

Programación Web Full Stack

Trabajo integrador - S3 - M03C11

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 sitio. Todo lo que hicieron hasta ahora es completamente estático y sin dudas hay mucho del código 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.



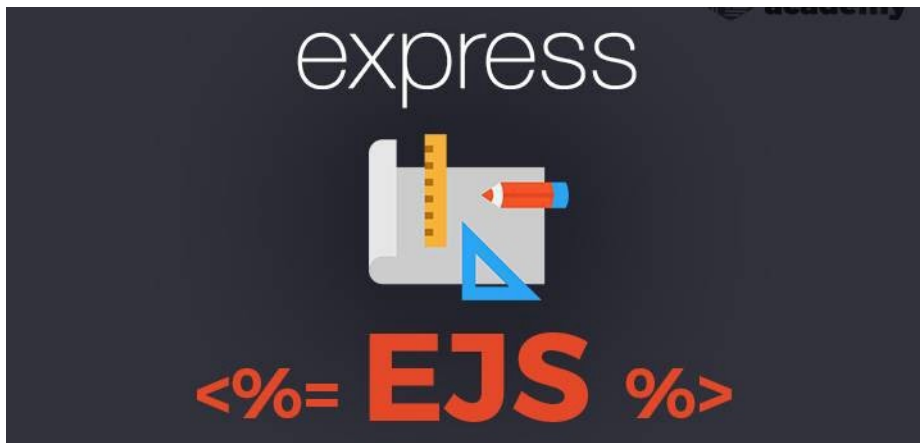
> Requisitos

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.

> Objetivo

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

- Reutilice los componentes compartidos: header, menu, footer, etc.
- Muestre contenido dinámicamente a través de 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, 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 quedan tareas pendientes del anterior sprint, será su responsabilidad priorizarlas y agregarlas a este sprint.

Recuerden que durante la planificación es importante:

- Debater cada tarea en conjunto para asegurarse de 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 esta requiere de que alguna otra tarea esté terminada antes de poder iniciarla (para determinar el orden).
- Asignar tentativamente a los 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 tres minutos de tres 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 se transmite la información más importante de las tareas y los impedimentos.

De esta manera, todo el equipo está al tanto de lo que está haciendo cada uno 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 en el sprint anterior, qué hicieron mal, qué deberían empezar a hacer, qué deberían dejar de hacer, sigan [esta 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 y asignen integrantes a cada tarea (recuerden que pueden cambiar la organización si es necesario).

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

3. Implementar daily/weekly standups (Opcional)

Cada equipo habla como máximo tres minutos de tres temas puntuales:

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

Entregable: Archivo `daily.md` o `weekly.md` con un resumen de las tareas completadas, los impedimentos encontrados y las soluciones propuestas indicando los integrantes.



Template engines

4. Implementar el motor de templates

Implementar el módulo EJS y renombrar todas las vistas actuales para que utilicen la extensión `.ejs`.

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

Entregable: sitio actualizado con todas las vistas y rutas implementando EJS.

5. Separar las vistas en carpetas (Opcional)

Si tenemos en cuenta que nuestro sitio va a crecer y que muy pronto tendremos un montón de páginas, nos conviene mantener el orden desde el principio.

Crear, dentro de la carpeta **views**, la carpeta **products** y la carpeta **users**. Dentro de **products** pondremos todas las vistas de productos que tengamos (por ejemplo: listado, detalle, creación, edición, etc.). Dentro de **users** pondremos todas las vistas de usuarios que tengamos (por ejemplo: registro, login, perfil, etc.).

- Usuarios: `src/views/users/`
- Productos: `src/views/products/`

Entregable: estructura actualizada de directorios y archivos de las vistas.

6. Separar los componentes repetidos en archivos parciales

Crear una carpeta llamada **partials** dentro de la carpeta de **views**, separar las áreas comunes del sitio. Como mínimo nos gustaría ver:

- Head (incluyendo todo el elemento `<head></head>`) → `head.ejs`
- Header (incluyendo barras de navegación) → `header.ejs`
- Footer (incluyendo todo el elemento `<footer></footer>`) → `footer.ejs`
- (Opcional) Otras secciones de su sitio que se repitan.

Pueden separar otros componentes de la misma manera si lo creen útil. Por ejemplo: los



productos dentro de un listado.

Recuerden implementar los archivos parciales en todas las páginas que correspondan.

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

Entregable: sitio actualizado con la implementación de los **partials**.

7. Página: creación y edición de productos

Formulario al que accede el usuario administrador para cargar nuevos productos y editar los existentes.

Un buen punto de partida para los campos de estos formularios puede ser el siguiente:

- Nombre del producto (*name*)

- Descripción (*description*)
- Imagen (*image*)
- Categoría (*category*)
- Colores (o cualquier otro campo similar como: tamaños, talles, etc)
- Precio (*price*)

Historias de usuario

- ★ **Como administrador *quiero* poder crear nuevos productos *para* agregarlos a los listados del sitio.**
- ★ **Como administrador *quiero* poder modificar los productos existentes *para* corregir información o actualizar precios.**

Entregable: páginas de creación y edición de productos en formato **.ejs** junto con todos los recursos necesarios, estilos de CSS e imágenes.

> Resumen de entregables

- ★ Archivo **retro.md** con el resultado de la retrospectiva.
- ★ Archivo **daily.md** con sus opiniones sobre las daylies/weeklies. (Opcional)
- ★ Tablero de trabajo actualizado.
- ★ Aplicación Node.js+Express+EJS con:
 - Archivos parciales (head, header, footer, etc.)
 - Home
 - Listado de productos
 - Detalle del producto
 - Carrito de productos
 - Formulario de registro y login
 - Formulario de carga y edición de productos

> Cierre

Una de las maravillas de la programación es que nos permite automatizar los procesos que se repiten. 📄 Ya vimos que JavaScript (así como la mayoría de los lenguajes de programación) nos permite escribir funciones para agrupar código que necesitamos reutilizar en el futuro.

Herramientas como [EJS](#) nos ayudan a hacer lo mismo con nuestras páginas, agrupando partes que se repiten en muchos lugares y generando componentes que podemos reutilizar en todos los lugares donde sean necesarios.

Otra de las ventajas de **EJS**, o de cualquier otro motor de plantillas, es que, si en algún momento tenemos que actualizar uno de esos componentes, basta con hacerlo en un solo lugar en vez de tener que ir a revisar cada página. 🧩🔗



¡Así que buena suerte con ese rompecabezas de componentes! 🧩🧩🧩

