

Buscando inspiración para realizar un proyecto con el que optar a “Dualiza” hemos recordado a nuestros amigos los **Jawas**.

Como estamos seguros que todos habéis visto las pelis sabéis perfectamente lo que hacían estos bichos, el minuto exacto en el que aparecen por primera vez, su influencia en el desarrollo de las pelis y hasta seguro que chapurreáis un poco de su idioma... no esperamos menos.

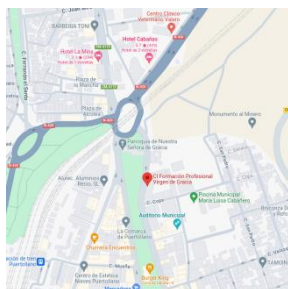
El caso es que esta gentecilla se dedicaba a traficar con cacharros electrónicos, androides principalmente, que compraban u obtenían de la forma que fuera, y en el estado que fuera, los medio reparaban y los ponían en circulación otra vez. Nosotros haremos algo parecido, cogeremos cacharros obsoletos o descatalogados que ya no funcionen y los convertiremos en joyas y abalorios varios con fines no lucrativos.



Nuestra aplicación permitirá lo siguiente:

- Que **empresas externas** se registren como colaboradoras cediendo material que ya no utilicen. Estas empresas indicarán el lote que donarán y dónde lo recogeremos.
- La aplicación contemplará la figura del usuario **clasificador**, este usuario desguazará el lote y almacenará las piezas clasificándolas por tipo y cantidad entre otros campos que sean útiles.
- En nuestra aplicación también podemos entrar como **administradores** que permitirá gestionar usuarios, tipos de cacharros y demás inventarios.
- Asimismo también podremos entrar como usuarios **diseñadores de joyas**.
- Para todos los usuarios existirán la opción de tener y gestionar su foto de perfil.

Empresa



Las empresas se registrarán y podrán indicar el **lote de hardware** que quieren donar y **dónde lo harán**.

El lote se marcará como entregado cuando sea recibido por el centro. El **lugar de ubicación** será uno elegido a través de **Google Maps**.

Las empresas podrán hacer *login* siempre que quieran para añadir nuevas entregas o comprobar la lista de entregas que han realizado. También podrán cancelar algún lote si finalmente no lo quieren

entregar.

Clasificador

El lote será despiezado y clasificado. En dicha clasificación este usuario indicará una **cantidad** de elementos, una **descripción** general de ellos y un **tipo** (placa, memoria, hd, monitor, etc...). Estos campos lógicamente serán ampliables y será el administrador el que los habilite o deshabilite. El clasificador por lo tanto tendrá acceso a un CRUD para los componentes existentes en el sistema.

Administrador

Los **administradores** gestionarán a los **usuarios registrados** permitiendo realizar un CRUD con ellos, incluyendo el cambio de rol si fuera necesario.

También gestionarán los **tipos de hardware** que el sistema tratará. Es decir, los tipos de hardware que admitiremos y que son necesarios para el diseño de los adornos que hagamos.

Los **administradores** también podrán **añadir componentes** al sistema pero **no podrán gestionar los lotes**, esto lo harán los clasificadores exclusivamente.



Diseñador



Los **diseñadores** podrán gestionar un CRUD de joyas. Para las joyas, el diseñador, establecerá unos *ingredientes* para realizar cada una. Por ejemplo, **pendienteHD**, necesitamos:

- 2 discos duros.
- Dos enganches de pendiente.
- 2 ositos de goma.

De forma que el diseñador pueda elegir qué joya quiere hacer y el sistema le diga si tiene componentes suficientes para hacerla. Incluso que le diga cuántas de esas joyas podría hacer con los componentes actuales.

Por lo tanto el **diseñador** podrá hacer:

- Gestionar joyas y “*recetas*” de elaboración de las mismas.
- CRUD de componentes para cada receta.
- Consulta de posibilidad de realizar una joya con los componentes actuales.
- Consulta de cuantas joyas de un tipo con los componentes





DESARROLLO DE APLICACIONES WEB

DAW 2

Desafío 1



actuales.

- También podrá guardar una foto de referencia de cada joya.
- Tendremos también un generador de joyas aleatorias que se pueda parametrizar, es decir, el diseñador indicará qué tipo de joya quiere y el sistema, teniendo en cuenta el histórico de joyas y los componentes actuales podrá sugerir combinaciones. Este algoritmo se deja libre al programador.

Valoraciones

Se tendrá en cuenta de modo general:

- Diseño de la BD que recoja toda la información de forma correcta.
- Que **lo programado se ajuste a lo pedido**.
- Una planificación de tareas coherente.
- Que los *sprints* de desarrollo cumplan lo planificado.
- El **proyecto deberá estar sincronizado con un repositorio Git y Board** de planificación. Todo esto será consultable por los profesores en cualquier momento del desarrollo del proyecto.

Diseño de interfaces

- Se usará Bootstrap 5.2.
- Se deben crear dos maquetaciones diferentes del sitio, suficientemente dispares.
- CSS específico realizado en SASS (lo más reusable posible). Dentro de CSS habrá que incluir, de la forma más coherente posible:
 - Transformaciones
 - Animaciones
 - Transiciones (que no estén incluidas en Bootstrap por defecto...)
- Se tendrán en cuenta los principales aspectos de usabilidad vistos en clase.
- El código generado debe ser *responsive*.

Desarrollo web en entorno cliente:

- Generación de código estructurado y modular. Utilización de POO. Utilización correcta de funciones.
- Utilización correcta de estructuras de almacenamiento con JS.
- Utilización correcta del proyecto con **webpack** e importación y exportación de módulos.
- Gestión correcta de eventos.
- Validación de formularios.
- Utilización correcta de promesas en la comunicación asíncrona con el servidor utilizando *fetch*.
- Planificación correcta y aplicación eficiente de la metodología *SCRUM*.



DESARROLLO DE APLICACIONES WEB

DAW 2

Desafío 1



Desarrollo web entorno servidor:

- Elección y agrupación de rutas.
- Verbos y respuestas estándar.
- *Eloquent* y *Query Builder*.
- Protección de rutas con *token* de seguridad. *Middlewares* y *abilities*.
- *Migrations*, *Factories* y *Seeders*.
- Validación de datos. *Validator*.
- Algoritmos diseñados para las tareas propuestas.
- Parametrización del sistema.
- Realización de *tests*.

Despliegue de Aplicaciones Web

- Utilización de *Sonarqube*
- Despliegue del cliente y el servidor en apache local



DESARROLLO DE APLICACIONES WEB

DAW 2

Desafío 1



ANEXO: METODOLOGÍA SCRUM

SCRUM es una metodología ágil en la que se aplican regularmente un conjunto de buenas prácticas para trabajar en equipo y obtener el mejor resultado en el proyecto.

En SCRUM se realizan entregas parciales y regulares del producto final, priorizándolas según el beneficio que aportan al receptor del proyecto (cliente).

Los puntos destacables de esta metodología de trabajo son:

ROLES:

- **Product Owner:** se encarga de analizar qué es lo que quiere el cliente final en su proyecto y lo detalla en una lista priorizada (**product backlog**).
- **Scrum Master:** organiza y facilita el trabajo. Defiende la metodología a llevar a cabo ante el product owner.
- **Team:** equipo de desarrolladores. Analizan las tareas del **product backlog**, detallan los ítems que van a contener, su prioridad y el tiempo que les va a llevar realizarlas.

Una persona puede tener más de un rol. Por ejemplo, el scrum master puede formar parte del equipo de desarrollo.

FUNCIONAMIENTO:

El trabajo se organiza en **sprints**. Un sprint tiene un tiempo acotado (por ejemplo 2 semanas) y en él se incluyen las tareas (historias) establecidas en el **product backlog**.

El sprint comienza con una planificación (**planning**): se reúne el **team, product owner y scrum master**. Se analiza cada tarea establecida por el product owner en el backlog. Se prioriza, determina la dificultad técnica por el equipo de desarrollo y se establece el tiempo que tardará en completarse. Una vez analizada, se decide si esa tarea “**cabe**” en dicho sprint, en cuyo caso, el team se **COMPROMETE** a realizarla en dicho sprint.

Cuando finaliza el sprint, se realiza una **retrospectiva** analizando lo que se ha hecho bien y lo que se ha hecho mal, para proponer mejoras al respecto. Después se realiza una **demo** de las historias del sprint que se han **completado**, ante el **product owner** (y a veces ante el cliente final).

Cada día del sprint se hará una pequeña reunión denominada **daily** (tiene que ser corta (10') y se obliga a estar de pie para que no se alargue), en la que cada miembro del equipo de desarrollo cuenta en lo que ha trabajado, las dificultades y si necesita ayuda por parte del resto del equipo.

Cuando comienza un sprint, existe un tablero donde se encuentran todas las tareas sin asignar pendientes del sprint (**backlog**), y que cada desarrollador irá escogiendo



DESARROLLO DE APLICACIONES WEB

DAW 2

Desafío 1



(teniendo en cuenta al equipo) y asignando según sus preferencias, ritmo de trabajo, etc.