

INTEGRA+506: Una Metodología para el diseño y desarrollo de juegos serios

Juan Carlos Sandí-Delgado^{1,2} [0000-0003-3932-3045] and Patricia Bazán² [0000-0001-6720-345X]

¹ Recinto de Guápiles - Sede del Atlántico, Universidad de Costa Rica, Costa Rica
juan.sandidelgado@ucr.ac.cr

² LINTI Facultad de Informática, Universidad Nacional de La Plata, Buenos Aires, Argentina
pbaz@mail.info.unlp.edu.ar

Resumen. La formación de competencias tecnológicas en educación superior se ha vuelto un factor imprescindible en el ámbito educativo. En este sentido, los juegos serios han sido utilizados para entrenamiento, la formación, capacitación y adquisición de habilidades en diferentes áreas del conocimiento. Por tal razón, se realizó una investigación con el objetivo de brindar una propuesta metodológica y arquitectónica que guíe paso a paso el diseño de juegos serios vinculados con la formación de dichas competencias. En consecuencia, se planteó una metodología y arquitectura de software para diseñar un prototipo no funcional del juego serio denominado SkillNet, el cual permite la adquisición de estas competencias en educación superior. Para llevar adelante la investigación, se realizó una indagación bajo un enfoque metodológico cualitativo y un diseño descriptivo, donde se realizó primeramente una revisión teórica acorde con los temas de investigación; principalmente los relacionados con metodologías y arquitecturas utilizadas para el diseño de juegos serios, así como las vinculadas con el diseño de prototipos. Posteriormente, se diseña y propone la metodología denominada INTEGRA+506 la cual se conforma de 6 etapas y quince pasos metodológicos que guían el diseño de juegos serios orientados a la formación de competencias tecnológicas. Por último, se concluye que la metodología INTEGRA+506 mostró efectividad para guiar el trabajo realizado en el diseño del prototipo no funcional de SkillNet y su aplicación ayudó a ordenar todo el proceso acorde con la validación realizada por el panel de expertos que revisó y analizó el prototipo.

Palabras clave: Juegos serios; diseño y desarrollo de juegos serios; desarrollo de competencias digitales; estrategias educativas; innovación educativa.

1. Introducción

Los juegos serios han sido utilizados para la formación en competencias digitales, entrenamiento y adquisición de habilidades en diferentes áreas del conocimiento, las cuales pueden contribuir a la generación de emociones y sensaciones, servir de apoyo para comprender diferentes procesos sociales, de salud, económicos y educativos (Khaleghi et al., 2022; Mercier & Lubart, 2023). En

esta línea, un juego serio se podría definir como aquel juego que tiene un enfoque que va más allá del entretenimiento o la diversión, el cual posee un propósito educativo explícito y cuidadosamente pensado, es decir, se puede utilizar para potenciar diferentes áreas del conocimiento, ofrecer un aprendizaje, apoyar procesos formativos, promover cambios de actitud o comportamiento, generar emociones, potenciar la adquisición de habilidades, entre otros. Caracterizado principalmente por ser atractivo, interactivo,

entretenido-dinámico, motivador, desafiante, fácil de utilizar, con la capacidad de estimular la colaboración, el razonamiento y el pensamiento crítico.

Actualmente, no se evidencia en la literatura consultada, diseñados o arquitecturas de estos juegos pensados exclusivamente para potenciar habilidades y competencias tecnológicas en el profesorado a nivel de educación superior. Por tal razón, resultó de interés elaborar una propuesta metodológica y arquitectónica que guíe paso a paso su diseño. En consecuencia, se planteó una metodología y arquitectura de software para diseñar un prototipo no funcional del juego serio denominado SkillNet, para permitir la formación de habilidades tecnológicas en el profesorado; específicamente las relacionadas con competencias y conocimientos básicos en redes de datos. La metodología se denominó INTEGRA+506 por la integración de metodologías de diseño y arquitectónicas (INTEGRA), luego, +506 por ser el código de Costa Rica, país de origen de la metodología.

Este tema es de alto interés, relevancia y trascendencia en la actualidad, debido a que aún existe una carencia en la definición de metodologías específicas para la creación y diseño de juegos serios en el ámbito del desarrollo de las competencias tecnológicas. Por ende, la metodología INTEGRA+506 es una contribución original dicha área de investigación, donde se realizó una investigación en profundidad en relación con todos los aspectos relacionados con procesos de diseño de juegos serios en el ámbito educativo. Se ha tenido en cuenta tanto el punto de vista educativo, como el metodológico y el tecnológico, incluyendo el estudio de las diferentes arquitecturas y los diferentes aspectos a considerar en el desarrollo de software.

2. Referente teórico

En esta sección se brinda una conceptualización de juegos serios y sus principales características, se adopta una definición propia, lo cual permite brindar contexto a la investigación realizada. Asimismo, se describe el concepto de competencias digitales.

2.1 Juegos Serios

El concepto de juegos serios se conoció por primera vez en 1970 por (Abt, 1970), quien indicó que los mismos tienen una intención educativa explícita, los cuales son cuidadosamente pensados y que su objetivo pedagógico va más allá del entretenimiento. Los juegos serios facilitan la adquisición de habilidades en diferentes áreas del conocimiento a través de la instrucción, simulación, entrenamiento o educación, facilitan principalmente el placer y la motivación (Abt, 1970; Kwon & Lee, 2016).

2.2 Competencias tecnológicas

El término competencias digitales ha sido asociado al dominio de variedad de herramientas tecnológicas, a las habilidades asociadas y requeridas para su correcta integración de estas en las diferentes áreas del conocimiento (Rodríguez-García et al., 2019).

En distintas investigaciones (Knezek et al., 2023; Sanz et al., 2022) se indica que las competencias digitales son consideradas como competencias claves, destacadas y transversales, debido a que potencian la formación de otras competencias, tales como la lingüística y la matemática, requeridas por la ciudadanía para garantizar una participación activa en la sociedad y la economía.

Al respecto, el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF, 2017), señala que las competencias digitales están relacionadas con el uso creativo, seguro y crítico de las TIC para alcanzar los objetivos afines al entretenimiento, el aprendizaje, la inserción y contribución en la sociedad, apoyadas en las competencias básicas relacionadas al uso de tecnologías digitales. Además, dichas competencias están relacionadas con el saber, saber hacer, saber estar y saber ser (Hernández et al., 2016; Lee et al., 2023).

Las competencias digitales se podrían definir como un conjunto de saberes relacionados con el correcto uso, dominio e integración de las tecnologías digitales y otras áreas del conocimiento relacionadas, que a través de diferentes habilidades, actitudes y valores permiten a las personas el saber ser, es decir, desarrollar un comportamiento propio ante un evento para ser capaz de potenciar el mejoramiento personal, académico o profesional, y con ello, garantizar su participación en la sociedad.

En la próxima sección, se aborda el marco metodológico seguido en la investigación, así como una descripción detallada de la metodología INTEGRA+506 (INTEGRA por la integración de metodologías de diseño y arquitectónicas y +506 por ser el código del país de origen de la metodología) la cual ha sido propuesta con el fin de integrar aspectos metodológicos y arquitectónicos en el diseño de juegos serios para potenciar en el profesorado la formación de competencias tecnológicas.

3. Metodología

Para alcanzar el objetivo propuesto, se realizó una investigación fundamentada en el diseño descriptivo, con un enfoque metodológico cualitativo (Hernández Sampieri et al., 2014), la cual se diseñó en tres fases; en la primera fase se realiza una búsqueda, revisión y selección de referencias bibliográficas relacionadas a los temas y objetivos propuestos en esta investigación. Lo cual permite consolidar el marco referencial de la investigación que incluye los temas de competencias digitales y juegos serios. En la segunda fase, se elabora la propuesta metodológica y arquitectónica (INTEGRA+506) que permite brindar solución al problema planteado (guiar paso a paso el proceso de planificación y desarrollo de Juegos Serios). Los pasos metodológicos propuestos, se formulan con base en las principales recomendaciones existentes en la literatura, identificación y análisis de metodologías y modelos propuestos para el diseño e implementación de video juegos y juegos serios que han sido comúnmente más utilizados, tales como EMERGO, EDoS, SAVIE, DODDEL, VGSCS, MECONESIS, MPIu+a y MPDSG, las cuales propiciaron el desarrollo del prototipo de juego serio denominado SkillNet.

En la tercera fase, se realiza la validación de la metodología propuesta a través de juicio de expertos y mediante su aplicación en el desarrollo de un prototipo de juego

serio denominado SkillNet, proceso que sienta las bases para la integración de metodologías y arquitecturas enfocadas en el diseño de juegos serios.

4. Propuesta metodológica y arquitectónica INTEGRA+506

En el presente apartado se aborda en detalle los pasos metodológicos y arquitectónicos presentes en la metodología INTEGRA+506 la cual guía paso a paso el diseño de juegos serios orientados a la formación de competencias tecnológicas, siguiendo las principales consideraciones teóricas en relación con el desarrollo de software, metodologías y arquitecturas para el diseño y desarrollo de juegos serios.

Por tanto, se propone una metodología compuesta por 6 fases, donde cada fase incluye diferentes pasos metodológicos y arquitectónicos (PM&A) que guía el diseño del prototipo no funcional del juego SkillNet. Para la definición de las fases de la metodología se consideraron aspectos teóricos del desarrollo de software, metodologías y arquitecturas para el desarrollo de videojuegos. Ahora bien, las fases como los pasos metodológicos podrían ser considerados de forma distinta dependiendo de la experiencia o área de formación de la persona o investigador que las analiza.

4.1 Fase 1. Consideraciones generales

Esta primera fase es la más importantes de la metodología, ya que se requiere definir los roles o actores participantes durante el desarrollo del prototipo, así como establecer las competencias a potenciar y el tipo de aplicación que se utilizará para ejecutarlo.

- **FM&A01 Definir roles o actores participantes** - El propósito de este paso consiste en identificar el equipo interdisciplinario, es decir, las personas expertas más recomendadas para participar en la definición de la propuesta del juego serio, tales como expertos pedagógicos, en el tema, de contenido, desarrolladores, pedagogos, entre otros (Spinelli & Massa, 2018), máxime que la definición de los requerimientos de usuario son influyentes en relación con la calidad de la pieza de software por desarrollar (Alshamrani & Bahattab, 2015).
- **FM&A02 Seleccionar competencias – Definir objetivos** - Al diseñar un juego serio para la formación de competencias tecnológicas, se debe considerar en primera instancia la definición de los requerimientos, luego, se define claramente los objetivos del juego, es decir, seleccionar previamente las competencias que se busca potenciar. En este sentido, existen estudios previos que han definido y clasificado las competencias en dimensiones e indicadores, tal es el caso de Prendes & Gutiérrez (2013) en España, Ministerio de Educación Nacional Colombiano (Sandí-Delgado, 2019) y por el Consejo Universitario de la Universidad de Costa Rica (UCR) (UCR, 2004). Lo anterior, coincide con lo indicado por (Yeh et al., 2017) quien agrega que para integrar las TIC en los procesos formativos es necesario poseer habilidades tecnológicas, pedagógicas y disciplinares, competencias que han sido también propuestas por el modelo de Conocimiento Técnico Pedagógico del Contenido (TPACK, *Technological Pedagogical Content Knowledge*).
- **FM&A03 Seleccionar el tipo de aplicación** - Posterior a la selección de las

competencias tecnológicas a potenciar en el profesorado a través del prototipo del juego serio por diseñar, se debe realizar la selección del tipo de aplicación sobre la cual se desarrollará el juego serio (Hainey et al., 2016), para ello, se debe considerar aquellos que hayan sido utilizados con mayor frecuencia para la ejecución de juegos serios. En este sentido, en una investigación realizada por (Sandí-Delgado et al., 2022), se consideran 5, entre ellas: para PC, las orientadas a trabajar en línea (web), video consolas, móviles y finalmente, las orientadas a la virtualidad. El tipo de aplicación a seleccionar va a depender de los objetivos propuestos y de los posibles recursos con que cuente la institución y el profesorado directamente involucrado. El prototipo de juego serio requiere de poder acceder a la aplicación desde cualquier computadora o dispositivo conectado a internet, el cual podría utilizar diferentes navegadores web (Albert & Torres, 2022a).

4.2 Fase 2. Lenguaje de programación

Para el diseño de una pieza de software y, en particular, el prototipo de un juego serio, se requiere de la definición de un paradigma y tipo de lenguaje de programación (IEEE Computer Society, 1990; Prasad & Verma, 2016).

- **FM&A04 Paradigma de programación** - Para el prototipo no funcional del juego serio, se requiere que se defina un paradigma de programación, ya que este indicará el método para su construcción, además, permitirá organizar las tareas que debe llevar a cabo la pieza de software, además, los mismos definen los estilos de programación (INET, 2016). Se podría elegir entre paradigmas de programación no estructurada, programación estructurada y programación orientada a objetos.
- **FM&A05 Tipo de lenguaje de programación** - En relación con el lenguaje de programación para crear la aplicación del prototipo no funcional, se puede utilizar una variedad de lenguajes enfocados en el desarrollo web. La estructura de una aplicación web consiste en el Frontend (la interfaz orientada al usuario), y el Backend (código interno para el manejo de datos del Frontend) acorde con lo indicado por (Gagliardi, 2021; Jones, 2020; Slivnik, 2022).
 1. Lenguajes para el desarrollo **Frontend**: Comúnmente se utiliza para el desarrollo de *Frontend* lenguajes como JavaScript (brinda dinamismo a los sitios web), CSS (aporta estilos, diseños y formatos a la página web) y el Lenguaje de Marcado de Hipertexto (HTML, *HyperText Markup Language*) el cual provee una estructura de etiquetas y maquetado para el desarrollo del sitio web.
 2. Lenguajes para el desarrollo del **Backend**: Existe variedad de lenguajes para desarrollo del *Backend*, entre ellos se encuentra PHP (*Hypertext Preprocessor*), Python, Java, C++, JavaScript, entre otros, los cuales pueden ser utilizados para el desarrollo del prototipo.

Dependerá del desarrollador cual lenguaje va a utilizar para el desarrollo del prototipo, máxime que la selección de este queda sujeta a la experiencia del usuario (desarrolladores) y de los objetivos por alcanzar.

4.3 Fase 3. Procesos de software

Acorde con lo indicado por (Sommerville, 2016), se debe definir un proceso de software para guiar todas las actividades relacionadas con la producción de la pieza de software, en este caso, el modelo de desarrollo y el diseño centrado en el usuario.

- **FM&A06 Modelo de desarrollo de software** - Para el diseño de la pieza de software por desarrollar, se podría utilizar un modelo de cascada, modelo de proceso incremental, modelo de prototipos, modelo espiral, modelo de desarrollo de aplicación rápida o bien, un modelo de ciclo de vida ágil.
- **FM&A07 Diseño centrado en el usuario** - Para el diseño del prototipo se requiere de un diseño centrado en el usuario para ofrecer una mayor satisfacción y experiencia a nivel del usuario (UX). Lo anterior, debido a que se requiere que los usuarios influyan sobre las decisiones de diseño y sobre los objetivos pretendidos por alcanzar con el prototipo. En relación con los estilos de componentes a nivel de interfaz para prototipos, se recomienda que para el manejo del diseño y presentación de la aplicación, se definan los estilos que presentará el sitio web, dentro del diseño el intérprete que asignará los estilos es el lenguaje Hojas de Estilo en Cascada (CSS *Cascading Style Sheets*), para la utilización de este, existe variedad de opciones a elegir, entre ellos *Frameworks* sencillos de utilizar, tales como Bootstrap, Bulma, MDBostrap, que permiten brindar una identidad distinta al estilo, según el *Frameworks* elegido. De igual manera se podría utilizar CSS en su versión pura “vanilla”, la cual se caracteriza por ser ligera, adaptable y de código abierto (Malgarejo-Torralba et al., 2022).

4.4 Fase 4. Ingeniería de requerimientos

Esta fase es de importancia en el desarrollo de piezas de software (Sommerville, 2016), las partes involucradas deben definir los requerimientos generales, ya que a como una buena especificación de requerimientos es determinante en la calidad del software (Gómez-Fuentes, 2011). Para ello, se debe considerar los requerimientos funcionales y no funcionales, así como definir claramente los casos de uso, las reglas de negocio y la ficha del juego.

- **FM&A08 Requerimientos funcionales y no funcionales** - Para el diseño de software se requiere definir claramente los servicios que el sistema debe proporcionar e indicar cuales serían las restricciones de esta.
- **FM&A09 Casos de uso** - Para el diseño del prototipo no funcional del juego serio, se requiere la descripción de las acciones o actividades que permitan concretar el proceso del diseño final (Sommerville, 2016).
- **FM&A10 Reglas de negocio** - Para el diseño del prototipo no funcional del juego serio, se requiere la definición de las reglas de negocios, las cuales tienen como objetivo definir restricciones o límites que permitan alcanzar el objetivo del proyecto.
- **FM&A11 Ficha de juego** - Los juegos serios usualmente, suelen tener asociada una ficha de juego, en la cual se cuenta brevemente su **historia** (de qué trata el juego), así como ofrecer las **indicaciones del cómo jugarlo** y, finalmente, incluir el aspecto más caracterizante de los juegos serios, el cual corresponde a la

estrategia de ludificación, es decir, el juego describe la metodología de ludificación que utiliza (medallas, ranking, insignia, puntajes y premiación) (Sandí-Delgado & Sanz, 2020), lo cual es un aspecto relevante para considerar, debido a que los juegos serios se caracterizan por la intención pedagógica y, por incorporar estrategias lúdicas (Sandí-Delgado, 2019).

4.5 Fase 5. Diseño de software – Aspectos arquitectónicos

En el diseño de software se requiere generar una estructura que permita relatar el “como” se logra cada función propuesta del software mediante una especificación de propiedades de objeto (Leung & Cockburn, 2021). Entre las especificaciones, se requiere describir como mínimo el estilo arquitectónico (con todos sus componentes) y el diagrama de flujo de datos.

FM&A12 Estilo arquitectónico - El estilo arquitectónico puede estar integrado por diferentes componentes, los cuales varían acorde con la experiencia del desarrollador, el objetivo del juego y tipo de aplicación a la que está orientada la pieza de software.

1. **Modelo, Vista, Controlador** – En modelo, vista, controlador para prototipos, se requiere para separar de manera lógica la estructura de programación en una segregación por capas (López Caballero et al., 2019).
2. **Microservicios (web services)** – Se requiere la implementar un web services para el consumo de recursos almacenados en la base de datos donde estará alojada la información del juego, tales como información general del usuario y métricas: tiempo y puntuaciones (López Caballero et al., 2019).
3. **Entorno de desarrollo** - En relación con los estilos de componentes a nivel de interfaz para prototipos, se requiere seleccionar un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*) adecuado al lenguaje que se va a utilizar (Bazán et al., 2017; Yulianto et al., 2017), en este sentido, las opciones a utilizar podrían ser amplias y con alta variedad de software entre el cual elegir. Dentro de las opciones más utilizadas en el mercado para el desarrollo web están: Visual Studio Code debido a su alta flexibilidad para trabajar como lenguaje, WebStorm y SublimeText, entre otros (Janssen et al., 2019; Yun et al., 2020).
4. **Framework o Librerías** - Para el desarrollo del prototipo no funcional del juego serio se podría utilizar un *Frameworks*, el cual permite el desarrollo a través de un estructura o esqueleto (es decir, es un marco de trabajo), el cual aporta flexibilidad para el manejo tanto del *Backend* como del *Frontend*, debido a que estos marcos de trabajo cuentan con librerías creadas por la misma comunidad de desarrolladores que los utilizan (Leung & Cockburn, 2021), estos podrían facilitar el desarrollo de piezas de software dependiendo de las circunstancias a implementar (Cabrera, 2017). En la actualidad, existe una variedad de Frameworks para JavaScript tales como React, AngularJS, Vue.js (Albert & Torres, 2022b), cada uno posee sus características particulares a nivel de usabilidad y robustez.
5. **Gestor de base de datos** - Se requiere utilizar un Sistema Gestor de Base de Datos (DBMS, *Database Management System*) para administrar el acceso por parte de los usuarios a la base de datos, así como permitir la realización de consultas y generación de informes en relación con los avances de los jugadores (Hueso-

Ibáñez, 2015). El DBMS usualmente cumple con el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones (Díaz-Eraza et al., 2022). Para el almacenamiento de los datos se podría utilizar varios gestores, tales como MySQL, SQL, Oracle, entre otras. Estos gestores permiten almacenar ordenadamente la información generada por los jugadores.

6. **Alojamiento** - En el prototipo no funcional del juego serio se requiere alojar la aplicación en la web para garantizar su disponibilidad y alcance al usuario. Lo más utilizado actualmente, son la contratación de servicios de proveedores de computación en la nube (Cobeña, 2021). Algunas opciones para alojar la aplicación (juego, API, entre otros) son Azure, AWS, *Google Cloud Platform*, entre otras (Albert & Torres, 2022a), las cuales usualmente ofrecen un costo económico por el servicio consumido de alojamiento. Igualmente, se podría valorar la opción de buscar alternativas gratuitas como GitHub.
7. **Autenticación de usuarios** - En el prototipo no funcional del juego serio se requiere de la autenticación de usuarios, la cual se podría realizar de distintas maneras; utilizar librerías como (JWT, *Json Web Token*) o Auth0, las cuales brindan seguridad garantizada en el control de acceso de los usuarios (Díaz-Eraza et al., 2022). Este proceso es requerido para garantizar la privacidad, integridad y calidad de los datos de cada jugador, así como mantener un registro individual de la actividad y alcances de cada usuario.
8. **Versionado** - El prototipo de juego serio requiere ofrecer una aplicación simple, que se mantenga una única versión compartida para todos los clientes (Sandí-Delgado & Bazán, 2017). Se podrían utilizar herramientas existentes para el control de versiones, tales como Git (sistema de control de versiones distribuido), el cual ofrece diferentes clientes tales como GitHub y GitLab, Subversión (sistema de control de versiones centralizado) y Mercurial (sistema de control de versiones multiplataforma), siendo este último el menos utilizado (Albert & Torres, 2022a). La diferencia más importante entre Git y Subversión consiste en que, en los sistemas centralizados (Subversión) se mantiene únicamente una copia de la aplicación y las copias permanecen respaldadas en un solo lugar. En los sistemas distribuidos que hacen uso de Git, se mantiene una copia del proyecto original en cada equipo local y los cambios se actualizan o suben al repositorio en línea, los cuales posteriormente deben ser descargados de dicho repositorio para actualizar la versión de cada uno de los desarrolladores (Albert & Torres, 2022a). El versionado de los diferentes repositorios (*frontend*, *backend*) de las aplicaciones web se pueden actualizar sin interrumpir su flujo, al desplegar la aplicación con la nueva versión del repositorio la misma se actualiza con sus nuevas funcionalidades o correcciones.

FM&A13 Diagramas de clases (UML) y Diagrama entidad relación (ER) - En el diseño del prototipo se requiere de la definición de un diagrama de clases y un diagrama entidad relación debido a que este permite identificar claramente cuál será el flujo de la información durante el desarrollo de la pieza de software. Para ello, se pueden utilizar herramientas gratuitas tales como Draw.io, StarUML.io, Diagrams.net, o bien de paga como Visio. En estos diagramas se hace uso de simbología tal como el

círculo, diferentes tipos de flechas, rectángulos, texto entre otros, los cuales permiten visualizar la entrada y salida de datos (Sommerville, 2016).

4.6 Fase 6. Codificación del software

En el diseño del prototipo se sugiere contemplar la codificación del software, debido a que permite traducir todas las especificaciones diseñadas en código fuente, haciendo más fácil el leer y comprender la pieza de software desarrollada (Prasad & Verma, 2016). Lo anterior, permite obtener un estilo de programación, pautas de codificación, técnicas de codificación estructuradas y técnicas de verificación del código. En esta fase se analiza tanto las características del código de software como la reutilización de sus componentes (Sommerville, 2016).

- **FM&A14 Características del código de software** - El código del software debe estar escrito en congruencia con los requisitos del usuario, es decir, apegado a los estándares y buenas prácticas de programación, eficiente en sus algoritmos, con manejo de excepciones, validaciones y medidas de seguridad, garantizando un mínimo margen de errores (Prasad & Verma, 2016). Se pueden implementar algunas de las siguientes medidas: simplicidad de código (simple y conciso), modularidad (dividido en varios módulos), diseño (estilo específico y consistente), eficiencia (uso óptimo de los recursos disponibles), claridad (simplicidad, legibilidad y modularidad), accesibilidad (fácilmente disponibles y accesibles) o estabilidad (funcionar correctamente en diferentes plataformas sin afectar su diseño y consistencia).
- **FM&A15 Reutilización de componentes** - En el diseño del prototipo se sugiere contemplar la reutilización de código y componentes dinámicos de software, máxime si se piensa en una aplicación web, ya que facilita el mantenimiento, escalabilidad y funcionalidad de la aplicación (Prasad & Verma, 2016). La reutilización del software es posible de implementar en diferentes niveles: Nivel de abstracción, nivel del objeto, nivel del componente y nivel del sistema (Sommerville, 2016).

En la Figura 1, se puede observar el resumen de las fases y pasos metodológicos y arquitectónicos de la metodología INTEGRA+506.

4.7 Validación de la propuesta a través de un prototipo

En el desarrollo de software se recomienda que las propuestas metodológicas sean validadas por profesionales y expertos en el área, por ello, en este caso, se recomienda diseñar un prototipo de juego serio siguiendo los pasos metodológicos propuestos a través de INTEGRA+506, posteriormente, hacer uso del juicio de expertos para validar el prototipo diseñado.

Fase	Pasos metodológicos y arquitectónicos
Fase 1. Consideraciones generales	FM&A01 Definir roles o actores participantes. FM&A02 Seleccionar competencias – Definir objetivos. FM&A03 Seleccionar el tipo de aplicación.
Fase 2. Lenguaje de programación	FM&A04 Paradigma de programación. FM&A05 Tipo de lenguaje de programación.
Fase 3. Procesos de software	FM&A06 Modelo de desarrollo de software. FM&A07 Diseño centrado en el usuario.
Fase 4. Ingeniería de requerimientos	FM&A08 Requerimientos funcionales y no funcionales. FM&A09 Casos de uso. FM&A10 Reglas de negocio. FM&A11 Ficha de juego.
Fase 5. Diseño de software – aspectos arquitectónicos	<p>FM&A12 Estilo arquitectónico:</p> <ul style="list-style-type: none"> ✓ Modelo, vista, controlador. ✓ Microservicios (web service). ✓ Entornos de desarrollo. ✓ Framework o Librerías. ✓ Gestor de base de datos. ✓ Alojamiento. ✓ Autenticación de usuarios. ✓ Versionado. <p>FM&A13 Diagramas de clases (UML) y Diagrama Entidad Relación (ER).</p>
Fase 6. Codificación del software	FM&A14 Características del código de software. FM&A15 Reutilización de componentes

Fig. 1. Fases y pasos metodológicos y arquitectónicos de INTEGRA+506

5. Prototipo SkillNet

El prototipo con funcionalidad mínima SkillNet puede ser accedido desde el siguiente enlace: <https://skillnetucr.github.io/Prototipe/> para iniciar sesión se debe utilizar en ambos casos como usuario y contraseña: admin123@gmail.com

Este juego serio tiene como propósito la formación de habilidades tecnológicas en el profesorado; específicamente las relacionadas con competencias y conocimientos básicos en redes de datos. El juego consta de tres etapas conformadas por diferentes niveles cada una, en las cuales el jugador debe enfrentar desafíos para superarlos y desbloquear el siguiente nivel. El juego tiene un personaje principal llamado Robotín.

Robotín es un dispositivo de red que siempre ha tenido un sueño, conocer la isla del Caribe (San Andrés - Colombia). Robotín es un equipo que se encuentra en un Centro de Datos en Limón (Costa Rica) y debe salir de su entorno para tomar una embarcación (WAN), eso sí, para salir no la tendrá tan fácil, debe superar una serie de desafíos para poder pasar de su ubicación a la Isla.

El Centro de Datos está compuesto por salidas de seguridad, cada puerta podría abrirse en base a conocimientos en el área de Redes y Comunicaciones de Datos. Para poder cumplir con el sueño de Robotín se tiene que pasar por 3 ambientes o escenarios, los cuáles son los siguientes:

- Robotín está conectado en un MDF (*Main Distribution Frame*), para poder desconectarse y salir debe resolver desafíos sobre clases de direcciones del protocolo de internet, asignación de máscaras variables y subdivisión de redes, al

completar estos ejercicios puede salir al salón principal de tecnologías de información.

- En el salón principal, la clave de salida es el ordenamiento e identificación del cableado necesario para la creación de una red y los dispositivos intermediarios que la componen, donde una vez que Robotín consiga ordenar dicho salón, puede salir y embarcarse en su travesía en el Crucero Island San Andrés.
- Ya en el crucero, debe Robotín enseñarle al capitán la trayectoria del viaje a San Andrés, para completar este desafío debe conocer la confección correcta de una red del protocolo de enrutamiento. Cada vez que supera un reto y desbloquea un nivel (paso de una ubicación a otra), la dificultad aumenta y recibe como premio un alimento básico de sobrevivencia (agua, proteína, medicamentos, entre otros).

¿Cómo se gana el juego? – El jugador debe superar todas las actividades evaluativas de un nivel para lograr desbloquear el siguiente, en este caso se le abrirá una puerta que le permitirá seguir avanzando en el juego. Para ganar, debe desbloquear los 3 niveles y completar absolutamente todas las actividades en cada uno. **Estrategia de ludificación** - El prototipo ofrece 5 estrategias a saber: Sistema de puntuación, Desafíos, Niveles, Intentos por etapa y Sistema de ranking.

La metodología INTEGRA+506 cumplió con el objetivo de guiar el diseño del prototipo no funcional del juego serio llamado SkillNet, el cual está orientado a la formación de competencias tecnológicas en el profesorado.

6. Conclusiones

Se definió una metodología y arquitectura de software para el diseño de juegos serios para la formación de competencias tecnológicas. A partir de la metodología propuesta denominada INTEGRA+506 se diseñó un prototipo con funcionalidad mínima de un juego serio denominada SkillNet, para potenciar competencias digitales en el profesorado, específicamente vinculadas con conocimientos básicos en redes de datos.

En relación con las TIC y competencias tecnológicas, se caracterizaron ampliamente ambos términos, a partir de sus aspectos caracterizantes identificados, se brindaron definiciones propias, resultado de una investigación teórica realizada. Luego, se indagó el concepto de juegos serios y se identificaron juegos que se están utilizando para el desarrollo de competencias tecnológicas. Se brindó una definición propia. Se evidenció que existe una ausencia de metodologías guíen el diseño de juegos serios específicamente para la formación de competencias tecnológicas. Por ello, se analizaron propuestas de metodologías y arquitecturas para el diseño de videojuegos y juegos serios en particular, a partir de esto, se elaboró una propuesta metodológica y arquitectónica para el desarrollo de juegos serios orientados a la formación de competencias tecnológicas denominada INTEGRA+506, la cual mostró efectividad para guiar el diseño del prototipo SkillNet. Su aplicación ayudó a ordenar todo el proceso. Sin embargo, resta aún la parte funcional y ampliar su aplicación a otros escenarios educativos.

La metodología INTEGRA+506 podrían resultar de muy alta relevancia y trascendencia a nivel nacional e internacional para investigadores y desarrolladoras de videojuegos serios, ya que les podría permitir tomarla como guía, base o referente al momento de desarrollar juegos serios. Es importante indicar que la metodología en mención permitió desarrollar el prototipo de SkillNet, pero el mismo aún no ha sido implementado o aplicado con docentes, lo cual se espera realizar en un mediano plazo, en su debido momento, se describirá los resultados que se obtengan en relación con el uso del prototipo en espacios de formación en competencias tecnológicas. Se debe trabajar en la aplicación de la propuesta metodológica INTEGRA+506 para el diseño de prototipos de juegos serios en diferentes ámbitos educativos, así como en distintas áreas del conocimiento y competencias.

Referencias

1. Abt, C. C. (1970). *Serious Games*. Viking Press.
2. Albert, M., & Torres, M. V. (2022a). Desarrollo de una aplicación web como catálogo de buenas prácticas de accesibilidad. Universitat Politècnica de València.
3. Albert, M., & Torres, M. V. (2022b). *Desarrollo de una aplicación web como catálogo de buenas prácticas de accesibilidad* [Universitat Politècnica de València]. <http://hdl.handle.net/10251/188224>
4. Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model , Spiral Model , and Incremental / Iterative Model. *International Journal of Computer Science*, 12(1), 106–111. <https://ijcsi.org/papers/IJCSI-12-1-1-106-111.pdf>
5. Bazán, P. A., Fernández, A., Del Río, N., Molinari, L., Pérez, J. P., & Banchoff, M. (2017). *Aplicaciones, servicios y procesos distribuidos: Una visión para la construcción de software* (P. A. Bazán (ed.); 1st ed.). Editorial de la Universidad Nacional de La Plata (EDULP). <http://hdl.handle.net/10915/62354>
6. Cabrera, J. (2017). *Modular Design Frameworks: A Projects-based Guide for UI/UX Designers* (K. Jameson (ed.); 1st ed.). Apress. <https://doi.org/10.1007/978-1-4842-1688-0>
7. Cobeña, J. A. (2021). *Plataformas de computación en la nube para el desarrollo de aplicaciones web* [Pontificia Universidad Católica del Ecuador Sede Esmeraldas]. <https://repositorio.pucese.edu.ec/handle/123456789/2595>
8. Díaz-Eraza, A. D., Morales-Morales, R. M., Pineda-Chávez, V. K., & Morales-Cardoso, L. S. (2022). Comparative Analysis of performance for SQL and NoSQL Databases. *17th Iberian Conference on Information Systems and Technologies*, 1–14. <https://doi.org/10.23919/CISTI54924.2022.9820292>
9. Gagliardi, V. (2021). Decoupled Django: Understand and Build Decoupled Django Architectures for JavaScript Front-ends. In J. Markham (Ed.), *Decoupled Django*. Apress. <https://doi.org/10.1007/978-14842-7144-5>
10. Gómez-Fuentes, M. del C. (2011). *Notas Del Curso Análisis De Requerimientos* (M. del C. Gómez (ed.); 1st ed.). Universidad Autónoma Metropolitana, Unidad Cuajimalpa. http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf
11. Hainey, T., Connolly, T. M., Boyle, E. A., Wilson, A., & Razak, A. (2016). A systematic literature review of games-based learning empirical evidence in primary education. *Computers & Education*, 102 (November), 202–223. <https://doi.org/10.1016/j.compedu.2016.09.001>
12. Hernández, C. A., Arévalo, M. A., & Gamboa, A. A. (2016). Competencias TIC para el desarrollo profesional docente en educación básica. *Praxis & Saber. Revista de Investigación y Pedagogía*, 7(14), 41–69. <https://doi.org/10.19053/22160159.5217>

13. Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). Análisis de datos cuantitativos. In *Metodología de la investigación* (6th ed.). Mc Graw Hill. <https://www.uca.ac.cr/wpcontent/uploads/2017/10/Investigacion.pdf>
14. Hueso-Ibáñez, L. (2015). *Administración de Sistemas Gestores de Bases de Datos* (2nd ed.). RA-Ma.
15. IEEE Computer Society. (1990). IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology. *Ieee*, *121990*(1), 1. <https://doi.org/10.1109/IEEESTD.1990.101064>
16. INET. (2016). *Programación Orientada a Objetos*. <http://ce aer.edu.ar/wp-content/uploads/2018/04/ApunteTeorico-de-Programacion-OO.pdf>
17. INTEF. (2017). *Marco Común de Competencia Digital Docente*. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). <http://educalab.es/documents/10180/12809/Marco+competencia+digital+docente+2017/afb079871ad6-4b2d-bdc8-58e9faeccc>
18. Janssen, J., Surendralal, S., Lysogorskiy, Y., Todorova, M., Hickel, T., Drautz, R., & Neugebauer, J. (2019). pyiron: An integrated development environment for computational materials science. *Computational Materials Science*, *163*, 24–36. <https://doi.org/10.1016/j.commatsci.2018.07.043>
19. Jones, C. B. (2020). *Understanding Programming Languages*. Springer Nature Switzerland. <https://doi.org/10.1007/s11023-007-9062-6>
20. Khaleghi, A., Aghaei, Z., & Behnamghader, M. (2022). Developing two game-based interventions for dyslexia therapeutic interventions using gamification and serious games approaches entertainment computing journal. *Entertainment Computing*, *42*, 100482. <https://doi.org/10.1016/j.entcom.2022.100482>
21. Knezek, G., Christensen, R., Smits, A., Tondeur, J., & Voogt, J. (2023). Strategies for developing digital competencies in teachers: Towards a multidimensional Synthesis of Qualitative Data (SQD) survey instrument. *Computers & Education*, *193*, 104674. <https://doi.org/10.1016/j.compedu.2022.104674>
22. Kwon, J., & Lee, Y. (2016). Serious games for the job training of persons with developmental disabilities. *Computers & Education*, *95*(April), 328–339. <https://doi.org/10.1016/j.compedu.2016.02.001>
23. Lee, G., Caton, E., & Ding, A. (2023). Evaluating digital competencies for pharmacists. *Research in Social and Administrative Pharmacy*, *19*(5), 753–757. <https://doi.org/10.1016/j.sapharm.2023.01.012>
24. Leung, J., & Cockburn, A. (2021). Design Framework for Interactive Highlighting Techniques. *Design Framework for Interactive Highlighting Techniques*, *14*(2–3), 1–176. <https://doi.org/10.1561/9781680839258>
25. López Caballero, V., Fragoso Díaz, O. G., Santaolaya Salgado, R., Rojas Pérez, J. C., & González Serna, J. G. (2019). Learning Web Services for E-learning in the Workplace. *IEEE Latin America Transactions*, *17*(11), 1894–1901. <https://doi.org/10.1109/TLA.2019.8986429>
26. Melgarejo-Torralba, M., Parras-Burgos, D., & Fernández-Pacheco, D. G. (2022). Hand-developed creative prototyping. Methodological proposal and experimentation. *Thinking Skills and Creativity*, *44*, 101025. <https://doi.org/10.1016/j.tsc.2022.101025>
27. Mercier, M., & Lubart, T. (2023). Video games and creativity: The mediating role of psychological capital. *Journal of Creativity*, *33*(2), 100050. <https://doi.org/10.1016/j.yjoc.2023.100050>
28. Prasad, R., & Verma, G. (2016). *Software Engineering* (1st ed.). Khanna Book Publishing Co. (P) Ltd.
29. Prendes, M. P., & Gutiérrez, I. (2013). Competencias tecnológicas del profesorado en las universidades españolas. *Revista de Educación*, *361*, 196–222. <https://doi.org/10.4438/1988-592X-RE-2011-361-140>

30. Rodríguez-García, A.-M., Raso-Sánchez, F., & Ruiz-Palmero, J. (2019). Competencia digital, educación superior y formación del profesorado: un estudio de meta-análisis en la Web of Science. *Píxel-Bit. Revista de Medios y Educación*, 54, 65–82. <https://doi.org/10.12795/pixelbit.2019.i54.04>
31. Sandí-Delgado, J. C. (2019). Juegos serios para la indagación de competencias tecnológicas que puedan integrarse en la práctica pedagógica del profesorado. Una propuesta de aplicación en la Sede del Atlántico de la Universidad de Costa Rica (UCR). *TE&ET. Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología*, 23, 103–105. <https://doi.org/10.24215/18509959.23.e13>
32. Sandí-Delgado, J. C., & Bazán, P. A. (2017). *Cloud computing: posibilidades para la ejecución de juegos serios educativos as a service (JSEaaS)* [(Tesis de especialización). Universidad Nacional de La Plata (UNLP)]. <http://hdl.handle.net/10915/63388>
33. Sandí-Delgado, J. C., & Sanz, C. V. (2020). Juegos serios para potenciar la adquisición de competencias digitales en la formación del profesorado. *Revista Educación*, 44(1), 1–18. <https://doi.org/10.15517/revedu.v44i1.37228>
34. Sandí-Delgado, J. C., Sanz, C. V., & Lovos, E. N. (2022). Acceptance of Serious Games to Develop Digital Competencies in Higher Education. *The Electronic Journal of E-Learning (EJEL)*, 20(3), 351–367. <https://doi.org/10.34190/ejel.20.3.2181>
35. Sanz, C., Gorga, G., González, A., Zangara, A., Iglesias, L., Ibáñez, E., Violini, L., Fachal, A., Archuby, F., Abásolo, M. J., Manresa-Yee, C., Paula, D., & Pesado, P. (2022). Entornos virtuales y herramientas digitales en escenarios educativos híbridos. *XXIV Edición Del Workshop de Investigadores En Ciencias de La Computación*, 744–748. <http://sedici.unlp.edu.ar/handle/10915/145252>
36. Slivnik, B. (2022). Context-sensitive parsing for programming languages. *Journal of Computer Languages*, 73, 101172. <https://doi.org/10.1016/j.cola.2022.101172>
37. Sommerville, I. (2016). *Software Engineering* (M. Borthakur (ed.); 10th ed.). Pearson Education.
38. Spinelli, A. T., & Massa, S. M. (2018). Elicitación de Requerimientos, Centrada en el Usuario, para el Desarrollo de un Serious Game. *XIX Workshop de Investigadores En Ciencias de La Computación*, 1194–1198. <http://hdl.handle.net/10915/62882>
39. UCR, U. de C. R. (2004). *Consejo Universitario, Acta de la Sesión N.º4932* (pp. 1–30). Consejo Universitario, UCR.
40. Yeh, Y.-F., Hsu, Y.-S., Wu, H.-K., & Chien, S.-P. (2017). Exploring the structure of TPACK with videoembedded and discipline-focused assessments. *Computers & Education*, 104(January), 49–64. <https://doi.org/10.1016/j.compedu.2016.10.006>
41. Yulianto, B., Prabowo, H., Kosala, R., & Hapsara, M. (2017). Harmonik = ++(Web IDE). *Procedia Computer Science*, 116, 222–231. <https://doi.org/10.1016/j.procs.2017.10.044>
42. Yun, Y.-S., Kim, S., Park, J., Kim, H., Jung, J., & Eun, S. (2020). Development of Neuromorphic Architecture Integrated Development Environment. *International Conference on Green and Human Information Technology*, 47–49. <https://doi.org/10.1109/ICGHIT49656.2020.00019>