Investigación

Instructor:

Luis Fernando Sanches

Aprendices :

Iyer Smith Torres Jaramillo

Juan Sebastián Quiceno Cano

Carloz Montoya Acevedo

16/02/2025

(1). Ambiente Personal De Aprendizaje:

Que es :

Es un conjunto de recursos que se utilizan para aprender y adquirir habilidades.

También conocida como (PLE).

Recursos para aprender:

* Bibliotecas digitales
* Blogs
* Canales de YouTube
* Programas de estudio
* Cursos en línea
* Hoy día (las inteligencias artificiales )

Características de un PLE :

* Se adapta a las necesidades del estudiante
* Permite prender a largo plazo
* Ayuda a desarrollar competencias digitales
* Permite establecer objetivos de aprendizaje

(2). Métodos De Aprendizaje :

Que es:

Son los métodos que usan los estudiantes para entender algo que le s llame la atención, por ejemplo:

* Auditiva
* Visual
* Lectura / Escritura
* Cinestésica

De la que voy a hacer ejemplo es la cinestésica, entonces que es cinestésica, son aquellos estudiantes que necesitan ponerse a la obra y les resulta más fácil aprender haciendo algo.

Link:

<https://www.kumon.com.bo/blog/estilos-de-aprendizaje/#:~:text=En%20%C3%A9l%2C%20cada%20una%20de,%2C%20asimilador%2C%20convergente%20y%20acomodador>.

Para la programación la forma más fácil de aprender es:

* Creación de proyectos
* Practica
* Mejorar la lógica (Algoritmos y Recursividad )

(3). Tipos De Aprendizaje:

* Aprendizaje Cooperativo:

Se trabaja en equipo para resolver problemas y construir conocimientos.

* Aprendizaje Autónomo:

La persona organiza su propio proceso de aprendizaje sin depender de un guia constante.

* Aprendizaje por observación:

Se aprende viendo a otros y modelando su comportamiento.

* Aprendizaje basado en Problemas:

Se resuelve casos o situaciones reales para adquirir conocimientos y habilidades.

Link :

https://www.ispring.es/blog/estrategias-de-aprendizaje

(4).  Que es una base de datos :

Que son:

Es el producto de nuestra necesidad de guardar información de gran escala organizado de manera sistemático para:

* Analizarla y transmitirla

Son las encargadas de guardar (datos) atreves del tiempo y el deterioro, ejemplo de bases de datos:

* Bibliotecas publicas
* Historiales médicos o registro de transacciones

Sistema de Gestiones De Bases De datos ejemplos de unos son:

* PostgreSQL
* SQLite
* Apache Derby
* My SQL
* FireBird
* Microsoft SQL Server

Sirven para facilitar a el usuario la recuperación de los datos  ejemplo:

* Insert
* Select \* from
* Delete
* Alter

Link :

<https://www.youtube.com/watch?v=6S8A-1jBD5Y>



(5).  Tipos De Bases De Datos:

Relacionales:

Usan filas y columnas para almacenar y organizar los datos usan el método relacional,

Donde los datos de organizan en tablas y se establecen relaciones entre ellas mediante claves Primarias y Claves Externas, ejemplo

Una tienda

Tabla que contiene productos y  la otra los proveedores la cual se establece una relación entre ellas.

No relacionales (noSQL):

Utilizan datos flexibles y no requieren de esquemas relacionales la cual se pueden usar otros tipos de datos como ejemplo:

* Grafos
* Clave de valor
* Documentos

Según las necesidades de la aplicación ejemplo :

Aplicación de recomendaciones en línea, se modifica lo que ya este predefinido la cual se guardan en documentos  jsons la cual se pueden acceder y actualizar de manera más rápida y flexible.

Sus estructuras

Relacionales : tiene una estructura rígida la cual deben estar predefinida de antemano la cual cada tabla tiene su esquema predefinido y se organizan en filas.

No SQL

Tienen una estructura mas flexible y no necesitan una estructura predefinida

Escalabilidad

Relacionales:

Son escalables verticalmente,

 lo cual se pueden mejorar usando mas recursos en servidores existentes

No SQL:

Escalables horizontalmente

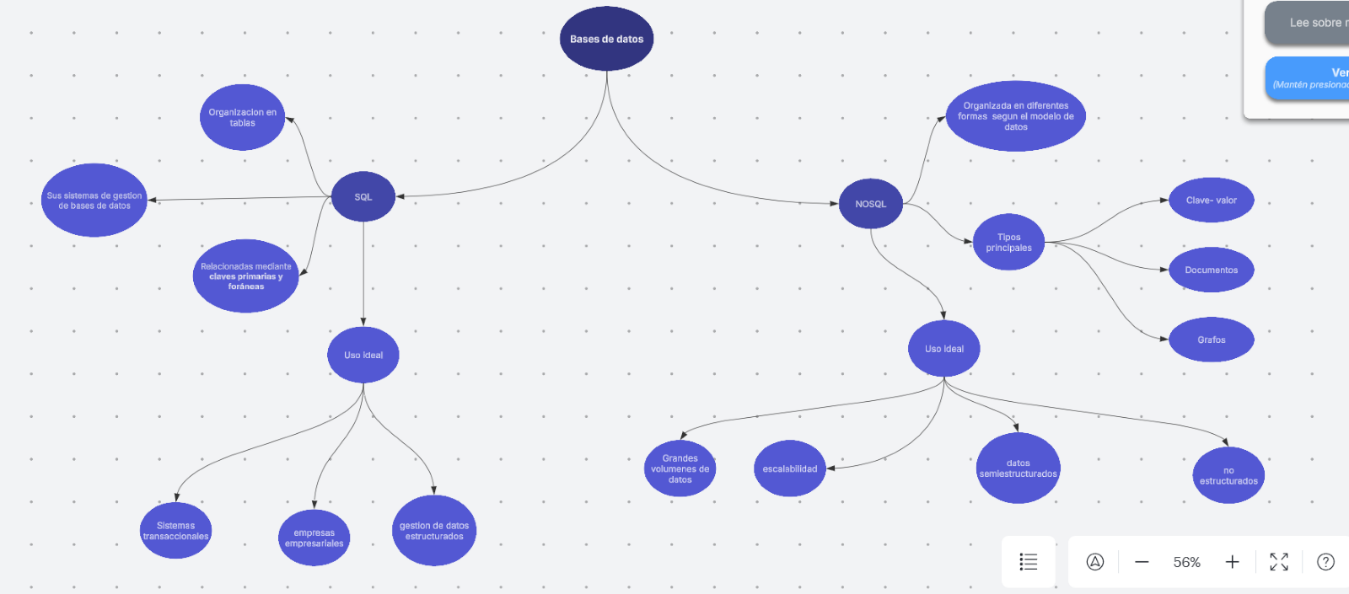
Pueden manejar grandes volúmenes de datos distribuyendo la carga en múltiples servidores

Link:

<https://www.youtube.com/watch?v=xkMDTulV8r4>



(6). Mapa Conceptual



Link:

[https://lucid.app/lucidspark/0cc76399-2b1d-471e-813a-2c13cda5ddcb/edit?beaconFlowId=1680867528A77295&invitationId=inv\_33654a8f-5d2b-4190-84a2-9cb9de6a1fad&page=gHjjwlkrc\_Jt#](https://lucid.app/lucidspark/0cc76399-2b1d-471e-813a-2c13cda5ddcb/edit?beaconFlowId=1680867528A77295&invitationId=inv_33654a8f-5d2b-4190-84a2-9cb9de6a1fad&page=gHjjwlkrc_Jt)

(7). marcas y fabricantes de gestores de bases de datos

* MY SQL :

El sistema gestor de bases de datos relacional es un SGBD (multihilo y multiusuario) utilizado en la gran parte de las paginas web actuales. Además, es el mas usado en aplicaciones creadas con software libre.

Ventajas:

* Facilidad de uso y gran rendimiento
* Facilidad para instalar y configurar
* Soporte multiplataforma
* Soporte SSL

Creador:

Desarrollada por la empresa sueca MYSQLAB, fundada por David Allan Larsson, y Michael Widenius. La primera versión de MySQL se lanzó en 1995

* MariaDB

Es una derivación de (MySQL) que cuenta con la mayoría de las características de este e incluye varias extensiones. Nace a partir  de la adquisicion de MYSQL por parte de ORACLE para seguir la filosofía (OPEN SOUCE)  y tiene la ventaja de que es totalmente compatible con MYSQL

Características:

* Aumento de motores de bases de almacenamiento
* Gran escalabilidad
* Seguridad y rapidez en transacciones
* Extensiones y nuevas características relacionadas con su aplicacion para base de datos

Creador: Michael "Monty" Widenius creó MariaDB, un sistema de gestión de bases de datos relacionales (SGBD). Widenius también fundó MySQL AB, la empresa que desarrolló MySQL, y es miembro fundador de la Fundación MariaDB.

SQLite

 Es más una biblioteca utilizada en magnitud de aplicaciones actuales ya que es OPEN SOURCE y las consultas son muy eficientes.

Características o mejoras

* El tamaño de tratarse de una biblioteca, es mucho menor que cualquier SGBD
* Reúne los cuatro criterios ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)
* Gran portabilidad y rendimiento

Creador: . Richard Hipp diseñó SQLite en la primavera de 2000 mientras trabajaba para General Dynamics por contrato con la Marina de los Estados Unidos.

Link: https://es.wikipedia.org/wiki/SQLite#:~:text=SQLite%20es%20un%20proyecto%20de,Richard%20Hipp.&text=A%20diferencia%20de%20los%20sistema,el%20programa%20principal%20se%20comunica.

PostgreSQL

Esta orientada a objetos y es libre, publicado bajo la licencia BSD

Características:

* Control de concurrencias multiverso (MVCC)
* Flexibilidad en cuanto a lenguajes de programación
* Dispone de una herramienta muy fácil e intuitiva para administrar las bases de datos.
* Robustez, eficacia y estabilidad

Creador:

El profesor Michael Stonebraker de la Universidad de California, Berkeley, creó PostgreSQL en 1986. El proyecto original se llamó POSTGRES.

Microsoft SQL Server

Es un sistema de bases de datos relacionales basado en el lenguaje TRANSACT-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Características:

* Soporte exclusivo por parte de Microsoft
* Escalabilidad, estabilidad y seguridad
* Posibilidad de cancelar consultas
* Potente entorno grafico de administracion que permite utilizar comandos DDL Y DML

 Creador: Microsoft creó SQL Server en 1988. Este sistema de gestión de bases de datos relacionales (RDBMS) fue desarrollado en colaboración con Sybase y Ashton-Tate.

MongoDB

 Es un SBGD NoSQL orientada a ficheros que almacena la información de estructuras BSON con un esquema dinámico que permite su facilidad de integración.

Características:

-Indexación y replicación

- Balaceo de carga

- Almacenamiento en ficheros

- Consultas ad hoc

- Escalabilidad Horizontal

- Open Source

Creador:  Dwight Merriman, Eliot Horowitz y Kevin Ryan fundaron MongoDB en 2007. Estos tres fueron parte del equipo que desarrolló DoubleClick, una empresa de publicidad en internet que ahora es propiedad de Google.

LINK de donde sacamos todo el texto reescrito, los creadores si fueron copias : https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/

(8). ¿Qué es Python, historia y documentación?

Su historia.

El origen de Python, fue desarrollado inicialmente a fines de la década de 1980 por el programador holandés Guido van Rossum, y su primera versión fue lanzada en 1991 en Centrum Wiskunde & Informatica - CWI (Instituto Nacional de Investigación en Matemáticas y Ciencias de la Computación), en los Países Bajos.

El lenguaje ha experimentado grandes avances en el desarrollo de código abierto, utilizando las PEPs ("Propuestas de Mejora de Python") como principal herramienta para sugerencias de mejora y discusiones dentro de la comunidad.

Estas PEPs se utilizan para describir cambios en el lenguaje o en sus normas, y son evaluadas por el público, aceptadas o rechazadas después de muchas discusiones. Cualquier persona puede escribir y enviar una PEP para su evaluación.

¿Qué Es?

Python es un lenguaje de programación de propósito general, lo que significa que se puede utilizar para crear una amplia variedad de aplicaciones y no está especializado en un problema específico.

Esta versatilidad, junto con su facilidad de uso para principiantes, lo ha convertido en uno de los lenguajes más utilizados en la actualidad.

Algunos de los puntos destacados de Python son:

Es un lenguaje interpretado, lo que significa que no necesita pasar por el proceso de compilación. En el caso de Python, el proceso de interpretación se realiza dentro de máquinas virtuales, donde el código pasa por una capa intermedia que traduce las instrucciones del programa a código binario. Esto acelera considerablemente la velocidad de desarrollo.

Su sintaxis es simple, fácil de aprender y muy cercana al lenguaje que utilizamos en nuestra comunicación cotidiana. Por esta razón, se puede decir que es un lenguaje de alto nivel.

Es multiparadigma, lo que significa que nos permite programar en varios paradigmas, como:

* Procedural: se basa en instrucciones que se pasan al ordenador en la secuencia en que deben ejecutarse.
* Funcional: consiste en construir programas aplicando y componiendo funciones.
* Orientado a objetos: brinda una perspectiva del mundo real a la programación, lo que facilita la comprensión de los programas debido a esta relación.

El propio programa "reconoce" el tipo de dato que se está utilizando, por lo que no es necesario declararlo previamente. Por eso decimos que Python tiene una semántica dinámica.

¿Por qué Se valora Python?

Basándonos en la investigación "Python Developers Survey 2021 Results" realizada por JetBrains, Python es uno de los lenguajes que crece más rápidamente en relevancia y es valorado por las empresas que lo utilizan debido a sus características, como agilidad, practicidad y versatilidad, especialmente para proyectos que requieren mayor complejidad.

En las áreas en las que se utiliza con mayor frecuencia, destacamos:

1. Análisis de datos: Python es ampliamente utilizado en el campo del análisis de datos debido a sus bibliotecas especializadas como NumPy, Pandas y Matplotlib, que ofrecen herramientas poderosas para el procesamiento, visualización y análisis de datos.
2. Aprendizaje automático (machine learning): Python se ha convertido en uno de los lenguajes más populares para el desarrollo de algoritmos de aprendizaje automático. Las bibliotecas como TensorFlow y scikit-learn proporcionan un conjunto completo de herramientas para construir y entrenar modelos de aprendizaje automático.
3. Desarrollo web: Python es ampliamente utilizado en el desarrollo web gracias a frameworks populares como Django y Flask. Estos frameworks ofrecen una estructura sólida y eficiente para crear aplicaciones web, facilitando tareas como el enrutamiento, la gestión de bases de datos y la generación de contenido dinámico.
4. DevOps: Python se utiliza en la automatización de tareas y la gestión de infraestructuras en el campo de DevOps. Su facilidad de uso, combinada con bibliotecas como Fabric y Ansible, permite a los desarrolladores automatizar procesos y administrar sistemas de manera eficiente.

**NumPy**

✨ **¿Qué es?**  
Es una librería para cálculos numéricos eficientes en Python. Permite trabajar con arreglos multidimensionales (arrays) y operaciones matemáticas avanzadas de manera rápida y optimizada.

✨ **¿Qué hace?**

* Manejo de **arrays** y **matrices** multidimensionales.
* Operaciones matemáticas rápidas con funciones optimizadas en C.
* Álgebra lineal, transformadas de Fourier y generación de números aleatorios.
* Ideal para cálculos científicos y de ingeniería.

**Pandas**

🔹 ¿Qué es?  
Es una librería para análisis y manipulación de datos en Python. Usa estructuras como DataFrames (tablas) y Series (columnas individuales).

🔹 ¿Qué hace?

* Carga y manipula datos desde CSV, Excel, SQL, JSON, etc.
* Limpieza, transformación y filtrado de datos.
* Agrupaciones y operaciones estadísticas.
* Manejo de datos faltantes y formatos de fecha.

**Matlolib**

👌 **¿Qué es?**  
Es una librería para crear **gráficos y visualizaciones** en Python. Es altamente personalizable y permite desde gráficos de líneas hasta histogramas y mapas de calor.

👌 **¿Qué hace?**

* Generación de gráficos de líneas, barras, dispersión, histogramas, etc.
* Personalización de colores, etiquetas y leyendas.
* Exportación de gráficos en PNG, PDF, SVG y más.
* Compatible con NumPy y Pandas.

WebGrafia: [https://www.aluracursos.com](https://www.aluracursos.com/blog/que-es-python-historia-guia-para-iniciar)

(9). Que es Replit y Github.

REPLIT

¿Qué es replit?

En su esencia, Replit es una plataforma basada en la nube que actúa como un entorno de desarrollo integrado (IDE). A diferencia de muchos IDEs tradicionales que requieren instalación en tu máquina, Replit opera directamente desde tu navegador web. Ya sea que uses Chrome, Firefox o cualquier otro navegador, puedes acceder a tu código, proyectos y archivos con facilidad.

**Lenguajes y programación**

Una de las mayores fortalezas de Replit es su amplio soporte para diferentes lenguajes de programación. Desde [Javascript](https://www.dongee.com/tutoriales/que-es-javascript-y-para-que-sirve/) hasta [Python](https://www.dongee.com/tutoriales/para-que-sirve-python/) y HTML, los desarrolladores tienen la libertad de trabajar en múltiples lenguajes sin la necesidad de cambiar de herramienta o entorno.

**¿Por qué usar Replit?**

Replit ha emergido rápidamente como una de las plataformas preferidas para muchos desarrolladores y educadores en todo el mundo. Su enfoque basado en la nube y las capacidades intuitivas lo hacen destacar entre las muchas opciones disponibles. A continuación, desglosamos algunas de las razones principales por las cuales deberías considerar usar Replit en tus proyectos:

* **Acceso desde cualquier lugar**: La belleza de Replit radica en su capacidad basada en la nube. No importa si estás en tu casa, en una cafetería o de viaje; siempre que tengas una conexión a **internet**, tu trabajo te espera justo donde lo dejaste.
* **Colaboración en tiempo real**: En un mundo donde el trabajo colaborativo se ha vuelto esencial, Replit brilla con su función de colaboración en tiempo real. Esto permite a varios **usuarios** trabajar simultáneamente en un mismo proyecto, facilitando la comunicación y el intercambio de ideas.
* **No se necesita instalación**: Dígale adiós a los largos procesos de instalación y configuración. Con Replit, simplemente abre tu **navegador**, inicia sesión y estás listo para empezar a codificar.

**Herramientas y aplicaciones integradas**

Más allá de ser un simple **editor**, Replit está equipado con una variedad de **herramientas** que facilitan el desarrollo:

* **Debugger**: Identifica y soluciona problemas en tu **código** con facilidad.
* **Consola integrada**: Ejecuta comandos, prueba fragmentos de código y más, todo sin salir de la plataforma.
* **Control de versiones**: Replit se integra con GitHub, permitiéndote gestionar tus **proyectos** y colaborar con otros **desarrolladores** con facilidad.

**Ventajas de usar Replit**

1. **Acceso Universal**: Como plataforma basada en la nube, puedes acceder a tus proyectos desde cualquier lugar con conexión a internet, sin necesidad de tener tu computadora personal.
2. **Sin Instalaciones**: Al ser una plataforma online, evitas la necesidad de instalar software y configuraciones en tu máquina, lo que ahorra tiempo y reduce posibles complicaciones.
3. **Múltiples Lenguajes de Programación**: Replit ofrece soporte para una variedad de lenguajes de programación, permitiéndote trabajar en distintos proyectos sin cambiar de plataforma.
4. **Colaboración en Tiempo Real**: La capacidad de colaborar en tiempo real con otros desarrolladores facilita la cooperación y el trabajo en equipo.
5. **Recursos Educativos**: Replit brinda herramientas para educadores, convirtiéndolo en una excelente opción para la enseñanza y el aprendizaje de programación.
6. **Integración con GitHub**: Esto facilita la gestión de versiones y la colaboración en proyectos más grandes.
7. **Comunidad Activa**: Con una base de usuarios en crecimiento, siempre hay oportunidades para aprender, compartir y recibir feedback.

**Desventajas de usar Replit**

1. **Dependencia del Internet**: Al ser una plataforma en línea, necesitas una conexión a internet para trabajar. Esto puede ser problemático si te encuentras en un lugar sin acceso o con una conexión inestable.
2. **Limitaciones de Rendimiento**: Aunque Replit es poderoso, no siempre puede igualar el rendimiento de un IDE local, especialmente con proyectos más grandes o complejos.
3. **Menos Personalización**: Aunque Replit es altamente funcional, puede no ofrecer el mismo nivel de personalización que otros IDEs tradicionales.
4. **Preocupaciones de Privacidad**: Al ser una plataforma en la nube, siempre existe el riesgo potencial de brechas de seguridad o problemas de privacidad, aunque Replit se esfuerza por mantener la seguridad de los datos de los usuarios.
5. **Puede no ser adecuado para proyectos grandes**: Para proyectos de gran envergadura con necesidades específicas, Replit puede no ser la opción más adecuada en comparación con soluciones locales más robustas.

**Historia de Replit2**:

Replit fue creado por los programadores Amjad Masad, Faris Masad y la diseñadora Haya Odeh en 2016.  Una vez que figuraba como cofundador junto con Masad, Max Shawabkeh dejó la empresa desde el principio.

Antes de crear Replit, **Amjad Masad** trabajó en roles de ingeniería en [Yahoo](https://es.wikipedia.org/wiki/Yahoo!" \o "Yahoo!) y [Facebook](https://es.wikipedia.org/wiki/Facebook) , donde creó herramientas de desarrollo. También ayudó a fundar [Codecademy](https://es.wikipedia.org/wiki/Codecademy" \o "Codecademy) . A Masad se le ocurrió la idea de Replit más de una década antes de su creación.

En 2009, Masad intentó escribir implementaciones de otros lenguajes de programación en [JavaScript](https://es.wikipedia.org/wiki/JavaScript), pero no resultó realizable. Vio grandes avances en las tecnologías web y de navegador y se inspiró en las capacidades web de [Google Docs](https://es.wikipedia.org/wiki/Documentos_de_Google) . Pensó en la idea de poder escribir código en un navegador y hacerlo fácil de compartir. Pasó dos años creando un producto de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) con Haya Odeh llamado "pyRepl".  Este producto le permitió compilar lenguajes en [JavaScript](https://es.wikipedia.org/wiki/JavaScript). Este producto impulsó los tutoriales de [Udacity](https://es.wikipedia.org/wiki/Udacity" \o "Udacity) y [Codecademy](https://es.wikipedia.org/wiki/Codecademy" \o "Codecademy). Después de convertirse en uno de los primeros empleados de [Codecademy](https://es.wikipedia.org/wiki/Codecademy" \o "Codecademy), este proyecto se pospuso hasta años más tarde, cuando él y Odeh decidieron revivir el proyecto de un entorno de [programación](https://es.wikipedia.org/wiki/Programaci%C3%B3n) en un [navegador](https://es.wikipedia.org/wiki/Navegador_web).

GITHUB3

¿Qué es Github?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

* **Presentar o compartir** el trabajo.
* **Seguir y administrar** los cambios en el código a lo largo del tiempo.
* Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
* **Colaborar** en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto **Git**, en el que se basa GitHub.

**Acerca de Git.**

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas y tú estáis haciendo cambios en los mismos archivos al mismo tiempo.

Normalmente, para hacerlo en un flujo de trabajo basado en Git, harías lo siguiente:

* **Crear una rama** a partir de la copia principal de archivos en los que tú (y tus colaboradores) estáis trabajando.
* **Realizar modificaciones** en los archivos de forma independiente y segura en tu propia rama personal.
* Dejar que Git **fusione mediante combinación** y de forma inteligente los cambios específicos en la copia principal de archivos, de modo que los cambios no afecten a las actualizaciones de otras personas.
* Dejar que Git **realice un seguimiento** de tus cambios y los de otras personas, por lo que todos siguen trabajando en la versión más actualizada del proyecto.

Al cargar archivos en GitHub, los almacenarás en un "repositorio de Git". Esto significa que al realizar cambios (o "confirmaciones") en los archivos de GitHub, Git se iniciará automáticamente para realizar el seguimiento de los cambios y administrarlos.

Hay muchas acciones relacionadas con Git que puedes completar en GitHub directamente en el navegador, como crear un repositorio de Git, crear ramas y cargar y editar archivos.

Pero la mayoría de los usuarios trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando empieces a colaborar con otros y todos necesitéis trabajar en el mismo repositorio al mismo tiempo, haréis lo siguiente continuamente:

* **Extraer** todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub.
* **Insertar** tus propios cambios en el mismo repositorio remoto en GitHub.

Git determina cómo combinar inteligentemente este flujo de cambios y GitHub te ayuda a administrarlo mediante características como las "solicitudes de cambios".

**Un poco de historia.**

[GitHub](https://github.com/) existe desde 2008, cuando el cofundador Chris Wanstrath creó el primer [code commit](https://en.wikipedia.org/wiki/Commit_(version_control)" \t "_blank) para lo que se convertiría en la principal plataforma del mundo para el desarrollo de software. Hoy en día, organizaciones de todos los tamaños y de todos los sectores, incluso Zendesk, aprovechan los repositorios, las comunidades, los flujos de trabajo y otros beneficios centrales del negocio de GitHub.

Webgrafias: [https://www.dongee.com](https://www.dongee.com/tutoriales/que-es-replit-una-guia-completa-para-principiantes/) \ 2 <https://es.wikipedia.org/wiki> \ 3[https://docs.github.com](https://docs.github.com/es/get-started/start-your-journey/about-github-and-git) \ 4<https://www.zendesk.com.mx/customer/github/>

(10). ¿Qué es ciencia de datos?

La ciencia de datos es el estudio de datos con el fin de extraer información significativa para empresas. Es un enfoque multidisciplinario que combina principios y prácticas del campo de las matemáticas, la estadística, la inteligencia artificial y la ingeniería de computación para analizar grandes cantidades de datos. Este análisis permite que los científicos de datos planteen y respondan a preguntas como “qué pasó”, “por qué pasó”, “qué pasará” y “qué se puede hacer con los resultados”.

¿Por qué es importante la ciencia de datos?

La ciencia de datos es importante porque combina herramientas, métodos y tecnología para generar significado a partir de los datos. Las organizaciones modernas están inundadas de datos; hay una proliferación de dispositivos que pueden recopilar y almacenar información de manera automática. Los sistemas en línea y los portales de pago capturan más datos en los campos del comercio electrónico, la medicina, las finanzas y cualquier otro aspecto de la vida humana. Disponemos de grandes cantidades de datos de texto, audio, video e imágenes.

Historia de la ciencia de datos

Aunque el término ciencia de datos no es nuevo, sus significados y connotaciones cambiaron con el tiempo. La palabra apareció por primera vez en los años 60 como nombre alternativo de la estadística. A finales de los 90, los profesionales de la computación formalizaron el término. Una propuesta de definición la consideraba un campo independiente con tres aspectos: diseño, recopilación y análisis de datos. Todavía tuvo que pasar otra década para que el término se utilizara fuera del ámbito académico.

Futuro de la ciencia de datos

La [inteligencia artificial](https://aws.amazon.com/what-is/artificial-intelligence/) y las innovaciones del [machine learning](https://aws.amazon.com/what-is/machine-learning/) han hecho que el procesamiento de datos sea más rápido y eficiente. La demanda del sector ha creado un ecosistema de cursos, grados académicos y puestos de trabajo en el campo de la ciencia de datos. Debido al conjunto de competencias multidisciplinarias y a la experiencia necesaria, la ciencia de datos promete un fuerte crecimiento en las próximas décadas.

¿Para qué se utiliza la ciencia de datos?

La ciencia de datos se utiliza para estudiar los datos de cuatro maneras principales:

**1. Análisis descriptivo**

El análisis descriptivo examina los datos para obtener información sobre lo que ha ocurrido u ocurre en el entorno de datos. Se caracteriza por las visualizaciones de datos, como los gráficos circulares, de barras o líneas, las tablas o las narraciones generadas. Por ejemplo, un servicio de reserva de vuelos registra datos como el número de billetes reservados cada día. El análisis descriptivo revelará los picos y las caídas de las reservas, así como los meses de alto rendimiento del servicio.

**2. Análisis de diagnóstico**

El análisis de diagnóstico es un examen profundo o detallado de datos para entender por qué ha ocurrido algo. Se caracteriza por técnicas como el análisis detallado, el descubrimiento y la minería de datos o las correlaciones. Se pueden llevar a cabo varias operaciones y transformaciones de datos en un determinado conjunto con el fin de descubrir patrones únicos en cada una de estas técnicas. Por ejemplo, el servicio de vuelos podría hacer el análisis detallado de un mes con un rendimiento particularmente alto para entender mejor el pico de reservas. Esto puede revelar que muchos clientes visitan una determinada ciudad para asistir a un evento deportivo mensual.

**3. Análisis predictivo**

El análisis predictivo utiliza los datos históricos para hacer previsiones precisas sobre los patrones de datos que pueden producirse en el futuro. Se caracteriza por técnicas como el machine learning, la previsión, la coincidencia de patrones y el modelado predictivo. En cada una de estas técnicas, se entrena a las computadoras para aplicar ingeniería inversa a las conexiones de causalidad en los datos. Por ejemplo, el equipo de servicios de vuelo podría utilizar la ciencia de datos para predecir los patrones de reserva de vuelos del año siguiente al inicio de cada año. El programa o algoritmo de la computadora pueden examinar datos anteriores y predecir picos de reservas de determinados destinos en mayo. Al anticiparse a las futuras necesidades de viaje de los clientes, la empresa podría empezar desde febrero a hacer publicidad específica para esas ciudades.

**4. Análisis prescriptivo**

El análisis prescriptivo lleva los datos predictivos al siguiente nivel. No solo predice lo que es probable que ocurra, sino que sugiere una respuesta óptima para ese resultado. Puede analizar las posibles implicaciones de las diferentes alternativas y recomendar el mejor curso de acción. Utiliza el análisis de gráficos, la simulación, el procesamiento de eventos complejos, las redes neuronales y los motores de recomendación del machine learning.

De vuelta al ejemplo de la reserva de vuelos, el análisis prescriptivo podría examinar las campañas de marketing históricas para maximizar la ventaja del próximo pico de reservas. Un científico de datos podría proyectar los resultados de las reservas de diferentes niveles de gasto en varios canales de marketing. Estas previsiones de datos dan a la empresa de reserva de vuelos una mayor confianza en sus decisiones de marketing.

¿Cuáles son los beneficios de la ciencia de datos para las empresas?

La ciencia de datos revoluciona el funcionamiento de las empresas. Muchas empresas, independientemente de su tamaño, necesitan una sólida estrategia de ciencia de datos para impulsar el crecimiento y mantener una ventaja competitiva. Algunos beneficios clave son:

**Descubrir patrones desconocidos de transformación**

La ciencia de datos permite a las empresas descubrir nuevos patrones y relaciones con el potencial de transformar la organización. Puede revelar cambios de bajo coste en la administración de recursos para obtener el máximo impacto en los márgenes de beneficio. Por ejemplo, una empresa de comercio electrónico utiliza la ciencia de datos para descubrir que se generan demasiadas consultas de clientes fuera del horario comercial. Las investigaciones revelan que es más probable que los clientes compren si reciben una respuesta rápida en lugar de una respuesta al día siguiente. Al implementar un servicio de atención al cliente las 24 horas del día, los 7 días de la semana, la empresa aumenta sus ingresos en un 30 %.

**Innovar con nuevos productos y soluciones**

La ciencia de datos puede revelar lagunas y problemas que de otro modo pasarían desapercibidos. Mejor información sobre las decisiones de compra, los comentarios de los clientes y los procesos empresariales puede impulsar la innovación en las operaciones internas y las soluciones externas. Por ejemplo, una solución de pago en línea utiliza la ciencia de datos para cotejar y analizar los comentarios que hacen los clientes sobre la empresa en redes sociales. Los análisis revelan que los clientes olvidan las contraseñas durante los periodos de pico de compra y que no están satisfechos con el actual sistema de recuperación de contraseñas. La empresa puede innovar para obtener una mejor solución y ver un aumento significativo en la satisfacción del cliente.

**Optimización en tiempo real**

Para las empresas, en especial las grandes, es un gran reto responder en tiempo real a las condiciones cambiantes. Esto puede causar importantes pérdidas o interrupciones en la actividad empresarial. La ciencia de datos puede ayudar a las empresas a predecir los cambios y reaccionar de forma óptima ante las distintas circunstancias. Por ejemplo, una compañía de envíos que use camiones utiliza la ciencia de datos para reducir el tiempo de inactividad si los camiones se rompen. Identifican las rutas y los patrones de turnos que propician averías más rápidas y ajustan los horarios de los camiones. Además, crean un inventario de piezas de repuesto comunes que se necesitan sustituir con frecuencia para que los camiones se puedan reparar con mayor rapidez.

¿Qué es el proceso de la ciencia de datos?

Un problema empresarial suele iniciar el proceso de la ciencia de datos. Un científico de datos trabajará con las partes interesadas del negocio para entender las necesidades del mismo. Una vez definido el problema, el científico de datos puede resolverlo con el proceso que consiste en obtener, depurar, explorar y modelar datos e interpretar los resultados (OSEMN):

**Obtener datos**

Los datos pueden ser preexistentes, recién adquiridos o un repositorio descargable de Internet. Los científicos de datos pueden extraerlos de las bases de datos internas o externas, del software CRM de la empresa, de los registros del servidor web, de las redes sociales o adquirirlos de terceros de confianza.

**Depurar datos**

La depuración o limpieza de datos consiste en el proceso de normalizarlos según un formato predeterminado. Incluye la gestión de los datos que faltan, la corrección de errores en estos y la eliminación de datos atípicos. Algunos ejemplos de la depuración de datos son:

* Cambiar todos los valores de fecha a un formato estándar común.
* Corregir las faltas de ortografía o los espacios adicionales.
* Corregir inexactitudes matemáticas o eliminar comas de números grandes.

**Explorar datos**

La exploración de datos es un análisis preliminar de estos que se utiliza para planificar otras estrategias para su modelado. Los científicos de datos obtienen una comprensión inicial de los datos mediante estadísticas descriptivas y herramientas de visualización de los mismos. A continuación, exploran los datos para identificar patrones interesantes que se puedan estudiar o utilizar.

**Modelar datos**

El software y los algoritmos de machine learning se utilizan para obtener información más profunda, predecir resultados y prescribir el mejor curso de acción. Las técnicas de machine learning, como la asociación, clasificación y agrupación, se aplican al conjunto de datos de entrenamiento. El modelo podría probarse con datos de prueba predeterminados para evaluar la precisión de los resultados. El modelo de datos se puede ajustar muchas veces para mejorar los resultados.

**Interpretar los resultados**

Los científicos de datos trabajan junto a los analistas y las empresas para convertir la información de datos en acción. Hacen diagramas, gráficos y tablas para representar tendencias y predicciones. La síntesis de datos ayuda a las partes interesadas a comprender y aplicar con eficacia los resultados.

**¿Las ETL y la Ciencia de Datos?**

En la ciencia de datos, ETL son las siglas de "extracción, transformación y carga" (Extract, Transform and Load). Es un proceso que se utiliza para combinar datos de diferentes fuentes en un repositorio central.

**¿Para qué se usa ETL?**

* Almacenar datos antiguos o agregar nuevos datos
* Analizar datos
* Tomar decisiones empresariales
* Preparar datos para el aprendizaje automático (ML)
* Facilitar el descubrimiento, la generación de informes, y el análisis

Etapas del proceso ETL

1. Extraer datos de varias fuentes
2. Transformar los datos según las reglas de negocio
3. Cargar los datos en un almacén de datos de destino

Herramientas ETL

[Google Cloud ofrece](https://cloud.google.com/products/dataflow) herramientas ETL como Dataflow, Cloud Data Fusion, y Dataproc.

Consideraciones sobre la extracción

* Existen dos maneras de extraer los datos: totales o incrementales
* La manera de extraer los datos depende del tipo de datos, su formato, y sus características
* Extraer todos los datos en cada ejecución puede ser ineficiente si se trabajan con muchos datos de diferentes formatos

WebGrafias: [https://aws.amazon.com](https://aws.amazon.com/es/what-is/data-science/%23:~:text=La%20ciencia%20de%20datos%20es,analizar%20grandes%20cantidades%20de%20datos.) \ [https://aws.amazon.com](https://aws.amazon.com/es/what-is/etl/#:~:text=es%20AWS%20Glue?-,¿Qué%20es%20ETL?,central%20llamado%20almacenamiento%20de%20datos.)

THE END