

# PORTADA

## **CAPÍTULO 1: INTRODUCCIÓN**

---

# Índice

## 1.- Capítulo 1: Introducción

- 1.1.- .....Introduccion del Proyecto
- 1.2.-.....Proposito
- 1.3.-.....Objetivos del Proyecto
- 1.4.-.....Costes del Proyecto
  - 1.4.1.-.....Costes de Desarrollo
  - 1.4.2.-.....Costes de Implementacion

## 2.- Capítulo 2: Análisis del Sistema

- 2.1.-.....Introducción
- 2.2.-.....Análisis de requisitos
  - 2.2.3.-.....Requerimientos funcionales
- 2.3.-Casos de Uso:

## 3.-Diseño del Sistema

- 3.1.-.....Introducción
- 3.2.-.....Diagrama de clases
- 3.3.-.....Diseño de la base de datos
- 3.4.-.....Diseño lógico
- 3.5.-.....Diseño de la intefaz

## 4.-Implementacion

- 4.1.-.....Introducción
- 4.2.-.....Arquitectura cliente/servidor
- 4.3.-.....Lenguajes de programación
- 4.4.-.....Herramientas de desarrollo
- 4.5.-.....Codificación

## 5.- Pruebas de Software

### 5.1.-.....Introducción

### 5.2.-.....Tecnicas de prueba

#### 5.2.1.-Pruebas de caja blanca

#### 5.2.2.-Pruebas de caja negra

## 6.-Conclusiones

### 6.1.-.....Conclusiones

### 6.2.-.....Ampliaciones

## 7.-Bibliografias y referencias

Anexo I: Manual de instalación

Anexo II:Manual de usuario

## **1.1.- INTRODUCCIÓN DEL PROYECTO**

Celeben es una aplicación e-commerce en la que podrás desde una interfaz fácil de usuario vender y comprar productos. Podrás registrarte como vendedor y crear y modificar tus propios productos.

## **1.2.- PROPÓSITO**

Celeben hace mas fácil tanto a los usuarios normales como a editores de contenido dentro del e-commerce y comerciales el uso de la misma mediante un backend fácil de usar y con roles administrados

## **1.3.- OBJETIVOS DEL PROYECTO**

- |   |
|---|
| <ul style="list-style-type: none"><li>1.- Desarrollo de un backend</li><li>2.- Desarrollo de interfaz</li><li>3.- Desarrollar de la logica de productos</li><li>4.- Desarrollar la logica de compra</li><li>5.- Desarrollar la logica de permisos</li></ul> |
|---|

## **1.4.- COSTE DEL PROYECTO**

Estimación del coste y esfuerzo del software y/ o hardware requerido para la realización del mismo. Podemos dividir en dos subapartados:

### **1.4.1.-COSTES DE DESARROLLO**

Disposicion a un ordenador con internet inestable(num horas + licencias)

### **1.4.2 COSTES DE IMPLANTACIÓN**

Implementacion en un servicio web como heroku o netlify, 20 euros al mes.

## **CAPÍTULO 2: ANÁLISIS DEL SISTEMA**

---

## 2.1.- INTRODUCCIÓN

Para empezar, el proyecto debe de tener una base de datos con tablas que se muestren en el backend. Una vez declarada la base de datos en el backend, podemos proceder a implementar la logica y relaciones entre ellas. Una vez hecho, preharemos queries que usaremos en el frontend para crear/leer/actualizar/eliminar productos en el frontend.

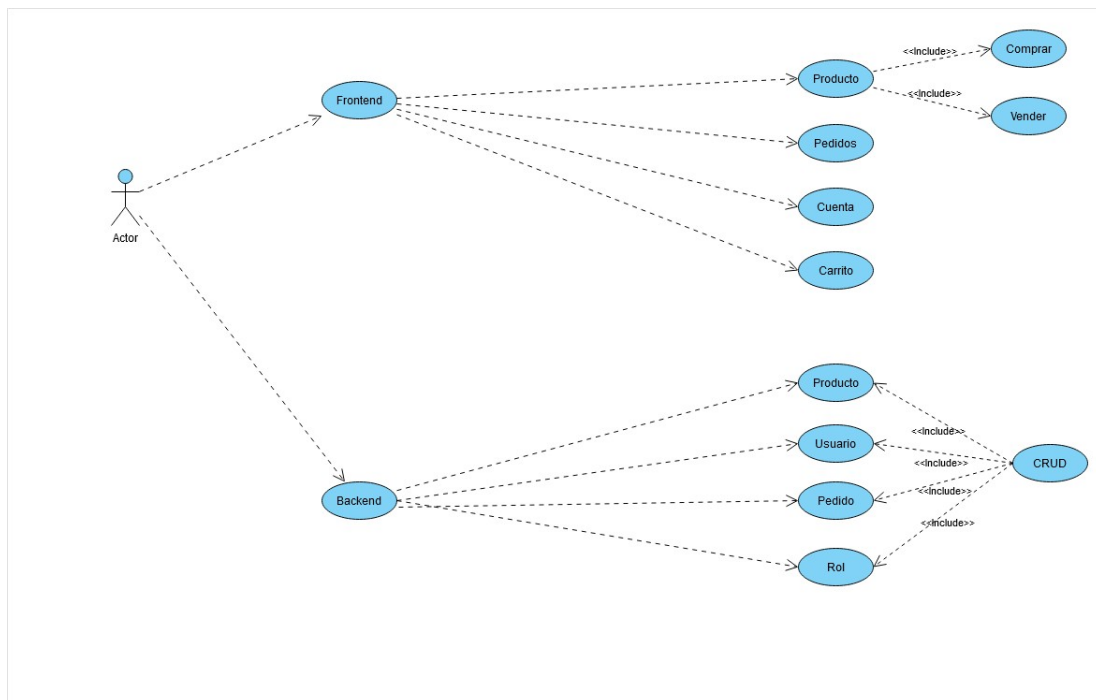
## 2.2.- ANÁLISIS DE REQUISITOS

El proposito del proyecto consiste en hacerlo fácil desde una interfaz con formularios tanto para el usuario normal como para el usuario avanzado que gestiona la aplicación desde el backend.

### 2.2.1. REQUERIMIENTOS FUNCIONALES.

Necesario una maquina virtual con nodejs instalado e internet, no es necesario tener la base de datos integrada, ya que se usan servicios cloud.

## 2.3.- CASOS DE USO: DIAGRAMAS Y NARRATIVAS DE CASOS DE USO



Paso 1: El actor solicita entrar al backend o al frontend

Paso 1: El actor solicita entrar al frontend

Paso 1.1: El actor puede elegir entre todas las opciones del menu.

Paso 1.1a: El actor elige Producto. Puede comprar productos o venderlos(si su rol se lo permite)

Paso 1.1b: El actor elige pedidos. Visualiza todos los pedidos asociados a su cuenta

Paso 1.1c: El actor elige cuenta. Visualiza los datos de su cuenta

Paso 1.1d:El actor elige carrito. Puede ver el carrito de compra eliminar productos del carrito y realizar la compra.

Paso 2: El actor solicita entrar al backend

Paso 2.1: El actor puede elegir entre todas las opciones del menu.

Paso 2.1a: El actor elige Producto. Puede realizar acciones CRUD sobre productos (si su rol se lo permite)

Paso 2.1b: El actor elige usuarios. Puede realizar acciones CRUD sobre usuarios (si su rol se lo permite)

Paso 2.1c: El actor elige pedido. Puede realizar acciones CRUD sobre pedidos (si su rol se lo permite)

Paso 2.1d: Puede realizar acciones CRUD sobre roles(si es el admin)



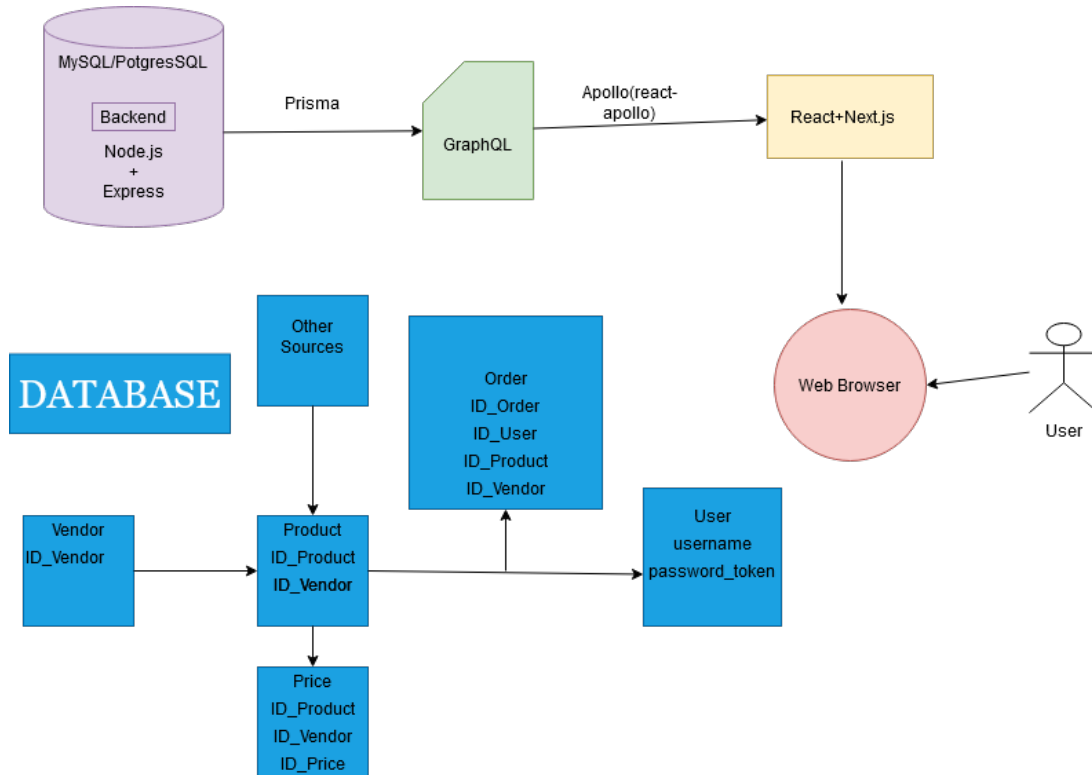
## CAPÍTULO 3: **DISEÑO DEL SISTEMA**

---

### 3.1.- INTRODUCCIÓN.

La base de datos será gestionada por un framework llamado KeyStone, que nos hara mucho más facil crear tablas y relacionarlas desde el backend con la base de datos.

### 3.2.- DIAGRAMA DE CLASES.



### 3.3.- DISEÑO DE LA BASE DE DATOS.

La base de datos está hecha en Mongo NOSQL, en ellas encontramos los siguientes documentos: Producto, Usuario, ImagenProducto, Order, OrderItem, Cart, CartItem, Roles

### 3.4.- DISEÑO LÓGICO.

User:

Id(autogenerado),name(required),email(required),password(required),cart,orders,role

Product:

Id(autogenerado),name(required),description,photo(required),status(required),price(resquired)

CartItem:

id,quantity,product,user

Order:

id,total,item,user,charge

OrderItem:

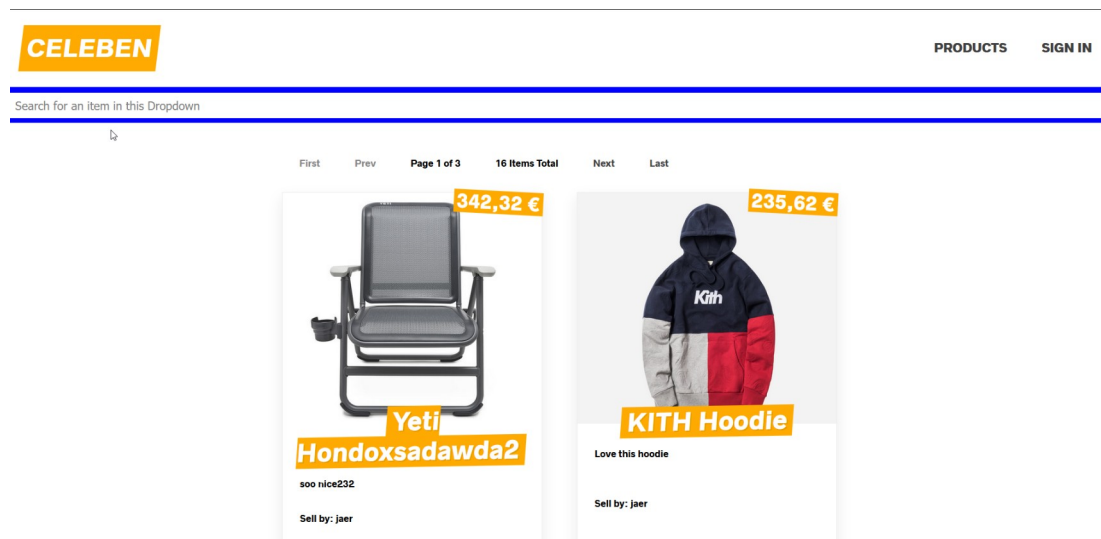
id,name,description,price,quantity,order,photo

Role:

id,name,assignedTo,permissonfields

### 3.5.-DISEÑO DE LA INTERFAZ.

El diseño de la interfaz está dividido en una barra de navegacion,con el logotipo y las diferentes páginas de las que dispone la aplicación. Justo debajo, una barra de busqueda global de la tienda, y un cuerpo donde se muestra el contenido de cada página en la que nos encontremos(productos,pedidos,informacion de cuenta...etc)



## **CAPÍTULO 4: IMPLEMENTACIÓN**

---

#### **4.1.- INTRODUCCIÓN.**

En este momento ya se encuentra definido el problema y la solución, por lo que lo siguiente será transformar el modelo obtenido en las actividades anteriores en código fuente.

#### **4.2.- ARQUITECTURA CLIENTE/SERVIDOR**

La arquitectura consiste en una relacion entre el front y el back con intermediadores. Por una parte, el front se comunica con el backend mediante Apollo(el cual hace como de REST api) y el backend con el front mediante GraphQL.

#### **4.3.- LENGUAJES DE PROGRAMACIÓN.**

JavaScript. En front end el framework React+ Next.js y en back end Node.js +Keystone.js

#### **4.4.- HERRAMIENTAS DE DESARROLLO.**

Visual Studio Code,MongoDBCompass,La base de datos hospedada en cloud.mongodb, Cloudinary como CDN para imagenes.

#### **4.5.- CODIFICACIÓN.**

Toda la codificación y este mismo documento se encuentran en:

<https://github.com/Jaespina/Celeben>

Donde podras descargarlo.

## **CAPÍTULO 5: PRUEBAS DE SOFTWARE**

---

## **5.1.-INTRODUCCIÓN.**

Las pruebas del software consistirán en cómo reacciona el servidor al recoger un error y como se mostrará al usuario.

## **5.2.- TÉCNICAS DE PRUEBA.**

Las tecnicas de pruebas aplicadas serán:

Pruebas de Caja blanca

Pruebas de Caja negra

### **5.2.1.- PRUEBAS DE CAJA BLANCA O ENFOQUE ESTRUCTURAL.**

Las pruebas de caja blanca se centran en el funcionamiento procedimental del software

### **5.2.2.- PRUEBAS DE CAJA NEGRA O ENFOQUE FUNCIONAL.**

Las pruebas de caja negra consistiran en la comprobacion del correcto funcionamiento de la aplicación web.

Partes de estas pruebas pueden consistir en entrar en páginas a las que no podamos acceder con nuestro rol(como puede ser /sell), o entrar en páginas sin estar logeados(lo cual nos saltará un formulario de Sign In).

Aqui también entrar a la hora de borrar sesiones/cookies/datosde aplicación dentro del navegador. Tras borrarlos, mostrara errores de sesion en lasque nos informara de que no estamos logeados, o al hacer alguna acción en la página se refrescarán.

También se encuentran la comprobacion de tarjetas de creditos, que está manejado mediante Stripe, la cual nos mostrará errores específicos que se mostraran en la página como pueden ser numero de tarjeta invalido/cvc invalido entre muchos otros.

## **CAPÍTULO 6: CONCLUSIONES.**

---



## **6.1.- CONCLUSIONES**

En definitiva, es un e-commerce que trata de acercarse a lo que puede ser WordPress o Magento, pero con otras tecnologías webs mas actuales. Tienen el mismo manejo de productos fácil para el usuario que quiere vender productos y para el usuario normal.

## **6.2.- PROPUESTAS FUTURAS (OPCIONAL).**

Ampliaciones podría tener Oauth con facebook/google(como usar Aliexpress), separar los productos por tags(informatica,limpieza y hogar...),

## CAPÍTULO 7: **BIBLIOGRAFÍA Y REFERENCIAS**

---

## 7.1.-REFERENCIAS BIBLIOGRÁFICAS.

<https://graphql.org/>

<https://nextjs.org/>

<https://es.reactjs.org/>

<https://github.com/downshift-js/downshift>

<https://www.keystonejs.com/>

<https://stripe.com/docs/api>

<https://cloudinary.com/>

<https://www.mongodb.com/es>

## ANEXO 1: MANUAL DE INSTALACIÓN

---

Primero, antes de todo, necesitas tener una version `>13.1` de Nodejs <https://nodejs.org/es/download/>. Para comprobarlo, usa el comando `node -v` en tu terminal.

Tras descargar/clonar el repositorio <https://github.com/Jaespina/Celeben>, abrir una terminal en la carpeta `front`, y ejecutar `npm install`. Y lo mismo con la carpeta `back`.

Una vez hecho esto, debemos ejecutar `npm run dev` tanto en `front` como en `back`.

En la carpeta de `front` necesitas las siguientes variables dentro del archivo `“.env.local”`:

`NEXT_PUBLIC_STRIPE_KEY=“Introducir clave de tu cuenta de Stripe”`

En la carpeta de back, necesitas las siguientes variables dentro del archivo “.env”:

CLOUDINARY\_CLOUD\_NAME=”Nombre de Usuario en cloudinary”

CLOUDINARY\_KEY=”Key de cloudinary”

CLOUDINARY\_SECRET=”Cloudinary Api Secret key”

DATABASE\_URL=”Url de la base de datos”

MAIL\_HOST=”smtp.ethereal.email”

MAIL\_PORT=587

MAIL\_USER=”Cuenta creada en ethereal.mail”

MAIL\_PASS=”Cuenta creada en ethereal.mail”

FRONTEND\_URL=”http://localhost:7777(estando en entorno desarrollo)”

## ANEXO 2: MANUAL DE USUARIO

---

Tras entrar al homepage, encontraras una barra de navegacion.

Desde esta barra puedes acceder a:

Productos: Todos los productos en la página web paginados.

Orders: Los pedidos guardados en tu cuenta

Sell: (Esta opción saldrá dependiendo del rol que tengas en la página web). Vender productos rellenando un formulario.

Account: Información básica de tu cuenta

Sign In: Acceder a tu cuenta

Sign Out: Salir de tu cuenta

My Cart: Carrito de compra.

Luego, encontrarás una barra donde buscar productos. Los controles de esta barra son: flecha arriba y abajo para moverse, esc para salir y enter para ir al producto.

Cada producto será mostrado en una carta, en la que verás el precio, el nombre, una breve descripción, quien vende el producto y añadir al carrito.

Si pulsas en el nombre, te llevará a la página del producto solo.

Una vez añadidos varios productos al carrito, encontraras unos campos donde ingresar tu tarjeta de crédito y un botón donde realizar la compra.

Si ocurre algún error, se detendrá y lo mostrará. Tras una compra exitosa, te mostrará el pedido.