DigitalHouse > Coding School

Desarrollo Web Full Stack Node

Trabajo integrador

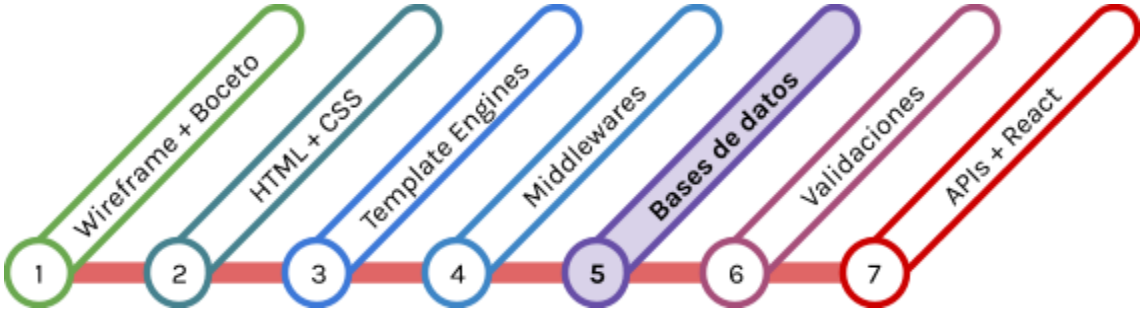
Trabajo Integrador - Sprint 5

Introducción

¿¿Qué?? ¡¡Que de a poco estamos llegando al final!! ✋😳✋. ¡Es hora de la **quinta 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!**


Wireframe + Boceto
HTML + CSS
Template Engines
Middlewares
Bases de datos
Validaciones
APIs + React
1
2
3
4
5
6
7

Requisitos

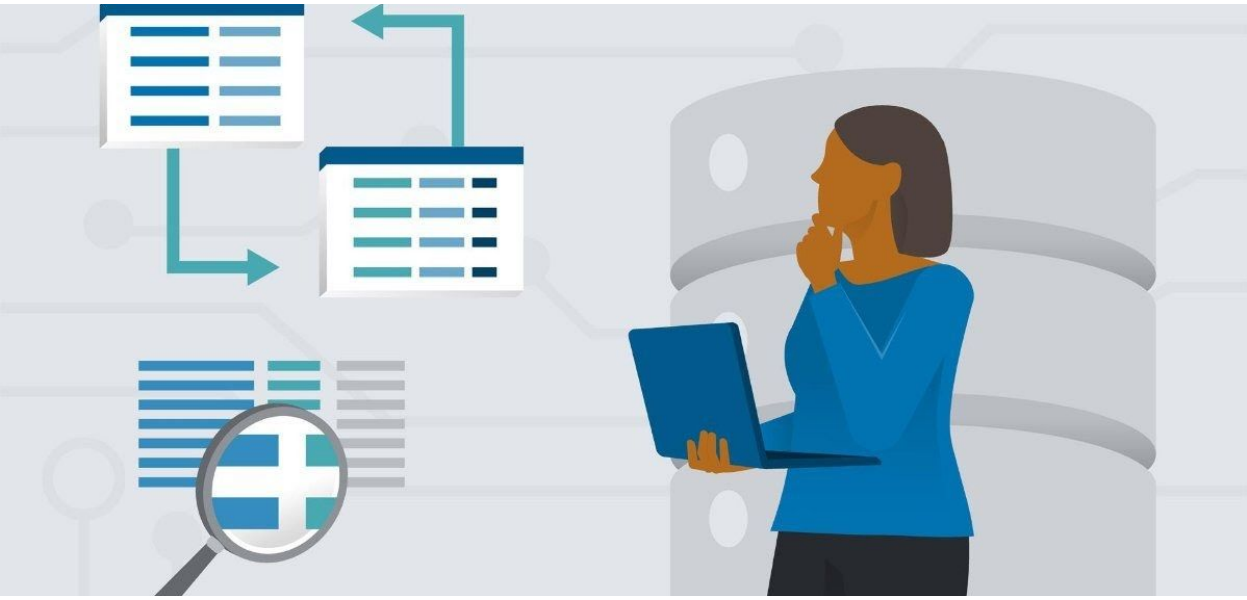
1. **Fuente de datos de usuarios y productos:** Los archivos JSON serán su fuente de inspiración. Recuerden que en el sprint 3 tienen una sugerencia de los campos mínimos.
2. **CDUD de productos y usuarios:** Que hoy funcionan para JSON y sobre los cuales implementarán la magia de Sequelize ✨🧙✨.

Objetivo

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

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**.



Metodología

¡Vamos con la quinta vuelta de retro y planificación 📝🤓👌✨!

Como siempre les recomendamos que no se salteen este paso, es muy importante 😊👆.

La retrospectiva

A esta altura es posible que el equipo haya alcanzado una buena velocidad de trabajo, si es así pueden enfocarse en mantenerla. De lo contrario deberán enfocarse en los puntos que se puedan mejorar.

Implementen nuevamente la dinámica de la **estrella de mar**, resaltando aquello que hay que:

1. Comenzar a hacer
2. Hacer más
3. Continuar haciendo
4. Hacer menos
5. Dejar de hacer

Pueden leer más sobre esta ceremonia aquí.

El tablero de trabajo

Momento de **reiniciar el tablero** para acomodar este sprint de bases de datos. Como siempre lo que haya quedado del anterior sprint se suma al actual.

Recuerden que durante la planificación es importante:

- Debatir cada tarea en conjunto para asegurarse que no haya dudas sobre su alcance (hasta dónde van a hacer) y sobre cómo van a resolverla.
- Estimar la dificultad de la tarea y si ésta requiere de que alguna otra tarea esté terminada antes de poder iniciarla (para determinar el orden).
- Asignar tentativamente las/os responsables de cada una de ellas.

(Opcional) La reunión daily o weekly

La **daily standup** es una reunión, que en los equipos de **Scrum** se realiza todos los días, donde cada integrante habla como máximo 3 minutos de 3 temas puntuales

- Qué hizo ayer
- Si se encontró con algún impedimento
- Qué va a hacer hoy

El formato está pensado para ser rápido y liviano, solo queremos la información más importante de las tareas y los impedimentos.

Importante: No es necesario que esto lo hagan todos los días, al menos una vez por semana sería ideal.

Consignas

Planificación y trabajo en equipo

1. Realizar un breve retrospectiva

   Piensen qué hicieron bien el sprint anterior, qué hicieron mal, qué deberían empezar a hacer, qué deberían dejar de hacer, sigan ésta dinámica.

   Entregable: Actualizar el archivo retro.md con las principales conclusiones de la retro del segundo sprint.

2. Actualizar el tablero de trabajo

   Discutan las tareas que se desprenden de este documento, determinen en qué orden deberán realizarlas, asignen integrantes a cada tarea.

   Entregable: Link al documento o plataforma que utilicen para organizar el trabajo.

3. (Opcional) Implementar daily / weekly standups

   Cada equipo habla como máximo 3 minutos de 3 temas puntuales

   - Qué hizo ayer
   - Si se encontró con algún impedimento
   - Qué va a hacer hoy

   Entregable: Archivo daily.md con sus principales impresiones (positivas, neutras o negativas) sobre la utilidad de esta ceremonia.

Bases de datos y Sequelize

4. Diagrama de base de datos

   Toda buena base de datos empieza en la mesa de dibujo. Les toca 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 implementan Sequelize y que por los general los ORMs como éste 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 ya que es fácil de usar y soporta diagramas DER.

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

5. Script de creación de estructura de base de datos

Tomando como referencia el diagrama del punto anterior, les toca ahora 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.

6. (Opcional) Script de población de base de datos

Ya tenemos la estructura ahora faltan los datos. El script de datos permite que cualquier desarrollador/a (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)
- (Opcional) Poblar la tabla de carrito de compras
- (Opcional) Poblar la tabla de productos de carritos de compras

Una vez definidos los campos de sus tablas, nuevamente pueden utilizar nuevamente Mockaroo, pero esta vez para generar el archivo SQL con datos 😉 👆🏻.

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

7. 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, seguirá crear los archivos de modelos para explicarle 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.

8. ¡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 detalle
- (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.

Resumen de entregables

- ★ Archivo **retro.md** con el resultado de la retrospectiva
- ★ (Opcional) Archivo **daily.md** con sus opiniones sobre las dailies / weeklies
- ★ Tablero de trabajo actualizado
- ★ Diagrama de base de datos
- ★ Script de creación de estructura de base de datos con:
    - Creación de la base de datos y de todas sus tablas
    - Tipos de datos de los campos y sus restricciones
    - Relaciones entre las diferentes tablas
- ★ (Opcional) Script de población de base de datos para:
    - Tabla de usuarios
    - Tabla de productos
    - Tablas secundarias (categorías, marcas, colores, talles, etc)
    - (Opcional) Tabla de carrito de compras y productos de carritos de compras
- ★ Creación de carpeta Sequelize con:
    - Archivos de configuración
    - Modelos con sus relaciones
- ★ CRUD
    - De productos
    - De usuarios
    - (Opcional) De tablas secundarias

Cierre

¡Wow! Las bases de datos son una herramienta que permite que el almacenamiento de tus datos crezca de forma escalable y veloz. Puede ser un trabajo arduo (sobretodo si ya tenías cosas hechas en JSON…) pero el resultado es increíble. No solo es increíble porque tu aplicación está lista para crecer radicalmente sino porque **¡completaste un backend completo!**

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

Si llegaste hasta acá, felicitaciones, sos crack 😎✌✨.