**Laboratorio**

El alcance de este laboratorio es la creación de una aplicación web que permita gestionar ordenes de compra para un negocio.

A nivel de funcionalidades se dividirán en 2 partes.

# **Funcionalidades de cliente**

Los clientes podrán visualizar el listado de productos y agregarlos a su carro de compras, pero deberán iniciar sesión para hacer checkout de la orden.

A su vez solo podrán acceder al listado de órdenes realizadas estando logueados.

## Registro de usuarios

Los usuarios se registrarán en la aplicación utilizando un correo electrónico y contraseña.

A la hora del registro adicionalmente se solicitarán:

* Nombre
* Apellido
* Fecha de nacimiento

Se deberá solicitar confirmación de contraseña.

Se deberá validar que no exista otro usuario con el mismo email.

Al finalizar el registro el usuario deberá ser dirigido a la página de inicio de sesión.

## Inicio de sesión

Los usuarios iniciarán sesión con su email y contraseña.

En caso que los datos sean incorrectos se deberá mostrar un mensaje de error.

Cuando el usuario inicia sesión correctamente debe ser dirigido al listado de productos.

## Listado y búsqueda de productos

Debe existir una página que liste los productos disponibles en formato grilla.

El listado debe tener paginado y mostrar las páginas disponibles.

Los resultados deben poder ser ordenados ascendente y descendentemente:

* Alfabéticamente
* Precio

Deberá existir un buscador por nombre de producto.

Deberá existir un filtro de producto por categoría.

De cada producto de mostrará la siguiente información:

* Nombre
* Precio
* Descripción
* Imagen

El producto debe tener un campo numérico que permita especificar la cantidad requerida y un botón que permita agregar dicha cantidad de productos al carro de compras.

## Generación de una orden mediante un carro de compras

Debe existir una página que permita ver los productos agregados al carro de compras.

Por cada producto existirá un botón para eliminarlo del carro y un campo editable precargado con la cantidad seleccionada.

Se debe mostrar el costo total de la orden el cual debe ser actualizado dinámicamente cuando se remueve un producto o se editan las cantidades.

La página debe tener un botón para concretar la compra el cual mostrará un dialogo al usuario para que confirme su compra. Dicho botón debe estar habilitado solo si existen productos en el carro.

## Listado de órdenes realizadas por un usuario

En esta página el usuario podrá visualizar sus órdenes ordenadas por fecha descendientemente.

Por cada orden se deberá mostrar:

* Estado (Pendiente o Aceptada)
* Fecha y Hora
* Precio Total

# **Funcionalidades del comercio**

## Listado de órdenes

El usuario deberá poder visualizar el listado de ordenes.

Se mostrará:

* Id
* Fecha y Hora
* Precio Total
* Nombre del cliente

## Procesamiento de órdenes

El usuario deberá poder marcar una orden como aceptada ingresando su ID.

# **Componentes de software**

## Base de datos SQL Server:

La base de datos deberá contener las siguientes entidades

* Usuarios
* Productos
* Ordenes

## Backend .Net 5 c#

El backend estará desarrollado en las siguientes capas:

* Capa de acceso a datos (Dapper).
* Capa de negocio
* Web Api que exponga endpoints a ser consumidos por la UI

## UI

La interface de usuario será desarrollada Angular y contendrá las siguientes páginas/Componentes

* Página de Login
* Página de Registro de usuarios
* Página de productos
* Componente producto
* Página de carro de compras
* Página de listado de órdenes
* Componente orden

## Backoffice

El backoffice será como mínimo una aplicación de consola que consuma directamente la capa de negocios del backend.