Sprint 3 - Template engine, GET y POST

Desarrollo Web Full Stack Node

Trabajo integrador - S3 - M03C11

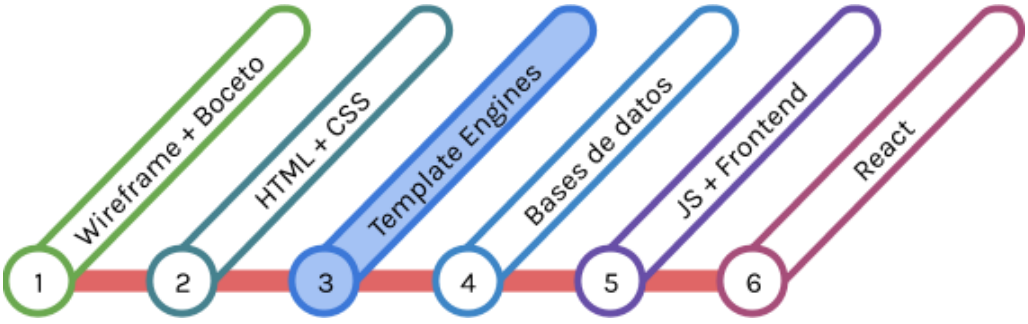
Trabajo Integrador - Sprint 3

Introducción

¡Llegamos a la **tercera iteración** del **Trabajo Integrador**!

En las etapas anteriores se encargaron de pensar, planificar y maquetar las principales páginas de su marketplace. Todo lo que hicieron hasta ahora es completamente estático y hay mucho del código que programaron que se repite innecesariamente.

Es hora de agregar algo de dinamismo a su web y empezar a reutilizar todos aquellos componentes que se comparten como: el header, el footer, la navegación y los productos.



Requisitos

1. **Sitio maquetado:** Necesitan un sitio completo con su HTML y CSS para poder empezar a separar los componentes y aplicar el motor de templates.

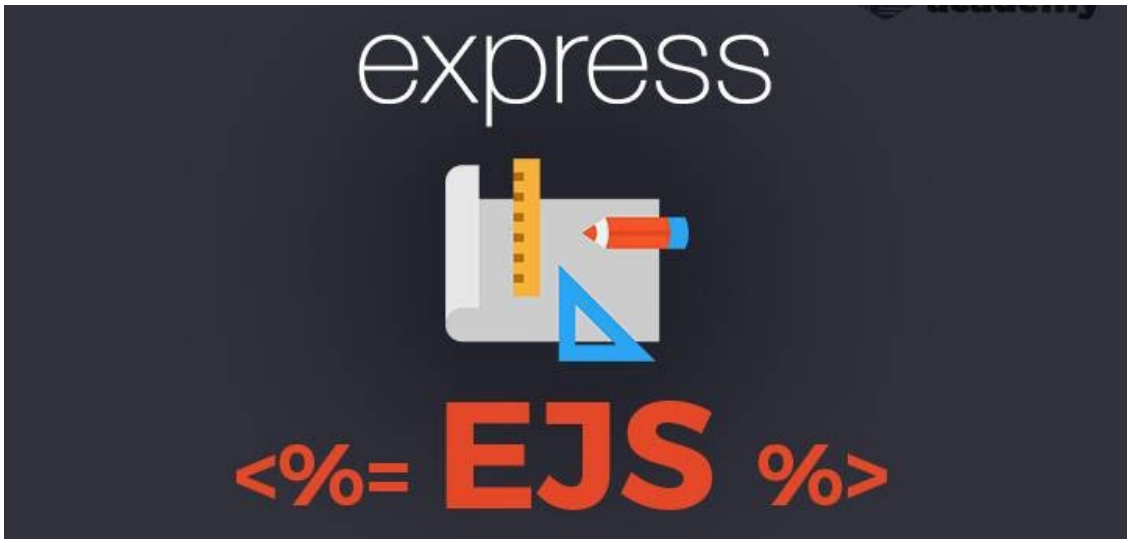
2. **Productos y usuarios definidos:** Indispensable para crear la primera versión de nuestra primera fuente datos de productos y usuarios. Deberán tener una buena idea de los campos que necesitan guardar para cada ítem y para

cada usuario.

Objetivo

Durante esta iteración su foco será el de modificar el sitio para que:

- Reutilice los componentes compartidos: header, menu, footer, etc
- Muestre productos dinámicamente a través de una fuente de datos (JSON) y un motor de templates (Express + EJS)



Metodología

Como ya se habrán dado cuenta, las **metodologías ágiles** son iterativas. Les toca otra vuelta de retro y planificación 📝🔌✨.

La retrospectiva

Ya con dos sprints terminados deberían empezar a marcarse tendencias. Se conocen un poco más entre ustedes, pudieron ver en cuáles aspectos el trabajo fluye y en cuáles no. También debería ser más evidente quién es más hábil para el código, quién para lo estético y quién para lo organizativo.

Usen toda esta experiencia y recuerden que la **retrospectiva** se centra en **mejorar como equipo**, su objetivo es mejorar la dinámica del grupo y no están evaluando el sitio o el producto en sí.

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

Llegó la hora de la **planificación** y les toca **reiniciar el tablero** para acomodar el nuevo sprint. Este proceso es muy importante si quieren tener más chances de llegar bien al final de este tercer tirón.

Si quedaron tareas pendientes del anterior sprint, será su responsabilidad priorizarlas y agregarlas a este sprint.

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

Recuerden que nada está escrito en piedra y que ser ágil se trata de que puedan ser flexibles para llegar a destino de la mejor manera. Si algo ocurre en el camino que altere los planes, lo mejor es saberlo cuanto antes y decidir un nuevo camino de acción.

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 la información más importante de las tareas y los impedimento se transmite.

De esta manera todo el equipo está al tanto de lo que está haciendo cada una/o y en el caso de que haya algún impedimento pueden aportar a la solución.

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

   Nuevamente 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

Tiempo de planificación, discutan las tareas que se desprenden de este documento, determinen en qué orden deberían realizarlas, asignen integrantes a cada tarea (y recuerden que pueden cambiar la organización si es necesario).

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.

Template engine, GET y POST

4. Separar los componentes repetidos en archivos parciales

Crear una carpeta llamada **partials** dentro de la carpeta de **views**, separar las áreas comunes del sitio que deberían incluir al menos:

- Head (desde el principio hasta la apertura del elemento BODY) → head.ejs
- Header (incluyendo barras de navegación) → header.ejs
- Footer (desde el cierre del elemento BODY) → footer.ejs

Pueden separar otros componentes de la misma manera si lo creen útil.

Entregable: carpeta partials dentro de views con todos los archivos parciales.

5. Implementar el motor de templates

Renombrar todas las vistas actuales para implementar la extensión .ejs.

Modificar los métodos de los controladores para que utilicen el método render().

Implementar en todas las páginas que correspondan, los archivos parciales generados en el punto anterior: head.ejs, header.ejs, footer.ejs, etc.

Entregable: Sitio actualizado con todas las vistas y rutas implementando los parciales creados anteriormente y el formato de EJS.

6. Definir los campos necesarios para los productos y generar archivo JSON

Como paso previo a tener una base de datos, vamos a estar trabajando con archivos JSON. Lo que necesitan hacer es decidir los campos que crean necesarios para sus productos. Una buena base para empezar sería:

- Identificador (ya hablaremos más sobre este campo): id

- Nombre del producto: name
- Descripción: description
- Imagen: image
- Categoría: category
- Colores (o cualquier otro campo similar como: talle): colors
- Precio: price

Una vez definidos los campos de sus productos, pueden utilizar la siguiente herramienta para generar el archivo JSON con datos: https://mockaroo.com/.

Aquí tienen un ejemplo ya generado que pueden tomar de referencia: https://mockaroo.com/9511ef20.

Entregable: Carpeta **data** con archivo **products.json** con los datos de productos generados.

7. Definir los campos necesarios para los usuarios y generar archivo JSON

Similar al punto anterior, solamente que esta vez deberán generar datos de usuarios., Una buena base para empezar sería:

- Identificador: id
- Nombre: firstName
- Apellido: lastName
- Email: email
- Contraseña: password
- Categoría: category
- Imagen: image

Utilizar la siguiente herramienta para generar el archivo JSON: https://mockaroo.com/

Entregable: Carpeta **data** con archivo **users.json** con los datos de usuarios generados.

8. CRUD de productos

Como hemos visto la abreviación CRUD (o ABM en español) refiere a las acciones que se desprenden de trabajar con un recurso: listado, altas, bajas, modificaciones.

Su tarea será la de implementar todos los métodos necesarios para poder trabajar con la fuente de datos JSON de productos que crearon en los puntos anteriores.

Recuerden que para cumplir ese objetivo necesitarán de 7 rutas:

1. **/products** (GET)
Listado de productos

2. **/products/:id** (GET)
Detalle de un producto particular

3. **/products/create** (GET)
Formulario de creación de productos

4. **/products** (POST)
Acción de creación (a donde se envía el formulario)

5. **/products/:id/edit** (GET)
Formulario de edición de productos

6. **/products/:id** (PUT)
Acción de edición (a donde se envía el formulario):

7. **/products/:id** (DELETE)
Acción de borrado

Entregable: Sección funcional con listado, detalle, alta, modificación y baja de productos.

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

★ Archivos **products.json** y **users.json** con datos de productos y usuarios

★ Aplicación Node + Express + EJS con:

- ○ Home
- ○ Listado de productos
- ○ Detalle del producto
- ○ Formulario de carga y edición de productos
- ○ Formulario de registro y login

★ Sección funcional con listado, detalle, alta, modificación y baja de productos.

Cierre

De la misma manera que usamos wireframes y bocetos para definir el sitio, corregirlo y determinar si es viable, vamos a estar haciendo lo mismo con los datos y la funcionalidad.

Herramientas como Mokaroo y EJS nos permiten darle vida a nuestro sitio y probar rápidamente como funciona cuando está poblado de productos y de usuarios. En la mayoría de los casos esta práctica nos permitirá detectar errores y puntos de mejora que jamás hubiéramos podido ver en un diseño o una maqueta estática.

Es lo que estamos buscando siempre cuando aplicamos agilidad, encontrar los fallos y puntos de mejora lo antes posible en el proceso de desarrollo. No es casualidad, a medida que avanzamos con el armado de un sitio o un producto, se torna más costoso hacer cambios, correcciones y ajustes.

Pero no sólo lo hacemos por ahorrar tiempo o costos, es un momento mágico cuando empezamos a ver que nuestro sitio cobra vida 🎩✨🐇, y ahora les toca a ustedes 😎👍.