# UD_Logo_Bilíngüe_Cas-Eus

**Ingeniería Web – Proyecto Web Colaborativo**

**Titulo:** Reto 4

**Curso:** 2º Grado en Industria Digital (Semestre 2º)

**Materia:** Ingeniería Web

**Estudiantes:** Javier Marín Díaz de Guereñu

Rubén Domínguez Tilve

Iker Reyes Navarro

**Grupo:** IW-09

**Profesor:** Jon Vadillo Romero

**Facultad de Ingeniería**

**Universidad de Deusto**

**Vitoria - Gasteiz, mayo de 2020**

# RESUMEN

Hemos desarrollado una aplicación que permite gestionar diferentes facetas de una empresa. El núcleo de la aplicación está basado en el catálogo, los pedidos y las tareas. La idea de nuestro sistema se basa en que los clientes puedan hacer pedidos, un responsable los desglose en tareas y un operario las marque como realizadas una vez terminadas. Los clientes pueden tener un seguimiento de su pedido, viendo el progreso del mismo.

Hemos contemplado varios tipos de usuarios para nuestra aplicación:

* Operario: podrá gestionar sus tareas asignadas, podrá consultar los equipos, los procesos y los perfiles de sus compañeros.
* Responsable: podrá gestionar todo lo anteriormente mencionado, además del catálogo y de los pedidos. Este además, es el encargado de desglosar los pedidos en tareas. Se entiende al responsable como el gestor principal de la aplicación.
* Cliente: su único cometido es hacer pedidos, con la capacidad de poder gestionarlos. Otra de las funcionalidades disponibles del clientees registrarse a sí mismo.
* Superadmin: este usará el apartado de Djangoadmin, es el encargado de hacer las cosas que no puede hacer el responsable, como dar de alta a los usuarios. Entendemos que este rol lo desempeñará un administador de sistemas.

# DESCRIPTORES

* Industrial.
* Producción.
* Gestión.
* Planificación.
* Seguimiento.

ÍNDICE

[RESUMEN iii](#_Toc41148148)

[DESCRIPTORES iii](#_Toc41148149)

[ÍNDICE DE ILUSTRACIONES v](#_Toc41148150)

[Capítulo 1: OBJETIVOS DEL PROYECTO 1](#_Toc41148151)

[1.1. Tareas principales 1](#_Toc41148152)

[1.2. Planificación temporal 2](#_Toc41148153)

[Capítulo 2: ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA 3](#_Toc41148154)

[2.1 Introducción 3](#_Toc41148155)

[2.1.1. Alcance del proyecto 3](#_Toc41148156)

[2.2 Descripción general 3](#_Toc41148157)

[2.2.1. Catálogo de requisitos 3](#_Toc41148158)

[2.3. Descripción de requisitos del nuevo sistema 4](#_Toc41148159)

[2.3.1. Modelo funcional 4](#_Toc41148160)

[2.4. Descripción de la interfaz del sistema 5](#_Toc41148161)

[2.4.1. Perfil de los usuarios 5](#_Toc41148162)

[Capítulo 3: ESPECIFICACIÓN DEL DISEÑO 6](#_Toc41148163)

[3.1. Introducción 6](#_Toc41148164)

[3.1.1. Principales funciones del software 6](#_Toc41148165)

[3.1.2. Descripción del entorno de desarrollo 6](#_Toc41148166)

[3.2. Arquitectura física y entorno tecnológico 7](#_Toc41148167)

[3.2.1. Descripción general 7](#_Toc41148168)

[3.3. Descripción del diseño 7](#_Toc41148169)

[3.3.1 Especificación de las interacciones 7](#_Toc41148170)

[3.3.2. Diseño de la estructura física de los datos 8](#_Toc41148171)

[Capítulo 4: MANUAL DE USUARIO 13](#_Toc41148172)

[Capítulo 5: INCIDENCIAS DEL PROYECTO Y CONCLUSIONES 19](#_Toc41148173)

[BIBLIOGRAFÍA 20](#_Toc41148174)

# ÍNDICE DE ILUSTRACIONES

[Ilustración 1: Commits **¡Error! Marcador no definido.**](#_Toc41146840)

[Ilustración 2: Modelo funcional 4](#_Toc41146841)

[Ilustración 3: Diagrama de uso 5](#_Toc41146842)

[Ilustración 4: Arquitectura física 7](#_Toc41146843)

[Ilustración 5: Estructura física 8](#_Toc41146844)

[Ilustración 6: Propuesta catálogo 9](#_Toc41146845)

[Ilustración 7: Catálogo actual 9](#_Toc41146846)

[Ilustración 8: Propuesta tareas 10](#_Toc41146847)

[Ilustración 9: Tareas actuales 10](#_Toc41146848)

[Ilustración 10: Propuesta menú principal 11](#_Toc41146849)

[Ilustración 11: Menú principal actual 11](#_Toc41146850)

[Ilustración 12: Propuesta tareas para empleado 12](#_Toc41146851)

[Ilustración 13: Tareas para empelados actual 12](#_Toc41146852)

[Ilustración 14: Catálogo actual 13](#_Toc41146853)

[Ilustración 15: Listado de equipos 14](#_Toc41146854)

[Ilustración 16: Listado de empleados 14](#_Toc41146855)

[Ilustración 17: Listado de procesos 15](#_Toc41146856)

[Ilustración 18: Listado de pedidos cliente 1 15](#_Toc41146857)

[Ilustración 19: Listado de pedidos cliente 2 15](#_Toc41146858)

[Ilustración 20: Listado de pedidos responsable 16](#_Toc41146859)

[Ilustración 21: Listado de tareas 16](#_Toc41146860)

[Ilustración 22: Listado de pedidos responsable 17](#_Toc41146861)

[Ilustración 23: Listado de pedidos no planificados 17](#_Toc41146862)

[Ilustración 24: Listado de tareas 17](#_Toc41146863)

[Ilustración 25: Listado de pedidos responsable 18](#_Toc41146864)

# Capítulo 1: OBJETIVOS DEL PROYECTO

## Tareas principales

Realizar un proyecto que cumpla las siguientes características:

* El proyecto debe aportar una solución a la siguiente premisa:

*Reto 4: Gestión de la producción*

*La empresa Deustubular S.L. quiere digitalizar parte de los procesos de producción de laempresa. La empresa cuenta con un amplio departamento de IT y se ha decidido encargara vuestro equipo su desarrollo.*

*La empresa quiere dejar de utilizar papel y poner un PC junto a cada puesto de trabajo, donde estará siempre abierta la aplicación web a desarrollar. La idea es que el responsableintroduzca los procesos a realizar desde la oficina y los operarios puedan ver un listado conlos procesos cuando están junto a las máquinas (e incluso actualizar la información, porejemplo, inicio y fin del proceso). Así se evitará tener que imprimir toda la información y llevarla a los puestos de trabajo.*

*Un proceso puede tener más de un empleado asignado, aunque solo se realiza con unúnico equipo.Esta es la información que han pensado manejar:*

*● Proceso:*

*○ Código de la orden de fabricación*

*○ Código del proceso*

*○ Nombre del proceso*

*○ Referencia*

*○ Inicio y Fin*

*○ Equipo empleado*

*○ Empleados asignados*

*● Equipo*

*○ Modelo*

*○ Marca*

*○ Tipo o categoría*

*○ Fecha de adquisición*

*○ Fecha de instalación*

*● Empleado*

*○ DNI*

*○ Nombre*

*○ Apellidos*

*○ Email*

*○ Teléfono*

*Siendo esta la información principal que se maneja, las funcionalidades más importantesrequeridas son:*

*● Gestión de procesos: creación, visualización (listado y detalle), modificación y baja.*

*● Gestión de equipos: creación, visualización (listado y detalle), modificación y baja.La empresa también está abierta a cualquier funcionalidad extra que se quiera incorporar:*

*● Gestión de órdenes de fabricación*

*● Gestión de empleados*

* Implementación de una interfaz de usuario que, a nivel usabilidad, mantiene un diseño que facilita la interacción con el usuario.
* Calidad del código óptima.

## 1.2. Planificación temporal

En cuanto a la planificación temporal, hemos seguido el siguiente esquema de trabajo:

* Implementar un horario que ha sido respetado por todos los miembros del grupo; de lunes a sábado, desde las 10:00 hasta las 14:00. De añadir otra jornada, si fuera necesario, y compatible con los horarios, se trabaja por la tarde.
* Mantener un ritmo de commits constantes. Y junto a eso, la participación equivalente de todos los miembros del grupo.
* Respetar la fecha y hora límite de la entrega.

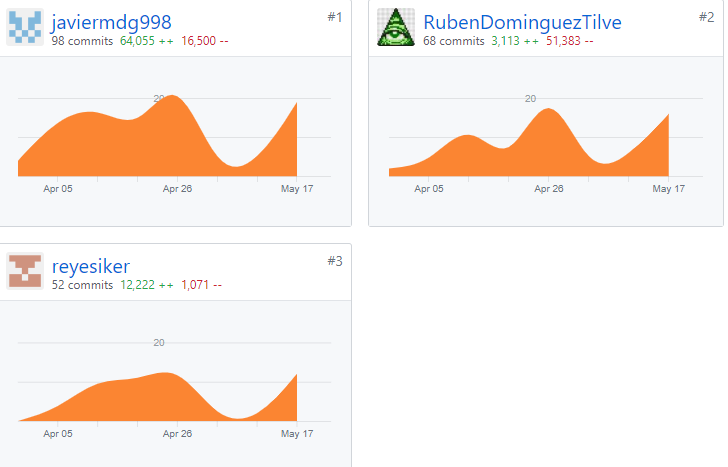


Ilustración 1: Commits

Estos fueron los recursos que utilizamos para la facilitación del trabajo:

* MySQL Workbench, para el diseño de la base de datos.
* Trello, para organizar las tareas y como nexo a la documentación y recursos adicionales.
* Drive, para subir los archivos que teníamos que compartir.
* Discord, para facilitar la comunicación.
* GitHub y GitKraken para el código de la aplicación.
* Herramientas de Microsoft Office.
* Librerías tipo Boostrap o Scss.
* Prepros.
* MockFlow, para la maquetación de vistas.
* FlatuiColors, para la paleta de colores.
* FlatIcons, para los iconos de la aplicación.

# Capítulo 2: ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

## 2.1 Introducción

Es una aplicación pensada para correr en un servidor, pero se usará tanto por los clientes (a través de Internet, desde su localización) como por los operarios que están trabajando en la línea.

### 2.1.1. Alcance del proyecto

Es una aplicación en las que se diferencian dos funcionalidades: la parte interna y la externa. En cuanto a la parte interna respecta, nos permitirá controlar las tareas realizadas por los empleados, y permitirá que los responsables se encarguen de la gestión de todos los elementos.

Por otra parte, en cuanto a la parte externa se refiere, la aplicación hará de portal para que los potenciales clientes puedan tener un fácil acceso a nuestro catálogo actualizado y realicen pedidos de forma cómoda y accesible para nosotros.

## 2.2 Descripción general

### 2.2.1. Catálogo de requisitos

Se incluye la gestión en la base de datos, el CRUD (create, remove, update y delete) para los siguientes elementos; empleados que trabajan en la planta, los productos que están en venta en el catálogo, los pedidos que realizan los clientes, las tareas asociadas a cada pedido, los equipos de la empresa en los que se realizarán las tareas y los procesos que realizan las tareas.

La autenticación de usuarios que defina los roles que pueden cumplir en el proyecto: cliente realiza pedidos, empleados completan tareas y responsables se dedican a la gestión de la base de datos, en la que tienen acceso para modificar cualquier dato almacenado del proyecto.

Si el tiempo de trabajo hubiese sido mayor, nos hubiera gustado que nuestra aplicación hiciese un seguimiento de los pedidos, que muestre el porcentaje de completación de dichos pedidos, la automatización de las tareas, mejorar la interfaz gráfica (añadirle complejidad), mejorar los gráficos presentes en la aplicación y añadirle una paginación a las tablas y/o listados en el catálogo. Por ejemplo, si las tareas que requiere cada pedido estuvieran definidas se podría automatizar el seguimiento del pedido dividido en: tareas sin planificar, tareas planificadas y tareas completadas.

Entre los inconvenientes con los que nos hemos encontrado, cabe destacar que en la parte de interacción con los clientes, nos hubiera gustado que cuando el cliente realice un pedido, tenga el suficiente tiempo para poder modificarlo.

# 2.3. Descripción de requisitos del nuevo sistema

### 2.3.1. Modelo funcional

Ilustración 2: Modelo funcional

## 2.4. Descripción de la interfaz del sistema

### 2.4.1. Perfil de los usuarios

Los clientes son los únicos capaces de crear sus propios usuarios, y además, solo pueden especificar su nombre de usuario, su contraseña y la empresa a la que pertenecen. El resto son creados por el equipo de Deustronics bajo petición de los responsables.

La idea es, que si esto fuera una empresa real, los usuarios se fueran dando de alta cuando los trabajadores de Recursos Humanos transmiten la contratación del nuevo personal.

La correcta creación y asignación de roles es esencial para el correcto funcionamiento del proyecto porque serán estos roles los que delimiten las acciones que puede desempeñar cada usuario.

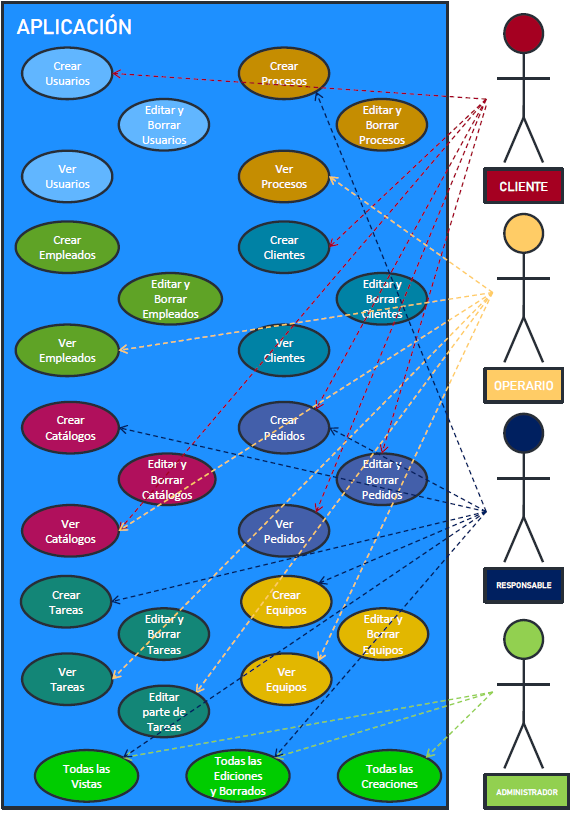


Ilustración 3: Diagrama de uso

# Capítulo 3: ESPECIFICACIÓN DEL DISEÑO

## 3.1. Introducción

Al comenzar el proyecto, decidimos que quería que hiciese nuestra aplicación. Y por ello, decidimos diseñar una base de datos. Aunque la base de datos iba a ser Django, decidimos hacer un diseño gráfico y visual en MySQL Workbench, porque era un programa que dominábamos e iba a servirnos de guía.

Por último, programamos el modelo de datos en Django, como requerían las especificaciones del proyecto.

### 3.1.1. Principales funciones del software

* Gestión de tareas.
* Gestión de catálogo.
* Gestión de pedidos.
* Gestión de procesos.
* Gestión de equipos.
* Gestión de empleados.
* Creación de perfiles por parte de los propios clientes.

### 3.1.2. Descripción del entorno de desarrollo

Estas son las versiones del software que hemos utilizado:

* Python 3.8.2.
* Django 3.0.5.
* Pillow 7.1.2.
* Css 3.
* Boostrap 4.
* Html 5.

Y estas son las aplicaciones que hemos usado:

* VisualStudioCode, porque entre los entornos que ofrecen una interfaz de Python, es con el que más cómodo nos sentíamos trabajando. Barajamos también la opción de trabajar con Sublime o PyCharm.
* Prepros, para compilar y aplicar el código Scss.

Aunque el desarrollo se ha realizado con VisualStudioCode para poder programar el código, no es necesario que durante el funcionamiento en producción de la aplicación se tenga instalado ningún IDE, ya que la aplicación sigue un esquema cliente-servidor. Lo único necesario es una servidor que corra la aplicación, y en cada puesto de producción en la que quieran gestionarse tarea, con cualquier navegador web servirá.

## 3.2. Arquitectura física y entorno tecnológico

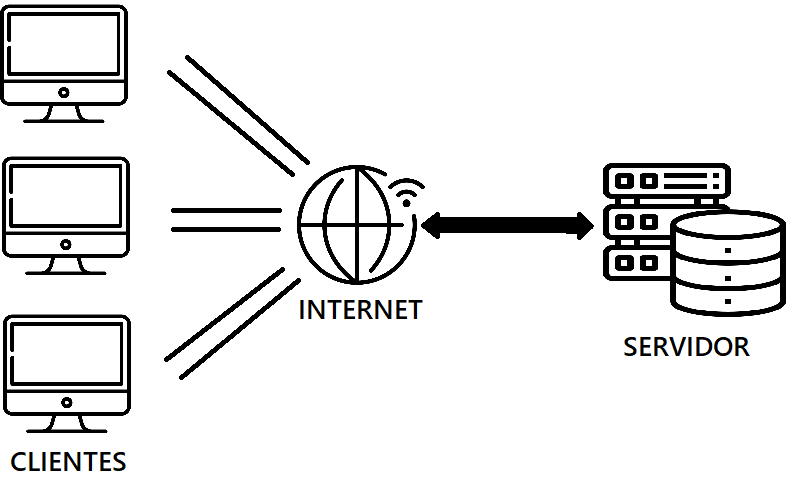


Ilustración 4: Arquitectura física

### 3.2.1. Descripción general

La aplicación no requiere de ningún entorno tecnológico complejo, ya que lo único que se necesita es una conexión entre el servidor y las estaciones. Si se quiere implementar la interfaz de venta a cliente, el servidor debe ser accesible por Internet para que los clientes puedan comunicarse con él.

## 3.3. Descripción del diseño

### 3.3.1 Especificación de las interacciones

Los clientes solicitan, mediante la aplicación, datos al servidor y la aplicación cuando recibe la respuesta del servidor se encarga de mostrar esos datos, como pueden, por ejemplo: las tareas que se le muestran a un operario y cuando este marca la tarea como realizada, la aplicación comunica esta acción al servidor para que lo actualice en la base de datos. Cuando el servidor confirma que se ha realizado la acción, devolviendo una respuesta. La aplicación queda a la espera de otra interacción.

### 3.3.2. Diseño de la estructura física de los datos

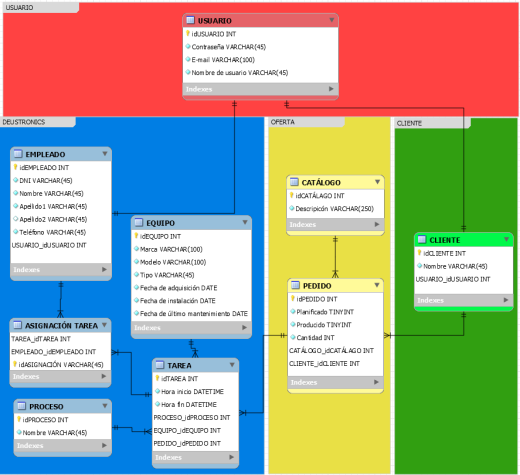


Ilustración 5: Estructura física

#### 3.3.2.1 Definición de vistas

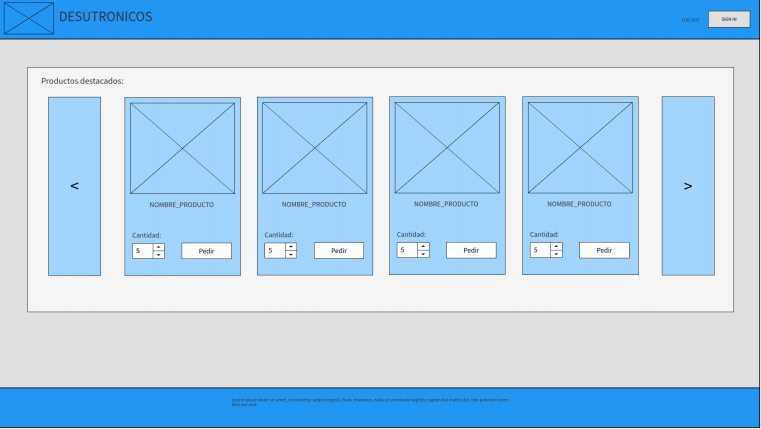


Ilustración 6: Propuesta catálogo

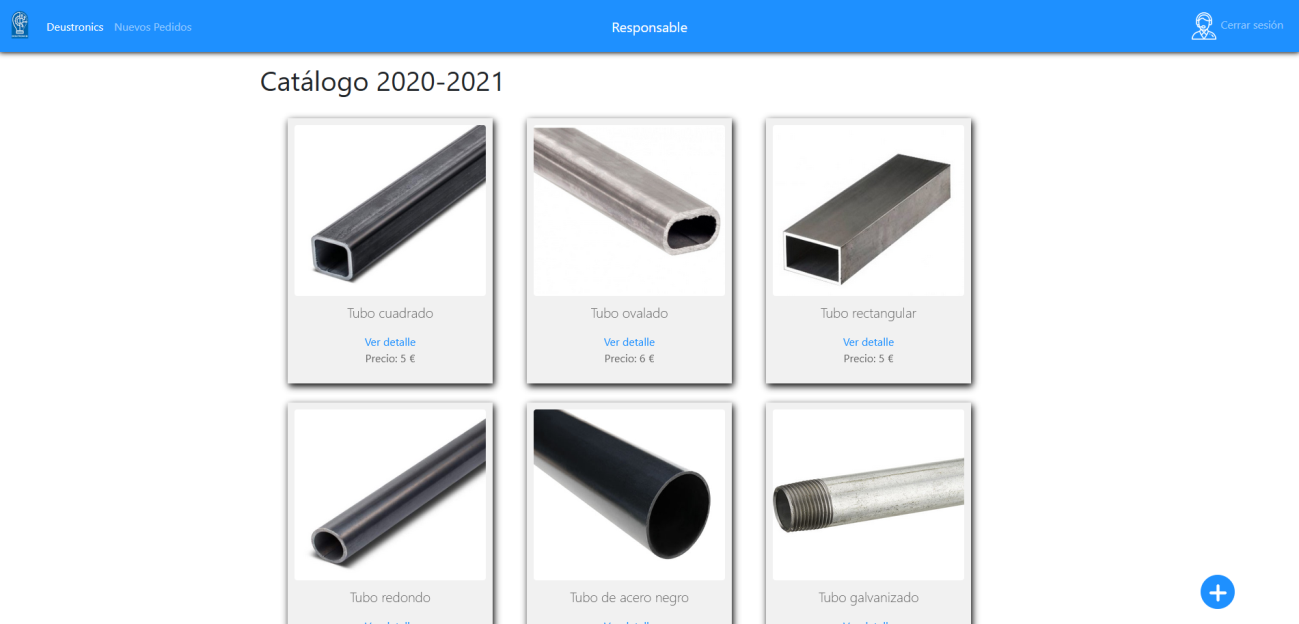


Ilustración 7: Catálogo actual

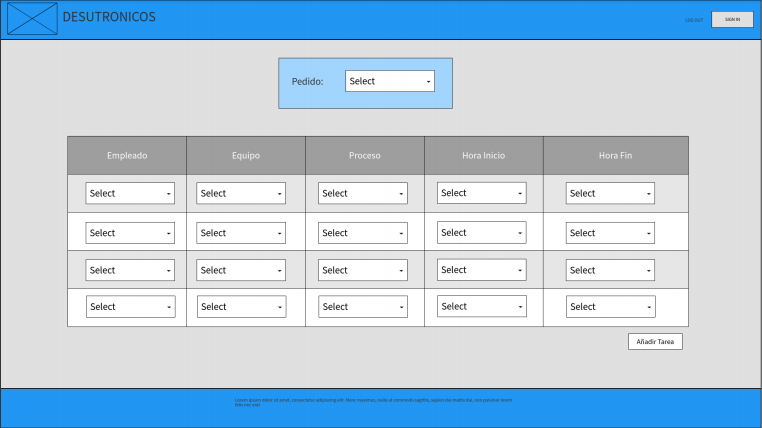


Ilustración 8: Propuesta tareas



Ilustración 9: Tareas actuales

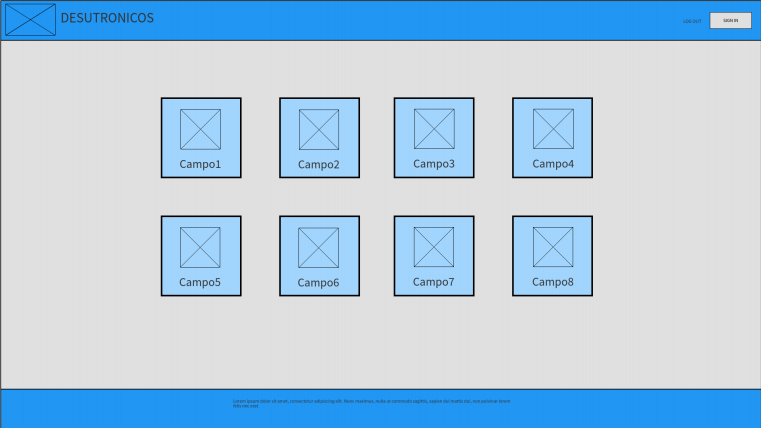


Ilustración 10: Propuesta menú principal

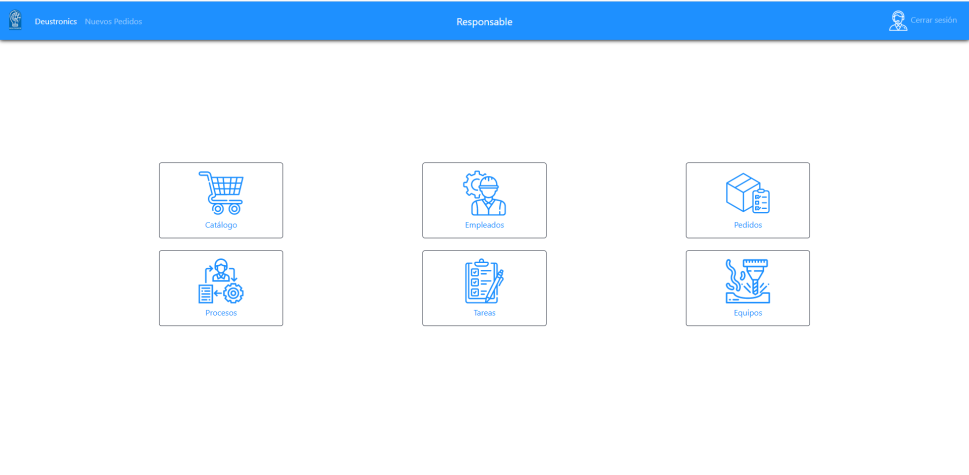


Ilustración 11: Menú principal actual



Ilustración 12: Propuesta tareas para empleado

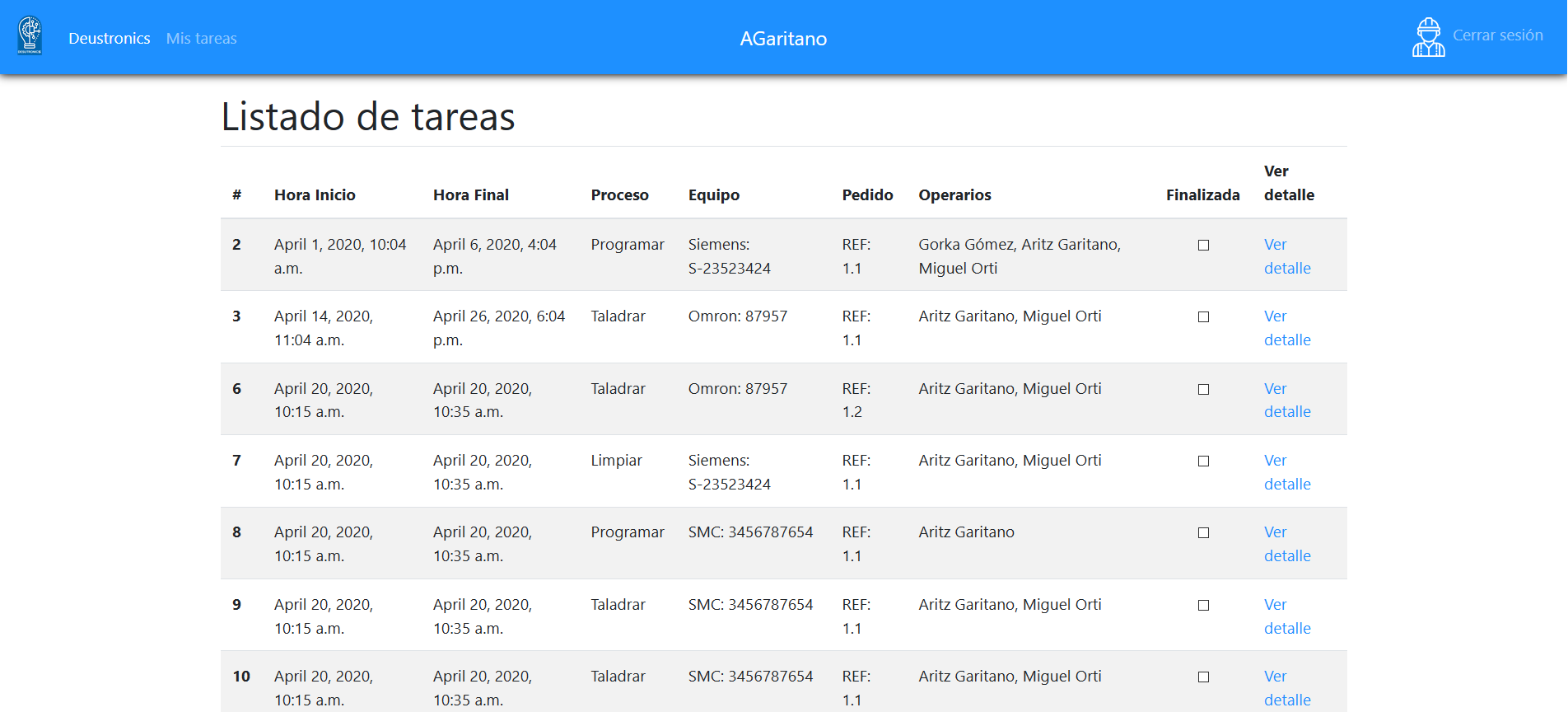


Ilustración 13: Tareas para empelados actual

# Capítulo 4: MANUAL DE USUARIO

**Paso 1:** una vez creado el *“superuser”,* crear los grupos de usuarios de la base de datos. Grupos necesarios: CLIENTE, OPERARIO y RESPONSABLE.

**Paso 2:** dar de alta como usuarios a todos los trabajadores que se quieran añadir, y crear, al menos, un responsable que se encargue de gestionar la aplicación.

**Paso 3:** asignar grupo a todos los usuarios anteriormente creados.

**Paso 4:** definir los tipos de los equipos con los que vamos a trabajar, porque luego el encargado no podrá modificarlos sin acceder a *“DjangoAdmin”.*

*Estos cuatro pasos, si se conserva la base de datos, no deberían ser necesarios.*

**Paso 5:** acceder a la aplicación como *“Responsable”,* y añadir productos en el catálogo.

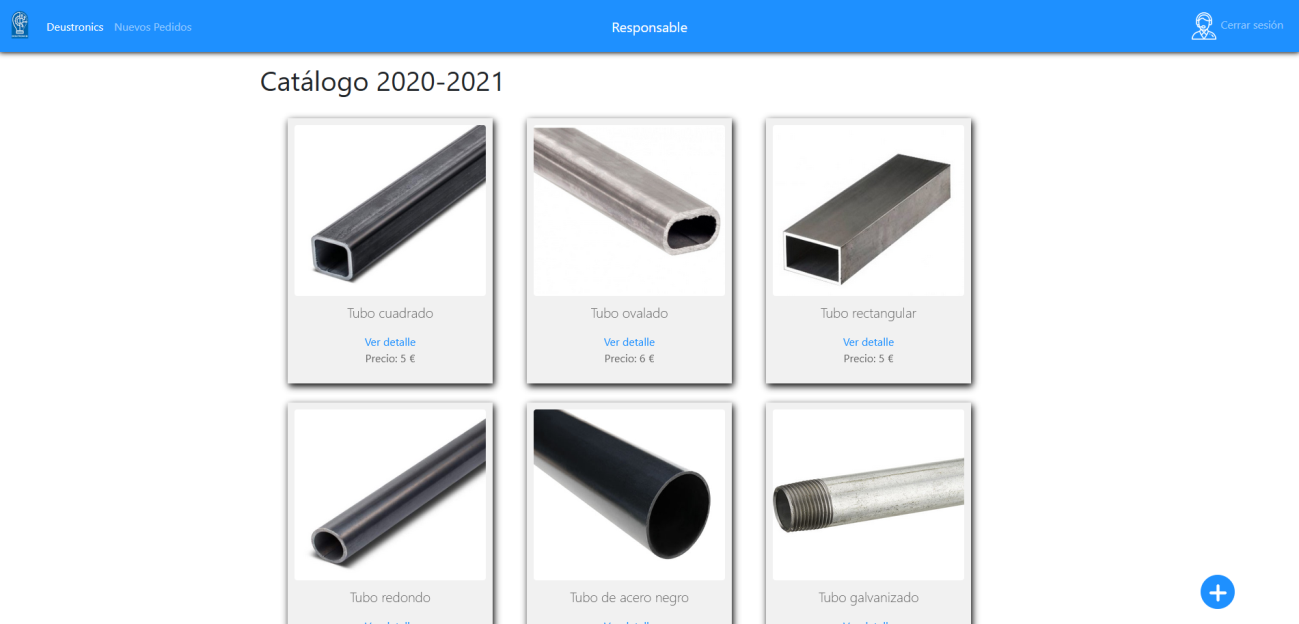


Ilustración 14: Catálogo actual

**Paso 6:**crear los equipos con los que vamos a trabajar.

Ilustración 15: Listado de equipos

**Paso 7:** dar de alta a los empleados en la base de datos. Teniendo en cuenta los usuarios que les hemos creado antes.



Ilustración 16: Listado de empleados

**Paso 8:** definir los procesos que van a seguir los empleados con sus equipos correspondientes.

Ilustración 17: Listado de procesos

**Paso 9:** cerrar la sesión de *“Responsable”,* y registrarse como cliente. Una vez dentro, entrar al catálogo y realizar un pedido. Una vez pedido, podemos comprobar su estado en el apartado *“Mis pedidos”.*

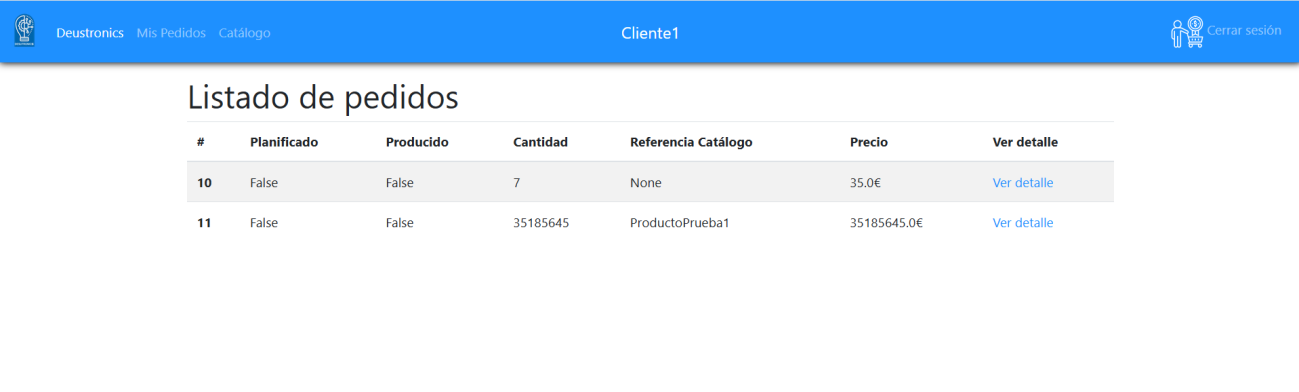


Ilustración 18: Listado de pedidos cliente 1

Si nos registramos con otro cliente, solo aparecerán sus pedidos.

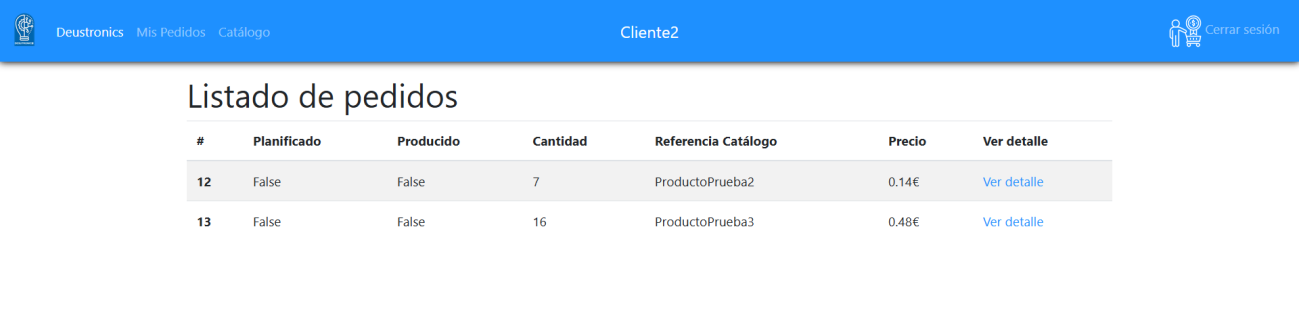


Ilustración 19: Listado de pedidos cliente 2

**Paso 10:** volvemos a iniciar sesión con el *“Responsable”,* y entramos al apartado *“Nuevos pedidos”.* Podremos ver que aparecen los pedidos anteriormente realizados por los clientes en cuestión.

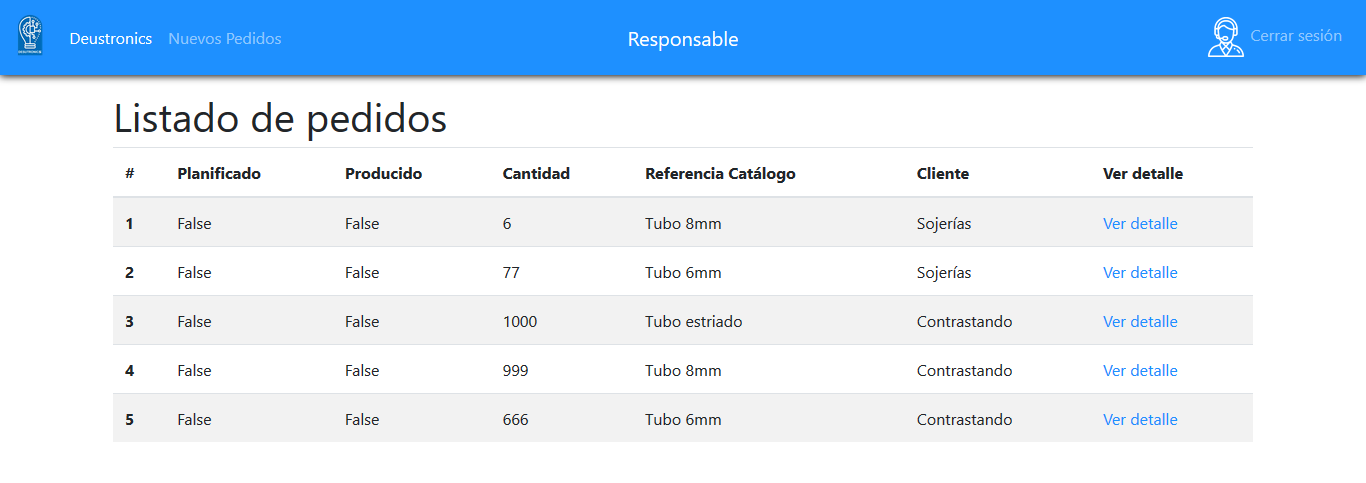


Ilustración 20: Listado de pedidos responsable

**Paso 11:** debemos asignar las diferentes tareas en las que se desglosa cada pedido.



Ilustración 21: Listado de tareas

**Paso 12:** cuando el *“Responsable”* considere que ya ha asignado todas las tareas correspondientes a un pedido, debería editar el pedido y marcarlo como planificado.



Ilustración 22: Listado de pedidos responsable

Después de marcarlo, se puede ver en el apartado *“Nuevos pedidos”* que los planificados ya no aparecen.



Ilustración 23: Listado de pedidos no planificados

**Paso 13:** iniciamos sesión con uno de los operarios, y vemos sus tareas asignadas.

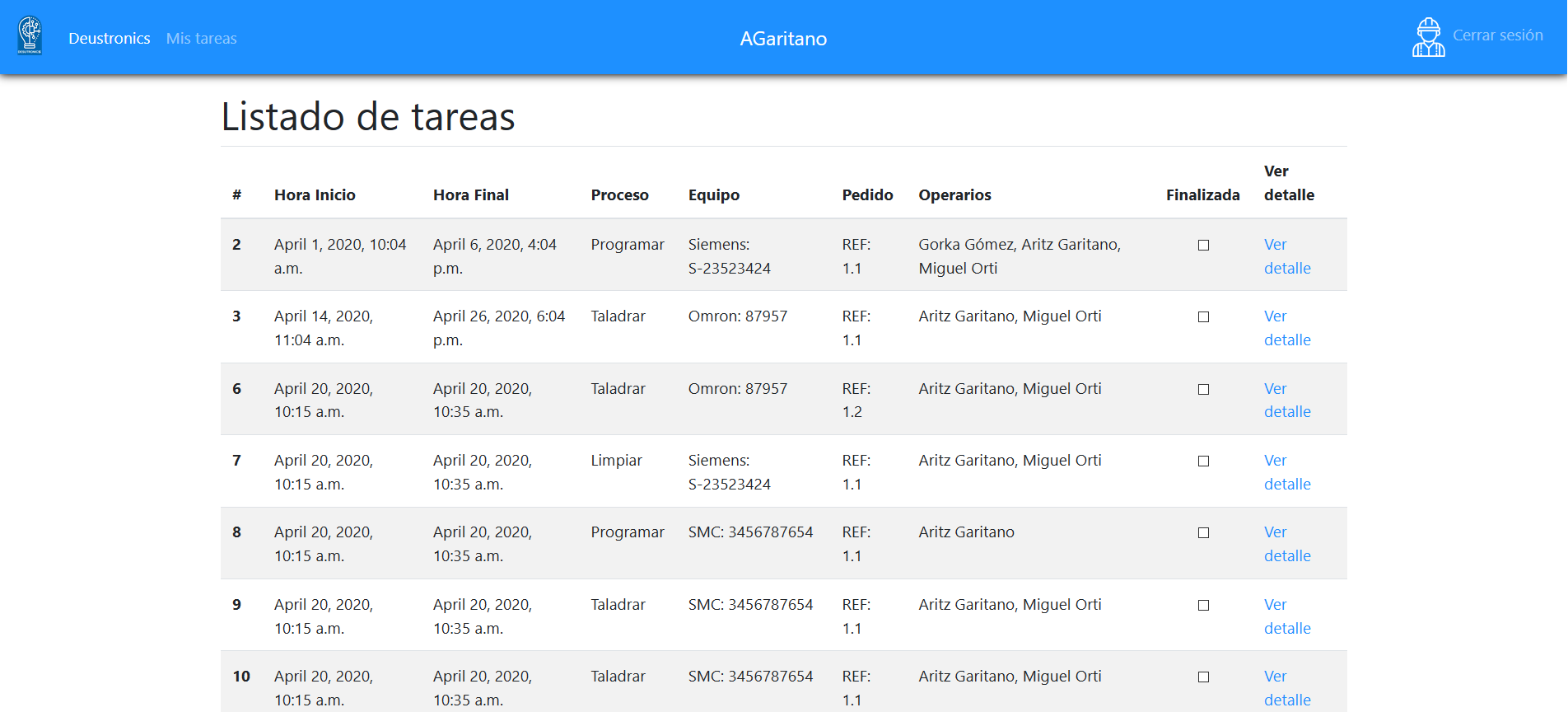


Ilustración 24: Listado de tareas

Una vez el operario termine la tarea, podrá marcarla como finalizada y desaparecerá con una elegante animación.

**Paso 14:** volvemos a iniciar sesión como *“Responsable”,* marcamos los pedidos cuyas tareas han sido finalizadas como producidos.



Ilustración 25: Listado de pedidos responsable

Ahora el cliente podrá ver que su pedido está producido.

# Capítulo 5: INCIDENCIAS DEL PROYECTO Y CONCLUSIONES

Estamos muy contentos con el resultado final del proyecto, aunque nos hemos quedado con ideas en el tintero. Al ser un proyecto tan largo, hay días que se han hecho más tediosos que otros, pero al mantener un ritmo constante y un horario predefinido, siempre hemos ido muy bien de tiempo. Ese mismo ritmo es el que nos ha facilitado esta última entrega, ya que muchos de los requerimientos de ella los habíamos completado en la anterior.

En cuanto a las incidencias de programación, al ser un lenguaje y framework que desconocíamos, hemos tenido que dedicarle muchas horas a la investigación. Hemos cometido muchos errores de “novatos”, ya que al tener poca experiencia con el lenguaje había mecanismos del lenguaje y del framework que hemos ido descubriendo.

# BIBLIOGRAFÍA

* <https://getbootstrap.com/>
* <https://www.djangoproject.com/>
* <https://es.stackoverflow.com/>
* <https://stackoverflow.com/>
* <https://www.flaticon.com/>
* <https://www.youtube.com/watch?v=yZDH6vH3iM8&list=PL8ZnVqiE4oiY6fh6_vvNKwkxfutf3CiMY>
* <https://www.mockflow.com/>
* <https://flatuicolors.com/>