



## Programación Web Full Stack

Trabajo integrador - S4 - M03C11

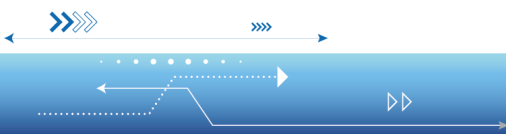
# Sprint 4

### > Introducción

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

Si completaron los tres sprints anteriores (los completaron, ¿verdad 😄?), deberían tener resuelta la parte visual de su sitio. Ya lo podemos navegar, pero todavía no lo podemos usar...

¡Muy bien! Nos toca ahora lograr que el sitio comience a cobrar vida haciendo que todos esos formularios y acciones trabajen con productos y usuarios reales almacenados en formato JSON.



## > Requisitos

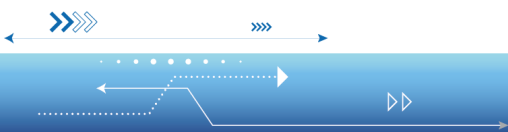
1. **Productos y usuarios definidos:** Indispensable para crear la primera versión de nuestra fuente de datos de productos y usuarios. Deberán tener una buena idea de los campos que necesitan guardar para cada ítem y para cada usuario.

## > Objetivo

Durante esta iteración su foco será el de modificar el sitio para que muestre productos dinámicamente a través de una fuente de datos (JSON).

{ JSON }

JavaScript Object Notation



## > Metodología

¡Vamos con otra vuelta de retro y planificación! 📝🤖💡✨

No se salteen este paso, es más importante de lo que piensan. 😊👉

### La retrospectiva

Para esta altura ya deberían ser expertos en encontrar puntos de mejora.

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

Otra vez toca **reiniciar el tablero** para acomodar este sprint.

Las tareas pendientes del anterior sprint deben 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 los 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

Nuevamente piensen qué hicieron bien 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 tercer 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.

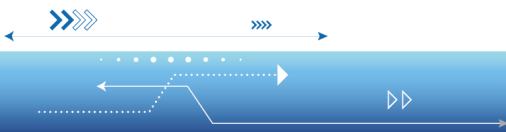
**Entregable:** link al documento o plataforma que utilicen para organizar el trabajo.

#### 3. Implementar daily/weekly standups (Opcional)

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



## JSON y métodos de HTTP

### 1. Definir los campos necesarios para los productos y generar archivo JSON

Como paso previo a tener una base de datos relacional, 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: talla): 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.

### 2. Definir los campos necesarios para los usuarios y generar archivo JSON

Similar al punto anterior, pero 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



Aquí tienen un ejemplo ya generado que pueden tomar de referencia: <https://mockaroo.com/0b5f3440>

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

### 3. CRUD de productos

CRUD es el acrónimo (en inglés) de "Crear, Leer, Actualizar y Borrar", es decir, poder crear, leer, actualizar y borrar un producto en particular.

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 siete rutas:

1. **/products** (GET)  
Listado de productos
2. **/products/create** (GET)  
Formulario de creación de productos
3. **/products/:id** (GET)  
Detalle de un producto particular
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.
- ★ Archivo **daily.md** con sus opiniones sobre las daylies/weeklies. (Opcional)
- ★ Tablero de trabajo actualizado.
- ★ Archivos **products.json** y **users.json** con datos de productos y usuarios.
- ★ Administración completa de productos con:
  - Listado
  - Detalle
  - Creación
  - Edición
  - Eliminación

## > 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](#) nos permiten darle vida a nuestro sitio y probar rápidamente cómo funciona cuando está poblado de productos y de usuarios (en nuestra jerga se denomina Mockear nuestros datos). 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 siempre estamos buscando 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 solo lo hacemos para ahorrar tiempo o costos, es un momento mágico cuando empezamos a ver que nuestro sitio cobra vida 🎩✨🐰, y ahora les toca a ustedes. 😎👍

