



## Programación Web Full Stack

Trabajo integrador - S5 - M03C14

# Sprint 5

### > Introducción

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

A esta altura deberían tener un **sitio dinámico utilizando EJS** como motor de plantillas, un **CRUD de productos** funcionando perfectamente con una fuente de datos **basada en JSON**. Es un buen momento para mirar atrás 🕒 y darse cuenta de cuánto avanzaron en tan poco tiempo 🙌😎🙌.

Nos queda entonces enfocarnos en los usuarios. En esta etapa vamos a estar generando el registro, el login, el perfil y, además, vamos a trabajar en las rutas a las cuales podrán acceder nuestros huéspedes (los visitantes que no hicieron login) y otras para los usuarios con login.



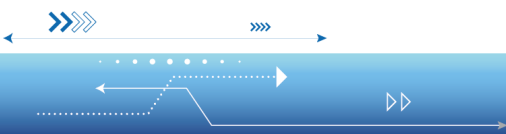
## > Requisitos

1. **Fuente de datos de usuarios:** indispensable para darle vida a los formularios de registro, login y a la página de perfil. Deberán tener una buena idea de los campos que necesitan guardar para cada usuario. **Recuerden que en el sprint anterior tienen una sugerencia de los campos mínimos que deberían tener.**

## > Objetivo

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

- Permita el flujo de registro, login y logout de usuarios.
- Permita recordar al usuario para que pueda ingresar sin volverse a loguear.
- Tenga rutas accesibles solo por huéspedes (visitantes sin login).
- Tenga rutas accesible solo por usuarios (que hicieron login).



## > 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 ser priorizadas y agregadas a este sprint.

Recuerden que durante la planificación es importante:

- Debatir 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 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 cuarto sprint.

---

#### 2. Actualizar el tablero de trabajo

Discutan las tareas que se desprenden de este documento, determinen en qué orden deberán ser realizadas, 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.

---

## Usuarios y middlewares

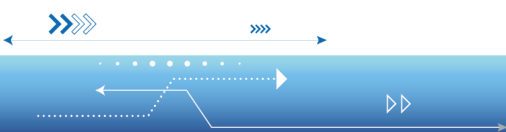
### 4. Implementar la entidad de usuarios

Tomando como ejemplo lo que hicieron para productos, replicar la estructura de archivos y directorios que necesitarán para implementar las funcionalidades.

Les sugerimos esta estructura, aunque si trabajaron diferente, vale igual. 😊

- Rutas: `src/routes/users.js`
- Controlador: `src/controllers/usersController.js`
- Vistas: `src/views/users/`
- Directorio para imágenes: `public/images/users/`
- Colección: `src/data/users.json`

**Entregable:** estructura de archivos y directorios de usuarios.



## 5. Implementar el registro de usuarios

Tomando como referencia el formulario de creación de productos, implementar el formulario de registro de usuarios.



- Deberá incluir los campos mínimos mencionados en el sprint anterior.
- Deberá permitir la subida de una imagen de perfil (con [Multer](#)).
- Deberá encriptar la contraseña ingresada por el usuario (con [bcrypt.js](#)).
- Deberá guardar los datos enviados en el archivo JSON de usuarios.

**Entregable:** formulario funcional de creación de usuarios.

---

## 6. Implementar el login de usuarios

Ahora, es momento de poner en práctica **middlewares**, **sesiones** y **cookies**. Deberán implementar un formulario de login que:

- Incluya los campos de email y password.
- Verifique la información enviada por el usuario y según el caso:
  - Redirija a la home o a la **página de perfil** en caso de éxito y muestre los datos del usuario en algún lugar del sitio, como el header.
  - Redirija nuevamente al login en caso de error.



**Entregable:** formulario funcional de login.

---

## 7. (Opcional) Implementar la función de recordar al usuario

Deberán agregarle al formulario de login la posibilidad de que se recuerde al usuario (checkbox). En caso de que el usuario decida ser recordado:

- Utilizar cookies para guardar esa información en el navegador.
- Implementar un middleware de aplicación que busque la cookie y loguee al usuario en caso de que exista y sus datos sean correctos.



## 8. Implementar rutas de huéspedes y de usuarios

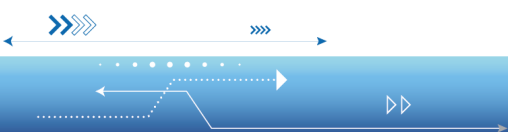
Ahora que tienen un login funcionando, su próximo desafío será el de separar las rutas que se pueden acceder en cualquier momento, de las que se puede acceder solo si uno no está logueado y, por último, de las que requieren estar logueado.

En cada caso deberán implementar el comportamiento que corresponda:

- Rutas accesibles por cualquiera → sin cambios
- Rutas accesibles solo **sin** login → redirigen al perfil
- Rutas accesibles solo **con** login → redireccionan al login

### > 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.
- Formulario de registro con:
  - Los campos mínimos mencionados en el sprint anterior.
  - Subida de una imagen de perfil.
  - Guardado en JSON con encriptación de contraseña.
- Formulario de login con:
  - Campos de email y password.
  - (Opcional) Función de recordar al usuario.
- Rutas de huéspedes y usuarios:
  - Las de huéspedes deberán redireccionar al perfil si el usuario está logueado.
  - Las de usuarios deberán redireccionar al login si el usuario no está logueado.



## > Cierre

Ya habíamos dicho que en programación siempre buscamos no repetirnos, así que cuando una tarea la estamos haciendo muchas veces en diferentes partes de nuestro código, la abstraemos. Es decir, que aislamos ese comportamiento en un bloque de código reutilizable.

Los middlewares nos permiten justamente eso. Un middleware de aplicación se ejecutará en cada pedido del cliente, mientras que un middleware de ruta se ejecutará cada vez que se pida esa ruta particular.

Con estas dos herramientas se pueden hacer cosas muy poderosas. De hecho, cada uno de los paquetes que han instalado (Express, EJS, method-override, Multer, etc.) están contruidos como middlewares. Ahora ya pueden construir los suyos. 🤖👉🌟

Por otro lado, las sesiones y las cookies son parte del funcionamiento normal de la mayoría de los sitios que visitan a diario. Es otra parte del desarrollo web que van a dominar si siguen poniéndole horas de práctica a todo esto.

El mundo digital que los rodea está escrito utilizando este tipo de tecnologías. A medida que avancen en su aprendizaje les va a ser más fácil entender cómo funciona todo este ecosistema de webs y aplicaciones.

