

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

Nuevos Requerimientos

- 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 (puntos) = x (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.

Interfaces/Vistas del sitio web re adaptadas a la app mobile:

Estas son algunas de las vistas que fueron adaptadas a la app mobile, con los nuevos requerimientos. No han sufrido grandes cambios con respecto a las vistas de la página web en la primera iteración, aunque hay algunos cambios y combinaciones.

Login:



Datos implicados o que se necesitan:

IdUsuario: Int (NOT NULL).

NombreUsuario/Email: String; (NOT NULL)

Contraseña: String; (NOT NULL).

Catálogo de productos:



Requerimientos implicados:

- Mostrar el catálogo de productos que se ofrecen.
- Implementar una API existente (de otra empresa)...
- Generar un pedido.

Se ha agregado la opción para usar la API en caso de que el usuario quiera saber con mayor precisión la cantidad de m2 que debe construir con cada material.

Datos implicados o que se necesitan:

IdProducto: Int; (NOT NULL)

Desc/Uso: String; (NOT NULL)

CantPallet: Int; (NOT NULL)

PrecioUnitario: Float; (NOT NULL)

Vista detallada para un producto:



Agregué a la app mobile un requerimiento extra: Permitir una vista con las características del producto, para una mejor visualización del mismo.

Si el usuario está interesado, puede pasar a la vista de cotización, sino puede volver al catálogo.

Datos implicados o que se necesitan:

IdProducto: Int; (NOT NULL)

Desc/Uso: String; (NOT NULL)

CantPallet: Int; (NOT NULL)

PrecioUnitario: Float; (NOT NULL)

Dimensiones: Float; (NOT NULL)

Origen: String; (NOT NULL)

Cotización y cálculo de materiales (si no utiliza la API)

The image shows a mobile application interface for product quotation. At the top, there is a search bar. Below it, the main header is "Cotización de producto" with a user profile icon and a settings gear icon. The main content area is divided into two sections. The left section shows an image of a brick and the text "Ladrillo Hueco". The right section contains a form for entering construction data. The form has a title "Ingrese la cantidad a construir:" and a subtitle "Las unidades deben estar en m2". It includes input fields for "Largo", "Ancho", and "Alto", and a "Calcular materiales" button. Below the form, the results of the calculation are displayed: "Estos datos corresponden a Ladrillo Hueco", "Cantidad de unidades necesarias: 18161", "Equivalen a: 25 pallet/s", "Precio unitario: 500\$", "Descuento: 5%", "Impuestos: 250\$", "Costo total: 8.080.750\$", and "Puntos que ganas con la compra de este producto: *Puntos*". At the bottom, there is a question "¿Desea agregar al pedido?" with two buttons: "Sí, agregar" and "No, volver a pedido".

Barra de búsqueda

Cotización de producto

MI perfil  

 **Ladrillo Hueco**

Por favor, ingrese los datos para cotizar el producto

Ingrese la cantidad a construir:
*Las unidades deben estar en m2

Largo

Ancho

Alto

Calcular materiales

Estos datos corresponden a Ladrillo Hueco

Cantidad de unidades necesarias: 18161

Equivalen a: 25 pallet/s

Precio unitario: 500\$

Descuento: 5%

Impuestos: 250\$

Costo total: **8.080.750\$**

Puntos que ganas con la compra de este producto: ***Puntos***

¿Desea agregar al pedido?

Sí, agregar **No, volver a pedido**

Requerimientos implicados:

- Realizar la cotización por separado de cada producto.
- Cálculo de materiales

Tengo que hacer un paréntesis, decidí conservar la funcionalidad de cotización de la primera iteración, pidiéndole al usuario que ingrese la cantidad a construir, esto precisamente porque un usuario puede que no necesite construir una gran estructura (por

ejemplo, pensé que este podría necesitar materiales para construir una casita para su perro o inclusive una parrilla) por lo que no va a necesitar de un plano autocad para el cálculo de los m2, probablemente la persona haya calculado los m2 que quiere construir y solo necesita la cantidad de materiales para cubrir esa cantidad.

Datos implicados o que se necesitan:

IdProducto: Int; (NOT NULL)

Largo: Float; (NOT NULL)

Alto: Float; (NOT NULL)

Ancho: Float; (NOT NULL)

CantUnidades: Int; (NOT NULL)

EqPallet: Int; (NOT NULL)

PrecioUnitario: Float; (NOT NULL)

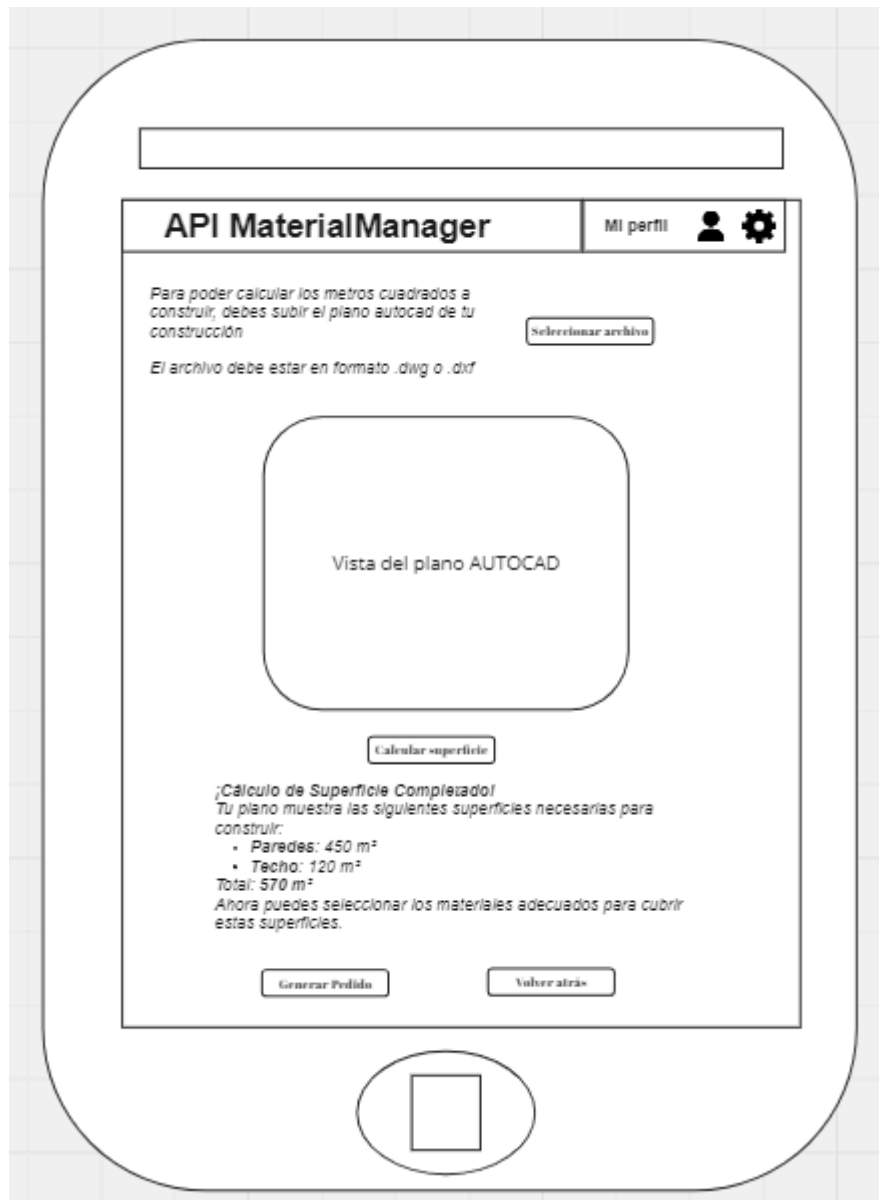
Descuento: Int; (NOT NULL)

Impuestos: Float; (NOT NULL)

CostoTotal: Float; (NOT NULL)

CantPuntos: Float; (NOT NULL)

Vista de la API (la llamé MaterialManager)



Requerimientos implicados:

- Generar un pedido
- Implementar una API existente...

En esta nueva vista, el usuario hace uso de la API para calcular la superficie que tiene que construir, subiendo a la plataforma su plano autocad. En base a este, el sistema calcula la superficie de paredes y techo junto al total en m2. Una vez el usuario vea la información, procede a generar el pedido o volver atrás al catálogo.

Datos implicados o que se necesitan:

CantPared: Float; (NOT NULL)

CantTecho: Float; (NOT NULL)

TotalSup: Float; (NOT NULL)

Cotización de producto (si se utilizó la API):

The screenshot shows a mobile application interface for product quotation. At the top, there is a search bar. Below it, the title "Cotización de producto" is displayed next to a user profile icon and a settings gear icon. The main content area features a product image of a brick labeled "Ladrillo Hueco". To the right of the image, it shows calculated surfaces: "Paredes: 450m2", "Techo: 120m2", and "Total: 570m2". Below this, there is a section titled "Seleccione las superficies a cubrir con el material:" with three radio button options: "Solo Paredes", "Solo Techo", and "Ambas". A "Calcular materiales" button is positioned below the options. The interface then displays the following information: "Estos datos corresponden a Ladrillo Hueco", "Cantidad de unidades necesarias: 16161", "Equivalen a: 25 pallet/s", "Precio unitario: 500\$", "Descuento: 5%", "Impuestos: 250\$", and "Costo total: 8.080.750\$". It also shows "Puntos que ganas con la compra de este producto:" followed by "*Puntos*". At the bottom, there is a question "¿Desea agregar al pedido?" with two buttons: "Sí, agregar" and "No, volver a pedido". A home button icon is located at the very bottom of the screen.

Requerimientos implicados:

- Realizar la cotización por separado de cada producto.
- Cálculo de materiales

Si el usuario utilizó la API, selecciona un producto para cotizar y calcula los materiales, primero debe seleccionar qué superficie va a cubrir con ese material. Cada material tiene sus propias dimensiones por lo que este es capaz de cubrir tantos m2s, por lo que el cálculo para los materiales va a tener en cuenta esta información a la hora de devolver las unidades de material necesarias.

Una vez que se calculan los materiales, el usuario decide si lo agrega al pedido o si vuelve a la vista de su pedido.

Por cada producto que va agregando, el sistema debe ir actualizando los m2 de las superficies.

Datos implicados o que se necesitan:

CantPared: Float; (NOT NULL)

CantTecho: Float; (NOT NULL)

TotalSup: Float; (NOT NULL)

IdProducto: Int; (Primary Key, NOT NULL)

Largo: Float; (NOT NULL)

Alto: Float; (NOT NULL)

Ancho: Float; (NOT NULL)

CantUnidades: Int; (NOT NULL)

EqPallet: Int; (NOT NULL)

PrecioUnitario: Float; (NOT NULL)

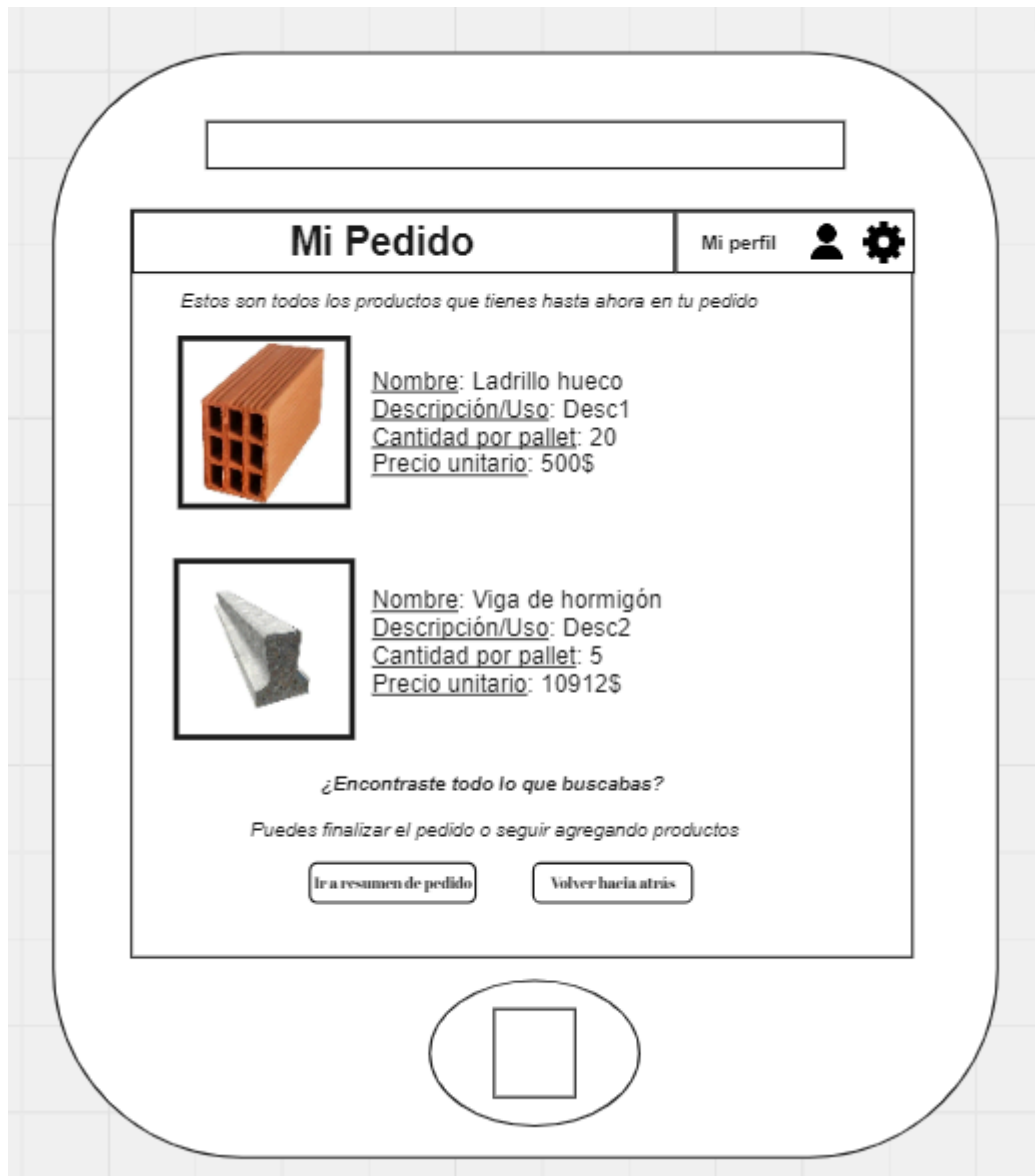
Descuento: Int; (NOT NULL)

Impuestos: Float; (NOT NULL)

CostoTotal: Float; (NOT NULL)

CantPuntos: Float; (NOT NULL)

Mi pedido, con todos los productos que fue agregando el cliente:



Esta es la vista del pedido, con todos los productos que fue agregando el cliente. Puede decidir si proceder a ver un resumen con más detalle del mismo o volver hacia atrás, cancelando el pedido.

Datos implicados o que se necesitan:

IdUsuario: Int; (NOT NULL)

IdPedido: Int; (NULL)

IdProducto: Int; (NOT NULL)

Desc/Uso: String; (NOT NULL)

CantPallet: Int; (NOT NULL)

PrecioUnitario: Float; (NOT NULL)

Resumen de pedido:

Resumen de Pedido

Mi perfil

Nro Pedido: Fecha:

Producto	Unidades	Precio Unitario	Total

Subtotal:

Descuentos:

Impuestos:

Puntos ganados con este pedido:

Total del pedido:

Proceder con el pago Cancelar Pedido

Requerimientos implicados:

- Mostrar un resumen del pedido realizado.
- Sumar puntos por cada compra realizada...

En la vista de resumen del pedido, lo que he agregado es la información de cuántos puntos va a ganar el cliente con este pedido.

Datos implicados o que se necesitan:

IdPedido: Int; (NULL)
Fecha: Date; (NOT NULL)
IdProducto: Int; (NOT NULL)
Unidades: Int; (NOT NULL)
PrecioUnitario: Float; (NOT NULL)
Total: Float; (NOT NULL)
Subtotal: Float; (NOT NULL)

Descuentos: Int; (NOT NULL)
Impuestos: Float; (NOT NULL)
CantPuntos: Float; (NOT NULL)

Método de pago:



Requerimientos implicados:

- Ofrecer variadas opciones de pago de un pedido.
- Pagar los pedidos realizados mediante diferentes plataformas de pago, modo, mercadopago, etc.



Aquí he agregado las opciones de plataformas de pago, si el usuario desea pagar en una de estas, lo redireccionará a la aplicación correspondiente, manejando el flujo de datos en la misma y realizando el pago del pedido por dicho medio.

Datos implicados o que se necesitan:

TipoPago: String; (NOT NULL)
TipoPlataforma: String; (NULL)

Monto: Float; (NOT NULL)
FechaPago: Date; (NOT NULL).

Sistema de puntos, desde el sitio web:

Sistema de Puntos Mi perfil  Rol: Admin 

En este apartado, puedes cambiar y/o consultar las reglas para el sistema de puntos:

Puntos que se otorgan:

Pesos que se deben gastar:

Por último, esta vista es para el administrador el cual una de sus funciones será el manejar el sistema de puntos de la aplicación. Podrá este modificar las características de este sistema en base a las decisiones que tome la empresa.

Los usuarios van acumulando estos puntos mediante los distintos pedidos que van realizando, seguramente puedan canjearlos por algún descuento o promoción, insisto que depende de qué es lo que le conviene a la empresa.

Datos implicados o que se necesitan:

IdUsuario: Int; (NOT NULL)

Rol: String; (NOT NULL)

CantPuntos: Float; (NOT NULL)

CantPesos: Float; (NOT NULL).

Flujo de datos entre componentes del sistema durante la cotización:

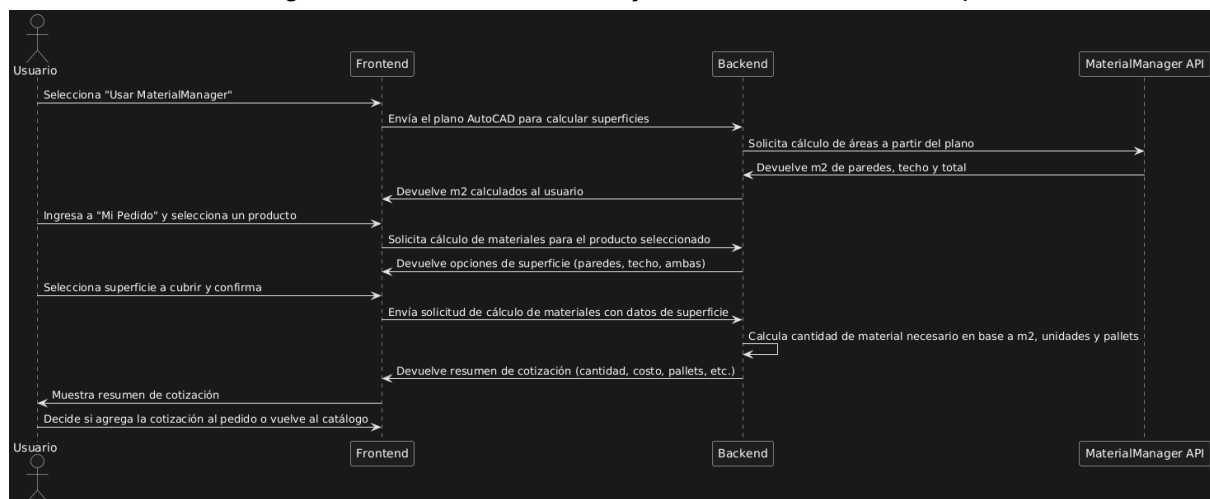
Voy a explicar cómo se da el flujo de datos entre los componentes que conforman el sistema durante la cotización, pero esta vez en el caso de que el usuario haya utilizado la API de MaterialManager.

El usuario ingresa al catálogo de productos y elige la opción de usar la API de MaterialManager. En esta vista, carga el plano autocad y se lo envía al servidor para procesarlo, este se comunica con el controlador de la API de MaterialManager para poder realizar el cálculo de las superficies. Una vez que se procesa toda la información, el

controlador de la API de MaterialManager devuelve los m2 de las superficies de paredes y techo y el total al servidor de mi sistema y este los envía a la interfaz de usuario. Luego, el usuario genera el pedido desde el front y va cotizando cada producto por separado, ahora el cálculo para los materiales que se necesitan están acompañados con los datos del cálculo de las superficies, por lo que antes de poder calcular los materiales el usuario necesita seleccionar qué tipo de superficie debe cubrir con ese material, una vez lo haya seleccionado pulsa en calcular materiales. El backend del servidor toma las dimensiones de dicho material, calcula cuánto cubre en m2 y en base esta información logra calcular cuántos m2 se cubren con dicho material, cuántas unidades del material se necesitan, el costo total, equivalencia en pallets, entre otros datos y se los devuelve a la interfaz. Finalmente, el usuario decide si agrega dicha cotización al pedido o si vuelve al catálogo.

Diagrama de secuencia del flujo de datos para la cotización, integrando la API de MaterialManager:

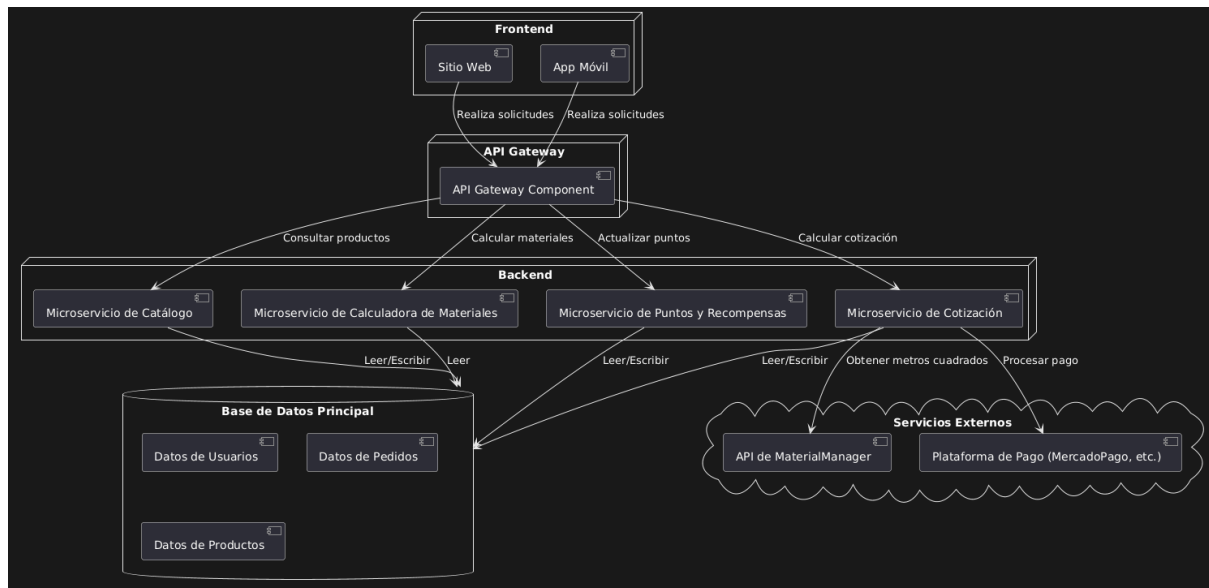
A continuación, el diagrama de secuencia del flujo de datos, realizado en plantUML.



Para una mejor visualización:

https://www.plantuml.com/plantuml/dpng/VLCzRjim5DvrYW-UKa3jqB53aJYfWGm60XGvm0lveXAb-0JoqK1xcnoWKu_WY_L9bXjRTRgltP-bwwAKjVQn8QUiYlWgvl3DOAV1XO6iOhN9afIJNu4ADkTtQtBVZrVhywijAUT0SKk9EiwuURYVODjjqym-UFyP6tp2LuxiN914M3qMobXmLyrcXAuD-YNjzhy87354leARgd9xSmUp9ZYAhaRZbpfmVhOeN9grKSrjq95eHsgu_Si857Z6_YKp5H2cy1d-ADqOPpQzurtbk6Fq7qUA8TbpQQ7ilWb-G4Kf7ems2_WSoOjPHDHvt-K0zwdBN0XaimKyi0zULYPSNeUf2KCMNvtAwneN6_NpU5mEgt3yopmHz78Xi0pZdH4zdso8g_F6BQX_elBkF-zuoYQu-fH3jZvEqdF8FVtpdkLGf7fpVgkCaU59PP6kiHwpsErXR3XA6ZpvKm4aTboc9nWyHba3zJQI2EKYyKRA4RMSc6OpANsnXIJOQVvCB-TyBLqQcruwBI7YLQIiBeFjj1KT9x91BULYsPwGsyvgFcx9ODE2vpHuXqvayHm_InG30djlD7QWjA9yqr9p_-tNy0

Arquitectura:



Para mejor visualización:

https://www.plantuml.com/plantuml/dpng/bLNDYXD14BxtKnIvgE3gtODit5r5C3YuY0V9eQQx4XexNMDrZXgh-30-mpv2NioQJWOdu8Gaa6AgvgkIVkg3kKmp9PTDCEO9NDFAHx_X147_aPtQ3WN1ebF3cuuZnKpGYO_MTnY08WG4bE_5PxAv21hD8vYz5LPaT3F01Akz0xyCwE_hhS-UuGkrlzW5vbq7Y-tZdGzByzliAURDUtY7cNxWGsN1pgztxi0pGi3LqDw8uGtQRqCFxUvvl5ruAvn8xhpr30g-mhpz4tZDooa0P_yJhTy-naa81bi2EXRitGLc4e-1qbH2KsBc10_mYVhrKaoQQlyECxQOwWng5NgDFR8PDbdzSkqmpkwjmZwdWk9PMGxZ3Jdl_XCMTiLctbMrWOk3sUs-mmGtzvaaSghLg3eOHjiVJxd0Y6kIVOKcOCOLomPh6LmpF5sGM7JSEy-1idtnJAIu2izof7klfb-AK41R7JDM25nSI9wwFRpILMBGGq7Ye4FauZHzhBZpiygKNACeAKdv51rJ2Ha5kc7Lvz6DTQU4WyO4xB_ukOoJCXtpRuRmUGrFY5gPvpQNkbo1haACEJvjp-Lch9-R3D5gaufAmdOWaxGSJg9zal1Qfb6s5BgkzQuENRIoknQpVb092zlaXNVUZdUpsJMEOSxv32d7MMoihcawEf7ulyQ

He elegido para mi sistema una arquitectura híbrida entre **Capas y Microservicios**, ya que permite tres características fundamentales: escalabilidad, flexibilidad y modularidad. Al dividir el backend en microservicios independientes, cada uno especializado en una funcionalidad específica, se facilita el escalamiento individual de cada servicio, optimizando el uso de recursos y simplificando futuras implementaciones. Combinando esta estructura con un enfoque en capas, obtenemos una organización más clara y ordenada. Además, esta arquitectura se adapta bien a un entorno con múltiples interfaces (tanto web como móvil), permitiendo que ambas interactúen con el mismo backend y se integren eficientemente con servicios externos, como la API de MaterialManager para el cálculo de superficies y las plataformas de pago (Mercado Libre, Modo...).