

Diseño de Sistemas de Información

INTEGRANTE:

- OSVALDO BARCOS

PROFESORES:

- PABLO PIOLI
- JUAN PABLO FERREYRA

Tabla de contenido

| | |
|-------------------------------------|----|
| Enunciado..... | 2 |
| Requerimientos No Funcionales | 3 |
| Requerimientos Funcionales..... | 4 |
| Prototipos y flujo de datos | 6 |
| Flujo de comunicación: | 6 |
| Registrarte..... | 7 |
| Inicio de Sesión..... | 9 |
| Catálogo | 11 |
| Cotización..... | 13 |
| Carrito de compras..... | 15 |
| Realizar compra..... | 21 |
| Diagrama de Arquitectura..... | 23 |
| Container..... | 23 |
| Diagrama Entidad Relación | 24 |
| Diagrama de Casos de Uso | 25 |

Enunciado

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 pallet.
- 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.

Segunda Iteración

Debido al éxito del sitio web desarrollado, la Gerencia está entusiasmada y desea desarrollar una app mobile que brinde las funciones actuales del sitio web, pero que además permita a los clientes:

- Pagar los pedidos realizados mediante diferentes plataformas de pago, modo, mercadopago, etc.
- Sumar puntos por cada compra realizada, los puntos se definen como una cierta cantidad n de puntos cada x cantidad de pesos gastados. Dicha regla: $n \text{ (puntos)} = x \text{ (pesos)}$ podrá ser modificada por un usuario administrador.
- Implementar una API existente (de otra empresa) que a partir de un plano Autocad devuelve los metros cuadrados de pared y techo necesarios. Esto le permitirá al cliente facilitar la determinación de metros cuadrados de pared y techo, esta información se deberá tomar de insumo en la cotización.

Requerimientos No Funcionales

1. Eficiencia

RNF1: El sistema debe tener un tiempo de respuesta de ≤ 2 segundos para el 95% de las solicitudes realizadas (navegación del catálogo, carrito de compras).

RNF2: El sistema debe soportar al menos 1000 transacciones por segundo durante períodos de alta demanda.

2. Escalabilidad

RNF3: El sistema debe soportar un crecimiento hasta 5000 usuarios concurrentes sin afectar significativamente el rendimiento.

3. Disponibilidad

RNF4: El sistema debe tener una disponibilidad mínima del 99.9%, lo que implica un tiempo de inactividad máximo de 8.77 horas al año.

RNF5: El sistema debe contar con mecanismos de redundancia para evitar la pérdida de información crítica (pedidos, pagos, etc.).

4. Usabilidad

RNF6: Los usuarios deben poder realizar sus primeras cotizaciones en menos de 5 minutos.

5. Seguridad

RNF7: El sistema debe implementar autenticación mediante tokens (JWT) para asegurar que sólo los usuarios autorizados puedan acceder a sus cuentas y datos.

6. Mantenibilidad

RNF8: El código debe estar bien documentado y contar con guías de mantenimiento para que futuros desarrolladores puedan trabajar en él sin dificultades.

7. AutoCAD

RNF9: El sistema debe soportar la carga de archivos AutoCAD de hasta 10 MB en formato .dwg o .dxf.

Requerimientos Funcionales

1. Gestión de usuarios

RF1: Los usuarios deben poder registrarse proporcionando datos como nombre, correo electrónico y contraseña.

RF2: Los usuarios deben poder iniciar sesión utilizando sus credenciales.

RF3: El sistema debe permitir a los usuarios recuperar o restablecer su contraseña en caso de olvido.

2. Carrito de compras y cotización

RF4: El sistema debe permitir a los usuarios visualizar un catálogo de ladrillos disponibles para la construcción. El catálogo debe mostrar el nombre del producto, precio unitario, y otras especificaciones como tipo de ladrillo (hueco, cocido, etc.).

RF5: Los usuarios deben poder buscar productos y filtrar el catálogo según características como el tipo de ladrillo y el precio.

RF6: El sistema debe calcular automáticamente la cantidad de ladrillos necesaria según las dimensiones proporcionadas (largo, ancho, altura) del área a construir.

RF7: El sistema debe mostrar el carrito con los productos seleccionados, la cantidad calculada de ladrillos, el subtotal de cada producto y el total general.

RF8: Si la cantidad total de ladrillos supera un umbral (ej. 10,000 unidades), el sistema debe aplicar un descuento automático (ej. 5%).

3. Método de pago

RF9: El sistema debe permitir realizar pagos a través de la API de Mercado Pago, mostrando opciones de pago como tarjetas de crédito/débito y otros métodos.

RF10: Después de calcular el total del carrito (con o sin descuentos), los usuarios deben poder proceder al pago y obtener una confirmación de este.

4. Entrega a domicilio

RF11: El usuario debe poder introducir o seleccionar su dirección de entrega.

5. autoCAD

RF12: El sistema debe permitir al usuario cargar un archivo AutoCAD (.dwg o .dxf) desde la aplicación móvil o web.

RF13: La aplicación debe procesar el archivo utilizando una API externa que calcule los metros cuadrados de paredes y techos necesarios.

6. Gestión de puntos por compras

RF14: Por cada compra realizada, los usuarios acumularán puntos en función de una regla configurable por el administrador (n puntos por x pesos gastados).

RF15: El sistema debe mostrar los puntos acumulados por cada usuario en su perfil.

RF16: El administrador podrá modificar las reglas de asignación de puntos desde un panel de administración.

Prototipos y flujo de datos

Flujo de comunicación:

1. **El frontend utiliza JavaScript** para crear una solicitud HTTP especificando el método, la URL del endpoint del backend y opcionalmente, un cuerpo de datos en formato JSON.
2. **La solicitud se envía al backend.**
3. **El backend recibe la solicitud**, procesa los datos JSON y realiza las operaciones necesarias.
4. **El backend envía una respuesta al frontend**, generalmente también en formato JSON.
5. **El frontend recibe la respuesta** y la procesa, actualizando la interfaz de usuario o realizando otras acciones según sea necesario.

[Registrarte](#)

Crea tu cuenta

Nombre

name@example.com

Apellido

Email

name@example.com

Password

.....



Número de teléfono

321831283

DNI

45406108

Sign In

Sign Up

| Paquete de datos (Front al back) |
|----------------------------------|
| nombre_apellido: string |
| email: string |
| contraseña: string |
| numero: string |
| dni: string |

-En caso de éxito:

```
1 {  
2   "mensaje": "Registro exitoso",  
3   "usuario_id": 456  
4 }
```

-En caso de error (por ejemplo, si el email ya existe):

```
{  
  "error": "El correo electrónico ya está registrado"  
}
```

Inicio de Sesión

Iniciar Sesión

Email

Password

[Sign In](#)

[Forgot Password?](#) [Registrarse](#)

| Paquete de datos(FRONT AL BACK) | |
|---------------------------------|--------|
| email: | string |
| contraseña: | string |

En caso de éxito:

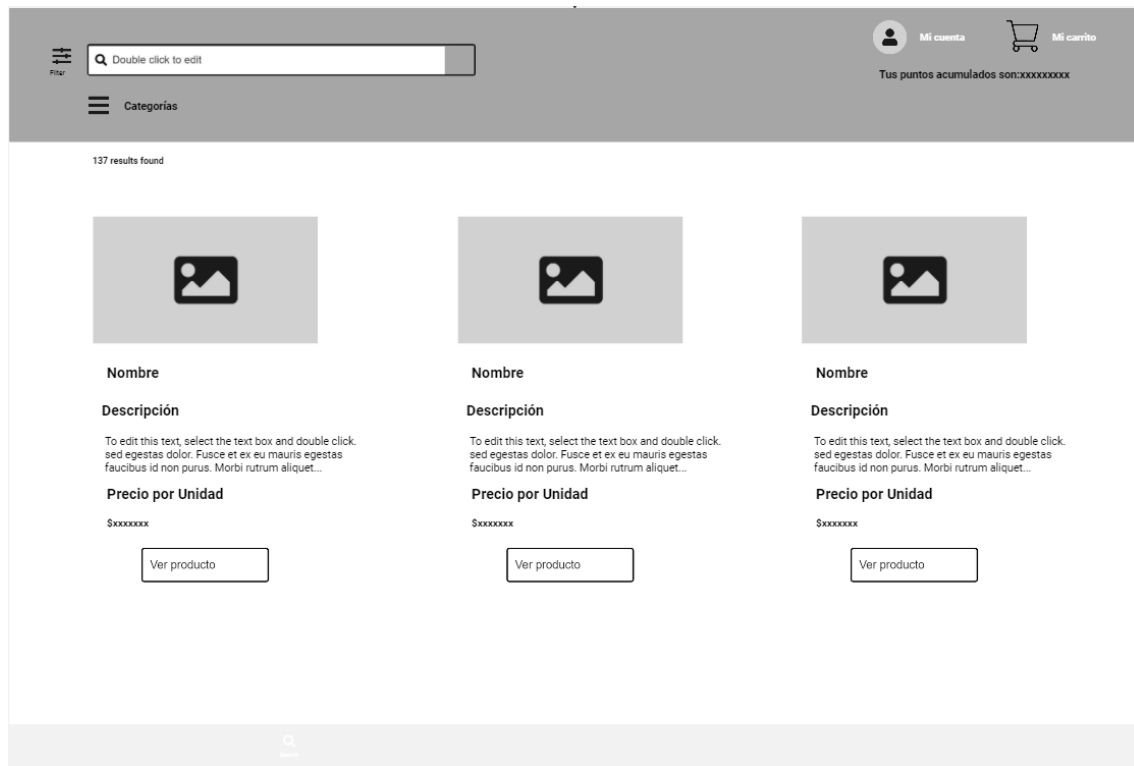
```
1 {  
2   "mensaje": "Inicio de sesión exitoso",  
3   "token": "eyJhbGciOiJIc2ciOiJ0eXAd3QilCjZxhwIfQ.e30mEX4P1r2r1J1pB_v10RLdX0C18K8",  
4   "usuario": {  
5     "id": 123,  
6     "nombre": "Juan Perez"  
7   }  
8 }
```

El **token** es el que el frontend utilizará para **mantener la sesión abierta** del usuario. Este token debe ser almacenado en el navegador (en localStorage, sessionStorage o una **cookie segura**).

En caso de error:

```
1 {  
2   "error": "Correo o contraseña incorrectos"  
3 }
```

Catálogo

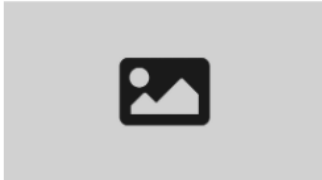


| Paquete de datos(BACK AL FRONT) |
|---------------------------------|
| ID_producto: string |
| nombre: string |
| descripcion: string |
| precio_unitario: float |
| imagen_url: string |
| largo_ladrillo: float |
| ancho_ladrillo: float |
| altura_ladrillo: float |

Cotización

Pantalla cuando se presiona en el botón ver producto

Ingrese su archivo AutoCAD aquí



Descripción

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Nombre

Precio por Unidad

\$xxxxxxx

Cotizar

Datos enviados cuando se agrega al carrito el producto con las medidas ingresadas manualmente.

| Paquete de datos (FRONT AL BACK) |
|----------------------------------|
| ID_producto: string |
| largo: float |
| ancho: float |
| altura: float |

Datos enviados cuando se carga el archivo autoCAD:

| |
|----------------------------------|
| Paquete de datos (FRONT AL BACK) |
| ID_producto: string |
| archivo:multipart/form-data |

Procesamiento de los datos en el backend:

- El backend recibe el archivo, valida nuevamente el tipo y tamaño.
- El backend invoca un endpoint de la API de AutoCAD para procesar el archivo:

```
POST /calcular-m2
Content-Type: application/json
{
  "archivo": "base64_encoded_file_data"
}
```

- La API devuelve los metros cuadrados de pared y techo necesarios:

```
{
  "largo": 120.5,
  "ancho": 80.2,
  "altura": 7
}
```

- Los datos obtenidos se almacenan en la base de datos junto con el archivo cargado.

Carrito de compras

Cotización/Carrito de compras

| | |
|--|--|
| <p>Nombre : Ladrillo hueco</p> <p>Cant. ladrillo: 7000</p> <p>Subtotal: \$490.000</p> | <p>Resumen de compra</p> <p>Cant. Total: 12000</p> <p>Precio Total: \$840.000</p> <p>Descuento: 5%</p> <p>Precio final: 798.000</p> <p>Puntos por esta compra: 798000</p> |
| <p>Nombre : Ladrillo cocido de tierra</p> <p>Cant. ladrillo: 5000</p> <p>Subtotal: \$350.000</p> | |

Agregar más productos



- **Cotización de cada producto cuando se agrega al carrito de compras.**

| Paquete de datos recibidos(BACK AL FRONT) |
|---|
| ID_producto: string |
| cantidad: int |
| subtotal: float |

- **Cálculos para obtener los datos:**

-perimetro= $2 \times (\text{Largo} + \text{Ancho}) = 2 \times (40 + 40) = 160$ (m)

-A_total = perimetro x altura = 160 (m) \times 6 (m) = 960 (m²)

-A_ladrillo = largo_ladrillo x altura = $0,0594$ (m²)

-cantidad = A_total/A_ladrillo= 960 (m²) / $0,0594$ (m²) = 16.161 ladrillos

-subtotal = cantidad x precio_unitario = $16.161 \times 390 = 6.302.790$

- Datos del resumen de compra.

| Paquete de datos recibidos(BACK AL FRONT) |
|---|
| cantidad_total: int |
| descuento_total: float |
| precio_total: float |
| precio_final |
| puntos: int |

- Cálculos para obtener los datos:

```
class SistemaPuntos:
    def __init__(self, x_pesos, n_puntos):
        """
        Inicializa la regla para calcular puntos.
        :param x_pesos: Cantidad de pesos que equivale a un bloque de puntos.
        :param n_puntos: Cantidad de puntos otorgados por cada x_pesos.
        """
        self.x_pesos = x_pesos
        self.n_puntos = n_puntos

    def calcular_puntos(self, monto_compra):
        """
        Calcula los puntos obtenidos por una compra.
        :param monto_compra: Monto total de la compra.
        :return: Puntos acumulados para el monto de compra.
        """
        if monto_compra <= 0:
            return 0 # No se otorgan puntos para montos no válidos.

        # Calcular los puntos como proporción del monto de compra
        puntos = (monto_compra // self.x_pesos) * self.n_puntos
        return int(puntos)
```

```
def calcular_total_y_descuento(carrito):
    cantidad_total = 0
    precio_total = 0.0 # Renombrado desde subtotal_total
    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"] # Renombrado desde subtotal_total

    # 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

    # Configuración del sistema de puntos
    sistema_puntos = SistemaPuntos(x_pesos=100, n_puntos=10)
    puntos_ganados = sistema_puntos.calcular_puntos(precio_final)

    return {
        "precio_final": precio_final,
        "descuento_aplicado": descuento_aplicado,
        "puntos_ganados": puntos_ganados,
    }

# Ejemplo de uso
carrito = {
    "productos": [
        {"cantidad": 5000, "subtotal": 25000},
        {"cantidad": 6000, "subtotal": 36000},
    ]
}

resultado = calcular_total_y_descuento(carrito)
print(f"Precio final: {resultado['precio_final']}")
print(f"Descuento aplicado: {resultado['descuento_aplicado']}")
print(f"Puntos ganados: {resultado['puntos_ganados']}")
```

-JSON de ejemplo

- Envío de todos los datos calculados en el backend.

```
{
  "resumen": {
    "productos": [
      {
        "nombre": "Ladrillo Hueco",
        "cantidad": 7000,
        "subtotal": 490000
      },
      {
        "nombre": "Ladrillo Cocido de Tierra",
        "cantidad": 5000,
        "subtotal": 350000
      }
    ],
    "descuento_total": 42000.0,
    "precio_total": 840000.0,
    "precio_final": 798000.0,
    "puntos_ganados": 7980
  }
}
```

Realizar compra

-Cargar datos del domicilio para la entrega

 Realizar compra

 Envío a domicilio

Provincia:
 Ciudad:
 Código postal:
 Dirección:


Cancelar
 Continuar compra

| Paquete de datos enviados(FRONT AL BACK) |
|--|
| provincia: string |
| ciudad: string |
| codigo_postal: string |
| direccion: string |
| ID_cotización: string |

-Seleccionar método de pago (por ahora solo tarjetas)



- Integración de la API checkout Pro de Mercado pago.



Checkout Pro Web y Mobile

Checkout Pro se puede abrir tanto vía Desktop como Mobile (smartphones y tablets). Para dispositivos móviles, el comprador es redirigido automáticamente a la aplicación de Mercado Pago (cuando esta se encuentra instalada), lo que garantiza una mayor agilidad y facilidad de pago, ya que no es necesario completar formularios.

miro

Diagrama de Arquitectura

Container

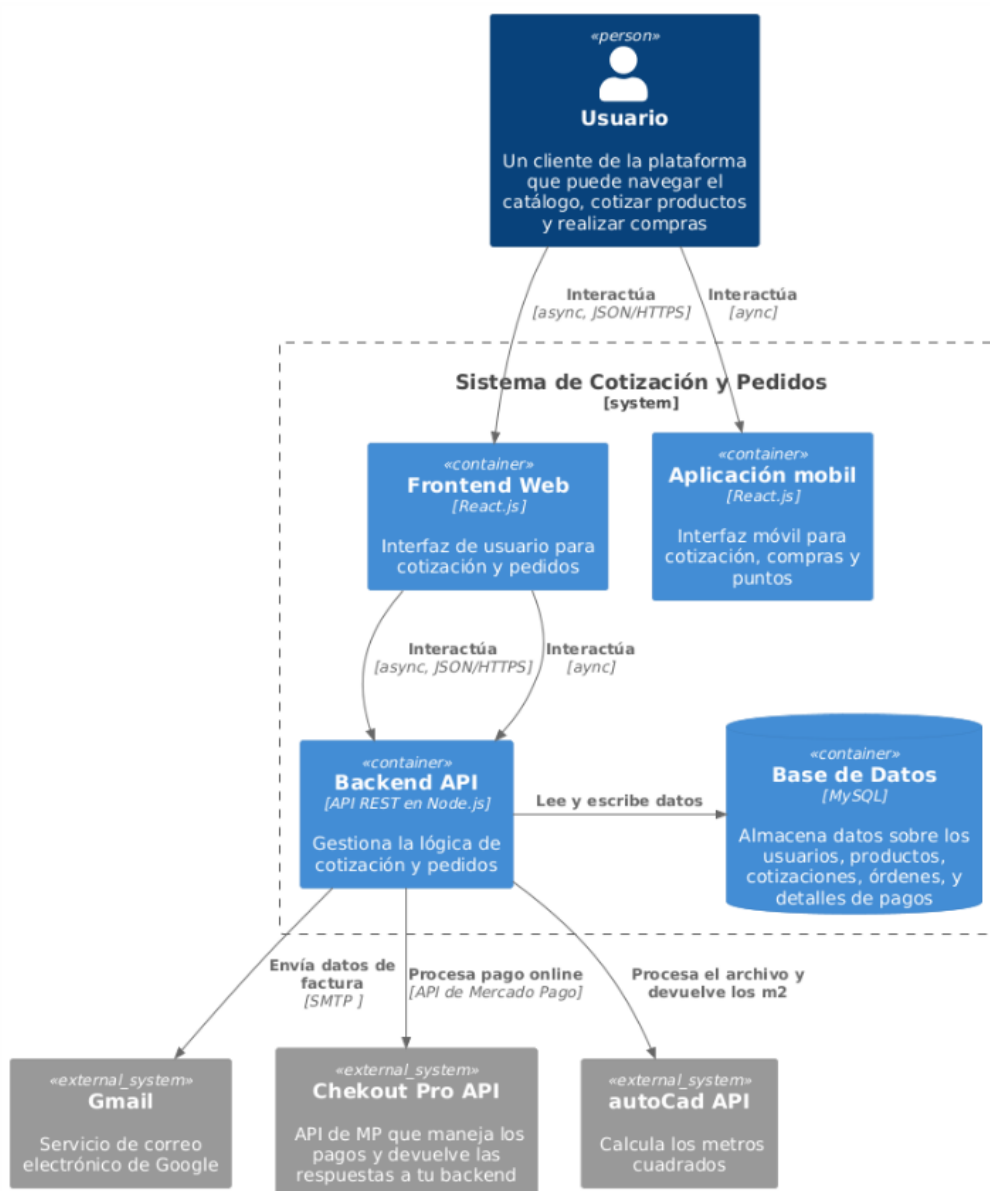


Diagrama Entidad Relación

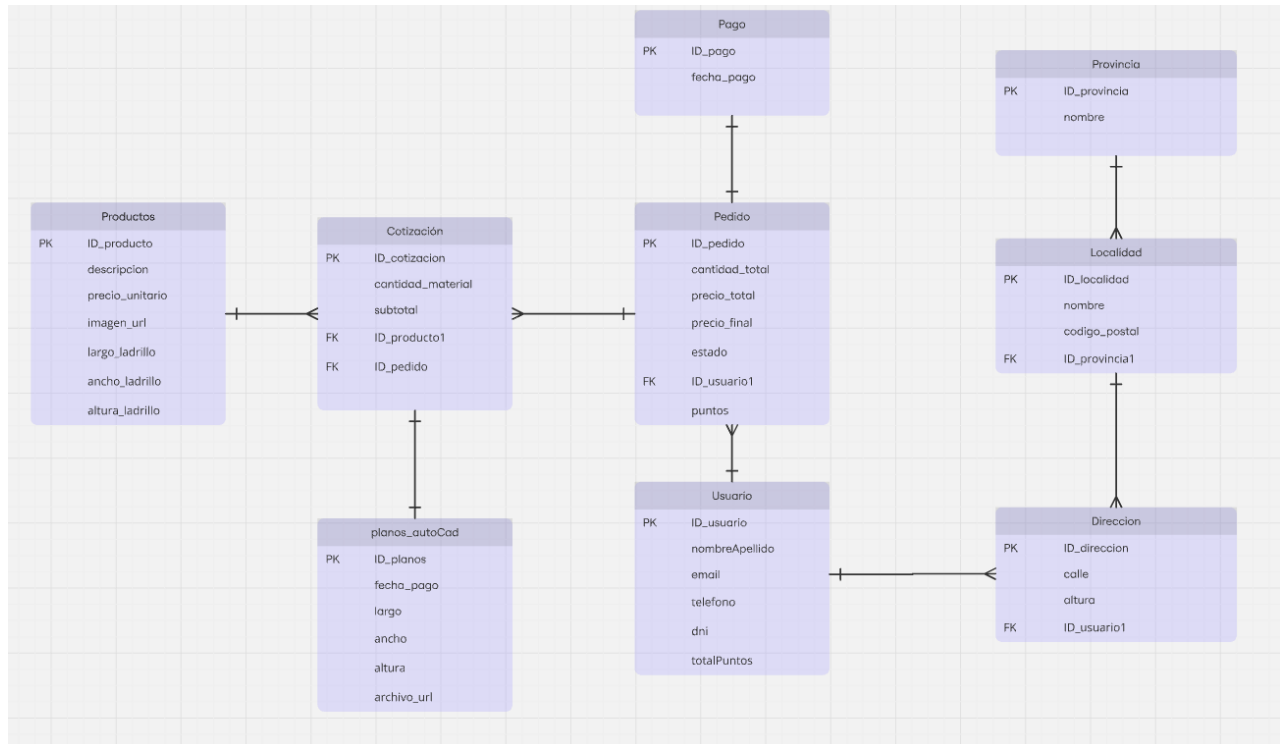
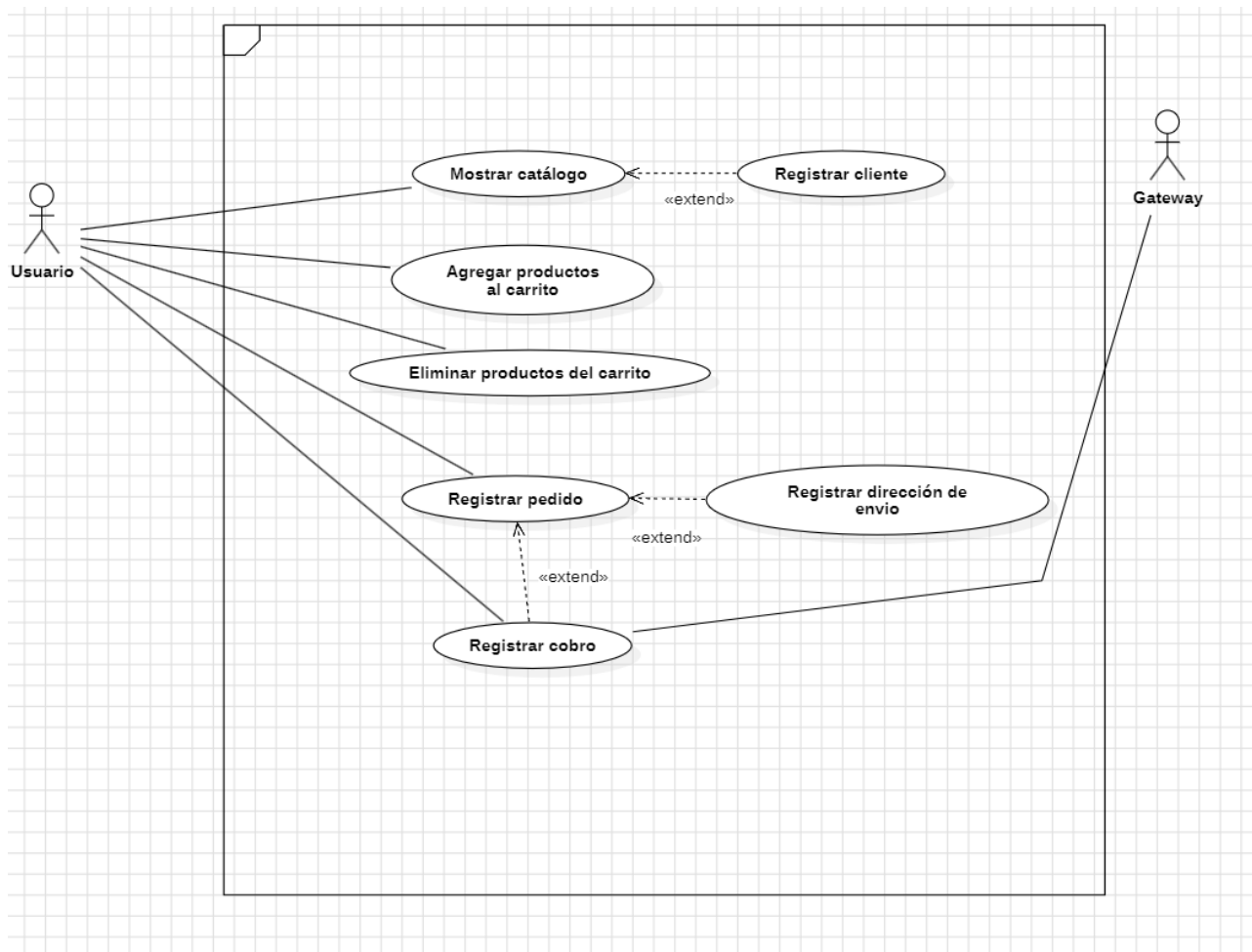


Diagrama de Casos de Uso



Integración App Mobile

Registrarte

Crea tu cuenta

Nombre

name@example.com

Apellido

Email

name@example.com

Password



Número de teléfono

321831283

DNI

45406108

Sign In

Sign Up

Inicio de Sesión

Iniciar Sesión

Email

name@example.com

Password



Sign In

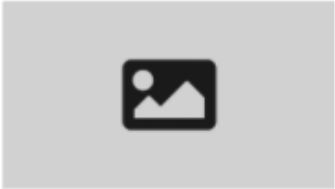
[Forgot Password?](#)

[Registrarse](#)

Catálogo

Categorías

137 results found



Nombre

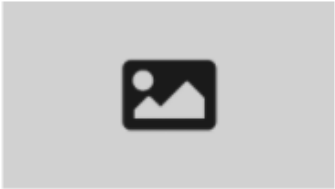
Descripción

To edit this text, select the text box and double click.
sed egestas dolor. Fusce et ex eu mauris egestas
faucibus id non purus. Morbi rutrum aliquet...

Precio por Unidad

\$xxxxxxx

Ver producto



Nombre

Descripción


To edit this text, select the text box and double click.
sed egestas dolor. Fusce et ex eu mauris egestas
faucibus id non purus. Morbi rutrum aliquet...

Precio por Unidad

\$xxxxxxx

Cotización

Ingrese su archivo
AutoCAD aquí



Nombre

Precio por Unidad
\$xxxxxxx

Cotizar

Agregar al carrito

Descripción

Lorem ipsum dolor sit amet, consectetur adipiscing
 elit, sed do eiusmod tempor incididunt ut labore et
 dolore magna aliqua. Ut enim ad minim veniam, quis
 nostrud exercitation ullamco laboris nisi ut aliquip ex
 ea com

Carrito de compras

Cotización/Carrito de compras

Nombre : Ladrillo hueco

Cant. ladrillo: 7000

Subtotal: \$490.000

Nombre : Ladrillo cocido de tierra

Cant. ladrillo: 5000

Subtotal: \$350.000

Resumen de compra

Cant. Total: 12000

Precio Total: \$840.000

Descuento: 5%

Precio final: 798.000

Puntos por esta compra: 798000

Agregar más productos



Realizar compra



Realizar compra



Envío a domicilio

Provincia:

Ciudad:

Código postal:

Dirección:

Cancelar

Continuar compra

Finalizar compra



Realizar compra



Elegir método de
pago



Tarjeta de crédito o débito (API
mercadopago)

Cancelar

Finalizar compra