

{desafío}
latam_

Modelamiento y Normalización _

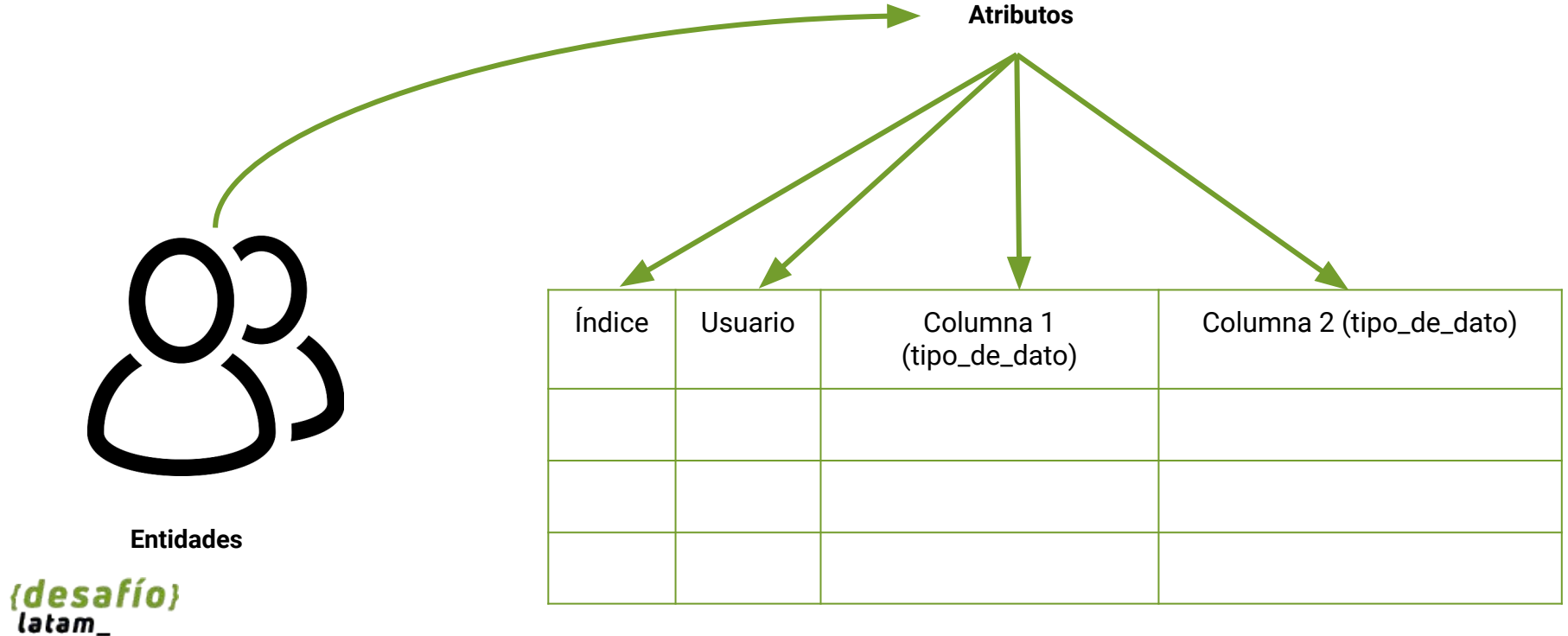


Modelación

Preliminares

- Modelación: Generar una representación de la realidad para un caso de uso específico.
- Antes de ingresar y preprocesar los datos para futuras aplicaciones, debemos modelar la naturaleza del problema.

Entidades y Atributos de una tabla



Pasos en la modelación

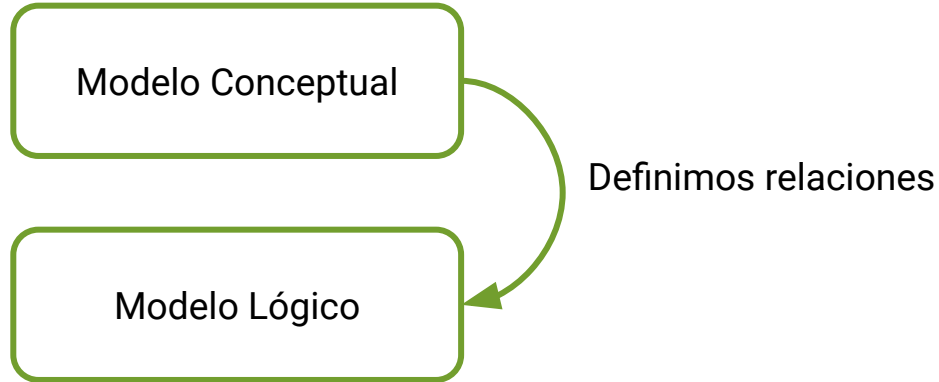
Modelo Conceptual

Pasos en la modelación

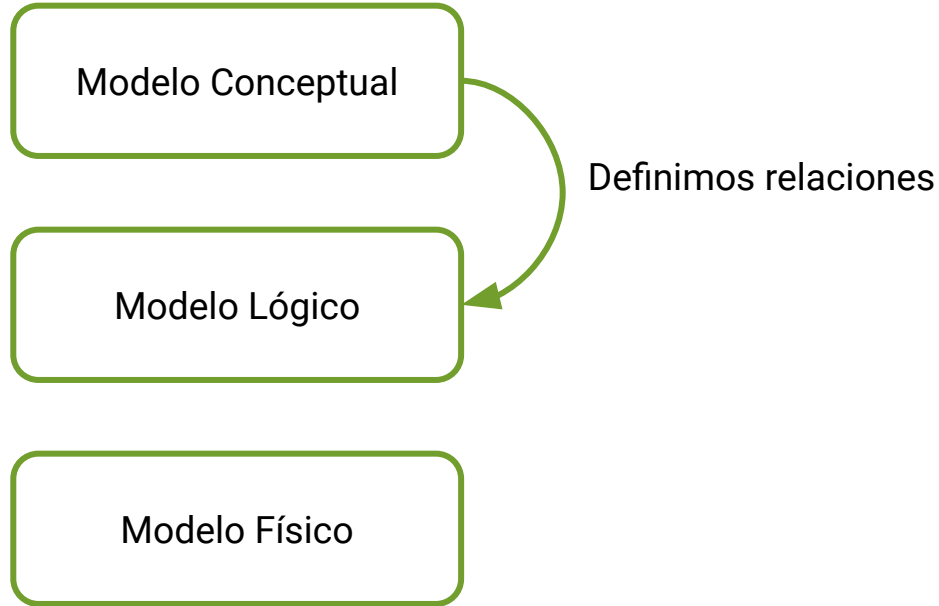
Modelo Conceptual

Modelo Lógico

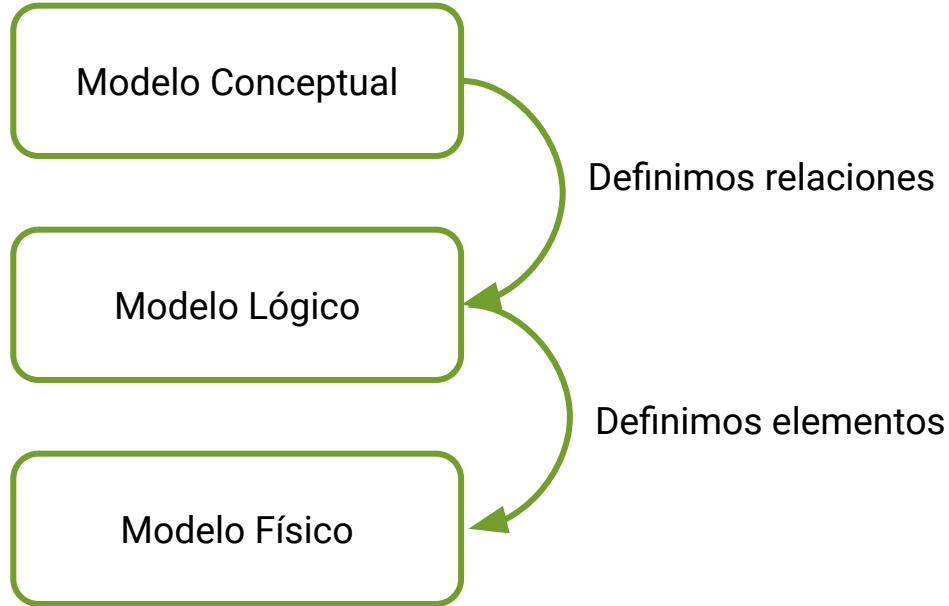
Pasos en la modelación



Pasos en la modelación



Pasos en la modelación

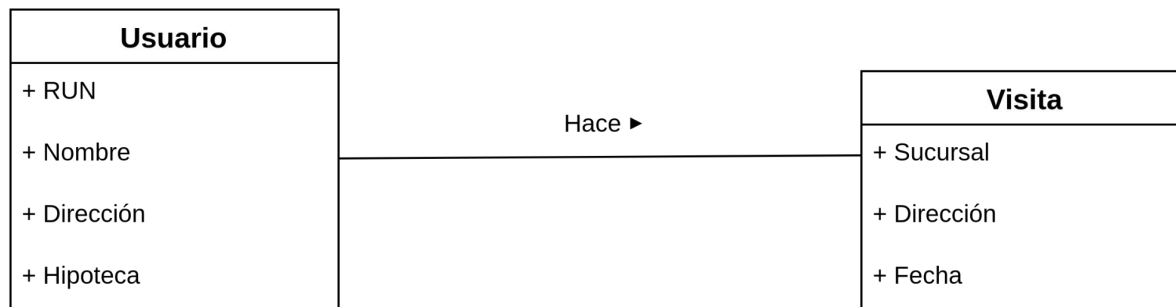


Modelo Conceptual

- Bajamos los requerimientos del cliente en un modelo que represente las entidades y atributos mediante tablas.
- Mediante éste logramos reducir la complejidad de los requerimientos.

Aplicación del Modelo Conceptual

- Contextualización del requerimiento del cliente: Deseamos levantar datos sobre la gestión de clientes actuales en un banco.
- Los datos existentes son: RUN, nombre, dirección, hipoteca, fecha, lugar y sucursal de visitas al banco.

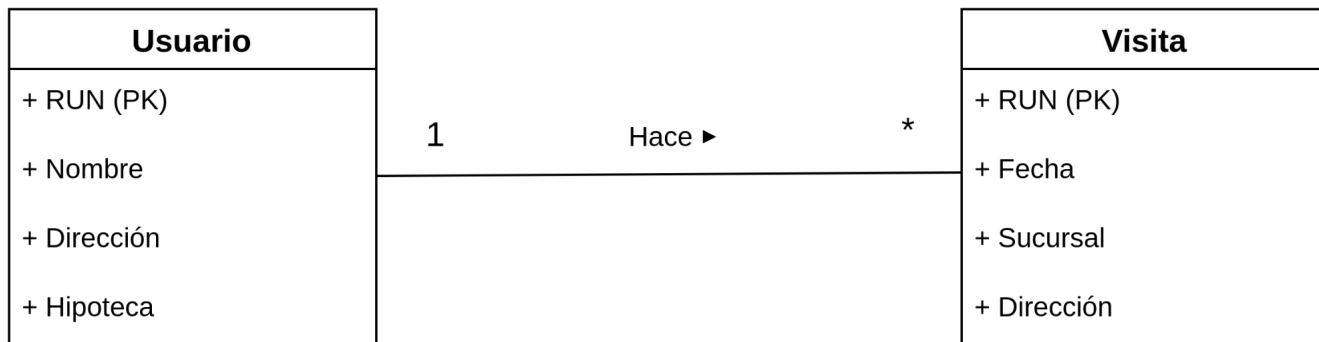


Modelo Lógico

- Con el modelo conceptual no estamos habilitados para armar nuestra base de datos.
- El siguiente paso es definir cómo vamos a lograr las asociaciones existentes entre las tablas.
- Preguntas que busca resolver:
 - ¿Cuáles serán nuestras claves primarias?
 - ¿Cómo implementaremos las relaciones existentes en el modelo?

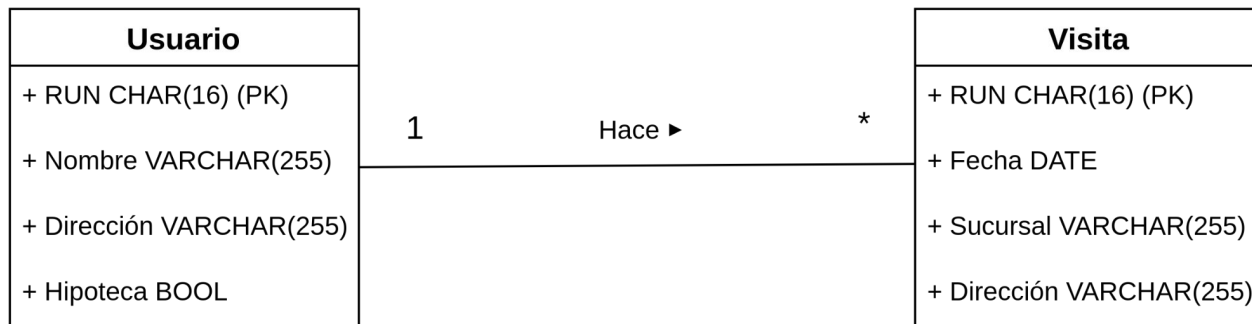
Aplicación del Modelo Lógico

- Definimos el flujo entre registros.
- El punto clave es considerar que un usuario específico puede tener múltiples visitas.



Aplicación del Modelo Físico

- Posterior a la definición de cómo asociamos las tablas mediante claves primarias y foráneas, estamos habilitados para definir los campos y sus tipos de datos correspondientes.



Normalización

Preliminares

- Problema recurrente en el modelado de datos: Existencia de problemas de redundancia y falta de integridad de datos.
- Solución: Iterar el modelado en el proceso de normalización.
- Para ello hacemos uso de las Formas Normales.

Nomenclatura

- Tabla original (sin claves)

```
Tabla(atributo_1, atributo_2, ..., atributo_n)
```

- Tabla con clave primaria

```
Tabla(#clavePrimaria, atributo_1, atributo_2, ..., atributo_n)
```

- Tabla con claves primarias y foráneas

```
Tabla(#clavePrimaria, claveForánea, ..., atributo_n)
```

- Tabla con claves primarias y compuesta

```
Tabla(#clavePrimaria, #clavePrimariayForánea, ..., atributo_n)
```

Ejemplo: Nuestros datos

- Structured Query Language: Creado para la manipulación de bases de datos relacionales.
- Se considera un lenguaje declarativo:
 - Generamos instrucciones para crear, actualizar, borrar y solicitar datos de tablas

Primera Forma Normal

- Normalización en primera forma:
 - Cada campo debe tener un solo valor.
 - No pueden existir grupos repetitivos.
- Solución: preservar cada atributo repetido como entradas en una nueva tabla.

Ejemplo: Aplicando la primera forma normal

- Volvamos a nuestro ejemplo

```
Cliente(nombre, #run, dirección, hipoteca {fecha, sucursal, dirección})
```

- Tabla cliente con 1FN

```
Cliente(nombre, #run, dirección, hipoteca)
```

- Tabla visita con 1FN

```
Visita(#run, #fecha, #sucursal, dirección)
```

Segunda Forma Normal

- Solución en segunda forma:
 - Se debe cumplir la primera forma normal
 - Los atributos deben depender de toda la clave primaria, y no solo una parte de ella.
- Hay que fijarse que existan claves primarias compuestas, dado que generar dependencia parcial.

Ejemplo: Aplicando la segunda forma normal

- Nuestro problema se presenta en la tabla visita

```
Visita(#run,#fecha, #sucursal, dirección)
```

```
Sucursal(#sucursal, dirección)
```

Tercera Forma Normal

- Solución de la tercera forma normal:
 - Debe satisfacer la segunda forma normal
 - Toda entidad debe depender directamente de la clave primera

Ejemplo: Aplicando la tercera forma normal

```
Cliente(nombre, #run, dirección)
```

```
Propiedad(#dirección, hipoteca)
```

```
Visita(#run, #fecha, #sucursal)
```

```
Sucursal(#sucursal, dirección)
```


Desnormalización

- Es una técnica de optimización donde se añade deliberadamente redundancia a una o más tablas.
- Se considera una técnica de optimización dado que puede reducir los tiempos de consultas en múltiples tablas a una sola.
- Desnormalizar no implica evitar la normalización: Es un proceso posterior a este que depende de nuestras necesidades como gestor de bases de datos.

Usos de Desnormalización

- Trueque entre redundancia y normalización.
- En Ciencia de Datos, a veces nos encontraremos con situaciones donde se debe trabajar con datos desnormalizados. Algunos:
 - Estructuras de datos multinivel.
 - Estructuras de datos panel.
 - Estructuras de redes sociales.

Estructuras de datos Desnormalizadas

$$\text{Ingreso}_{ijk} = \beta_0 + \zeta_j \text{Mes} + \gamma_k \text{Año} + \varepsilon_i + \delta_j + \nu_k$$

Individuo	Mes	Año	Ingreso
Juan	Enero	1999	1000
Juan	Febrero	1999	1200
...
Juan	Diciembre	2005	45000
Claudio	Enero	1999	12000

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com