DigitalHouse > Coding School

# Desarrollo Web Full Stack Node

**Trabajo integrador - S8 - M10C24**

# Trabajo Integrador - Sprint 8

## Introducción

Parece mentira que hayamos llegado al final 🤯. Hace algunos meses nomás empezamos con la consola a descubrir de qué iba esto de Node 💻🔍; después vimos HTML, CSS y los motores de plantillas con EJS 📋➡️📄📄📄; también aprendimos cómo podíamos hacer que un usuario se loguee 👥🔑 y permanezca logueado con Middlewares, Sessions y Cookies 🍪😛, luego nos tocó validar tanto en el backend como en el frontend 🔒👮; a lo último vimos que podemos enviar y recibir datos de manera eficiente a través de las APIs 🌐↔️🌐 y que React nos permite trabajar el frontend de una manera modularizada y súper eficiente 🧙✨.

Con estos últimos dos conceptos vamos a trabajar este sprint final. **¡A darle átomos 🚀⚛🚀!**

## Requisitos

1. **Base de datos de usuarios y productos:** en este sprint vamos a trabajar con los datos que ya existen en nuestro sitio, por eso necesitamos la base completa y funcionando.

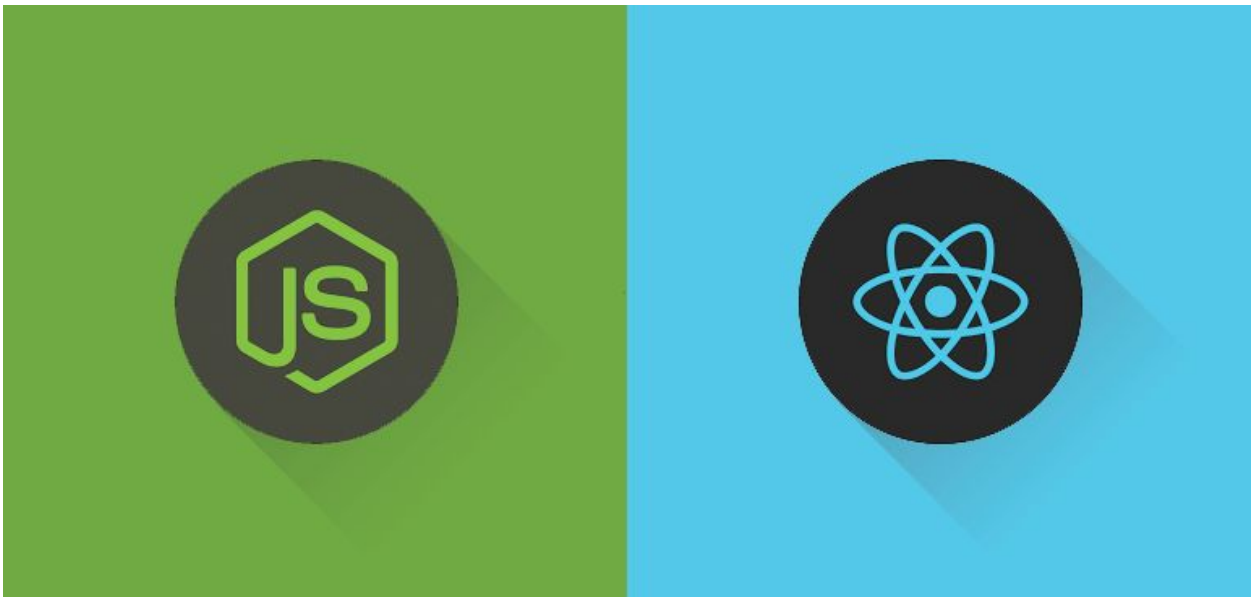
## Objetivo

Al igual que el sprint anterior, éste se divide en dos partes:

En la **primera parte** van a estar armando una **API de usuarios y de productos** que exponga los datos más importantes de su aplicación.

En la **segunda parte** van a estar armando un **dashboard hecho en React** que consuma los datos de la API y muestre de manera resumida las principales métricas de su negocio.

Un dashboard nos permite ver a simple vista si todo está funcionando bien. Pueden pensarlo como el tablero de un auto, donde toda la información que necesitamos está a simple vista.



# Metodologías ágiles

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

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

## La retrospectiva ¡con yapa!

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

Ya estamos casi al final del viaje, y cuando terminen este sprint, les sugerimos que prueben la retrospectiva de **línea de tiempo**.

Que a resumidas cuentas consiste en dibujar una línea de tiempo que incluya todos los sprints y en ella ubicar post-its con cosas positivas (arriba de la línea) y negativas (debajo de la línea) que hayan ocurrido en cada etapa. Al terminar tendrán un resumen gráfico de todo el proceso.

## 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.

**APIs y dashboard**

4. API de usuarios

   Nuestra API de usuarios va a proveernos de dos endpoints muy importantes. El primero nos entregará la lista completa de usuarios y el segundo nos permitirá consultar los detalles de un usuario particular.

   Vamos con esas consignas:

   - **api/users/**
     - Deberá devolver un objeto literal con la siguiente estructura:
       - count → cantidad total de usuarios en la base
       - users → array con la colección de usuarios, cada uno con:
         - id
         - name
         - email
         - detail → URL para obtener el detalle
   - **api/users/:id**
     - Deberá devolver un objeto literal con la siguiente estructura:
       - Una propiedad por cada campo en base
       - Una URL para la imagen de perfil (para mostrar la imagen)
       - Sin información sensible (ej: password y categoría)

**Entregable:** URL funcionales devolviendo datos de usuarios en formato JSON.

5. API de Productos

Nuestra API de productos será muy similar. Sus dos endpoints entregarán la lista completa de productos y el detalle de un producto en particular.

Vamos con esas consignas:

- **api/products/**
  - Deberá devolver un objeto literal con la siguiente estructura:
  - count → cantidad total de productos en la base
  - countByCategory → objeto literal con una propiedad por categoría con el total de productos
  - products → array con la colección de products, cada uno con:
    - id
    - name
    - description
    - un array con principal relación de uno a muchos (ej: categories)
    - detail → URL para obtener el detalle
- **api/products/:id**
  - Deberá devolver un objeto literal con la siguiente estructura:
    - una propiedad por cada campo en base
    - un array por cada relación de uno a muchos (categories, colors, sizes, etc)
    - Una URL para la imagen del producto (para mostrar la imagen)

**Entregable:** URL funcionales devolviendo datos productos en formato JSON.

6. (Opcional) Paginado

   Agregar a los endpoints de listado, la posibilidad de paginar los resultados.

   - **api/users/**
   - **api/products/**
     - 10 resultados por página (recuerden limit y offset 😉👆)
     - next → URL a la próxima página (si corresponde)
     - previous → URL a la página previa (si corresponde)

   Pueden tomar de referencia la API de Star Wars:

   ➔ https://swapi.co/
   ➔ https://swapi.co/api/people/?page=3

7. Dashboard en React

   Ya tenemos nuestra fuente de datos y ahora solo queda consumirlas para darle vida a nuestro dashboard.

   Para este punto te recomendamos que partas de los archivos que te compartimos durante las ejercitaciones presenciales de React.

   El dashboard deberá contener al menos:

   - 3 a 6 paneles simples con totales
     - Total de productos
     - Total de usuarios
     - Total de categorías
   - Panel de detalle de último producto o usuario creado
   - Panel de categorías con el total de productos de cada una
   - Panel con el listado de productos

(Opcional) Agregar datos adicionales

- Total de productos vendidos / total de ventas
- Últimos 5 productos vendidos / los 5 más vendidos

## 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
- ★ Endpoints de **usuarios**:
  - Listado de usuarios
    - (Opcional) Paginado
  - Detalle de usuario
- ★ Endpoints de **productos**:
  - Listado de productos
    - (Opcional) Paginado
  - Detalle de producto
- ★ Dashboard del sitio hecho en React
  - 3 a 6 paneles simples con totales
  - Panel de detalle de último producto o usuario creado
  - Panel de categorías con el total de productos de cada una
  - Panel con el listado de productos
  - (Opcional) Datos adicionales

## No es un adiós, es un hasta pronto

Decíamos en el anterior sprint que *"nada que valga la pena es fácil de conseguir"* y si llegaste hasta aquí no queda más que aplaudirte 👏😊, podés considerarte con orgullo Fullstack Developer 💻🥇🎉.

La programación se parece mucho a la carpintería y eso es porque si bien es posible entender cómo se arma un mueble en la teoría 🧠, no es hasta que lo armamos que realmente empezamos a internalizar cada pequeño detalle que supone construir ese objeto 🤯.

No solo eso, así como la carpintería, la programación es un oficio requiere mucha práctica para llegar a ser realmente hábil. Y en nuestra humilde opinión, hay pocas cosas más satisfactorias que encontrar un oficio al que dedicarse con empeño 👨💻💖👩💻.

Esperamos que le hayan agarrado el gusto al mundo de la programación y que éste sea el comienzo de una gran aventura 🏞️🏕️🚶🚶.

Por aquí seguiremos para cuando necesiten alguna guía 👉🗺️, o por qué no, para recibir su guía. Después de todo ya somos colegas ¿verdad? 😉👌

**¡Hasta pronto y gracias por venir!**