



Trabajo Práctico 3 -Diseño de Sistemas

Año: 2024

Docentes: Ing. Pablo Pioli

Ing. Juan Pablo Ferreyra

Alumno: Gonzalo Nazzetta

Índice

Enunciado del problema	2
Requerimientos	3
Funcionales	3
No Funcionales	4
Futuros	5
Propuestas tecnológicas	5
Diseño de la arquitectura	6
Especificaciones de la arquitectura	6
Diagrama de Entidad Relación (DER)	6
Caso de uso	7
Diagrama de secuencia	8
Diagrama del container	11
Prototipos:.....	12
Tabla de datos para transferencias	16

Enunciado del problema

Una empresa dedicada a la fabricación de materiales para la construcción se encuentra distribuida en diferentes 3 plantas productivas, una oficina comercial y vendedores que atienden a clientes mayoristas en diferentes zonas.

La sucursal A extrae materia prima que se utiliza como insumo en la planta C. La planta B elabora productos semi-terminados en base a alambres de acero que se utilizan para producir en la planta C. Por su parte, la planta C utiliza elabora ladrillos, vigas de cemento y bloques pre-armados de diferentes medidas. Desde la planta C se realiza el envío de los pedidos directamente al cliente.

Cada planta productiva realiza ingresos de stock de materias primas, consulta de stock, generación de órdenes de producción de los diferentes productos y envío de productos a las diferentes plantas.

Por decisión de la gerencia se necesita reducir los tiempos de atención a clientes minoristas, para ello se pretende ofrecer la posibilidad de cotizar y generar pedidos directamente en el sitio web de la empresa, para ello, una vez identificados los clientes podrán consultar los productos.

Podrán cotizar, ingresando cantidad de metros cuadrados a construir y tipos de materiales, en base a dicha información se debería poder determinar la cantidad de materiales necesarios, por ejemplo: Para construir un galpón de 40m x 40m, de 6m de altura, con ladrillo de tipo bloques de 18cm x 33cm se necesitaría cubrir una superficie de 960 metros cuadrados, con lo cual la cantidad de ladrillos, considerando una separación de 40 cm entre vigas, se necesitaría:

- 16161 ladrillos, equivalentes a 112,23 pallets.
- Importe \$ 6.302.790.-

Se debería poder gestionar los descuentos por cantidad, por ejemplo, a partir de los 10mil ladrillos ofrecer un 5% de descuento sobre el valor del producto. A partir de dicha cotización el cliente podrá realizar un pedido, debiendo completar información de domicilio de envío. La empresa cuenta con servicio de envío. Una vez aprobado el pedido, se acuerda una forma de pago. Una vez que el cliente realiza el pago se envía el pedido.

Requerimientos

Funcionales

- 1RF-Debe permitir registrar un usuario.
- 2RF-Debe permitir loguear un usuario previamente registrado.
- 3RF-Poder seleccionar los productos y ver la información sobre este. Mostrando así el monto con los descuentos aplicados
- 4RF-Se debe poder ingresar las medidas (en metros) y cotizar estas, devolviendo la cantidad de materiales y palets que se necesitan
- 5RF-Debe permitir aplicar descuentos a los productos según las cantidades solicitadas u otros requisitos.
- 5RF-Se debe permitir registrar y solicitar la aprobación de un pedido un pedido.
- 6RF-Se debe poder ver el estado de un pedido (aprobado, no aprobado, cancelado y en espera) y además se debe poder mandar un mail al usuario cuando el estado del pedido cambie.
- 7RF-Se debe poder cancelar un pedido, ya sea que esté aprobado o en espera.
- 8RF-Se debe poder confirmar un pedido aprobado y que se reciba un mail en el caso de que sea aprobado.
- 9RF-Se debe poder ingresar otra dirección en el caso de que así se desee.
- 10RF-Se debe poder poner todos los productos ya cotizados en un carrito y poder pedir la aprobación de este.
- 11RF-Para el caso del pago, se debe conectar con el api de mercado pago y que se realice el pago desde allí.

12RF-Se debe poder buscar productos con su nombre y filtrar según sus características.

13RF-Se debe crear un código único a la hora de crear un pedido nuevo.

14RF-Debe gestionar los pedidos.

No Funcionales

Capacidad de Interacción (Normas ISO 9000 y 25010):

Operabilidad: Debe ser fácil de operar e intuitivo para cualquier empleado del minorista, incluso para usuarios con poca experiencia con el internet.

Protección contra errores de usuario: Debe prevenir los errores del usuario a la hora de ser operado por esto, evitando así fallas.

Fiabilidad (Normas ISO 9000 y 25010):

Ausencia de fallos: El sistema debe ser capaz de llevar a cabo todas las funciones correctamente y sin fallos, bajo condiciones normales de operación.

Disponibilidad: Debe estar activo las 24 horas del día y toda la semana para ser utilizado por los usuarios.

Tolerancia a fallos: El sistema debe ser capaz de adecuarse a un sistema en el caso de que aparezca un fallo relacionado con el hardware y/o ser capaz de operar en presencia de fallos de software.

Capacidad de recuperación: El sistema debe ser capaz de recuperar la información que se tenía anteriormente ante la presencia de un error y restablecer el estado que se tenía antes del error.

Seguridad (Normas ISO 9000 y 25010):

Confidencialidad: Los datos de los usuarios como datos personales, pedidos, entre otros se deben mantener de forma confidencial y solo pueden ser accesibles para usuarios autorizados.

Integridad: Los datos deben estar protegidos en contra de modificaciones no autorizadas y contar con un historial de versiones para los casos de que estos sean vulnerados.

Autenticidad: El acceso al sistema debe estar restringido a usuarios no autenticados. Se debe implementar un sistema de gestión de usuarios con roles y permisos adecuados.

Mantenibilidad (Normas ISO 9000 y 25010):

Modularidad: El sistema debe estar diseñado de forma modular para que sea fácil de actualizar y mantener.

Capacidad para ser probado: Se deben crear pruebas unitarias y de integración para asegurar la calidad del código y facilitar la detección de errores.

Futuros

1RF-Debe permitir que los administradores tengan las funciones de carga de nuevos productos, la creación/modificación/eliminación de ofertas para los productos.

2RF-Debe tener funciones según los roles existentes (Usuario, Administrador).

3RF-El Administrador va a ser el encargado de designar los estados a un pedido según corresponda.

Propuestas tecnológicas

La sección de propuestas tecnológicas del documento no parece estar completamente detallada en las partes visibles, pero podemos inferir algunos componentes claves basados en los requerimientos funcionales y no funcionales:

- **Frontend:** Será la interfaz de usuario a la que los clientes acceden a través de un navegador web. Aquí se detallan varias pantallas (Loguin, cotización, registrar usuario, etc.). Será desarrollada en tecnologías como **React**, **Vue** o **Angular** para proveer una experiencia interactiva y rápida al usuario.
- **Backend:** Procesa las solicitudes desde el frontend. Desarrollada en **Node.js**, **Java** o **Python**. También se menciona la conexión con el API de Mercado Pago para la gestión de pagos.
- **Base de datos:** El sistema debe ser capaz de gestionar usuarios, productos, pedidos, entre otros, Se usará una base de datos relacional.
- **Servicios de autenticación y gestión de roles:** Dado que el sistema gestiona diferentes roles de usuario (usuario y administrador), se podrá implementar un sistema de autenticación seguro con **JWT** (JSON Web Tokens).

Se implementarán 2 API:

La primera es un api de mercado pago que permitirá realizar pagos con tarjetas u online. Esta facilitará los pagos, dándonos así más seguridad ante cualquier intento de fraude u otros.

La segunda es un api que vamos a utilizar para poder contactarse con los clientes via mail. Informando a estos sobre estados de su pedido, información importante y facturas electrónicas.

Diseño de la arquitectura

Especificaciones de la arquitectura

La arquitectura del sistema es de tipo **cliente-servidor**, donde las capas están bien diferenciadas:

- **Cliente (Frontend):** Esta capa es responsable de la interfaz de usuario y las interacciones con el servidor a través de peticiones JSON. Los datos como la cotización de productos, creación de pedidos y visualización de estados se manejarán desde aquí. La comunicación con el backend se hace principalmente a través de llamadas API.
- **Servidor (Backend):** Aquí se maneja toda la lógica de negocio, validación de datos y procesamiento de las solicitudes. El backend se comunica con la base de datos para leer/escribir datos, y con APIs externas (como la de Mercado Pago) para procesar pagos.
- **Base de datos:** Una base de datos separada, que contiene toda la información sobre los productos, usuarios, cotizaciones y pedidos. Esta separación permite una mejor escalabilidad y manejo de los datos.

Se van a aplicar roles dentro del sistema, estos van a ser:

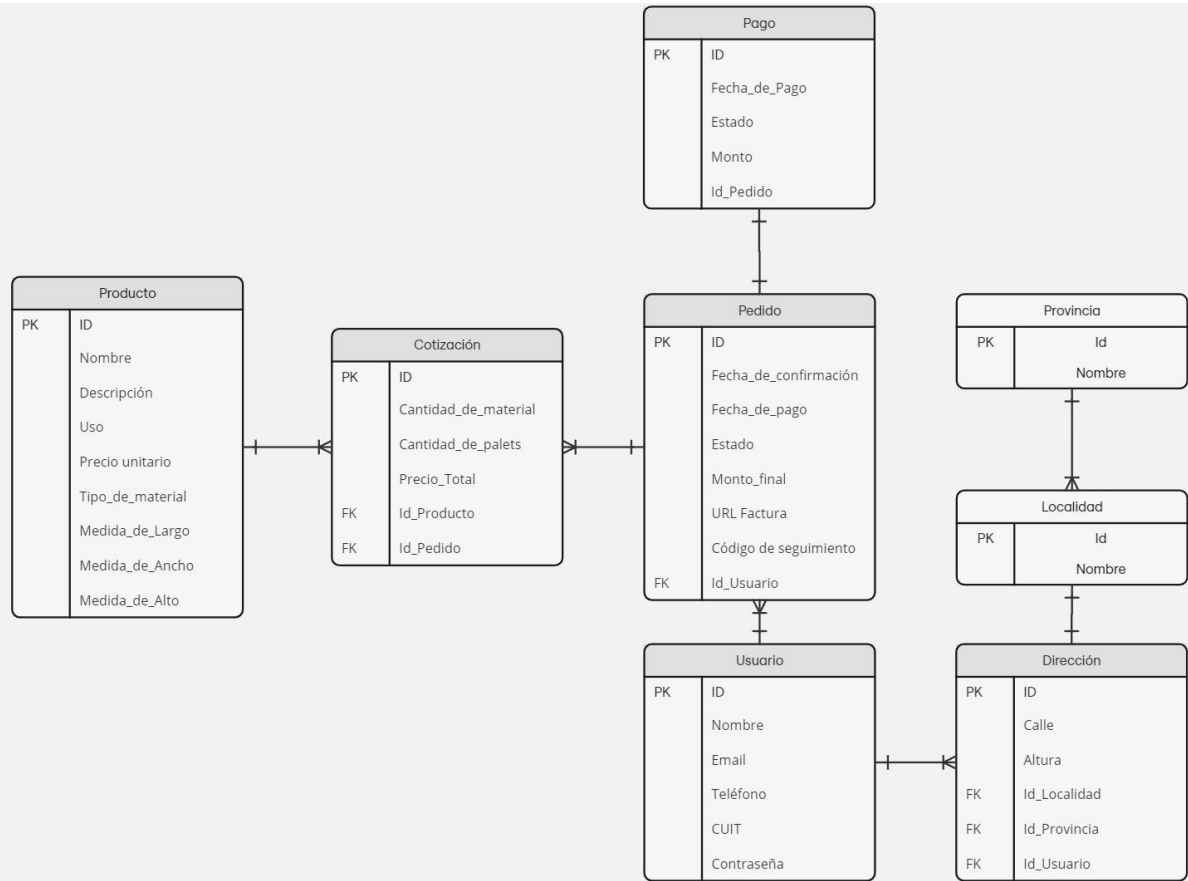
Cliente: es quien va a agregar productos para cotizarlos y quien va a comprar.

Administrador: es quien va a agregar productos al catálogo, actualizar precios, modificar los descuentos y Actualizar el estado de los pedidos.

Otro aspecto que se va a tener en cuenta es el de la validación de los datos ingresados a la hora de cotizar un producto. Se validará la entrada correcta de datos desde que los ingresa en el frontend, como para cuando se envían al backend. Así se realiza correctamente la cotización sin errores y no se ingresa nada fuera de lugar en la base de datos.

Diagrama de Entidad Relación (DER)

Este es el esquema que se va a utilizar en la base de datos, así podremos mantener toda la información ordenada.



Caso de uso

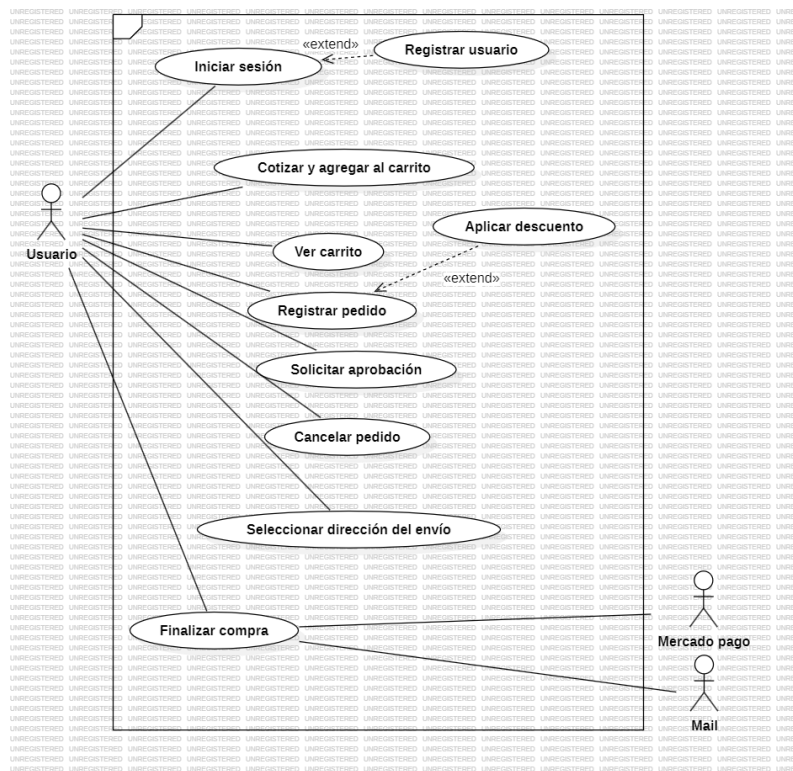
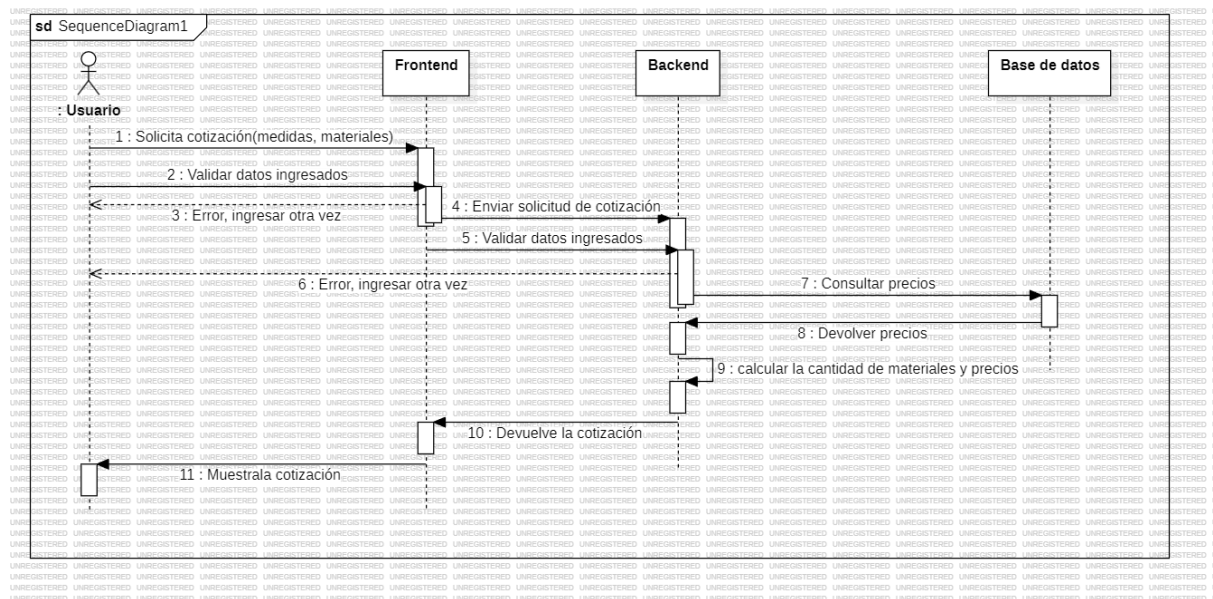


Diagrama de secuencia

El siguiente diagrama de secuencia muestra como se realizaría una cotización paso a paso.



Así quedaría la cotización:

```

def calcular_total_y_descuento(carrito):
    cantidad_total = 0
    precio_total = 0.0
    descuento_aplicado = 0.0

    # Calcular la cantidad total y el precio total para cada producto
    for producto in carrito["productos"]:
        cantidad_total += producto["cantidad"]
        precio_total += producto["subtotal"]

    # Aplicar un 5% de descuento si la cantidad total supera 10,000 unidades
    if cantidad_total >= 10000:
        carrito["descuento"] = 5 # Aplicamos un 5% de descuento
        descuento_aplicado = precio_total * (carrito["descuento"] / 100)
    else:
        carrito["descuento"] = 0 # No se aplica descuento

    # Calcular el precio final después del descuento
    precio_final = precio_total - descuento_aplicado
    return cantidad_total, precio_total, descuento_aplicado, precio_final
    
```


Para manejar el intercambio de datos entre el backend y el frontend se utilizarán paquetes de datos de JSON. Estos tendrán una estructura como la que se mostrará como ejemplo a continuación:

Paquetes del frontend al backend:

```
//Registro de usuario
{
  "nombre": "Juan Pérez",
  "email": "juan.perez@example.com",
  "contraseña": "password123",
  "telefono": "1234567890",
  "cuit": "20-12345678-9",
  "responsabilidad_afip": "Responsable Inscripto"
}

//Cotización del pedido
{
  "productos": [
    {
      "nombre": "Ladrillo Hueco",
      "cantidad": 7000
    },
    {
      "nombre": "Ladrillo Cocido de Tierra",
      "cantidad": 5000
    }
  ]
}

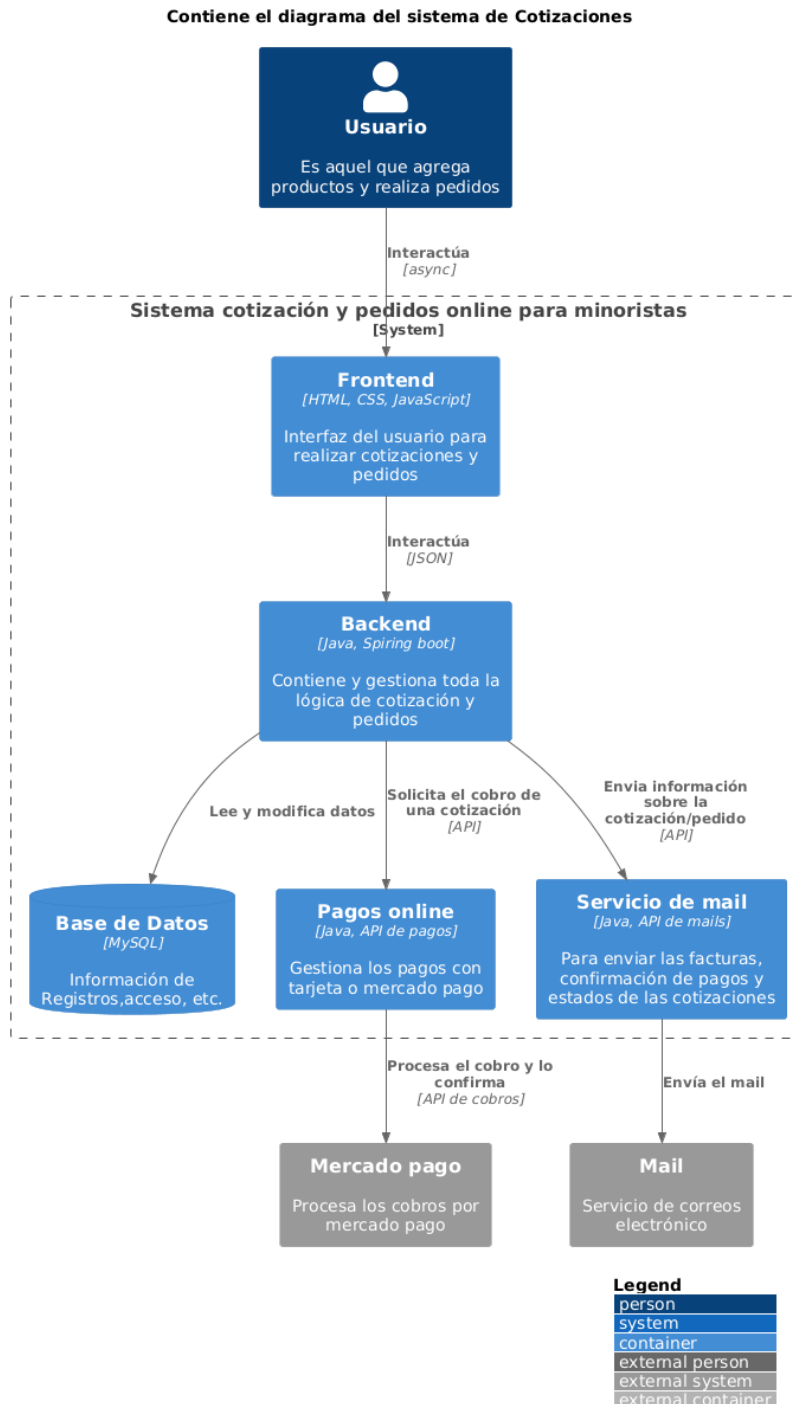
//Solicitud de pago
{
  "id_pedido": 12345,
  "metodo_pago": "Mercado Pago",
  "monto_total": 798000
}
```

Paquetes de datos del backend al frontend

```
//Respuesta de registro exitoso
{
  "mensaje": "Registro exitoso",
  "usuario_id": 456
}

//Resultado de una cotización
{
  "resumen": {
    "productos": [
      {
        "nombre": "Ladrillo Hueco",
        "cantidad": 7000,
        "subtotal": 490000,
        "palets_requeridos": 14
      },
      {
        "nombre": "Ladrillo Cocido de Tierra",
        "cantidad": 5000,
        "subtotal": 350000,
        "palets_requeridos": 10
      }
    ],
    "descuento_total": 42000, // Descuento aplicado
    "precio_total": 840000, // Suma de subtotales
    "precio_final": 798000, // Total después del descuento
    "palets_totales": 24 // Suma total de palets
  }
}
```

Diagrama del container



Prototipos:

Iniciar sesión



Inicio de sesión

Email

nombre@ejemplo.com

Contraseña

.....

[Olvidaste tu contraseña?](#)

Iniciar sesión

Registrar Usuario

LOGO

Registrar usuario

Nombre

Apellido, Nombre

Responsabilidad AFIP

Responsabilidad AFIP

Contraseña

.....

Email

nombre@ejemplo.com

CUIT

xx-xxxxxxx-x

Repetir contraseña

.....

Teléfono

Número de teléfono

Dirección

Calle

Provincia

Altura

Localidad

Registrar Usuario

Nombre del producto





[Iniciar sesión](#)
[Registrarse](#)

Nombre del producto



Descripción:

Uso:

Precio unitario: \$XX [m2]

[Cotizar](#)

Nombre del producto



Descripción:

Uso:

Precio unitario: \$XX [m2]

[Cotizar](#)

Cotización







Nombre del producto



Descripción:

Precio unitario: \$XX [m2]

Ingresar datos para la cotización:

Alto

[Calcular precio final](#)

Largo

Cantidad de ladrillos: XX

Ancho(opcional)

Cantidad de palets: XX

[Agregar al carrito](#)

Carrito

Carrito



Nombre del producto

Cantidad de ladrillos: XX

Cantidad de palets: XX

Eliminar

Modificar

Precio total del producto \$XX



Nombre del producto

Cantidad de Material: XX

Eliminar

Modificar

Precio total del producto \$XX

Monto total: \$XX

Continuar comprando

Confirmación de pago

Pagar

Ingresar datos para pagar

☒ Dirección XXX, Ciudad, Provincia

☒ Ingresar otro domicilio

Confirmar dirección de envío

Domicilio, ciudad, provincia

Seleccionar Domicilio

Monto total: \$XX

Cancelar

Confirmar compra

Cancelar Pedido

¿Está seguro que quiere cancelar su Pedido?

No se eliminará el pedido al cancelar la compra

Aprobación del pedido

Aprobación del pedido

ID: XXXX

Productos:

Nombre

Cantidad de Material: XX
Precio total del producto \$XX

Nombre

Cantidad de Material: XX
Precio total del producto \$XX

Nombre

Cantidad de Material: XX
Precio total del producto \$XX

Estado:....

Solicitar aprobación

En espera
Aprobado
No Aprobado
Cancelado

Se le envía un mail a la oficina para que se revise el pedido y se apruebe si se quiere. Puede ser cancelado por el cliente

Si fue aprobado el pedido se puede pagar o cancelar, en el caso de que no fuera aprobado no aparece ninguna opción y si está en espera solo se puede cancelar

Cancelar

Pagar pedido

Cancelar Pedido

¿Está seguro que quiere cancelar su Pedido?

Al cancelar se eliminará el pedido, independientemente del estado que este tenga

Detalles del envío

Detalles del envío

Factura:

[Ver factura](#)

Descripción:

Número de pedido: XXXX

Domicilio:...

Tu Pedido está: (estado)

Llegará el: DD/MM

El código de seguimiento es:

Detalles de la compra:

Nombre del producto
Cantidad de ladrillos: XX
Cantidad de palets: XX
Precio total del producto \$XX

Nombre del producto
Cantidad de Material: XX
Precio total del producto \$XX

Monto total: \$XX

Tabla de datos para transferencias

Pantalla: Inicio de sesión	
Datos del backend al frontend	
Foto (logo de la empresa)	String
Datos del frontend al backend	
Email	String
Contraseña	String

Pantalla: registrar Usuario	
Datos de entrada	
Foto (logo de la empresa)	String
Datos de salida	
Nombre	String
Email	String
Contraseña	String
Teléfono	String
Responsabilidad AFIP	String
CUIT	String
Contraseña	String
Calle	String
Altura	String
Provincia	String
Localidad	String

Pantalla: Cotización	
Datos del back al front	
Foto (logo de la empresa)	String
Id-Material	Int
Foto (Imagen del material)	String
Nombre-del-material	String
Descripción	String
Precio-unitario	Float
Precio-final	Float
Cantidad-de-ladrillos	Int
Cantidad de palets	Int
Datos del front al back	
Alto	Float
Ancho	Float
Largo	Float
Id-Material	Int

Pantalla: Principal	
Datos del back al front	
Foto (logo de la empresa)	String
Id-Material	Int
Foto (Imagen del material)	String
Nombre-del-material	String
Descripción	String
Precio-unitario	Float
Datos del front al back	
Id-Material	Int

Pantalla: Ver estado del pedido	
Datos del back al front	
Foto (logo de la empresa)	String
Estado del pedido	String
Id-Pedido	Int
Nombre-del-material	String
Precio del total	String
Cantidad de material	Float
Datos del front al back	
Id-Pedido	Int

Pantalla: Detalles del envío	
Datos del back al front	
Foto (logo de la empresa)	String
Factura	String (URL)
Id-Pedido	Int
Nombre-del-material	String
Precio del total por material	Float
Cantidad de material	Float
Cantidad de palets	Int
Monto total	Float
Facha de llegada	String
Estado(del pedido)	String
Código de seguimiento	String
Datos del front al back	
Id-Pedido	Int