

## Sprint 2

En lineas generales, les pedimos tres grandes cosas:

- Retrospectiva del sprint 1
- Creacion del proyecto de express
- Creacion de las maquetas del sitio usando HTML+CSS

### La retrospectiva

Para la retro vimos que tenemos que usar la metodologia de “Starfish Retrospective” (Estrella de mar).

En este link pueden encontrar más informacion sobre la retrospectiva. Tambien en las consignas del sprint van a encontrar más informacion.

Lo que yo recomiendo para hacerlo de manera remota es utilizar un tablero de trello (nuevo, diferente al del proyecto) que vamos a reutilizar para todas nuestras retrospectivas, pero pueden hacerlo como prefieran.

El proceso de la retro en Trello deberia ser una cosa asi:

- Creamos las 5 secciones de la retrospectiva como listas de tarjetas. (“Start doing”, “More of”, “keep doing”, “Less of”, “Stop doing”)
- Empezamos por “Start Doing”.
  - Cada uno empieza a agregar tarjetas (como **minimo** una tarejta por miembro del equipo)
  - Una vez que nadie quiere agregar más, pasamos al siguiente, “More of”
  - Repetimos hasta que hayamos completado con una tarjeta por miembro del equipo en todas las listas.
- Empezamos a charlar sobre cada una de las tarjetas, otra vez en el orden de arriba.
- Vamos anotando en la descripcion de cada tarjeta lo que charlamos sobre esto y [MUY IMPORTANTE] los pasos que tenemos que hacer para lograr ese objetivo.
- Generamos el documento de markdown que nos pide el sprint. Eso seria escribir cada una de las secciones de la retro, con sus tarjetas y sus descripciones. En este archivo anoten la fecha de esta retro, porque para el proximo sprint vamos a crear una nueva más arriba

### Ejemplo de Retro.md

```
# Retrospectivas
```

```
## Sprint 1 - 28/4/2020
```

```
### Start Doing
```

- **\*\*Reuniones semanales\*\***  
 Los Sprints requieren cada vez más cohesión de equipo. Sería ideal que podamos reunirnos dos veces por semana para mantenernos en sincronía. Vamos a utilizar Google Calendar para generar eventos recurrentes todas las semanas, en principio los `_lunes_` y los `_jueves_` con todos los miembros del equipo, y de esta manera definir horarios y días fijos de reunión.
- **\*\*Repartición de tareas\*\***  
 Ahora mismo, es difícil de ver quién está haciendo que tarea en el Trello. En el nuevo sprint, al momento de crear las tareas, vamos a estar en una reunión por hangouts y en este momento vamos a definir quién se va a ocupar de cada tarea, al menos inicialmente, agregando como miembro de la tarjeta al miembro del equipo que corresponda.
- **\*\*Visibilidad de prioridades\*\***  
 Es difícil ver qué tareas son más importantes para el sprint. Vamos a crear las etiquetas: Prioridad Alta (Rojo), Prioridad Media (Amarillo), Prioridad Baja (Verde) para definir la prioridad de todas las tareas del sprint.

### More of

(...)

## Estructura del proyecto.

Podemos utilizar `express-generator` para generar la estructura del proyecto o hacerlo a mano.

### Usando `express-generator` (RECOMENDADO)

Para eso podemos utilizar el siguiente comando:

```
npx express-generator --ejs --force <carpeta-del-repo>
```

- `--ejs`: Esto define que el template engine que vamos a usar es `ejs`. Todavía no lo vamos a utilizar, pero es importante definirlo correctamente.
- `--force`: Esto es para poder usar una carpeta que no esté vacía. Como objetivo estamos usando la carpeta del repositorio que ya tiene las cosas que entregamos en el primer sprint así que va a ser necesario.

Lo correcto es que nuestro proyecto quede en la raíz del proyecto, por eso es que tendríamos que hacerlo así.

## Creando el proyecto a mano

Por otro lado, si lo hacemos a mano, deberíamos preparar el proyecto de npm, crear la estructura minima requerida para un proyecto de express, etc.

Para eso podemos utilizar los siguientes comandos:

```
cd <carpeta-del-repo>
```

```
npm init -y
npm install express
```

```
mkdir -p public/img
mkdir -p public/src
mkdir src
```

```
touch src/app.js
```

## Estructura minima

Asi debería quedarles el proyecto como minimo. Más abajo detallo las cosas más importantes.

```
/
|- README.md
|- package.json
|- mockups (1)
|- public (2)
|   |- css
|   |   |- master.css
|   |   |- home.css
|   |   |- productDetail.css
|   |- img
|   |- index.html
|   |- productDetail.html
|   |- productCart.html
|   |- register.html
|   |- productAdd.html
|- src (3)
|   |- app.js
```

Detalles Importantes:

1. (mockups) En esta carpeta van a guardar los archivos y materiales que entregaron en el sprint anterior. Es decir los bocetos, las imagenes, referencias, etc. La pueden llamar como mejor les parezca, pero que sea claro qué es lo que tiene.
2. (public) En esta carpeta van a crear todos los htmls y van a tener los css y las imagenes que necesiten (en las carpetas css y img). Se tiene que llamar

`public.`

3. (`src`) En esta carpeta vamos a tener el código fuente de `expressjs`. Por ahora no es necesario que tenga nada más que un archivo `app.js` (o `index.js`). Se tiene que llamar `src`.