

# SPRINT 10. COMENTARIOS, REESTRUCTURA, MODIFICACIONES, VISTA LOCAL Y FUNCIONALIDAD REAL

---

## COMENTARIOS

En primer lugar, he comenzado haciendo los comentarios con lo que se espera que haga cada función siguiendo la nomenclatura de **Javadoc**. Esto ha transcurrido sin ningún problema.

## REESTRUCTURA

Tras comentar las funciones, he comenzado a reestructurar los archivos siguiendo los nuevos nombres que se habían comentado en clase:

- **v**NOMBRE para las vistas.
- **m**NOMBRE para los modelos.
- **p**NOMBRE para los presentadores.

Tras realizar estos cambios en el sketch, cambiar los nombres de los archivos .cpp y .h pertinentes y cambiar los includes del código, he intentado compilar el programa para comprobar que todo seguía funcionando correctamente pero no. El IDE de Arduino seguía llamando a los archivos por su nombre antiguo (pese que ya los hubiera cambiado) y, por tanto, daba error. Para arreglarlo he debido de borrar el caché del IDE accediendo a `C:\Users\nombre_usuario\AppData\Local\arduino\sketches` y borrando las carpetas que se encontraban en su interior. Tras esto, el código funcionó correctamente.

## MODIFICACIONES

En cuanto a las modificaciones, en primer lugar he creado `vSetup` y `mSetup`, encargados de configurar el usuario y contraseña de la WiFi del domicilio en el que se vaya a instalar la placa, apoyandome de Gemini para conseguir que funcionaran correctamente. Además he tenido que modificar `AppNetwork.cpp` para que se conectara a la WiFi utilizando las credenciales proporcionadas por el usuario y `pBase.cpp` (el presentador) para crear una ruta en la que la nueva vista sea accesible. Finalmente he desarrollado la lógica para tener esa persistencia, guardando dichas credenciales en la memoria flash del ESP. Adicional a esto, debido a la innecesaria creación de un modo AP, he optado por eliminar esta funcionalidad del código, ya que el Arduino se conectará a la WiFi de casa y nosotros con nuestro telefono también pero, en ningún momento debemos estar conectados como tal al ESP. Se ha comentado todo siguiendo la nomenclatura de Javadoc.

## VISTA LOCAL

Ahora trataremos la parte local. En primer lugar, he creado los archivos `vLocalSetup` (.cpp y .h) los cuales serán los encargados de modificar las credenciales de WiFi. Este archivo es muy importante porque sin el no podríamos hacer la primera conexión al router y, por tanto, la aplicación no podría funcionar correctamente porque no podríamos mostrar el contenido en la red (dado que no nos sabemos las credenciales de estas). Por otra parte, he creado una vista adicional `vLocalPrincipal` (tanto el .cpp como el .h) la cual se encarga de mostrar la información de la base de datos (los productos que tenemos, botones para aumentar y disminuir stock, eliminarlos, etc). No he podido hacer esto sin modificar el código de la librería `TFT_eSPI.h` por el

siguiente motivo: La librería se compila ciegamente, sin ver nuestra aplicación. Si no encuentra la funcionalidad *TOUCH\_CS*, piensa que el usuario no tiene pantalla táctil y, por tanto, obvia esa funcionalidad. Como borra esa funcionalidad, cuando accedo a dicha propiedad desde mi código da error porque realmente esa funcionalidad no existe. Esta funcionalidad existe en la librería, simplemente debemos descomentarla para que funcione.

## FUNCIONALIDAD REAL, UTILIZANDO LA PLACA

Llego el momento de probar la aplicación en la pantalla. Lo primero, el COM adecuado y los baudios a 115200 en el serial monitor. En primer lugar he tenido un problema. Si no hay WiFi guardada porque es la primera vez que lo enciendes y tampoco hay AP porque vimos que no era necesario, el ESP se queda en tierra de nadie y se cuelga. La solución ha sido sencilla. Si no tenemos conexión, no necesitamos el servidor web sino solamente la vista *vLocalSetup* para configurar la conexión y una vez tengamos las credenciales, lanzar el servidor (el presentador). Tras esto, he conseguido el mensaje de *SISTEMA LISTO* que tanto esperaba. Aún así, no se mostraba nada por pantalla. Esto fue por lo que me paso hace varios SPRINTs. No le había puesto brillo a la pantalla. Tras arreglar esto, ya mostraba el mensaje "SIN CONEXIÓN, TOCA PARA CONFIGURAR". De todas formas, el táctil no funciona, a sí que debo hacer códigos de prueba para comprobar que no sea problema de la calibración. Finalmente, he tenido que instalar una nueva librería llamada **XPT2046\_Touchscreen** porque en esta placa el táctil y la pantalla están conectados a pines diferentes y, por tanto, `touchScreen()` nunca funcionará. Para ello, he cambiado todo lo relacionado a la parte táctil de la librería *TFT\_eSPI.h* por la nueva, creando los archivos *TouchDrv* (tanto *.cpp* como *.h*) y he conseguido así que el táctil funcione. Ahora cuando inicio el ESP, toco la pantalla y tengo acceso a un teclado (lo más grande posible, la pantalla es la que es y hay que escribir no puedo hacer otra cosa) para poner las credenciales de la WiFi.

## MOCKUPS

Para el diseño de los mockups, he hecho una vista html + css donde se podrá ver de forma totalmente fidedigna lo que se va a mostrar tanto en la pantalla local como en el navegador.