

# Investigación de Ciencias de Datos

Adrian Posada Torres

2795599

Luis Sánchez

20/02/2025

## Tabla de contenido

Investigación de Ciencias de Datos.....	1
<b>Ambiente personal de aprendizaje .....</b>	<b>6</b>
<b>Métodos de aprendizaje.....</b>	<b>6</b>
<i>Aprendizaje basado en problemas.....</i>	<i>6</i>
<i>Aprendizaje Cooperativo:.....</i>	<i>6</i>
<i>Aprendizaje Experiencial:.....</i>	<i>6</i>
<i>Aprendizaje Autodidacta:.....</i>	<i>6</i>
<i>Aprendizaje activo: .....</i>	<i>7</i>
<b>Tipos de Aprendizajes .....</b>	<b>7</b>
<i>Visual: .....</i>	<i>7</i>
<i>Auditivo:.....</i>	<i>7</i>
<i>Kinestésico: .....</i>	<i>7</i>
<input type="checkbox"/> <i>Memorístico:.....</i>	<i>8</i>
<b>¿Qué es Base de Datos? .....</b>	<b>8</b>
<i>Propósito .....</i>	<i>8</i>
<b>¿Tipo de Base de Datos? .....</b>	<b>8</b>
<i>Relacionales .....</i>	<i>8</i>
<i>No Relacionales (NoSQL).....</i>	<i>8</i>
<b>¿Qué es Ciencia de Datos? .....</b>	<b>9</b>

<b>Estadística.....</b>	<b>9</b>
<b>Algoritmos: .....</b>	<b>9</b>
<b>Bases de Datos.....</b>	<b>9</b>
<b>¿Qué es Python – Historia?.....</b>	<b>9</b>
<b>Lenguaje de Programación: .....</b>	<b>9</b>
<b>Versatilidad:.....</b>	<b>9</b>
<b>Comunidad Activa.....</b>	<b>9</b>
<b>Historia de Python: .....</b>	<b>10</b>
<b>Evolución de las Versiones .....</b>	<b>10</b>
<b>Python 1.x (1994-2000): .....</b>	<b>10</b>
<b>Python 2.x (2000-2010): .....</b>	<b>10</b>
<b>Python 3.x (desde 2008):.....</b>	<b>11</b>
<b>Bases de Datos Relacionales (SQL).....</b>	<b>11</b>
<input type="checkbox"/> <b>¿Qué son? .....</b>	<b>11</b>
<input type="checkbox"/> <b>¿Cómo se integran?.....</b>	<b>11</b>
<input type="checkbox"/> <b>¿Para qué se usan? .....</b>	<b>12</b>
<b>Bases de Datos No Relacionales (NoSQL).....</b>	<b>12</b>
<input type="checkbox"/> <b>¿Qué son? .....</b>	<b>12</b>
<input type="checkbox"/> <b>¿Cómo se integran?.....</b>	<b>12</b>
<input type="checkbox"/> <b>¿Para qué se usan? .....</b>	<b>12</b>

Diferencias Claves entre SQL y NoSQL.....	13
Mapa Conceptual.....	14
¿Cómo se usa? .....	14
<input type="checkbox"/> <i>Identificar el tema principal:</i> .....	14
<input type="checkbox"/> <i>Agregar conceptos secundarios</i> .....	14
<input type="checkbox"/> <i>Establecer conexiones</i> .....	14
<input type="checkbox"/> <i>Refinar y mejorar:</i> .....	14
¿Para qué sirve?.....	14
Características de un Mapa Conceptual .....	15
Elementos de un Mapa Conceptual.....	15
Tipos de Mapas Conceptuales.....	15
¿Cómo se una en diferentes áreas? .....	16
Replit +Github. ....	16
¿Qué es Replit? .....	16
¿Qué es GitHub? .....	16
¿Cómo funciona la integración entre Replit y GitHub? .....	17
<input type="checkbox"/> <i>Clonar repositorios</i> .....	17
<input type="checkbox"/> <i>Sincronizar cambios</i> .....	17
<input type="checkbox"/> <i>Colaborar en proyectos</i> .....	17
<input type="checkbox"/> <i>Hacer commits y push</i> .....	17

<b>Ventajas de usar Replit con GitHub .....</b>	<b>17</b>
---	-----------

# Ambiente personal de aprendizaje.

Se refiere al conjunto de herramientas, recursos, estrategias y personas que cada individuo utiliza para aprender. Incluye: Recursos digitales, Redes sociales y comunidades de aprendizaje, Métodos de estudio y estrategias personales, Fuentes de información como libros, revistas y experiencias prácticas.

## Métodos de aprendizaje.

- ***Aprendizaje basado en problemas:***

Resuelve problemas del mundo real. Mejora el pensamiento crítico.

- ***Aprendizaje Cooperativo:***

Trabaja en equipo para aprender. Fomenta la comunicación y el respeto.

- ***Aprendizaje Experiencial:***

Aprende haciendo y reflexionando. Aplica la teoría a la práctica.

- ***Aprendizaje Autodidacta:***

Se aprende de manera independiente a través de la investigación y la práctica.

- ***Aprendizaje activo:***

El estudiante participa activamente (proyectos, debates, experimentos).

## **Tipos de Aprendizajes.**

- ***Visual:***

Características: Aprendes mejor con imágenes y gráficos. Prefieres mapas mentales y diagramas.

Estrategias: Usa videos y presentaciones visuales. Toma notas con dibujos y símbolos.

- ***Auditivo:***

Características: Aprendes mejor escuchando y hablando. Prefieres debates y discusiones.

Estrategias: Escucha podcasts y audiolibros. Participa en grupos de estudio y explica conceptos.

- ***Kinestésico:***

Características: Aprendes mejor moviéndote y tocando. Prefieres actividades prácticas y experimentos.

Estrategias: Realiza experimentos y proyectos. Usa modelos y simulaciones.

- **Memorístico:**

Repetición: Repite la información varias veces.

Asociación: Conecta la nueva información con algo conocido.

Mnemotecnia: Crea acrónimos y rimas.

## ¿Qué es Base de Datos?

Una base de datos es un conjunto organizado de datos estructurados, almacenados electrónicamente en un sistema informático. Permite almacenar, acceder, gestionar y actualizar datos de manera eficiente.

**Propósito:** Su propósito principal es facilitar el acceso, la gestión y la actualización de grandes cantidades de información. Se utilizan en una amplia variedad de aplicaciones, desde la gestión de inventarios hasta la administración de información de clientes.

## ¿Tipo de Base de Datos?

**Relacionales:** Organizan los datos en tablas con filas y columnas, utilizando claves primarias y foráneas para establecer relaciones entre las tablas. Ejemplos: MySQL, PostgreSQL.

**No Relacionales (NoSQL):** Almacenan datos en formatos diferentes a las tablas, como documentos, grafos o clave-valor. Son más flexibles y escalables. Ejemplos: MongoDB, Cassandra.



## ¿Qué es Ciencia de Datos?

**Estadística:** La ciencia de datos utiliza métodos estadísticos para analizar datos y extraer patrones significativos.

**Algoritmos:** Se emplean algoritmos de aprendizaje automático para predecir resultados y tomar decisiones basadas en los datos.

**Bases de Datos:** La ciencia de datos necesita acceso a datos almacenados en bases de datos para realizar análisis y modelos.

## ¿Qué es Python – Historia?

**Lenguaje de Programación:** Python es un lenguaje de programación de alto nivel, interpretado, de propósito general y fácil de aprender.

**Versatilidad:** Se utiliza en una amplia variedad de aplicaciones, desde desarrollo web hasta ciencia de datos e inteligencia artificial.

**Comunidad Activa:** Cuenta con una gran comunidad de desarrolladores que contribuyen con bibliotecas y herramientas.

## Historia de Python:

Python nació de la mente de Guido van Rossum. Él buscaba un sucesor para el lenguaje ABC. Este era usado en el centro de investigación CWI.

Su creación comenzó en diciembre de 1989. Van Rossum lo concibió como un proyecto "hobby". Quería un lenguaje fácil de usar.

Se inspiró en Monty Python. De ahí proviene el nombre del lenguaje. Buscaba algo corto, único y ligeramente misterioso.

## Evolución de las Versiones:

***Python 1.x (1994-2000):*** Python 1.0 fue lanzado en enero de 1994. Esta versión consolidó muchas de las características que se encontraban en la versión 0.9.0 y presentó nuevas funcionalidades, incluyendo:

- Comprensión de listas.
- Soporte para módulos de extensión.

***Python 2.x (2000-2010):*** Python 2.0, lanzado en octubre de 2000, marcó el inicio de una nueva era para el lenguaje. Esta versión trajo mejoras significativas y nuevas características, incluyendo:

- **Recogida de basura:** Un sistema de recolección de basura para el manejo automático de memoria.
- **List comprehensions:** Para la creación concisa de listas.
- **El módulo unittest:** Introducción de un framework de pruebas unitarias estándar.

***Python 3.x (desde 2008):*** trajo cambios importantes respecto a Python 2, aunque no era totalmente compatible hacia atrás. Sus principales mejoras incluyen:

- **print() como función,** requiriendo paréntesis.
- **División de enteros,** ahora devuelve un float por defecto.
- **Soporte Unicode,** haciendo que las cadenas sean Unicode por defecto.

## Bases de Datos Relacionales (SQL)

- **¿Qué son?**

Son bases de datos estructuradas en tablas con filas y columnas, donde cada fila representa un registro y cada columna un atributo. Utilizan SQL (Structured Query Language) para gestionar la información.

- **¿Cómo se integran?**

Se conectan a aplicaciones mediante controladores y lenguajes de programación como Python, Java, PHP o Node.js. Se usa SQL para realizar consultas y manipulaciones de datos.

- **¿Para qué se usan?**

Son ideales para aplicaciones que requieren integridad, consistencia y relaciones bien definidas, como sistemas bancarios, ERPs, CRMs y comercio electrónico.

- **Ejemplos de fabricantes:**
  - **MySQL** (Oracle)
  - **PostgreSQL**
  - **Microsoft SQL Server**
  - **Oracle Database**

## **Bases de Datos No Relacionales (NoSQL)**

- **¿Qué son?**

No almacenan datos en tablas, sino en formatos como documentos, grafos, clave-valor o columnas. Son más flexibles y escalables.

- **¿Cómo se integran?**

Se acceden mediante APIs y drivers específicos, con lenguajes como JavaScript, Python o Go. Usan modelos distintos a SQL, como JSON o BSON.

- **¿Para qué se usan?**

Son ideales para big data, análisis en tiempo real, aplicaciones web y redes sociales. Se usan en sistemas que manejan grandes volúmenes de datos y requieren escalabilidad horizontal.

- **Ejemplos de fabricantes:**
  - **MongoDB** (documentos JSON)

- **Redis** (clave-valor)
- **Cassandra** (columnares, usado por Facebook)
- **Neo4j** (grafos, usado en redes sociales)

## Diferencias Claves entre SQL y NoSQL.

Característica	SQL (Relacionales)	NoSQL (No Relacionales)
Modelo de datos	Basado en tablas con filas y columnas	Flexible (documentos, grafos, clave-valor, columnas)
Lenguaje	SQL	JSON, BSON, APIs propietarias
Escalabilidad	Vertical (aumentando recursos del servidor)	Horizontal (distribuyendo en varios servidores)
Transacciones	ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)	Eventualmente consistente, CAP Theorem
Uso recomendado	Finanzas, ERP, CRM, e-commerce	Big Data, redes sociales, IoT, aplicaciones en tiempo real

# Mapa Conceptual

Un mapa conceptual es una herramienta gráfica que organiza y representa visualmente información, mostrando relaciones entre conceptos de manera jerárquica. Se compone de nodos (conceptos clave) y líneas o enlaces que los conectan, a menudo con palabras de enlace que explican la relación entre ellos.

## ¿Cómo se usa?

- **Identificar el tema principal:** Coloca el concepto central en la parte superior o centro del mapa.
- **Agregar conceptos secundarios:** Conecta ideas relacionadas al concepto principal, formando una estructura jerárquica.
- **Establecer conexiones:** Usa líneas y palabras de enlace para explicar cómo se relacionan los conceptos.
- **Refinar y mejorar:** Organiza la información de forma clara y lógica para facilitar su comprensión.

## ¿Para qué sirve?

- Organizar ideas y estudiar de manera estructurada.
- Facilitar la comprensión de temas complejos.
- Mejorar la memoria y el aprendizaje visual.

- Relacionar conceptos de manera lógica.

## Características de un Mapa Conceptual

***Jerárquico:*** Los conceptos se organizan de lo general a lo específico.

***Conectado:*** Usa líneas y palabras de enlace para mostrar relaciones.

***Sencillo y claro:*** Debe ser fácil de entender y sintetizar la información.

***Visualmente estructurado:*** Usa colores, formas y tamaños para resaltar ideas.

## Elementos de un Mapa Conceptual

⚡ ***Conceptos:*** Palabras clave o ideas principales.

⚡ ***Enlaces o Conectores:*** Líneas que unen los conceptos.

⚡ ***Palabras de Enlace:*** Explican la relación entre los conceptos.

⚡ ***Proposiciones:*** Frases formadas por la unión de conceptos y palabras de enlace.

## Tipos de Mapas Conceptuales

***Jerárquico:*** Va de lo más general a lo más específico.

***Araña:*** Tiene el tema principal en el centro y conceptos alrededor.

***Secuencial:*** Representa procesos paso a paso.

***Sistemático:*** Muestra interacciones entre conceptos en diferentes niveles.

## ¿Cómo se usa en diferentes áreas?

📖 Educación: Para resumir temas y estudiar mejor.

🏢 Empresas: Para organizar proyectos y estrategias.

🔬 Ciencia e investigación: Para analizar relaciones entre teorías.

# Replit +Github.

## ¿Qué es Replit?

Replit es un entorno de desarrollo en línea (IDE en la nube) que permite a los programadores escribir, ejecutar y colaborar en código desde un navegador sin necesidad de instalar software adicional. Es compatible con más de 50 lenguajes de programación, incluidos Python, JavaScript, C++, y más.

## ¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo basada en Git, que permite a los desarrolladores almacenar, gestionar y compartir código en repositorios. Es ampliamente utilizado para el control de versiones, la colaboración en proyectos y el despliegue de software.



## ¿Cómo funciona la integración entre Replit y GitHub?

Replit ofrece integración con GitHub para que los desarrolladores puedan:

- **Clonar repositorios** directamente desde GitHub a Replit.
- **Sincronizar cambios** entre un proyecto en Replit y un repositorio

en GitHub.

- **Colaborar en proyectos** con otros programadores usando Git y

GitHub.

- **Hacer commits y push** a GitHub desde el mismo entorno de

Replit.

## Ventajas de usar Replit con GitHub

**Desarrollo en la nube:** No necesitas instalar nada en tu computadora.

**Colaboración en tiempo real:** Múltiples usuarios pueden trabajar en un mismo proyecto.

**Fácil integración:** Puedes sincronizar tu código sin problemas.

**Compatible con múltiples lenguajes:** Admite más de 50 lenguajes de programación.

**Ambiente de ejecución en línea:** Puedes probar y ejecutar código en Replit sin configuraciones adicionales.