

Desarrollo Web Full Stack Node

Trabajo integrador - S2 - M05C08

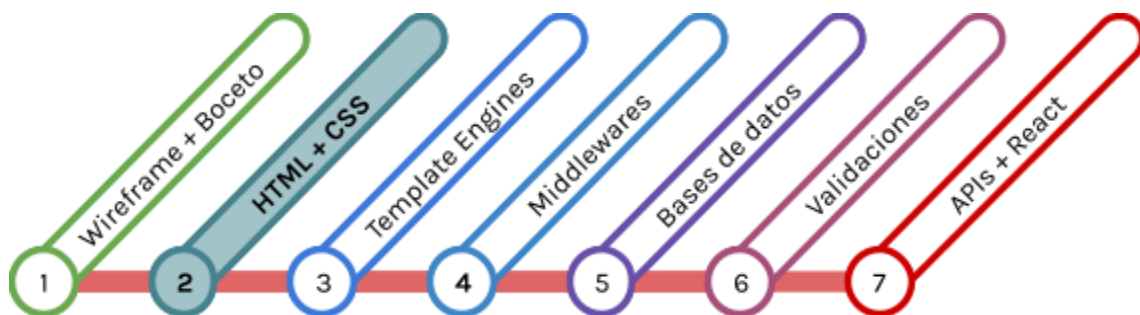
Trabajo Integrador - Sprint 2

Introducción

Bienvenidas y bienvenidos a esta **segunda iteración** del **Trabajo Integrador**.

En la etapa anterior se encargaron de planificar un poco, definiendo la temática del sitio, armando los bocetos y creando el diseño que servirá de guía para las posteriores etapas.

En esta etapa es momento de arrancar a codear, van a estar escribiendo HTML y CSS. Con estas dos herramientas van a poder darle “vida” a los bocetos que prepararon.



Requisitos

1. **Repositorio creado:** Indispensable para trabajar en equipo y realizar la entrega de cada sprint. Recuerden que todos los integrantes deben contribuir con código.
2. **Sitio definido y bocetos terminados:** La temática del sitio y los bocetos deberán estar definidos. Sin ellos lo más probable es que tengan problemas para avanzar con rapidez y para trabajar por separado de manera eficiente. Los bocetos también sirven de guía a la hora de resolver dudas o conflictos.

Objetivo

Durante esta iteración su foco deberá ser el construir un sitio que:

- Sea estéticamente agradable (UI - User Interface)
- Sea fácil de usar (UX - User Experience)



Metodología

Vamos a seguir practicando las **metodologías ágiles**.

La retrospectiva

A esta altura ya tienen un sprint terminado (o deberían tenerlo), eso quiere decir que tienen evidencia de cómo funcionó el grupo durante estas semanas y es un muy buen momento para hacer una retro.

La **retrospectiva** se centra en **mejorar como equipo**. Les invitamos a que implementen la dinámica de la **estrella de mar**, que busca resaltar 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

Es una herramienta indispensable para realizar el seguimiento del proyecto, con él pueden:

- Organizar y dar seguimiento a todas las tareas
- Saber quién está haciendo cada tarea en cada momento
- Saber qué tareas están pendientes para comenzarlas
- Saber qué criterios debe cumplir una tarea para considerarse lista
- Saber si el sprint viene a buen ritmo o si está atrasado

Consignas

¡Importante! Durante este sprint estaremos trabajando solamente la estructura y estética del sitio, lo que comúnmente se conoce como la maqueta del sitio.

No será necesario implementar ninguna funcionalidad más que los enlaces entre las diferentes páginas. Para consignas como la validación de formularios deberán mostrar cómo se verán los errores aún si el formulario no esté funcional.

En todas las consignas que corresponda, se les presentarán las historias de usuario (user stories). Será su responsabilidad determinar las tareas que se desprenden de ellas.

1. Realizar un breve retrospectiva

Nuestro equipo es ágil y uno de los secretos de la agilidad es el de fallar rápido para ser exitosos más pronto. Piensen qué hicieron bien el sprint anterior, qué hicieron mal, qué deberían empezar a hacer, qué deberían dejar de hacer, les recomendamos [ésta dinámica](#).

Entregable: Crear un archivo retro.md en el repositorio y anotar las principales conclusiones de la retro del primer sprint.

2. Crear un tablero de trabajo

Indispensable para trabajar en equipo. Les dejamos dos recomendaciones, aunque son libres de utilizar cualquier herramienta o método que deseen.

- [Trello](#), una herramienta muy flexible que permite para dar seguimiento a todo tipo de proyectos o procesos. Les dejamos un tablero de muestra:

<https://trello.com/b/mDz4bKmd/proyecto-integrador>

- [GitHub](#), otra manera de organizar el trabajo a través tableros. Les dejamos el artículo de GitHub que explica su funcionamiento:

<https://help.github.com/es/github/managing-your-work-on-github/about-project-boards>

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

3. Crear la estructura de archivos utilizando Node + Express

Misma estructura que vimos en la clase de Express. A elección si desean crearla manualmente o tomar el boilerplate disponible en:

<https://github.com/javi-dh/baseAppHtml>

Entregable: Implementación de express-generator con estructura MVC.

4. Página: home del sitio

Home del sitio con la oferta de productos y / o servicios.

Historias de usuario

- ★ *Como cliente quiero ver un resumen de la oferta de productos para poder decidir rápidamente si quedarme o buscar el producto en otro sitio.*
- ★ *Como cliente quiero que el sitio funcione bien en mi teléfono celular para poder acceder en cualquier momento.*

Entregable: Archivo index.html junto con todos los recursos necesarios, estilos de css e imágenes.

5. Página: detalle de producto

Página a la que accede el cliente al hacer click en un producto.

Historias de usuario

- ★ *Como cliente quiero poder ver toda la información relevante del producto para decidir si voy a comprarlo.*

- ★ *Cómo cliente quiero poder agregar fácilmente el producto a mi carrito para poder comprarlo.*

Entregable: Archivo productDetail.html (o similar) junto con todos los recursos necesarios, estilos de css e imágenes.

6. Página: carrito de compras

Página donde el cliente puede visualizar su compra actual y proceder al pago.

Historias de usuario

- ★ *Como cliente quiero poder acceder al detalle de mi carrito de compras en cualquier momento para finalizar la compra.*
- ★ *Como cliente quiero poder modificar productos del carrito para poder eliminarlos de la compra o modificar la cantidad.*

Entregable: Archivo productCart.html (o similar) junto con todos los recursos necesarios, estilos de css e imágenes.

7. Página: registro

Historias de usuario

- ★ *Como cliente quiero registrarme para acceder a mi historial de compras.*
- ★ *Como cliente quiero registrarme para facilitar futuras compras.*

Entregable: Archivo register.html (o similar) junto con todos los recursos necesarios, estilos de css e imágenes.

8. Página: formulario de carga de producto

Página del backend de administración donde el personal del sitio carga nuevos productos y gestiona los existentes.

Historias de usuario

- ★ *Como administrador quiero poder cargar productos con facilidad para hacer eficiente mi trabajo.*
- ★ *Como administrador quiero saber si un campo es obligatorio para acelerar la carga.*
- ★ *Cómo administrador quiero que el formulario me marque los errores para evitar cargar productos incompletos o erróneos.*

Entregable: Archivo productAdd.html (o similar) junto con todos los recursos necesarios, estilos de css e imágenes.

Resumen de entregables

- ★ Archivo **retro.md** con el resultado de la retrospectiva
- ★ Enlace al tablero de trabajo
- ★ Aplicación Node + Express con:
 - Home (index.html)
 - Detalle del producto (productDetail.html)
 - Carrito de compras (productCart.html)
 - Formulario de registro (register.html)
 - Formulario de carga de producto (productAdd.html)

Cierre

El desarrollo web moderno pone en el centro al usuario, eso quiere decir que vamos a **pensar primero en la necesidad o el problema** que tiene nuestro usuario **antes de programar la solución**.

En el caso del cliente la necesidad podría ser la de comprar un artículo en el sitio y en el caso del administrador podría ser la de gestionar los productos y las ventas. Queda entonces de su

lado, responsabilidad de asegurarse que los usuarios tengan lo necesario para poder cumplir tus tareas con facilidad.

Las metodologías ágiles son iterativas y toman como fuente de mejora las iteraciones pasadas, no se preocupen si las cosas no salen como esperaban en un comienzo, tomen lo bueno, lo malo y úsenlo para ir mejorando como equipo en cada entrega.

Cuidadito con este sprint, si bien las consignas son simples, van a estar escribiendo código en forma colaborativa por primera vez, lleva tiempo organizarse y muy posiblemente haya conflictos en el camino. **Les recomendamos comenzar lo antes posible** y recuerden arrancar con GIT desde el principio para agarrar velocidad con la práctica.

¡Les toca brillar a ustedes! ¡Buena suerte!