



*Universitat
Abat Oliba CEU*

Ciencia de datos en el aula: proyecto de matemáticas y programación

TRABAJO FIN DE MÁSTER

Autor: Natàlia Conesa Parras
Tutor: Sergio Gómez Moyano
Máster Universitario en Formación del Profesorado de Educación Secundaria
Obligatoria y Bachillerato, Formación Profesional y Enseñanzas de Idiomas
Año: 2023

DECLARACIÓN

El que suscribe declara que el material de este documento, que ahora presento, es fruto de mi propio trabajo. Cualquier ayuda recibida de otros ha sido citada y reconocida dentro de este documento. Hago esta declaración en el conocimiento de que un incumplimiento de las normas relativas a la presentación de trabajos puede llevar a graves consecuencias. Soy consciente de que el documento no será aceptado a menos que esta declaración haya sido entregada junto al mismo.

Firma:..........
Natàlia CONESA PARRAS

*Poco importa quién llegue primero a una idea,
lo que es significativo es hasta dónde puede llegar esa idea,*

SOPHIE GERMAIN

Resumen

En este Trabajo de Final de Máster se presenta el diseño de una situación de aprendizaje para la enseñanza del análisis de datos en el área de matemáticas y programación. La propuesta se enfoca en alumnos de primer año de Bachillerato y se extiende a lo largo de un trimestre. En el proyecto se emplea la metodología de Aprendizaje Basado en Proyectos y se apoya en el lenguaje de programación Python, en concreto, la librería Pandas para tratar y mostrar datos y aplicar conceptos de estadística básica.

Resum

En aquest Treball de Final de Màster es presenta el disseny d'una situació d'aprenentatge per a l'ensenyament de l'anàlisi de dades en l'àrea de matemàtiques i programació. La proposta s'enfoca en alumnes de primer any de Batxillerat i s'estén al llarg d'un trimestre. En el projecte s'utilitza la metodologia d'Aprenentatge Basat en Projectes i es recolza en el llenguatge de programació Python, en concret, la llibreria Pandas per a tractar i mostrar dades i aplicar conceptes d'estadística bàsica.

Abstract

This Master's Thesis presents the design of a learning situation for teaching data analysis in the areas of mathematics and programming. The proposal is aimed at first-year high-school students and extends over one term. The project employs the methodology of Project-Based Learning and relies on the Python programming language, specifically the Pandas library, to handle and display data and apply basic statistical concepts.

Palabras claves / Paraules clau / Keywords

Trabajo Fin de Máster / Treball Final de Màster / Master's Thesis — Investigación / Investigació / Research — Universidad / Universitat / University — Matemáticas / Matemàtiques / Mathematics — Programación / Programació / Programming — Análisis de datos / Anàlisi de dades — Situación de aprendizaje / Situació d'aprenentatge / Learning situation — Aprendizaje Basado en Proyectos / Aprenentatge Basat en Projectes / Project-based learning — Estadística / Estadística / Statistics

Índice

Introducción	11
1. Marco teórico	15
1.1. La programación en la educación	15
1.1.1. Beneficios de la programación en la educación	15
1.1.2. Conceptos básicos de programación que se deben aprender	16
1.1.3. Dificultades de los estudiantes frente al aprendizaje con programación	17
1.1.4. Cómo introducir a los estudiantes en el aprendizaje de la programación	18
1.2. Aprendizaje basado en proyectos	19
1.2.1. Características del Aprendizaje Basado en Proyectos	19
1.2.2. Elementos esenciales para Aprendizaje Basado en Proyectos	21
1.2.3. Beneficios e inconvenientes del Aprendizaje Basado en Proyectos	24
1.3. La estadística en la ciencia de datos	26
1.3.1. Elementos estadísticos utilizados en el análisis de datos	27
1.3.2. La carrera de moda en el siglo XXI	28
2. Contexto de la propuesta de intervención y análisis de necesidades	30
2.1. Contexto de la propuesta de intervención	30
2.2. Análisis de necesidades	30
2.3. Propuesta de intervención	31
3. Conclusiones	47
4. Bibliografía	51

Índice de anexos

Anexo I. Rutina de pensamiento: “Los datos son el nuevo petróleo”	53
Anexo II. Temas para explorar con Google Trends	55
Anexo III. Guía para acceder al perfil de anuncios de Google	56
Anexo IV. Google Colab: ¡Analiza y visualiza datos!	57
Anexo V. Google Colab: Librería Pandas. Series	59
Anexo VI. Google Colab: Librería Pandas. DataFrame	64
Anexo VII. Presentación: “Una buena visualización”	73
Anexo VIII. Google Colab: Visualización de datos	78
Anexo IX. Google Colab: Estadística con DataFrames	88
Anexo X. Google Colab: Un análisis completo	92
Anexo XI. Descripción de la actividad: “¿Qué hemos aprendido?	100
Anexo XII. Rúbrica para la actividad: “¿Qué hemos aprendido?	101
Anexo XIII. Descripción de la actividad: “Analizando Datos”	102
Anexo XIV. Descripción de la actividad: “¿Qué historia nos cuentan los datos?.....	103
Anexo XV. Rúbrica docente para la actividad: “¿Qué historia nos cuentan los datos?”	104
Anexo XVI. Rúbrica coevaluación para la actividad: “¿Qué historia nos cuentan los datos?”	105
Anexo XVII. Rúbrica coevaluación y autoevaluación	106

Índice de Figuras

Figura 1. Adaptación del “cono del aprendizaje” de Edgar Dale.	20
Figura 2. Diferencias entre Proyecto y Aprendizaje Basado en Proyectos	21
Figura 3. Elementos esenciales que debe incluir un buen proyecto.	23
Figura 4. Beneficios del Aprendizaje Basado en Proyectos.	25
Figura 5. Inconvenientes del Aprendizaje Basado en Proyectos.	26
Figura 6. Evolución del interés del término ciencia de datos en Google	28

Introducción

A lo largo de los años, el enfoque de la educación ha cambiado significativamente. Desde la primera ley, la Ley Moyano de 1857, que garantizaba la educación obligatoria y gratuita hasta los 12 años hasta la actualidad, se han realizado cambios muy relevantes. Uno de los autores que más ha contribuido a este cambio es Jean Piaget, conocido por su teoría constructivista del desarrollo cognitivo de los niños. Piaget ofreció una nueva perspectiva sobre cómo aprenden los niños y cómo se desarrolla su conocimiento, enfatizando el hecho de que los niños construyen el conocimiento a partir de sus propias experiencias y que el aprendizaje debe estar centrado en el alumno y no en el profesor.

Este enfoque centrado en el alumno se ha vuelto tan relevante que los métodos de enseñanza actuales se centran en dar a los alumnos un papel más participativo en su propio aprendizaje. Actualmente, se utilizan varios métodos para este propósito, como el aprendizaje basado en proyectos, el aprendizaje basado en juegos o la clase invertida.

Si nos centramos en la didáctica de las matemáticas, la programación ha tenido un gran impacto. Para la mayoría de los estudiantes, los métodos formales de las matemáticas siguen siendo misteriosos, artificiales, poco motivadores y muy relacionados con el pensamiento intuitivo (Feurzeig et al., 1969). La introducción de herramientas y tecnologías informáticas ha permitido una mayor interacción y participación del estudiante en su proceso de aprendizaje, consiguiendo también una mayor personalización de la educación. Sin embargo, la relación de los niños con las matemáticas sigue siendo profundamente desconcertante. A menudo, los prejuicios populares sobre las matemáticas dificultan la comprensión de la materia (Feurzeig et al., 1969). Otro aspecto señalado por Feurzeig et al. (1969), es la importancia de la programación en las matemáticas y cómo contribuyen en la enseñanza de estas. Destacaremos tres argumentos: la facilitación en la adquisición de pensamiento y expresión rigurosa, la ayuda al estudiante a luchar por la autoconciencia y la alfabetización sobre el proceso de resolución de problemas y el uso del ordenador como un laboratorio matemático para fomentar un enfoque experimental hacia la resolución de problemas. Más recientemente, Mulder (2002, citado por Saeli et al., 2011), destaca que, en los Países Bajos, la informática se ha definido como una disciplina de nueva generación, porque se vincula con las Matemáticas, la Física, la Ingeniería, la Lingüística, la Filosofía, la Psicología, la Economía, los Negocios y las Ciencias Sociales en general. Soloway (1993) argumenta, tras considerar las opiniones de distintos expertos, que la programación debería estar más integrada en

la educación para empoderar a los estudiantes y ayudarlos a desarrollar nuevas formas de ver el mundo. Apoda a la programación como el nuevo latín, la base para aprender muchos lenguajes. Aunque sea tan relevante, diversos especialistas en educación (Kurland et al., 1989; Papert, 1980, citados por Saeli et al., 2011), explican que la programación es una habilidad que se considera difícil de aprender e incluso después de dos años de instrucción, el nivel de comprensión de la programación es bajo, no obstante, si se apoya en estrategias y herramientas didácticas adecuadas, los alumnos pueden llegar a dominarlo hasta cierto punto.

En la actualidad, la programación y las matemáticas están muy unidas y son esenciales para la ciencia de datos. El auge de la ciencia de datos como campo de estudio es reciente, a partir de la década de 1960, cuando surgió la necesidad de analizar grandes cantidades de datos resultantes de fenómenos naturales, transacciones comerciales, etc. El estadístico estadounidense John W. Tukey fue uno de los pioneros en la utilización de las estadísticas y la computación para analizar datos.

Durante mucho tiempo pensé que era un estadístico interesado en inferencias de lo particular a lo general. Pero a medida que observé la evolución de las estadísticas matemáticas, tuve motivos para preguntarme y dudar (...) Llegué a sentir que mi interés central está en el análisis de datos... El análisis de datos, y las partes de las estadísticas que se adhieren a él, deben (...) asumir las características de la ciencia en lugar de las matemáticas (...) el análisis de datos es intrínsecamente una ciencia empírica. (W. Tukey, 1962, citado por Santamaría, 2021)

Estas declaraciones fueron las primeras en abordar el desarrollo de la estadística matemática como ciencia de datos. Sin embargo, el término "ciencia de datos" en sí mismo fue acuñado por el científico danés Peter Naur en 1974, quien es conocido por su trabajo en las ciencias computacionales y recibió el Premio Turing en 2005. (Santamaría, 2021)

En los últimos años, el análisis de datos ha avanzado significativamente debido al avance tecnológico y la mayor disponibilidad de datos. Las organizaciones pueden recopilar y analizar grandes cantidades de información para tomar mejores decisiones. Además, los métodos y herramientas utilizados para analizar los datos han mejorado, volviéndose más potentes y sofisticados, lo que permite identificar patrones y tendencias más complejas.

El crecimiento de la ciencia de datos es tan importante que actualmente existen grados universitarios completos dedicados a ella. El alto interés de estos programas de grado refleja la creciente necesidad de científicos de datos.

La estadística juega un papel fundamental en el análisis de datos: permite resumir y describir un conjunto de datos, identificar patrones y tendencias, hacer predicciones y tomar decisiones informadas basadas en los datos. Proporciona las herramientas necesarias para transformar los datos en información útil y valiosa. Los estadísticos utilizan una variedad de herramientas y técnicas para analizar datos, como visualización de datos, regresión y análisis de varianza. La estadística también ayuda a diseñar experimentos y encuestas para recopilar datos y garantizar la validez y confiabilidad de los resultados.

Batanero (2000) argumenta que el interés en la enseñanza de la estadística en las matemáticas se debe al rápido crecimiento de la estadística como ciencia y su utilidad en la investigación, la técnica y la vida profesional debido a los avances tecnológicos y de comunicación. También destaca que los currículos de educación primaria y secundaria incorporan la estadística, pero todavía hay escasez de profesores que enseñan este tema. Prácticamente, no existe una preparación especial de los docentes en esta área.

En este trabajo, se quiere diseñar una asignatura trimestral que ayudará a los estudiantes a convertirse en analistas de datos mediante un proyecto de programación. En este proceso, el alumnado ha de ser capaz de aprender de manera progresiva los conceptos estadísticos necesarios y aplicará los conocimientos adquiridos al análisis de datos que le resultan familiares, como son los de su propia localidad. Usaremos datos locales para lograr nuestro segundo objetivo: motivar a los estudiantes a obtener un aprendizaje significativo y duradero en el análisis de datos. Al finalizar el proyecto, los estudiantes deberán ser capaces de preparar y entender los datos, extraer información relevante de ellos, responder a preguntas específicas sobre los datos y mostrar los resultados de una manera entendible y atractiva para presentarlos delante de la audiencia.

1. Marco teórico

1.1. La programación en la educación

En el ámbito educativo se están empezando a introducir competencias relacionadas con los lenguajes de programación y la programación en el aula. Los estudiantes de nueva generación están inmersos en una era tecnológica, rodeados de instrumentos relacionados con la informática y posiblemente desarrollarán un trabajo que aún no se ha inventado.

Es posible aprender desde edades muy tempranas, pero es importante tener en cuenta que los ejercicios, métodos y objetivos deben adaptarse a los distintos niveles educativos.

Se puede usar la programación para solucionar problemas en matemáticas, lengua, ciencia, ... Aprender a programar enseña a pensar, fomenta el pensamiento crítico, la lógica y la creatividad. En este trabajo trabajaremos la programación en la educación secundaria en el ámbito de las matemáticas.

1.1.1. Beneficios de la programación en la educación

Para la mayoría de los estudiantes, los métodos formales de las matemáticas son misteriosos, artificiales, poco motivadores y con un pensamiento poco intuitivo.

Una enseñanza adecuada con un lenguaje de programación adecuado puede contribuir a la educación matemática de varias maneras.

La programación tiene múltiples beneficios para la educación matemática si se enseña de manera adecuada, Feurzeig et al. (1969) hacen un listado de algunos de ellos:

1. Facilita la adquisición de pensamiento y ayuda a adquirir una expresión más rigurosa. El alumnado tiene la necesidad de entender bien los procedimientos para transmitir la información al ordenador.
2. Impulsa una mejor comprensión a diversos conceptos de álgebra, como las variables o las funciones.
3. Proporciona herramientas mentales o estrategias que se utilizan para abordar problemas complejos y llegar a soluciones creativas y elaboradas. La forma concreta del programa y el aspecto interactivo de la máquina permiten identificar la detección de errores como una actividad definida, constructiva y planificable.

4. Integra en el conocimiento del alumno pequeños problemas que van aumentando gradualmente el nivel de dificultad. Las funciones, por ejemplo, se vuelven estrategias familiares que se usan para propósitos reales. Estas funciones se usan como bloques de construcción para construir funciones más complejas que a menudo son elementos de construcciones aún más grandes, muy similar a la forma en que los matemáticos usan proposiciones para probar teoremas y usan estos teoremas para probar teoremas más complejos.
5. Ayuda al estudiante a desarrollar confianza en sí mismo y habilidades en el proceso de resolución de problemas. Los estudiantes de secundaria no tienen la costumbre de discutir los problemas y disponen del lenguaje necesario para hacerlo. Los programas son más discutibles que las actividades matemáticas tradicionales: pueden indicar su estructura, su desarrollo, la relación con el problema original, ...
6. Otorga un enfoque experimental a la resolución de problemas usando el ordenador como laboratorio matemático. Un lenguaje de programación puede usarse como instrumento para controlar una realidad exterior. Los estudiantes aprenden a escribir programas por experiencia y experimentación.
7. Relaciona las matemáticas con otras materias, mostrando así la interdisciplinariedad y la conexión entre ellas.

1.1.2. Conceptos básicos de programación que se deben aprender

Saeli et al. (2011) hacen un análisis sobre cómo sería el currículo ideal de programación combinando distintos currículos.

Hay dos tipos de conocimientos en la programación: la generación de programas y la comprensión de programas (Van Merriënboer y Krammer, 1987; Robins et al., 2003; Mannila, 2007, citados por Saeli et al., 2011). En la generación de programas, los estudiantes deben aprender el proceso de resolución de problemas, la reflexión sobre este proceso y el desarrollo de formas algorítmicas de pensamiento. Para la comprensión del programa es importante enseñar cómo se crean y entienden los programas.

En el proceso de aprender a programar, se identifican tres aspectos principales: datos, instrucciones y sintaxis. Los datos se refieren a los conceptos de variables y tipos de datos. En cuanto a las instrucciones, se necesita comprender las estructuras de control y subrutinas para la programación de procedimientos. Por último, la sintaxis

denota el grupo de reglas que determinan qué está permitido y qué no dentro de un lenguaje de programación (Govender, 2006, citado por Saeli et al., 2011).

Sin embargo, esta clasificación no tiene en cuenta dos aspectos clave para el diseño y la comprensión programas: la modularidad, que consiste en la capacidad de dividir un programa en partes más pequeñas y manejables, y la abstracción de la programación, que implica ocultar detalles innecesarios. Abelson y Sussman (1996, citados por Saeli et al., 2011), distinguen tres aspectos principales: expresiones primitivas, representan los objetos más simples tratados por el lenguaje; medios de combinación, a partir de elementos simples se construyen elementos compuestos; y medios de abstracción, elementos compuestos pueden ser nombrados y manipulados como unidades. Al usar estos tres mecanismos entre sí, se pueden crear programas complejos a partir de otros más simples.

Saeli et al. (2011) destacan otro aspecto fundamental de la programación: la semántica del programa, es decir, su significado y funcionalidad. Un programa se considera semánticamente correcto si es capaz de realizar la tarea para la que fue diseñado. Programas escritos con diferente sintaxis pueden realizar la misma tarea.

1.1.3. Dificultades de los estudiantes frente al aprendizaje con programación

Saeli et al. (2011) explican los problemas más comunes que tienen los estudiantes mientras aprenden a programar. Lo hace primeramente a partir de la tesis de DuBoulay (1989, p. 283-284), quién identifica cinco tipos de dificultades:

- Orientación: saber para qué sirve la programación y cuáles son los beneficios de aprender a programar.
- Comprender las propiedades conceptuales de la máquina y saber diferenciar entre el concepto abstracto y el físico de la máquina que se está queriendo controlar. Saber diferenciar entre la máquina nocional y la física.
- Notación: incluye los problemas de aspectos de los diversos lenguajes formales como la sintaxis y la semántica.
- Estructuras: comprendiendo los esquemas o planes que se pueden utilizar para alcanzar objetivos de pequeña escala (p. ej., utilizando un bucle).
- Dominar la funcionalidad de la programación, aprender la habilidad de especificar, desarrollar, probar y depurar un programa.

Desde el punto de vista de la relación estudiante-ordenador, Pea (1986, citado por Saeli et al., 2011), distingue tres tipos de errores conceptuales:

- El error de paralelismo, diferentes líneas de código en un programa se ejecutan simultáneamente o en paralelo, lo que puede generar resultados inesperados o errores de ejecución del programa.
- La intencionalidad, los estudiantes deben comprender que los ordenadores no pueden ir más allá de la información proporcionada en las líneas de código que se ejecutan. Los estudiantes suelen atribuir algún tipo de inteligencia a los ordenadores, lo que puede generar confusión o errores de programación.
- El egocentrismo, la tendencia de los estudiantes a sobrestimar su propio conocimiento y comprensión del programa que están desarrollando.

Otra dificultad que señalan Saeli et al. (2011) es que los alumnos tienden a centrarse en puntos de vista locales en lugar de una visión global del problema. Es posible que los estudiantes no siempre puedan traducir intuiciones mentales de manera comunicativa, o puede que no entiendan la semántica de un programa. La semántica se considera un aspecto problemático de la programación, porque los estudiantes deben integrar diferentes partes de un programa, como variables, expresiones, operadores, estructuras de control, objetos y métodos para crear una solución funcional.

1.1.4. Cómo introducir a los estudiantes en el aprendizaje de la programación

En la enseñanza de la programación queremos evitar las dificultades y motivar a los estudiantes de una manera efectiva y atractiva. Es difícil concluir cuál es el mejor método, ya que en un aula es donde mayor diversidad se puede encontrar.

Saeli et al. (2011) hacen mención a la importancia de un lenguaje de programación simple, con pocos errores de sintaxis, por ejemplo Python. También explica la importancia del uso de ejemplos prácticos, se debe llevar a los estudiantes a escribir primero programas simples y luego combinar las soluciones simples para obtener soluciones a problemas más complicados.

Para lograr el pensamiento algorítmico, los estudiantes deben resolver muchos problemas. Sin embargo, los estudiantes no logran traducir su razonamiento correcto en un conjunto de instrucciones para la máquina.

En el artículo de Saeli et al. (2011), se explica una cadena ideal para aprender a programar, desarrollada por Linn y Dalbey (1989, p. 58-62). Esta cadena va desde la comprensión del programa y termina con la generación del programa. La cadena tiene tres eslabones principales:

1. Características de un solo idioma, debería comenzar con la comprensión de las características del lenguaje, conocimiento que se puede evaluar pidiendo a los estudiantes que reformulen o cambien una característica del lenguaje en un programa para que el programa haga algo ligeramente diferente.
2. Habilidades de diseño, grupo de técnicas utilizadas para combinar características del lenguaje para formar un programa. Repetir patrones y reformular procedimientos ya conocidos.
3. Habilidades generales de resolución de problemas, se pueden evaluar pidiendo a los estudiantes que resuelvan problemas utilizando un sistema formal desconocido, como un nuevo lenguaje de programación.

1.2. Aprendizaje basado en proyectos

El Aprendizaje Basado en Proyectos, ABP, se define como una metodología activa¹ en la que los estudiantes, en grupo, se enfrentan a la solución de un problema de la vida real. A través de la investigación y mediante preguntas, afrontan soluciones a problemas reales y aportan ideas que aportarán soluciones al crearlas. El alumnado, mediante esta metodología busca, elige, discute, aplica, corrige y ensaya, hace muchísimo más que entender y memorizar.

1.2.1. Características del Aprendizaje Basado en Proyectos

Esta estrategia, junto con otras dentro del ámbito del aprendizaje activo, como el aprendizaje basado en tareas, el aprendizaje basado en problemas, el aprendizaje por descubrimiento o el aprendizaje basado en retos, tienen algunas características muy importantes:

- El conocimiento no es una posesión del docente que se transmite al alumno, sino que es el resultado de un proceso de trabajo entre el alumno y el docente en el que se formulan preguntas, se busca información y se procesa esta información para llegar a una conclusión.

¹ Estrategia de aprendizaje enfocada en el estudiante y en lograr su capacitación en una determinada disciplina a través de un proceso activo y constructivo.

- El papel del alumno no se limita a escuchar activamente, sino que también se espera que participe activamente en procesos como el reconocimiento de problemas, la priorización, la recogida de información, la comprensión e interpretación de datos o el planteamiento de conclusiones.
- El papel del docente va más allá de la transmisión de contenidos. La función principal del docente es crear una situación de aprendizaje que le permita a los estudiantes desarrollar un proyecto.

La Pirámide de Aprendizaje de Edgar Dale, véase la Figura 1, nos indica que nuestro aprendizaje es más efectivo cuando participamos activamente e integramos diferentes canales en el procesamiento de información.

Figura 1

Adaptación del “cono del aprendizaje” de Edgar Dale.



Nota. La figura representa una adaptación del cono de aprendizaje de Edgar Dale. Se trata de una representación gráfica de los diferentes modos de aprendizaje y su eficacia relativa. Fuente: Fernández Solo de Zaldívar (s.f.).

Es importante entender que no es lo mismo trabajar con proyectos que trabajar por proyectos. El ABP está conectado a una situación real, el 'producto' que creará el

estudiante variará de un grupo de trabajo a otro y, en el proceso, los estudiantes aprenderán lo que necesitan y quieren saber. El ABP permite a los estudiantes jugar, experimentar, usar simulaciones y trabajar con amigos para buscar conocimiento.

Fuentes (2018) recoge en una tabla, véase la Figura 2, las diferencias entre las características de un proyecto y el ABP.

Figura 2

Diferencias entre Proyecto y Aprendizaje Basado en Proyectos

Proyecto	Aprendizaje basado en proyecto (ABP)
Son proyectos cerrados, suelen tener los mismos objetivos o metas.	Son proyectos abiertos, las elecciones de los alumnos determinan los resultados y, así se da libertad para que determinen libremente (asesorados) su camino.
No se pueden utilizar en la vida real, para resolver problemas reales.	Pueden dar lugar a situaciones aplicables en situaciones reales.
No son particularmente relevantes para la vida de los alumnos.	Son relevantes para los alumnos en su presente y/o futuro.
No tiene ninguna relación con el trabajo en la vida real.	Es igual o semejante al trabajo en la vida real.
No incluye escenarios ni información de fondo pueden estar basados en sucesos que ya hayan sido resueltos.	El escenario o la simulación es real, o si es ficticia es realista y suficientemente flexible a lo largo del tiempo.
Se basan más en el uso de una herramienta concreta, que en una pregunta a responder.	Los alumnos eligen las herramientas de acuerdo al proyecto, las cuales les darán bases para el futuro laboral.
Se entregan resultados para ser evaluados	Se presentan los resultados a toda la clase o a la escuela.
Siempre es lo mismo	Cada día es diferente

Nota. La figura representa una comparación entre las características de un proyecto y el ABP. Fuente: Fuentes (2018).

1.2.2. Elementos esenciales para Aprendizaje Basado en Proyectos

Todo buen proyecto debe cumplir dos criterios principales: debe ser importante para los alumnos, deben percibirlo como algo importante que quieren hacer y debe ser significativo, debe tener un propósito educativo.

Trujillo (2015) recoge los ocho elementos esenciales que debe incluir un buen proyecto:

1. Contenido significativo

Los docentes deben planificar el proyecto de modo que se centre en competencias específicas; debe reflejar lo que se considera importante en el currículo. Y los estudiantes deben encontrarlo significativo, es decir, real y relacionado con su entorno e intereses.

2. Necesidad de saber

Los docentes deben abrir el proyecto con una actividad que el estudiante haga preguntas al profesor (y no al revés). El estudiante ha de querer saber más, se ha de generar una necesidad de saber.

3. Una pregunta que dirija la investigación

Se ha de iniciar el proyecto con una buena pregunta, ha de comunicar claramente la esencia del proyecto. Debe ser provocadora, abierta y exigente, y estar relacionada con el temario que se quiere enseñar.

4. Voz y voto para los alumnos

Es importante que los estudiantes tengan capacidad de elección en el proyecto.

5. Competencias del siglo XXI

Un buen proyecto dará a los estudiantes la oportunidad de practicar y así adquirir habilidades demandadas en el siglo XXI: demostrar pensamiento crítico, comunicarse efectivamente, usar tecnología y hacer trabajo en grupo.

6. Investigación lleva a la innovación

Es importante crear un ambiente en clase que promueva a los estudiantes a hacer nuevas preguntas, formular hipótesis y estar abiertos a nuevas perspectivas.

7. Evaluación, realimentación y revisión

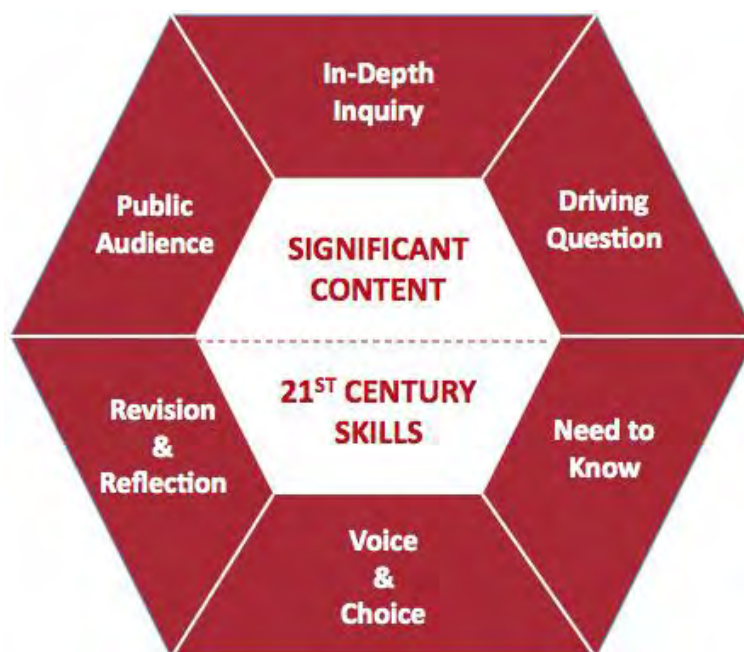
La evaluación a lo largo del proyecto es muy importante. Los docentes deben guiar a los estudiantes para que hagan críticas constructivas y aprendan que la calidad del trabajo no siempre sale a la primera.

8. Presentación el producto final ante una audiencia

Los resultados del proyecto deben ser presentados a otras clases, presidentes de clase, padres, ... Tiene más sentido que trabajar para una audiencia real que para un examen. Es importante que se sientan orgullosos del trabajo han hecho.

Figura 3

Elementos esenciales que debe incluir un buen proyecto.



Nota: La figura representa los ocho elementos esenciales que debe incluir un buen proyecto.

Fuente: Trujillo (2015, p. 8)

Estos elementos los podemos dividir en tres etapas:

Primera etapa. El docente hace la pregunta inicial que guiará todo el proyecto, por lo que debe ser una pregunta relacionada con lo que se aprenderá y lo que involucrará a los estudiantes.

Segunda etapa. Los alumnos deberán buscar toda la información que consideren relevante para la elaboración del producto final, han de poder responder la pregunta original junto a las que les han ido surgiendo.

Tercera etapa. Los grupos deberán preparar una exposición de su producto final para exponerla frente a un público con el fin que el trabajo tenga un valor real.

1.2.3. Beneficios e inconvenientes del Aprendizaje Basado en Proyectos

Existen numerosos artículos que destacan los beneficios del ABP para los estudiantes (De Miguel, s.f.; *Aprendizaje Basado En Proyectos: Características, Beneficios, Etc.*; 2021; Orientación Anduja, 2015). A continuación, destacaremos los más relevantes (ver Figura 4).

El ABP es una metodología que permite a los estudiantes involucrarse profundamente con el contenido, lo que genera un aprendizaje altamente significativo y permanente a largo plazo. Además, desarrolla y mejora el uso de habilidades de pensamiento crítico, creativo y social. El aprendizaje cooperativo es una parte fundamental del ABP, posibilita al alumnado a trabajar en grupos, resolver problemas y tomar decisiones más efectivas.

También destacan que esta metodología es más motivadora para los estudiantes, ya que les permite trabajar en proyectos que les interesan. Asimismo, son capaces de integrar los conocimientos aprendidos en la vida real, lo que les permite desarrollar habilidades y competencias del siglo XXI necesarias en la vida cotidiana. El ABP promueve el aprendizaje autónomo y el espíritu emprendedor. Mientras trabajan en un proyecto, los estudiantes tienen la oportunidad de decidir qué información necesitan para su proyecto y establecer relaciones entre conceptos. Seleccionan, evalúan y utilizan la información de manera crítica, y aprenden a manejar una gran cantidad de información. Por último, este método promueve la diversidad, porque estimula tanto a los estudiantes con problemas de aprendizaje como a los alumnos más avanzados.

Figura 4

Beneficios del Aprendizaje Basado en Proyectos.



Nota: La figura representa diversos beneficios del Aprendizaje Basado en Proyectos. Fuente: Elaboración propia.

Si bien el ABP ofrece muchos beneficios, es importante considerar que también existen ciertos inconvenientes que deben ser tenidos en cuenta. Sánchez (2017), recoge algunos de ellos (ver Figura 5).

En primer lugar, se debe enfatizar que el ABP es un proceso complejo que puede ser difícil de aplicar, especialmente en las primeras etapas, como ocurre con cualquier tipo de metodología. Es muy importante que los docentes estén debidamente formados y preparados. Por otro lado, a algunos docentes les puede resultar difícil evaluar a los estudiantes, especialmente en el uso de las tecnologías de la información y la comunicación (TIC).

De manera similar, los estudiantes también pueden enfrentarse a algunas dificultades cuando trabajan con el ABP, sobre todo en lo que respecta a la organización y gestión del tiempo, el manejo de la información y la colaboración en equipo.

Figura 5

Inconvenientes del Aprendizaje Basado en Proyectos.



Nota: La figura representa distintos inconvenientes del Aprendizaje Basado en Proyectos. Fuente: Elaboración propia.

1.3. La estadística en la ciencia de datos

El desarrollo de la ciencia y la tecnología en la actualidad nos da acceso a cantidades enormes de datos, ya que con la ayuda de sensores y dispositivos móviles podemos registrar y procesar una gran cantidad de información antes desconocida. Toda esta red de información es un fenómeno conocido como *big data*, que requiere de métodos avanzados capaces de analizar estos datos en toda su complejidad para poder relacionarse. La ciencia de datos quiere desarrollar modelos que nos permitan procesar grandes cantidades de información y, a través de la programación informática, realizar procedimientos analíticos complejos para convertirlos en lógica.

La ciencia de datos es una disciplina que, gracias a una combinación de modelos matemáticos y estadísticos, programación informática y técnicas de visualización de datos, maximiza el uso de los datos para ayudar en la toma de decisiones.

El procesamiento de datos implica varios pasos:

1. Definir qué pregunta se quiere responder.

Al trabajar con datos, es importante tener claro qué se quiere estudiar. Sin un objetivo claro, el análisis de datos puede convertirse en una tarea sin sentido,

sin una dirección clara, sin proporcionar información relevante o generar conocimiento útil.

2. Recopilar datos

La recopilación de datos es esencial para la toma de decisiones y la generación de conocimiento en cualquier campo. Los datos son la materia prima para la ciencia de datos y, por lo tanto, la base para generar información valiosa y conocimiento útil.

3. Limpiar datos

La limpieza de datos consiste en la identificación y eliminación de errores, inconsistencias y datos duplicados en un conjunto de datos. Esto es importante porque los datos sucios o incorrectos pueden conducir a un análisis deficiente y en consecuencia conclusiones erróneas. La limpieza de datos garantiza la calidad y la integridad de los datos utilizados en el análisis. Además, reduce el tiempo de procesamiento y los costes de almacenamiento.

4. Analizar los datos

El análisis de los datos implica explorar y visualizar datos para identificar patrones y tendencias. Esto puede incluir técnicas de visualización como gráficos y tablas dinámicas para resumir datos y detectar valores atípicos.

5. Interpretar los datos

Saber interpretar los datos implica poder presentar unos resultados y conclusiones de manera clara y comprensible. Esto permite confirmar o rechazar la que originó el estudio. Para lograrlo, se pueden emplear diversas herramientas, como visualizaciones interactivas, informes y presentaciones.

1.3.1. Elementos estadísticos utilizados en el análisis de datos

Hay muchos factores estadísticos utilizados para el análisis de datos en la ciencia de datos. Estos son algunos de los más populares:

Medidas de tendencia central. Se utilizan para resumir la distribución de un conjunto de datos y pueden incluir la media, la mediana y la moda.

Medidas de dispersión. Mide la variabilidad de los datos y puede incluir la desviación estándar, el rango y la varianza.

Análisis de correlación. Se utilizan para evaluar la relación entre dos variables y determinar si existe una relación entre ellas.

Prueba de hipótesis. Se utiliza para determinar si la diferencia observada en los datos es estadísticamente significativa o aleatoria.

Regresión. Se utiliza para modelar la relación entre dos o más variables y para predecir el valor de una variable a partir de otra.

Análisis de series temporales. Se utiliza para modelar y predecir patrones y tendencias en los datos a lo largo del tiempo.

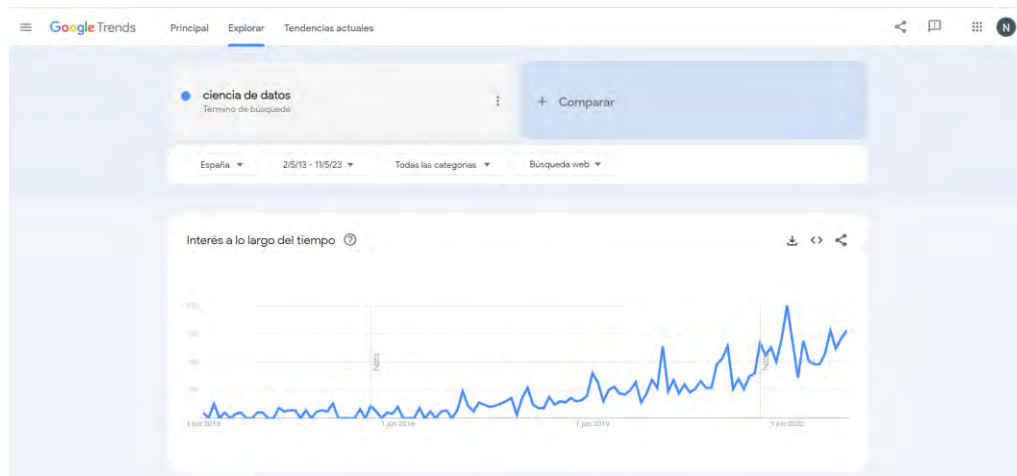
Análisis de clúster. Se utiliza para agrupar datos en función de su similitud en variables específicas.

1.3.2. La carrera de moda en el siglo XXI

La popularidad de la ciencia de datos se ha disparado en los últimos años, convirtiéndose en uno de los campos de mayor demanda laboral a pesar de ser relativamente nuevo. La Figura 6 refleja cómo ha aumentado el interés por este tema los últimos diez años, tal y como se puede observar en el interés de búsqueda en Google.

Figura 6

Evolución del interés del término ciencia de datos en Google



Nota: El gráfico representa cómo se ha popularizado el término ciencia de datos a partir de las búsquedas de Google en los últimos 10 años. En mayo de 2013 había un interés de búsqueda del 5%, mientras que en mayo del 2023 está en un 78%. Fuente: Google Trends².

² Google Trends es una herramienta en línea de Google que permite a los usuarios ver y analizar cómo ha cambiado con el tiempo el interés de búsqueda de un término o un tema en particular.

Este aumento de popularidad se debe en parte a la mayor cantidad de datos disponibles y la mayor velocidad de procesamiento de las computadoras modernas y la computación en la nube.

2. Contexto de la propuesta de intervención y análisis de necesidades

2.1. Contexto de la propuesta de intervención

La situación de aprendizaje propuesta tiene lugar en el centro Maristes Champagnat de Badalona (Barcelona). Un aspecto muy importante de su plan educativo son las nuevas tecnologías. Uno de sus proyectos educativos más relevantes es la programación y la robótica. Se proponen trabajar la fabricación, programación y robótica para buscar soluciones creativas a las necesidades que nos vamos encontrando en cualquier ámbito.

Cabe destacar que el centro ofrece la optativa de programación desde 1.º de Educación Secundaria Obligatoria (ESO) hasta 1.º de Bachillerato, lo cual demuestra su compromiso con el desarrollo de habilidades tecnológicas desde temprana edad. La propuesta de la situación de aprendizaje de este trabajo final de máster está diseñada específicamente para el grupo de primero de Bachillerato, con el objetivo de brindar a los estudiantes una experiencia enriquecedora y relevante en programación y estadística.

En el centro Maristes Champagnat, se ha establecido un sólido programa de enseñanza de la programación, comenzando con Scratch en los primeros cursos y posteriormente introduciendo a los estudiantes al lenguaje de programación Python a partir de tercero de la ESO. Esta progresión en el aprendizaje de la programación permite a los estudiantes adquirir habilidades sólidas en el área y estar preparados para enfrentar desafíos más complejos.

Es en este contexto que se plantea la situación de aprendizaje en Python, utilizando el paquete específico de Pandas para manipular y presentar datos, así como aplicar conceptos básicos de estadística. Al aprovechar el conocimiento previo de los estudiantes en programación y su familiaridad con Python, se busca potenciar su aprendizaje al aplicar estos conocimientos a la resolución de problemas reales y al análisis de datos.

2.2. Análisis de necesidades

Actualmente, el centro está apostando por la innovación tecnológica. El nuevo currículum, que apuesta por las situaciones de aprendizaje, tiene como objetivo aprender con sentido. Aprendemos con sentido cuando acercamos la escuela a las situaciones, cuestiones y problemas reales de la vida cotidiana. El objetivo no es

simplemente adquirir saberes, sino hacerlos significativos, aprender a utilizar esos conocimientos para resolver las necesidades del mundo real.

Para mantenerse al día con las últimas innovaciones, el centro necesita dedicar grandes esfuerzos a la preparación de nuevo material didáctico actualizado. Preparar una situación de aprendizaje de programación y estadística puede ser una excelente forma para estar siempre al día. Además de brindar a los estudiantes habilidades valiosas y relevantes para el mercado laboral, esta combinación fomenta un enfoque interdisciplinario que muestra cómo la estadística puede ser una herramienta útil en diversos campos y no simplemente una materia aislada. Asimismo, la enseñanza de programación y estadística contribuye al desarrollo de habilidades digitales y analíticas en los estudiantes, que son fundamentales en la sociedad actual.

Si nos centramos en la parte académica, muchos profesores de matemáticas del centro han expresado su preocupación por no tener tiempo para enseñar estadística. Sin embargo, esta es un área que se considera muy importante en términos de competencias. De hecho, si observamos las competencias básicas de matemáticas o científicotecnológicas de cuarto de Educación Secundaria Obligatoria³, veremos que hay una gran cantidad de tareas que requieren interpretación de datos y gráficos. Es esencial que la enseñanza de la estadística se incorpore lo suficiente en las escuelas para garantizar que los estudiantes adquieran las habilidades necesarias en el futuro.

2.3. Propuesta de intervención

Situación de aprendizaje

Título	¿Qué nos dicen los datos?
Curso (nivel educativo)	Primero de Bachillerato
Área / Materia / Ámbito	Programación

³ Se pueden consultar todas las distintas pruebas de competencias básicas para 4 de Educación Secundaria Obligatoria en el enlace siguiente: <http://csda.gencat.cat/ca/arees-actuacio/avaluacions-consell/avaluacio-quart-eso/>

DESCRIPCIÓN (contexto + reto)

¿Por qué esta situación de aprendizaje? ¿Está relacionada con alguna otra? ¿Cuál es el contexto? ¿Qué reto plantea?

¿Qué nos dicen los datos?, es una situación de aprendizaje que se desarrolla en primero de Bachillerato en la materia optativa de Programación.

La ciencia de datos es una disciplina que cada vez cobra más importancia debido a la cantidad de datos que se generan en el mundo. La explosión de datos en Internet y las redes sociales han creado una gran cantidad de información que puede resultar confusa y difícil de usar sin las herramientas y técnicas adecuadas. Actualmente, la ciencia de datos es una disciplina importante para aprovechar al máximo la información que pueden llegar a proporcionar los datos y ayuda a las personas en la toma de decisiones.

A lo largo de esta situación de aprendizaje, los alumnos aprenderán a tratar, analizar y contar historias a partir de los datos. Esta actividad busca fomentar la motivación de los alumnos hacia la estadística mediante la programación, mostrándoles su utilidad práctica.

Se propondrá al alumnado trabajar con datos de su ciudad para elaborar un informe y un póster, que se presentarán en el ayuntamiento de su localidad. Podrán observar cómo un buen análisis de datos puede llegar a influir en la toma de decisiones importantes.

COMPETENCIAS ESPECÍFICAS

Con la realización de esta situación de aprendizaje se favorece el logro de las competencias específicas siguientes:

Competencias específicas	Materia
Competencia 2 Crear prototipos visuales de programas y aplicaciones y utilizarlos para crear interfaces de usuario con las herramientas informáticas adecuadas.	Programación

<p>Competencia 3</p> <p>Partiendo de la descripción funcional de una solución informática, desarrollar el programa, la aplicación o el servicio web que las implementa utilizando un lenguaje y un entorno de programación adecuados.</p>	<p>Programación</p>
<p>Competencia 3</p> <p>Seleccionar, utilizar y configurar las herramientas digitales necesarias, aplicando conocimientos interdisciplinarios, para resolver tareas y comunicar los resultados de manera ordenada y precisa, con el uso esmerado del lenguaje gráfico y la terminología técnica adecuada.</p>	<p>Tecnología</p>
<p>Competencia 6</p> <p>Vincular y contextualizar las matemáticas en otras áreas de conocimiento, abordando las situaciones que se desprendan, por modelizar y resolver problemas y desarrollar la capacidad crítica, creativa e innovadora en situaciones diversas.</p>	<p>Matemáticas</p>
<p>Competencia 7</p> <p>Comunicar y representar, de forma individual y colectiva, conceptos, procedimientos y resultados matemáticos usando el lenguaje oral, escrito, gráfico y multimedia, mediante diferentes tipos de apoyos, incluyendo los tecnológicos, para dar significado en el conocimiento, transferirlo y compartirlo.</p>	<p>Matemáticas</p>

<p>Competencia 9</p> <p>Cooperar, desarrollando las destrezas sociales necesarias para participar activamente en los equipos de trabajo inclusivos reconociendo la diversidad y el valor de las aportaciones de los otros, para compartir y construir conocimiento matemático de manera colectiva.</p>	<p>Matemáticas</p>
---	--------------------

TRATAMIENTO DE LAS TRES COMPONENTES TRANSVERSALES DE LAS COMPETENCIAS CLAVE DEL BACHILLERATO

<p>Gestión y comunicación de la información: Tanto en la actividad inicial como en otras actividades relacionadas con la situación de aprendizaje, los estudiantes tratarán con diferentes tipos de documentos para dar respuesta a las diferentes preguntas que se vayan planteando. Mientras que en la etapa de estructuración se proporcionan los documentos y los datos procesados, en la etapa de aplicación deben buscar y seleccionar la información que consideren necesaria para lograr el objetivo. Además, los estudiantes también deben hacer una presentación oral de los resultados de su investigación.</p> <p>Resolución de problemas a partir de la aplicación integrada de los aprendizajes: La situación de aprendizaje, en general, parte de la resolución de un problema. Al analizar los datos, identificamos problemas y buscamos soluciones. Se pueden descubrir patrones y tendencias que son invisibles a simple vista. Una vez que se identifican los problemas, los datos pueden proporcionar la información necesaria para tomar una decisión informada.</p> <p>Juicio y pensamiento crítico: El alumnado tendrá que decidir qué datos son los más importantes. Al aprender a seleccionar y evaluar datos, pueden desarrollar habilidades analíticas y de pensamiento crítico.</p>
--

OBJETIVOS DE APRENDIZAJE Y CRITERIOS DE EVALUACIÓN

<p>Objetivos de aprendizaje ¿Qué queremos que aprenda el alumnado y para qué? CAPACIDAD + SABER + FINALIDAD</p>	<p>Criterios de evaluación ¿Cómo sabemos que lo ha aprendido? ACCIÓN + SABER + CONTEXTO</p>
<p>1. Diseñar un informe mediante herramientas informáticas a partir de unos datos para concluir un análisis. (Programación, CE2)</p>	<p>1. Hacer uso de las herramientas de programación aprendidas para desarrollar parte del producto final.</p>
<p>2. Comprender y analizar los datos proporcionados mediante el lenguaje de programación Python para conocer e implementar la tecnología más actual en el campo del análisis de datos. (Programación, CE3)</p>	<p>2.1. Usar correctamente los aprendizajes aprendidos en la fase de estructuración en un nuevo conjunto de datos. 2.2. Manejar correctamente el uso de la librería Pandas de Python en el entorno de Google Colab.</p>
<p>3. Presentar conclusiones de datos de manera efectiva, adaptando la visualización de datos a la audiencia para lograr una comunicación clara y efectiva. (Tecnología, CE3)</p>	<p>3.1. Diseñar y presentar los datos del estudio utilizando las herramientas y las aplicaciones digitales adecuadas para lograr adaptarse a la audiencia objetivo (CA 3.1).</p>
<p>4. Comprender y aplicar las matemáticas en la programación mediante la estadística para generar conocimiento y dar respuesta a los problemas durante el análisis de datos. (Matemáticas, CE6)</p>	<p>4.1. Analizar e interpretar la información con sentido crítico de forma individual y colectiva para construir nuevo conocimiento estadístico (CA 6.1). 4.2. Argumentar a partir del análisis realizado sobre diferentes aspectos socioculturales de su localidad, como, por ejemplo, sobre la distribución de instalaciones según el barrio y si corresponde con la edad predominante de este (CA 6.5).</p>

<p>5. Comunicar y visualizar los procesos seguidos en la resolución de un problema para estructurar el propio pensamiento y colaborar en la construcción del conocimiento grupal. (Matemáticas, CE7)</p>	<p>5.1. Comunicar de manera organizada, utilizando el lenguaje de la estadística, las ideas matemáticas (CA 7.1)</p> <p>5.2. Representar los datos con los gráficos adecuados (CA 7.2).</p> <p>5.3. Escribir un informe usando el lenguaje, la simbología y el rigor preciso (CA 7.4).</p>
<p>6. Desarrollar habilidades sociales como la colaboración, participando activamente en grupos de trabajo inclusivos, compartiendo y recopilando conocimientos, con relación al análisis de datos y la estadística (Matemáticas, CE9)</p>	<p>6.1. Aportar y compartir estrategias y razonamientos matemáticos con los compañeros y valorar el éxito colectivo como una estrategia de mejora personal. (CA 9.1)</p> <p>6.2. Plantear preguntas sobre los datos que puedan contribuir activamente con el grupo (CA 9.3).</p> <p>6.3. Ayudar a identificar errores y dificultades de aprendizaje de los compañeros haciendo aportaciones constructivas y concretas (CA 9.4).</p>

SABERES

Con la realización de esta situación de aprendizaje se tratarán los saberes siguientes:

	Saber	Materia
1	Uso de una de las librerías de Python más populares para el análisis de datos estadísticos.	Programación
2	Comprensión de los fundamentos algorítmicos en el análisis de datos.	Programación
3	Autoconfianza e iniciativa. Identificación y gestión de emociones.	Programación

4	Resolución de tareas y proyectos con una actitud emprendedora, resiliente, perseverante y creativa para abordar problemas desde una perspectiva interdisciplinaria.	Tecnología
5	Análisis, representación e interpretación de relaciones cuantitativas usando herramientas tecnológicas cuando sea necesario.	Matemáticas
6	Organización de los datos procedentes de variables bidimensionales. Análisis de la dependencia estadística.	Matemáticas
7	Diseño de estudios estadísticos usando las herramientas digitales para responder a retos o problemas susceptibles de ser tratados con métodos estadísticos.	Matemáticas
8	Análisis de muestras bidimensionales con herramientas tecnológicas con el fin de emitir juicios y de tomar decisiones.	Matemáticas
9	Destrezas para explorar y valorar diferentes estrategias en el tratamiento matemático de un problema o de una situación.	Matemáticas
10	Capacidad de consensuar opiniones y estrategias diversas en la hora de tomar una decisión colectiva en el desarrollo de una actividad matemática.	Matemáticas

DESARROLLO DE LA SITUACIÓN DE APRENDIZAJE

¿Cuáles son las principales estrategias metodológicas que se prevén utilizar? ¿Qué tipos de agrupamiento realizaremos? ¿Cuáles son los principales materiales que necesitaremos? Etc.

Para desarrollar esta situación de aprendizaje, se ha aplicado el modelo Diseño Universal para el Aprendizaje, con el objetivo de crear un entorno de aprendizaje que sea accesible y significativo para todos los estudiantes. Permite que cada estudiante aprenda de acuerdo con sus necesidades e intereses.

Durante la situación de aprendizaje, el trabajo en el aula se llevará a cabo en grupos cooperativos, ya que esto permite un mejor ajuste de las diferentes habilidades y niveles de rendimiento de los estudiantes. Los profesores pueden organizar los grupos de forma heterogénea en cuanto a las capacidades y el desempeño de los alumnos en el aula. También podrán preguntar individualmente a cada alumno por escrito con quién quiere trabajar y con quién no, justificándolo, y formar grupos con toda esa información.

Todos los estudiantes tendrán un ordenador, acceso a internet y papel para tomar notas, hacer esquemas y resolver los distintos problemas que vayan surgiendo. Las actividades entregables se depositarán a través de una plataforma escolar como Microsoft Teams u otras herramientas similares.

Al inicio de cada tipo de actividad (inicial, desarrollo, estructuración y aplicación), el profesor hará una explicación a toda la clase, detallando las actividades propuestas y el tiempo asignado para realizarlas. Las primeras partes de las distintas actividades de desarrollo se pueden realizar con o sin la explicación del profesor, dependiendo de las habilidades que vaya demostrando el alumnado.

Durante la fase de aplicación, el profesor acompañará a los alumnos y les hará preguntas para ayudarlos a seguir adelante, si el grupo está bloqueado, no darán instrucciones detalladas. Al final de esta fase, los alumnos expondrán sus trabajos en la escuela y entregarán un informe en el ayuntamiento. Es muy importante poder adaptarse a diferentes audiencias. Se organizará un pequeño foro donde cualquier alumno de la escuela podrá conocer los resultados obtenidos en su localidad.

ACTIVIDADES DE APRENDIZAJE Y DE EVALUACIÓN

Actividad	Descripción de la actividad de aprendizaje y de evaluación	Temporización
Actividades iniciales <i>¿Qué sabemos?</i>	Actividad 1. ¿Para qué sirven los datos? La actividad inicial de la situación de aprendizaje consta de tres partes. - Iniciaremos con la rutina de pensamiento “veo, pienso, me pregunto” para reflexionar sobre la importancia de los	1 sesión

	<p>datos en la actualidad. Utilizaremos la frase de Clive Humby: "Los datos son el nuevo petróleo" como punto de partida.</p> <ul style="list-style-type: none"> - A continuación, utilizaremos la herramienta Google Trends, https://trends.google.es/home, para explorar cómo ha cambiado con el tiempo el interés de búsqueda de un término o un tema en particular. Junto con el alumnado, visualizaremos las tendencias más populares de ese momento y analizaremos cómo cambian a lo largo del tiempo. También podremos observar cómo la búsqueda de ciertas palabras, como "resfriado", aumenta en invierno y disminuye en verano. - Por último, les enseñaremos el perfil que Google crea de cada uno a partir de nuestras búsquedas para mostrarnos anuncios. Utilizaremos la web https://myadcenter.google.com/controls?hl=es. <p>Material de soporte:</p> <ul style="list-style-type: none"> - Rutina de pensamiento para la frase de frase de Clive Humby. Se puede encontrar en el anexo I. - Documentos de posibles temas para explorar con la herramienta Google Trends. Se puede encontrar en el anexo II. - Instrucciones detalladas para acceder a los perfiles de Google. Se puede encontrar en el anexo III. 	
--	--	--

<p>Actividades de desarrollo</p> <p><i>Aprendemos nuevos saberes</i></p>	<p>Actividad 2. Google Colab: ¡Analiza y Visualiza Datos!</p> <p>En la primera sesión de desarrollo introduciremos la plataforma Google Colab. Esta plataforma será la que usaremos a lo largo de toda la situación de aprendizaje para trabajar con datos y realizar análisis y visualizaciones. Para usar esta plataforma se requiere un correo de Google, es posible que no todos los alumnos dispongan de unos. El objetivo de esta sesión es que todo el alumnado pueda registrarse, navegar y utilizar la plataforma.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Cuaderno de Google Colab de la actividad 'Introducción a Google Colab'. Se puede encontrar la actividad en el anexo IV. 	<p>1 sesión</p>
	<p>Actividad 3. Librería Pandas. Series</p> <p>En esta actividad de desarrollo introduciremos la librería de Python que más utilizaremos a lo largo de toda la situación de aprendizaje. Comenzaremos trabajando el elemento más básico de la librería, las Series. Aprenderemos cómo crear y manipular Series y cómo realizar operaciones básicas con ellas.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Cuaderno de Google Colab para los alumnos de la actividad 'Librería Pandas. Series'. Se puede encontrar la actividad en el anexo V. 	<p>3 sesiones</p>

	<p>Actividad 4. Librería Pandas. DataFrame</p> <p>Continuaremos con la librería pandas. En esta ocasión, introduciremos el elemento más importante, los DataFrames. Es fundamental entender bien los DataFrames, ya que son la base del análisis de datos con pandas.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Cuaderno de Google Colab para los alumnos de la actividad 'Librería Pandas. DataFrame. Se puede encontrar la actividad en el anexo VI. 	3 sesiones
	<p>Actividad 5. Visualización de datos</p> <p>Esta actividad consta de dos partes:</p> <ul style="list-style-type: none"> - Presentaremos en clase los conceptos fundamentales que se deben tener en cuenta para realizar una buena visualización. Utilizaremos una presentación realizada con la plataforma genial.ly. En la última parte, se mostrarán ejemplos de visualizaciones de datos con errores y fallos, ejemplos de lo que no se debe hacer. Esta parte puede generar mucho debate con el alumnado. - Introduciremos la librería Matplotlib junto con Pandas para realizar visualizaciones de datos. Explicaremos en qué situaciones es más adecuado utilizar cada tipo de gráfico, así los estudiantes podrán llegar a elegir la mejor opción según los datos y el objetivo de la visualización. Al comprender cuándo y 	<p>3 sesiones:</p> <ul style="list-style-type: none"> - 1 sesión de presentación - 2 sesiones de Matplotlib

	<p>cómo utilizar cada tipo de gráfico, podrán crear visualizaciones efectivas que ayuden a transmitir información de manera clara y comprensible.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Presentación 'Una buena visualización'. Se puede encontrar en el anexo VII. - Cuaderno de Google Colab para los alumnos de la actividad 'Visualización de datos'. Se puede encontrar la actividad en el anexo VIII. 	
	<p>Actividad 6. Estadística con DataFrames</p> <p>Esta actividad consta de dos partes:</p> <ul style="list-style-type: none"> - Explicaremos los conceptos fundamentales de estadística que deben ser abordados en primer año de Bachillerato. Se explicarán conceptos como población, individuo, muestra, frecuencia, moda, media, mediana, desviación típica y correlación. Es importante que los estudiantes comprendan los conceptos básicos de la estadística para poder interpretar y analizar datos de manera efectiva. - Aprenderemos a calcular los distintos conceptos estadísticos que hemos explicado anteriormente utilizando la librería Pandas. 	<p>3 sesiones:</p> <ul style="list-style-type: none"> - 2 sesión explicación - 1 sesión de cálculos

	<p>Material de soporte:</p> <ul style="list-style-type: none"> - Cuaderno de Google Colab para los alumnos de la actividad 'Estadística'. Se puede encontrar la actividad en el anexo IX. 	
	<p>Actividad 7. Un análisis completo</p> <p>El alumnado pondrá en práctica los conceptos aprendidos en las actividades 5 y 6 mediante la realización de un pequeño estudio y la visualización de los resultados obtenidos. Este ejercicio práctico les permitirá aplicar los conocimientos adquiridos en una situación real.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Cuaderno de Google Colab para los alumnos de la actividad 'Un análisis completo'. Se puede encontrar la actividad en el anexo X. 	3 sesiones
<p>Actividades de estructuración</p> <p><i>¿Qué hemos aprendido?</i></p>	<p>Actividad 8. ¿Qué hemos aprendido?</p> <p>En esta actividad, el alumnado utilizará los datos recopilados y visualizados en la actividad 7 para crear un póster e informe y presentarlos ante la clase. Pueden usar herramientas como Canva o Word para crear el póster e informe. El objetivo es que el alumnado pueda tener un primer contacto con el producto final que deberán realizar. Se trata de una actividad rápida, no se espera un gran nivel de profundidad en los resultados.</p>	<p>3 sesiones:</p> <ul style="list-style-type: none"> - 2 sesiones preparación - 1 sesión de presentación

	<p>Material de soporte:</p> <ul style="list-style-type: none"> - Documento con la actividad detallada. Se puede encontrar la actividad en el anexo XI. - Rúbrica de evaluación. Se puede encontrar la actividad en el anexo XII. 	
<p>Actividades de aplicación</p> <p><i>Aplicamos lo que hemos aprendido</i></p>	<p>Actividad 9. Analizando datos</p> <p>En la primera actividad de aplicación, les proporcionaremos al alumnado dos conjuntos de datos nuevos. En este caso trabajaremos con el listado de equipamientos de Badalona dividido por barrio, junto a la población segmentada por edad en cada uno de los barrios. En lugar de guiarlos en un análisis específico, se les dará libertad para decidir qué información consideran relevante y qué aspectos quieren investigar. La toma de decisiones sobre qué variables son más importantes y cómo quieren presentar los resultados ha de ser completamente del alumnado.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Documento con la actividad detallada. Se puede encontrar la actividad en el anexo XIII. - Conjuntos de datos. Se pueden encontrar la actividad en el anexo XIII. 	6 sesiones
	<p>Actividad 10. ¿Qué historia nos cuentan los datos?</p> <p>Para concluir la situación de aprendizaje, el alumnado deberá presentar los resultados de su proyecto en forma de un informe y un póster. El</p>	<p>6 sesiones:</p> <ul style="list-style-type: none"> - 5 sesiones preparación

	<p>informe estará dirigido al Ayuntamiento de Badalona, incluirá las conclusiones y recomendaciones para mejorar los equipamientos en los distintos barrios. El póster se presentará en un congreso organizado en la escuela para compartir los resultados con otros estudiantes y profesores.</p> <p>Material de soporte:</p> <ul style="list-style-type: none"> - Documento con la actividad detallada. Se puede encontrar la actividad en el anexo XIV. - Rúbrica para la evaluación del profesor. Se puede encontrar la actividad en el anexo XV. - Rúbrica para la coevaluación de posters. Se puede encontrar en el anexo XVI. - Rúbrica para la coevaluación y autoevaluación del trabajo en grupo. Se puede encontrar en el anexo XVII. 	<p>informe y póster</p> <p>- 1 sesión congreso</p>
--	---	--

MEDIDAS I SOPORTES UNIVERSALES

La situación de aprendizaje incluye un conjunto de medidas y apoyos para facilitar el aprendizaje en todo el alumnado:

- Fomenta la colaboración y el trabajo en equipo. Los estudiantes siempre tienen la oportunidad de apoyo entre iguales, además de la acción del o de la docente.
- Promover la diversidad y la igualdad en el aula teniendo en cuenta las diferentes perspectivas.
- Utiliza tecnología para mejorar el acceso a los materiales del curso y aumentar la interacción y la participación de los estudiantes.
- Permite a los estudiantes flexibilidad tanto en el contenido del curso como en su organización.

MEDIDAS I SOPORTES ADICIONALES O INTENSIVOS

Qué medidas o apoyos adicionales o intensivos se proponen para cada uno de los alumnos siguientes:

Alumno/a	Medida y soporte adicional o intensivo

3. Conclusiones

Al inicio de este trabajo final de máster se habían planteado dos objetivos:

- Diseñar una asignatura trimestral que ayudará a los estudiantes a convertirse en analistas de datos mediante un proyecto de programación.
- Motivar a los estudiantes a obtener un aprendizaje significativo y duradero en el análisis de datos.

Además, resulta complicado identificar las limitaciones que podrían surgir sin haberlo implementado.

Para alcanzar estos objetivos, se ha implementado una situación de aprendizaje utilizando la metodología de Aprendizaje Basado en Proyectos. Aunque no ha sido posible llevar a cabo el proyecto en la práctica y obtener retroalimentación directa de los estudiantes, es importante destacar que la situación de aprendizaje diseñada cumple con los ocho elementos esenciales que debe incluir un buen proyecto. El contenido es altamente significativo, existe una clara necesidad de conocimiento, el título del proyecto es provocador y transmite la esencia del mismo, y los estudiantes tienen una participación activa y voz en el proceso. Además, se abordan competencias del siglo XXI, se fomenta la constante formulación de preguntas por parte de los estudiantes, se brinda una guía efectiva por parte del docente y se presenta un producto final ante una audiencia.

Con relación al segundo objetivo, debido a la limitación de no poder verificar directamente si los estudiantes obtuvieron la motivación deseada o adquirieron conocimientos significativos a través de esta situación de aprendizaje, es importante considerar otros factores que respaldan la posibilidad de alcanzar estos resultados. La revisión bibliográfica sobre los beneficios e inconvenientes del Aprendizaje Basado en Proyectos ha demostrado consistentemente que, cuando se implementa de manera efectiva, este enfoque pedagógico puede ser altamente motivador y propiciar un aprendizaje profundo y significativo. Por lo tanto, aunque no se cuente con una retroalimentación directa de los estudiantes en este contexto particular, existe una base teórica sólida que respalda la conclusión de que un proyecto bien diseñado puede, en efecto, generar la motivación necesaria y facilitar un aprendizaje de calidad en el análisis de datos.

En cuanto a futuras perspectivas para esta situación de aprendizaje, se pueden considerar varias opciones. Una de ellas es la incorporación de elementos de gamificación de actividades para hacerlas más atractivas y motivadoras, siempre

teniendo en cuenta la importancia de diseñar cuidadosamente estos elementos para no complicar la adquisición de los conceptos fundamentales.

Otra opción interesante para ampliar la situación de aprendizaje es la capacidad de analizar diferentes tipos de datos, como los de redes sociales, deportes y medio ambiente. Una opción adicional sería permitir que cada grupo de trabajo se centre en el tipo de datos que más les gusta, lo que fomentaría la exploración de diferentes perspectivas en el análisis de datos. El estudio de datos de redes sociales permite al alumnado explorar cómo se recopilan, analizan y utilizan los datos en las plataformas de redes sociales. Podrían aprender a extraer información, identificar tendencias, analizar el impacto de las publicaciones y comprender el comportamiento de los usuarios. El análisis de datos en el deporte ofrece a los estudiantes la oportunidad de explorar cómo se puede aplicar el análisis de datos en disciplinas deportivas. Podrían aprender a analizar estadísticas deportivas, identificar patrones y utilizar los datos para tomar decisiones estratégicas. El estudio de datos ambientales permite a los estudiantes explorar y analizar los datos ambientales para abordar problemas relacionados con el cambio climático, la conservación de la biodiversidad y la sostenibilidad.

Además, se podría considerar la colaboración con instituciones educativas, organizaciones locales o incluso empresas relacionadas con el análisis de datos, con el fin de brindar al alumnado experiencias más cercanas a la realidad. Esto podría implicar la realización de visitas a laboratorios o empresas, participación en proyectos de investigación o mentorías por parte de profesionales del campo. Estas oportunidades prácticas no solo enriquecerían el aprendizaje de los estudiantes, sino que también les proporcionarían una visión más amplia de las aplicaciones y perspectivas de futuro en el análisis de datos.

Asimismo, sería beneficioso fomentar la participación de los estudiantes en competiciones o desafíos relacionados con el análisis de datos a nivel local o nacional. Esto les permitiría aplicar y demostrar sus habilidades en un contexto competitivo, desarrollar su pensamiento crítico y creativo, y establecer conexiones con otros estudiantes y profesionales del campo. Estas experiencias extracurriculares podrían impulsar su motivación, confianza y reconocimiento en el ámbito del análisis de datos.

Otra forma de ampliar este trabajo final de máster sería preparando nuevas situaciones de aprendizaje para abordar los dos trimestres restantes. En estos proyectos podrían enfocarse temas como los algoritmos de los videojuegos, la programación de una habitación inteligente, la construcción de un robot o la creación de una inteligencia artificial. También se podría enriquecer el aprendizaje explorando el uso de herramientas de aprendizaje automático y análisis de datos avanzado en proyectos futuros.

4. Bibliografía

- Batanero, C. (2000). ¿Hacia dónde va la educación estadística?. *Blaix*, 15, 2-13. <https://www.ugr.es/~batanero/pages/ARTICULOS/BLAIX.pdf>
- De Miguel, R. (s.f.). Qué es, qué beneficios aporta y cómo se pone en marcha el ABP en el aula. *Educación* 3.0. <https://www.educaciontrespuntocero.com/noticias/abp-en-el-aula-beneficios/#Principales-beneficios-del-ABP>
- Fernández Solo de Zaldivar, I. (s.f.). *Juego serio: gamificación y aprendizaje*. Centro de Comunicación y Pedagogía. <http://www.centrocp.com/juego-serio-gamificacion-aprendizaje/>
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., Solomon, C. (1969). Programming-Languages as a Conceptual Framework for Teaching Mathematics. *ACM SIGCUE Outlook*, 4(20), 13-17. <https://doi.org/10.1145/965754.965757>
- Fuentes, R. (22 de febrero de 2018). *La diferencia entre los proyectos y el aprendizaje basado en proyectos*. Yo Profesor. <https://yoprofesor.org/2018/02/22/15-herramientas-para-un-mejor-aprendizaje-basado-en-proyectos/>
- Orientacion Anduja. (11 de mayo de 2015). ABP Siete ventajas del aprendizaje basado en proyectos. Orientación Andújar Recursos Educativos. <https://www.orientacionandujar.es/2015/05/11/abp-siete-ventajas-del-aprendizaje-basado-en-proyectos/>
- Saeli, M., Perrenet, J., Jochems, W., Zwaneveld, B. (2011). Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education*, 10(1), 73-88. <https://files.eric.ed.gov/fulltext/EJ1064282.pdf>
- Sánchez, J. (15 de mayo de 2017). *Qué dicen los estudios sobre el Aprendizaje Basado en Proyectos*. Educación para la solidaridad. <https://educacionparalasilididad.com/2017/05/15/que-dicen-los-estudios-sobre-el-aprendizaje-basado-en-proyectos/>
- Santamaría, R. (2 de julio de 2021). *El origen y la evolución de la ciencia de datos (data science)*. Fundación iS+D para la Investigación Social Avanzada. <https://isdfundacion.org/2021/07/02/el-origen-y-evolucion-de-la-ciencia-de-datos-data-science/>
- Soloway, E. (1993). Should we Teach Students to Program? *Communications of the ACM*, 36(10), 21-24. <https://dl.acm.org/doi/10.1145/163430.164061>
- Trujillo, F. (2015). *Aprendizaje basado en proyectos. Infantil, Primaria y Secundaria*. Ministerio de Educación y Formación Profesional.
- Aprendizaje basado en proyectos: características, beneficios, etc.* (1 de diciembre de 2021) Classlife Education. <https://www.classlife.education/blog/aprendizaje-basado-en-proyectos/>

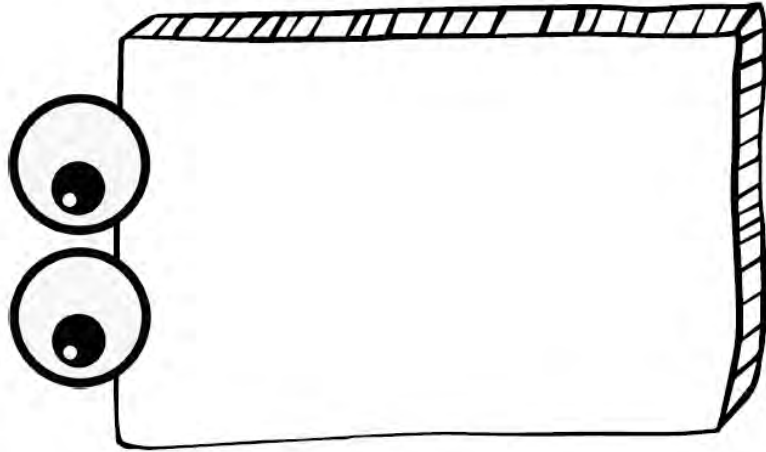
Anexo I. Rutina de pensamiento: “Los datos son el nuevo petróleo”

Se puede descargar la rutina de pensamiento en el siguiente enlace:
https://www.canva.com/design/DAFiiurqciU/5K0qNw0ELkbQGSFZL1lw1Q/edit?utm_content=DAFiiurqciU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

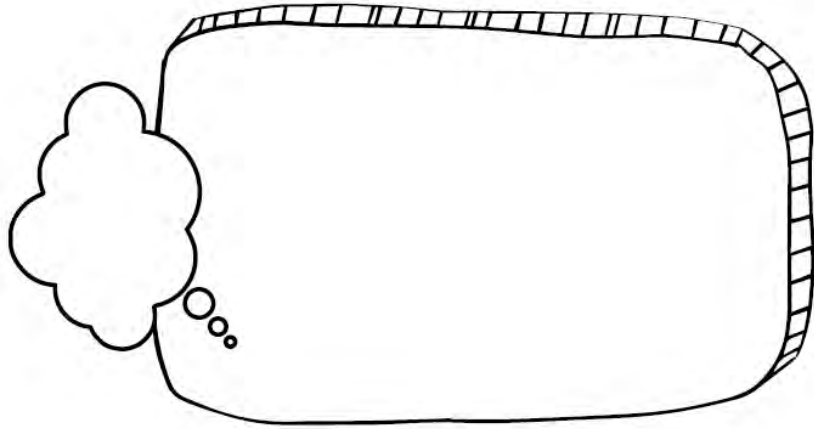


¿QUÉ IMPORTANCIA TIENEN LOS DATOS EN LA ACTUALIDAD?

VEO

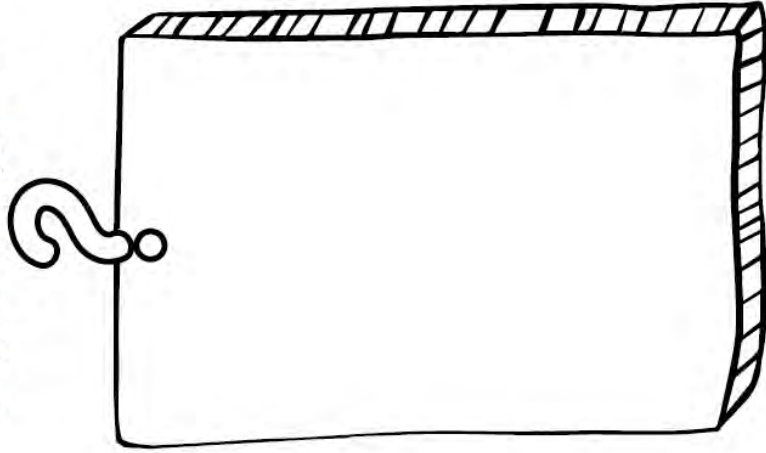


PIENSO



ME

PREGUNTO



Anexo II. Temas para explorar con Google Trends

TEMAS PARA EXPLORAR CON LA HERRAMIENTA GOOGLE TRENDS

1. Accede al enlace <https://trends.google.es/trends/> para utilizar Google Trends.
2. Explora la pestaña 'Tendencias Actuales', donde encontrarás los términos más populares en la actualidad.
3. Ingresa a la pestaña 'Explorar', que ofrece varios ejemplos que puedes utilizar como punto de partida. A partir de ellos, podrás variar los términos de acuerdo con las inquietudes del alumnado. Es muy importante animar a los estudiantes a explorar diferentes comparaciones y discutir los resultados conjuntamente.

A continuación, se presentan algunas ideas de comparaciones que se pueden realizar:

- Compara los términos fútbol y el fútbol americano en Estados Unidos. Puedes acceder a la comparación en el siguiente enlace:
https://trends.google.es/trends/explore?q=%2Fm%2F02vx4,%2Fm%2F0jm_&date=all&geo=US
- Realiza comparaciones entre dos personas célebres, como cantantes, actores, escritores, etc.
- Compara dos tipos de comidas, por ejemplo, cupcake contra madalena o frankfurt contra hamburguesa.

Anexo III. Guía para acceder al perfil de anuncios de Google

INSTRUCCIONES DETALLAS PARA ACCEDER AL PERFIL DE ANUNCIOS DE GOOGLE

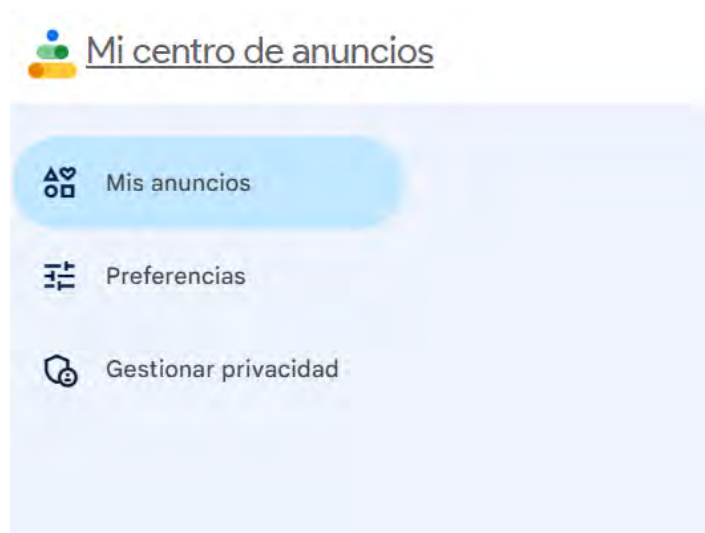
Google recopila información sobre los usuarios y decide que anuncios mostrarles según el perfil que crea de cada uno. Se accede a este perfil mediante el siguiente enlace:

Se debe iniciar sesión en Google o en uno de los servicios de Google, como Gmail o YouTube, para acceder a esta página.

Los perfiles recopilados por Google incluyen una amplia variedad de información, desde intereses y hábitos hasta modelo de automóvil, sistema operativo utilizado, país de origen, actividad deportiva, estado civil y actividad laboral.

¿Qué sabe Google de nosotros?

Accede al menú 'Gestionar privacidad' en el lado izquierdo de la página, podrás ver y gestionar tu información privada y las deducciones que Google ha hecho según tus búsquedas



Menú del centro de anuncios de Google

Anexo IV. Google Colab: ¡Analiza y visualiza datos!

Se puede acceder a esta actividad en el siguiente enlace:

https://colab.research.google.com/drive/10yOkbm1OjKONFHtyuLXEp8IOAI_rr0Ya?usp=sharing.

▼ 2. Google Colab: ¡Analiza y Visualiza Datos!

En este documento encontraréis un pequeño resumen sobre cómo funciona el aplicativo Google Colab y algunos ejemplos de lo que se puede llegar a hacer.

Lo primero que debes saber es que este documento no es una página web estática, sino un entorno interactivo denominado cuaderno de Colab que te permite escribir y ejecutar código.

Existen dos tipos de celdas: de texto y de código. En las casillas de texto solo podréis escribir, en las de código, en cambio, podréis ejecutar diferentes funciones realizadas con lenguaje Python. A continuación tiene un ejemplo de una celda de código.

```
# Celda de código
```

```
# Celda de código
mensaje = "Hola"
print(mensaje)
```

Si quieres ejecutar el código de la celda anterior, debes hacer clic en el botón de reproducir situado a la izquierda de la celda o pulsar "Mayus + Intro".

Para editar, solo tienes que hacer clic en la celda.

Las variables que definas en una celda se pueden utilizar más adelante en otras celdas:

```
mensajeLargo = mensaje + " món"
print(mensajeLargo)
```

▼ Librerías

Con esta herramienta se pueden importar las más famosas librerías de Python.

Una librería es un conjunto de funciones, escritas previamente, que pueden reutilizarse en diferentes programas para realizar tareas concretas.

En el siguiente ejemplo importamos la librería time:

```
import time
print("Durmiendo...")
time.sleep(10) # la máquina duerme 10 segundos
print("Buenos días!")
```

▼ Ciencia de datos

Para analizar y visualizar datos podemos utilizar dos de las librerías más populares de Python, estas son Pandas, Numpy y Matplotlib.

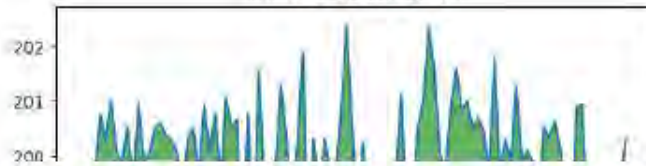
```
import numpy as np
from matplotlib import pyplot as plt

ys = 200 + np.random.randn(100)
x = [x for x in range(len(ys))]

plt.plot(x, ys, '-.')
plt.fill_between(x, ys, 195, where=(ys > 195), facecolor='g', alpha=0.6)

plt.title("Primera Visualización")
plt.show()
```

Primera Visualización



Podemos llegar a realizar visualizaciones realmente complejas, más adelante lo veremos en detalle.

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Generamos dadas aleatorias
num_points = 20
x = 5 * np.arange(num_points) + np.random.randn(num_points)
y = 10 + np.arange(num_points) + 5 * np.random.randn(num_points)
sns.regplot(x=x, y=y)
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')

x1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y1 = np.random.randint(10, size=10)
z1 = np.random.randint(10, size=10)

x2 = [-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]
y2 = np.random.randint(-10, 0, size=10)
z2 = np.random.randint(10, size=10)

ax.scatter(x1, y1, z1, c='b', marker='o', label='blue')
ax.scatter(x2, y2, z2, c='g', marker='D', label='green')

ax.set_xlabel('x axis')
ax.set_ylabel('y axis')
ax.set_zlabel('z axis')
plt.title("3D Scatter Plot Example")
plt.legend()
plt.tight_layout()
plt.show()
```

Incluso podemos realizar visualizaciones interactivas:

```
import altair as alt
from vega_datasets import data
cars = data.cars()

alt.Chart(cars).mark_point().encode(
    x='Horsepower',
    y='Miles_per_Gallon',
    color='Origin',
).interactive()
```

Ahora es el momento de recordar sus habilidades en Python. Podéis modificar este archivo, crear nuevas celdas, experimentar y jugar para ver todo lo que recordáis.

Si no recordáis mucho, os recomiendo hacer el siguiente tutorial, os será de gran ayuda en este curso:

<https://www.codcademy.com/learn/learn-python-3>

Anexo V. Google Colab: Librería Pandas. Series

Se puede acceder a esta actividad en el siguiente enlace:

https://colab.research.google.com/drive/1cB3JQxqANM0yf1Nr4wGqPI4YMsspe_2P?usp=sharing

3. Librería Pandas. Series

- Introducción a la librería Pandas

Pandas es una de las librerías más famosas para Python, se utiliza para la manipulación y análisis de datos. Esta librería proporciona estructuras de datos, eficientes y fáciles de utilizar. Con pandas, puede realizar varias operaciones de datos, tales como la selección, filtrado, agrupación y transformación de datos.

Importamos la librería. Cada vez que escribimos 'pd.' Estaremos accediendo a la librería en cuestión:

```
import pandas as pd
```

- Estructuras básicas

Las estructuras de datos que más se utilizan de la librería Pandas son:

- **Serie:** es una sola columna.
- **DataFrame:** consiste en una tabla de datos relacionales, con filas y columnas con nombre.

Un DataFrame contiene una o más Series y cada Serie tiene un nombre.

En este documento trabajaremos las Series.

Estas estructuras se identifican por un valor único llamado índice. Este valor se inicia en 0 y cuando hemos generado la tabla siempre aumenta de 1 en 1. Es muy útil para acceder a filas específicas.

- Series

Son estructuras similares en las listas de una dimensión. Son homogéneas, es decir, sus elementos deben ser del mismo tipo, y su tamaño es inmutable, es decir, no puede cambiarse, aunque sí su contenido.

Dispone de un índice que asocia un valor a cada elemento de la Serie, a través del cual se accede al elemento.

Índex	Nom
0	Anna
1	Josep
2	Cristina

- Creación de Series

Se puede crear una Serie a partir de una lista o de un diccionario:

- **Lista:** `Series(data=lista, index=índexs, dtype=tipus)`
- **Diccionario:** `Series(data=diccionario, index=índexs)`

```
# Creación a partir de una lista
# Si no se pasa la lista de índices se utilizan como índices los enteros del 0 al n-1,
# donde n es la longitud de la serie. Si no se pasa el tipo de dato, se deduce.
```

```
pd.Series(['Ana', 'Josep', 'Cristina'], dtype='string')
```

```
0    Ana
1   Josep
2  Cristina
dtype: string
```

```
# Creación a partir de un diccionario
# Si no se pasa una lista de índice se utilizan las claves del diccionario.

pd.Series({0:'Ana', 1:'Jose', 2: 'Cristina'})

0      Ana
1      Jose
2  Cristina
dtype: object
```

▼ Atributos de una Serie

Una Serie tiene distintas propiedades:

- | `s.size` : Devuelve el número de elementos de la Serie `s`.
- | `s.index` : Devuelve una lista con los nombres de las filas de la Serie `s`.
- | `s.dtype` : Devuelve el tipo de datos de los elementos de la Serie `s`.

```
s = pd.Series(['Ana', 'Jose', 'Cristina'], index=['A1_1', 'A1_2', 'A1_3'])
s.size
```

```
3
```

```
s.index
```

```
Index(['A1_1', 'A1_2', 'A1_3'], dtype='object')
```

```
s.dtype
```

```
dtype('O')
```

▼ Acceso a los elementos de una Serie

A una Serie podemos acceder según:

- La posición

Es similar a cómo se accede a los elementos de un array.

- | `s[i]` : Devuelve el elemento que ocupa la posición `i+1` en la Serie `s`.
- | `s[posiciones]` : Devuelve otra Serie con los elementos que ocupan las posiciones de la lista de `posiciones`.

- El índice

- | `s[nombre]` : Devuelve el elemento con el índice de nombre `nombre`.
- | `s[nombres]` : Devuelve otra Serie con los elementos correspondientes a los nombres de la lista `nombres` en el índice.

! El índice es único, nunca podrán existir dos índices repetidos

```
# Acceso per posición
print(s)
print("Imprimimos el valor de la posición 0:", s[0])
print("Imprimimos el valor de la posición 2:", s[2])
print("Imprimimos los valores de las posiciones 0 y 1: \n", s[0:2])
```

```
A1_1      Ana
A1_2      Jose
A1_3  Cristina
dtype: object
Imprimimos el valor de la posición 0: Ana
Imprimimos el valor de la posición 2: Cristina
Imprimimos los valores de las posiciones 0 y 1:
A1_1      Ana
A1_2      Jose
dtype: object
```

```
# Acceso per índice
print(s)
```

```
print("Imprimimos el valor con índice Al_1:", s['Al_1'])
print("Imprimimos el valor con índice Al_1:", s['Al_3'])
print("Imprimimos los valores de las posiciones 0 y 1: \n", s[['Al_1','Al_3']])
```

```
Al_1      Ana
Al_2      Jose
Al_3      Cristina
dtype: object
Imprimimos el valor con índice Al_1: Ana
Imprimimos el valor con índice Al_1: Cristina
Imprimimos los valores de las posiciones 0 y 1:
Al_1      Ana
Al_3      Cristina
dtype: object
```

▼ Filtrar una Serie

Para filtrar una Serie y quedarse con los valores que cumplen una determinada condición se utiliza el siguiente método:

`s[condición]`: Devuelve una Serie con los elementos de la Serie `s` que se corresponden con el valor `True` de la lista booleana `condición`. `condición` debe ser una lista de valores booleanos de la misma longitud que la Serie.

```
notas = pd.Series({'Ana':9, 'Jose':4, 'Cristina':7.5})
print(notas)
```

```
# Buscamos las notas que sean más grandes que 5
print("\nLos alumnos que han aprobado son: \n")
notas[notas>5]
```

```
Ana      9.0
Jose     4.0
Cristina 7.5
dtype: float64
```

Los alumnos que han aprobado son:

```
Ana      9.0
Cristina 7.5
dtype: float64
```

▼ Ordenar una Serie

Para ordenar una serie se utilizan los siguientes métodos:

`s.sort_values(ascending=boolean)`: Devuelve la Serie `s` ordenada según los valores. Si el argumento del parámetro `ascending` es `True`, el orden es creciente y si es `False` decreciente.

`s.sort_index(ascending=boolean)`: Devuelve la Serie `s` ordenada según el índice. Si el argumento del parámetro `ascending` es `True`, el orden es creciente y si es `False` decreciente.

```
notas = pd.Series({'Ana':9, 'Jose':4, 'Cristina':7.5})
# Ordenamos por valor
print(notas.sort_values(ascending = False))
```

```
Ana      9.0
Cristina 7.5
Jose     4.0
dtype: float64
```

```
# Ordenamos per índice
print(notas.sort_index(ascending = True))
```

```
Ana      9.0
Cristina 7.5
Jose     4.0
dtype: float64
```

▼ Eliminar dadas desconocidas de una Serie

En los estudios no siempre tenemos la información completa, por eso es muy importante aprender a limpiar los datos para poder trabajar con ellos.

En la mayoría de los estudios, la información recopilada está a menudo incompleta o contiene errores. Por esta razón, es crucial saber cómo limpiar los datos para poder trabajar con ellos. Un conjunto de datos limpio y bien estructurado puede proporcionar una base sólida para la interpretación y toma de decisiones informadas.

En Pandas, los datos numéricos desconocidos se representan por NaN* y los valores no numéricos desconocidos por None*.

- NaN es un valor numérico especial que representa "Not a Number". Este valor se utiliza para representar datos faltantes o indefinidos en operaciones matemáticas.
- None, es un valor de objeto que indica la falta de un valor válido en estructuras de datos no numéricas, tales como listas, tuplas y diccionarios.

Tanto unos como otros suelen ser un problema a la hora de realizar algunos análisis de datos, por lo que es habitual eliminarlos. Para eliminarlos de una serie se utiliza el siguiente método:

`s.dropna()`: Elimina los datos desconocidos o nulos de la Serie.

```
import numpy as np # Necesitamos esta librería para crear los Nans
```

```
s = pd.Series(['a', 'b', None, 'c', np.NaN, 'd'])
s = s.dropna()
s
```

```
0    a
1    b
3    c
5    d
dtype: object
```

```
s = s.reset_index(drop=True)
```

```
s
```

```
0    a
1    b
2    c
3    d
dtype: object
```

▼ Ejercicios

▼ Ejercicio 1

Crear una Serie que recoja cuántas mascotas tiene cada compañero de clase. Nombra a esta Serie `mascotasClase`.

¿Cuántos alumnos hay en clase? ¿Cómo saberlo?

Ordena la Serie según el número mascotas, del valor menor al mayor.

¿Qué compañeros tienen más de dos mascotas? ¿Todo el mundo tiene mascota, o hay compañeros que no tienen ninguna?

▼ Ejercicio 2

En la clase de programación se habían inscrito inicialmente 25 alumnos, pero al principio del curso se realizó una reestructuración y ahora el grupo está formado por 17 alumnos. La lista de clase ha quedado de la siguiente manera:

```
import numpy as np
classe = pd.Series(['Sergi', 'Ferran', np.NaN, 'Laura', None, 'Luis', 'Mireia', 'Lorena', None, 'Sara', 'Lluisa', 'David', None, 'Dolors',
                   'Rogelio', None, 'Ramón', np.NaN, 'Maria', 'Júlia', None, 'Isabel', np.NaN, 'Pablo'])
classe.head() # mostramos los primeros valores de la tabla
```

En primer lugar, **limpiala**, elimina aquellos valores desconocidos.

A continuación reescribe **los índices** de forma ascendente.

▼ Ejercicio 3

A continuación tiene **la lista** de artistas más escuchados mensualmente en **el** año 2023. (Datos extraídos **el** 25 de abril de 2023):

```
spotify = pd.Series({'The Weeknd': 107.37, 'Taylor Swift': 85.13, 'Ed Sheeran': 84.15, 'Miley Cyrus': 79.99, 'Shakira': 76.27,
                    'Rihanna': 75.89, 'Ariana Grande': 74.43, 'Justin Bieber': 73.33, 'Drake': 70.95, 'David Guetta': 70.79, 'SZA': 67.34,
                    'Eminem': 66.76, 'Harry Styles': 65.32, 'Bad Bunny': 65.23, 'Calvin Harris': 64.16, 'Coldplay': 63.37,
                    'Dua Lipa': 61.59, 'Sam Smith': 60.67, 'Imagine Dragons': 60.66, '21 Savage': 59.75, 'Bruno Mars': 59.44,
                    'Selena Gomez': 57.87, 'Maroon 5': 55.94, 'Lady Gaga': 55.88, 'Post Malone': 55.16, 'Marshmello': 54.84, 'Karol G': 53.66,
                    'Kanye West': 53.36, 'Metro Boomin': 53.32, 'Doja Cat': 53.08})
spotify.head() # mostramos los primeros valores de la tabla
```

```
The Weeknd    107.37
Taylor Swift   85.13
Ed Sheeran    84.15
Miley Cyrus   79.99
Shakira       76.27
dtype: float64
```

¿Quién es **el** octavo artista más escuchado?, ¿y **el** décimo?

¿Cuántos artistas tienen entre 60 y 70 millones de reproducciones mensuales?

¿Qué artista se encuentra justo en medio de **la** lista?

Anexo VI. Google Colab: Librería Pandas. DataFrame

Se puede acceder a esta actividad en el siguiente enlace:

https://colab.research.google.com/drive/1cB3JQxqANM0yf1Nr4wGqPI4YMsspe_2P?usp=sharing

4. Librería Pandas. DataFrame

Importamos la librería. Cada vez que escribimos 'pd.' Estaremos accediendo a la librería en cuestión:

```
import pandas as pd
```

Recordamos. Estructuras básicas

Las estructuras de datos que más se utilizan de la librería Pandas son:

- **Series:** es una sola columna.
- **DataFrame:** consiste en una tabla de datos relacionales, con filas y columnas con nombre.

Un DataFrame contiene una o más Series y cada Series tiene un nombre.

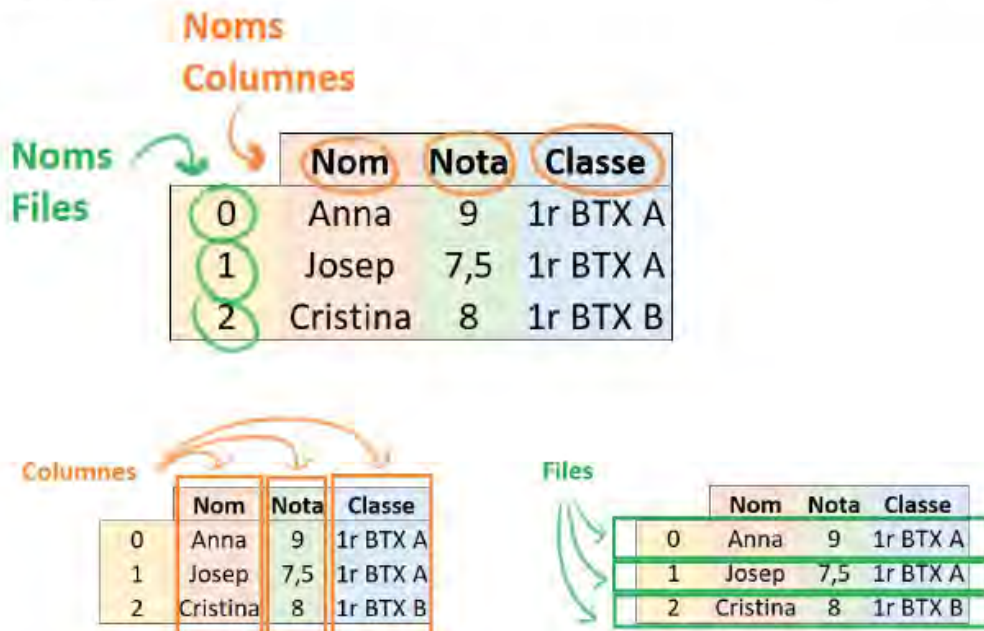
En este documento trabajaremos las Series.

Estas estructuras se identifican por un valor único llamado índice. Este valor se inicia en 0 y cuando hemos generado la tabla siempre aumenta de 1 en 1. Es muy útil para acceder a filas específicas.

DataFrame

Un objeto del tipo DataFrame define un conjunto de datos estructurado en forma de tabla en el que cada columna es un objeto de tipo Series, es decir, todos los datos de una misma columna son del mismo tipo.

Un DataFrame contiene dos índices, uno para las filas y otro para las columnas, pudiendo acceder a sus elementos mediante los nombres de las filas y las columnas.



Creación de DataFrames

Se puede crear un DataFrame a partir de los siguientes elementos:

- **Diccionario de listas.**
- **Lista de diccionarios.**
- **Archivo csv o excel.**

Diccionario de listas

```
pd.DataFrame(archivo=diccionario, indice=filas, columnas=columnas, type=tipo)
```

Las **columnas** son las **listas** contenidas en los **valores** del diccionario.

Los nombres de **filas** están indicados en la **lista de filas**.

Los nombres de **columnas** están indicados en la **lista de columnas**.

Los tipos están indicados en la **lista tipo**.

!! La lista filas debe tener el mismo tamaño que el total de diccionarios, mientras que las listas columnas y tipos deben tener el mismo tamaño que el diccionario.

```
# Creación a partir de un diccionario de listas

# Si no se pasa la lista de filas se utilizan como nombres los enteros comenzando en 0.
# Si no se pasa la lista de columnas se utilizan como nombres las claves del diccionario.
# Si no se pasa la lista de tipos, se deduce.

dades = {'Nombre':['Ana', 'Jose', 'Cristina'],
         'Nota':[9, 7.5, 8],
         'Clase':['1r BCH A', '1r BCH A', '1r BCH B' ]}

df = pd.DataFrame(dades)
df
```

	Nombre	Nota	Clase
0	Ana	9.0	1rBCHA
1	Jose	7.5	1rBCHA
2	Cristina	8.0	1rBCHB

Lista de diccionarios

```
pd.DataFrame(archivo=diccionarios, indice=filas, columnas=columnas, type=tipo)
```

Las **filas** son los **valores** de los diccionarios.

Los nombres de **filas** están indicados en la **lista de filas**.

Los nombres de **columnas** están indicados en la **lista de columnas**.

Los tipos están indicados en la **lista tipo**.

!! La lista filas debe tener el mismo tamaño que el total de diccionarios.

!! Si los diccionarios no tienen las mismas claves, las claves que no aparecen en el diccionario se rellenan con valores NaN.

```
# Creación a partir de una lista de diccionarios

# Si no se pasa la lista de filas se utilizan como nombres los enteros comenzando en 0.
# Si no se pasa la lista de columnas se utilizan como nombres las claves del diccionario.
# Si no se pasa la lista de tipos, se deduce.

dades = [{'Nombre':'Ana', 'Nota':9, 'Clase': '1r BCH A'},
         {'Nombre':'Jose', 'Nota':7.5, 'Clase':'1r BCH A'},
         {'Nombre':'Cristina', 'Nota':8, 'Clase':'1r BCH B'}]

df = pd.DataFrame(dades)
df
```

	Nombre	Nota	Clase
0	Ana	9.0	1rBCHA
1	Jose	7.5	1rBCHA
2	Cristina	8.0	1rBCHB

Archivo CSV o excel

Dependiendo del tipo de archivo, existen diferentes funciones para importar un DataFrame desde un archivo:

- Archivo CSV:

```
read_csv(archivo.csv, sep=separador, header=n, index_col=m)
```

Los separadores más utilizados son ';' o ','.

Como nombres de columnas se utiliza los valores de la fila n y como nombres de filas los valores de la columna m, si no se indica son 0.

- Archivo Excel

```
read_excel(archivo.xlsx, header=n, index_col=m)
```

Como nombres de columnas se utiliza los valores de la fila n y como nombres de filas los valores de la columna m, si no se indica son 0.

Se pueden encontrar los conjunto de datos en los siguientes enlaces:

csv https://drive.google.com/file/d/1zOtcaB7QlyW_rZI6PlhaKHseHHb-mBrj/view?usp=sharing

xlsx

https://docs.google.com/spreadsheets/d/1jb8tlvgqxlP4y_sGgt85ayOtxOGfhE/edit?usp=sharing&ouid=116154351428130274414&rtpof=true&sd=true

```
# Creación a partir de un fichero CSV
```

```
df = pd.read_csv('/content/2021_ALTAS_NACIMIENTOS.csv', sep=';')  
df.head()
```

	CODIGO_DIVISION_TERRITORIAL	BARRIO	HOMBRES	MUJERES	TOTAL
0	B-01	Centre	33	26	59
1	B-02	Coll i Pujol	13	15	28
2	B-03	Dalt de la Vila	15	14	29
3	B-04	Gorg	34	25	59
4	B-05	Progrés	34	41	75

```
# Creación a partir de un fichero xlsl
```

```
df = pd.read_excel('/content/2021_ALTAS_NACIMIENTOS.xlsx')
df.head()
```

	CODIGO_DIVISION_TERRITORIAL	BARRIO	HOMBRES	MUJERES	TOTAL
0	B-01	Centre	33	26	59
1	B-02	Coll i Pujol	13	15	28
2	B-03	Dalt de la Vila	15	14	29
3	B-04	Gorg	34	25	59
4	B-05	Progrés	34	41	75

▼ Atributos de un DataFrame

Un DataFrame tiene distintas propiedades:

`df.info()` : Devuelve información (número de filas, número de columnas, índices, tipos de las columnas y memoria utilizada) del DataFrame `df`.

`df.shape` : Devuelve una tupla con el número de filas y columnas del DataFrame `df`.

`df.size` : Devuelve el número de elementos de DataFrame.

`df.columns` : Devuelve una lista con los nombres de las columnas del DataFrame `df`.

`df.index` : Devuelve una lista con los nombres de las filas de DataFrame `df`.

`df.dtypes` : Devuelve una Serie con los tipos de datos de las columnas del DataFrame `df`.

`df.head(n)` : Devuelve las `n` primeras filas del DataFrame `df`.

`df.tail(n)` : Devuelve las `n` últimas filas del DataFrame `df`.

```
df.info()
```

```
df.shape
```

```
df.size
```

```
df.columns
```

```
df.index
```

```
df.dtypes
```

```
df.head(7)
```

```
df.tail(7)
```

▼ Acceso a los elementos de un DataFrame

A un Dataframe podemos acceder según:

- La posición

`df.iloc[i, j]` : Devuelve el elemento que se encuentra en la fila *i* y la columna *j* del DataFrame *df*.

`df.iloc[files, columnas]` : Devuelve un DataFrame con los elementos de las filas de la lista *files* y de las columnas de la lista *columnas*.

`df.iloc[i]` : Devuelve una Serie con los elementos de la fila *i* del DataFrame *df*.

```
# Acceso por posición
print(df.head(10))
print("\n")
print("Imprimimos el valor de la fila 0 y columna 1:", df.iloc[0,1])
print("Imprimimos el valor de la fila 7 y columna 2:", df.iloc[7,2])
print("Imprimimos el valor de las tres primeras filas y la columna 3: \n", df.iloc[0:3,3])
print("Imprimimos la fila 4: \n", df.iloc[4])
```

- El nombre de las filas y columnas

`df.loc[nombreFila, nombreColumna]` : Devuelve el elemento que se encuentra en la fila con nombre *nombreFila* y la columna de con nombre *nombreColumna* de

DataFrame df.

`df.loc[nombreFilas, nombreColumnas]` : Devuelve un DataFrame con los elementos que se encuentran en las filas con los nombres de la lista *nombreFilas* y las columnas con los nombres de la lista *nombreColumnas* de DataFrame df.

`df[nombreColumna]` : Devuelve una Serie con los elementos de la columna de nombre *nombreColumna* del DataFrame df.

```
# Acceso por nombre de filas y columnas
print(df.head(10))
print("\n")
print("Imprimimos el valor de la fila 0 y columna BARRI:", df.loc[0,'BARRI'])
print("Imprimimos el valor de la fila 7 y columna HOMES:", df.loc[7,'HOMES'])
print("Imprimimos el valor de las tres primeras filas y las columnas HOMES i DONES: \n", d
print("Imprimimos la columna TOTAL: \n", df['TOTAL'])
```

▼ Filtrar un DataFrame

Para filtrar un DataFrame y quedarse con los valores que cumplen una determinada condición se utiliza los siguientes métodos:

- Filtrar por columna

Podemos seleccionar una columna.

```
df['nombreColumna']
```

Pero si queremos unas cuantas filas únicamente, podemos especificar cuáles:

```
df['nombreColumna'][filas]
```

- Filtrar por condiciones

Para filtrar con condiciones, debemos escribir la condición dentro de corchetes.

```
df[df['nombreColumna'] comparación]
```

Se crea una matriz de valores booleanos True o False, de acuerdo con la condición expresada (`df['nombreColumna']comparación`). Esta matriz es pasada al dataframe df y se queda con las celdas que son True.

- Filtrar con más condiciones

Para filtrar con más condiciones, debemos escribir las diferentes condiciones uniéndolas mediante la unión, |, o la intersección, &.

```
df[condicio1 | condicio2]
```

```
df[condicio1 & condicio2]
```

```
df.head()
```

```
# Filtrar toda una columna  
df['TOTAL']
```

```
# Filtramos las 7 primeras filas de una columna  
df['TOTAL'][0:7]
```

```
# Filtramos por condiciones  
df[df['DONES']>=60]
```

```
# Filtrar con más condiciones  
df[(df['HOMES']>=50) | (df['DONES']>=50)]
```

▼ Ordenar un DataFrame

Para ordenar un DataFrame se utilizan los siguientes métodos:

```
df.sort_values(by=nombreColumna, axis=0, ascending=True)
```

by: Nombre o lista de nombres para ordenar.

axis: Si axis es 0 o 'índice' corresponde a etiquetas de columna. Si el eje es 1 o "columnas", corresponde a etiquetas de índice.

```
df.sort_values(by='TOTAL', axis=0, ascending=False)
```

▼ Eliminar datos desconocidos de un DataFrame

En los estudios no siempre tenemos la información completa, por eso es muy importante aprender a limpiar los datos para poder trabajar con ellos.

En la mayoría de los estudios, la información recopilada está a menudo incompleta o contiene errores. Por esta razón, es crucial saber cómo limpiar los datos para poder trabajar con ellos. Un conjunto de datos limpio y bien estructurado puede proporcionar una base sólida para la interpretación y toma de decisiones informadas.

En Pandas, los datos numéricos desconocidos se representan por NaN* y los valores no numéricos desconocidos por None*.

- *NaN* es un valor numérico especial que representa "Not a Number". Este valor se utiliza para representar datos faltantes o indefinidos en operaciones matemáticas.
- *None*, es un valor de objeto que indica la falta de un valor válido en estructuras de datos no numéricas, tales como listas, tuplas y diccionarios.

Tanto unos como otros suelen ser un problema a la hora de realizar algunos análisis de datos, por lo que es habitual eliminarlos. Para eliminarlos de una serie se utiliza el siguiente método:

`df.dropna()` : Elimina los datos desconocidos o nulos de la Series.

```
import numpy as np # Necesitamos esta libreria para crear los NaNs

df=df.dropna()
df.head()
```

Fíjese en cómo ha quedado el índice, si nos conviene podemos resetearlos:

```
df = df.reset_index(drop=True)
df.head()
```

▼ Añadir y Eliminar columnas y filas

▼ Añadir columnas

```
# Declaramos la lista que será la nueva columna

nuevaColumna = ['Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona',
                'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona',
                'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona',
                'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona', 'Badalona']

# Añadimos la nueva columna
```

```
df['CIUTAT'] = nuevaColumna
```

```
df.head()
```

	CODIGO_DIVISION_TERRITORIAL	BARRIO	HOMBRES	MUJERES	TOTAL	CIUTAT
0	B-01	Centre	33	26	59	Badalona
1	B-02	Coll i Pujol	13	15	28	Badalona
2	B-03	Dalt de la Vila	15	14	29	Badalona
3	B-04	Gorg	34	25	59	Badalona
4	B-05	Progrés	34	41	75	BadalonaBadalona

▼ Eliminar Columnas

`del df[nombreColumna]` : Elimina la columna de nombre `nombreColumna` del DataFrame `df`.

```
del df['CIUTAT']  
df.head()
```

▼ Añadir Filas

Para añadir una fila a un DataFrame se utiliza el siguiente método:

```
df.append(Serie, ignore_index=True)
```

Los nombres del índice de la Serie deben corresponderse con los nombres de las columnas de `df`.

```
s = pd.Series(['B-Test', 'Test', 1, 2, 3], index=['CODI_DIVISIO_TERRITORIAL', 'BARRI', 'HOMES', 'DONES', 'TOTAL'])  
print(s)  
df = df.append(s, ignore_index=True)  
df = df.append(s, ignore_index=True)  
df.tail()
```

▼ Eliminar Filas

```
print(df.tail())  
df = df.drop([34,35]) # Indicamos las filas que queremos eliminar  
df.tail()
```

▼ Ejercicios

Genera un DataFrame para recopilar información sobre tus compañeros de clase. Por ejemplo, detalles sobre sus mascotas, deportes que practican, sus series favoritas. El tema es de elección libre.

Una vez que hayas creado el DataFrame, realiza un análisis estadístico del mismo y extrae algunas conclusiones. Por ejemplo, identifica el valor más frecuente o si existe alguna correlación.

Anexo VII. Presentación: “Una buena visualización”

Se puede acceder a la presentación ‘Una Buena Visualización’ en el siguiente enlace:

<https://view.genial.ly/6452695d274b2b0012b2dfd0/presentation-una-buena-visualizacion>

UNA BUENA VISUALIZACIÓN

EMPEZAR ▶

NATÀLIA CONESA

ÍNDICE

1. ¿Qué es la visualización de datos?
2. Visualización de información
3. Audiencia
4. Color
5. Qué no hacer

The screenshot shows a Genially presentation interface. The title 'UNA BUENA VISUALIZACIÓN' is prominently displayed in large, bold, black letters. Below the title is a blue button with the text 'EMPEZAR ▶'. The author's name 'NATÀLIA CONESA' is visible in the top right corner. The presentation content includes a table of contents with five items: '1. ¿Qué es la visualización de datos?', '2. Visualización de información', '3. Audiencia', '4. Color', and '5. Qué no hacer'. The background is decorated with various data visualization charts, including line graphs, bar charts, and pie charts, in shades of blue, green, and yellow. The Genially logo is visible in the bottom left corner.

1. ¿QUÉ ES LA VISUALIZACIÓN DE DATOS?

La visualización de datos es la práctica de traducir información a un contexto visual para hacer que los datos sean más fáciles para el cerebro humano.



genially

2. VISUALIZACIÓN DE INFORMACIÓN

La visualización de información es el proceso de representar datos de una manera visual y significativa para que un usuario pueda comprenderlos mejor.



Campañas publicitarias de Spotify

genially

3. AUDIENCIA

IMPORTANCIA DE LA AUDIENCIA

Considerar a la audiencia al crear visualizaciones de datos es de vital importancia. ayuda a garantizar que la información sea comprensible y relevante para el público objetivo.

NECESIDADES E INTERESES

Diferentes audiencias pueden tener diferentes necesidades e intereses.

ACCESIBILIDAD Y COMPRENSIÓN

La visualización ha de ser accesible y comprensible para todos los usuarios.

genially

4. COLOR

PALETAS

Cómo elegir colores para visualizaciones de datos

<https://chartio.com/learn/charts/how-to-choose-colors-data-visualization/>

HERRAMIENTAS

<https://colorbrewer2.org/>

<https://www.learnui.design/tools/data-color-picker.html>

genially

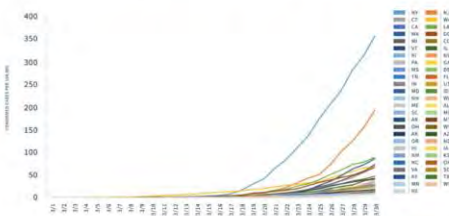
5. QUÉ NO HACER

A continuación veréis unos ejemplos de malas prácticas en la visualización de datos

genially

MUCHA INFORMACIÓN

CUMULATIVE CASES PER 100,000: ALL STATES



<https://twitter.com/Scavino45/status/1245112211829334018>

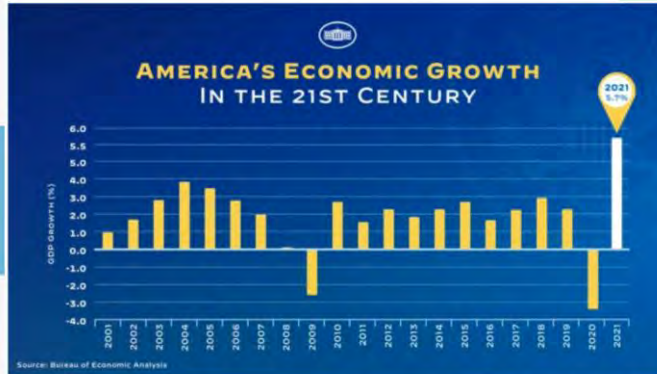
CORONAVIRUS IN THE USA



<https://twitter.com/visualisingdata/status/1238407753258963428>

genially

ESCALAS INADECUADA



<https://twitter.com/jschwabish/status/1488267927405342722>

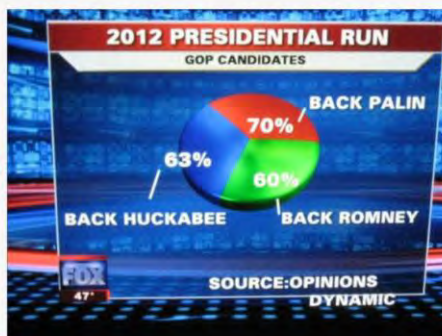
genially

ESCALAS INADECUADA



genially

MATEMÁTICA BÁSICA



Fox News

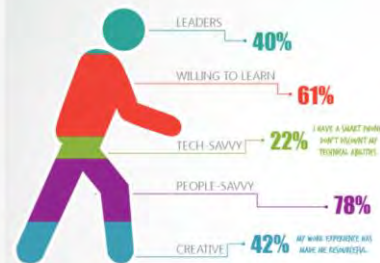


<https://twitter.com/MarkBradbourne/status/1243264357406068744>

genially

MATEMÁTICA BÁSICA

HOW BABY BOOMERS DESCRIBE THEMSELVES



<https://twitter.com/MaartenSmeden/status/1442765690591760384>

VOTERS TRUST TRUMP OVER CLINTON

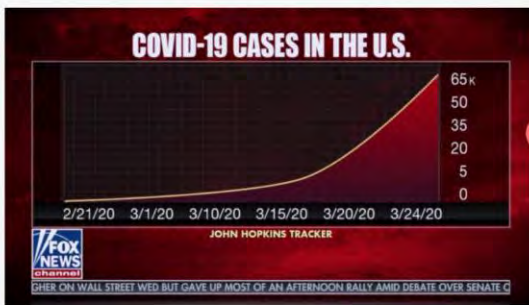


<https://twitter.com/BarMartland/status/1384941249560522752>

MANIPULACIÓN DE LOS EJES Y-X



<https://twitter.com/Birdbassador/status/1246439050426363906>



Fox News

CONFUNDIR AL USUARIO



<https://twitter.com/dsn/status/123776289907515395>

Anexo VIII. Google Colab: Visualización de datos

Se puede acceder a esta actividad en el siguiente enlace:

<https://colab.research.google.com/drive/1PnN9xyPLMnYtAxho-mNOUURd5x-gdgZB?usp=sharing>

5. Visualización de datos

En este documento encontraréis un pequeño resumen sobre cómo visualizar a partir de la librería matplotlib.

Realizaremos distintos gráficos, y comentaremos cuándo usar cada uno de ellos.

En primer lugar, necesitamos importar las librerías necesarias

```
# Librería matplotlib
import matplotlib.pyplot as plt

# Librería pandas
import pandas as pd
```

En este documento seguiremos trabajando con el DataFrame de altas de nacimientos en Badalona.

Se puede encontrar el conjunto de datos en el enlace:

https://docs.google.com/spreadsheets/d/1jb8tlvgqxl_pS4v_sGgtn85ayOtxOGfhF/edit?usp=sharing&ouid=116154351428130274414&rtpof=true&sd=true

```
df = pd.read_csv('/content/2021_ALTAS_NACIMIENTOS.csv', sep=';')
df.head()
```

	CODIGO_DIVISION_TERRITORIAL	BARRIO	HOMBRES	MUJERES	TOTAL	
0	B-01	Centre	33	26	59	
1	B-02	Coll i Pujol	13	15	28	
2	B-03	Dalt de la Vila	15	14	29	
3	B-04	Gorg	34	25	59	
4	B-05	Progrés	34	41	75	

Gráfico de barras

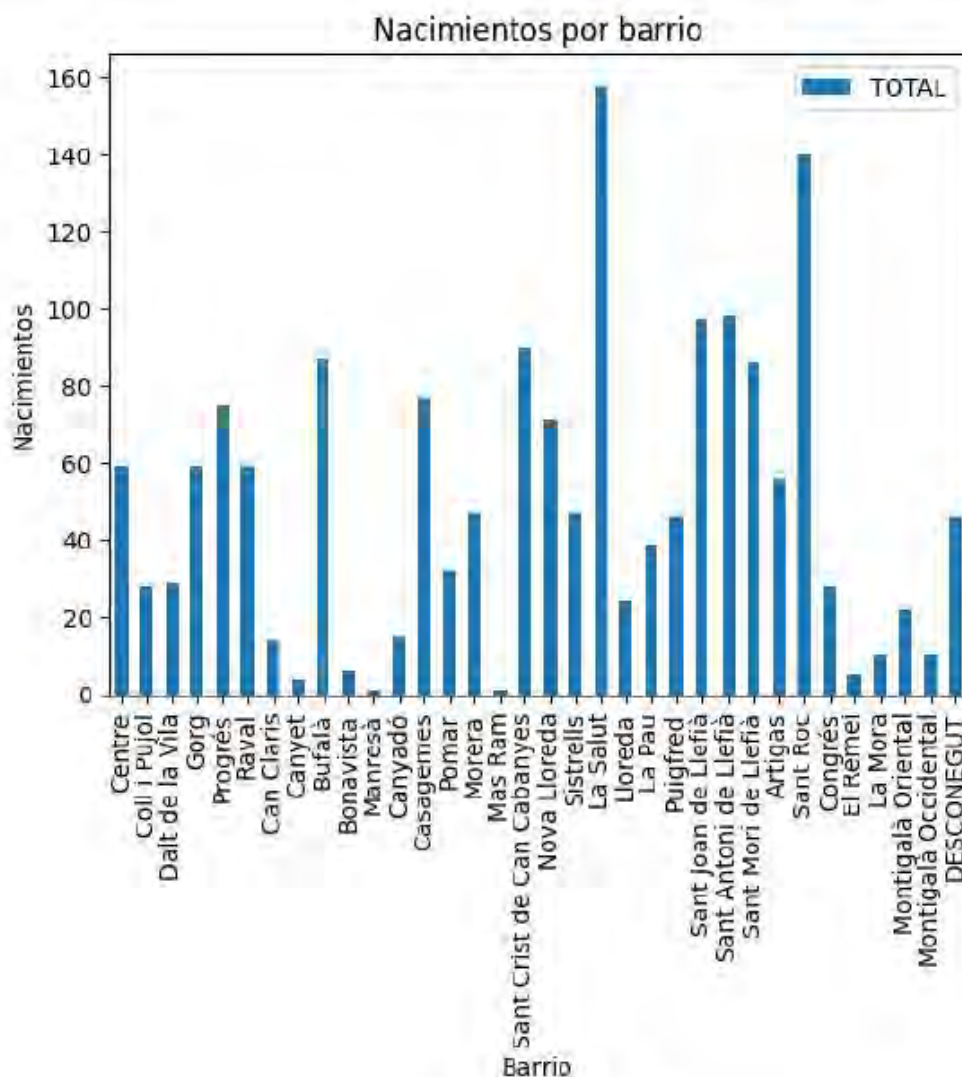
Un gráfico de barras es un tipo de gráfico que se utiliza para mostrar y **comparar** diferentes medidas para diferentes categorías de datos o series de datos.

Este tipo de gráfico puede tener una orientación horizontal o vertical.

Usamos el método `plot()` del `DataFrame` y especificamos el tipo de gráfico que se quiere dibujar, en este caso, un gráfico de barras. Para indicarlo al parámetro `kind` le asignamos el valor `'bar'`.

También puedes especificar otras opciones, como el título del gráfico y las etiquetas de los ejes:

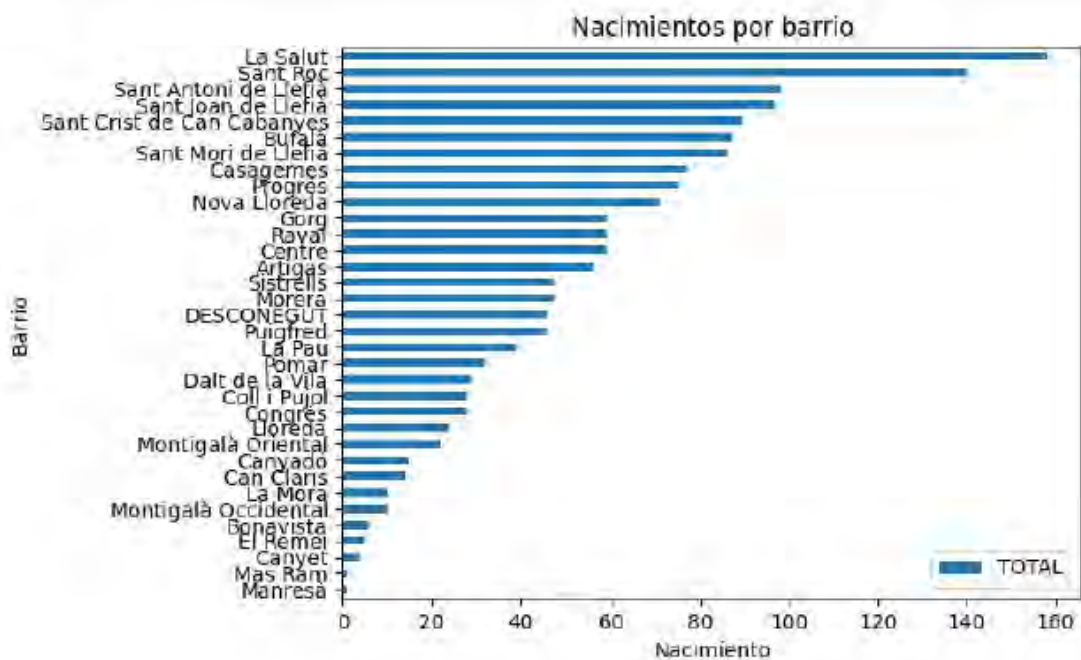
```
# Dibujamos el gráfico
df.plot(kind='bar', x='BARRIO', y='TOTAL', title='Nacimientos por barrio')
# Personalizamos el gráfico
plt.xlabel('Barrio')
plt.ylabel('Nacimientos')
plt.show()
```



Este gráfico consta de dos ejes: un eje cuantitativo, muestra la escala de los valores; y un eje cualitativo, indica la categoría a la que pertenecen los datos. Cada categoría está representada por una barra.

Para hacer el gráfico de barras horizontal indicaremos el valor del parámetro kind como 'barh':

```
# Ordenamos el DataFrame
df_ordenat = df.sort_values('TOTAL', ascending=True)
# Dibujamos el gráfico
df_ordenat.plot(kind='barh', x='BARRIO', y='TOTAL', title='Nacimientos por barrio')
# Personalizamos el gráfico
plt.xlabel('Nacimiento')
plt.ylabel('Barrio')
plt.show()
```



El gráfico de barras horizontales es muy útil cuando se quiere comparar un conjunto de valores y establecer un ranking. No solo muestra los valores más altos y más bajos, sino que también ayuda a dar una impresión de la diferencia entre unos valores y otros.

!! El eje cuantitativo debe comenzar siempre en cero, porque, de lo contrario, las diferencias de valores se magnifican y engañan al lector.

↳ Gráfico de barras agrupadas

El gráfico de barras agrupadas es una variación del gráfico de barras. Permite comparar dos o más conjuntos de datos relacionados en una misma visualización. En este tipo de gráfico, las


barras se agrupan por categorías y se muestran en diferentes colores para facilitar la comparación entre los conjuntos de datos.

!! Los datos han de estar en la misma escala para poderlos comparar

```
df.head()
```

	CODIGO_DIVISION_TERRITORIAL	BARRIO	HOMBRES	MUJERES	TOTAL	
0	B-01	Centre	33	26	59	
1	B-02	Coll i Pujol	13	15	28	
2	B-03	Dalt de la Vila	15	14	29	
3	B-04	Gorg	34	25	59	
4	B-05	Progrés	34	41	75	

```
# Construimos un nuevo dataframe con los datos que queremos mostrar
df_agrupado = pd.DataFrame(data=df[['HOMBRES', 'MUJERES']].values, index=df['BARRIO'], col
df_agrupado
```

	HOMBRES	MUJERES	
BARRIO			
Centre	33	26	
Coll i Pujol	13	15	
Dalt de la Vila	15	14	
Gorg	34	25	
Progrés	34	41	
Raval	34	25	
Can Claris	7	7	
Canyet	3	1	
Bufalà	46	41	
Bonavista	5	1	
Manresà	0	1	
Canyadó	6	9	
Casagemes	37	40	
Pomar	21	11	
Morera	22	25	

```
# Visualizamos
df_agrupado.plot.bar()
```

<Axes: xlabel='BARRIO'>

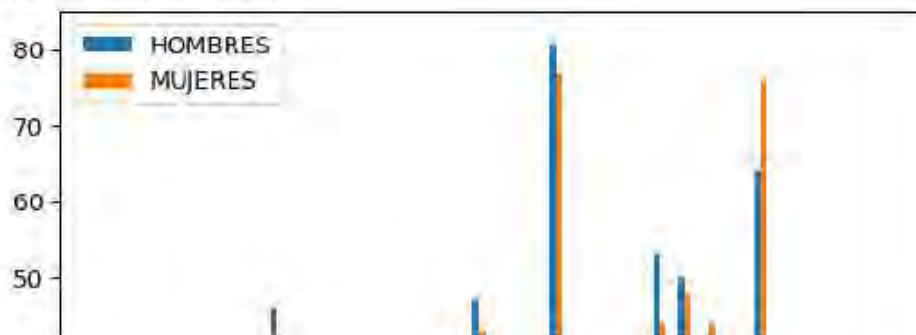


Gráfico de Líneas



Un gráfico de **líneas** es un tipo de gráfico que representa la relación entre dos variables numéricas a través de una línea continua que conecta los puntos de datos.

En un gráfico de **líneas**, la variable independiente se traza en el eje x (horizontal) y la variable dependiente se traza en el eje y (vertical).

Los gráficos de **líneas** son útiles para **mostrar tendencias y patrones** en los datos.

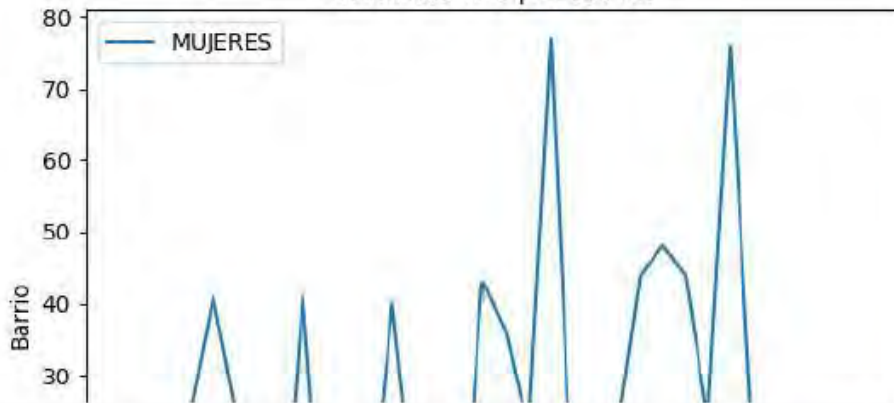
En un gráfico de **líneas** también se pueden utilizar **múltiples líneas** para representar diferentes variables o para **comparar las tendencias** de varias series de datos.

Usamos el método `plot()` del `DataFrame` y especificamos el tipo de gráfico que se quiere dibujar, en este caso, un gráfico de **líneas**. Para indicarlo al parámetro `kind` le asignamos el valor `'line'`.

π

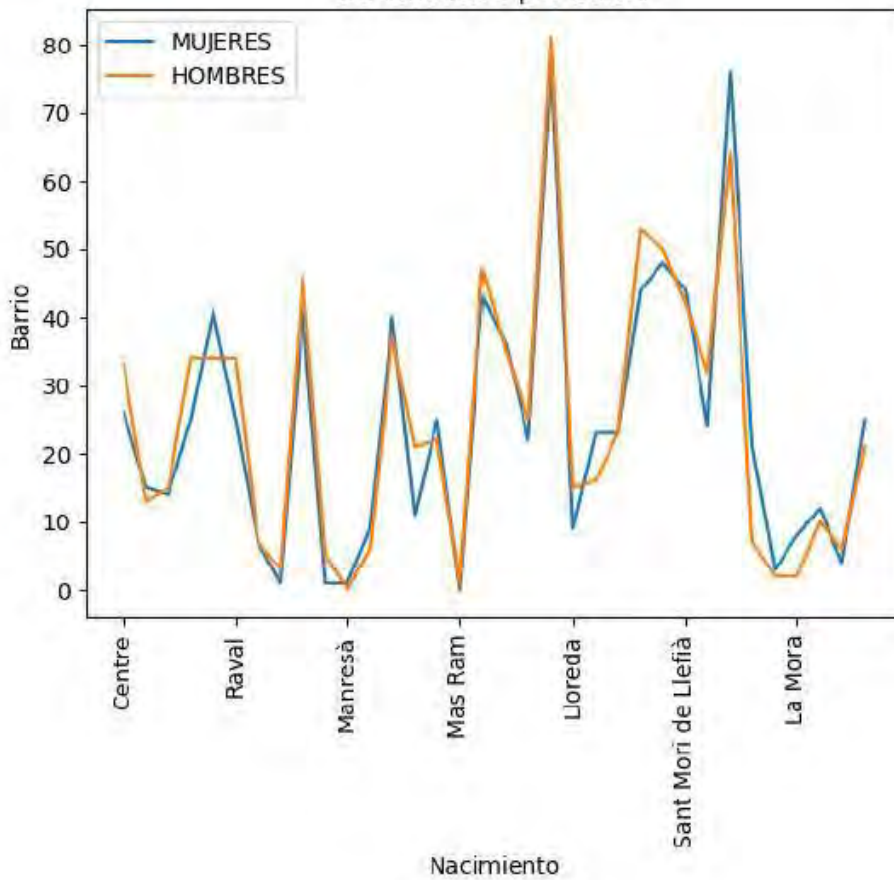
```
# Dibujamos el gráfico
df.plot(kind='line', x='BARRIO', y='MUJERES', title='Nacimientos por barrio', rot=90)
# Personalizamos el gráfico
plt.xlabel('Nacimiento')
plt.ylabel('Barrio')
plt.show()
```

Nacimientos por barrio



```
# Dibujamos el gráfico
df.plot(kind='line', x='BARRIO', y=['MUJERES','HOMBRES'], title='Nacimientos por barrio',
# Personalizamos el gráfico
plt.xlabel('Nacimiento')
plt.ylabel('Barrio')
plt.show()
```

Nacimientos por barrio



▼ Gráfico de puntos

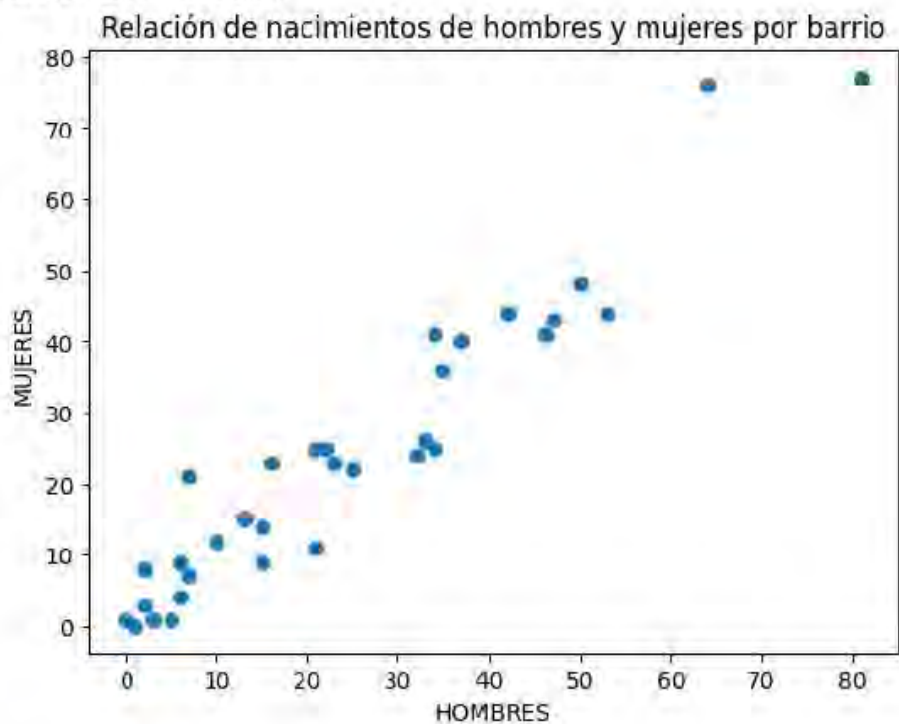
Un gráfico de puntos (también conocido como gráfico de dispersión) se utiliza para representar la relación entre dos variables numéricas.

Cada punto en el gráfico representa un valor de las dos variables. La posición en el eje horizontal corresponde a un valor de una variable, mientras que la posición en el eje vertical corresponde a un valor de la otra variable.

Los gráficos de puntos son útiles para **explorar la correlación o la asociación entre dos variables y para identificar posibles patrones o tendencias en los datos.**

```
fig, ax = plt.subplots()
# Dibujamos el gráfico de puntos
ax.scatter(df['HOMBRES'], df['MUJERES'])
# Personalizamos el gráfico
ax.set_title('Relación de nacimientos de hombres y mujeres por barrio')
ax.set_xlabel('HOMBRES')
ax.set_ylabel('MUJERES')
```

```
Text(0, 0.5, 'MUJERES')
```



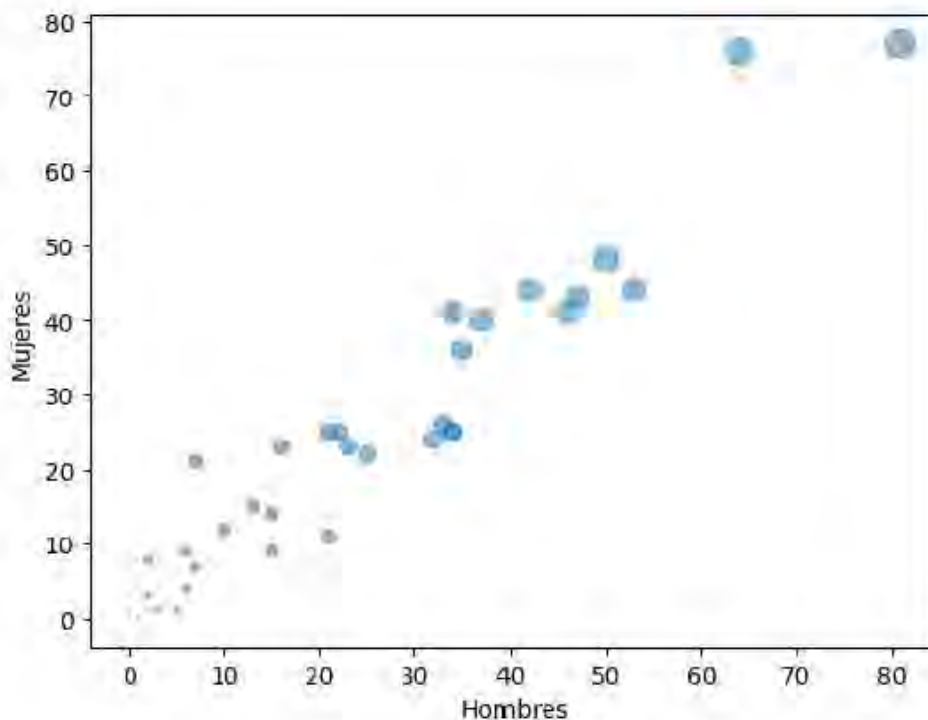
▼ Gráfico de burbujas

Un gráfico de burbujas es un gráfico multivariable que utiliza burbujas para representar los datos en lugar de puntos simples.

Si estamos representando la relación entre dos variables numéricas (como en un gráfico de dispersión), el tamaño de las burbujas puede representar un tercer valor, como la frecuencia o la importancia relativa de los datos.

Los gráficos de burbujas se utilizan normalmente para **comparar y mostrar las relaciones**.

```
# Dibujamos el gráfico de burbujas
plt.scatter(df['HOMBRES'], df['MUJERES'], s=df['TOTAL'], alpha=0.5)
# Personalizamos el gráfico
plt.xlabel('Hombres')
plt.ylabel('Mujeres')
plt.show()
```



▼ Gráfico circular

Un gráfico circular, **es utilizado para mostrar proporciones o porcentajes de diferentes categorías**. Este gráfico consiste en una circunferencia dividida en secciones proporcionales a los valores que se están representando.

Es especialmente adecuado cuando se quiere **comparar** las proporciones entre categorías de una muestra o población.

```

# Seleccionar la columna con los valores a graficar
valores = df['TOTAL']

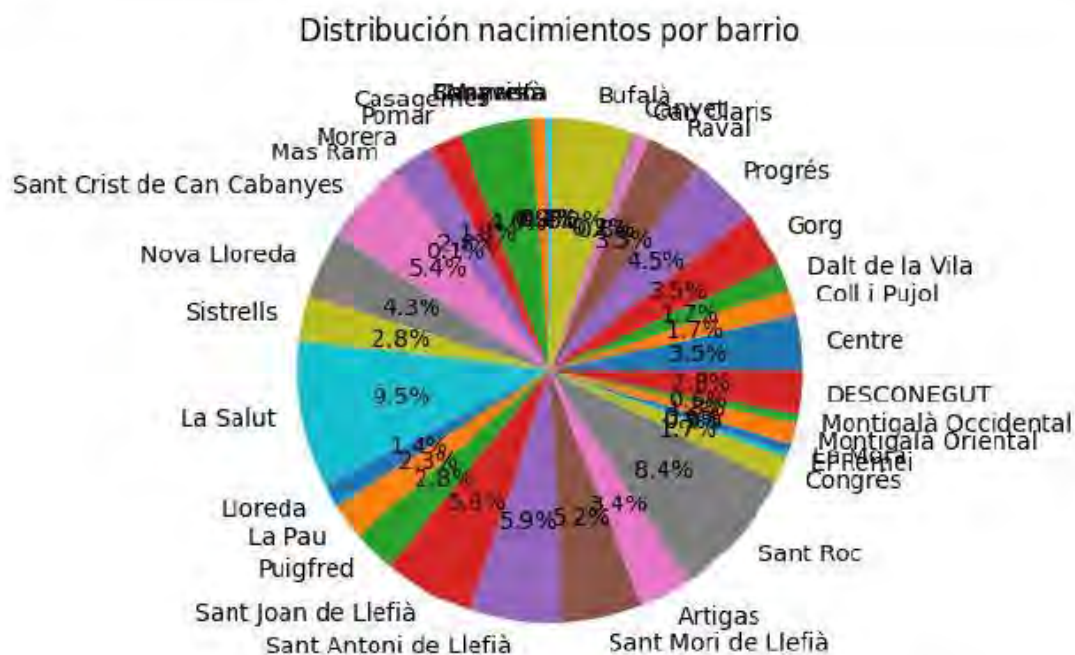
# Crear la figura y el eje para el gráfico circular
fig, ax = plt.subplots()

# Generar el gráfico circular
ax.pie(valores, labels=df['BARRIO'], autopct='%1.1f%%')

# Agregar un título y leyendas
ax.set_title('Distribución nacimientos por barrio')
# ax.legend(df['BARRIO'])

# Mostrar el gráfico
plt.show()

```

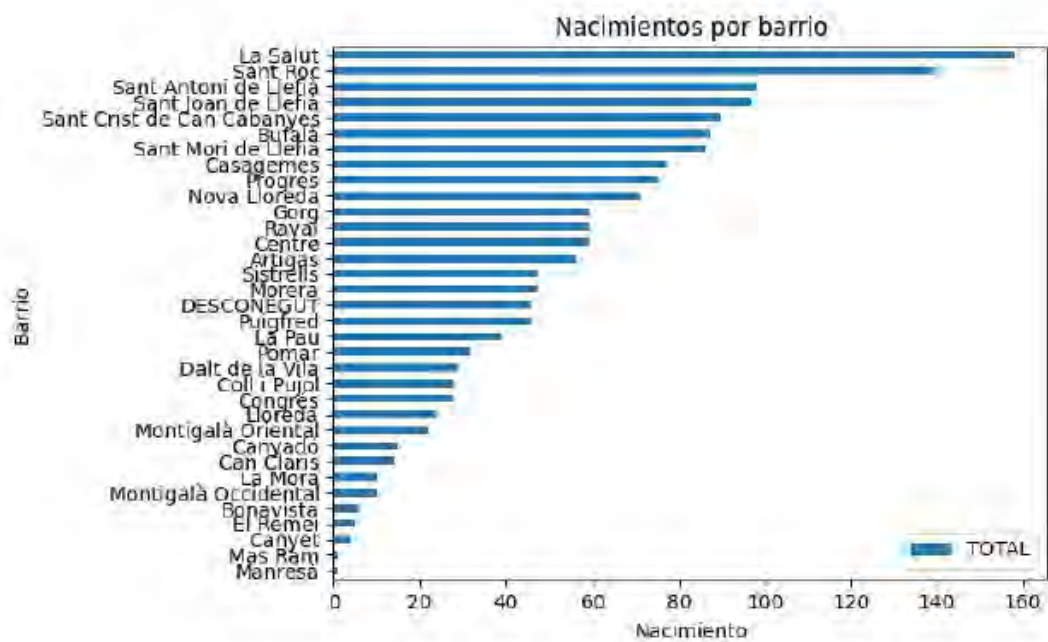


El gráfico que acabamos de mostrar no es una buena visualización. En los casos en que queremos comparar muchos valores, no es recomendable utilizar un gráfico circular para representar más de dos o tres partes de un total. En ese caso, sería mucho mejor utilizar un gráfico de barras.

```

# Ordenamos el DataFrame
df_ordenat = df.sort_values('TOTAL', ascending=True)
# Dibujamos el gráfico
df_ordenat.plot(kind='barh', x='BARRIO', y='TOTAL', title='Nacimientos por barrio')
# Personalizamos el gráfico
plt.xlabel('Nacimiento')
plt.ylabel('Barrio')
plt.show()

```



↳ Ejercicios

A partir del DataFrame creado en la actividad anterior, visualiza los datos utilizando los gráficos que mejor se adecúen al tipo de información que deseas transmitir. Selecciona los tipos de gráficos más apropiado para resaltar los patrones, relaciones o distribuciones de los datos.

Anexo IX. Google Colab: Estadística con DataFrames

Se puede acceder a esta actividad en el siguiente enlace:

https://colab.research.google.com/drive/1igKxnXg2-XNrCeJ381Fic_4LcWsl7Q7y?usp=sharing

6. Estadística con DataFrames

Importamos la librería. Cada vez que escribamos 'pd' estaremos accediendo a la librería pandas:

```
import pandas as pd
```

En esta primera parte, una vez recordados algunos conceptos estadísticos en clase, vamos a ver como los podemos aplicar en un DataFrame.

El DataFrame que usaremos en este documento consiste en el conjunto de propinas que han recibido los empleados de un restaurante. Este conjunto de datos se puede encontrar en la librería seaborn y consta de 244 registros y 7 características: la cuenta en dólares, la propina en dólares, el sexo del pagador de la cuenta, si había fumadores en la fiesta, el día de la semana, la hora del día y el tamaño de la fiesta.

```
from seaborn import load_dataset
df = load_dataset("tips")
df.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16,99	1,01	Female	No	Sun	Dinner	2
1	10,34	1,66	Male	No	Sun	Dinner	3
2	21,01	3,50	Male	No	Sun	Dinner	3
3	23,68	3,31	Male	No	Sun	Dinner	2
4	24,59	3,61	Female	No	Sun	Dinner	4

Podemos aplicar en las distintas columnas cada una de las siguientes funciones:

- df.count(): Cuenta los valores no nulos para cada columna o fila.
- df.sum(): Devuelve la suma de los valores sobre el eje solicitado.
- df.cumsum(): Devuelve la suma acumulada sobre un eje DataFrame o Series.
- df.unique(): Devuelve valores únicos según su orden de aparición. (NO ordena).
- df.min(): Devuelve el mínimo de los valores sobre el eje solicitado.
- df.max(): Devuelve el máximo de los valores sobre el eje solicitado.
- df.mean(): Devuelve la media de los valores sobre el eje solicitado.
- df.var(): Devuelve la varianza sobre el eje solicitado.
- df.std(): Devuelve la desviación estándar de la muestra sobre el eje solicitado.
- df.cov(): Devuelve las covarianzas de los datos de las columnas numéricas, excluyendo los valores nulos.
- df.corr(): Devuelve la correlación por pares de columnas, excluyendo los valores nulos.
- df.describe(): Devuelve un DataFrame con un resumen estadístico de las columnas del DataFrame df.

```
# ¿Cuántas propinas tengo?
df["tip"].count()
```

```
244
```

```
# ¿Cuánto se deja de propina de media?
df["tip"].mean()
```

```
2.99827868852459
```

```
# ¿Existe correlación entre las variables?
df.corr()
```



```
df.corr()

```

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315

Hacemos un resumen de los diferentes datos estadísticos
df.describe()

```
df.describe()

```

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.988279	2.669672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.582500	3.000000
max	50.810000	10.000000	6.000000

Sin embargo, podemos obtener información más detallada y útil si trabajamos un poco los datos. Podemos agrupar los datos para analizarlos por categorías específicas.

↳ Agrupación de un DataFrame

En muchas aplicaciones es útil agrupar los datos de un DataFrame de acuerdo con los valores de una o varias columnas, como por ejemplo la nota, el sexo, la clase, ...



Para dividir un DataFrame en grupos se utiliza el siguiente método:

```
df.groupby(columnasAgrupacion).groups
```

Devuelve un diccionario con las combinaciones columnasAgrupacion agrupadas

```
# Agrupamos por día
print("\n Agrupamos por día: ")
grupos = df.groupby(['day']).groups
print(grupos)
```

```
Agrupamos por día:
{'Thu': [77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
```

A estas agrupaciones se les puede aplicar funciones, por ejemplo, calcular la media de propina por día

```
# Agrupamos por día y seleccionamos la columna que queremos calcular la media.
# Si no selecciono la columna, calcularía la media de todas las columnas numéricas.
df.groupby(['day'])['tip'].mean()
```

```
day
Thur    2.771452
Fri     2.734737
Sat     2.993103
Sun     3.255132
Name: tip, dtype: float64
```

También se puede hacer una **doble** agrupación. Por ejemplo, podemos contar cuántos fumadores hay según el **sexo**

```
# Grupo por sexo y cuento cuántos fumadores hay
df.groupby(['sex', 'smoker'])['tip'].count()
```

```
sex    smoker
Male   Yes     60
       No     97
Female Yes     33
       No     54
Name: tip, dtype: int64
```

▼ Tablas dinámicas

Una **tabla dinámica** es una herramienta que permite reorganizar y resumir grandes cantidades de datos en una **tabla** más compacta y fácil de analizar.

En una **tabla dinámica**, los datos se organizan en **filas** y **columnas**, y se pueden aplicar diferentes funciones de resumen (como la **suma**, el **promedio**, el **mínimo**, el **máximo**, etc.) para resumir los datos de interés.

Para convertir un DataFrame en otro, intercambiando **filas** y **columnas**, se utiliza el siguiente método:

```
df.pivot_table(index=filas, columns=columna, values=valores, aggfunc=funcion)
```

```
# Construimos un DataFrame para ver la propina según la hora del día
df.pivot_table(index='day', columns='time', values='tip', aggfunc='mean')
```

```
time    Lunch    Dinner
day
Thur    2.767705    3.000000
Fri     2.382857    2.940000
Sat     NaN        2.993103
Sun     NaN        3.255132
```

▼ Ejercicios

Calcula la **media** del total de la cuenta (total_bill).

Calcula la **mediana** de la propina (tip).

¿Cuántas horas del día podemos pagar ir a los restaurantes?

Calcula el **porcentaje** de registros donde el **sexo** del comensal es masculino (sex).

Calcula la desviación estándar del total de la cuenta agrupado por día de la semana (day). ¿Es mejor tener una desviación pequeña o grande?
¿Por qué motivo?

Calcula el coeficiente de correlación entre el total de la cuenta y la propina. ¿Hay relación entre las dos variables?

¿Quién deja una propina mayor, los fumadores o los no fumadores?

Anexo X. Google Colab: Un análisis completo

Se puede acceder a esta actividad en el siguiente enlace:

https://colab.research.google.com/drive/17IMV_77VXDy02ofPclAuJlQIbznfWkI?usp=sharing

7. Un análisis completo

▼ ¿Qué queremos responder?

En este documento vamos a hacer un análisis de las precipitaciones acumuladas mensuales de la ciudad de Barcelona desde 1786. Veremos como han variado anual y mensualmente las precipitaciones. Aplicaremos todo lo que hemos ido aprendiendo hasta ahora.

▼ 1. Importamos las librerías necesarias

A continuación tenemos las librerías que vamos a utilizar:

- pandas
- numpy
- matplotlib

Esto lo hacemos siempre al principio.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

▼ Recopilamos datos

▼ 2. Cargamos el archivo de datos

El documento que vamos a tratar se llama 'PRECIPITACIONES_BCN.csv'. Podemos encontrarlo en el grupo de Teams de clase en castellano o en catalán en este enlace:

<https://opendata-ajuntament.barcelona.cat/data/es/dataset/precipitacio-hist-bcn/resource/5da03f48-020e-4f46-9199-a919feac2034>

```
Recordar importar el archivo cada vez que lo queráis tratar, Google Colab no guarda los archivos e
```

Este documento recoge las precipitaciones de Barcelona distribuidas por mes desde el año 1786. Tiene 4 columnas:

- Año
- Mes: valor numérico del mes
- Desc_Mes: mes escrito
- Precipitaciones: precipitaciones acumuladas medidas en milímetros

```
df = pd.read_csv("/content/PRECIPITACIONES_BCN.csv", sep=';')
df.head() # Visualizamos los primeros datos
```

	Año	Mes	Desc_Mes	Precipitaciones
0	1786	1	Enero	32.8
1	1786	2	Febrero	28.4
2	1786	3	Marzo	84.4
3	1786	4	Abril	42.3
4	1786	5	Mayo	8.5

Estudiamos un poco el DataFrame para ver cuantos elementos deberemos tratar:

```
# ¿Con cuántas filas trabajaremos?
len(df)
```

```
2844
```

```
# ¿Qué forma tiene el DataFrame? ¿Cuántas filas y cuántas columnas?
df.shape
```

```
(2844, 4)
```

```
# ¿Qué resumen estadístico podemos saber?
df.describe()
```

```
# No nos aporta demasiada información
```

	Año	Precipitaciones
count	2844,000000	2844,000000
mean	1904,000000	49,220886
std	68,427400	45,000000

▼ Limpiamos los datos

50%	1904,000000	37,400000
------------	-------------	-----------

En este bloque limpiamos los datos y los estructuramos para según nos interese para el análisis que queremos realizar.

▼ 3. Datos inexistentes

Primero trataríamos los datos inexistentes, las casillas con NaN o None.

```
# Primero comprobamos en todas las casillas si hay algún elemento que sea nulo (isnull()),
# a continuación miramos si algún elemento de la columna ha sido nulo (any())
df.isnull().any()
```

```
Año           False
Mes           False
Desc_Mes      False
Precipitaciones False
dtype: bool
```

En este caso no tendremos que tratar ningún dato, lo tenemos completo.

▼ 4. Eliminamos y renombramos columnas

En este punto también podemos eliminar las columnas que no nos interesen y renombrarlas como nos interese.

Con el dataset de precipitaciones, nos interesan todas las columnas, pero si por ejemplo, no nos interesa la columna numérica de meses, la eliminaríamos y posteriormente modificaremos el nombre de las columnas:

```
# Eliminamos columnas

df2 = df.drop(['Mes'], axis=1)

# También podríamos usar la siguiente función:
```

```
# del df['Mes']
```

```
df2.head()
```

	Año	Desc_Mes	Precipitaciones
0	1786	Enero	32.8
1	1786	Febrero	28.4
2	1786	Marzo	84.4
3	1786	Abril	42.3
4	1786	Mayo	8.5

```
# Renombramos columnas
```

```
df2 = df2.rename(columns={"Desc_Mes": "Mes"})
```

```
df2.head()
```

	Año	Mes	Precipitaciones
0	1786	Enero	32.8
1	1786	Febrero	28.4
2	1786	Marzo	84.4
3	1786	Abril	42.3
4	1786	Mayo	8.5

▼ Analizamos los datos

El primer análisis que haremos es ver cada año cuántas precipitaciones ha habido:

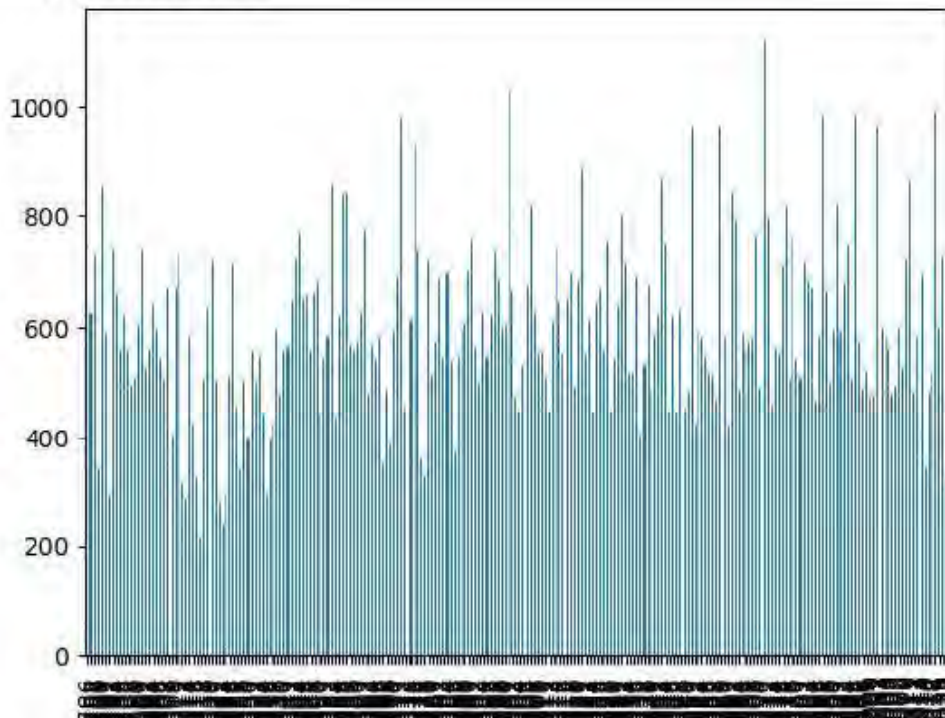
```
precip_ano_df = df.groupby(['Año'])['Precipitaciones'].sum()
```

```
precip_ano_df.head()
```

```
Año
1786    722.2
1787    625.1
1788    728.8
1789    342.5
1790    855.7
Name: Precipitaciones, dtype: float64
```

```
precip_ano_df.plot(kind='bar')
```

```
<Axes: xlabel='Año'>
```



¡Esta visualización no es buena! Hay demasiada información y no podemos verla bien. En este caso vamos a estudiar si existe alguna tendencia.

Una tendencia se refiere a la dirección y patrón general de cambio en los datos a lo largo del tiempo.

Para estudiar la tendencia agruparemos los años por décadas.

```
# Creamos una nueva columna para indicar la década
precip_aux_df = pd.DataFrame({'Precipitaciones': precip_ano_df, 'Decada': (precip_ano_df.index // 10).astype(int)}
precip_aux_df.head()
```

	Precipitaciones	Decada
Año		
1786	722.2	1780
1787	625.1	1780
1788	728.8	1780
1789	342.5	1780
1790	855.7	1790

```
# Agrupamos por década y contamos de cuántos años de esa década tendríamos información.
agg_functions = {'Precipitaciones': 'sum', 'Decada': 'count'}
precip_decada_df = precip_aux_df.groupby('Decada').agg(agg_functions)
```



```
precip_decada_df = precip_decada_df.rename(columns={"Decada": "Num_años_decada"})
precip_decada_df.head()
```

	Precipitaciones	Num_años_decada
Decada		
1780	2418,6	4
1790	5866.9	10
1800	5777.8	10
1810	4681.7	10
1820	4551.9	10

```
# Eliminamos aquellas filas que no tengan la década completa
precip_decada_df = precip_decada_df.loc[precip_decada_df['Num_años_decada']==10,:]
# Eliminamos la columna que ya no nos interesa
del precip_decada_df['Num_años_decada']
precip_decada_df.head()
```

	Precipitaciones
Decada	
1790	5866.9
1800	5777.8
1810	4681.7
1820	4551.9
1830	4614,8

```
precip_decada_df.plot(kind='bar')
```


Precipitaciones

	count	mean	std	min	25%	50%	75%	max
Decada								
1780	4.0	604.650000	181.081980	342.5	554.450	673.65	723.850	728.8
1790	10.0	586.690000	150.615371	291.8	518.750	573.70	646.125	855.7
1800	10.0	577.780000	93.554783	402.5	528.975	577.85	630.600	736.6
1810	10.0	468.170000	179.915629	215.6	319.525	463.85	618.500	729.0
1820	10.0	455.190000	169.594395	239.4	305.175	477.10	508.325	719.2
1830	10.0	461.480000	91.015991	294.6	403.825	456.10	534.125	597.0
1840	10.0	644.760000	71.110543	555.6	582.350	650.00	674.275	768.0
1850	10.0	628.730000	159.857145	435.9	548.025	576.25	782.375	858.6
1860	10.0	535.830000	121.103721	352.6	477.425	553.70	578.400	774.9
1870	10.0	639.620000	218.025844	329.2	487.825	650.95	729.400	980.0
1880	10.0	567.330000	91.907019	373.7	537.925	559.20	603.800	696.4
1890	10.0	632.880000	85.075062	497.6	571.875	623.75	695.475	757.8
1900	10.0	640.550000	174.908365	448.0	531.400	615.20	671.000	1030.6
1910	10.0	587.870000	94.479863	447.3	517.075	581.45	646.075	740.0
1920	10.0	622.490000	134.763987	443.8	551.950	625.25	675.875	886.3
1930	10.0	602.440000	119.640101	403.2	521.100	589.15	686.175	801.9
1940	10.0	589.080000	140.680282	443.6	457.925	601.90	627.275	867.8
1950	10.0	604.360000	196.489814	421.8	488.250	530.40	589.800	964.7
1960	10.0	617.250000	136.585702	417.1	564.900	580.40	719.125	839.0
1970	10.0	676.500000	206.715231	456.6	516.025	634.00	787.600	1122.7
1980	10.0	630.180000	151.245245	463.9	515.950	623.45	676.450	982.3
1990	10.0	650.000000	158.396401	485.1	534.050	596.70	728.425	982.4
2000	10.0	574.460000	146.702899	469.1	479.350	541.35	596.425	963.5
2010	10.0	627.040000	193.593252	345.8	489.750	589.90	713.550	988.0
2020	3.0	452.933333	234.528769	307.7	317.650	327.60	525.550	723.5

Anexo XI. Descripción de la actividad: “¿Qué hemos aprendido?”

ACTIVIDAD 8. ¿QUÉ HEMOS APRENDIDO?

En esta actividad, el alumnado utilizará los datos recopilados y visualizados en la actividad 7 para crear un póster y presentarlos ante la clase. Pueden usar herramientas como Canva o Word para crear el póster e informe. El objetivo es que el alumnado pueda tener un primer contacto con el producto final que deberán realizar. Se trata de una actividad rápida, no se espera un gran nivel de profundidad en los resultados.

Para el póster se pedirá que tenga los siguientes apartados:

- Título
- Autores
- Resumen o introducción
- Resultados
- Interpretación
- Conclusiones

En el siguiente enlace se pueden encontrar posters de ejemplo junto con consejos para realizar un buen póster científico: <https://neoscientia.com/como-hacer-un-poster-cientifico/>

Anexo XII. Rúbrica para la actividad: “¿Qué hemos aprendido?”

RÚBRICA ACTIVIDAD 8. ¿QUÉ HEMOS APRENDIDO?

		ASPECTOS	4	3	2	1
TER	Secciones del Póster	Todas las secciones necesarias están presentes y presentadas de forma clara y completa.	Todas las secciones necesarias están presentes, aunque algunas no están completamente desarrolladas.	Falta una sección necesaria.	Algunas secciones necesarias no se han agregado.	
	Organización de los contenidos	La información está muy bien organizada. El conjunto se complementa con imágenes y gráficos.	La información escrita está estructurada con párrafos complementados con imágenes.	Información estructurada, incluyendo algunos gráficos, pero párrafos mal redactados.	La información escrita no contiene gráficos y no está estructurada.	
	Resultados	El análisis de los datos se explica en forma de texto y gráficos.	Los datos del análisis se presentan con algunas explicaciones en forma de texto.	Los datos se presentan sin ningún análisis.	No se han proporcionado cifras finales.	
	Conclusión	Las conclusiones se apoyan en los resultados obtenidos y cumplen con los objetivos asumidos.	Las conclusiones presentadas corresponden al análisis realizado y se confirman con los resultados obtenidos.	Las conclusiones solo contienen el análisis realizado.	Las conclusiones incluyen algunos aspectos que no han sido analizados.	
	Estética	Soporte visual especialmente atractivo y de mucha calidad.	Soporte visual adecuado e interesante.	Soporte visual adecuado, pero poco interesante.	Soporte visual inadecuado.	
EXPOSICIÓN	Exposición	Capta la atención de la audiencia y mantiene el interés a lo largo de la exposición.	Interesa bastante en principio, pero se hace un poco monótono.	Le cuesta ganar o mantener el interés público.	Apenas utiliza recursos para mantener la atención del público.	
	Expresión oral	Habla claro durante toda la presentación. Su pronunciación es correcta. Su tono de voz es apropiado.	Habla claramente en la mayor parte de la presentación. Su pronunciación es aceptable, pero a veces realiza pausas innecesarias. Su tono de voz es apropiado.	Algunas veces habla claramente. Pronuncia correctamente, pero a menudo utiliza pausas innecesarias. Su tono de voz no es el apropiado.	Durante la mayor parte de la presentación no habla con claridad. Mala pronunciación, muchas pausas y uso de palabras de acompañamiento. Su tono de voz no despierta el interés del oyente.	
	Lenguaje no verbal	Tiene buena postura, y demuestra confianza durante la presentación. Establece contacto visual con todos los presentes.	Tiene una buena postura la mayor parte del tiempo y mantiene contacto visual con todos los presentes. A veces no demuestra confianza.	A veces tiene buena postura y pocas veces establece contacto visual con todos los presentes. No muestra confianza.	Tiene mala postura y no hace contacto visual con los presentes. No muestra confianza.	

Anexo XIII. Descripción de la actividad: “Analizando Datos”

ACTIVIDAD 9. ANALIZANDO DATOS

En esta actividad, les proporcionaremos al alumnado dos conjuntos de datos nuevos. En este caso, trabajaremos con el listado de equipamientos de Badalona dividido por barrio, junto a la población segmentada por edad en cada uno de los barrios.

Los datos se extraen del portal opendata de Badalona, que se encuentra en <http://opendata.badalona.cat/>. Se puede encontrar el conjunto de datos del equipamiento de Badalona en el siguiente enlace: <http://opendata.badalona.cat/od/lLista-dequipaments-de-la-ciutat-de-badalona/>. El conjunto de datos de la población segmentada por edad no se encuentra directamente en el portal de opendata, ya que ha sido tratado previamente. Se le ha añadido una columna con el código del distrito territorial por si algún grupo de alumnos quisiera relacionar los dos conjuntos de datos. Se puede encontrar en el siguiente enlace: <https://drive.google.com/file/d/1jNWQzsTjkXo9apd1ud8m64e4SVIiB8Ew/view?usp=sharing>

En lugar de guiarlos en un análisis específico, se les dará libertad para decidir qué información consideran relevante y qué aspectos quieren investigar. La toma de decisiones sobre qué variables son más importantes y cómo quieren presentar los resultados ha de ser completamente del alumnado.

Anexo XIV. Descripción de la actividad: “¿Qué historia nos cuentan los datos?”

ACTIVIDAD 10. ¿QUÉ HISTORIA NOS CUENTAN LOS DATOS?

Para concluir la situación de aprendizaje, el alumnado deberá presentar los resultados de su proyecto en forma de un informe y un póster.

El informe estará dirigido al Ayuntamiento de Badalona, el alumnado deberá redactarlo de manera formal y estructurada. El informe debe contener las conclusiones obtenidas a partir del análisis de los conjuntos de datos proporcionados, así como las recomendaciones que los estudiantes consideren importantes para mejorar las condiciones en diferentes puntos de Badalona. Es importante que los estudiantes hagan sus conclusiones y recomendaciones basadas en los datos analizados y en sus propios argumentos.

Por otro lado, el póster se presentará en un congreso organizado en la escuela. Esto significa que los estudiantes deberán diseñar un póster visualmente atractivo y claro, que resuma los aspectos más relevantes de su proyecto. El póster servirá como una forma de comunicar los resultados de la investigación a otros estudiantes y profesores. Es importante que el póster contenga los principales hallazgos, gráficos o visualizaciones que respalden las conclusiones del proyecto y alguna información adicional que pueda ser relevante.

Esta actividad se evaluará de la siguiente manera:

1. Evaluación del docente: El docente utilizará una rúbrica que abarcará tanto el informe como el póster.
2. Coevaluación: Cada grupo evaluará el póster de los demás grupos utilizando una rúbrica proporcionada por el profesor.
3. Coevaluación y autoevaluación del trabajo en grupo: Los estudiantes llevarán a cabo una coevaluación sobre el trabajo y la implicación con su grupo.

Anexo XV. Rúbrica docente para la actividad: “¿Qué historia nos cuentan los datos?”

RÚBRICA DOCENTE, ACTIVIDAD 10. ¿QUÉ HISTORIA NOS CUENTAN LOS DATOS?

		4	3	2	1
PÓSTER	Secciones	Todas las secciones necesarias están presentes y presentadas de forma clara y completa.	Todas las secciones necesarias están presentes, aunque algunas no están completamente desarrolladas.	Falta una sección necesaria.	Algunas secciones necesarias no se han agregado.
	Organización	La información está muy bien organizada. El conjunto se complementa con imágenes y gráficos.	La información escrita está estructurada con párrafos complementados con imágenes.	Información estructurada, incluyendo algunos gráficos, pero párrafos mal redactados.	La información escrita no contiene gráficos y no está estructurada.
	Resultados	El análisis de los datos se explica en forma de texto y gráficos.	Los datos del análisis se presentan con algunas explicaciones en forma de texto.	Los datos se presentan sin ningún análisis.	No se han proporcionado cifras finales.
	Conclusión	Las conclusiones se apoyan en los resultados obtenidos y cumplen con los objetivos asumidos.	Las conclusiones presentadas corresponden al análisis realizado y se confirman con los resultados obtenidos.	Las conclusiones solo contienen el análisis realizado.	Las conclusiones incluyen algunos aspectos que no han sido analizados.
	Estética	Soporte visual especialmente atractivo y de mucha calidad.	Soporte visual adecuado e interesante.	Soporte visual adecuado, pero poco interesante.	Soporte visual inadecuado.
INFORME	Estructura	Todas las secciones necesarias (título, introducción, cuerpo, conclusión) están presentes y presentadas de forma clara y completa.	Todas las secciones necesarias (título, introducción, cuerpo, conclusión) están presentes, aunque algunas no están completamente desarrolladas.	Falta una sección necesaria (título, introducción, cuerpo, conclusión).	Algunas secciones necesarias (título, introducción, cuerpo, conclusión) no se han agregado.
	Información	La información presentada es totalmente consistente con la investigación realizada.	La información presentada corresponde a la investigación realizada.	La información presentada es consistente con la investigación realizada. No está bien argumentada.	La información presentada no es relevante para la investigación realizada. No está bien argumentada.
	Propuestas de mejora	Se realizan dos o más propuestas para mejorar la localidad. Las propuestas se basan en investigaciones, están bien explicadas y son realistas.	Se realiza una propuesta para mejorar la localidad. La propuesta se basa en investigaciones, está bien explicada y es realista.	Se realiza una propuesta para mejorar la localidad. La propuesta no se basa en investigaciones, pero está bien explicada y es realista.	No se realiza ninguna propuesta para mejorar la localidad, o bien es irreal.

Anexo XVI. Rúbrica coevaluación para la actividad: “¿Qué historia nos cuentan los datos?”

RÚBRICA COEVALUACIÓN, ACTIVIDAD 10. ¿QUÉ HISTORIA NOS CUENTAN LOS DATOS?

Se puede descargar la rúbrica de coevaluación en el siguiente enlace:

https://www.canva.com/design/DAFjqjJ5sYQ/VbnPXmUy8vPwLeeEbtQLNg/edit?utm_content=DAFjqjJ5sYQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

		Evaluador: Grupo EVALUADO/DA:			
		1	2	3	4
¿QUÉ HISTORIA NOS CUENTAN LOS DATOS?	Secciones	Algunas secciones necesarias no se han agregado.	Falta una sección necesaria.	Todas las secciones necesarias están presentes, aunque algunas no están completamente desarrolladas.	Todas las secciones necesarias están presentes y presentadas de forma clara y completa.
	Organización	La información escrita no contiene gráficos y no está estructurada.	Información estructurada, incluyendo algunos gráficos, pero párrafos mal redactados.	La información escrita está estructurada con párrafos complementados con imágenes.	La información está muy bien organizada. El conjunto se complementa con imágenes y gráficos.
	Resultados	No se han proporcionado cifras finales.	Los datos se presentan sin ningún análisis.	Los datos del análisis se presentan con algunas explicaciones en forma de texto.	El análisis de los datos se explica en forma de texto y gráficos.
	Conclusión	Las conclusiones incluyen algunos aspectos que no han sido analizados.	Las conclusiones solo contienen el análisis realizado.	Las conclusiones presentadas corresponden al análisis realizado y se confirman con los resultados obtenidos.	Las conclusiones se apoyan en los resultados obtenidos y cumplen con los objetivos asumidos.
	Estética	Soporte visual inadecuado.	Soporte visual adecuado, pero poco interesante.	Soporte visual adecuado e interesante.	Soporte visual especialmente atractivo y de mucha calidad.

Anexo XVII. Rúbrica coevaluación y autoevaluación

COEVALUACIÓN Y AUTOEVALUACIÓN DEL TRABAJO EN GRUPO⁴

INFORMACIÓN DEL GRUPO EVALUADO

Grupo: _____

Evaluador: _____

Compañero/a EVALUADO/DA: _____

INDICADORES DE EVALUACIÓN

1.COOPERACIÓN: Ha puesto sus talentos al SERVICIO de la mejora del grupo, trabajando CONJUNTAMENTE e integrando las OPINIONES de los otros.

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

2.RESPECTO: Los conflictos han sido solucionados dentro del grupo; ha sabido argumentar su parecer, escuchar, comunicarse y llegar a acuerdos de manera asertiva.

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

3. ROL Y RESPONSABILIDAD: Se ha encargado de las tareas de las cuales ha estado responsable y las ha realizado o coordinado adecuadamente.

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

4. INICIATIVA Y PRO-ACTIVIDAD: Ha aportado ideas y compromisos, ha colaborado activamente en la INVESTIGACIÓN De INFORMACIÓN y la GENERACIÓN De IDEAS.

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

5. CALIDAD DE LAS TAREAS: Ha contribuido que las entregas cumplen con los criterios de presentación, corrección lingüística, formato y contenido.

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

⁴ Esta rúbrica es una adaptación de las rúbricas de coevaluación que se emplean en el centro Maristes Champagnat de Badalona

6. AUTONOMÍA Y PLANIFICACIÓN: Ha contribuido a hacer una muy buena PLANIFICACIÓN del trabajo y ha aprovechado las HORAS DE TRABAJO de forma óptima haciendo entregas puntualmente y a tiempo.

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

7. LOGRO DE OBJETIVOS: En general, se ha preocupado para lograr los OBJETIVOS individuales/grupales y el ÉXITO final del proyecto cooperativo.

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

VALORACIÓN CUALITATIVA

Si lo consideras oportuno, destaca y comenta de forma cualitativa los aspectos positivos (puntos fuertes) y/o las áreas de mejora por el que hace el trabajo en equipo.

11.PUNTOS FUERTES:

--

12.ÁREAS DE MEJORA:

--