



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Pandalyze: aprender Ciencia de Datos con programación en bloques

**AUTORES:** Marcón Paula y Duggan Tomás

**DIRECTOR/A:** Banchoff Tzancoff, Claudia

**CODIRECTOR/A:** Martin, Sofía

**ASESOR/A PROFESIONAL:** -

**CARRERA:** Licenciatura en Sistemas

### Resumen

*En esta tesina analizamos cómo la Ciencia de Datos puede enseñarse en escuelas, utilizando herramientas diseñadas específicamente para este entorno. Abordamos los conceptos clave de esta disciplina, su relevancia en diferentes sectores y su impacto en el ámbito educativo. Además, evaluamos diversas herramientas que facilitan la enseñanza de la programación y el análisis de datos, destacando sus ventajas y desventajas. Finalmente, presentamos y evaluamos Pandalyze, una aplicación desarrollada para enseñar Ciencia de Datos a estudiantes de secundaria y primeros años de universidad mediante la programación en bloques.*

### Palabras Clave

*Ciencia de Datos, Python, bloques, programación, enseñanza, Pandalyze.*

### Conclusiones

*Exploramos el creciente impacto de la Ciencia de Datos en diversos sectores y su limitada integración en la educación inicial. En respuesta, desarrollamos Pandalyze, una aplicación intuitiva basada en programación en bloques, que, a través de las pruebas realizadas, ha demostrado ser efectiva para que los estudiantes sin experiencia previa se introduzcan en la Ciencia de Datos de manera autónoma.*

### Trabajos Realizados

*Realizamos las siguientes actividades:*

- *Elaboramos un análisis comparativo de las herramientas existentes utilizadas para enseñar programación y Ciencia de Datos.*
- *Desarrollamos la aplicación basada en bloques Pandalyze.*
- *Llevamos a cabo dos talleres, con grupos diferentes de estudiantes, para realizar pruebas utilizando Pandalyze y obtener una devolución.*

### Trabajos Futuros

*Extender la funcionalidad de la aplicación:*

- *Implementar perfiles de usuario para guardar el progreso y retomar el aprendizaje en cualquier momento.*
  - *Ampliar la aplicación con más ejemplos y ejercicios interactivos.*
  - *Añadir bloques adicionales, especialmente para pandas y Plotly, haciendo la herramienta más versátil.*
  - *Realizar la traducción de código a bloques para mayor flexibilidad.*
  - *Soportar otros formatos de archivos.*
- Elaborar secuencias didácticas para el uso de Pandalyze.*

**Fecha de la presentación: Septiembre 2024**



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## Agradecimientos

Queremos agradecer a todas las personas que hicieron posible esta tesina.

A nuestras familias, amigos y a quienes contribuyeron en nuestra formación, gracias por su apoyo incondicional y por la paciencia a lo largo de estos años de aprendizaje.

Gracias a la universidad pública por brindarnos las herramientas y recursos necesarios para desempeñarnos como profesionales.

A nuestras directoras, Claudia y Sofía, un agradecimiento especial por su orientación y constante motivación durante todo este proceso.



# Índice

<b>Capítulo 1. Presentación y estructura del proyecto.....</b>	<b>5</b>
1.1 Introducción.....	5
1.2 Objetivos.....	5
1.3 Motivación.....	6
1.4 Estructura del informe.....	7
<b>Capítulo 2. Ciencia de Datos.....</b>	<b>9</b>
2.1 ¿Qué es la Ciencia de Datos?.....	9
2.1.1 Las etapas de la Ciencia de Datos.....	10
2.2 Áreas de aplicación de la Ciencia de Datos.....	11
2.3 Desafíos y consideraciones éticas de la Ciencia de Datos.....	12
2.4 Futuro de la Ciencia de Datos.....	13
<b>Capítulo 3. Ciencia de Datos en las escuelas.....</b>	<b>15</b>
3.1 Informática en las escuelas.....	15
3.1.1 ¿Qué planes hay hoy en las escuelas?.....	16
3.2 Herramientas más utilizadas.....	17
3.2.1 Scratch.....	18
3.2.2 Pilas Bloques.....	19
3.3 ¿Por qué llevar la Ciencia de Datos a las escuelas?.....	20
3.4 Ciencia de Datos a nivel universitario.....	21
<b>Capítulo 4. Herramientas existentes.....</b>	<b>23</b>
4.1 Aspectos analizados de las herramientas.....	23
4.2 Herramientas analizadas.....	24
4.2.1 BlockPy.....	24
4.2.2 PandasGUI.....	26
4.2.3 Jupyter Notebook.....	27
4.3 Comparación de las herramientas analizadas.....	29
<b>Capítulo 5. Pandalyze.....</b>	<b>30</b>
5.1 Introducción.....	30
5.1.1 El por qué de la programación en bloques.....	30
5.2 Tecnologías utilizadas.....	31
5.2.1 Tecnologías del frontend.....	31
5.2.2 Tecnologías del backend.....	33
5.2.3 Limitaciones técnicas.....	35
5.2.4 Alternativas consideradas y decisiones finales.....	35
5.3 Aspectos destacados de la interfaz de usuario.....	36
5.4 Funcionalidad: flujo y experiencia de usuario.....	42
<b>Capítulo 6. Pruebas y validaciones.....</b>	<b>43</b>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

6.1 Evaluación de funcionalidad.....	43
6.1.1 Nuevas funcionalidades.....	43
6.1.2 Modificaciones para mejorar la Experiencia de Usuario.....	46
6.2 Evaluación con estudiantes.....	49
6.2.1 Evaluación con los estudiantes de la PPS.....	49
6.2.2 Evaluación en el marco del proyecto Ciencia de Datos en la escuela.....	51
6.3 Aspectos a destacar.....	53
6.3.1 Aspectos positivos.....	53
6.3.2 Aspectos negativos.....	54
<b>Capítulo 7. Conclusiones y consideraciones finales.....</b>	<b>55</b>
7.1 Conclusiones.....	55
7.2 Líneas de trabajo futuro.....	55
<b>Referencias Bibliográficas.....</b>	<b>57</b>



## Capítulo 1. Presentación y estructura del proyecto

En este primer capítulo, presentamos los elementos fundamentales que sustentaron el desarrollo de esta tesina. Detallamos los objetivos específicos de nuestro proyecto, la motivación que impulsó esta iniciativa y finalizamos exponiendo la estructura general del informe que describe brevemente los aspectos clave de cada capítulo.

### 1.1 Introducción

En esta tesina exploramos la relación entre la Ciencia de Datos y la enseñanza de los conceptos de esta ciencia, enfocándonos en cómo se puede introducir esta disciplina en las escuelas a través de herramientas didácticas y adecuadas para ese nivel educativo. A lo largo del documento, presentamos los conceptos fundamentales de la Ciencia de Datos, su importancia en diversos sectores y su relevancia en el contexto educativo. Analizamos las herramientas existentes que facilitan la enseñanza de la programación y el análisis de datos, evaluando las ventajas y limitaciones de cada una. El informe culmina con la descripción y evaluación de Pandalyze, una aplicación desarrollada para introducir los conceptos de la Ciencia de Datos a estudiantes de nivel secundario y primeros años de universidad, utilizando un enfoque de programación en bloques.

### 1.2 Objetivos

El objetivo de esta tesina fue desarrollar una herramienta visual que permita introducir, de forma simple, los aspectos básicos de la Ciencia de Datos y programación en Python a estudiantes de secundaria, promoviendo así el interés en esta temática.

Para lograr este objetivo, propusimos los siguientes objetivos secundarios:

1. Analizar estrategias pedagógicas y herramientas existentes para acercar la Ciencia de Datos a la escuela secundaria.
2. Diseñar y crear Pandalyze, una herramienta que permite a estudiantes introducir conceptos fundamentales de la Ciencia de Datos y programación en Python junto con la librería pandas, de una manera intuitiva y atractiva, utilizando el enfoque de programación basada en bloques.
3. Facilitar la comprensión de los conceptos y operaciones relacionadas con la Ciencia de Datos, como ser la manipulación de datos y visualización gráfica mediante la traducción automática de los bloques gráficos a código Python, lo que le permite a los estudiantes visualizar resultados de manera interactiva.
4. Evaluar la capacidad de los estudiantes para aplicar las técnicas básicas del análisis de datos utilizando Pandalyze.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

### 1.3 Motivación

La motivación detrás de esta tesina surgió de diversas consideraciones relevantes en el contexto educativo actual: la creciente importancia de la Ciencia de Datos, la necesidad de pensar en distintos enfoques para introducir la programación en la escuela y las limitaciones de las herramientas actuales.

Relevancia de la Ciencia de Datos: la Ciencia de Datos se ha convertido en un área de mucha importancia en la actualidad [1]. La capacidad de analizar y comprender datos es esencial en un amplia gama de disciplinas, desde la toma de decisiones empresariales hasta la investigación científica [2]. Sin embargo, este tema a menudo se considera avanzado y se aborda principalmente en niveles educativos superiores. La motivación nació por la necesidad de acercar esta disciplina a un público más joven y prepararlo para los desafíos de un mundo orientado a datos en constante evolución.

Además, algunas de las habilidades que fomenta son el trabajo en equipo, la visualización, manipulación, comprensión de datos y análisis significativo de los mismos [3].

Enseñanza de programación: la programación permite incorporar habilidades valiosas en el mundo digital actual, abre una puerta a la exploración, promueve la competencia digital, desarrolla el pensamiento lógico y matemático, facilita el trabajo autónomo y fomenta la creatividad [4], sin embargo, su enseñanza suele ser limitada y, en ocasiones, percibida como compleja.

Existen numerosas herramientas que permiten introducir los aspectos básicos de la programación a través de lenguajes basados en bloques, algunas de ellas son MIT App Inventor<sup>1</sup> en la cual la utilización de bloques permite a los usuarios crear aplicaciones móviles funcionales e interactivas y las ofrecidas por la iniciativa Code.org que ofrece actividades para enseñar programación y ciencias de la computación de manera accesible y entretenida<sup>2</sup>. Estas aplicaciones eliminan la complejidad inicial y brindan a los estudiantes una experiencia gradual y accesible en el aprendizaje de la programación.

Teniendo en cuenta esto, pensamos en una herramienta que utiliza programación basada en bloques para proporcionar una forma más amigable e intuitiva de introducir aspectos de programación orientados al análisis de datos.

Limitaciones de las herramientas actuales: las herramientas disponibles en la actualidad para manipular datos presentan limitaciones significativas al intentar introducir su uso en la educación secundaria. Por un lado, en la mayoría de los casos donde se aborda esta

---

<sup>1</sup> <https://appinventor.mit.edu/>

<sup>2</sup> <https://code.org/about>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

temática se utilizan herramientas basadas en el lenguaje Python, con las librerías Pandas<sup>3</sup> y Matplotlib<sup>4</sup>, a través de Jupyter Notebook<sup>5</sup> o Google Colaboratory<sup>6</sup>. Si bien estas son dos poderosas herramientas, obligan a comenzar con un lenguaje textual, como lo es Python.

Existen otras herramientas visuales, tales como PandasGUI<sup>7</sup> que, aunque poderosa, a menudo resulta complicada para principiantes debido a su excesiva funcionalidad y que requiere de una instalación ya que se trata de una aplicación de escritorio. Por otro lado, otra herramienta disponible es BlockPy<sup>8</sup> que, si bien es una aplicación web con una interfaz mucho más sencilla, tiene restricciones en la lectura de archivos CSV genéricos y está disponible únicamente en idioma inglés.

Dentro del panorama de herramientas en línea, también se destaca el proyecto Concord<sup>9</sup>, el cual ofrece diversas opciones para analizar conjuntos de datos. Esta plataforma brinda la posibilidad de trabajar con conjuntos de datos precargados, así como la posibilidad de analizar datos generados a raíz de partidas de un juego sencillo, con algunas limitaciones sobre su uso, por ejemplo, que no permite visualizar el código generado en base a las operaciones realizadas por el usuario.

Esta falta de herramientas adecuadas [5] y sencillas para estudiantes principiantes, fue un factor motivador para desarrollar una solución que combine la potencia de Python y la programación visual, dando origen a Pandalyze.

## 1.4 Estructura del informe

Este informe se encuentra estructurado en 7 capítulos donde desarrollamos los aspectos más importantes de la construcción de nuestra aplicación, Pandalyze. Desde nuestros objetivos iniciales, el estudio de la Ciencia de Datos, el análisis y comparación con otras herramientas existentes y sus usos en entornos educativos, hasta la puesta en marcha de la aplicación para todo público.

A continuación, y a modo de resumen, explicamos qué aborda cada capítulo.

En el **Capítulo 1**, describimos los objetivos y la motivación por la cual surgió la necesidad de crear Pandalyze, proporcionando un marco general y las razones que llevaron a su realización.

El **Capítulo 2** está dedicado a la Ciencia de Datos. Aquí explicamos qué comprende esta ciencia, sus etapas principales, las áreas de aplicación, los desafíos y consideraciones éticas asociadas y cómo será su impacto en el futuro.

---

<sup>3</sup> <https://pandas.pydata.org/>

<sup>4</sup> <https://matplotlib.org/>

<sup>5</sup> <https://jupyter.org/>

<sup>6</sup> <https://colab.google/>

<sup>7</sup> <https://github.com/adamerose/PandasGUI>

<sup>8</sup> <https://think.cs.vt.edu/blockpy/>

<sup>9</sup> <https://codap.concord.org/>





FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

En el **Capítulo 3**, luego de haber definido la importancia de la Ciencia de Datos, explicamos cuál es el papel que cumple en las escuelas y universidades actualmente. Antes de esto, hacemos un análisis de qué lugar ocupa la Informática en las escuelas. Luego, en aquellas que la han integrado en su plan de estudio, revisamos dos de las herramientas más utilizadas para enseñarla, Scratch y Pilas Bloques. Por último, llegando al final del capítulo, profundizamos en las razones por las cuales acercar la Ciencia de Datos a los distintos niveles educativos y nombramos algunas instituciones que han comenzado a introducir esta disciplina en sus programas.

Al avanzar hacia el **Capítulo 4**, nos enfocamos en algunas de las herramientas existentes y utilizadas para la enseñanza de la Ciencia de Datos. Realizamos un análisis de tres de ellas: BlockPy, PandasGUI y Jupyter Notebook, investigando cómo funcionan, evaluando sus ventajas y desventajas, considerando su aplicabilidad en contextos educativos, y explorando las características que mejor se ajustan a las necesidades de los usuarios. Finalizamos con un cuadro comparativo que resume el análisis realizado.

En el transcurso del **Capítulo 5**, nos adentramos en Pandalyze. Examinamos las tecnologías empleadas en su creación, detallamos su estructura y explicamos las decisiones clave que nos guiaron en el proceso de desarrollo hasta el objetivo final. Posteriormente, presentamos el diseño de la interfaz de usuario de nuestra aplicación y la dividimos en secciones describiendo qué rol cumple cada una de ellas.

Una vez finalizada la construcción de Pandalyze, realizamos algunas pruebas con estudiantes para probar su correcto funcionamiento y obtener devoluciones al respecto. En el **Capítulo 6**, determinamos cuáles fueron estas pruebas realizadas, describimos los destinatarios que las realizaron y exploramos los resultados obtenidos.

Para finalizar, en el **Capítulo 7**, presentamos nuestras conclusiones obtenidas a lo largo de esta tesina, junto con posibles líneas de trabajo futuro a tener en cuenta en caso de querer expandir la aplicación.





## Capítulo 2. Ciencia de Datos

A lo largo de este capítulo nos introducimos en el mundo de la Ciencia de Datos. Comenzamos definiendo esta disciplina, atravesando sus principales etapas y áreas de aplicación. Luego, exploramos y analizamos los desafíos y consideraciones éticas que enfrenta el uso de la Ciencia de Datos y finalizamos revisando los desarrollos futuros que podrían influir en su evolución.

### 2.1 ¿Qué es la Ciencia de Datos?

Vivimos en un mundo lleno de datos. Con la llegada de Internet, la acumulación de información ha crecido de manera exponencial, impulsada tanto por las interacciones directas e indirectas de los usuarios como por la proliferación de dispositivos que conforman la Internet de las Cosas (IoT)<sup>10</sup>, que generan y transmiten datos de manera autónoma y constante. Los sitios web rastrean cada movimiento del mouse, los teléfonos inteligentes registran nuestra ubicación y nuestras preferencias de aplicaciones, las tiendas utilizan cámaras y sensores para monitorear el comportamiento y patrones de compra de los clientes, las redes sociales almacenan cada publicación, comentario, y los servicios de transmisión rastrean nuestros hábitos de visualización. Entonces, Internet es una inmensa fuente de datos con respuestas a incontables preguntas, ¿cómo podemos extraer las respuestas? [6].

La Ciencia de Datos es un campo interdisciplinario que utiliza métodos, procesos y algoritmos para extraer conocimiento y obtener información valiosa a partir de datos crudos. Combina diversas áreas como la Estadística, la Informática y el conocimiento de dominio específico para analizar grandes cantidades de datos y descubrir patrones, tendencias y asociaciones que pueden ser útiles para la toma de decisiones.

Como consecuencia del acelerado crecimiento del volumen de datos, la Ciencia de Datos se ha convertido en un área indispensable en sectores como el comercial, la salud, el gubernamental, el educativo, entre otros [7]. Esto ha generado una demanda significativa de profesionales capacitados en análisis de datos, quienes extraen información y conocimientos de conjuntos de datos complejos y desordenados. Además, utilizan habilidades en programación, estadísticas y conocimientos específicos del dominio para limpiar, procesar y analizar datos, desarrollando modelos predictivos y algoritmos para interpretar y comunicar hallazgos de manera efectiva.

---

<sup>10</sup> **IoT (Internet of Things o Internet de las Cosas)**: red de dispositivos y objetos físicos conectados a Internet que pueden recolectar, intercambiar y actuar sobre datos de manera autónoma, facilitando la automatización y el análisis en tiempo real.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

### 2.1.1 Las etapas de la Ciencia de Datos

La Ciencia de Datos ha emergido como un campo fundamental en la era digital, transformando la manera en que las organizaciones abordan la toma de decisiones. Su relevancia surge por su capacidad para convertir grandes volúmenes de datos en información valiosa. Por esto, es esencial comprender los pasos clave en el análisis de datos y cómo cada uno de ellos forma un ciclo continuo de trabajo. En este sentido, el proceso de análisis de datos se puede resumir en las siguientes etapas [7] [8]:

- 1) Definir objetivo: esta primera etapa implica determinar qué datos se van a recopilar y por qué. Este proceso se centra en definir qué información es vital para abordar problemas específicos, estableciendo una conexión directa entre los datos a seleccionar y los objetivos estipulados.
- 2) Adquirir los datos: en esta etapa, se utilizan herramientas y algoritmos avanzados para analizar grandes volúmenes de datos y extraer información útil. Es esencial reconocer que no todos los datos tienen igual valor y, por lo tanto, es fundamental recopilar y organizar únicamente aquellos que contribuyen al logro de las metas definidas en el primer paso.
- 3) Comprender los datos: en este tercer paso, se procesan y organizan los datos recopilados, utilizando herramientas y técnicas específicas, para luego extraer conclusiones significativas.
- 4) Comunicar los resultados: finalmente, en esta última etapa, se deben presentar los resultados y conclusiones obtenidas en el paso anterior de manera clara y efectiva mediante gráficos o informes y que estos puedan ser utilizados y comprendidos por los involucrados para tomar decisiones al respecto.

En la **Figura 1** exhibimos un gráfico con los pasos mencionados previamente.



Figura 1. Pasos en el análisis de datos

## 2.2 Áreas de aplicación de la Ciencia de Datos

La Ciencia de Datos puede ser aplicada en diversos tipos de áreas, entre ellas ciberseguridad, finanzas, seguros, industria, comercial, salud, gubernamental y educativa. Detallamos algunos ejemplos de su aplicación en estos campos a continuación [7] [9]:

- Ciberseguridad - identificación de ciberamenazas: en ciberseguridad, la identificación de amenazas se basa en el análisis de datos de acceso a sistemas y redes. Se buscan patrones de actividad normal y se emiten alertas ante desviaciones. Los datos provienen de registros de actividad, los cuales son analizados para detectar comportamientos anómalos y prevenir posibles ataques.
- Finanzas - detección de fraudes: en el sector financiero, la detección de fraudes, como los relacionados con tarjetas de crédito, se realiza mediante el cruce de datos de diversas fuentes. Al analizar patrones de comportamiento normal del cliente y compararlos con actividades sospechosas, los sistemas pueden identificar y alertar sobre posibles fraudes antes de que se concreten, protegiendo así a los usuarios y a las entidades financieras.
- Seguros - cálculo de primas: las aseguradoras utilizan la Ciencia de Datos para calcular las primas personalizadas. Analizando datos de sensores en vehículos, pueden evaluar los hábitos de conducción y determinar el riesgo de accidentes. Esto permite ofrecer tarifas ajustadas a cada cliente, considerando incluso variaciones estacionales en su comportamiento.



- Industria - mantenimiento predictivo: en el ámbito industrial, los sensores en maquinaria y sistemas logísticos recogen datos sobre múltiples variables operativas. Estos datos se analizan para predecir fallos y optimizar el mantenimiento, evitando costosas paradas no planificadas y reduciendo la necesidad de reparaciones inesperadas.
- Comercial - preferencias de consumidores: en el ámbito comercial, la Ciencia de Datos permite analizar las preferencias y comportamientos de los consumidores. Esta información ayuda a personalizar las estrategias de marketing, optimizar la oferta de productos y mejorar la experiencia del cliente, ajustando las campañas y promociones para satisfacer mejor las necesidades y deseos del público objetivo.
- Salud - diagnóstico temprano: en el área de la salud, la Ciencia de Datos facilita la identificación de patrones en datos médicos que apoyan el diagnóstico temprano de enfermedades. Esto permite a los profesionales de la salud detectar condiciones antes de que se conviertan en problemas graves, mejorando la precisión del diagnóstico y posibilitando una atención médica más eficiente y personalizada para los pacientes.
- Gubernamental - calidad de vida: en este campo, la Ciencia de Datos ayuda a desarrollar y evaluar políticas públicas que buscan mejorar la calidad de vida de los ciudadanos. Analizando datos sobre diversos aspectos de la vida urbana, salud, educación y seguridad, los gobiernos pueden diseñar intervenciones más efectivas y basadas en evidencia, optimizando recursos y abordando las necesidades y prioridades de la comunidad de manera más precisa.
- Educativo - rendimiento estudiantil: en el campo educativo, se puede analizar el rendimiento de los estudiantes, identificando áreas que requieran atención. Esta información permite desarrollar estrategias de enseñanza más efectivas y adaptadas a las necesidades individuales. Además, facilita la implementación de intervenciones tempranas y personalizadas para apoyar a los estudiantes a optimizar su desempeño académico.

En el capítulo siguiente, hacemos foco en la aplicación en el campo educativo, el cual es el objetivo de este trabajo.

## 2.3 Desafíos y consideraciones éticas de la Ciencia de Datos

La Ciencia de Datos enfrenta una serie de desafíos técnicos y éticos que deben ser considerados. Estos no solo influyen en la calidad de los resultados obtenidos, sino también en aspectos como la equidad, la privacidad y la confianza en los sistemas. A continuación, presentamos algunos desafíos relevantes en este campo [10] [11]:

- Calidad y sesgo de los datos: uno de los principales desafíos en la Ciencia de Datos es garantizar la calidad y la integridad de los datos utilizados para el análisis. Datos incompletos, inexactos o desactualizados pueden llevar a conclusiones erróneas y



decisiones equivocadas. Por ejemplo, si un conjunto de datos contiene sesgos históricos contra ciertos grupos demográficos, un modelo de Inteligencia Artificial entrenado con estos datos puede replicar estas desigualdades, perjudicando a estos grupos. Por lo tanto, es esencial implementar técnicas de limpieza de datos y utilizar métodos estadísticos para detectar y corregir sesgos, asegurando que los modelos sean justos y representativos.

- Privacidad y seguridad de los datos: la privacidad y seguridad de los datos son aspectos fundamentales en la ética de la Ciencia de Datos. Con el aumento de la cantidad de datos recolectados y almacenados por las organizaciones, proteger la información sensible y personal se ha vuelto más crítico que nunca. Por lo tanto, las organizaciones deben invertir en robustas medidas de seguridad, como la encriptación y el acceso controlado, así como en la capacitación continua de sus empleados sobre las mejores prácticas en privacidad y seguridad de datos.
- Transparencia y explicabilidad en los algoritmos de recopilación de datos: en un mundo cada vez más dependiente de los algoritmos, es importante comprender cómo funcionan y cómo toman decisiones. La transparencia implica que tanto las organizaciones como los científicos de datos ofrezcan información detallada y accesible sobre la recolección, el procesamiento y el uso de los datos. La explicabilidad hace referencia a la capacidad para interpretar y justificar las decisiones generadas por estos algoritmos. Contando con estos dos aspectos, se asegura un uso ético y responsable de los datos y su recolección.
- Datos abiertos: son datos que pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona. Es por esto que es esencial que los datos sean limpiados para eliminar información inexacta, estandarizados para garantizar coherencia en su formato, y que sean anónimos para proteger la privacidad de las personas. La Ciencia de Datos depende de la calidad de los datos, por lo que estos procesos son muy importantes antes de publicar o utilizar datos abiertos para evitar sesgos y respetar el propósito original de su recolección [12].

## 2.4 Futuro de la Ciencia de Datos

El futuro de la Ciencia de Datos se ve prometedor y expansivo, impulsado por avances tecnológicos continuos y la creciente disponibilidad de datos. Aquí, presentamos varias tendencias y desarrollos clave que probablemente moldearán el futuro de esta disciplina [13]:

- Integración con IA y ML: a medida que la Inteligencia Artificial y el Machine Learning<sup>11</sup> evolucionen, su integración con la Ciencia de Datos se volverá más profunda. Esto permitirá análisis más sofisticados y capacidades predictivas,

---

<sup>11</sup> Machine learning: campo de la Inteligencia Artificial que permite a los sistemas aprender de datos y tomar decisiones sin intervención humana [14].



automatizando procesos complejos y haciendo predicciones más precisas a gran escala.

- Avances en Deep Learning: el Deep Learning<sup>12</sup> seguirá revolucionando las capacidades de la Ciencia de Datos, particularmente en campos como el reconocimiento de imágenes y voz, el procesamiento del lenguaje natural y la detección de anomalías. Esto mejorará la automatización del reconocimiento de patrones y los procesos de toma de decisiones.
- Computación cuántica: la aparición de la computación cuántica promete ofrecer avances significativos en la potencia de procesamiento, lo que podría revolucionar la forma en que se procesan y analizan los grandes volúmenes de datos. Esto podría resolver problemas complejos mucho más rápido que los métodos de computación actuales.
- IA ética y responsable: habrá un enfoque creciente en las consideraciones éticas y el uso responsable de la IA y la Ciencia de Datos. Esto incluye preocupaciones sobre la privacidad, seguridad, equidad y transparencia. Las organizaciones deberán adoptar directrices y prácticas éticas para garantizar que sus iniciativas no causen daño o sesgo de manera inadvertida.
- Análisis automatizado y aumentado: se espera un crecimiento en la automatización en la Ciencia de Datos, a través de tecnologías como AutoML<sup>13</sup>. Esta herramienta puede analizar datos automáticamente y generar descubrimientos sin intervención humana, haciendo la Ciencia de Datos más accesible para no expertos y aumentando la productividad.

---

<sup>12</sup> Deep learning: subconjunto de ML que utiliza redes neuronales profundas con múltiples capas para modelar y extraer patrones complejos a partir de grandes volúmenes de datos [14].

<sup>13</sup> AutoML: la Automatización del Aprendizaje Automático (AutoML) proporciona métodos y procesos para hacer que el Machine learning sea accesible para personas que no son expertas en este campo [15].





## Capítulo 3. Ciencia de Datos en las escuelas

En este capítulo, continuamos explorando la Ciencia de Datos, pero ahora explorando su relevancia en el ámbito educativo. Empezamos revisando el estado actual de la enseñanza de la Informática en las escuelas y estudiando las herramientas más utilizadas para esto. Llegando al final, discutimos sobre la importancia de incorporar la Ciencia de Datos en las escuelas y repasamos universidades que han comenzado a incluir el estudio de esta temática.

### 3.1 Informática en las escuelas

La incorporación de contenidos de la Informática en la educación de nivel obligatorio ha avanzado en los últimos años a través de diferentes experiencias y propuestas.

Hace diez años, en el año 2014, la Fundación Sadosky lanzó una campaña en donde personas destacadas en distintos ámbitos, como Enamuel Ginobili o Mario Pergolini, explican por qué es importante aprender a programar. El video está disponible en YouTube<sup>14</sup>. Esto acompañó numerosas iniciativas para promover la enseñanza de programación en niños y jóvenes. Algunos ejemplos de esto pueden ser [16]:

- *“Clases virtuales de programación: para escuelas primarias y secundarias, con un equipo integrado por docentes formados en nuestros cursos o especializaciones.”*
- *“Sitio web Estudiar Computación: para acompañar la búsqueda de carreras y promover el interés por ser parte del proceso creativo que define la tecnología que usamos.”*
- *“Talleres en escuelas: sobre Programación, Inteligencia Artificial, Videojuegos y Seguridad Informática, para despertar vocaciones y promover la participación de mujeres y diversidades en las carreras relacionadas con las Ciencias de la Computación.”*

Por otra parte, el Consejo Federal de Educación (CFE) de la Argentina declaró el aprendizaje de la programación como una herramienta de "importancia estratégica para el sistema educativo argentino"[17]. Otras iniciativas que potenciaron la implementación de la Informática en las escuelas fueron los planes gubernamentales como:

- *“El Programa Conectar Igualdad creó “HUAYRA”, el sistema operativo libre, público, gratuito y abierto, el cual ha sido incorporado en todos los equipos distribuidos y a distribuir a docentes y alumnos, y que provee múltiples herramientas y facilidades para la enseñanza y el aprendizaje de la programación.”*
- *“El portal Educ.ar acompañó la implementación del Programa Conectar Igualdad con diversas actividades y contenidos dirigidos a estudiantes y docentes vinculados a la programación aplicada en diferentes áreas, generando la participación de escuelas*

<sup>14</sup> [https://www.youtube.com/watch?v=HrBh2165KjE&ab\\_channel=programar2020](https://www.youtube.com/watch?v=HrBh2165KjE&ab_channel=programar2020)





FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

*artísticas, como orientadas y técnicas.”*

### 3.1.1 ¿Qué planes hay hoy en las escuelas?

Según la ADICRA (Asociación de Docentes de Informática y Computación de la República Argentina) [18], un espacio curricular debe contar con un diseño específico que incluya objetivos claros y contenidos definidos, además de tener asignada una materia propia de Informática y otorgar prioridad a los profesores especializados. Estos criterios son determinados por las jurisdicciones provinciales, afectando tanto a escuelas estatales como privadas.

Sin embargo, la realidad muestra que la presencia de la Informática en las escuelas es irregular. En el nivel primario, la materia Informática no figura en los planes de estudio de ninguna provincia y en el nivel secundario, aunque algunas han introducido cursos bajo la denominación de TIC (Tecnologías de la Información y la Comunicación)<sup>15</sup>, estos no siempre constituyen una enseñanza completa de la Informática como disciplina autónoma. Además, hay escuelas que han incorporado la materia Informática pero de manera extracurricular o extraprogramática. La oferta de títulos como Bachiller en Informática o Técnico en Computación es limitada y no llega a todos los estudiantes de manera uniforme. La **Tabla 1** la obtuvimos del sitio oficial de ADICRA y muestra la distribución de la carga horaria dedicada a las materias de Informática en las escuelas argentinas de nivel bachiller, desglosada por provincia.

---

<sup>15</sup> En la provincia de Buenos Aires, la materia se denomina NTICX.



Provincia (Jurisdicción)	Nombre de la materia	Año	Horas cátedra
Buenos Aires (Provincia)	NTICx	4to.	3 hc
Catamarca	Nuevas TIC	4to.	4 hc
Ciudad de Buenos Aires	TI - Tecnologías de la Información	3ro., 4to., 5to.	2 hc
Corrientes	TIC - Tecnologías de la Información y la Comunicación	4to.	4 hc
Entre Ríos	TIC - Tecnologías de la Información y la Comunicación	5to.	2 hc
La Pampa	TIC - Tecnologías de la Información y la Comunicación	4to.	4 hc
Mendoza	TIC - Tecnologías de la Información y la Comunicación	3ro.	3 hc
Misiones	TIC - Tecnologías de la Información y la Comunicación	3ro.	3 hc
Neuquén	Informática (interareal con Matemática).	1ro., 2do, 3ro.	2 hc
Santa Cruz	Informática	4to. o 5to.	3 hc
Santiago del Estero	TIC - Tecnologías de la Información y la Comunicación	5to.	3 hc
Tucumán	TIC - Tecnologías de la Información y la Comunicación	4to.	3 hc

**Tabla 1. Enseñanza de Informática en nivel bachiller**

La Informática está siendo gradualmente introducida en las escuelas, pero predominantemente se enseña los conceptos básicos en lugar de abordar temas más avanzados y orientados específicamente a la programación de computadoras.

### 3.2 Herramientas más utilizadas

La programación de computadoras es uno de los temas que se abordan a la hora de introducir Informática y justamente el que puede ser un vínculo para trabajar aspectos de Ciencia de Datos.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

Existen diversas herramientas que se utilizan en escuelas actualmente con el fin de introducir la programación, entre las más populares se encuentran Scratch<sup>16</sup> y Pilas Bloques<sup>17</sup>.

### 3.2.1 Scratch

Scratch es un lenguaje de programación con una interfaz sencilla que permite a los jóvenes crear historias digitales, juegos y animaciones y compartirlas con el mundo. Cuando estos crean y comparten proyectos de Scratch, aprenden a pensar de forma creativa, a razonar sistemáticamente y a trabajar colaborativamente.

Algunas características de la aplicación:

- Scratch está diseñado, desarrollado y moderado por la Fundación Scratch, una organización sin fines de lucro.
- Está diseñado especialmente para edades entre los 8 y 16 años, aunque usado por personas de todas las edades.
- Está disponible en más de 70 idiomas.
- Se puede usar a través de un navegador web o mediante una aplicación de escritorio descargable para los sistemas operativos Windows, Mac OS, Chrome OS y Android.
- Se encuentra bajo la licencia “*Creative Commons Attribution-ShareAlike 2.0*”: cualquiera puede adaptar y modificar los proyectos publicados por otros sin ninguna penalización, siempre y cuando den crédito al autor original y licencien sus contribuciones bajo la misma licencia.
- Cuenta con una extensa comunidad. En 2022, alcanzó los 50 millones de usuarios, 120 millones de proyectos nuevos y 8 millones de descargas [19]. Entre los proyectos más populares se encuentran videojuegos originales como *Stickman Hook* y *Tower Climb*, así como clones de juegos populares como *Minecraft* y *Subway Surfer* para dispositivos móviles. Esta información se encuentra disponible en la sección “Explorar” del sitio oficial de Scratch.

Seguidamente, mostramos algunas experiencias de docentes utilizando la aplicación en Argentina:

- Alejandra Valdez es Ingeniera en Sistemas de Información y Especialista Docente de Nivel Superior en Educación y TIC en San Juan, en donde, en 2017, tuvo a cargo la materia Informática I. Su opinión nos parece importante porque resume la herramienta y describe su propósito. Remarca la importancia de que los estudiantes conozcan de qué trata la programación y se familiaricen con los conceptos básicos para adquirir pensamiento lógico y la resolución de problemas. También consideramos acertado el hecho de que no es importante que se vuelvan

---

<sup>16</sup> <https://scratch.mit.edu/>

<sup>17</sup> <https://pilasbloques.program.ar/>



programadores ya que además la herramienta no fue diseñada para tal fin. Alejandra comenta:

*“Fue una experiencia muy rica, porque si bien no pretendo que salgan programadores, sí me propuse que los chicos conozcan de qué se trata la programación. Es muy importante que ellos mismos atraviesen estas experiencias, de nada sirve que el alumno aprenda conceptos sin desarrollar una serie de habilidades y competencias digitales como las que permite la programación” [20].*

- En 2017, en el Club de Ciencia Albert Einstein de la Escuela Primaria Provincial N.º82 de Santa Cruz, los docentes organizaron un taller con Scratch para que los chicos den sus primeros pasos en el mundo de la programación y puedan desarrollar competencias relacionadas al pensamiento computacional. Compartimos el punto de vista de la profesora de Informática Silvana Salica, ya que destaca acertadamente la efectividad de Scratch para enseñar programación a niños pequeños mostrando que es una herramienta valiosa y útil en edades tempranas:

*“Participaron alumnos de 8 y 9 años, y lo que me sorprendió fue que, siendo tan pequeños, puedan responder tan bien a la explicación e incluso plantear sus propias soluciones frente a determinados problemas, eso fue muy motivador para todos” [21].*

### 3.2.2 Pilas Bloques

Nos parece que la siguiente definición de la página oficial de Pilas Bloques refleja de manera precisa y efectiva la esencia de la herramienta, destacando su enfoque en la enseñanza y el aprendizaje de la programación a través de una metodología accesible:

*“Pilas Bloques es una aplicación para enseñar y aprender a programar por medio de bloques de forma simple y divertida. Posee desafíos con diversos niveles de dificultad para que niñas, niños y adolescentes puedan conocer y dar los primeros pasos en el mundo de la programación.”*

Algunas características de la aplicación:

- Está desarrollada en Argentina, desde la iniciativa Program.AR de la Fundación Sadosky.
- La herramienta está orientada a quienes tengan entre 5 y 12 años.
- Está disponible en español, inglés y portugués.
- Es libre, gratuita y puede usarse a través de un navegador web o descargarse para los sistemas operativos Linux/Huayra, Linux portable, Windows y Mac OS X y luego usarse sin necesidad de internet.
- Los desafíos de Pilas Bloques están organizados en secuencias de complejidad ascendente (principiante, intermedio y avanzado) e invitan a los estudiantes a resolver un problema concreto. Les brindan tiempo de exploración para que luego, junto a sus docentes, puedan revisar las diferentes estrategias de solución y construir las ideas fundamentales.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

A continuación, compartimos el ángulo de Carlos Gregorio Artilles Fontales, profesor de Tecnología en el I.E.S. Carmen Martín Gaité de Morzarzal (Madrid), quien en 2019 utilizó la aplicación y con quien coincidimos en el hecho de que el estudiante tiene la sensación de estar jugando al utilizarla. Creemos que esto es parte de lo que ayuda a mantener el interés y la motivación:

*“El entorno gráfico de Pilas Bloques puede recordar, en una primera impresión, que se está ante una aplicación informática de Infantil-Primaria, y que quizás, pueda quedarse corta en cuanto al alcance y desempeño esperado en el nivel de Secundaria; pero una vez que los estudiantes acceden a las dinámicas planteadas, como docente, te das cuenta, de que se trata de una pequeña joya muy cuidada para que el chaval mantenga el interés y la atención en lo que está realizando a la vez que tiene la sensación de estar “jugando” [22].*

### 3.3 ¿Por qué llevar la Ciencia de Datos a las escuelas?

Como mencionamos en secciones previas, la Informática está siendo gradualmente incorporada en el ámbito educativo a través de diversas estrategias. Sin embargo, la enseñanza se centra principalmente en los conceptos básicos, como programación y ciberseguridad, dejando de lado otros temas más avanzados.

Mencionamos también que la Ciencia de Datos se ha convertido en una disciplina que utiliza la Informática, más específicamente la programación, para transformar grandes cantidades de datos en información valiosa para la toma de decisiones, es por esto que las nuevas generaciones deben adquirir habilidades para manejar esta disciplina de manera efectiva. Sin embargo, a pesar de su relevancia, aún no ha sido trabajada en la escuela secundaria, menos aún integrada en la currícula. Motivar a los jóvenes a explorar esta disciplina es esencial, pero enfrentamos el desafío de que, tanto para enseñar como para aprender Ciencia de Datos, se requieren herramientas especializadas y un enfoque pedagógico adaptado a su complejidad.

En [23] se sostiene que:

*“Los establecimientos educativos han ido incorporando contenidos de Ciencia de la Computación en forma muy dispar y su avance está relacionado con la implementación de planes gubernamentales, nacionales y provinciales, que incluyen la entrega de diferentes tipos de recursos tecnológicos acompañados de propuestas pedagógicas. En particular la Ciencia de Datos no es aún tenida en cuenta como contenido a trabajar dentro de la escuela, si bien cada día se pone en evidencia su creciente importancia a través de generación de carreras, especializaciones universitarias, es escasa su implementación en nivel secundario... La introducción de este tipo de análisis en la escuela media permite generar propuestas para fomentar el pensamiento crítico de los estudiantes y entregar herramientas que permitan una mejor toma de decisiones a los equipos docentes para el seguimiento de sus estudiantes”.*



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

Por lo dicho hasta el momento, es importante contar con herramientas que faciliten la enseñanza de esta disciplina. Estas deberían ser interactivas e intuitivas, que simplifiquen conceptos complejos de programación y estadística y que permitan a los estudiantes explorar y experimentar con conjuntos de datos reales de manera guiada. Esto no solo haría más accesible la Ciencia de Datos, sino que también fomentaría el interés y la participación activa de los estudiantes en su aprendizaje.

### 3.4 Ciencia de Datos a nivel universitario

A nivel universitario, la oferta educativa en Informática y Ciencia de Datos es significativamente más amplia y especializada en comparación con los niveles educativos anteriores. Las universidades ofrecen una variedad de programas y cursos que permiten a los estudiantes profundizar en temas avanzados de Informática y análisis de datos.

Coincidimos con el director de la Licenciatura en Ciencias de la Computación de Exactas-UBA Santiago Ceria, quien asegura que:

*“Un profesional en Ciencias de la Computación tiene muchísimas oportunidades para insertarse en el mercado laboral porque hay muchísima demanda de profesionales y la oferta está bastante restringida. Esa demanda no es sólo local, sino global, porque hay mucha gente que trabaja de manera remota desde Argentina para otros países del mundo” [24].*

Creemos que es razonable pensar que una de las principales razones de esta situación es la insuficiente exposición de la Informática y la Ciencia de Datos (como una rama emergente) en las escuelas secundarias. Además, es posible que la baja cantidad de inscriptos se deba también a que muchos estudiantes perciben que pueden acceder a oportunidades laborales en el campo de la Informática con cursos y certificaciones, sin necesidad de un título de grado formal. Relacionado a esto último, nos parece acertado lo que el director de la Licenciatura argumenta:

*“Es muy conveniente terminar la carrera... El principal argumento es que la carrera está pensada como un conjunto de conocimientos que se integran unos con otros. Quienes diseñan los planes de estudio piensan en los conocimientos y habilidades que son deseables para que un profesional de Ciencias de la Computación los desarrolle, si uno no termina la carrera deja de aprender ciertas cosas que quienes diseñan las carreras consideran útiles. Uno podría decir que esas cosas las puede aprender en otro lado o en la práctica. Si bien en algunos casos eso puede ser cierto, creemos que el lugar ideal para recibir todo eso es el ambiente de la universidad” [24].*





FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

No obstante, en los últimos años se ha observado un crecimiento notable en el interés por estudiar Informática y Ciencia de Datos<sup>18</sup>. Este incremento se debe en parte a la creciente demanda del mercado laboral de profesionales capacitados en estas áreas, así como al reconocimiento de la importancia de los datos y la tecnología en prácticamente todos los sectores de la economía. Los avances tecnológicos y la transformación digital han resaltado la necesidad de expertos que puedan manejar y analizar grandes volúmenes de datos, impulsando así el interés por estas carreras.

En la Universidad Nacional de La Plata (UNLP), se ha respondido a esta demanda creando la carrera de Ciencia de Datos en Organizaciones, que comenzó a dictarse en el año 2024. Esta carrera es ofrecida de manera conjunta por la Facultad de Informática y la Facultad de Ciencias Económicas, y está diseñada para formar expertos en el manejo y análisis de datos. El programa tiene como objetivo principal capacitar a los estudiantes en la combinación de habilidades de gestión y técnicas informáticas a gran escala, aplicables en diversos ámbitos de la economía, incluyendo empresas, gobiernos, organizaciones públicas, de la sociedad civil y emprendimientos digitales.

Queda claro que a nivel universitario el abordaje de esta disciplina es específico y completamente enfocado en formar profesionales en el área. Muy distinto es el enfoque que se quiere dar en la escuela secundaria, donde se pretende realizar un acercamiento a la disciplina de manera tal de trabajar las habilidades relacionadas y proponer otro uso de la programación.

En el próximo capítulo, analizamos cuáles son las herramientas existentes para enseñar Ciencia de Datos en el nivel medio y cuáles son sus usos, ventajas y desventajas en relación a Pandalyze.

---

<sup>18</sup> La cantidad de inscriptos en la Licenciatura en Ciencias de la Computación en la UBA creció un 88% entre 2020 y 2022. Junto con la Licenciatura en Ciencias de Datos, es de las carreras más elegidas en la Facultad de Ciencias Exactas y Naturales [25].





## Capítulo 4. Herramientas existentes

En este capítulo describimos algunas herramientas analizadas que permiten el trabajo con datos. Definimos los aspectos analizados en cada una y concluimos con una tabla comparativa entre las mismas, destacando sus ventajas y desventajas de acuerdo a las propuestas de uso pensadas para la escuela secundaria.

### 4.1 Aspectos analizados de las herramientas

En el **capítulo 1** mencionamos algunas herramientas como BlockPy, PandasGUI, Jupyter Notebook, App Inventor y CODAP. En esta sección realizamos un análisis comparativo de tres de ellas: BlockPy, PandasGUI y Jupyter Notebook. Seleccionamos estas aplicaciones por usar Python y por su relevancia en la enseñanza de la programación y el análisis de datos. Nuestro objetivo fue evaluar sus características, fortalezas y limitaciones en contextos educativos, de manera tal de analizar su uso o el desarrollo de una nueva herramienta.

Entre los aspectos que analizamos incluimos:

- Facilidad de instalación y uso: evaluamos tanto la facilidad con la que las herramientas pueden ser instaladas y puestas en funcionamiento, ya sea en servidores locales o en línea, como su usabilidad. Consideramos si utilizan lenguajes textuales o interfaces basadas en bloques, un factor clave para la iniciación de estudiantes con poca o ninguna experiencia en entornos digitales, análisis de datos y programación.
- Conocimientos previos de Python: consideramos si las herramientas requieren tener conocimientos previos del lenguaje de programación Python.
- Soporte para pandas y Plotly: evaluamos la capacidad que tienen para trabajar con las librerías especializadas pandas y Plotly<sup>19</sup>, fundamentales para el análisis de datos y la creación de gráficos, y cómo estas herramientas facilitan la manipulación de datos complejos.
- Visualización de gráficos interactiva: analizamos la capacidad de cada herramienta para generar y manipular gráficos interactivos, permitiendo a los usuarios explorar los datos de manera dinámica.
- Soporte para archivos CSV personalizados: analizamos la facilidad con la que cada aplicación permite a los usuarios trabajar con datasets propios, lo cual es importante para una experiencia de aprendizaje más personalizada y realista.
- Interfaz en español: evaluamos la disponibilidad de las herramientas en español, considerando como un factor que puede limitar el acceso y la efectividad del aprendizaje.

---

<sup>19</sup> <https://plotly.com/>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

- Licencias y código abierto: examinamos las licencias bajo las cuales se distribuyen y si son de código abierto, lo que afecta la capacidad de personalización, flexibilidad de uso, y posibilidad de adaptación a diferentes contextos educativos.

## 4.2 Herramientas analizadas

En esta sección profundizamos sobre las herramientas seleccionadas, BlockPy, PandasGUI y Jupyter Notebook, describiendo los aspectos principales de cada una.

### 4.2.1 BlockPy

BlockPy es una aplicación educativa diseñada para facilitar la enseñanza de programación y análisis de datos a estudiantes, especialmente en contextos educativos. Fue desarrollada por Austin Cory Bart, Luke Gusukuma, Javier Tibau, Dennis Kafura, Clifford A. Shaffer y Eli Tilevich en 2015. Esta aplicación web permite a los usuarios aprender conceptos fundamentales de programación utilizando una interfaz de bloques visuales, lo que la hace accesible incluso para aquellos con poca o ninguna experiencia en programación.

BlockPy se puede utilizar desde un servidor provisto por la comunidad de desarrolladores, o bien, instalarse en un servidor local, adaptándose a las necesidades de cada institución educativa. Al ser una herramienta de código abierto, no solo es accesible para un amplio público, sino que también ofrece la flexibilidad de ser modificada y personalizada por educadores y desarrolladores para mejorar y adaptar su funcionalidad según el contexto educativo específico. Esta combinación de accesibilidad, flexibilidad y enfoque educativo hace de BlockPy una herramienta poderosa para introducir a los estudiantes en el mundo de la programación y el análisis de datos.

En la **Figura 2** presentamos su interfaz de usuario, junto con un ejemplo elaborado, donde se muestra un gráfico de barras con la cantidad de estudiantes varones que fueron inscriptos en la Facultad de Informática de la UNLP durante los años 2018, 2019 y 2020. Dado que BlockPy solo permite trabajar con un conjunto de datos precargados que se encuentran en la sección “import datasets” y no admite la importación de archivos CSV personalizados, para crear el gráfico de este ejemplo, tuvimos que generar una lista utilizando el bloque “create list with” de la categoría “Lists” y cargar manualmente los datos que deseábamos exponer. Luego mostrar el gráfico en consola con el bloque “show plot canvas”.



**Figura 2. Interfaz de usuario de BlockPy**

Si bien BlockPy es simple y permite abordar un análisis de datos simple, presenta varias limitaciones que impiden su utilización en las escuelas argentinas y justifican el desarrollo de nuevas herramientas:

- **Idioma:** una de las principales desventajas es su disponibilidad exclusiva en inglés. Este hecho puede representar un desafío considerable para usuarios que no dominan el idioma, limitando su capacidad para aprender de manera efectiva y aprovechar al máximo la herramienta.
- **Archivos CSV personalizados:** como indicamos anteriormente, BlockPy limita a los usuarios a trabajar con un conjunto precargado de datasets y no permite la importación de archivos CSV personalizados. Esto restringe la capacidad de los usuarios para explorar y analizar datos específicos y únicos que podrían ser relevantes para sus necesidades o intereses particulares. La capacidad de trabajar con datos personalizados es esencial para una experiencia de aprendizaje más completa y realista, permitiendo a los usuarios aplicar técnicas analíticas a datos que sean relevantes para sus proyectos o investigaciones personales.
- **Utilización de pandas y Plotly:** otro aspecto en el que BlockPy muestra limitaciones es en su enfoque hacia la Ciencia de Datos. Se dedica principalmente a la enseñanza básica de Python y a la creación de gráficos estáticos con Matplotlib, sin embargo, no ofrece un enfoque más avanzado en el análisis de datos. Es decir, la capacidad de manipular datos con herramientas especializadas como la librería pandas y de generar gráficos interactivos con Plotly, que permiten visualizaciones más dinámicas y detalladas, está ausente. A modo de ejemplo, no provee bloques para explorar las características de los datasets, como encontrar valores nulos, agrupamientos, procesamiento de columnas específicas, etc.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD NACIONAL DE LA PLATA

La **Figura 3** muestra la lista de datasets disponibles del lado izquierdo y los bloques generados a partir de la inclusión de “Aids” y “Airlines” del lado derecho. Estos utilizan la función “get” que devuelve una lista con los valores de la columna seleccionada y “filter” que permite aplicar limitadas opciones de filtrado sobre dichos valores.

The screenshot shows the PandasGUI interface. On the left, a window titled 'Lista de datasets disponibles' lists several datasets with their descriptions and 'Load' buttons. The 'Aids' and 'Airlines' datasets are marked as 'Loaded'. On the right, a workspace shows two generated blocks: 'airlines.get Airport.Code' and 'aids.get Country', both with 'filter (None)' options. A central menu lists various block types like Variables, Decisions, Iteration, Functions, Calculation, Output, Input, Turtles, Values, Conversion, Lists, Dictionaries, and Data - Aids/Airlines. At the top, there are buttons for 'Run', 'Blocks', 'Split', 'Text', 'Reset', and 'Import datasets'.

**Figura 3. Datasets disponibles junto a los bloques generados**

## 4.2.2 PandasGUI

PandasGUI es una herramienta desarrollada por Adam Rose en 2019 que proporciona una interfaz gráfica de usuario (GUI) para la manipulación y análisis de datos, aprovechando la potencia de la librería pandas de Python. Esta aplicación permite a los usuarios, incluso aquellos con poca experiencia en programación, interactuar de manera visual con sus conjuntos de datos, posibilitando su carga y visualización, la aplicación de filtros, la realización de agrupaciones y la ejecución de funciones de análisis sin necesidad de escribir código. Entre sus funcionalidades destacadas, PandasGUI ofrece la posibilidad de cargar datasets personalizados y explorar los datos de manera gráfica e interactiva utilizando Plotly, permitiendo detectar patrones, inconsistencias o realizar ajustes directamente desde la interfaz gráfica. Además, al ser de código abierto, esta aplicación puede integrarse en flujos de trabajo existentes y personalizarse según las necesidades específicas de análisis de datos, convirtiéndose en un recurso valioso para profesionales.

En la **Figura 4**, mostramos la interfaz de usuario de PandasGUI. En ella visualizamos un gráfico de dispersión creado a partir del dataset “load\_breast\_cancer” de la biblioteca Scikit-Learn<sup>20</sup>, que contiene información sobre diferentes características de los núcleos celulares. Se indica la relación entre dos columnas, “mean área” (promedio del área de las células) en el eje X y “mean concavity” (promedio de la concavidad de los núcleos celulares)

<sup>20</sup> <https://scikit-learn.org/stable/>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

en

el

eje

Y.

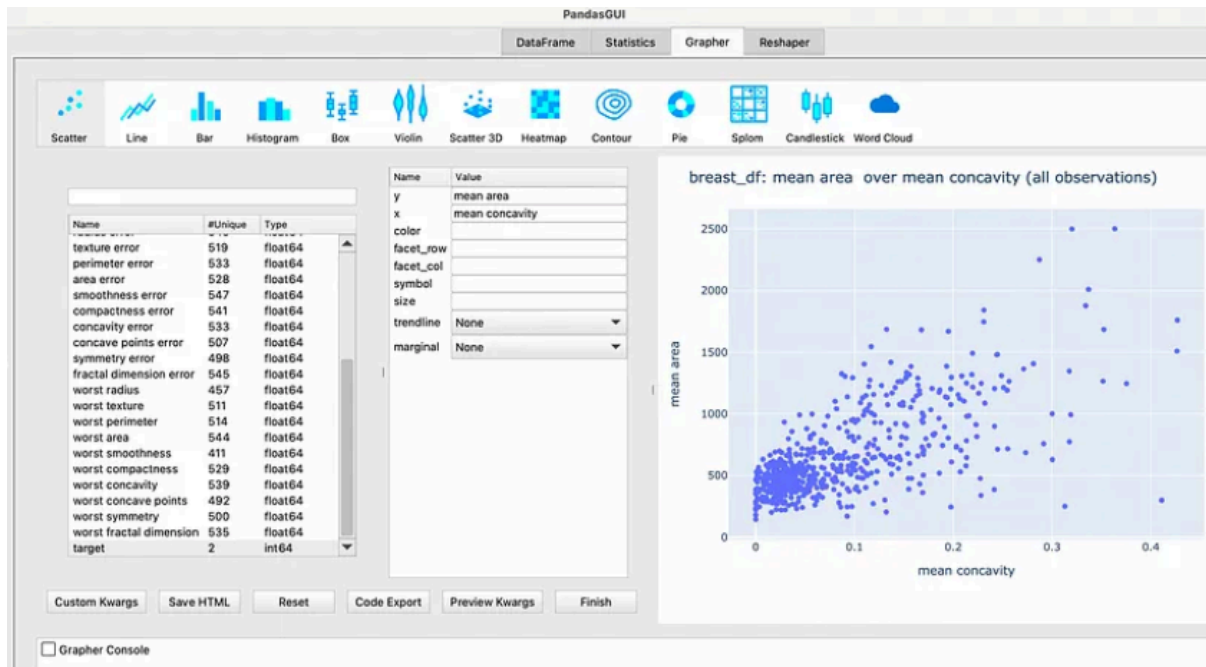


Figura 4. Interfaz de usuario de PandasGUI

Si bien, como mostramos en el ejemplo anterior, PandasGUI sí permite trabajar con datasets externos, presenta algunas desventajas que hacen difícil su uso:

- **Idioma:** la interfaz de PandasGUI está en inglés, lo que añade una barrera adicional para quienes no dominan el idioma, complicando aún más el proceso de navegación y el aprovechamiento de sus funcionalidades.
- **Interfaz poco amigable:** para aquellos estudiantes de nivel escolar o principiantes que estén dando sus primeros pasos en análisis de datos, la interfaz puede ser abrumadora y poco intuitiva. La abundancia de opciones y la falta de una guía clara hacen que la herramienta sea difícil de entender y utilizar sin asistencia. Esto puede generar frustración y dificultar el aprendizaje autónomo.
- **Instalación:** otro aspecto a considerar, es que se trata de una aplicación de escritorio y requiere ser instalada en entornos locales, lo cual puede ser un obstáculo para quienes no están familiarizados con la configuración de entornos de desarrollo y buscan una solución más sencilla y accesible.

### 4.2.3 Jupyter Notebook

Jupyter Notebook es un proyecto de código abierto sin fines de lucro que nació a partir del proyecto IPython en 2014. Es ampliamente utilizado en el ámbito educativo por su flexibilidad y capacidad de integrar texto y código en un mismo documento. Una de las



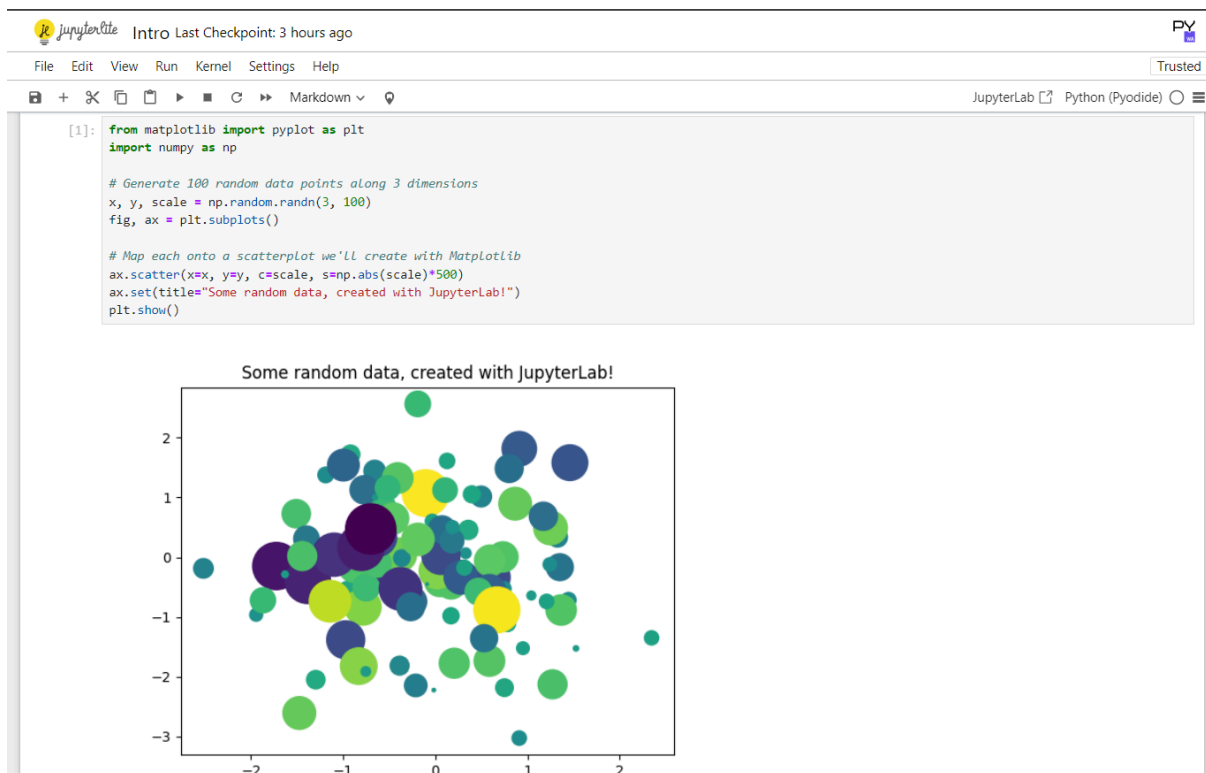
FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

grandes ventajas de esta aplicación es que permite ejecutar código Python en celdas de código dentro de un documento, permitiendo a los usuarios ver los resultados inmediatamente. Esta característica es especialmente útil para realizar pruebas y ajustes en tiempo real. Jupyter Notebook debe ser utilizado a través del navegador y, además, soporta una amplia gama de extensiones y librerías, incluyendo pandas y Plotly. Estas pueden ser utilizadas para cargar y analizar datasets personalizados lo que lo convierte en una herramienta muy poderosa.

En la **Figura 5** mostramos la interfaz de usuario de Jupyter Notebook donde puede visualizarse una celda de código Python y el resultado de su ejecución, una representación de un gráfico de dispersión, donde los puntos están distribuidos aleatoriamente.



**Figura 5. Interfaz de usuario de Jupyter Notebook**

A pesar de presentar numerosas ventajas, principalmente para aquellas personas que ya tienen conocimientos de Python, Jupyter Notebook tiene algunas desventajas que son importantes de considerar:

- **Idioma:** la herramienta se puede utilizar solo en inglés, francés o chino, lo que puede limitar su uso a aquellos usuarios que tienen poco o nulo dominio en estos idiomas.
- **Conocimientos previos de Python:** si bien es una ventaja para quienes ya programan en Python, Jupyter Notebook puede ser desafiante para principiantes. La necesidad





de tener conocimientos previos en Python, con su propia sintaxis y semántica, añade una capa adicional de complejidad.

### 4.3 Comparación de las herramientas analizadas

En las secciones previas describimos las herramientas analizadas de acuerdo a los criterios definidos en el apartado 4.1. A modo de resumen, generamos una tabla (**Tabla 2**) donde sintetizamos las características con las que cumple cada aplicación:

	<b>BlockPy</b>	<b>PandasGUI</b>	<b>Jupyter Notebook</b>
<b>Interfaz en español</b>	No	No	No
<b>Es código abierto</b>	Sí	Sí	Sí
<b>Interfaz basada en bloques</b>	Sí	No	No
<b>Interacción del usuario con pandas</b>	No	Sí	Sí
<b>Interacción del usuario con Plotly</b>	No	Sí	Sí
<b>Trabajo con archivos CSV personalizados</b>	No <sup>21</sup>	Sí	Sí
<b>Visualización de gráficos interactiva</b>	No	Sí	Sí
<b>Aplicación web</b>	Sí	No	Sí
<b>Aplicación de escritorio</b>	No	Sí	No
<b>Requiere conocimiento previo de Python</b>	No	No	Sí

**Tabla 2. Características de herramientas analizadas**

<sup>21</sup> BlockPy no permite cargar archivos CSV personalizados, pero en la **Figura 2** mostramos como se puede simular esta carga.





FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## Capítulo 5. Pandalyze

En este capítulo presentamos Pandalyze, una herramienta desarrollada para introducir los aspectos básicos de la Ciencia de Datos en personas que no requieren necesariamente conocimientos previos de programación.

Queremos destacar que Pandalyze fue presentada en el artículo 15479 - "Herramientas Informáticas utilizadas para Introducir Aspectos Básicos de Ciencia de Datos en la Escuela Secundaria", y ha sido aceptada para su exposición y publicación en el Congreso Argentino de Ciencias de la Computación (CACIC) [26]. Además, bajo el nombre "Pandalyze: una introducción al análisis de datos programado con bloques", también ha sido seleccionada para formar parte de las Jornadas Argentinas de Didáctica de las Ciencias de la Computación (JADICC) [27].

### 5.1 Introducción

El nombre "Pandalyze" surgió de la combinación entre "pandas", biblioteca de Python utilizada para el análisis de datos que explicaremos en detalle más adelante, y "analyze", por su traducción al español, analizar, examinar y comprender datos. La elección de este nombre refleja el objetivo central de nuestra aplicación: enseñar Ciencia de Datos de una manera accesible y visualmente intuitiva. A través de esta aplicación, los usuarios pueden utilizar los distintos bloques sin necesidad de tener conocimientos avanzados de programación desde el inicio y aprender conceptos fundamentales de Ciencia de Datos.

Es importante resaltar que Pandalyze es un proyecto de software libre y de código abierto, lo que permite a otros desarrolladores contribuir en su mejora y expansión. Esta característica, garantiza que la herramienta pueda evolucionar y adaptarse a las necesidades cambiantes de los usuarios y al avance de la Ciencia de Datos.

Se ha instalado, de forma temporal, una versión de prueba en este enlace: <https://668fdd58badbd2000869cde0--pandalyze.netlify.app/>

El código fuente del proyecto y las guías de instalación se encuentran en:

<https://github.com/paulamarcon/pandalyze-frontend>

<https://github.com/Tomgit97/pandalyze-backend>

#### 5.1.1 El por qué de la programación en bloques

Como mencionamos en el apartado 1.3, la enseñanza tradicional de la programación puede resultar intimidante para los estudiantes que se están familiarizando con este campo, por lo que la programación en bloques es una estrategia didáctica, amigable y muy utilizada para introducirlos en este mundo.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

A continuación, citamos la definición y el por qué de utilizar bloques según la “Fundación Sadosky”, ya que creemos que resume correctamente la simplicidad y efectividad de este enfoque, destacando su capacidad para facilitar el aprendizaje:

*“Programar por medio de bloques es desarrollar programas con acciones e instrucciones incorporadas en bloques o piezas prediseñadas. El resultado de encastrar los bloques entre sí es el programa que resuelve el problema o desafío planteado. Los bloques se seleccionan, arrastran, encastran y listo.*

*Esta modalidad es muy simple e intuitiva, ya que no requiere de que sepamos lidiar con los problemas de escribir código (como poner llaves, espacios o paréntesis). Esto ayuda a disminuir errores iniciales y acelerar el aprendizaje de quienes recién están aprendiendo a programar, porque cada concepto abstracto asociado a la programación tiene su representación visual” [28].*

Por este motivo, decidimos implementar Pandayze, proveyendo una interfaz con instrucciones asociadas a bloques vinculados a un determinado conjunto de funcionalidades provistas por las librerías pandas y Plotly.

## 5.2 Tecnologías utilizadas

Pandalyze fue pensada como una aplicación web que provee un frontend a través del cual es posible cargar archivos CSV y construir programas arrastrando bloques. Estos bloques se traducen automáticamente a código Python el cual es enviado a un backend para su ejecución, devolviendo los resultados en forma de texto o imágenes interactivas que permiten visualizar los datos.

### 5.2.1 Tecnologías del frontend

Para la construcción del frontend, utilizamos React<sup>22</sup> para construir la interfaz de usuario, la librería Blockly de Google<sup>23</sup> para la creación y utilización de bloques y la herramienta CodeMirror<sup>24</sup> para mostrar su traducción a código Python. Seguidamente explicamos con qué fin utilizamos cada herramienta:

- **React**: es una biblioteca de JavaScript ampliamente utilizada para construir interfaces de usuario interactivas. Su capacidad para funcionar con componentes reutilizables nos permitió construir una interfaz de usuario modular y dinámica [29]. Además, React cuenta con una gran comunidad de desarrolladores<sup>25</sup>, lo que facilitó la resolución de problemas y la incorporación de buenas prácticas en el desarrollo.

<sup>22</sup> <https://es.react.dev/>

<sup>23</sup> <https://developers.google.com/blockly>

<sup>24</sup> <https://codemirror.net/>

<sup>25</sup> <https://es.react.dev/community>



De acuerdo a nuestra experiencia, a diferencia de frameworks más completos como Angular, consideramos que React ofrece una mayor flexibilidad y una menor sobrecarga, lo que resultó en un enfoque más ágil y eficiente para desarrollar nuestra aplicación.

- **Blockly**: es una librería de código abierto que permite incluir un editor de código basado en bloques en una aplicación. Blockly se destacó como una opción ideal debido a su capacidad para simplificar la introducción a la programación mediante el uso de bloques visuales. Como dijimos al inicio de este capítulo, este enfoque permite construir programas arrastrando y encajando bloques, facilitando la comprensión de conceptos de programación y eliminando la necesidad de lidiar con la sintaxis textual del código.

Otra fortaleza de Blockly es su documentación completa y accesible. Esta proporciona guías detalladas, ejemplos prácticos y tutoriales que simplifican la integración y personalización de la librería.

Además, Blockly es personalizable, lo que nos permitió adaptar los bloques a los requerimientos específicos de nuestra herramienta. Pudimos diseñar bloques que se alinean con la manipulación de datos usando pandas y la creación de gráficos con Plotly, proporcionando una experiencia de usuario coherente y específica para la Ciencia de Datos.

- **CodeMirror**: para la traducción de los bloques utilizados a su correspondiente código Python, utilizamos CodeMirror, una herramienta que proporciona un editor de código enriquecido dentro del navegador. CodeMirror nos permitió fácilmente visualizar el código generado, y contar con sus funcionalidades avanzadas como el resaltado de sintaxis. Esto ayuda a los usuarios a familiarizarse con la estructura y la sintaxis del código Python de manera más intuitiva.

Si bien implementamos la traducción de bloques a código Python, no consideramos necesario realizar la conversión inversa, es decir, permitir la edición del código en CodeMirror para que se traduzca nuevamente a bloques. Esta decisión se basó en la idea de que, si los usuarios ya tienen la capacidad de escribir y entender código Python, entonces volver a hacer uso de bloques sería un retroceso en su aprendizaje. El objetivo de nuestra aplicación es que, hagan uso de los bloques, analicen y comprendan la sintaxis y cómo se comporta el código generado a partir de esos bloques y, de esta manera, puedan avanzar y profundizar en sus conocimientos de programación textual. En todo caso, si los usuarios desean modificar el código traducido, pueden copiar el mismo, editarlo y ejecutarlo en aplicaciones más avanzadas tales como Jupyter Notebook. No obstante, la implementación de esta funcionalidad de conversión inversa queda como una posible línea de trabajo futura para aquellos que prefieran una mayor integración



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

entre bloques y código.

## 5.2.2 Tecnologías del backend

Para el desarrollo del backend, utilizamos Flask<sup>26</sup> como framework, PostgreSQL<sup>27</sup> como base de datos, la librería pandas para la manipulación de datos y Plotly para su visualización. A continuación, ofrecemos una descripción detallada de cada herramienta y su aplicación en nuestro proyecto:

- **Flask**: es un framework minimalista escrito en Python que se adaptó perfectamente a nuestras necesidades debido a su simplicidad y flexibilidad. Flask es conocido por su enfoque "micro", lo que significa que proporciona solo los componentes esenciales para el desarrollo web, sin imponer una estructura rígida o una sobrecarga significativa. Este enfoque fue especialmente útil para nosotros, ya que solo necesitábamos implementar un número pequeño de funcionalidades.

Flask facilita la integración con otras librerías y herramientas en Python, lo cual fue clave para nuestro trabajo, ya que usamos pandas y Plotly para el análisis y visualización de datos. La integración fluida con estas librerías es una de las razones por las que Flask fue una elección adecuada para nuestro backend.

Otro factor importante en nuestra decisión fue que ya contábamos con experiencia previa en Flask. Esto nos permitió comenzar el desarrollo de manera más rápida y eficiente, evitando el tiempo de aprendizaje que habría requerido un framework diferente. Asimismo, en comparación con otros frameworks más complejos y completos como Django, Flask ofrece una solución más ligera y minimalista sin características innecesarias [30].

- **PostgreSQL**: como sistema de gestión de base de datos, elegimos PostgreSQL versión 16, por ser una base de datos relacional potente y de código abierto, conocida por su fiabilidad y capacidad para manejar grandes cantidades de datos, además es ampliamente reconocida por su sólida documentación y su soporte para una variedad de funciones avanzadas, lo cual resultó beneficioso durante el desarrollo [31].

Otro aspecto clave en nuestra elección fue la integración sencilla con el entorno de despliegue utilizado, Render<sup>28</sup>, que ofrece una configuración fácil para bases de datos PostgreSQL. Esta compatibilidad redujo la complejidad del proceso de despliegue y facilitó la gestión de la base de datos en producción.

<sup>26</sup> <https://flask.palletsprojects.com/en/3.0.x/>

<sup>27</sup> <https://www.postgresql.org/>

<sup>28</sup> <https://render.com/>



La decisión de utilizar una base de datos, surgió para poder almacenar de manera eficiente los archivos CSV que cargan los usuarios, evitando la necesidad de enviarlos constantemente al backend en cada operación. Al hacer esto, reducimos la carga de red y mejoramos el rendimiento general de la aplicación y del cliente. Además, dado que Pandalyze está orientada a la educación, es probable que los usuarios utilicen el mismo archivo CSV para resolver múltiples ejercicios, lo que hace que la capacidad de la base de datos para gestionar y acceder a estos archivos de manera rápida y organizada sea aún más valiosa.

Por último, contar con una base de datos optimiza la eficiencia del sistema y proporciona flexibilidad para futuras funcionalidades, como el historial de operaciones o la colaboración entre usuarios.

- Pandas: como parte de nuestros objetivos planteados en el **capítulo 1**, utilizamos pandas para la manipulación y análisis de datos. Con esta herramienta, permitimos que los usuarios puedan manejar grandes volúmenes de datos de manera eficiente, facilitando operaciones complejas como la carga, limpieza, transformación y análisis de datos. Sus estructuras de datos principales, el DataFrame<sup>29</sup> y las Series<sup>30</sup>, proporcionan una forma poderosa y flexible de trabajar con datos, posibilitando realizar operaciones como filtrado, agrupamiento y fusión con gran facilidad.
- Plotly: esta librería ofrece una gama de gráficos interactivos que permiten una visualización clara y atractiva de los datos y también facilitan la exploración dinámica de los mismos. A diferencia de Matplotlib, otra librería muy popular utilizada para la visualización de datos que se centra en gráficos estáticos, elegimos Plotly que proporciona una experiencia más rica y envolvente mediante la interacción en tiempo real. Los gráficos creados con Plotly pueden ser manipulados y explorados por el usuario, lo que ayuda a descubrir patrones y detalles que podrían no ser evidentes en visualizaciones estáticas.

La integración de Plotly con pandas es especialmente beneficiosa, ya que permite crear visualizaciones interactivas directamente a partir de los DataFrames de pandas. Esta combinación no solo mejora la eficiencia del proceso de visualización, sino que también facilita una comprensión más profunda y detallada de los datos. Pandas y Plotly juntas ofrecen una solución robusta y flexible para el análisis y la visualización de datos.

---

<sup>29</sup> Un DataFrame es una tabla flexible que organiza datos en filas y columnas, permitiendo realizar fácilmente operaciones y análisis [32].

<sup>30</sup> Una series es una columna de datos (valores), similar a una lista, pero con un índice (etiqueta) que identifica cada elemento. Es una estructura unidimensional para almacenar datos de un solo tipo [33].



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

### 5.2.3 Limitaciones técnicas

En nuestra aplicación, contamos con diversas limitaciones que surgen tanto de las tecnologías utilizadas como de las necesidades de eficiencia y experiencia del usuario. Entre las restricciones establecidas, permitimos la carga de archivos con formato CSV de hasta 10 MB (megabytes), y una vez cargados, los usuarios no pueden eliminarlos. La decisión de limitar el tamaño fue tomada para garantizar un rendimiento óptimo del sistema, ya que trabajar con archivos de gran tamaño podría resultar en tiempos de carga más largos y una experiencia de usuario menos fluida.

Por otra parte, Pandalyze está en funcionamiento y disponible para su uso. El frontend está alojado en Netlify<sup>31</sup>, mientras que el backend, que incluye el servidor y la base de datos, está desplegado en Render. Ambos componentes se encuentran en servicios gratuitos, lo que conlleva ciertas limitaciones. En particular, Render tiene un sistema de suspensión automática que se activa cuando no hay actividad en la aplicación durante un período de 15 minutos. Esto significa que, una vez suspendido, el primer usuario en acceder después de este intervalo experimentará tiempos de respuesta más largos, ya que el servicio tarda aproximadamente cinco minutos en volver a activarse. Esta demora inicial puede afectar la experiencia de los usuarios que acceden a la aplicación tras un período de inactividad. Asimismo, Render ofrece 256 MB de RAM y limita el espacio de la base de datos a 1 GB (gigabyte). Por último, debido a las restricciones del servicio gratuito, la base de datos debe ser desplegada de nuevo mensualmente, ya que la instancia se desactiva automáticamente después de un mes.

Cabe aclarar que estas restricciones están dadas por el servicio que elegimos para instalar la versión de prueba. En caso de instalarse en un servidor con diferentes restricciones de infraestructura, esto dependerá de las características particulares del mismo, y que son independientes de Pandalyze.

### 5.2.4 Alternativas consideradas y decisiones finales

Tomamos la decisión de contar con un backend ya que prescindir de él habría conllevado algunas desventajas. Ejecutar toda la lógica del código directamente en el cliente habría implicado desafíos significativos tanto en términos de seguridad como de eficiencia. La ejecución de código Python en el navegador requiere convertirlo en un formato interpretable por el mismo, lo que resulta en un proceso complejo y menos eficiente. Para esta conversión existen soluciones como son las herramientas Thebe<sup>32</sup> y Skulpt<sup>33</sup> que abordan la ejecución de código en el cliente, pero ambas presentan limitaciones notables:

<sup>31</sup> <https://www.netlify.com/>

<sup>32</sup> <https://thebe.readthedocs.io/en/stable/>

<sup>33</sup> <https://skulpt.org/>





FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

- Thebe: permite ejecutar código en el navegador mediante la integración con Jupyter Notebook. Sin embargo, la carga y procesamiento de archivos CSV directamente en el cliente es lenta y requiere una configuración avanzada y un manejo complejo de sesiones. Esta solución puede resultar en una implementación menos elegante y con una sobrecarga significativa en el lado del cliente.
- Skulpt: es un intérprete de Python escrito en JavaScript que permite ejecutar Python en el navegador. Sin embargo, su rendimiento puede ser limitado y no siempre es compatible con todas las características y bibliotecas de Python, lo que puede restringir la funcionalidad de la aplicación.

Por lo tanto, inclinarnos por un backend nos brindó ciertas ventajas:

- 1) Nos permitió separar el cliente del servidor, lo que mejoró la seguridad y flexibilidad de la aplicación al distribuir responsabilidades y optimizó la ejecución de procesos más complejos. Además, esta división facilita la escalabilidad de la aplicación, ya que las actualizaciones y mejoras pueden implementarse de manera más eficiente. A modo de ejemplo, si en el futuro se desea agregar funcionalidades como sesiones de usuario y administración, contar con un backend ofrece una base sólida para implementar estas características de manera sencilla minimizando los cambios en el frontend.
- 2) Manejar el backend nos proporcionó un control más preciso sobre diversos aspectos críticos de la aplicación. Por ejemplo, la carga de archivos CSV en la base de datos la administramos directamente en el servidor, esto nos garantizó un manejo eficiente y seguro de los datos. También pudimos controlar la salida de la aplicación de manera más efectiva, gestionando la salida de texto, imágenes generadas con Plotly y errores personalizados. Esto último ayuda al usuario a entender mejor el problema y trabajar para solucionarlo de manera autónoma, fomentando así un aprendizaje más profundo y significativo mejorando su experiencia.

### 5.3 Aspectos destacados de la interfaz de usuario

Hay distintas secciones específicas en Pandalyze y a continuación hacemos un recorrido por todas ellas.

Empezamos visualizando en la **Figura 6** la interfaz de usuario completa de nuestra aplicación.



**Pandalyze: aprender Ciencia de Datos con programación en bloques**

Ver tutorial | Sección de herramientas | Ejecutar código

Cargar CSV | Ejemplos | Eliminar bloques

**Sección de bloques**

- Variables
- Literales
- Pandas
  - Acceso a datos
  - Información
  - Cálculos
  - Filtros
  - Distribución
- Plotly
- Salida

**Sección de espacio de trabajo (workspace)**

```
print("Primeras 10 filas")
print(data_frame = read_csv(Lagos.csv) . head(10) )
plotly.bar(data_frame = read_csv(Lagos.csv) . x = data_frame . read_csv(Lagos.csv) . Nombre . y = data_frame . read_csv(Lagos.csv) . Superficie)
```

**Sección de código Python**

```
1 print("Primeras 10 filas:")
2 print(pandas.read_csv(Lagos.csv).head(10))
3 plotly.bar(data_frame = pandas.read_csv(Lagos.csv), x = pandas.read_csv(Lagos.csv).Nombre, y = pandas.read_csv(Lagos.csv).Superficie)
4
```

**Consola**

Primeras 10 filas:

	Nombre	Ubicación	Superficie (km²)	Profundidad máxima (m)	Profundidad media (m)	Coordenad
0	Lago Argentino	Santa Cruz	1435	300.0	150.0	50°14'53" S 72°57'15" W
1	Lago Viedma	Santa Cruz	1160	NaN	NaN	49°27'27" S 72°57'15" W
2	Lago Nahuel Huapi	Río Negro / Neuquén	540	464.0	157.0	40°51'16" S 71°57'15" W
3	Lago Musters	Chubut	427	39.0	20.0	45°24'53" S 67°57'15" W
4	Lago Cardiel	Santa Cruz	358	NaN	49.0	48°55'37" S 71°57'15" W
5	Lago Epecuén	Buenos Aires	156	11.0	7.0	37°27'45" S 62°57'15" W
6	Lago Pellegrini	Río Negro	116	NaN	NaN	38°42'25" S 67°57'15" W
7	Lago Strobel	Santa Cruz	100	NaN	NaN	48°25'58" S 72°57'15" W
8	Lago Fontana	Chubut	84	NaN	79.0	44°55'17" S 71°57'15" W
9	Lago Huechulafquen	Neuquén	84	NaN	142.0	39°46'28" S 72°57'15" W

**Superficies**

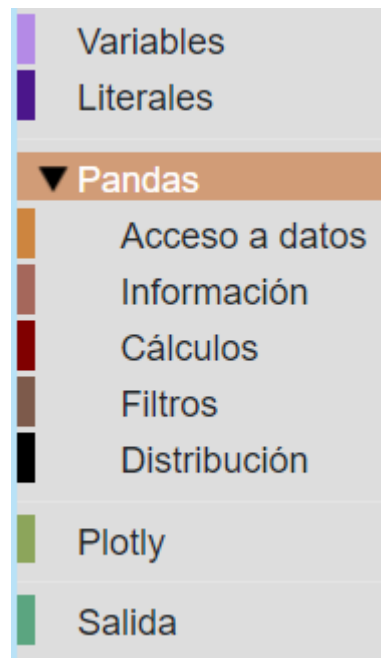
**Figura 6. Interfaz de usuario de Pandalyze**

Luego, pasando a las distintas secciones, contamos con:

- **Sección de bloques:** los diferentes bloques a utilizar se encuentran divididos por categoría según su funcionalidad. En la **Figura 7** podemos observar esta distinción distribuida en:
  - **Variables:** permite crear variables y ofrece bloques para asignarles valores y recuperarlos cuando sea necesario.
  - **Literales:** proporciona bloques para trabajar con valores primitivos. Permite utilizar datos fundamentales como números enteros, texto y valores de verdad.
  - **Pandas:** dividida en subcategorías con bloques relacionados a la librería pandas:
    - **Acceso a datos:** esta subcategoría está enfocada en la importación y el acceso a los datos dentro del DataFrame. Permite cargar datos desde archivos CSV y acceder a la información estructural del DataFrame, como los valores y el índice.
    - **Información:** proporciona herramientas para explorar y comprender la estructura y el contenido de los datos. Obtiene una vista preliminar de los datos, una descripción general del DataFrame y estadísticas descriptivas básicas.
    - **Cálculos:** está diseñada para realizar análisis estadísticos y agregaciones sobre los datos. Facilita contar valores, encontrar los valores máximos y mínimos, y calcular medias, entre otras operaciones básicas.
    - **Filtros:** aquí el usuario puede realizar la selección y manipulación de datos basados en condiciones específicas. Posibilita aplicar criterios para extraer subconjuntos de datos dentro del DataFrame y facilita la

asignación de resultados filtrados a nuevas variables, ayudando a personalizar y ajustar el análisis de datos según necesidades particulares.

- **Distribución:** se ocupa de la organización y agrupación de los datos. Facilita ordenar los datos y agruparlos en función de una o más columnas, lo que ayuda en el análisis y la visualización de la distribución de la información.
- **Plotly:** ofrece bloques para visualizar datos en forma de gráficos interactivos. Es posible representar conjuntos de datos mediante gráficos de líneas, barras, dispersión y torta.
- **Salida:** permite mostrar los resultados de las operaciones realizadas. Incluye bloques para mostrar texto en pantalla y para visualizar imágenes.



**Figura 7. Categorías de los bloques**

A continuación, en la **Figura 8**, mostramos a modo de ejemplo desplegada la subcategoría 'Acceso a datos' y en ella los bloques 'read\_csv', 'values' e 'index'.



Figura 8. Bloques dentro de la subcategoría 'Acceso a datos'

- Sección de espacio de trabajo: en esta sección se visualizan los bloques utilizados que conforman el programa. Estos son elegidos y arrastrados por el usuario al espacio de trabajo, donde los arma y encastra según las necesidades y lo que desee hacer.

En la **Figura 9**, podemos observar que también cuenta con un cesto de basura en la esquina inferior derecha para desechar aquellos bloques que no queramos usar y un + (más) y - (menos) para hacer zoom in y zoom out respectivamente.

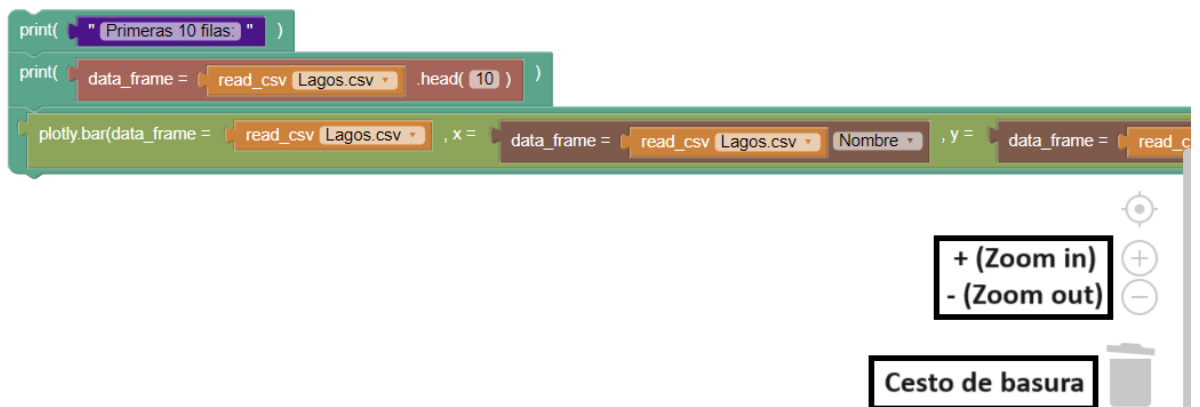


Figura 9. Espacio de trabajo

- Sección de código Python: aquí exhibimos el código Python equivalente al programa realizado con bloques. Cada bloque que el usuario arrastre, se traducirá a su

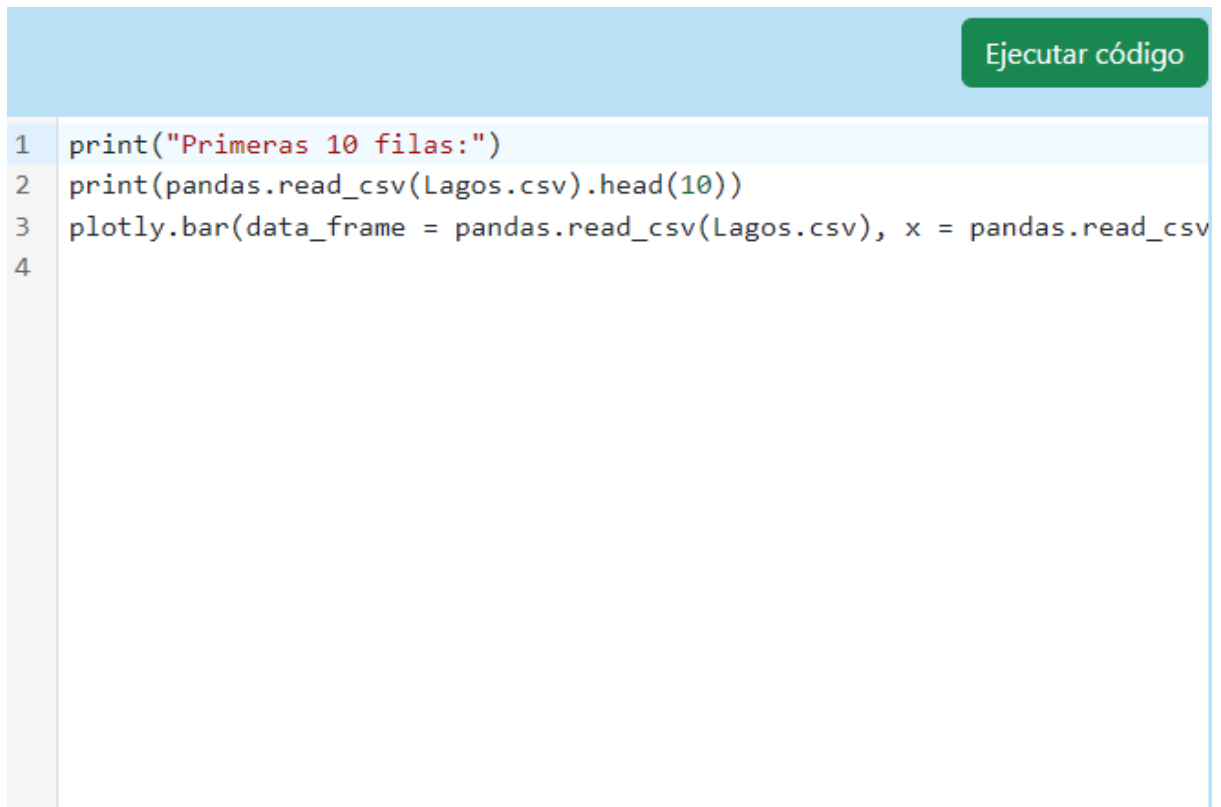


FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

equivalente en código y se mostrará en esta sección. También, como observamos en la **Figura 10**, presentamos el botón “Ejecutar código”, que permite la ejecución del programa.



```
1 print("Primeras 10 filas:")
2 print(pandas.read_csv(Lagos.csv).head(10))
3 plotly.bar(data_frame = pandas.read_csv(Lagos.csv), x = pandas.read_csv
4
```

**Figura 10. Código Python**

- Sección de herramientas: esta sección contiene los distintos botones a utilizar. Ver **Figura 11**.
  - **Ver tutorial**: le permite al usuario visualizar un tutorial donde se demuestra un pequeño paso a paso del uso de Pandalyze. Este se puede omitir y, sino, cuenta con 6 diapositivas sobre cómo cargar un CSV, cómo arrastrar bloques al espacio de trabajo, cómo graficar, entre otros. En la **Figura 12** podemos observar la primera dispositiva del tutorial.
  - **Cargar CSV**: permite seleccionar y cargar un archivo CSV desde el sistema de archivos del usuario.
  - **Ejemplos**: despliega un ejemplo prearmado que, al seleccionarlo, se cargan automáticamente los bloques correspondientes en el espacio de trabajo, generando el código Python listo para ser ejecutado.
  - **Eliminar bloques**: funciona para limpiar y eliminar todos los bloques del espacio de trabajo.

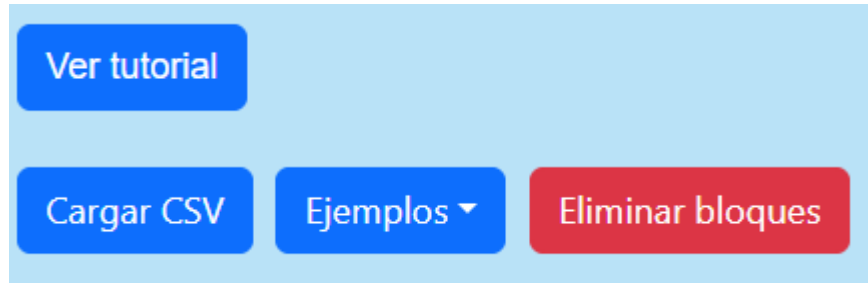
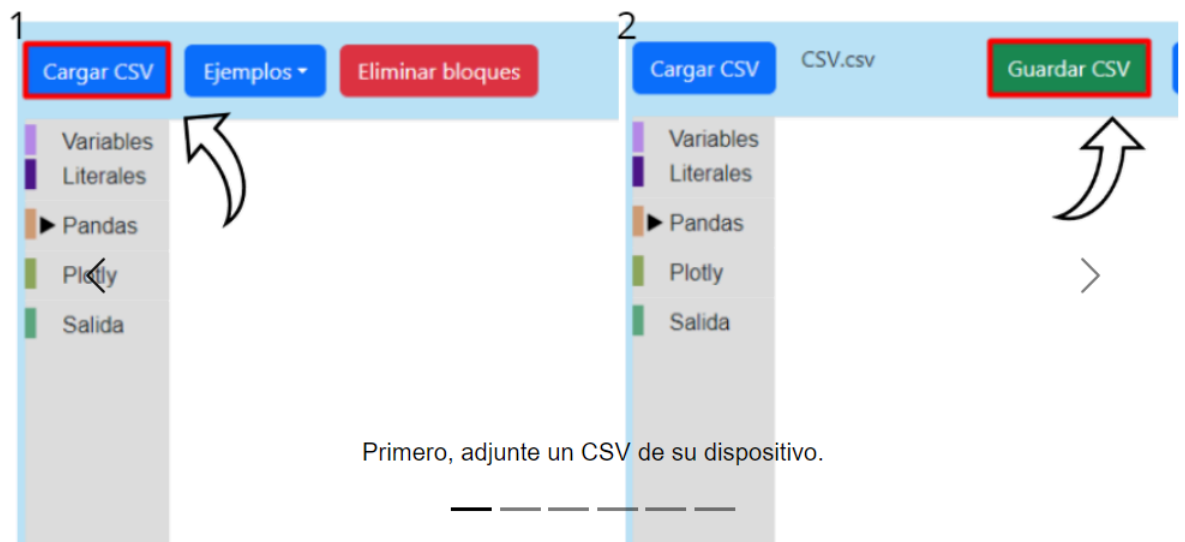


Figura 11. Botones

## ¡Bienvenidos a Pandalyze!



A continuación, un pequeño tutorial sobre algunos usos de la aplicación. Presione **Omitir tutorial** si lo desea.



Omitir tutorial

Figura 12. Primera diapositiva del tutorial

- Consola: cuando el usuario presiona 'Ejecutar código', es en la consola donde se visualiza el resultado de la ejecución. En la **Figura 13** vemos lo obtenido de "print" y "show", bloques utilizados para mostrar texto y gráficos respectivamente.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD NACIONAL DE LA PLATA

Primeras 10 filas:

	Nombre	Ubicación	Superficie (km <sup>2</sup> )	Profundidad máxima (m)	Profundidad media (m)	Coordenad
0	Lago Argentino	Santa Cruz	1435	500.0	150.0	50°14'53"S 7
1	Lago Viedma	Santa Cruz	1166	NaN	NaN	49°37'27"S 7
2	Lago Nahuel Huapi	Río Negro / Neuquén	540	464.0	157.0	40°51'16"S 7
3	Lago Musters	Chubut	427	39.0	20.0	45°24'53"S 6
4	Lago Cardiel	Santa Cruz	358	NaN	49.0	48°55'3"S 71
5	Lago Epecuén	Buenos Aires	156	11.0	7.0	37°7'45"S 62
6	Lago Pellegrini	Río Negro	116	NaN	NaN	38°42'25"S 6
7	Lago Strobel	Santa Cruz	100	NaN	NaN	48°25'58"S 7
8	Lago Fontana	Chubut	84	NaN	79.0	44°55'17"S 7
9	Lago Huelchulafquen	Neuquén	84	NaN	142.0	39°46'28"S 7

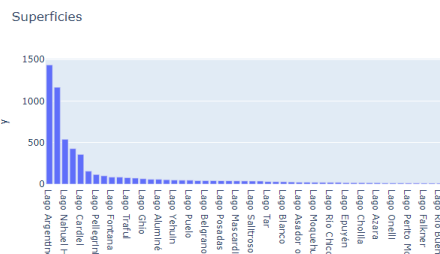


Figura 13. Consola

## 5.4 Funcionalidad: flujo y experiencia de usuario

Como dijimos a lo largo de este informe, Pandalyze es una herramienta diseñada para facilitar la introducción a la Ciencia de Datos. La aplicación se basa en la plataforma Blockly, ampliada con bloques específicos para utilizar las bibliotecas pandas y Plotly, permitiendo a los usuarios crear visualizaciones y realizar análisis de datos de manera intuitiva y visual.

A continuación, enumeramos cómo sería una secuencia de uso de la aplicación:

1. **Inicio y tutorial:** al acceder a Pandalyze, el usuario es recibido con una interfaz diseñada para ser intuitiva y accesible. Antes de comenzar, presentamos un tutorial que guía al usuario a través de las funcionalidades clave de la aplicación, desde la carga de archivos CSV hasta la ejecución del código, incluyendo algunas recomendaciones útiles para maximizar su uso. Este tutorial se despliega en un conjunto de diapositivas con imágenes y GIFs que ilustran cada paso de manera visual y sencilla. Cabe destacar que este tutorial puede omitirse si así se desea.
2. **Ejemplo inicial:** tras completar el tutorial o para quienes prefieran aprender practicando, la aplicación viene precargada con un ejemplo inicial básico. Este ejemplo muestra, en el espacio de trabajo, los bloques necesarios para exhibir la información de las diez primeras filas del CSV "Lagos"<sup>34</sup> que ya se encuentra cargado por defecto. Luego, al hacer clic en "Ejecutar código", la aplicación muestra el resultado de este ejemplo en la consola.
3. **Uso de bloques y su ejecución:** una vez que el usuario ha adquirido una noción general de cómo funciona Pandalyze, está listo para cargar los archivos CSV que desee desde la sección de herramientas o continuar con el archivo de datos por defecto que ofrece la aplicación. Podrá navegar por las distintas categorías de bloques y arrastrar y encastrar los necesarios en el espacio de trabajo para llegar a su objetivo y luego, en la sección de código Python, ver su traducción. Finalmente, al ejecutar el programa verá en consola el texto y/o los gráficos generados y estará en condiciones de realizar un análisis y sacar conclusiones al respecto.

<sup>34</sup> Se utilizó el dataset "Lagos", descargado del sitio del Instituto Geográfico Nacional. Este dataset contiene información sobre los lagos de la República Argentina.

<https://www.ign.gob.ar/NuestrasActividades/Geografia/DatosArgentina/Lagos>



## Capítulo 6. Pruebas y validaciones

En este capítulo presentamos los dos tipos de pruebas que desarrollamos para probar el correcto funcionamiento de la aplicación. Las primeras están relacionadas con la funcionalidad de Pandalyze y los cambios que fuimos realizando durante su construcción y, las segundas, corresponden a dos talleres desarrollados con estudiantes en el marco de actividades de extensión realizadas en la Facultad de Informática.

En las siguientes secciones detallamos cada una de estas pruebas.

### 6.1 Evaluación de funcionalidad

En un principio, teníamos una idea clara de cómo construir y estructurar la aplicación a nivel funcional. Sin embargo, a lo largo del proceso y en base a la retroalimentación recibida, surgieron necesidades adicionales que no habíamos previsto inicialmente. Tuvimos que ampliar las capacidades de la herramienta, sumando nuevos bloques y reorganizando las categorías en las que están divididos para mejorar la claridad y el uso de los mismos. Además, incorporamos un tutorial y nubes de diálogo sobre los bloques facilitando la navegación de la aplicación. Por último, en algunos bloques agregamos texto para ayudar a reconocer qué tipo de parámetro requiere cada espacio vacío.

#### 6.1.1 Nuevas funcionalidades

Cuando concluimos el desarrollo de Pandalyze, comenzamos a planificar la evaluación con un grupo de estudiantes que serían los primeros en utilizar la herramienta. Luego de diseñar la guía con los desafíos a realizar, que describimos en detalle en la sección **6.2**, surgió la necesidad de agregar algunos bloques adicionales a la aplicación. Esto nos llevó a ampliar las capacidades de la herramienta, añadiendo cuatro bloques clave: “columna”, “groupby”, “values” e “index”:

- columna: agregar este bloque fue valioso para poder, a partir de un DataFrame, recuperar una columna específica del mismo. Una vez obtenida la columna, el usuario puede realizar la operación que desee sobre ella. Por ejemplo, puede ejecutar un “count” para contar la cantidad de valores no nulos que hay en la columna, “max” para obtener su valor máximo, “min” para obtener su valor mínimo, entre otros.

Inicialmente, habíamos implementado el acceso a las columnas mediante un bloque de texto (string). Aunque la idea era sencilla, resultaba incómoda, ya que los usuarios tenían que recordar y escribir exactamente el nombre de la columna del DataFrame a la cual querían acceder, lo que iba a generar errores frecuentes. Para mejorar esta experiencia, creamos un bloque que primero espera un DataFrame y luego despliega los nombres de todas sus columnas. Esto permite a los usuarios seleccionar la columna deseada de manera más sencilla y eficiente. En la **Figura 14**





presentamos el resultado final de este bloque donde, a modo de ejemplo, a partir del archivo “Lagos.csv”, podemos obtener cualquiera de sus columnas: “Nombre”, “Ubicación”, “Superficie”, etc.

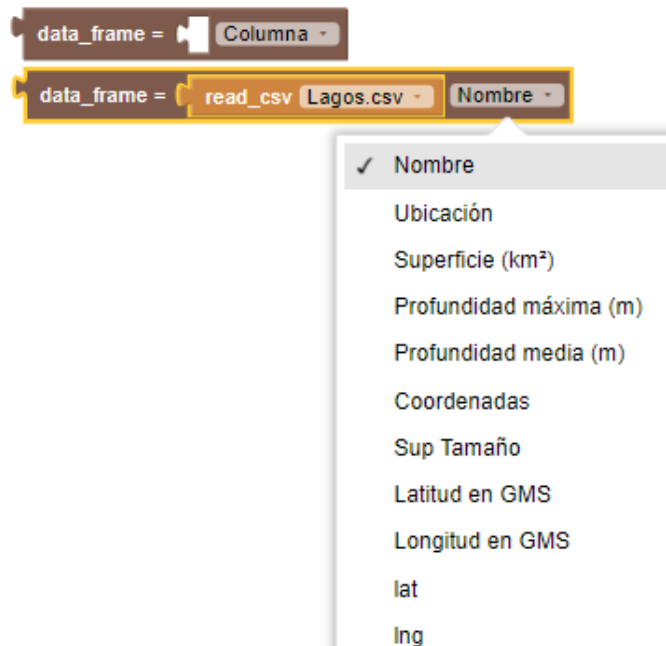


Figura 14. Bloque “columna”

- groupby: incorporamos este bloque para facilitar a los estudiantes la agrupación de datos en función de una columna específica dentro de un DataFrame. Por ejemplo, luego de hacer esta agrupación, se pueden aplicar funciones de agregación como suma, promedio o conteo a cada grupo. El “groupby” es fundamental en el análisis de datos, ya que permite analizar patrones y tendencias dentro de subgrupos específicos. En la **Figura 15** mostramos un ejemplo en el que utilizamos este bloque para obtener, por cada provincia (columna “Ubicación”), el valor máximo de cada columna. Para esto, primero creamos tres variables, “data\_frame”, “locations” y “grouped\_lakes”, donde almacenamos el archivo “Lagos.csv”, obtenemos todas las ubicaciones de los lagos y agrupamos utilizando el bloque “groupby” por la columna “Ubicación” respectivamente. Finalmente utilizamos la función “max” para calcular el valor máximo para cada columna dentro de cada grupo.

En la **Figura 16** presentamos la salida resultante de utilizar la función “max”. En ella podemos observar que, por ejemplo, en la provincia de Buenos Aires el lago de mayor superficie es aquel con 156km<sup>2</sup> y el lago más profundo alcanza los 11m.



```
set data_frame to read_csv Lagos.csv
set locations to data_frame = get data_frame Ubicación
set grouped_lakes to data_frame = get data_frame .groupby(columna = get locations )
print( get grouped_lakes .max() )
```

Figura 15. Ejemplo utilizando el bloque “groupby”

Ubicación	Nombre	Superficie (km <sup>2</sup> )	Profundidad máxima (m)	Profundidad media (m)	Coordenadas
Buenos Aires	Lago Epecuén	156	11.0	7.0	37°7'45"S 62°51'...
Chubut	Lago Río Chico	427	287.0	149.0	45°53'16"S 71°13...
Neuquén	Lago Tromen	84	277.0	166.0	40°9'45"S 71°26'...
Río Negro	Lago Perito Moreno	116	218.0	111.0	41°3'58"S 71°32'...
Río Negro / Neuquén	Lago Nahuel Huapi	540	464.0	157.0	40°51'16"S 71°36...
Santa Cruz	Lago Viedma	1435	500.0	150.0	50°6'30"S 73°20'...
Tierra del Fuego,...	Lago Yehuín	49	NaN	43.0	54°41'19"S 65°48...

Figura 16. Resultado de la ejecución de los bloques de la Figura 15

- values e index: inicialmente, los bloques de Plotly en la aplicación estaban diseñados para generar gráficos a partir de un DataFrame. Sin embargo, al probar la herramienta, nos dimos cuenta de que también sería útil poder graficar directamente a partir de una Series. Para lograr esto, era necesario extraer los valores y las etiquetas por separado, ya que algunos gráficos requieren estos componentes de datos en un formato específico. Por eso, añadimos los bloques “values” e “index”, que permiten a los usuarios extraer fácilmente estos componentes de una Series y usarlos en las visualizaciones.

En la **Figura 17** mostramos un ejemplo utilizando estos bloques para graficar la cantidad de lagos que hay por Ubicación. Primero, asignamos el archivo “Lagos.csv” a la variable “data\_frame” usando el bloque “read\_csv”. Luego, obtenemos la cantidad de valores únicos en la columna “Ubicación” y las almacenamos en la variable “location\_frecuencias” como una Series, utilizando el bloque “value\_counts”. A partir de esta Series, extraemos los valores (frecuencia de aparición de cada ubicación) y las etiquetas (ubicaciones), almacenándolos en las variables “values” e “indexes”, respectivamente. Por último, creamos un gráfico de torta utilizando el bloque “plotly.pie” y en la **Figura 18** presentamos el resultado obtenido.

```

set data_frame to read_csv(Lagos.csv)
set location_frequencies to data_frame = get(data_frame, Ubicación).value_counts()
set values to get(location_frequencies).values
set indexes to get(location_frequencies).index

plotly.pie(data_frame = get(location_frequencies), values = get(values), names = get(indexes), title = "Example") .show()

```

Figura 17. Ejemplo utilizando los bloques values e index

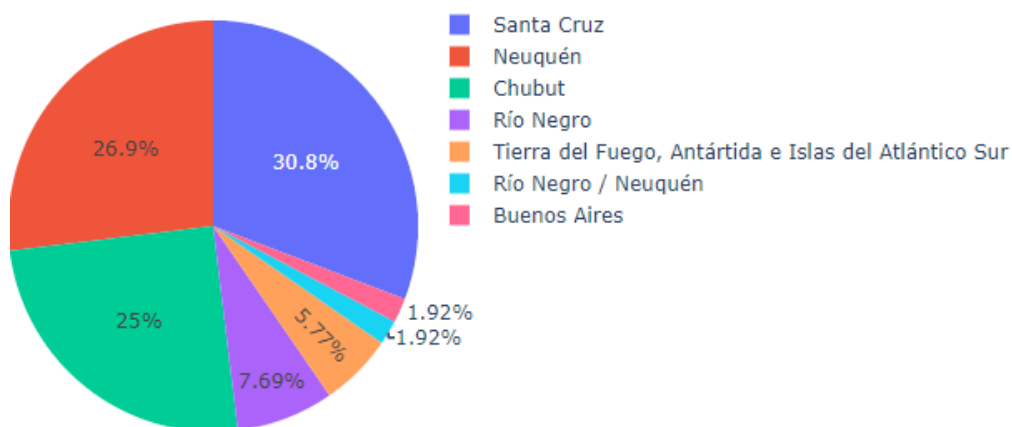


Figura 18. Gráfico de torta generado con el bloque “plotly.pie”

### 6.1.2 Modificaciones para mejorar la Experiencia de Usuario

Al darle uso a Pandalyze tras su desarrollo, también identificamos varias oportunidades para mejorar la experiencia de usuario. Como resultado, implementamos algunos cambios significativos: reorganizamos las categorías de los bloques, simplificamos los parámetros de las funciones, agregamos textos explicativos en algunos parámetros para facilitar su comprensión y un tutorial junto con nubes de diálogo en los bloques para ofrecer una mejor guía a los usuarios. A continuación, detallamos cada uno de estos cambios:

- Reorganización de categorías: habiendo incorporado los nuevos bloques y funcionalidades nombradas en la sección anterior, notamos que, aunque estas mejoras ampliaron las capacidades de la herramienta, la distribución previa de las categorías no facilitaba una navegación fluida. Esta limitación se hizo evidente al tratar de integrar los nuevos bloques, lo que nos llevó a reevaluar y reorganizar las categorías.

De la estructura original, eliminamos la categoría “Control” y renombramos “Primitivos” por “Literales”. Luego, la categoría “Pandas” fue la que sufrió mayores modificaciones. En un inicio, esta categoría contenía bloques y también estaba

dividida en dos subcategorías, “Información” y “Filtros”. Esta división quedó un poco limitada y confusa para los usuarios ya que algunos bloques eran difíciles de encontrar y no había una relación clara entre los bloques dentro de una misma categoría. Por lo tanto, para mejorar la usabilidad, decidimos reestructurarlas. Mantuvimos “Pandas” como un contenedor sin bloques directos, y creamos tres nuevas subcategorías: “Acceso a datos”, “Cálculos” y “Distribución”, que fueron definidas en la sección 5.3 junto a “Información” y “Filtros”. Esta nueva estructura nos permite tener una organización más lógica y coherente facilitando así su uso y localización.

En la **Figura 19** podemos visualizar el antes y después de estas categorías.



**Figura 19. Antes y después de la organización de las categorías**

- Simplificación de parámetros: la mayoría de los bloques utilizados en Pandalyze están basados en funciones de pandas y Plotly. Estas funciones originalmente incluyen una gran cantidad de parámetros opcionales para dar versatilidad a la operación, pero estos pueden resultar abrumadores para usuarios principiantes. Por lo tanto, para mejorar la usabilidad y focalizarnos en el objetivo principal de la funcionalidad, decidimos reducir el número de parámetros en los bloques, enfocándonos solo en aquellos que puedan resultar más relevantes y frecuentemente utilizados. A modo de ejemplo, en la **Figura 20**, mostramos como es originalmente la función “groupby” de pandas con sus parámetros “by”, “axis”, “level”, “as\_index”, “sort”, “group\_keys”, “observed” y “dropna” y cómo decidimos simplificar la función utilizando solo el parámetro “columna” que equivale al “by”.



```
DataFrame.groupby(by=None, axis=_NoDefault.no_default, Level=None, as_index=True, sort=True, group_keys=True, observed=_NoDefault.no_default, dropna=True)
```

```
data_frame = .groupby(column = )
```

Figura 20. Parámetros de pandas vs. parámetros de Pandalyze en “groupby”

- Textos explicativos en parámetros: para mejorar la comprensión, añadimos texto explicativo en los parámetros de algunos bloques para indicar claramente qué tipo de bloque debe insertarse en cada lugar. Por ejemplo, continuando con el bloque "groupby", en la **Figura 21** mostramos el diseño final donde incluimos las etiquetas "data\_frame" y "columna" para indicar que se debe utilizar un bloque de tipo "DataFrame" y uno de tipo "columna" respectivamente.

```
data_frame = .groupby(column = )
```

```
data_frame = read_csv Lagos.csv .groupby(column = data_frame = read_csv Lagos.csv Nombre )
```

Figura 21. Bloque “groupby” con etiquetas explicativas en los parámetros

- Tutorial y nubes de diálogo: como mencionamos en secciones anteriores, para que resulte más fácil e intuitivo para el usuario utilizar la aplicación, creamos un tutorial inicial descriptivo de 6 diapositivas que puede omitirse o volverse a ver cuando se desee. Además, nos pareció importante que los usuarios sepan y entiendan la función que cumple cada bloque al momento de utilizarlo. Es por esto, que decidimos incorporar nubes de diálogo informativas que pueden ser visualizadas al presionar sobre los bloques. En la **Figura 22** mostramos, a modo de ejemplo, los bloques “read\_csv”, “info”, “max” y “show” con sus respectivas nubes de diálogo.

**'Read CSV'**  
Convierte a DataFrame el CSV ingresado.

**'Info'**  
Obtiene características del DataFrame, incluyendo las columnas, tipos de datos y memoria utilizada.

**'Max'**  
Obtiene el valor máximo de cada columna del DataFrame.

**'Show'**  
Imprime en consola el gráfico generado.



Figura 22. Bloques con nubes de diálogo

## 6.2 Evaluación con estudiantes

Hicimos dos talleres con grupos diferentes de estudiantes, quienes tuvieron la oportunidad de explorar la aplicación y completar una serie de ejercicios diseñados para evaluar su comprensión. Gracias a esta experiencia, pudimos observar cómo interactuaban con Pandalyze recopilando comentarios sobre su desempeño y funcionalidad.

Las pruebas que realizamos con los estudiantes fueron realizadas en dos talleres:

1. En el primer taller, participaron 13 estudiantes de 7mo. año de la EET 9 de La Plata que se encuentran realizando la PPS (Práctica Profesional Supervisada) en la Facultad de Informática. Les dimos una guía<sup>35</sup> con las indicaciones y las consignas a realizar usando Pandalyze que debieron completar sin la intervención de la docente a cargo.
2. En el segundo encuentro, intervinieron 4 estudiantes de 5to año de la EET 1 de Berisso y 8 estudiantes de 5to año de la EET 9 de La Plata, a quienes se les dio una guía con desafíos similares a la dada en el primer taller<sup>36</sup>. Estas dos escuelas participan del proyecto "Ciencia de Datos en la escuela", aprobado en el marco de la convocatoria Extensión Universitaria 2022, número de proyecto EU71-UNLP19270.

### 6.2.1 Evaluación con los estudiantes de la PPS

Este grupo de estudiantes ya había trabajado con la librería pandas en Jupyter Notebook, con lo cual ya estaban familiarizados con las funcionalidades y los conceptos básicos de la Ciencia de Datos. La guía especificaba hacer capturas de los programas realizados con Pandalyze para resolver cada ejercicio y, además, debían desarrollar los desafíos sin ningún tipo de ayuda por parte de la profesora a cargo. Esto último nos permitió analizar si las ayudas y el tutorial de la herramienta eran suficientes para poder utilizarla.

Algunos de los ejercicios del instructivo incluían familiarizarse con la herramienta, explorar las características del archivo CSV "Lagos", operar con filtros y generar gráficos utilizando los bloques correspondientes.

Esta primera evaluación se llevó a cabo en dos jornadas, en las cuales los estudiantes pudieron completar todas las consignas dadas. Se utilizó la sala de PC del primer piso de la

---

<sup>35</sup>

[https://docs.google.com/document/d/1i6sx3wkkyb1JEpJaH-TCMEohMts9PquYO3\\_kwDMGSOs/edit#heading=h.5ot7b3rpf1e](https://docs.google.com/document/d/1i6sx3wkkyb1JEpJaH-TCMEohMts9PquYO3_kwDMGSOs/edit#heading=h.5ot7b3rpf1e)

<sup>36</sup>

[https://docs.google.com/document/d/1RCnEioTd\\_6Zj5uuDfDqH-mf3XcbXDnSgzcq41r7xoqQ/edit#heading=h.5ot7b3rpf1e](https://docs.google.com/document/d/1RCnEioTd_6Zj5uuDfDqH-mf3XcbXDnSgzcq41r7xoqQ/edit#heading=h.5ot7b3rpf1e)



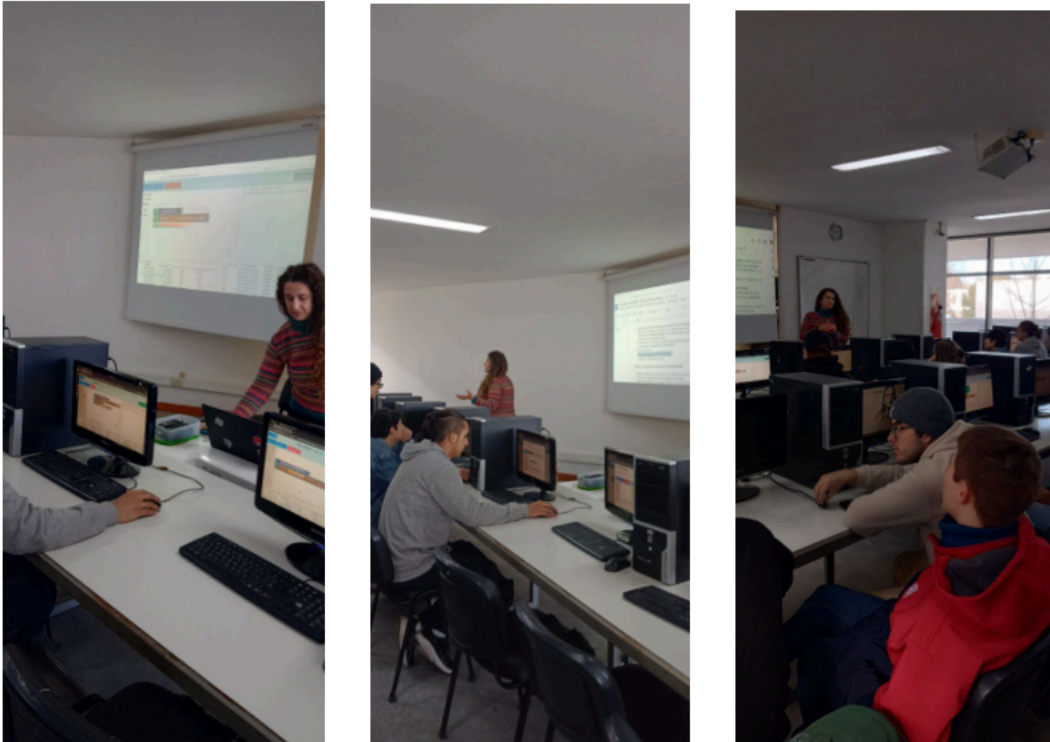


FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

Facultad de Informática para su implementación. La **Figura 23** muestra unas fotos tomadas durante el taller.



**Figura 23. Fotos realizando el primer taller**

Al finalizar la actividad, les pedimos a los estudiantes que completaran una encuesta<sup>37</sup> para evaluar su experiencia general con la aplicación. El objetivo era obtener una devolución sobre diversos aspectos de la herramienta. Entre las preguntas incluimos qué fue lo que más y lo que menos les gustó de Pandalyze, si les resultó fácil encontrar los bloques necesarios para resolver las consignas y si el tutorial les fue útil para guiarlos durante la actividad.

A modo de ejemplo, la **Figura 24** presenta tres gráficos de torta que muestran los resultados de tres de las preguntas que consideramos más relevantes:

- El primer gráfico revela que el 84.6% de los estudiantes consideró que el tutorial fue útil, mientras que el 15.4% opinó lo contrario.
- En el segundo gráfico, observamos que el 23.1% de los encuestados encontró sencillo el proceso de generación de gráficos, el 7.7% no y el 69.2% tuvo dificultades para identificar los parámetros a usar en este tipo de bloques.
- El tercer gráfico muestra que el 30.8% de los usuarios encontró todos los bloques necesarios con facilidad y el 69.2% encontró algunos fácilmente, pero tuvo

37

<https://docs.google.com/document/d/1x0ua9mP22lqfdJm Js2iLDynnNefuwLApiCqIRDUwC E/edit>





FACULTAD DE INFORMÁTICA



UNIVERSIDAD NACIONAL DE LA PLATA

dificultades con otros. Cabe destacar que ningún estudiante necesitó pedir ayuda para encontrar todos los bloques.



Figura 24. Resultados de tres preguntas de la encuesta (1er taller)

Podemos concluir, luego de analizar los resultados obtenidos de la encuesta, que a nivel general los estudiantes encontraron la aplicación bastante sencilla y más fácil que usar Python directamente. Destacaron la simplicidad e intuitividad de los bloques que facilitó la resolución de las consignas. Además, consideraron que el tutorial fue útil para resolver los ejercicios y que con práctica podrán avanzar al uso de código.

### 6.2.2 Evaluación en el marco del proyecto Ciencia de Datos en la escuela

En este taller, como en el anterior, algunos de los estudiantes incluidos ya habían realizado una actividad previa con pandas usando Jupyter Notebook.

Esta segunda evaluación se realizó el 9 de agosto de 2024 y también se utilizó la sala de PC del primer piso de la Facultad de Informática. La **Figura 25** muestra unas fotos tomadas durante el taller.

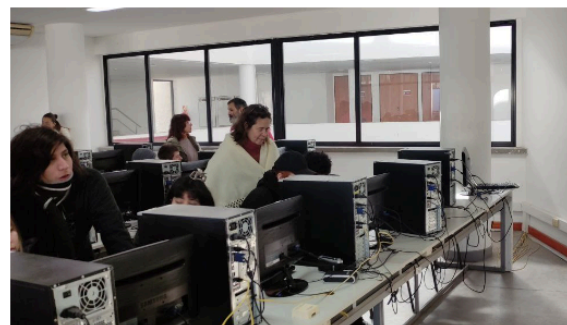
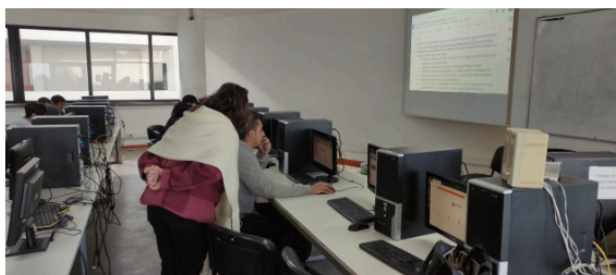


Figura 25. Fotos realizando el segundo taller

Adecuamos la guía que utilizamos en el primer taller, eliminando algunos ejercicios y las capturas de pantalla, ya que se disponía de menos tiempo. Sin embargo, este grupo de estudiantes finalizó el instructivo en la mitad del tiempo previsto, con lo cual, agregamos

varios ejercicios más en el momento para comprobar el uso del resto de las funcionalidades no incluidas en los ejercicios básicos.

Les propusimos a los estudiantes que encontrarán diferentes métodos para resolver un mismo desafío, como por ejemplo determinar el lago con mayor y menor superficie. Esta actividad les permitió profundizar en las operaciones disponibles dentro de la herramienta, explorando cómo podían llegar a la misma solución mediante distintos enfoques. Además, les permite potenciar la creatividad y la flexibilidad cognitiva, a través de la evaluación de los distintos bloques disponibles.

Para obtener los resultados de esta evaluación, reutilizamos las preguntas de la encuesta anterior. La **Figura 26** muestra los resultados de las mismas tres preguntas analizadas con el anterior grupo, facilitando su comparación.

En este segundo taller, los resultados obtenidos fueron:

- En el primer gráfico, podemos observar que el 91.7% de los estudiantes consideró que el tutorial fue útil, mientras que el 8.3% opinó lo contrario.
- En el segundo gráfico, visualizamos que el 58.3% de los encuestados encontró sencillo el proceso de generación de gráficos, el 8.3% no y el 33.3% tuvo dificultades para identificar los parámetros a usar en este tipo de bloques.
- El tercer gráfico muestra que el 33.3% de los usuarios encontró todos los bloques necesarios con facilidad y el 58.3% encontró algunos fácilmente, pero tuvo dificultades con otros. En este caso, además, el 8.3% tuvo que pedir ayuda para encontrar los bloques necesarios.



**Figura 26. Resultados de tres preguntas de la encuesta (2do taller)**

En esta segunda evaluación, los resultados fueron más satisfactorios en comparación con la primera. Esto se refleja comparando los datos presentados en la **Figura 24** con los de la **Figura 26**, donde en esta última observamos mejoras notables en los porcentajes. En particular, la mayor diferencia se destaca en el segundo gráfico de torta, que aborda la facilidad para generar gráficos en Pandalyze. El porcentaje de estudiantes que encontraron dificultades para identificar qué parámetros usar ha disminuido considerablemente.



### 6.3 Aspectos a destacar

Las dos experiencias con estudiantes nos permitieron realizar un análisis de Pandalyze, de acuerdo al uso que le dieron los destinatarios finales. A continuación, en la **Figura 27**, combinamos los resultados de ambos talleres y luego describimos los aspectos positivos y negativos que surgieron de estas dos actividades.



Figura 27. Resultados de tres preguntas de la encuesta (ambos talleres)

#### 6.3.1 Aspectos positivos

Las pruebas que realizamos han resaltado varios aspectos positivos de Pandalyze. En general, los usuarios encontraron la aplicación fácil de usar y menos compleja que el uso de código Python directo. Destacamos los siguientes puntos importantes:

- Facilidad de uso y diseño intuitivo: los estudiantes apreciaron la simplicidad y la facilidad de uso de Pandalyze. Al preguntarles qué les gustó más de la aplicación, la mayoría destacó su diseño intuitivo y la interfaz amigable. El enfoque basado en bloques no solo facilitó la comprensión de los conceptos, sino que también hizo que la resolución de ejercicios fuera un proceso accesible.
- Efectividad de las ayudas y el tutorial: consideraron que el tutorial fue de gran ayuda para la realización de las consignas y para entender cómo utilizar la aplicación. Esto nos indica que la guía es efectiva y necesaria para orientar a los usuarios en el uso de la herramienta.
- Resolución autónoma de ejercicios: pudieron completar múltiples desafíos con mínima asistencia por parte de la profesora a cargo, mostrando que Pandalyze les permite abordar y resolver problemas de manera efectiva y autónoma.

Estos resultados reflejan que nuestra aplicación cumple con los objetivos planteados en el **capítulo 1**. A través de una herramienta amigable, intuitiva y basada en bloques, los estudiantes pudieron introducirse en la Ciencia de Datos y la programación sin contar con demasiado conocimiento en esta temática.



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

### 6.3.2 Aspectos negativos

A pesar de los aspectos positivos, también surgieron áreas de mejora a partir de las evaluaciones realizadas:

- Dificultad en la identificación de parámetros: en particular, a la hora de realizar gráficos, algunos estudiantes se encontraron con dificultades para identificar los parámetros correctos que debían utilizar. Aunque el porcentaje de quienes experimentaron esta dificultad disminuyó en la segunda evaluación, sigue siendo un aspecto para mejorar la usabilidad.
- Facilidad para encontrar bloques: si bien la mayoría de los estudiantes encontró los bloques con facilidad, un porcentaje notable experimentó dificultades con ciertos bloques. Por esto, creemos que es importante continuar optimizando la disposición y organización de las categorías de los bloques garantizando encontrarlos con mayor rapidez.

Estos puntos destacan la importancia de realizar ajustes y mejoras continuas en Pandalyze para potenciar la experiencia de usuario y asegurar que la herramienta sea completamente accesible para todos los niveles de habilidad.



## Capítulo 7. Conclusiones y consideraciones finales

Este último capítulo, está dedicado a presentar las conclusiones finales derivadas del trabajo realizado y a discutir las posibles líneas de trabajo futuro, incluyendo la implementación de nuevas funcionalidades y mejoras en Pandalyze.

### 7.1 Conclusiones

Comenzamos este trabajo investigando y explorando el uso actual de la Ciencia de Datos en diversos ámbitos, incluyendo la educación, el gobierno y el área de la salud. Observamos también la relevancia y protagonismo que tiene esta disciplina en la gestión y análisis de grandes volúmenes de datos. Sin embargo, a pesar de la importancia de este campo, notamos que la enseñanza de la Ciencia de Datos en los niveles educativos iniciales se encuentra limitada y hay escasez de herramientas adecuadas para su aprendizaje.

La creciente demanda del mercado laboral en relación a la programación y Ciencia de Datos, remarca la necesidad de integrar estos conocimientos desde etapas tempranas en el ámbito educativo. Aprender a programar y manejar datos no solo es esencial para desarrollar habilidades técnicas, sino también para fomentar el pensamiento crítico y mejorar la toma de decisiones.

En respuesta a esta necesidad, desarrollamos Pandalyze, una herramienta intuitiva y amigable basada en programación en bloques y diseñada para facilitar el aprendizaje de la Ciencia de Datos a estudiantes con poca o ninguna experiencia previa. Gracias a las pruebas que realizamos con estudiantes y a las devoluciones obtenidas, podemos concluir que la aplicación ha cumplido su propósito inicial. Ellos han podido comprender rápidamente el funcionamiento de la herramienta, desenvolverse con facilidad y autonomía, utilizar los bloques para construir gráficos y ver los resultados en consola para luego analizarlos y sacar conclusiones al respecto.

### 7.2 Líneas de trabajo futuro

Como líneas de trabajo futuro, pensamos en varias opciones que podrían perfeccionar y enriquecer la experiencia del usuario y ampliar las capacidades de la aplicación:

- Sistema de usuarios y administradores: una de las principales mejoras sería la implementación de un sistema de usuarios y administradores. Esto permitiría a cada uno tener su propio perfil y guardar su progreso a lo largo del tiempo, pudiendo retomar su aprendizaje exactamente donde lo dejaron.



- Ejemplos y desafíos: la inclusión de más ejemplos para una mejor comprensión del uso de la aplicación y añadir desafíos interactivos y ejercicios harían que se pongan a prueba los conocimientos adquiridos, incentivando a los usuarios a aplicar lo que han aprendido y a resolver problemas por su cuenta.
- Crear más bloques: incorporar bloques, especialmente aquellos relacionados con las librerías pandas y Plotly, permitiría realizar un mayor análisis y visualización de los datos. Esta modificación haría a la herramienta más versátil y mejoraría las actividades que los usuarios pueden realizar. Además, añadir bloques fomentaría el procesamiento de datos más complejos y la generación de gráficos más avanzados e interactivos y, de esta manera, se podrían realizar análisis estadísticos más detallados.
- Traducción de código a bloques: aunque nosotros no lo consideramos necesario por lo explicado en el **capítulo 5**, realizar la traducción inversa de código a bloques nos parece una buena línea futura para una aplicación más completa y flexible.
- Comunidad y documentación: así como el desarrollo de esta aplicación se pudo llevar a cabo gracias a la distinta documentación existente sobre las tecnologías seleccionadas, nos parece importante crear un espacio colaborativo para los usuarios de Pandalyze. Este espacio podría incluir foros de discusión, tutoriales y documentación adicional, recursos educativos y la posibilidad de compartir proyectos y scripts entre usuarios. Esto fomentaría el apoyo entre todos y el crecimiento colectivo y de la aplicación.
- Soportar otros formatos de archivos: consideramos que otra mejora significativa podría ser la posibilidad de importar una variedad de formatos de archivos más allá de CSV. Incorporar la capacidad de manejar formatos como JSON y XML, entre otros, permitiría a los usuarios trabajar con diferentes tipos de datos y adaptarse a diversas necesidades.

Además de las anteriores líneas, todas ellas pensadas en extender el desarrollo de Pandalyze, pensamos que otra línea interesante que se podría seguir sería la de elaborar secuencias didácticas que permitan utilizar Pandalyze en distintos contextos educativos.





## Referencias Bibliográficas

- [1] Banchoff, Martin, Venosa y Hurtado (2023). *La incorporación de la Ciencia de Datos como estrategia transversal para reforzar la enseñanza de la programación en el aula*.  
<https://wicc2023.unnoba.edu.ar/2023/04/12/la-incorporacion-de-la-ciencia-de-datos-c-omo-estrategia-transversal-para-refozar-la-ensenanza-de-la-programacion-en-el-aul-a/>
- [2] Danyluk A. y Leidig. P (2021). *Computing Competencies for Undergraduate Data Science Curricula: ACM Data Science Task Force*  
<https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1008&context=cispeerpubs>
- [3] A. Ravishankar Rao, Yashvi Desai y Kavita Mishra (2019). *Data science education through education data: an end-to-end perspective*.  
<https://ieeexplore.ieee.org/document/8881970>
- [4] Faraón Llorens (2015). *Dicen por ahí que la nueva alfabetización pasa por la programación*.  
[https://rua.ua.es/dspace/bitstream/10045/49092/1/2015\\_Llorens\\_ReVision.pdf](https://rua.ua.es/dspace/bitstream/10045/49092/1/2015_Llorens_ReVision.pdf)
- [5] Mara Borchardt e Inés Roggi (2017). *Ciencias de la computación en los sistemas educativos de América Latina*.  
<https://unesdoc.unesco.org/ark:/48223/pf0000372138.locale=en>
- [6] Joel Grus (2019). *Data Science from Scratch. First Principles with Python*.  
<http://repository.universitatumigora.ac.id/2217/1802/2019%20Data%20Science%20from%20Scratch%20First%20Principles%20with%20Python%20by%20Joel%20Grus.pdf>
- [7] Sofía Martín, Claudia Banchoff, Paula Venosa y Liliana Hurtado (2024). *Ciencia de Datos en la escuela: herramientas y estrategias aplicadas en experiencias en La Plata*. Artículo aceptado para su publicación en el Congreso Tecnología en Educación y Educación en Tecnología (TEYET).  
<https://teyet2024.unimoron.edu.ar/#about-section>
- [8] United States Data Science Institute. *Role Of Major Components In Data Science Projects*.  
<https://www.usdsi.org/data-science-insights/role-of-major-components-in-data-science-projects>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

- [9] Expertos en Branded Content (2020). *Nueve industrias que aplican la Ciencia de Datos para solucionar problemas reales*.  
<https://www.xataka.com/n/nueve-industrias-que-aplican-ciencia-datos-para-solucionar-problemas-reales>
- [10] Candela García Fernández (2024). *Ética en Data Science: Desafíos y consideraciones clave*.  
<https://openwebinars.net/blog/etica-en-data-science/>
- [11] Analitika (2023). *Importancia de la ética en la ciencia de datos: análisis de sesgos y privacidad*.  
<https://analitikacentroamerica.com/importancia-de-la-etica-en-la-ciencia-de-datos-analisis-de-sesgos-y-privacidad/>
- [12] Opendatasoft. *What is open data? - Practical Guide*.  
<https://www.opendatasoft.com/en/what-is-open-data-practical-guide/>
- [13] Simplilearn (2024). *Predicting the Future of Data Science: What Lies Ahead?*  
<https://www.simplilearn.com/the-future-of-data-science-article>
- [14] Ayushi Chahal y Preeti Gulia (2019). *Machine Learning and Deep Learning*.  
[https://www.researchgate.net/publication/364097061\\_Machine\\_Learning\\_and\\_Deep\\_Learning](https://www.researchgate.net/publication/364097061_Machine_Learning_and_Deep_Learning)
- [15] AutoML.org (2024). *AutoML*.  
<https://www.automl.org/automl/>
- [16] Program.AR. *Vocaciones TIC*.  
<https://program.ar/vocaciones-tic/>
- [17] Consejo Federal de Educación (2015). *“Resolución del Consejo Federal de Educación de la República Argentina N° 263/15”*.  
[http://www.bnm.me.gov.ar/giga1/normas/RCFE\\_263-15.pdf](http://www.bnm.me.gov.ar/giga1/normas/RCFE_263-15.pdf)
- [18] Gustavo Cucuzza (2024). *La Informática como materia en las escuelas secundarias de Argentina*.  
<https://adicra.org.ar/licmensecu/>
- [19] Scratch Foundation (2022). *Growing a Global Creative Learning Movement. 2022 Annual Report*.  
<https://www.scratchfoundation.org/annualreport>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

- [20] Educ.ar (2017). *Alejandra Valdez: «Con Scratch busco despertar el pensamiento lógico»*  
<https://www.educ.ar/recursos/131974/alejandra-valdez-con-scratch-busco-despertar-el-pensamiento-logico2?from=150002>
- [21] Nahir Di Tullio (2017). «*Programa2 con Scratch*»  
<https://www.educ.ar/recursos/131973/programa2-con-scracth>
- [22] Carlos Gregorio Artilles Fontales para INTEF (2019). *Pilas Bloques Aprende a programar jugando*.  
<https://intef.es/wp-content/uploads/2019/06/Pilas-Bloques-1.pdf>
- [23] Sofía Martín, Claudia Banchoff, Paula Venosa y Lilita Hurtado (2022). *Ciencias de datos en escuelas secundarias*.  
[http://sedici.unlp.edu.ar/bitstream/handle/10915/139959/Documento\\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/139959/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y)
- [24] Departamento de computación - UBA. *Computación: una carrera que aumenta su demanda laboral y necesita cada vez más graduados*.  
<https://www.dc.uba.ar/computacion-una-carrera-que-aumenta-su-demanda-laboral-y-necesita-cada-vez-mas-graduados/>
- [25] Departamento de computación - UBA. *Computación: entre las carreras más demandadas de la UBA*.  
<https://www.dc.uba.ar/computacion-entre-las-carreras-mas-demandadas-de-la-uba/>
- [26] Marcón Paula, Duggan Tomás, Sofía Martín y Claudia Banchoff (2024). *Herramientas Informáticas utilizadas para Introducir Aspectos Básicos de Ciencia de Datos en la Escuela Secundaria*. Artículo aceptado para su publicación en el Congreso Argentino de Ciencias de la Computación (CACIC).  
<https://cacic2024.info.unlp.edu.ar/>
- [27] Marcón Paula, Duggan Tomás, Sofía Martín y Claudia Banchoff (2024). *“Pandalyze: una introducción al análisis de datos programado con bloques.”* Artículo aceptado para su publicación en las Jornadas Argentinas de Didáctica de las Ciencias de la Computación (JADICC).  
<https://jadicc2024.dc.exa.unrc.edu.ar/>
- [28] Fundación Sadosky. *Pilas Bloques es una aplicación para enseñar y aprender a programar*.  
<https://pilasbloques.program.ar/acerca-de-pilas-bloques/>



FACULTAD DE INFORMÁTICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

- [29] Codigital (2024). *Los Mejores Frameworks para Utilizar en 2024*.  
<https://codigital.ec/frameworks-2024/>
- [30] Denis Mashutin (2023). *Django vs Flask: Which is the Best Python Web Framework?*  
<https://blog.jetbrains.com/pycharm/2023/11/django-vs-flask-which-is-the-best-python-web-framework/>
- [31] PostgreSQL (2024). *About PostgreSQL*.  
<https://www.postgresql.org/about/>
- [32] pandas (2024). *pandas.DataFrame*.  
<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>
- [33] pandas (2024). *pandas.Series*.  
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.html>