

# UTN FRSFCO

---

## DISEÑO DE SISTEMAS DE INF.

# TP Nº3



ARROYO SANTINO

PROFESOR:  
PABLO PIOLI  
JUAN PABLO FERREYRA



# Índice

Enunciado del Problema.....	3
Propuesta Tecnológica.....	4
Diseño de Arquitectura.....	6
Diagrama de Container.....	6
Diagrama de Entidad - Relación.....	7
Requerimientos Específicos.....	8
Requerimientos Futuros:.....	8
Requerimientos Funcionales:.....	9
Requerimientos No Funcionales:.....	11
Diagrama de Casos de Uso.....	13
Diagramas de Secuencia.....	14
Cotización.....	14
Validación.....	16
Tabla de Datos.....	17
Interfaces.....	20



## 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.



## Propuesta Tecnológica

El nombre del sistema será “ConstruWeb” y estará disponible como página web, tanto en la versión de pc como en la versión para smartphones. El mismo servirá para poder comercializar materiales de construcción a clientes minoristas, el sistema contendrá:

- **Catálogo de Materiales:** Muestra una lista de materiales organizados por categoría (por ejemplo, “Ladrillos”).
- **Creación de Artículos Comercializables:** Los administradores pueden crear nuevos artículos y categorías.
- **Gestión de Cotizaciones:** El sistema deberá poder calcular cotizaciones y cantidad de materiales con medidas de alto y largo ingresadas por el cliente.
- **Gestión de Clientes:** Los clientes van a poder registrarse y la empresa gestionar sus usuarios.
- **Gestión de Cobros:** La plataforma permitirá calcular un presupuesto final (contemplando las situaciones de descuentos) y le permitirá a la empresa cobrarle a los clientes.
- **Gestión de Puntos:** La plataforma contará con un sistema de puntos el cuál tendrá la métrica  $\rightarrow 100.000\$ = 1000\text{pts}$ . Se podrán utilizar para solicitar descuentos o para canjear distintas cosas.

Los usuarios que van a interactuar con el sistema son:

- **Administrador:** Éste se encargará de realizar la carga de materiales/categorías e interactuar con las funcionalidades del sistema.
- **Cliente:** Cliente virtual que comprará los productos y consultará por cotizaciones.



### Arquitectura de la Plataforma:

- **Frontend:** Utilizaremos React.js para crear una interfaz de usuario amigable y receptiva. Esto permitirá a las empresas y empleados acceder a la plataforma desde cualquier dispositivo.
- **Backend:** Python para desarrollar la lógica de negocio y la gestión de datos, utilizando Django como framework.
- **Base de Datos:** Base de datos relacional (MySQL) para almacenar información sobre empresas, empleados, capacitaciones y progreso.
- **Autenticación y Seguridad:** Autenticación segura mediante tokens JWT (JSON Web Tokens) para que los usuarios puedan iniciar sesión y acceder a sus datos de manera segura. Además de que permite validar al usuario para no hacer el login cada vez que se ingresa.

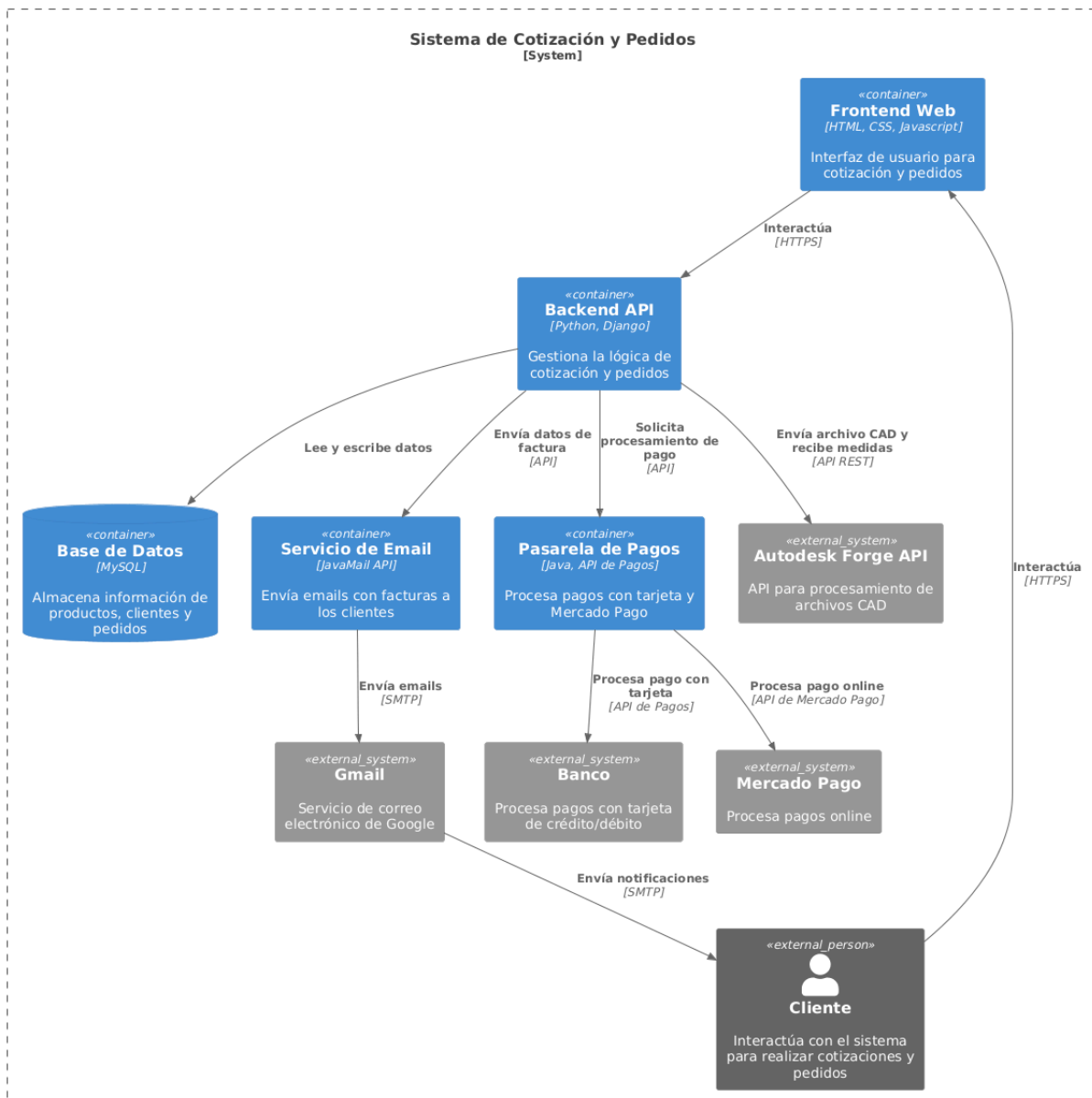
### Comunicaciones y Notificaciones

- **Correo Electrónico:** Google como servicio de envío de correo electrónico para enviar ofertas y/o facturas de cobro.
- **Mensajes Internos:** Sistema de mensajería interno para que los clientes puedan comunicarse con los empleados directamente desde la plataforma.



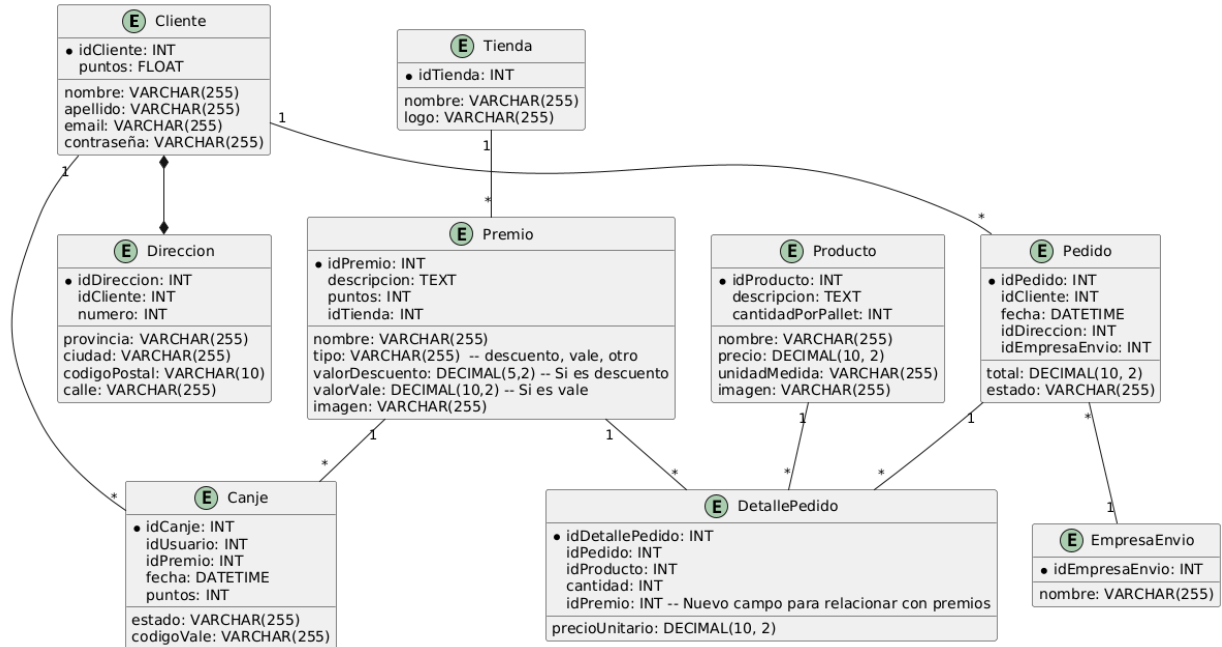
## Diseño de Arquitectura

### Diagrama de Container





## Diagrama de Entidad - Relación





## Requerimientos Específicos

### Requerimientos Futuros:

RF1 - Implementar un sistema de recomendaciones que sugiera a los clientes productos complementarios o alternativos en base a su historial de compras y a las preferencias de otros clientes.

RF2 - Ofrecer recomendaciones personalizadas de materiales en base a las dimensiones del área a construir y al tipo de proyecto.

RF3 - Implementar un sistema de notificaciones para informar a los clientes sobre el estado de sus pedidos, nuevas ofertas y promociones.

RF4 - Integrar herramientas de diseño que permitan a los clientes visualizar el proyecto en 3D y realizar simulaciones de construcción.

RF5 - Generar informes detallados sobre las ventas, los productos más vendidos, los clientes más activos y otros indicadores clave de rendimiento.

RF5 - Implementar herramientas de análisis de datos para identificar patrones de compra, tendencias de mercado y oportunidades de mejora.





## Requerimientos Funcionales:

RF1 - El sistema debe permitir a los clientes ingresar las dimensiones del área a construir (metros cuadrados).

RF2 - El sistema debe permitir a los clientes seleccionar los tipos de materiales deseados.

RF3 - El sistema debe calcular la cantidad de materiales necesarios en base a las dimensiones y tipos de materiales seleccionados.

RF4 - El sistema debe mostrar el costo total de los materiales, incluyendo descuentos por cantidad.

RF5 - El sistema debe permitir a los clientes registrados realizar pedidos en base a una cotización previa.

RF6 - El sistema debe permitir a los clientes ingresar la información de envío (dirección, etc.).

RF7 - El sistema debe permitir a los clientes seleccionar la forma de pago (tarjeta de crédito/débito, Mercado Pago).

RF8 - El sistema debe generar un número de pedido único para cada pedido realizado.

RF9 - El sistema debe enviar una confirmación del pedido al cliente por email, incluyendo la factura.

RF10 - El sistema debe aplicar descuentos por cantidad según la política definida (ej: 5% de descuento por más de 10mil ladrillos).

RF11 - El sistema debe procesar los pagos online a través de Mercado Pago.

RF12 - El sistema debe gestionar el envío de los pedidos a los clientes.

RF13 - El sistema debe proporcionar información de seguimiento del envío al cliente.

RF14 - El sistema debe ajustar automáticamente el diseño y tamaño de los elementos para pantallas de distintos tamaños, proporcionando una experiencia de usuario óptima en dispositivos móviles.

RF15 - En la versión móvil, el menú debe adaptarse a un diseño más compacto.

RF16 - Los botones deben ser lo suficientemente grandes y tener el espaciado adecuado para evitar toques accidentales, mejorando la accesibilidad en pantallas táctiles.



- Juan Pablo Ferreyra.
- Pablo Pioli

RF17 - La interfaz de pago en la versión móvil debe estar optimizada para facilitar el ingreso de datos de tarjetas y la integración con servicios como Mercado Pago mediante botones grandes y accesibles.

RF18 - El sistema deberá permitir la carga de archivos de AutoCad y deberá conectarse con la api de AutoDesk Forge para obtener las medidas para la cotización.



## Requerimientos No Funcionales:

### 1. Confiabilidad (ISO 9001 e ISO 25010):

RNF1 - Disponibilidad: El sistema debe estar disponible para los clientes minoristas las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad mínimo.

RNF2 - Tolerancia a fallos: El sistema debe ser capaz de manejar errores y fallos de manera eficiente, minimizando el impacto en los usuarios y la pérdida de datos. Se deben implementar mecanismos de recuperación y respaldo para asegurar la continuidad del servicio.

RNF3 - Recuperabilidad: En caso de fallos o interrupciones, el sistema debe ser capaz de recuperar los datos y volver a un estado operativo de manera rápida y eficiente.

### 2. Seguridad (ISO 9001 e ISO 25010):

RNF4 - Confidencialidad: La información de los clientes, incluyendo datos personales, de pago y de pedidos, debe ser confidencial y solo accesible para personal autorizado.

RNF5 - Integridad: Los datos del sistema deben estar protegidos contra modificaciones no autorizadas. Se deben implementar mecanismos para asegurar la integridad de los datos, como registros de auditoría y control de versiones.

RNF6 - Autenticación: El acceso al sistema debe estar restringido a usuarios autenticados. Se debe implementar un sistema de gestión de usuarios con roles y permisos adecuados.

### 3. Usabilidad (ISO 9001 e ISO 25010):

RNF7 - Facilidad de aprendizaje: La interfaz del sitio web debe ser intuitiva y fácil de usar para los clientes minoristas, incluso para aquellos con poca experiencia en el uso de internet.

RNF8 - Eficiencia: El sistema debe permitir a los usuarios realizar las tareas de forma eficiente, con un mínimo de pasos y clics.

RNF9 - Protección contra errores: El sistema debe guiar al usuario y prevenir errores comunes, por ejemplo, mediante validaciones de datos y mensajes de error claros.



#### 4. Mantenibilidad (ISO 9001 e ISO 25010):

RNF10 - Modularidad: El sistema debe estar diseñado de forma modular para facilitar el mantenimiento y las actualizaciones.

RNF11 - Testabilidad: Se deben crear pruebas unitarias y de integración para asegurar la calidad del código y facilitar la detección de errores.

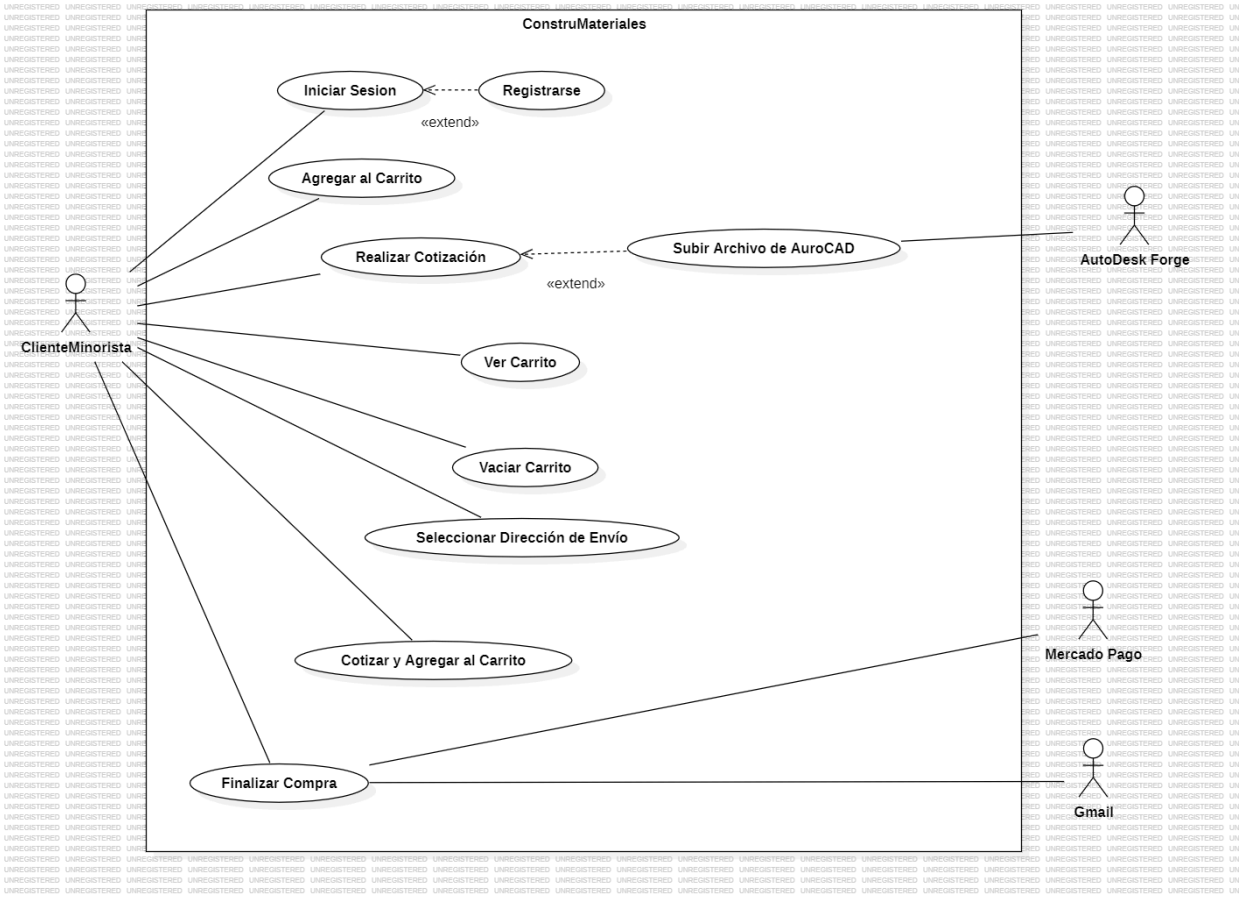
#### 5. Rendimiento (ISO 25010):

RNF12 - Tiempo de respuesta: El sistema debe responder de forma rápida a las solicitudes de los usuarios, con tiempos de carga de página y de procesamiento de datos optimizados.

RNF13 - Escalabilidad: El sistema debe ser capaz de manejar un aumento en el número de usuarios y de transacciones sin afectar negativamente al rendimiento.



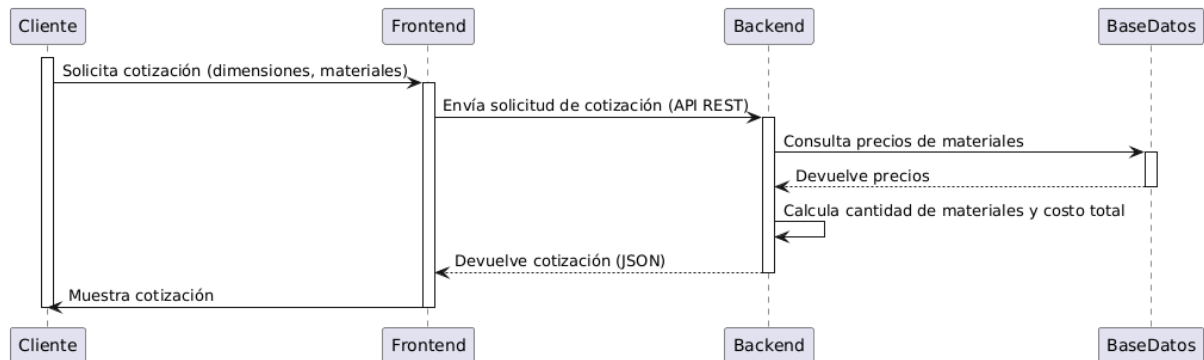
# Diagrama de Casos de Uso





## Diagramas de Secuencia

### Cotización



Ejemplo de los datos de la solicitud:

```
[
  {
    "materialId": 1, / ID del material (Ladrillo hueco)
    "alto": 0.18, double / en metros
    "largo": 0.33, double / en metros
    "ancho": 0.12, double / en metros
  },
  {
    "materialId": 2, / ID del material (Viga 4 mts)
    "alto": 0.20, Double / en metros
    "largo": 4.00, Double / en metros
    "ancho": 0.15, Double / en metros
  }
]
```



### Ejemplo de respuesta

```
{  
  "costoTotal": 12345.67, Double  
  "descuento": 500, Double  
  "materiales": [  
    {  
      "materialId": 1,  
      "nombre": "Ladrillo Hueco",  
      "cantidad": 1000,  
      "costo": 5000  
    },  
    {  
      "materialId": 2,  
      "nombre": "Viga 4 mts",  
      "cantidad": 20,  
      "costo": 7345.67  
    }  
  ]  
}
```



- Juan Pablo Ferreyra.
- Pablo Pioli

## Validación

Se tiene que realizar tanto del lado del cliente como del lado del servidor, ya que al validar del lado del cliente, se puede mandar una respuesta automática para que el cliente sepa que mandó los datos bien. Del lado del servidor se validaría por consiguiente ya que permitiría tener mucha más seguridad y consistencia acerca de los datos recibidos.





## Tabla de Datos

Dato	Tipo	Descripción	Ejemplo JSON
Productos	Array de objetos	Información de cada producto disponible para la cotización y venta.	[{"id": 1, "nombre": "Ladrillo Hueco", "descripcion": "Ladrillo hueco cerámico 12x18x33 cm 9 tubos", "precio": 390, "unidadMedida": "unidad", "cantidadPorPallet": 144, "imagen": "url_imagen"}, {"id": 2, "nombre": "Viga 4 mts", "descripcion": "...", "precio": 10619, "unidadMedida": "unidad", "cantidadPorPallet": 1, "imagen": "url_imagen"}, ... ]
Clientes	Array de objetos	Información de los clientes registrados en el sistema.	[{"id": 1, "nombre": "Juan Perez", "apellido": "Gomez", "email": "juan.perez@example.com", "contraseña": "password123",



- Juan Pablo Ferreyra.
- Pablo Pioli

			“puntos”: 100.0}, ... ]
Direcciones	Array de objetos	Información de las direcciones de los clientes.	[{"id": 1, "idCliente": 1, "provincia": "Buenos Aires", "ciudad": "CABA", "codigoPostal": "1425", "calle": "Calle Falsa", "numero": 123}, ... ]
Pedidos	Array de objetos	Información de los pedidos realizados por los clientes.	[{"id": 1, "idCliente": 1, "fecha": "2023-10-26T10:00:00", "idDireccion": 1, "total": 1234.56, "estado": "Pendiente", "idEmpresaEnvio": 2}, ... ]
Detalle Pedido	Array de objetos	Información de los productos incluidos en cada pedido.	[{"id": 1, "idPedido": 1, "idProducto": 2, "cantidad": 5, "precioUnitario": 10619.00}, {"id": 2, "idPedido": 1, "idProducto": 1, "cantidad": 20, "precioUnitario": 390.00}, ... ]
Envío	Array de objetos	Información de las empresas de envío	[{"id": 1, "nombre": "Correo Argentino"},



		disponibles.	{ "id": 2, "nombre": "Andreani"}, { "id": 3, "nombre": "OCA" }
Tiendas	Array de objetos	Información de las tiendas que participan en el sistema de canjes.	[ { "id": 1, "nombre": "Tienda A", "logo": "url_logo_tienda_a"}, { "id": 2, "nombre": "Tienda B", "logo": "url_logo_tienda_b" } ]
Premios	Array de objetos	Información de los premios disponibles para canjear.	[ { "id": 1, "nombre": "Cafetera", "descripcion": "Cafetera eléctrica de 12 tazas", "puntos": 1000, "tiendaId": 1, "imagen": "url_imagen_cafetera"}, { "id": 2, "nombre": "Parlante Bluetooth", "descripcion": "...", "puntos": 750, "tiendaId": 2, "imagen": "url_imagen_parlante" } ]
Canjes	Array de objetos	Información sobre los canjes realizados por los usuarios.	[ { "id": 1, "usuarioId": 1, "premioId": 1, "fecha": "2024-07-26",



			<pre>"puntos":          1000, "estado": "completado"}, {"id": 2,  "usuarioId":    2, "premioId": 2, "fecha": "2024-07-27", "puntos":          750, "estado": "pendiente"}]</pre>
--	--	--	--



## Interfaces

Link de la página (está adaptado tanto para web como para la versión mobile):

[https://santinoarroyo.github.io/TP3\\_DSI/](https://santinoarroyo.github.io/TP3_DSI/)

Para ir hasta el carrito, se deben añadir productos e ir hasta el final de la página, ahí estará el botón para ir al carrito.

### Interfaz de Inicio

Interacción	Nombre	Tipo	Descripción
<b>Frontend -&gt; Backend</b>			
Buscar Productos	busqueda	string	Término de búsqueda para filtrar productos
Agregar al Carrito	idProducto	int	ID del producto a agregar
Agregar al Carrito	cantidad	int	Cantidad del producto a agregar
<b>Backend -&gt; Frontend</b>			
Mostrar Productos	productos	array	Array de objetos con información de los productos (nombre, descripción, precio, imagen, etc.)
Actualizar Carrito	cantidadTotal	int	Cantidad total de



			productos en el carrito
--	--	--	----------------------------

#### Interfaz de Inicio de sesión

Interacción	Nombre	Tipo	Descripción
<b>Frontend -&gt; Backend</b>			
Iniciar Sesión	email	string	Email del usuario
Iniciar Sesión	contraseña	string	Contraseña del usuario
<b>Backend -&gt; Frontend</b>			
Confirmación de Inicio de Sesión	exito	boolean	Indica si el inicio de sesión fue exitoso
Confirmación de Inicio de Sesión	mensaje	string	Mensaje de éxito o error



### Interfaz de Registro

Interacción	Nombre	Tipo	Descripción
Frontend -> Backend			
Registrar Usuario	nombreCompleto	string	Nombre completo del usuario
Registrar Usuario	email	string	Email del usuario
Registrar Usuario	provincia	string	Provincia del usuario
Registrar Usuario	ciudad	string	Ciudad del usuario
Registrar Usuario	codigoPostal	string	Código Postal del usuario
Registrar Usuario	calle	string	Calle del usuario
Registrar Usuario	numeroCalle	int	Número de calle del usuario
Registrar Usuario	telefono	string	Teléfono del usuario
Registrar Usuario	fechaNacimiento	date	Fecha de nacimiento del usuario
Registrar Usuario	contraseña	string	Contraseña del usuario
Registrar Usuario	confirmarContraseña	string	Confirmación de la contraseña



Backend -> Frontend			
Confirmación de Registro	exito	boolean	Indica si el registro fue exitoso
Confirmación de Registro	mensaje	string	Mensaje de éxito o error

#### Interfaz de Cotizaciones

Interacción	Nombre	Tipo	Descripción
Frontend -> Backend			
Calcular Cotización	metrosCuadrados	float	Metros cuadrados a construir
Calcular Cotización	materiales	array	IDs de los materiales seleccionados
Agregar al Carrito	cotizacion	object	Objeto con la cotización (materiales y cantidades)
Subida de Archivo			Se sube el archivo de AutoCad para su procesamiento
Backend -> Frontend			
Mostrar Cotización	cotizacion	object	Objeto con la cotización (cantidades,





			pallets, precio total)
Cotización por archivo			Se manda el archivo a la API y se obtienen las medidas de largo y ancho para la cotización.

#### Interfaz de Carrito

Interacción	Nombre	Tipo	Descripción
Frontend -> Backend			
Finalizar Compra	direccionEnvio	object	Objeto con la información de la dirección de envío
Finalizar Compra	idEmpresaEnvio	int	ID de la empresa de envío seleccionada
Backend -> Frontend			
Mostrar Carrito	carrito	array	Array de objetos con la información del carrito (productos, cantidades, precios)



TP3 - Diseño de Sistemas de Información

Profesores:

- Juan Pablo Ferreyra.
- Pablo Pioli

Sistema para: Empresa de Fabricación de Materiales

Mostrar Carrito	subtotal	float	Subtotal del pedido
Mostrar Carrito	descuento	float	Descuento aplicado al pedido
Mostrar Carrito	total	float	Total del pedido



## Interfaz de canjes

Interacción	Nombre	Tipo	Descripción
Frontend -> Backend			
Obtener Premios		El frontend realiza una petición GET al endpoint /premios para obtener la lista de premios disponibles y los puntos del usuario	
Canjear Premio	premioId	Número	ID del premio que el usuario desea canjear.
Backend -> Frontend			
Mostrar Premios	puntosDisponibles premios	Número array	Puntos totales disponibles del usuario para canjear.  Lista de premios disponibles para canjear.
Mostrar Resultado Canje	estado mensaje puntosRestantes	string string número	Estado del canje (ej: "exito", "error").



	CodigoVale	string	Mensaje informativo sobre el resultado del canje.  Puntos restantes del usuario después del canje.  Código del vale generado, si corresponde.
--	------------	--------	--