 y así sucesivamente.

Por ejemplo, para escribir el texto:

hola10 se puede digitar lo siguiente: **44666555211110L**

Nota: El símbolo ' L ' representa el fin del texto

CÓMO DISTINGUIMOS LAS LETRAS DE UNA MISMA TECLA

El símbolo 'P' (pausa) representa una separación entre dos repeticiones de la misma tecla con el fin de que no se interprete como una única repetición.

Por ejemplo, para escribir el texto:

casa se puede digitar lo siguiente: **222P277772L**

Para las teclas # y 0 no es necesario introducir P entre dos apariciones consecutivas de estas teclas, ya que estas teclas tienen 1 símbolo asociado cada una. Por ejemplo para escribir el siguiente texto (un 0, seguidos de dos espacios, seguidos de tres 0):

0 000 se puede digitar lo siguiente: **0##000L**

CAMBIAR ENTRE MAYÚSCULA Y MINÚSCULA

El programa comienza por defecto a escribir en minúscula. Para cambiar el modo (mayúscula/minúscula) se utiliza el símbolo * (asterisco), luego de ingresar un * el texto siguiente cambiará a mayúscula o minúscula según el modo anterior. En el texto pueden existir símbolos consecutivos de asterisco. Por ejemplo, para escribir el texto:

hOIA se puede digitar la línea de abajo

44*666*555*2L

ENTRADA Y SALIDA DEL PROGRAMA

El programa tomará como entrada cada línea de texto y mostrará el texto asociado debajo de cada línea leída. Todas las líneas terminan con el carácter : **L**, excepto la última que solo contiene el símbolo: **S** indicando el fin del programa (o Salir del programa).

Luego de cada fin de línea L hay un retorno de carro, es decir se espera un enter.

Ejemplos de prueba para el programa. Se muestra en negrita el texto ingresado por el usuario

y en normal la salida del programa. Recuerde que el texto a ingresar por el usuario podría tener **1 o más líneas**. En este ejemplo el texto 4 líneas

446665552L

hola

***222P277772L**

CASA

6*33#555P55526P666#*7*33P37776661L

Me llamo Pedro.

***23###444###66677771L**

AD I OS.

S

Ejemplo 2

778833#82555111L

que tal?

S

FORMA DE ENTREGA

Las entregas se realizarán a través del aula virtual y por los siguientes recursos del aula. Se indican en negrita los recursos que deben utilizarse en forma obligatoria.

| ETAPA | CARACTERÍSTICAS | RECURSOS AULA VIRTUAL |
|-------------------|---|-----------------------------------|
| Análisis y Diseño | Los alumnos se reúnen en grupos en forma sincrónica o asincrónica | Foro Mensajería interna |

| | | |
|------------------------|--|--|
| | y analizan el problema. Se plantean diferentes modelos. Se diseña el algoritmo. | chat WIKI |
| Implementación | Los alumnos en forma colaborativa implementan el algoritmo diseñado usando Pascal. | Foro Mensajería interna VPL |
| Presentación y Defensa | Los alumnos elaboran una presentación tipo Power Point en la que se presenta la solución al problema. Luego en forma presencial realizan la presentación y defensa de su trabajo | Foro Mensajería interna Tarea |
| Evaluación | Los alumnos realizan una evaluación del proceso de aprendizaje a nivel personal, grupal y del tutor | Encuestas |

FECHA DE ENTREGA

Se debe entregar hasta la etapa de implementación hasta el día 06/04/2013 hasta las 23:30 hs.

CONSIDERACIONES

Se puede utilizar los siguientes conceptos visto en las clases teóricas y prácticas:

- Tipos de datos:
 - Integer
 - Boolean
 - Char
- Instrucciones:
 - Asignación
 - Entrada y Salida(read, write, WriteLn, ReadLn)
 - Secuencia
 - Selección (if, case)
 - Repetición (for, while, repeat)

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas prácticas de programación que permitan lograr un código legible, bien documentado y mantenible, para esto es importante tener en cuenta:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas, nombres mnemotécnicos para variables, constantes, etcétera.

Cuadrado Mágico Grupos Pares – (G.Nilles /E. Lovos)

Definición

Un cuadrado mágico es una cuadrícula de 3 x 3, o de 4 x 4, o de 5 x 5 o, en general, de n x n, en la que se acomodan ciertos números que cumplen con la siguiente regla: la suma de cualquier fila, la suma de cualquier columna y la suma de cualquiera de cualquiera de las dos diagonales es siempre la misma.

| | | | |
|------|------|------|------|
| 4 | 9 | 2 | → 15 |
| 3 | 5 | 7 | → 15 |
| 8 | 1 | 6 | → 15 |
| ↓ 15 | ↓ 15 | ↓ 15 | |

Si la *condición de mágico* no se cumple para las diagonales, entonces se dice que el cuadrado es latino (Figura b). El origen de los cuadrados mágicos es muy antiguo. Los chinos y los indios los conocían antes del comienzo de la era cristiana.

Propiedades

El orden de un cuadrado mágico es el número de renglones o el número de columnas que tiene. Así un cuadrado de 3 x 3 se dice que es de orden 3.

Al sumar los números de cualquier renglón, cualquier columna o cualquiera de las dos diagonales el resultado es el mismo, y a ese valor se lo denomina **constante mágica**. Existen diversos métodos para encontrar la constante mágica:

a) Si se conoce el cuadrado mágico basta sumar cualquier renglón o columna o diagonal.

b) Si el cuadrado no se conoce, una manera es sumar todos los números que se colocarán en el cuadrado y dividir el resultado entre el orden de éste. Por ejemplo: en un cuadrado mágico de orden 3 los números que se colocarán son: **1, 2, 3, 4, 5, 6, 7, 8, 9**

Así la constante mágica para un cuadrado de orden N= 3 será $(1+2+3+4+5+6+7+8+9)/3 = 15$

c) Otra manera de calcular la constante mágica de un cuadrado mágico es acomodar en la cuadrícula los números que se van a utilizar en su orden natural (no en forma de cuadrado mágico) y sumar los números de cualquiera de las diagonales; el resultado será la constante mágica de ese cuadrado

Problema

Se pide que implemente un programa modularizado que trabaje con cuadrados de N * N, donde N para la implementación como máximo será 5, y disponga de las siguientes funcionalidades:

a) Determinar dado un cuadrado de N*N (cuyos valores ingresa el usuario), si el mismo es un cuadrado mágico o si es un cuadrado latino. En el caso que el cuadrado sea mágico, deberá indicar también la constante mágica.

Por ejemplo para el cuadrado de la figura a) El programa debería indicar “Cuadrado Mágico . Constante mágica : 15

En cambio para el cuadrado de la figura b) El programa debería indicar “Cuadrado Latino”

| | | |
|---|---|---|
| 8 | 3 | 4 |
| 1 | 5 | 9 |
| 6 | 7 | 2 |

Figura a- Cuadrado Mágico

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 1 | 4 | 3 |
| 3 | 4 | 1 | 2 |
| 4 | 3 | 2 | 1 |

Figura b- Cuadrado Latino

b) A partir de un número N que ingresa el usuario y tomando como base el cuadrado de origen chino “Lo-Shu” el cual se muestra en la figura de abajo, devuelva un cuadrado mágico de orden 3.

| | | |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

Para obtener otro cuadrado mágico a partir del cuadrado de “Lo-Shu”, se procede así, al número N que ingresa el usuario se lo suma, resta o multiplica con cada uno de los números del cuadrado original, acomodando los resultados en los mismos lugares.

El cuadrado que queda también es mágico.

a

| | | |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

cuadrado "lo-shu"

b

| | | |
|---|---|---|
| 4 | 9 | 2 |
| 3 | 5 | 7 |
| 8 | 1 | 6 |

cuadrado "lo-shu"

| | | |
|----|----|----|
| 12 | 27 | 6 |
| 9 | 15 | 21 |
| 24 | 3 | 18 |

Se multiplica cada número del original por 3

| | | |
|----|----|----|
| -1 | 4 | -3 |
| -2 | 0 | 2 |
| 3 | -4 | 1 |

A cada número del cuadro original se le resta 5

Por ejemplo:

Forma de Entrega

Las entregas se realizarán a través del aula virtual y utilizando los recursos del aula virtual que se indican en la siguiente tabla. En negrita los recursos que deben utilizarse en forma obligatoria. Recuerden que la actividad debe desarrollarse en forma colaborativa entre los miembros del grupo y con la asistencia del tutor asignado al grupo. Recomendamos antes de comenzar con la primer etapa, que cada integrante del grupo haga una lectura crítica de la consigna y luego reflexionen sobre la misma con los demás integrantes y el tutor.

| Etapa | Fecha Entrega | Recursos |
|-------------------|----------------------------------|---|
| Debate inicial | 30/04/13 a partir de las 14:30hs | Clase Presencial - Aula Magna |
| Analisis y Diseño | 08/05/13 hasta las 23:30 hs | Wiki – Foro – mensajería - Chat |
| Implementación | 13/05/13 hasta las 23:30 hs | VPL -Wiki – Foro – mensajería - Chat |

| | | |
|--|-------------------------|-------------------------------|
| Presentación y Defensa | 17/05/13 14:30 | Wiki + Aula Presencial |
| Evaluación del equipo de trabajo, tutores y personal | A partir del 14/05/2013 | Encuesta |

Consideraciones

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas prácticas de programación que permitan lograr un código legible, bien documentado y mantenible, para esto es importante tener en cuenta:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

Más Información en:

http://www.albertocoto.com/index.php?option=com_content&view=article&id=251&Itemid=43

Tres en Línea – Grupos Pares – (H.Segura /E. Lovos)

El Juego

El **tres en línea**, también conocido como **tres en raya**, **tatetí** y otros tantos nombres más, es un juego de lápiz y papel, en el que se dibuja un [tablero](#) de 3×3 y pueden jugar hasta 2 jugadores. Los jugadores se identifican en el tablero por los símbolos O y X, y van en cada jugada ocupando los espacios vacíos del tablero. Un jugador gana si consigue tener una línea de tres de sus símbolos: la línea puede ser horizontal, vertical o diagonal.

Existe como en el fútbol la posibilidad del empate, y es precisamente, la habilidad de romper dicho empate la que determina al verdadero héroe del juego.

Problema

Se pide que cada grupo modele e implemente el juego Tres en Línea, donde se pueda jugar contra la máquina o entre dos jugadores.

El juego comienza con la selección del modo de juego, para esto el programa dispondrá del siguiente menú

- 1- Contra PC
- 2- Dos Jugadores
- S – Salir

Si se selecciona la opción S, entonces se finalizará el programa.

Si se selecciona la opción 1, Se pide el nombre del jugador y se le asigna para la partida el carácter **X** y a la PC el carácter **O**. Se debe preguntar al jugador si desea jugar en primer lugar, caso contrario inicia la PC

Si se selecciona la opción 2, se debe preguntar nombre Jugador 1 y asignarle **X** y preguntar si desea jugar en primer lugar. Luego se preguntará por el nombre del jugador 2 y asignarle **O**.

En cada jugada, se mostrará el tablero en su estado actual, y jugará el jugador correspondiente, indicando la fila y columna donde quiere ubicar su ficha. A continuación, se evaluará si es un movimiento válido o no:

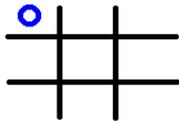
- Si es válido, se hará efectivo el movimiento y se evaluará si es una jugada ganadora o si hubo empate. En cualquier caso se debe mostrar el tablero y si la jugada fue ganadora el nombre del ganador. Por último se debe preguntar sobre la posibilidad de seguir jugando, en cuyo caso se limpiara el tablero y se comenzará con la nueva partida.
- Si no es válido, el programa deberá indicarlo, mostrando el tablero y solicitando nuevamente los datos de fila y columna.

Cuando se juega contra la PC, los movimientos de la misma se determinarán ubicando la ficha **O** en el primer espacio vacío. Esto es así para simplificar el juego, en su versión más compleja la idea consiste en determinar la mejor jugada.

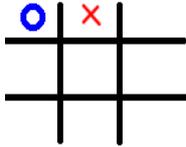
El programa deberá llevar la cuenta de los turnos, y si en un turno intenta jugar el jugador que no corresponde debe indicarlo. El jugador 1 siempre se identificará con **X** y el jugador 2 será identificado con **O**.

El jugador realiza el movimiento indicando fila y columna .

Ejemplo **Fila 1 Columna 1**



Entonces el programa debe controlar que el siguiente movimiento sea para X, caso contrario indicará el error y volvera a pedir que se ingrese el movimiento.



Se pide que el programa este modularizado y que incluya los siguientes módulos

- ³⁵₁₇ **Inicializar** : Dado un tablero de 3 * 3, lo inicializa, de manera que todas las casillas queden libres.
- ³⁵₁₇ **MostrarTablero** : Dado el tablero del juego, lo muestra en pantalla.
- ³⁵₁₇ **ObtenerMovJugador** : Dado el tablero del juego y un jugador, **obtiene** las coordenadas de fila y columna del movimiento. E **indica** si el movimiento es válido.
- ³⁵₁₇ **ObtenerMovPC**: Dado un tablero, obtiene las coordenadas de fila y columna libres y válidas para ubicar la ficha **O**
- ³⁵₁₇ **JugadaGanadora**: dado un tablero, determina si hay ganador o empate

Forma de Entrega

Las entregas se realizarán a través del aula virtual y utilizando los recursos del aula virtual que se indican en la siguiente tabla. En negrita los recursos que deben utilizarse en forma obligatoria. Recuerden que la actividad debe desarrollarse en forma colaborativa entre los miembros del grupo y con la asistencia del tutor asignado al grupo. Recomendamos antes de comenzar con la primer etapa, que cada integrante del grupo haga una lectura crítica de la consigna y luego reflexionen sobre la misma con los demás integrantes y el tutor.

| Etapa | Fecha Entrega | Recursos |
|--|----------------------------------|---|
| Debate inicial | 30/04/13 a partir de las 14:30hs | Clase Presencial - Aula Magna |
| Análisis y Diseño | 08/05/13 hasta las 23:30 hs | Wiki – Foro – mensajería - Chat |
| Implementación | 13/05/13 hasta las 23:30 hs | VPL -Wiki – Foro – mensajería - Chat |
| Presentación y Defensa | 17/05/13 14:30 | Wiki + Aula Presencial |
| Evaluación del equipo de trabajo, tutores y personal | A partir del 14/05/2013 | Encuesta |

Consideraciones

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas prácticas de programación que permitan lograr un código legible, bien documentado y mantenible,

para esto es importante tener en cuenta:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

Más información

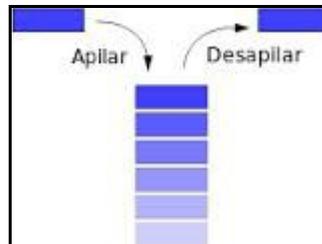
<http://bloglascosaspasan.blogspot.com.ar/2010/10/tres-en-rama-una-breve-historia.html>

APE NRO 3 – GRUPOS IMPARES - PILAS

PILA Usando Listas Grupos Impares – (E. Lovos)

Definición

Una pila representa una **estructura de datos lineal** en la que es posible agregar o quitar elementos únicamente por uno de sus dos extremos. En consecuencia, los elementos de una pila se **sacan en el orden inverso al que se insertaron**. Esta característica hace que se las conozca como estructuras **LIFO** (last input, first output). Es decir último en entrar, primero en salir. En todo momento, sólo se puede acceder directamente al último elemento agregado (el que está más arriba en la pila, que es llamado TOPE de la pila).



En la práctica podemos encontrar muchos usos para esta estructura: pila de platos, la pila de ejecución de un programa; etc

Las pilas con estructuras lineales como los arreglos, ya que sus componentes ocupan lugares sucesivos en la estructura de datos y cada elemento tienen un único sucesor o predecesor, con excepción del primero y el último.

Las pilas no son estructuras fundamentales de datos; es decir no están definidas como tales en los lenguajes de programación. Para su representación se hace necesario el uso de otras estructuras de datos como los arreglos y las listas enlazadas.

Problema

a)-Teniendo en cuenta la definición de pila dada, se pide que implemente una pila de caracteres, utilizando una lista enlazada como estructura de datos. La pila debe proveer la siguiente funcionalidad y los módulos deben denominarse como se indica en negrita:

- ³⁵₁₇ Insertar un caracter en la pila (**poner**).
- ³⁵₁₇ Recuperar un caracter de la pila (**sacar**).
- ³⁵₁₇ Obtener el caracter del tope de la pila, sin extraerlo (**Tope**).
- ³⁵₁₇ Averiguar si hay algún caracter almacenado en la pila (**EstaVacía**).
- ³⁵₁₇ Devolver el número de caracteres almacenados en la pila (**CantElem**).

Para evaluar esta primer parte se usará el programa de prueba **PruebaAPE3Pila.pas** que se adjunta con la consigna.

b)Una vez que haya implementado la pila de caracteres, se pide que utilice la implementación de la pila para resolver un problema clásico de compiladores: el emparejamiento de llaves { }, corchetes [], paréntesis (), y comentarios estilo /* */ de un lenguaje de programación (típicamente Java, Pascal, C ó C++). El programa debe leer una secuencia de caracteres hasta que se ingresa el símbolo #, e indicar si los caracteres anteriores están bien o mal emparejados.

El programa deberá presentar un menú con las siguientes opciones

- 1- Ingresar secuencia
- 2 – Imprimir Secuencia
- 3 – Evaluar Emparejamiento
- 4 – Salir

Ejemplo si el usuario ingresa la siguiente secuencia

/* comentario */ los parentesis ((()())(#

el programa debería indicar “**secuencia mal emparejada**”

Para la secuencia

(2 + 3 * (5 * 6 + [5-6]) # debería indicar “**secuencia mal emparejada**”

c) Si en lugar de pedir la implementación de una pila usando listas vinculadas, se utilizará una estructura de datos arreglo, en cambiaría la implementación? En qué casos convendría usar una o la otra implementación?. Explique características de ambas estructuras de datos.

Forma de Entrega

Las entregas se realizarán a través del aula virtual y utilizando los recursos que se indican en la siguiente tabla. Se resaltan en negrita los recursos que deben utilizarse en forma obligatoria. Recuerden que la actividad debe desarrollarse en forma colaborativa entre los miembros del grupo y con la asistencia del tutor asignado al grupo. Recomendamos antes de comenzar con la primer etapa, que cada integrante del grupo haga una lectura crítica de la consigna y luego reflexionen sobre la misma con los demás integrantes y el tutor.

| Etapa | Fecha Entrega | Recursos |
|--|---|---|
| Debate Inicial | 30/05/13 1430 hs | Clase Presencial – Aula Magna |
| Analisis y Diseño | 05/06/13 hasta las 23:30 hs | Wiki – Foro – mensajería - Chat |
| Implementación | Del punto a) 10/06/13 hasta las 23:30 hs Para aprobar la cursada Del punto b) 14/06/2013 hasta las 23:30 hs Para Promoción de la cursada | VPL -Wiki – Foro – mensajería - Chat |
| Presentación y Defensa | 21/06/13 14:30 | Wiki + Aula Presencial |
| Evaluación del equipo de trabajo, tutores y personal | A partir del 13/06/2013 | Encuesta |

Consideraciones

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas prácticas de programación que permitan lograr un código legible, bien documentado y mantenible, para esto es importante tener en cuenta:

- indentación adecuada

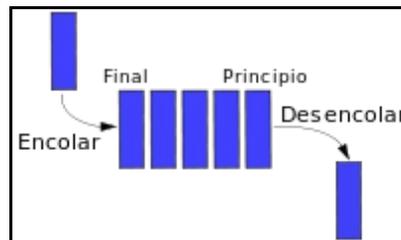
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

APE NRO 3 – GRUPOS PARES -COLAS

COLA Usando Listas Grupos Pares – (H.Segura)

Definición

Una cola es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la operación de inserción (*push*) se realiza por un extremo y la operación de extracción (*pop*) por el otro. También se le llama estructura **FIFO** (del inglés First In First Out), debido a que **el primer elemento en entrar será también el primero en salir**.



Las colas se utilizan en sistemas informáticos, transportes y operaciones de investigación (entre otros), donde los objetos, personas o eventos son tomados como datos que se almacenan y se guardan mediante colas para su posterior procesamiento.

Las colas son estructuras lineales como los arreglos, ya que sus componentes ocupan lugares sucesivos en la estructura y cada elemento tienen un único sucesor o predecesor, con excepción del primero y el último.

Las colas no son estructuras fundamentales de datos; es decir *no están definidas* como tales en los lenguajes de programación. Para su representación se hace necesario el uso de otras estructuras de datos como los arreglos y las listas enlazadas.

Problema

Se pide que cada grupo modele e implemente las siguientes funcionalidades:

a)-Teniendo en cuenta la definición de cola, se pide que implemente una cola que permita almacenar números enteros, utilizando una lista enlazada como estructura de datos. La cola debe proveer la siguiente funcionalidad y los módulos deben denominarse como se indica en negrita:

³⁵₁₇ Agregar un entero en la cola (**encolar**).

³⁵₁₇ Eliminar un entero de la cola (**desencolar**).

³⁵₁₇ Obtener el entero que está en la primer posición de la cola, sin extraerlo (**primero**).

³⁵₁₇ Averiguar si hay algún entero almacenado en la cola (**EstaVacía**).

³⁵₁₇ Devolver la cantidad de enteros almacenados en la cola (**CantElem**).

Para evaluar esta primer parte se usará el programa de prueba **PruebaAPE3Cola.pas** que se adjunta con la consigna.

b)-Una vez que haya implementado la estructura de datos cola de enteros y sus operaciones, diseñe e implemente un módulo que recibe una cola de enteros C y un número entero N, y elimine todas las ocurrencias de N en la cola. El módulo debe denominarse **ElimElem**.

El programa deberá presentar un menú con las siguientes opciones:

1- Ingresar datos cola. A través de esta funcionalidad el usuario ingresará números para encolar

con un máximo de 10 números.

2 – Eliminar Elemento. A través de esta funcionalidad se le pedirá al usuario que ingrese un número N y si procederá a eliminarlo de la cola si es que la misma tiene datos.

3 – Mostrar cola. Muestra la cola de números.

4 – Salir. Finaliza el programa

Ejemplo:

Si se ingresa la secuencia 5 10 1 4 6 1 debería quedar una cola como la siguiente

5 10 1 4 6 1

y luego se selecciona la opción 2 del menú ingresando el valor N = 1, la cola debería quedar como muestra la imagen.

5 10 4 6

c)- Si en lugar de pedir la implementación de una cola usando listas vinculadas, se utilizará una estructura de datos arreglo, en cambiaría la implementación?. En qué casos le convendría usar una u otra implementación?. Explique características de ambas estructuras de datos.

Forma de Entrega

Las entregas se realizarán a través del aula virtual y utilizando los recursos que se indican en la siguiente tabla. Se resaltan en negrita los recursos que deben utilizarse en forma obligatoria. Recuerden que la actividad debe desarrollarse en forma colaborativa entre los miembros del grupo y con la asistencia del tutor asignado al grupo. Recomendamos antes de comenzar con la primer etapa, que cada integrante del grupo haga una lectura crítica de la consigna y luego reflexionen sobre la misma con los demás integrantes y el tutor.

| Etapa | Fecha Entrega | Recursos |
|--|---|---|
| Debate Inicial | 30/05/13 1430 hs | Clase Presencial – Aula Magna |
| Análisis y Diseño | 05/06/13 hasta las 23:30 hs | Wiki – Foro – mensajería - Chat |
| Implementación | Del punto a) 10/06/13 hasta las 23:30 hs Para aprobar la cursada Del punto b) 14/06/2013 hasta las 23:30 hs Para Promoción de la cursada | VPL -Wiki – Foro – mensajería - Chat |
| Presentación y Defensa | 21/06/13 14:30 | Wiki + Aula Presencial |
| Evaluación del equipo de trabajo, tutores y personal | A partir del 13/06/2013 | Encuesta |

Consideraciones

En la resolución de la APE se valorará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios vistos en clase. Haciendo énfasis en las buenas

prácticas de programación que permitan lograr un código legible, bien documentado y mantenible, para esto es importante tener en cuenta:

- indentación adecuada
- utilización correcta y apropiada de las estructuras de control
- código claro y legible
- algoritmos razonablemente eficientes
- utilización de comentarios que documenten y complementen el código
- utilización de constantes simbólicas
- nombres mnemotécnicos para variables, constantes, etcétera.

ANEXO II - ENCUESTAS

ENCUESTAS APE 1 Y 2

Evaluación de la Actividad

A través de este formulario, se busca que Ud. auto-evalúe su participación en la APE, la de equipo de trabajo y la del tutor asignado. Esta evaluación es anónima, le pedimos que responda con total sinceridad a la misma.

*Obligatorio

1- Indique Nro de Grupo *

Autoevaluación

2 - Participé de las e-actividades (foros, wiki, VPL) propuestas *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

3- Asistí a los encuentros (presenciales o virtuales) establecidos por el grupo *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

4- Durante la actividad tuve dominio (conocimiento, manejo) de la información que se discutió para el desarrollo de la APE *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

5- Aporte información nueva y relevante en las discusiones que realizó el grupo *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

6- Realice preguntas que ayudaron a la comprensión. *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

7- Respete las opiniones de los demás. *

- Siempre
- Casi Siempre
- Alguna Vez
- Nunca

8- Cómo calificaría su interés en la materia, luego de realizar la APE *

- Nulo
- Poco
- Normal
- Grande
- Muy Grande

9- El desarrollo de la APE le permitió una mejor comprensión de los conceptos de las unidades involucradas *

- Nada
- Poco
- Normal
- Mucho

10- Hubiese podido realizar esta misma actividad en forma individual *

- Si
- Tal Vez
- No

Evaluación del Equipo de Trabajo

11- Respecto a la participación en las e-actividades (foros, wiki, VPL) propuestas *

- Participaron Todos
- Participaron Casi Todos
- Participaron Algunos
- No participo ninguno

12- Respecto a la asistencia a los encuentros (presenciales o virtuales) establecidos por el grupo *

- Participaron Todos
- Participaron Casi Todos
- Participaron Algunos
- No participo ninguno

13- Respecto al dominio de la información que se discutió para el desarrollo de la APE *

- Todos tenían conocimiento
- Casi todos tenían conocimiento
- Solo algunos tenían conocimiento
- Ninguno tenía conocimiento

14- Respecto al aporte de información nueva y relevante en las discusiones *

- Todos aportaron
- Casi todos aportaron
- Solo algunos aportaron
- Ninguno aportó

15-Se respetaron las opiniones de los demás *

- Todos los integrantes
- Casi todos los integrantes
- Algunos
- Ninguno

16-Le gustaría volver a trabajar con este grupo *

- Si, con todos sus integrantes
- Si, con algunos integrantes
- No, con ningún integrante.

17-Podría indicar lo mejor y lo peor de trabajar con este grupo

Cuadro de texto

Evaluación del Tutor

18-El tutor/a ha sido dinámico y activo durante la actividad, fomentando la colaboración (foros, wiki, VPL) *

- Siempre
- Casi Siempre
- Alguna vez
- Nunca

19-El tutor nos ha animado a preguntar y nos han dado respuestas satisfactorias *

- Siempre
- Casi Siempre
- Alguna vez
- Nunca

20-El tutor tenía dominio de la información que se discutió para el desarrollo de la APE *

- Siempre
- Casi Siempre
- Alguna vez
- Nunca

21-Aportó información nueva y relevante en las discusiones que realizó el grupo *

- Siempre
- Casi Siempre
- Alguna vez
- Nunca

22-Respecto del uso del Laboratorio Virtual (VPL), el tutor les brindó asistencia *

- Si
- No
- No fue necesario.

ENCUESTA APE 3

Se indican solo las preguntas agregadas a cada tipo de evaluación.

Auto-evaluación

1-Solo sino participo del desarrollo de la APE indique porque.

Cuadro de texto

2- El desarrollo de la APE le permitió prepararse mejor para el examen parcial *

- Si
- No
- No Sabe

3-Le gustaría que esta forma de trabajo se aplicará a otras materias *

- Si
- No
- No Sabe

4- Si tuviera que hacer cambios en la materia. Cuáles serían? Porque?

Cuadro de texto

ANEXO III - OBSERVACIÓN DE CLASE

La siguiente observación de clase, corresponde a la fase de presentación y defensa de la APE Nro1. Esta observación fue realizada por una docente de la carrera del área de matemáticas, que no tenía ninguna relación académica con los alumnos que participaron de la actividad.

Nota:

Los alumnos en cada grupo expositor son numerados 1°, 2°, etc, de acuerdo al orden en que tomaron la palabra durante la exposición. Sólo se hace referencia por nombre en caso que explícitamente ellos se hayan referido por sus nombres.

| Hora | Observación | Impresión |
|-------------|--|---|
| 16:32:00 | Un alumno consulta por el aula, estaban esperando afuera. Cuando se le confirma el aula, entra y llama a los otros | |
| 16:35:00 | La profesora presenta la actividad, consigna: cada grupo expone la solución al problema Los alumnos no hacen comentarios, ni sacan ningún material. Algunos comienzan a charlar entre sí mientras esperan que la docente acomode el equipo. | Los alumnos escuchan en silencio, le hacen algún comentario sobre el cañon (fuera de foco) |
| 16:37:00 | La profesora. muestra plataforma en cañon para ir viendo cada grupo La profesora propone comenzar con actividad 1, y sugiere a grupo 1 que inicie, abre algoritmo y propone si quieren mostrar presentación. Pasan los 3 alumnos involucrados, entregan presentación en pendrive a profesora. y esperan que la abra. La docente pide que lo hagan en forma breve dado los tiempos. | Se observan que se muestran cohibidos...hablan entre sí mientras esperan. Parece que se les hace larga la espera hasta que se abre presentación. |
| 16:41:00 | Entran unos 3 alumnos (no saludan), ya hay aprox 20 alumnos, más los 3 al frente del grupo 1 | |
| 16:43:00 | Entra una alumna, saluda y la docente responde La profesora presenta nuevamente la actividad haciendo un revisión general sobre lo realizado por todos los grupos. Ante la dificultad en abrir, la profesora menciona en general que convendría realizarlo en pdf | Los alumnos se ríen nerviosamente por los “problemas” en abrir la presentación |
| 16:45:00 | Llega un alumno que se queda esperando afuera (la puerta está abierta), la profesora. lo invita a entrar, el alumno pasa y saluda. Enseguida entra otro alumno detrás (este sin saludar). Continúan los problemas para abrir archivo, los alumnos proponen pasar a pdf, por lo que la docente pido a grupo 3 si puede pasar, a lo que le responden que falta un integrante. Luego llama a grupo 5, quienes pasan y entregan pendrive a profesora. Esperan mientras se abre archivo. | El grupo 5 parece más confiado, son una alumna que parece ser la líder y dos varones |
| 16:49:00 | Algunos grupos trabajan con sus respectivas pc, una de ellas emite mensaje de vos “la base...ha sido...” lo que provoca risas. | |
| | Inicia gupo 5. Un alumno explica el algoritmo, la profesora. pide explique como encararon el problema, responde el otro alumno, explicando detalles técnicos (contador y funcionamiento) la profesora retoma y explica en general. La profesora consulta que modelaron, responde le primer alumno que intervino, explicando los pasos. profesora. pregunta por asignación en particular, responde el mismo alumno, y complementa el otro, con afirmación de la alumna, quien intenta intervenir. Vuelve a explicar primer alumno. profesora. consulta nuevamente, responde 2° alumno y completa 1°. profesora vuelve a consultar, continua el dialogo en el que intervienen 1° y 2°. | |
| 16:54:00 | Entra un alumno y cierra puerta. La profesora. consulta en general al grupo y escribe en pizarrón, un alumno del grupo gral responde, y luego completa uno de los del grupo 5 (2°). La profesora retoma diciendo que eso era lo que se hablaba en el foro. Explica interpretando el código de los alumnos, y espera que ellos completen, los alumnos intentan seguir el razonamiento. La profesora. aprovecha para explicar al grupo general. La profesora pregunta si quieren agregar algo más. Explica 1°, la profesora consulta, responde 2°. Los alumnos grupo gral escuchan. Algunos trabajan un su pc. La profesora vuelve a dirigirse al grupo gral., explica sobre lo que se está exponiendo. Pregunta al grupo...se hace un silencio, la docente “se responde”. Uno | Los alumnos tiene dificultad en seguir la rapidez del razonamiento de la docente, intentan seguirlo y aportar. |

| | | |
|----------|---|--|
| | de los integrantes grupo 5 aporta (1°). | |
| 17:00:00 | La profesora consulta a grupo gral, sólo uno (del fondo) responde, ante próxima pregunta intervienen 2 o tres con respuestas cortas. Luego un alumno (del frente) explica un poco más. La profesora. repregunta, no hay respuesta, vuelve a formular preguntar, sólo alumno 1° grupo 5 responde. Luego interviene un alumno sentado en primera fila, explicando con mas detalle. | |
| 17:03:00 | Suena un celular, el alumno sale a atender. Entra el ayudante y se queda parado cerca de la puerta. La profesora cierra presentación del grupo, y les consulta sobre como usaron plataforma, uso del foro y wiki, dicen que usaron al plataforma, wiki, se organizaron en clase y luego se “fueron pasando archivos”. | |
| 17:05:00 | Se sienta el primer grupo, el ayudante se acerca a profesora. pasan alumnos de grupo 1, entregan pendrive a profesora, el ayudante les hace comentario. La profesora. pide al resto comentarios respecto de la presentación que se muestra, responde algunos. La profesora. valora la presentación (carátula). Inicia 1° diciendo que se juntaron en la casa, luego cambiaron,,,explica el proceso que hicieron para el trabajo. Luego empiezan explicando el código. La profesora. interviene explicando la importancia de interfase para explicar código. El alumno 1° continua con la explicación del código, con preguntas de la docente, siempre dirigiéndose al grupo gral, aunque responde 1°, y 2° completa afirmando. | Parece que la presencia de el ayudante les da algo de tranquilidad, aunque se muestran nerviosos. |
| 17:11:00 | La profesora. consulta a grupo gral, no hay respuesta...al rato responde 1 sentado al frente, repregunta, intervienen algunos más. Pide comparación con otro código. el ayudante pregunta, y responde 1°. La docente dice que el código no está tan comentado, lo explica, y platea importancia de identificar variables (banderas). Pide si hay preguntas al grupo, nadie interviene. La profesora pregunta como se organizaban y dice que sólo 1 participa en foros, explican que se juntaban físicamente. Se da por finalizada explicación, los alumnos del grupo se retiran del aula. El ayudante les hace el comentario de si se quieren quedar... | |
| 17:17:00 | La profesora. llama varios grupos pares (del ayudante). Pasa grupo 6 (dos varones, 2 chicas). el ayudante que se había quedado parado aprovecha a acercarse a escritorio y se sienta. La docente explica problema mientras se abre presentación del grupo. Un alumno sale y vuelve a entrar (otro hace lo mismo). La profesora pide expliquen como pensaron problema. Empieza 1° alumno y el otro acota, las chicas se quedan como detrás de los varones que son los que hablan. La profesora. explica problema. Una de las alumnas acota en algunas intervenciones del compañero. Explica alumno 2° detallando el algoritmo. La profesora. lo explica al grupo general, sugiere a todos que en próximas presentaciones estén explicados los códigos. Sigue alumno 1° que plantea algunas acotaciones que les hizo el ayudante, el ayudante acota, el alumno explica. | Los alumnos escuchan con atención, grupo 3 que ya paso se muestran aburridos. |
| 17:28:00 | Sigue explicación grupo 6, hablando alumno 1° explicando código. La profesora pregunta al grupo. La docente consulta a el ayudante como evalúa para entender el código, el ayudante le explica, el alumno acota. Alumno 1° da pie a que intervenga una de las alumnas, que se adelanta y explica muy acotada, a lo que 1° retoma. Sigue alumna 3° explicando, toma la palabra la otra alumna habla muy rápido explicando. La profesora. pregunta, acota el ayudante, y los alumnos explican. La profesora. pregunta, responde ultima alumna. Algunos alumnos del grupo gral hablan por lo bajo. La profesora. pregunta por intervenciones en el foro, dos de ellos (1 chica y 1 chico) y explican que se juntaron 3 o 4 veces personalmente, luego por chat preguntas puntuales entre ellos que les permitieron ir mejorando, cambiando algoritmo. Finaliza exposición. Alumnos se sientan | La profesora parece no estar del todo de acuerdo con las opciones del código, que entiendo fueron sugeridas por el ayudante. La ultima alumna se nota muy nerviosa, aunque parece que entiende el problema. |

| | | |
|----------|--|---|
| 17:33:00 | <p>profesora llama a grupo 8 que no entregó. Pide de pasar grupo 3 que faltaba integrante, pero deciden pasar si él (son dos alumnos varones, dicen que falta “Nicolás”)</p> <p>El grupo gral se distiende, hablan, un grupo prepara mate. La profesora habla a todos, se hace silencio. La profesora pregunta como se organizaron, dicen que usaron plataforma sólo para presentar trabajo, sino que prefieren encontrarse personalmente. La profesora cuestiona la poca participación, pide que quede registro de interacción en plataforma.</p> <p>Inicia presentación, comenta alumno diciendo dificultades para mostrar caracteres, dudas para iniciar el algoritmo. La profesora decide buscar versiones previas para mostrar esto al grupo gral. Lo busca, hay silencio del grupo gral.</p> | |
| 17:41:00 | <p>La profesora explica sobre uno de los “errores” iniciales de los alumnos. La profesora pregunta al grupo general, alumnos intervienen, entendiendo la situación. Una alumna del grupo que expuso antes (ultima) se retira. Uno de alumnos grupo gral sigue el razonamiento, explica lo que haría el código, otro también interviene.</p> | Los alumnos del grupo (1 de ellos) se sienten avergonzados por comentarios profesora. |
| 17:46:00 | <p>Los alumnos explican su código, frente a consultas docente, mayor intervención de uno de ellos (1°), el otro acota. Vuelven a presentación ultima, arreglada. Alumno 2° explica dificultades para llegar esa versión final. La docente da por finalizada la exposición.</p> | |
| 17:49:00 | <p>La profesora llama a grupo 10 (de el ayudante) quien “los llama” a pasar. Se retira primer grupo que expuso, la profesora pide que suban exposición a plataforma. Grupo 10 son 3 alumnos (todos muy jóvenes)</p> | Los alumnos parecen no tener muchas intenciones de pasar. |
| 17:52:00 | <p>Se muestra presentación muy gráfica...y se hacen comentarios graciosos al respecto.</p> <p>Luego muestran código. Uno de ellos 1° comienza explicando código. La profesora acota, pregunta, sigue explicando 1°. Acotan otros alumnos del grupo, y también el ayudante (tutor del grupo). Sigue 1°.</p> <p>La profesora explica el código, y cómo podría haber sido. Sigue 1°, profesora pregunta porque eligieron estructura repeat, habla 2° y luego 3° la profesora pregunta a grupo general, uno de ellos responde. profesora explica diferencia de como hubiesen funcionado distintas estructuras.</p> | Quedan unos 15 alumnos, escuchan atentamente. |
| 17:57:00 | <p>Sigue 1° explicando. Alumnos están en silencio. Uno de ellos escribe en el celular, otro grupo sigue con el mate. Interviene 2° en explicación, ante pregunta profesora. habla 3°. el ayudante acota a explicaciones. La profesora explica como podrían haber reducido el código. La profesora les pregunta como usaron plataforma, dicen que los 3 usaron foro, mas que nada para dudas con tutor, ya que se juntaban en la casa, dice que no supieron como usar la wiki.</p> | el ayudante interviene como “apoyando” al grupo |
| 18:02:00 | <p>Pasa grupo 6, profesora abre presentación. Se retira un alumno. Dicen que eran 4, uno dejó y el otro no participo.</p> <p>Pasan sólo 2. Comienza 1° diciendo como entendieron el problema, pero empezaron queriendo programar y se “trabaron”. La profesora pregunta si tenían ideas iniciales de programación dicen que algo con Scratch, dicen que uno de ellos ganó concurso nacional con Alice. La profesora aprovecha para comentar sobre este tipo de software. Alumno 1° explica algoritmo, se refiere a que utilizaron idea similar al grupo 5. La profesora pregunta como usaron “salida” del programa. el ayudante acota, profesora explica. Da por terminada presentación alumnos se sientan, la profesora retoma como realizar salida, no usar “exit”</p> | A alumno 1° parece que le cuesta expresarse para explicar lo que hicieron |
| 18:11:00 | <p>Pasa grupo 11. Se retiran 3 alumnos.</p> <p>Son 4 alumnos (3 varones, 1 chica). profesora abre presentación que tiene “animación”, por lo que se hacen comentarios en tono de broma. Comienzan 1° (alumno que parece mayor) explicando que se juntaron personalmente por lo que no usaron mucho plataforma. Usaron un diagrama de flujo que la profesora</p> | Este grupo era el que tomaba mate. |

| | | |
|----------|--|---|
| | <p>comenta que no lo vieron en el curso, pero explica su utilidad. Sigue 1° explicando código, interviene la chica 2°. La profesora explica para el grupo gral lo que hicieron. Sigue 1° usando diagrama de flujo. El alumno propone ver el código, la profesora. lo busca en la plataforma, aprovecha a consultarles porque no todos usaron la plataforma, la chica explica que porque se reunían personalmente o en clase. La profesora busca programa en laboratorio virtual y lo ejecuta, el ayudante acota si todos sabían como ejecutarlo, varios dicen que aun no. La profesora. explica brevemente como usar el VPL. Hace algunas pruebas. Pregunta por condiciones de salida, la verifica sin esperar respuesta. Da por finalizada la exposición, dice que mejoró de la presentación anterior.</p> | |
| 18:20:00 | <p>Los alumnos se sientan. Pasa grupo 12, son 3 alumnos varones que incluye al alumno ciego. Piden usar su propia pc (una tablet), la conecta al cañón con la ayuda de el ayudante. La profesora. dice que sólo lo vio a Javier en la plataforma, y a Gonzalo, los otros explican que les falto un poco mas de confianza, y que a veces se juntaron personalmente. Los alumnos están parados de espaldas al grupo, no les anda el cañón. La profesora. aprovecha para explicar la importancia de participar en plataforma. La profesora dice como evacuaban dudas,...no responden, luego dicen que tiene organizada exposición, están esperando intervenga German. Empieza alumno ciego, pero empieza explicando el código. Sigue el dueño de la tablet, dice que no quisieron usar la mensajería, porque no les resultaba, Dicen que Javier había empezado, éste interviene y explica como lo hizo. Sigue dueño tablet diciendo que usaron turbo pascal, y muestra código. Sigue Javier diciendo como pensó y usó la función. La profesora pregunta que hicieron los demás. Interviene alumnos grupo gral diciendo que no entendieron como hacerlo. Sigue Javier tratando de explicar y pide escribir en el pizarrón. Hace un “esquema”, explica.</p> | |
| 18:32:00 | <p>Se retira un alumno, la profesora consulta a grupo gral, responde uno del grupo que no participó hasta ahora. profesora repregunta al grupo gral, nadie responde, la profesora explica sobre los errores que observa en el código del grupo. Se retira otro alumno, quedan 9 mas 4 que exponen. La profesora. explica importancia de responder a la consigna. Continúa Javier, luego dueño de la tablet, dice que empezó Javier, luego se reunieron vieron lo que hizo y trataron de arreglarlo, Retoma último tratando de explicar el código, aunque se queda “trabado”, sigue el dueño tablet. La profesora consulta, el ayudante acota, explicando. La profesora, retoma la importancia de elegir nombres representativos de variable. El dueño tablet pide a último que explique pero este dice que no puede porque se “pierde” en la pantalla del turbo pascal. La profesora comienza a preguntar a grupo gral. Nadie responde. Sigue grupo, hablando Javier</p> | <p>Los alumnos están cansados, ya pocos siguen explicación.</p> |
| 18:40:00 | <p>Vuelve a intervenir el dueño tablet mostrando, el ayudante pide si puede mostrar ejecución. Mientras lo prueba la profesora pregunta si todos programaron en pascal, dueño tablet explica, dice que el que más sabía era Javier, y que German también aportó ideas, aunque no pudo programar. Cuando muestra ejecución dueño tablet pide a Marcelo explique. Este intenta explicar, profesora interviene, no espera respuesta. Intentan algunos casos.</p> | |
| 18:48:00 | <p>La profesora da por finalizada exposición. Todos comienzan a pararse, la profesora. pide que suban al foro, y da consignas para semana próxima, dice que por plataforma va a explicar como organizar grupos y actividad. Los alumnos escuchan de parados y con mochilas en mano, se acercan a la puerta. La profesora. dice que subió algo más para hacer individual, le pide que prueben la herramienta. Un alumno consulta dice que si tiene dudas sobre eso, la profesora. dice que usen el foro para eso. Los despide. Se retiran casi todos, quedan último grupo, Marcelo se acerca a el ayudante, le explica dificultades en la exposición. Otro (pelo largo) se queda consultando a la profesora sobre trabajar en maquinas de la universidad.</p> | |



ANEXO IV - PROGRAMA DE LA ASIGNATURA

| | |
|------------------------------------|-----------------------------------|
| Asignatura: Programación I | |
| Año calendario: 2012 | Cuatrimestre: primero |
| Carga horaria semanal: 6 hs | Créditos (si corresponde): |
| Carga horaria total: 96 hs | |

Días y horario de cursada: Martes de 14:30 a 17:30hs

Jueves de 18:00 a 21:00 hs

Horarios, días y lugar de consulta para alumnos: viernes de 09:00hs a 12:00hs

Horas de estudio recomendadas (extra clase): 6 hs. Semanales

Profesor: Ing. Edith Lovos

Email: elovos@unrn.edu.ar

Jefes de Trabajos Prácticos: Lic. Horacio Segura

Email: hsegura@unrn.edu.ar

Email:

Programa Analítico de la asignatura

Contenidos mínimos establecidos por Plan de Estudio:

1. Algoritmos: conceptos básicos y definiciones. Resolución de problemas por computadora. Tipos de datos simples. Modularización. Procedimientos. Parámetros. Estructuras de datos compuestas: registros. Estructuras de datos indexados: arreglos. Estructuras de datos lineales: listas, caso particular de acceso en forma de pilas y colas.

Objetivos de la asignatura:

- ³⁵/₁₇ Analizar problemas resolubles con computadora, poniendo énfasis en la modelización, la abstracción de funciones y en la descomposición funcional de los mismos. Obtener una expresión sintética y precisa de los problemas.
- ³⁵/₁₇ Estudio, expresión simbólica, implementación y evaluación de algoritmos, orientando los mismos a la resolución de las partes (módulos) en que se descomponen los problemas, a partir de un paradigma procedural/ imperativo.
- ³⁵/₁₇ Introducción de las nociones de estructuras de datos, tipos de datos y abstracción de datos.

Propuesta Metodológica:

Se dictan clases teóricas y prácticas semanales con evaluaciones teóricas y prácticas. Se realiza un acompañamiento a través de la plataforma Moodle donde los alumnos disponen de los materiales utilizados en la teoría y la práctica; así como recursos adicionales para profundizar y ejemplificar cada tema. A través del aula virtual se desarrollan y se entregan las actividades prácticas requeridas por la cátedra. Aunque la metodología de las clases teóricas es expositiva se promueve la participación de los alumnos en la resolución de las situaciones problemáticas que se presentan como ejemplos para mostrar la aplicación de los conceptos explicados. Las clases prácticas tienen como objetivo la aplicación de los conceptos vistos en la teoría y otros que serán descubiertos por ellos mismos, en la resolución de problemas computacionales, a través del diseño algoritmos y luego

implementando esas soluciones en un lenguaje de programación de alto nivel tipo Pascal. El programa consta de seis unidades didácticas, cada una con su correspondiente trabajo práctico y tres Actividades Prácticas Entregables (APE) integradoras. Estas consisten en la resolución colaborativa en equipos de trabajo, de problemas de mediana complejidad, cuya solución (programa computacional) tendrá que implementarse en el lenguaje de programación elegido por la cátedra. Las APE se desarrollarán combinando la metodología de ABP (Aprendizaje Basado en Problemas) y las TIC, promoviendo la participación de los alumnos y el desarrollo de competencias transversales tales como el razonamiento crítico, la capacidad de análisis, el trabajo en equipo, la autorregulación y la comunicación. El proceso de investigación estará dirigido por el docente.

Forma de aprobación: La aprobación de los trabajos prácticos se obtiene mediante un examen parcial práctico, con dos instancias recuperatorias, realizados durante el cuatrimestre que dura el curso; los mismos deben ser con nota: *APROBADOS*. Más la resolución de 3 Actividades Prácticas Entregables (APE) integradoras.

³⁵₁₇ Para obtener la aprobación final de la asignatura se debe primero aprobar el examen parcial en cualquiera de sus instancias, en las fechas que publique la cátedra, participar en el desarrollo de las 3 APE y luego rendir una evaluación final teórica, en las fechas de exámenes finales fijados por el calendario académico de la UNRN.

³⁵₁₇ Ver anexo 1 al final del documento, donde se detalla el reglamento interno de la asignatura.

Unidad o eje temático: 1 – Resolución de problemas. Algoritmos.

Contenidos:

Modelización de problemas del mundo real.

Elementos de un algoritmo. Operadores lógicos & expresiones lógicas. Operadores relacionales. Flujo de control de algoritmos (secuenciación, iteración, selección).

Fechas tentativas de Inicio y finalización del dictado de la unidad o eje temático:
06/03 al 16/03

Bibliografía básica:

Cómo Plantear y Resolver Problemas

Polya, G. Editorial Trillas. 1965

Estructuras de Datos y Algoritmos.

Hernández R., Dormido R., Lazaro J. Ros S. Pearson Education. 2000.

Introduction to algorithms

Comen, Leiserson. MIT Press 2001.

Bibliografía complementaria:

Estructuras de Datos. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Unidad o eje temático: 2 – Datos y Tipos de datos simples. Estructuras de control

Contenidos: Tipos de datos primitivos.

Constantes y variables.

Funciones predefinidas.

Tipos ordinales.



Tipos de datos definidos por el usuario.

Estructuras de control: if, while, repeat...until, for

Fechas tentativas de Inicio y finalización del dictado de la unidad o eje temático:

17/03 al 26/03

Bibliografía básica:

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Estructuras de Datos y Algoritmos. Weiss, M.A. Addison Wesley. 1995.

Bibliografía complementaria:

Pascal Estructurado.

Tremblay, Jean Paul. Mc Graw Hill.1980.

Unidad o eje temático: 3 - Modularización. Procedimientos y funciones. Parámetros

Contenidos: Descomposición de problemas. Utilidad e importancia.

Subprogramas o módulos.

Procedimientos.

Funciones.

Conceptos de argumentos y parámetros.

Conceptos de variables locales y variables globales.

Procedimientos y funciones con parámetros.

Noción de reusabilidad.

Fechas tentativas de Inicio y finalización del dictado de la unidad o eje temático:

27/03 al 19/04

Bibliografía básica:

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Apuntes de Cátedra.

Data structures, algorithms and software principles.

Standish, T. A. Addison Wesley Publishing Company. 1994.

Estructuras de Datos y Algoritmos

Weiss, M.A. Addison Wesley. 1995.

Fundamentos de Programación.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Bibliografía complementaria:

PASCAL Programming and Problem Solving.

Leestma Sanford. Macmillan Publishing Company. 1984.

Estructuras de Datos.

Lipschutz, S. Mc Graw Hill. 1997.

Unidad o eje temático: 4 - Estructuras de datos compuestas

Contenidos:



Introducción.

Registros.

Operaciones sobre cada una de las estructuras mencionadas.

Concepto de tipo definido por el usuario.

Fechas tentativas de Inicio y finalización del dictado de la unidad o eje temático:

20/04 al 10/05

Bibliografía básica:

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Apuntes de Cátedra.

Data structures, algorithms and software principles.

Standish, T. A. Addison Wesley Publishing Company. 1994.

Estructuras de Datos y Algoritmos

Weiss, M.A. Addison Wesley. 1995.

Fundamentos de Programación.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Bibliografía complementaria:

Estructura de Datos y Algoritmos.

Sisa, Alberto Jaime. Editorial Prentice. 2002.

Pascal Estructurado.

Tremblay, Jean Paul. Mc Graw Hill. 1980.

Unidad o eje temático: 5 - Datos compuestos indexados: arreglos

Contenidos: Clasificación de las estructuras de datos.

Arreglos. Operaciones con arreglos de una dimensión.

Matrices. Tratamiento de información estructurada en vectores y matrices.

Algoritmos de búsqueda.

Algoritmos de ordenación. Ordenación por índice.

Métodos de ordenación eficientes.

Fechas tentativas de Inicio y finalización del dictado de la unidad o eje temático:

11/05 al 01/06

Bibliografía básica:

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Apuntes de Cátedra.

Data structures, algorithms and software principles.

Standish, T. A. Addison Wesley Publishing Company. 1994.

Estructuras de Datos y Algoritmos

Weiss, M.A. Addison Wesley. 1995.

Fundamentos de Programación.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.

Bibliografía complementaria:

Unidad o eje temático: 6 - Estructura de datos compuestas enlazadas: listas

Contenidos:

Alocación dinámica. Punteros.
Listas. Operaciones con listas.
Relaciones entre los accesos a listas, vectores.
Tipos de listas: dobles, circulares.
Merge de listas

Fechas tentativas de inicio y finalización del dictado de la unidad o eje temático:

02/06 al 20/06

Bibliografía básica:

Algoritmos, datos y programas con aplicaciones en Pascal, Delphi y Visual Da Vinci.

De Giusti, Armando et al. 1er edición. Prentice Hall 2001.

Apuntes de Cátedra.

Introduction to algorithms

Comen, Leiserson. MIT Press 2001.

Estructuras de Datos y Algoritmos.

Aho Alfred, Hopcroft John y Ullman Jeffrey. Addison Wesley Publishing Company. EUA. 1988.

Bibliografía complementaria:

Estructuras de Datos. Libro de Problemas.

Joyanes Aguilar L., Fernandez M., Rodríguez L. Mc Graw Hill. 1999.



ANEXO 1 Del Programa De La Asignatura

Reglamento de la cursada 2013

INFORMACION

Toda la información disponible es publicada en el espacio virtual de la cátedra a través de la plataforma Moodle.

<http://licenciaturaensistemas.unrn.edu.ar/plataformavirtual/login/index.php>

Los alumnos podrán consultar temas específicos de práctica o teoría.

Los docentes de la cátedra atenderán personalmente consultas durante sus clases.

ALUMNOS

El curso incluye asistencia obligatoria a las clases prácticas, rendir exámenes parciales de los trabajos prácticos en condiciones equivalentes a los de todos los alumnos de la asignatura.

REGLAMENTO GENERALIDADES

- El curso consta de clases teóricas, explicaciones de práctica y prácticas en el aula y actividades de laboratorio.
- La teoría es dictada por un Profesor a cargo. La coordinación de las tareas de los jefes de trabajo práctico y ayudantes está a cargo del profesor a cargo. La explicación de práctica está a cargo de los Jefes de Trabajos Prácticos. La explicación de práctica podrá darse en la misma teoría, en un horario especial o distribuido en los horarios de práctica.
- Los Jefes de trabajos prácticos asignados son responsables de los aspectos administrativos relacionados con los alumnos, y a ellos deben dirigirse las consultas de este tipo, en forma presencial en el horario de práctica u horarios especiales que fijen y/o virtualmente a través del aula de la cátedra.
- El mecanismo de promoción de la Teoría será supervisado por el Profesor.
- Durante la cursada los Profesores coordinarán con los alumnos la fecha para rendir el parcial de teoría. La aprobación del mismo con nota 7 o superior significará la aprobación del examen final para los alumnos que aprueben los trabajos prácticos de la asignatura.

REGLAMENTO DE TEORIAS

- Es aconsejable pero no obligatoria, la asistencia a las clases de teoría. En ellas, se exponen los temas del cronograma y se trabaja sobre los mismos. Se atienden consultas referidas a los temas tratados y se supervisa el avance en las prácticas.

REGLAMENTO DE PRÁCTICAS

- Los docentes de la práctica presentarán situaciones problemáticas, promoviendo la participación de los alumnos para su resolución. Luego los alumnos podrán organizarse en grupos para la resolución de la práctica contando con la guía de los docentes. Los docentes contestarán durante el horario de clase preguntas relacionadas con la práctica que indique el Cronograma, y de serle solicitado, de la práctica anterior.
- En los horarios de consulta que se publiquen se atenderán dudas de cualquier práctica. Es necesario que los alumnos estén al inicio de dichos horarios, pues de lo contrario el ayudante supone que no asistirán a consultar y se retira.
- A través de las herramientas de comunicación provistas por el aula virtual es posible realizar consultas a cualquiera de los docentes de la cátedra

1. Asistencia a Prácticas

- ³⁵₁₇ Es obligatoria la asistencia a las prácticas. En ellas, se atienden consultas referidas a los trabajos prácticos de cada semana.
- ³⁵₁₇ En cada clase práctica los alumnos tendrán presente, ausente, o ausente justificado.
- ³⁵₁₇ La asistencia a cada clase práctica será tomada durante el horario de clase, por el ayudante a cargo del aula o el jefe de trabajos prácticos. Si un alumno no se encuentra en el aula por cualquier motivo, tendrá ausente.
- ³⁵₁₇ Los alumnos, en las clases prácticas, pueden consultar la cantidad de ausentes al ayudante. La atribución de poner el presente/ausente es exclusiva de la cátedra.
- ³⁵₁₇ Pueden justificarse ausentes solamente por razones de salud o trabajo, presentando certificado.
- ³⁵₁₇ El certificado, para ser tenido en cuenta, debe ser entregado al jefe de trabajos prácticos.
- ³⁵₁₇ Para poder rendir los parciales es necesario contar con el 60% (presentes + ausentes justificados), sobre el total de clases prácticas del cuatrimestre.

2. Entregas de actividades de laboratorio

Durante el cuatrimestre se desarrollarán 3 actividades de laboratorio, denominadas **actividades prácticas entregables (APE)**. Estas son integradoras de los conceptos tratados en las unidades de teoría y práctica. Y su desarrollo servirá de información para los docentes y de orientación para el alumno. El rendimiento de los alumnos en estas actividades será considerado, a favor del alumno, en el caso de que alguno de sus parciales resulte dudoso.

La fecha de publicación y *entrega de las APE* las establecen los Jefes de Trabajos Prácticos en conjunto con los profesores, y serán publicadas en el cronograma de la materia. Si alguna de estas fecha se modifica, esta situación se publicará en las novedades del aula virtual. En la práctica siguiente a la fecha de publicación de la APE, se destinará un tiempo, para que los grupos de trabajo puedan reunirse y trabajar con el tutor sobre la misma.

Las evaluaciones de las APE no tienen una nota cuantitativa, sino que tendrán como resultado entregado o no entregado.



El proceso de investigación que generan estas actividades estará a cargo de los Jefes de Trabajos Prácticos y los Profesores quienes se desempeñarán como tutores de los grupos de trabajo.

3. Parciales de trabajos prácticos

Para obtener la aprobación de los trabajos prácticos se debe rendir y aprobar un examen parcial individual.

El alumno tendrá una fecha de examen parcial con dos instancias recuperatorias. En el cronograma del curso 2013 estarán indicadas las fechas tentativas. Cualquier cambio se avisará en las clases presenciales o en la cartelera del aula virtual.

Antes de cada fecha de examen se publicarán las listas con los alumnos en condiciones de rendir, para que en caso de error, el alumno pueda plantear su situación al jefe de trabajos prácticos, en el horario que se establezca.

Los parciales deben entregarse escritos en forma legible, con las hojas numeradas y firmadas.

4. Muestra de Parciales de práctica

- ³⁵₁₇ Los parciales son corregidos por los docentes de la práctica y revisados por los profesores.
- ³⁵₁₇ Los parciales pueden ser vistos y consultados por los alumnos en forma individual, personal y exclusivamente en el día y hora que se indique para su muestra. Una vez que el alumno recibe el parcial **NO** puede retirarlo del aula y debe devolverlo.
- ³⁵₁₇ Los resultados se publicarán a través del aula virtual de la cátedra y en la cartelera de primer año.