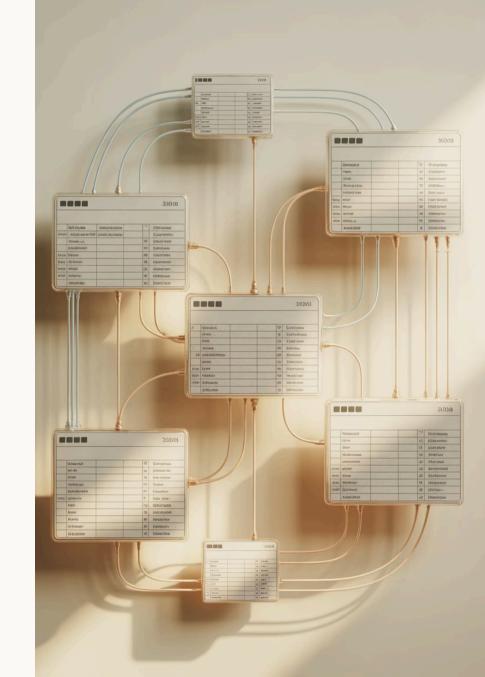
# Definiciones básicas de elementos del Modelo Relacional

**Objetivo:** Introducir los conceptos fundamentales que definen la estructura y el funcionamiento de las bases de datos relacionales, destacando el rol de las **tablas**, **tuplas**, **claves** e **índices**, y su vínculo con estructuras de datos comunes.

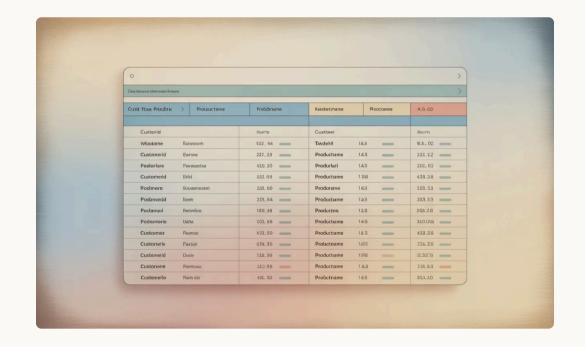




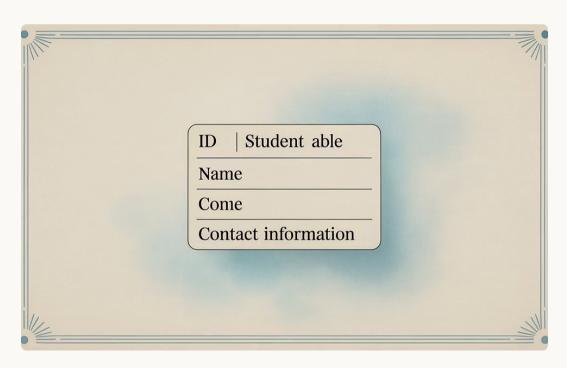
### ¿Qué es una Tabla o Relación?

Una **tabla** en una base de datos representa un conjunto de datos organizados en **filas** y **columnas**. Cada tabla tiene un **nombre** (por ejemplo: Alumnos) y una **lista de atributos** o columnas (como DNI, Apellido, Nombre, Teléfono, Acceso).

**Importante**: El orden de los atributos en el esquema no afecta los datos almacenados, pero sí su definición.

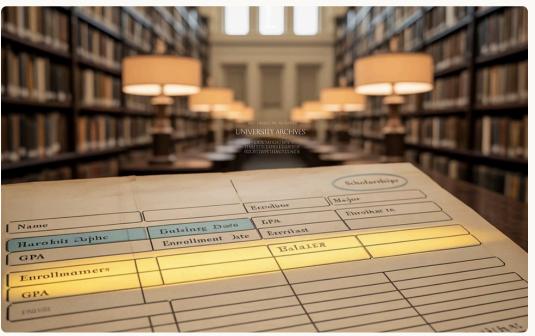


### Instancia de una Tabla: la Tupla



#### Concepto de Tupla

Cada **registro individual** o fila en una tabla se llama **tupla**. Representa una entidad específica (por ejemplo, un alumno) y contiene un **conjunto de valores**, uno por cada atributo de la tabla.



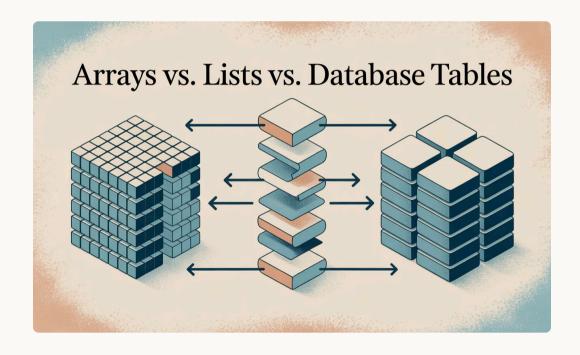
#### Ejemplo Práctico

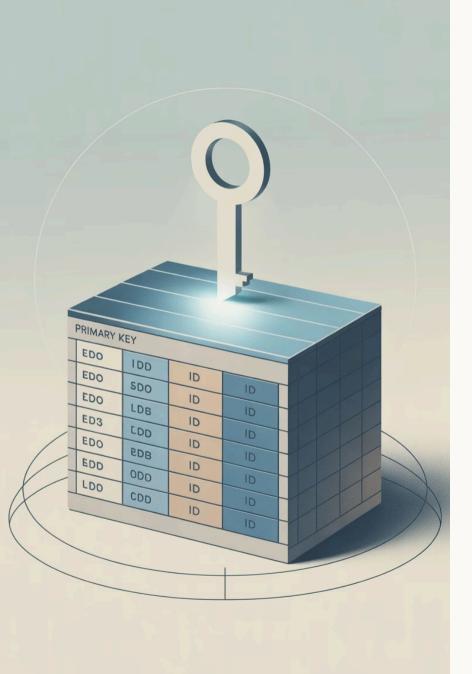
Una tupla de la tabla Alumnos podría ser: (35234123, "Gómez", "Ana", "1134567890", true)

### Colecciones de Datos: Vectores y Listas

Las tablas pueden pensarse como una **colección de tuplas**. En programación, estamos familiarizados con **vectores** o **listas**, donde se almacenan elementos de forma secuencial o dinámica.

Relación útil: Una **tabla** es una colección como una lista, pero cada elemento (tupla) tiene múltiples componentes (atributos).





### La Clave: Identificador Único



#### Definición de Clave

Una **clave** es un atributo (o conjunto de atributos) que **identifica de forma única** a cada tupla. En Alumnos, el DNI puede ser clave, porque no se repite.



#### Claves Artificiales

Cuando no hay un identificador natural, se pueden usar **claves artificiales** (como un ID autogenerado).

### Índices: Localización Eficiente de Datos

Un **índice** funciona como el índice de un libro: permite **buscar rápidamente** la ubicación de ciertos datos. Ayudan a **acelerar consultas**, especialmente sobre columnas usadas frecuentemente en búsquedas.

#### Ejemplo:

- El número de factura debe ser único, ideal para un índice.
- El **CUIT** del cliente puede repetirse (porque un cliente puede tener muchas facturas).



## Estructuras de Almacenamiento y Búsqueda

Estructura	Características principales	
Vector	Tamaño fijo, búsqueda secuencial o binaria (si está ordenado)	
Lista ordenada	Tamaño dinámico, requiere más procesamiento para ordenar	
Árbol binario	Alta eficiencia en búsqueda, no permite duplicados, inserción dinámica	

Cada estructura tiene ventajas y limitaciones según el contexto del sistema.

### Profundizando en las Tablas



#### Definición

Estructura fundamental que organiza datos en filas y columnas



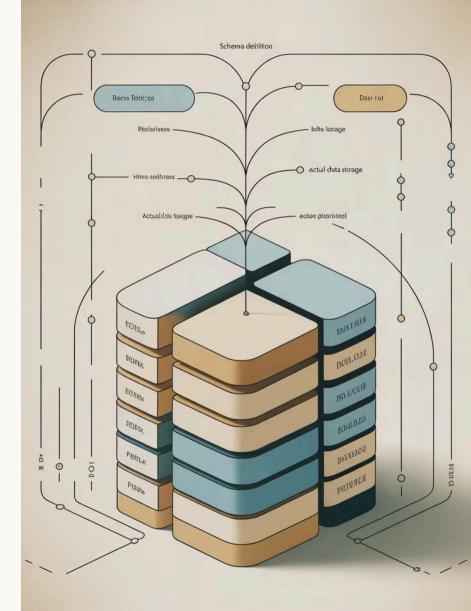
#### **Atributos**

Columnas que definen las propiedades de los datos almacenados



#### **Tuplas**

Filas que contienen los valores específicos para cada atributo



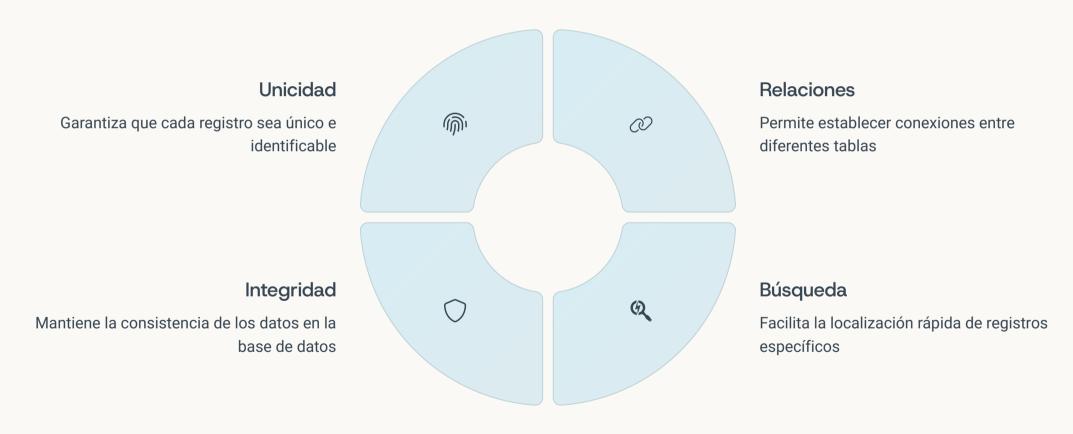
# Ejemplo Práctico: Tabla Alumnos

Veamos un ejemplo concreto de la tabla "Alumnos" mencionada anteriormente:

DNI	Apellido	Nombre	Teléfono	Acceso
35234123	Gómez	Ana	1134567 890	true
28456789	Pérez	Juan	1145678 901	true
39876543	López	María	1156789 012	false



### Importancia de las Claves



# Índices en Acción

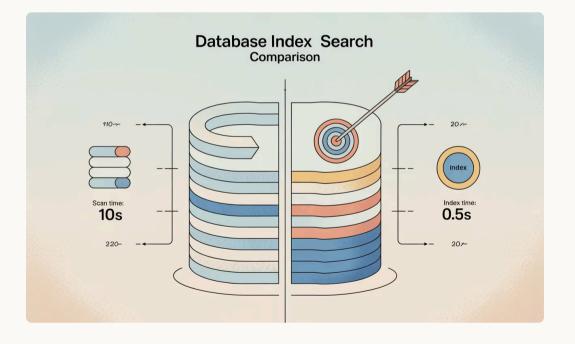
#### Sin Índice

Buscar un alumno por DNI requeriría revisar cada registro secuencialmente hasta encontrarlo.

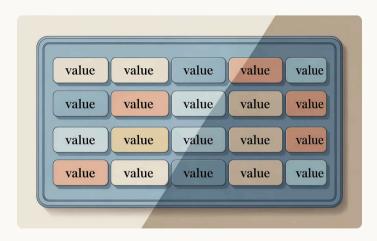
FOR cada tupla en Alumnos
IF tupla.DNI = 35234123 THEN
RETURN tupla
END IF
END FOR

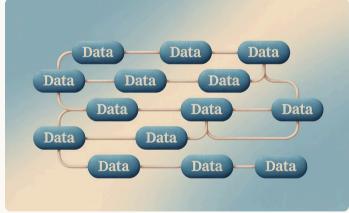
#### Con Índice

Un índice sobre DNI permite localizar el registro directamente, similar a buscar en un diccionario.



### Comparación de Estructuras de Datos







#### **Vectores**

Estructura de tamaño fijo con acceso directo a elementos por índice. Eficiente para búsquedas si está ordenado, pero ineficiente para inserciones y eliminaciones.

#### Listas

Estructura dinámica que facilita inserciones y eliminaciones, pero requiere recorrido secuencial para búsquedas.

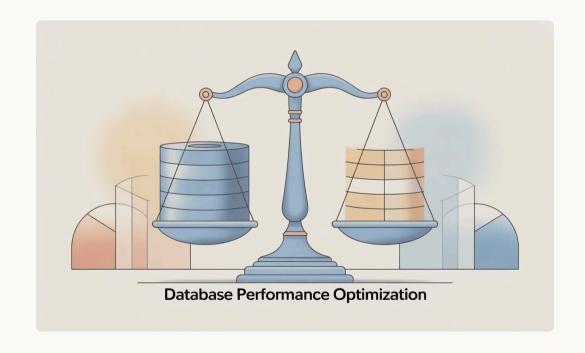
#### Árboles

Estructura jerárquica que optimiza búsquedas, inserciones y eliminaciones, ideal para índices de bases de datos.

### Reflexión Final

Los sistemas de bases de datos relacionales buscan **almacenar** y **recuperar información** de manera eficiente. La elección de claves, índices y estructuras de datos influye directamente en el **rendimiento** y **complejidad** del sistema.

La comprensión de estos conceptos fundamentales es esencial para diseñar bases de datos eficientes y escalables.



### Conclusión

Este recorrido inicial nos da un marco conceptual básico para comprender cómo funcionan las bases de datos relacionales. Seguiremos explorando estos temas con mayor profundidad a lo largo de la asignatura.

Tablas y Tuplas

Estructuras fundamentales para organizar y representar datos

Claves e Índices

Mecanismos para identificar y acceder eficientemente a los datos

Estructuras de Datos

Fundamentos que sustentan la implementación de bases de datos eficientes

