

# Desafío Profesional Full Stack

## Sprint 6 | Bases de datos

¿Qué? ¿Qué de a poco estamos llegando al final! 🙌😬🙌 ¡Es hora de la **sexta iteración** del **Trabajo Integrador**!

Nos toca ahora dejar atrás el viejo y querido JSON para pasar a algo más profesional que escale mejor cuando nuestro sitio salga a producción. 🚀 Ya casi estamos para salir, ya casi. 😎🔥

En este sprint estaremos trabajando con SQL por un lado y con Sequelize por el otro. ¡**Vamos!**

### Requisitos previos

1. **Fuente de datos de usuarios y productos:** los archivos JSON serán su fuente de inspiración. Recuerden que en el sprint 3 tienen una sugerencia de los campos mínimos.
2. **CRUD de productos y usuarios:** que hoy funcionan para JSON y sobre los cuales implementarán la magia de Sequelize. ✨👷‍♂️✨

### Objetivos

Durante esta iteración su foco será el de crear e implementar la base de datos de su sitio web.

En la **primera parte**, van a estar pensando en la **estructura** que será necesaria para que la base de datos cumpla con los requisitos del negocio: tablas, campos, tipos de datos y relaciones.

La **segunda parte** la van a pasar implementando la base de datos que crearon en la primera parte, utilizando el módulo de **Sequelize**.

### Consignas

1. Diagrama de base de datos

Toda buena base de datos empieza en la mesa de dibujo. Tendrán que armar el **Diagrama de Entidades y Relaciones (DER)**. Piensen en un buen diseño, armen un diagrama legible, con relaciones correctas y las claves foráneas para representarlas.

Recuerden que luego deberán implementar Sequelize y que, por lo general, los ORMs como este trabajan mejor con los nombres de tablas en inglés.

Les proponemos la siguiente estructura, aunque la pueden ajustar a la necesidad de su proyecto. 😊

- Usuarios (recuerden ver los campos sugeridos en el sprint 3).
- Productos (recuerden ver los campos sugeridos en el sprint 3).
- Tablas secundarias (según lo requiera su proyecto).
  - Para productos: categorías, marcas, colores, talles, etc.
  - (Opcional) Para usuarios: categorías.
- (Opcional) Carrito de compras.
  - Con detalle de quién hizo la compra, cantidad de ítems y precio total.
  - (Opcional) Productos de cada carrito de compras.

Les sugerimos utilizar [draw.io](https://draw.io) ya que es fácil de usar y soporta **diagramas DER**. También puede ser útil usar la interfaz que provee [MySQL Workbench](https://mysqlworkbench.dev/) para crear el diagrama y a partir de este crear la estructura de la base de datos.

**Entregable:** diagrama de entidad-relación de su base de datos en formato PDF.

## 2. Script de estructura

Tomando como referencia el diagrama del punto anterior, tienen que escribir las sentencias de SQL que crearán las tablas y sus relaciones.

- Deberá incluir la creación de la base de datos (**create database...**).
- Deberá incluir la creación de todas las tablas del sitio (**create table...**).
- Deberá incluir los tipos de datos de los campos y sus restricciones (**primary keys, (not) null, unique, default**, etc).
- Deberá incluir las relaciones entre las diferentes tablas (**foreign keys**).

**Entregable:** archivo **structure.sql** que permita crear la base de datos completa.

## 3. Script de datos

Ya tenemos la estructura, ahora faltan los datos. El script de datos permite que cualquier desarrollador (o docente 😊) descargue el proyecto, ejecute el script y ya pueda ver el sitio

funcionado sin más pasos.

El script debería:

- Poblar la tabla de usuarios.
- Poblar la tabla de productos.
- Poblar las tablas secundarias (categorías, marcas, colores, talles, etc).

Una vez definidos los campos de sus tablas, nuevamente pueden utilizar [Mockaroo](#), pero esta vez para generar el archivo SQL con datos. 😊👉

**Entregable:** archivo con extensión data.sql que permita poblar la base de datos.

## 4. Creación de carpeta Sequelize y archivos de modelos

Mediante la herramienta **sequelize-cli** deberán crear la carpeta que contenga los archivos de configuración de **Sequelize**. Una vez configurado Sequelize, se debe crear los archivos de modelos para explicarle a Sequelize cómo es la estructura de la base de datos.

La carpeta database deberá incluir:

- Los archivos de configuración para que Sequelize se conecte a la base de datos.
- Los archivos de modelos para representar las tablas de:
  - Usuarios.
  - Productos.
  - Tablas secundarias (categorías, marcas, colores, talles, etc).
  - (Opcional) Carrito de compras.
  - (Opcional) Productos de cada carrito de compras.
- Los modelos deben incluir todas las relaciones existentes en la base de datos.

**Entregable:** carpeta database que incluya los archivos de configuración y archivos de modelos junto con sus relaciones.

## 5. ¡CRUD!

Ya es hora de tener un CRUD como la gente. Qué bueno que tenemos a Sequelize de nuestro lado. Les pedimos que en su sitio se pueda:

- Para productos:

- Crear
- Editar
- Eliminar
- Listar
- Ver el detalle
- Buscar
- Para usuarios:
  - Crear
  - Editar
  - Ver el perfil
- (Opcional) CRUDs de tablas secundarias.
- (Opcional) Agregar paginado a los listados y buscadores.

**Entregable:** rutas, controladores y vistas necesarias para que suceda lo detallado previamente utilizando Sequelize para trabajar con la base de datos.

## Cierre

¡Wow! Las bases de datos son una herramienta que permite que el almacenamiento de los datos crezca de forma escalable y veloz. Puede ser un trabajo arduo (sobre todo si ya teníamos cosas hechas en JSON... 😊), pero el resultado es increíble. No solo es increíble porque nuestra aplicación está lista para crecer radicalmente, sino porque **¡completamos un back-end completo!**

Pensar, diseñar, implementar una base de datos no es sencillo, pero un buen diseño inicial va a asegurar nuestro éxito en el tiempo. Además, implementarlo en Sequelize nos permite correr un poco de SQL y aprovechar todas las prestaciones de un buen ORM.

Si llegamos hasta acá, felicitaciones, somos cracks. 🤖🙌✨