UPBEats

DOCUMENTO DE ARQUITECTURA

**Equipo de Trabajo**

Juan Camilo Restrepo Velez

Wilder Valencia Ocampo

**Asesor**

William Yovany Franco Rua

Historia

|  |  |  |
| --- | --- | --- |
| Versión | Fecha | Descripción de Cambios |
| 1.0 | 10/05/2020 | Construcción de documento de arquitectura. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

CONTENIDO

**Pág.**

[1 Descripción del Documento 3](#_Toc40220181)

[1.1 Propósito 3](#_Toc40220182)

[1.2 Contenido 3](#_Toc40220183)

[1.3 Audiencia 3](#_Toc40220184)

[1.4 Definiciones y acrónimos 3](#_Toc40220185)

[2 Panorama del Proyecto 4](#_Toc40220186)

[2.1 Contexto del negocio 4](#_Toc40220187)

[2.2 Contexto del sistema 4](#_Toc40220188)

[3 Drivers de la Arquitectura 5](#_Toc40220189)

[3.1 Restricciones de negocio 5](#_Toc40220190)

[3.2 Restricciones técnicas 5](#_Toc40220191)

[4 Estilos Arquitectónicos Usados 6](#_Toc40220192)

[5 Vista Conceptual 7](#_Toc40220193)

[5.1 Diagrama de Paquetes 7](#_Toc40220194)

[5.2 Casos de Uso Claves 8](#_Toc40220195)

[6 Vista Lógica 9](#_Toc40220196)

[6.1 Diagramas de Paquetes 9](#_Toc40220197)

[6.2 Diagramas de Clases 10](#_Toc40220198)

[6.3 Diagramas de secuencias 11](#_Toc40220199)

[7 Vista Física 18](#_Toc40220200)

[7.1 Diagramas de Despliegue 18](#_Toc40220201)

[7.2 Descripción de la plataforma tecnológica 18](#_Toc40220202)

[7.3 Consideraciones de implementación de los componentes a construir 18](#_Toc40220203)

[7.4 Consideraciones de componentes a reutilizar 18](#_Toc40220204)

[8 Vista de Implementación 20](#_Toc40220205)

[8.1 Diagramas de Componentes 20](#_Toc40220206)

[8.2 Diagramas de Paquetes 20](#_Toc40220207)

[8.3 Interfaces y operaciones de la capa de Interfaz 21](#_Toc40220208)

[8.4 Interfaces y operaciones de la capa de Aplicación 21](#_Toc40220209)

[8.5 Interfaces y operaciones de la capa de Datos 22](#_Toc40220210)

[9 Calidad 23](#_Toc40220211)

[9.1 Árbol de utilidad 23](#_Toc40220212)

# Descripción del Documento

## Propósito

El propósito de este documento es tener un consenso entre los miembros del equipo de trabajo en el diseño de la arquitectura de software de la aplicación UPBEats. La arquitectura planteda en este documento se basa en 4 vistas que defienen aspectos en los siguientes frentes:

* Vista Conceptual: visión que los usuarios tienen de la aplicación.
* Vista Lógica: visión desde los principales elementos y principios del diseño.
* Vista Física: visión desde la distribución del procesamiento entre los dispositivos.
* Vista de Implementación: visión que muestran cómo serán montados los diferentes componentes de la aplicación y la forma en que interactúan.

## Contenido

Este documento consta de 9 secciones:

* Las secciones 1 y 2 describen este documento y el proyecto.
* Las secciones 3 y 4 muestran los drivers y los estilos arquitectónicos usados.
* De la sección 5 a la 8 se ilustran las vistas arquitectónicas de UPBEats.
* La sección 9 muestra las condiciones de tamaño, rendimiento y calidad.

## Audiencia

Este documento se crea para los miembros del equipo de la compañía .Sostenible. La audiencia objetivo de este documento incluye el equipo de trabajo y el asesor.

## Definiciones y acrónimos

UPB: Universidad Pontificia Bolivariana

Browser: Es un navegador de internet, es decir, un software que permite la visualización de los contenidos que presenta una página web

# Panorama del Proyecto

## Contexto del negocio

En la universidad existen muchos pequeños emprendedores que son estudiantes de esta, pero como no se cuenta con un aplicativo o plataforma para conocerlos a todos, muchos microempresarios quedan en el olvido y no pueden obtener muchas ventajas fácil y óptimamente.

Los clientes serán los estudiantes de la universidad que tengan el perfil de pequeños emprendedores vendiendo sus productos dentro de la misma. Actualmente son alrededor de 300 personas, que pueden ir aumentando o disminuyendo dependiendo de los estudiantes que desean ingresar a esta opción de comercio.

## Contexto del sistema

Se busca desarrollar una plataforma donde todos los estudiantes de la universidad puedan conocer y buscar todos los diferentes vendedores y productos que se ofrecen dentro de la UPB, así mismo realizar los pedidos como comprador y gestionar los mismo como el vendedor, implementar un chat para que se puedan comunicar desde la misma plataforma, y con todo esto poder facilitar y optimizar el tiempo de compra y venta de productos dentro de la misma aplicación.

# Drivers de la Arquitectura

|  |  |  |  |
| --- | --- | --- | --- |
| # | Driver Arquitectónico | Importancia | Dificultad |
| 1 | Comunicar con un Usuario | Alta | Media |
| 2 | Realizar Pedidos | Alta | Media |
| 3 | Gestionar Pedidos | Alta | Media |
| 4 | Pagar Virtual | Alta | Baja |
| 5 | Recibir Pago Virtual | Alta | Baja |
| 6 | Conexión a Internet | Alta | Alta |
| 7 | Transacciones virtuales | Alta | Baja |
| 8 | Canales seguros | Alta | Media |
| 9 | Disponibilidad de la Información | Media | Baja |
| 10 | Transmisión de la información | Alta | Baja |
| 11 | Alto tráfico de usuarios | Alto | Alta |

## Restricciones de negocio

* Solo es permitido el acceso a personas que cuente con correo institucional de la UPB.

## Restricciones técnicas

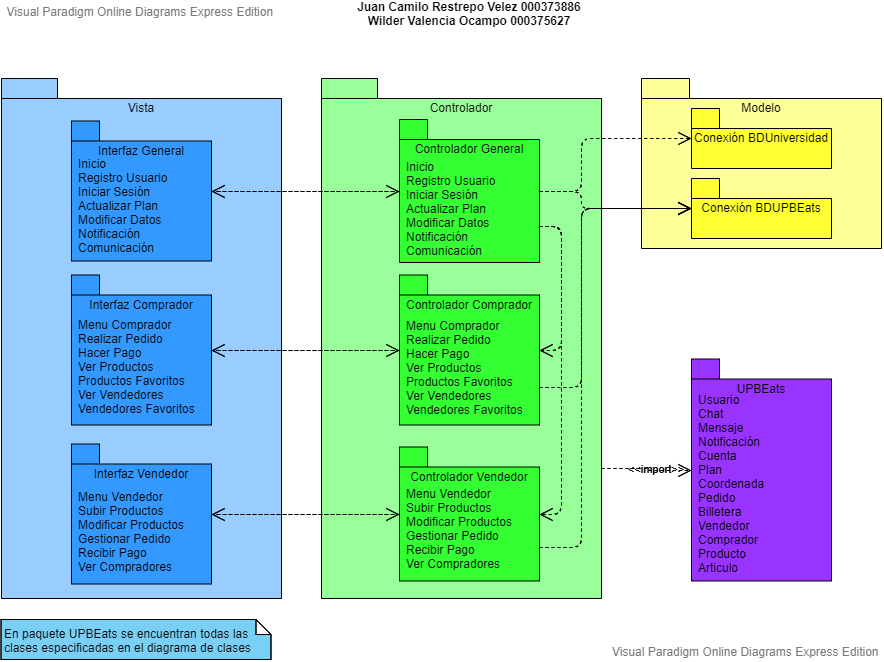
* Para el uso de la plataforma se requiere dispositivo que soporte cualquier tipo de browser.
* Tanto el comprador como el vendedor que se van a comunicar requieren acceso a internet.

# Estilos Arquitectónicos Usados

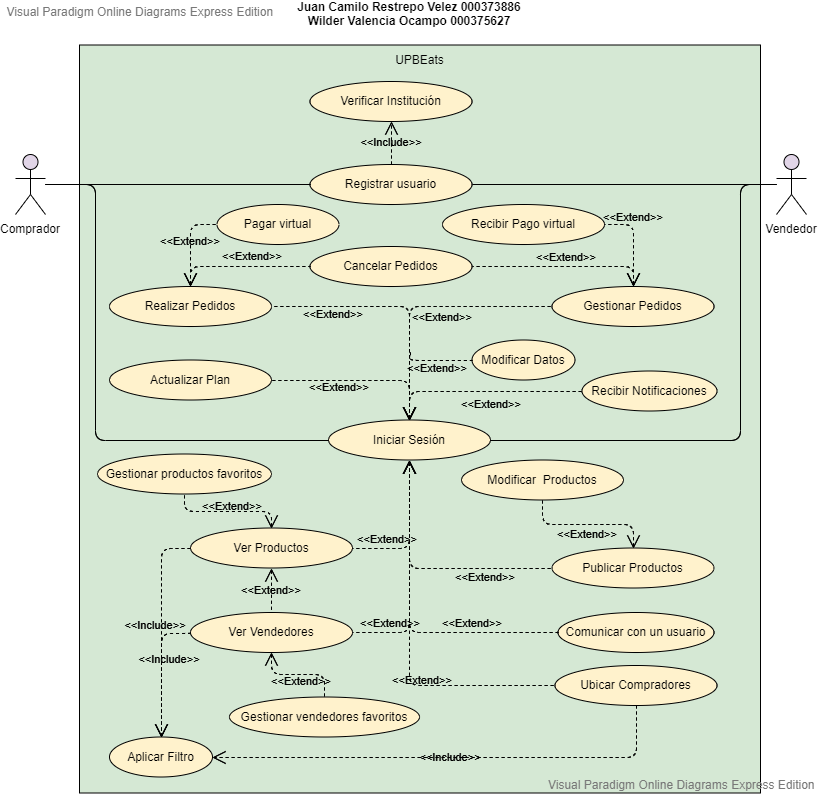
* **SOA (Comunicación):** la funcionalidad de UPBEats se proporcione como un conjunto de servicios. Los servicios se acoplan libremente porque utilizan interfaces basadas en estándares que se pueden invocar, publicar y descubrir. Los servicios en SOA se centran en proporcionar un esquema y una interacción basada en mensajes con una aplicación a través de interfaces que tienen un alcance de aplicación, y no componentes u objetos.
* **3-Tiers / N-Tiers (Implementación):** la separación de la funcionalidad en segmentos de la misma manera que el estilo en capas, pero cada segmento es un nivel que se puede ubicar en una computadora físicamente separada. Evolucionaron a través del enfoque orientado a componentes, generalmente utilizando métodos específicos de plataforma para la comunicación en lugar de un enfoque basado en mensajes.
* **Domain Driven Design (Dominio):** diseñar software basado en el dominio empresarial, sus elementos y comportamientos, y las relaciones entre ellos. Se logra habilitar sistemas de software que sean una realización del dominio empresarial subyacente mediante la definición de un modelo de dominio expresado en el lenguaje de los expertos en el dominio empresarial. El modelo de dominio se puede ver como un marco desde el cual las soluciones se pueden racionalizar.
* **Object-Oriented (Estructura):** la división de responsabilidades para la UPBEats en objetos individuales reutilizables y autosuficientes, cada uno de los cuales contiene los datos y el comportamiento relevante para el objeto. Esto hace que UPBEats se vea como una serie de objetos cooperantes, en lugar de un conjunto de rutinas o instrucciones de procedimiento. Los objetos son discretos, independientes y poco acoplados; se comunican a través de interfaces, llamando a métodos o accediendo a propiedades en otros objetos, y enviando y recibiendo mensajes.

# Vista Conceptual

## Diagrama de Paquetes

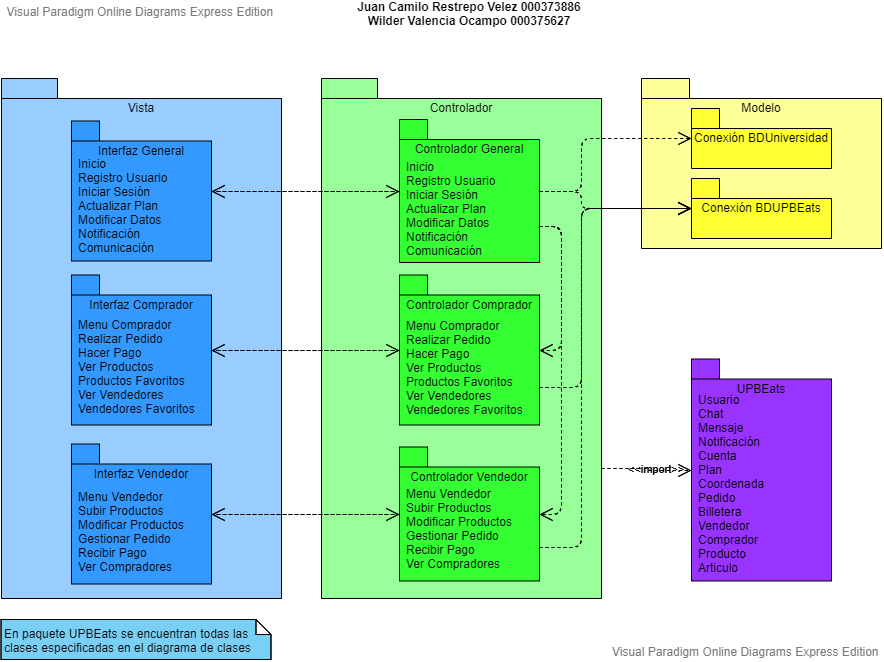


## Casos de Uso Claves

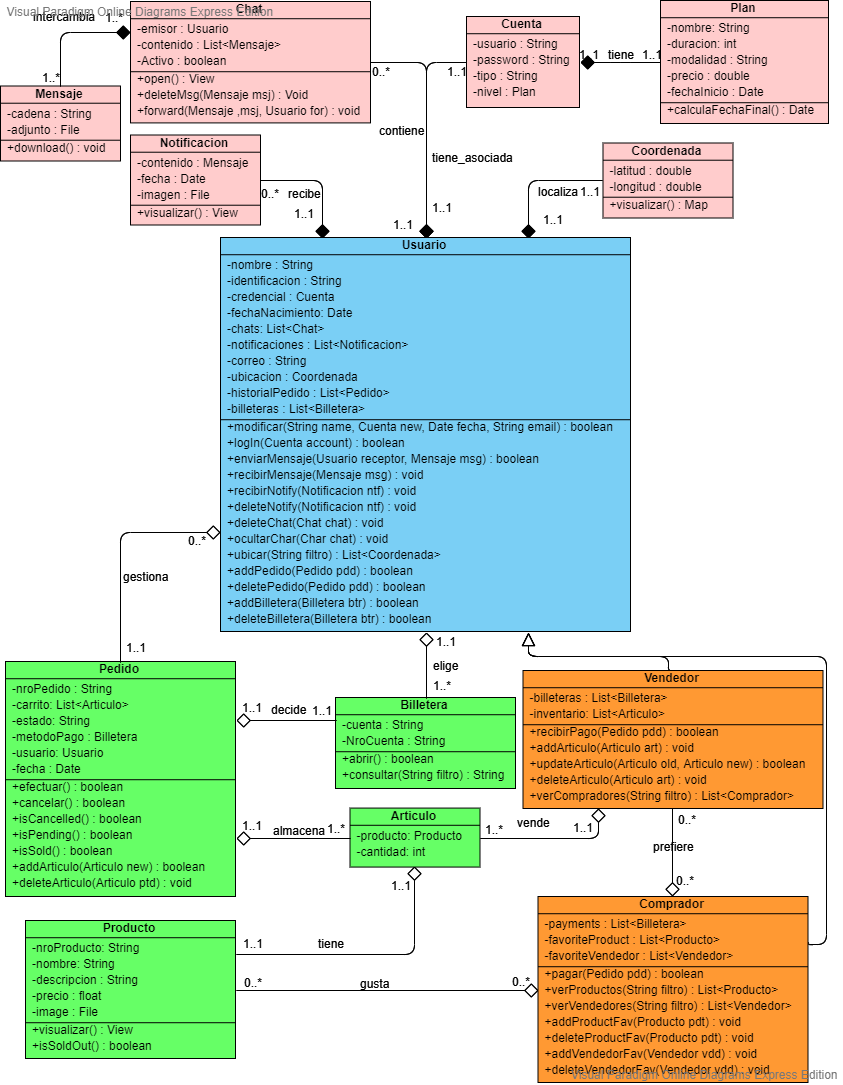


# Vista Lógica

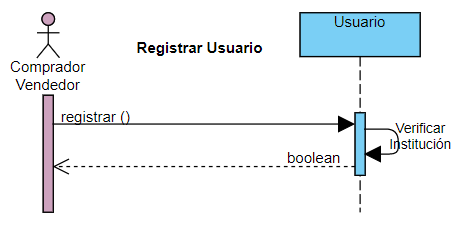
## Diagramas de Paquetes

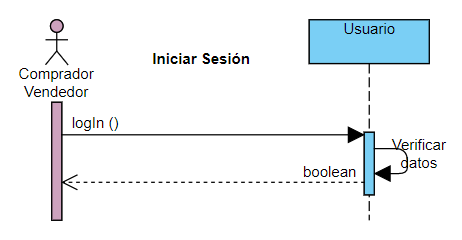


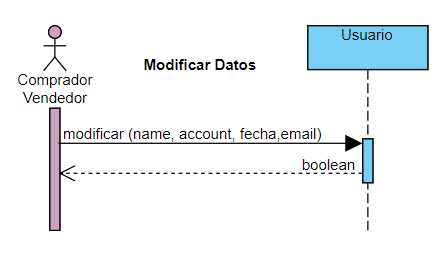
## Diagramas de Clases

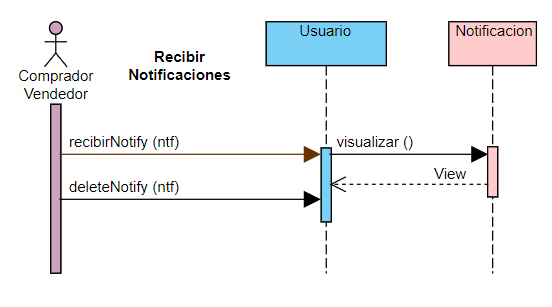


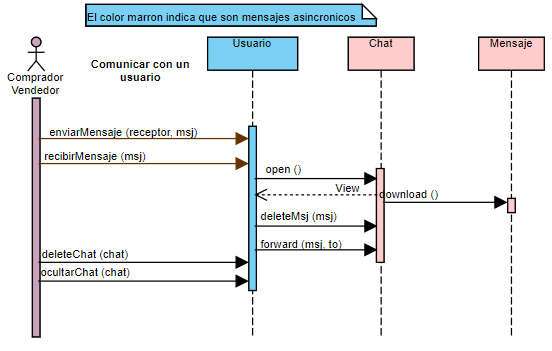
## Diagramas de secuencias

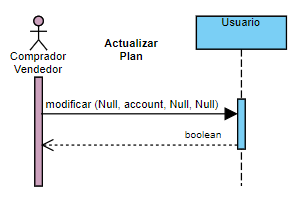


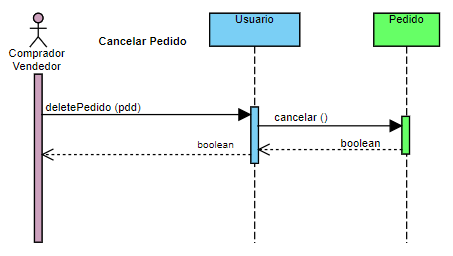


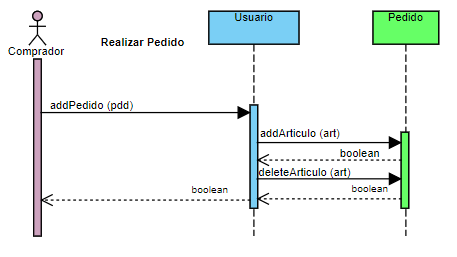


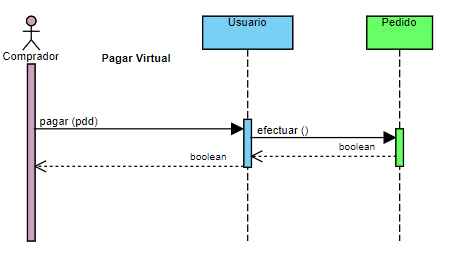


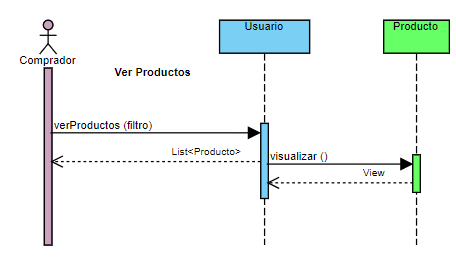


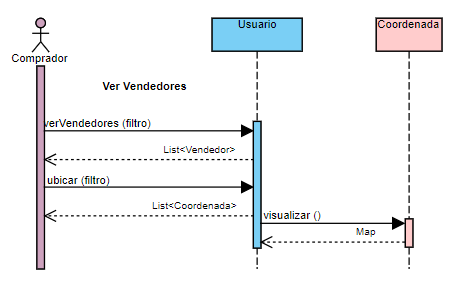
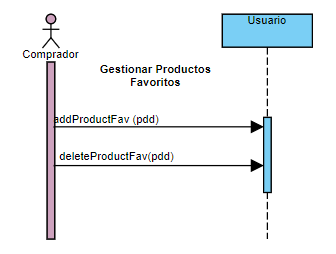


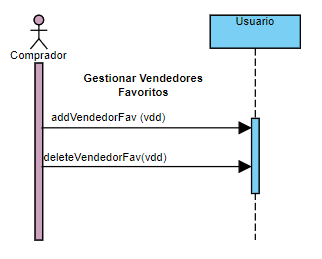


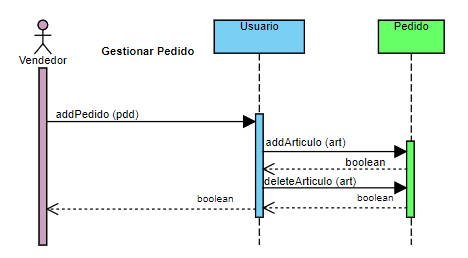


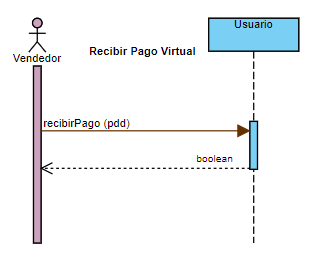


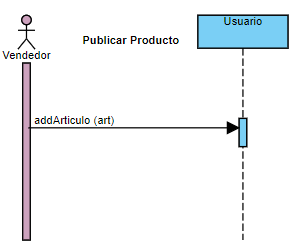


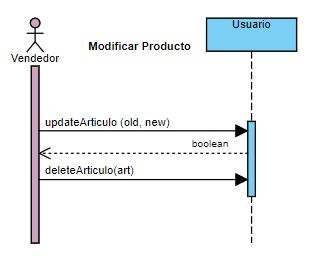


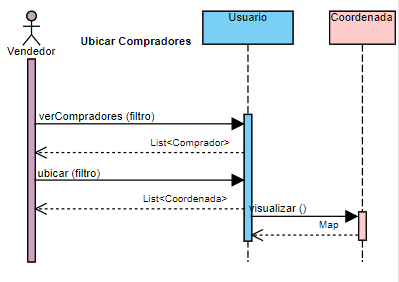






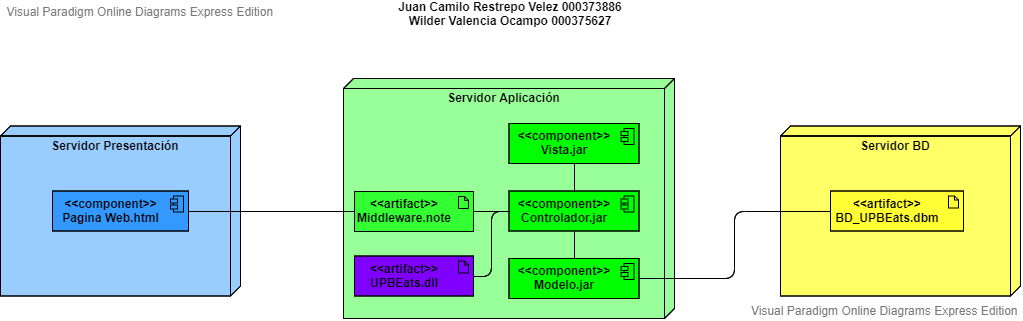






# Vista Física

## Diagramas de Despliegue



## Descripción de la plataforma tecnológica

Se utilizará la plataforma Amazon Web Services (AWS), que es una colección de servicios de computación en la nube pública (servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas por Amazon.com.

Se realiza la implementación en dicha nube debido a que soporta los de Java, Node.js, así como HTML y el motor de base de datos SQL Server, que son las herramientas necesarias para el despliegue de la aplicación. Además, gracias a AWS Lambda las respuestas a los eventos que se puedan generar ocurren en milisegundos ofreciendo una mejor experiencia de usuario y optimizando los costos cuando no se está utilizando la aplicación.

## Consideraciones de implementación de los componentes a construir

Todos los componentes como lo son: el Menú comprador y vendedor, actualizar plan, modificar datos, notificaciones, comunicación, los afines a los productos, pedidos, vendedores y compradores, se implementarán de forman que sirvan como un micro-servicio que exponga la información para posteriormente desde otro componente poder obtenerla, y así tener un menor acoplamiento lo que facilita el desarrollo y su mantenimiento.

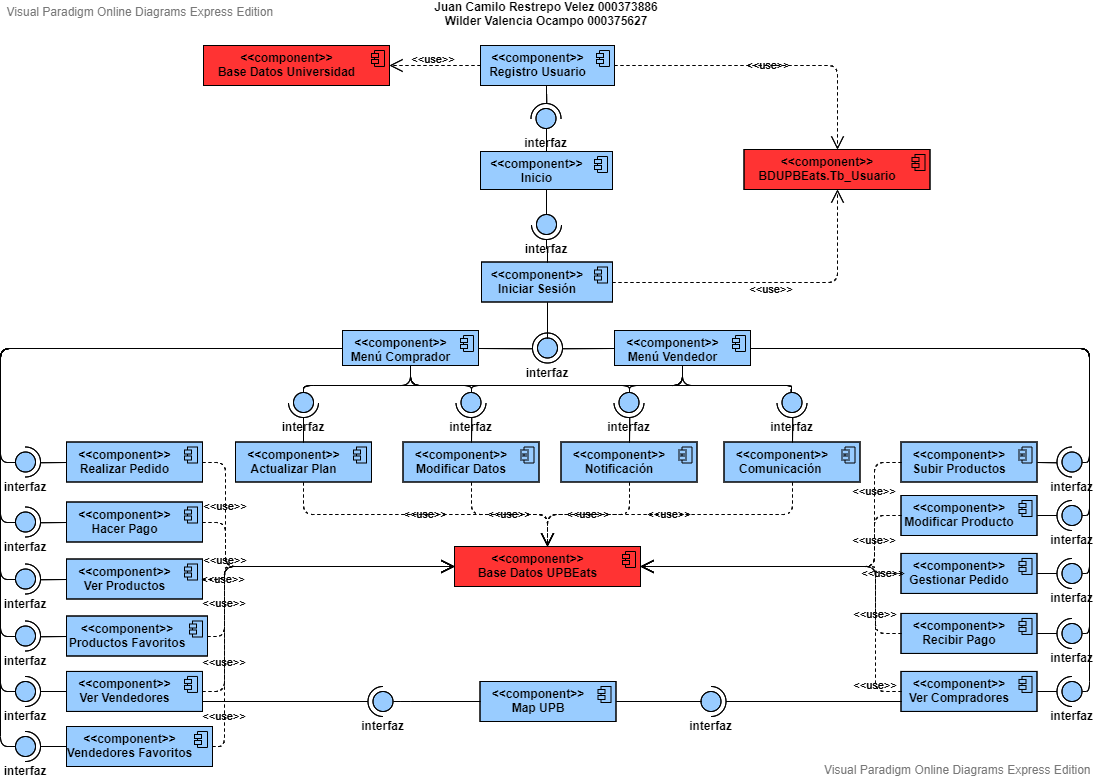
## Consideraciones de componentes a reutilizar

El principal componente a reutilizar es el de autentificación debido a que todos los usuarios de la plataforma deben contar con correos instituciones de la universidad, por lo que se hace necesario reutilizar la autentificación por parte de Office 365.

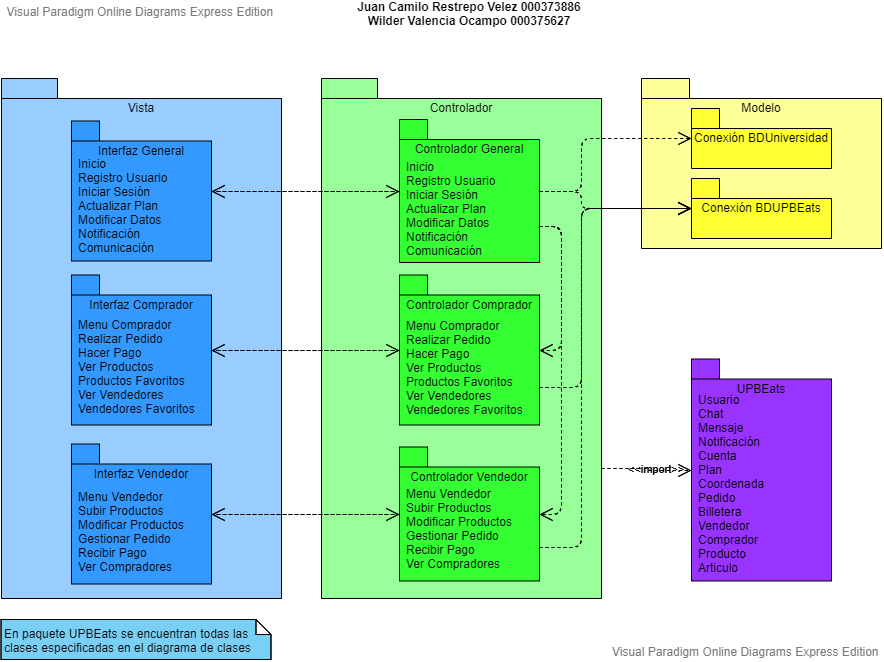
Además, otro componente a reutilizar es la visualización por un mapa que muestre los estudiantes dentro de la universidad.

# Vista de Implementación

## Diagramas de Componentes



## Diagramas de Paquetes



## Interfaces y operaciones de la capa de Interfaz

Esta capa se encarga de proveer las vistas al usuario, esto es, mostrar la información y permitir la interacción de los componentes de la capa de aplicación:

* Interfaz general: Incluye las ventanas del inicio de la página, de registro, inicio de sesión, de gestión de los datos del usuario, las notificaciones y los chats con los otros usuarios.
* Interfaz del comprador: Abarca las vistas de los productos con todas sus funcionalidades, así como todas las vistas necesarias para crear un pedido.
* Interfaz del vendedor: Contiene todas las opciones que el usuario tiene como vendedor para gestionar los productos que publicas y los pedidos que recibe.

## Interfaces y operaciones de la capa de Aplicación

En esta capa se aplica toda la lógica de negocio en los siguientes componentes:

* Autenticación: Se encarga de la verificación con la UPB y autentificación de la aplicación por medio del consumo de una API para el registro y logeo de Office 365.
* Comunicación: Encargado de comunicar usuarios a través de un chat, se hace vital su correcto funcionamiento.
* Información de Usuario: Es un complemento para el Usuario, le permite interactuar con su información.
* Producto: UPBEats se encarga de conectar personas, cuyo fin es adquirir y suministrar productos, por ello el manejo de los productos es esencial.
* Pedidos: si bien los productos son importantes, estos toman importancia cuando hacen parte de un pedido, además de que son parte del core domain de UPBEats.

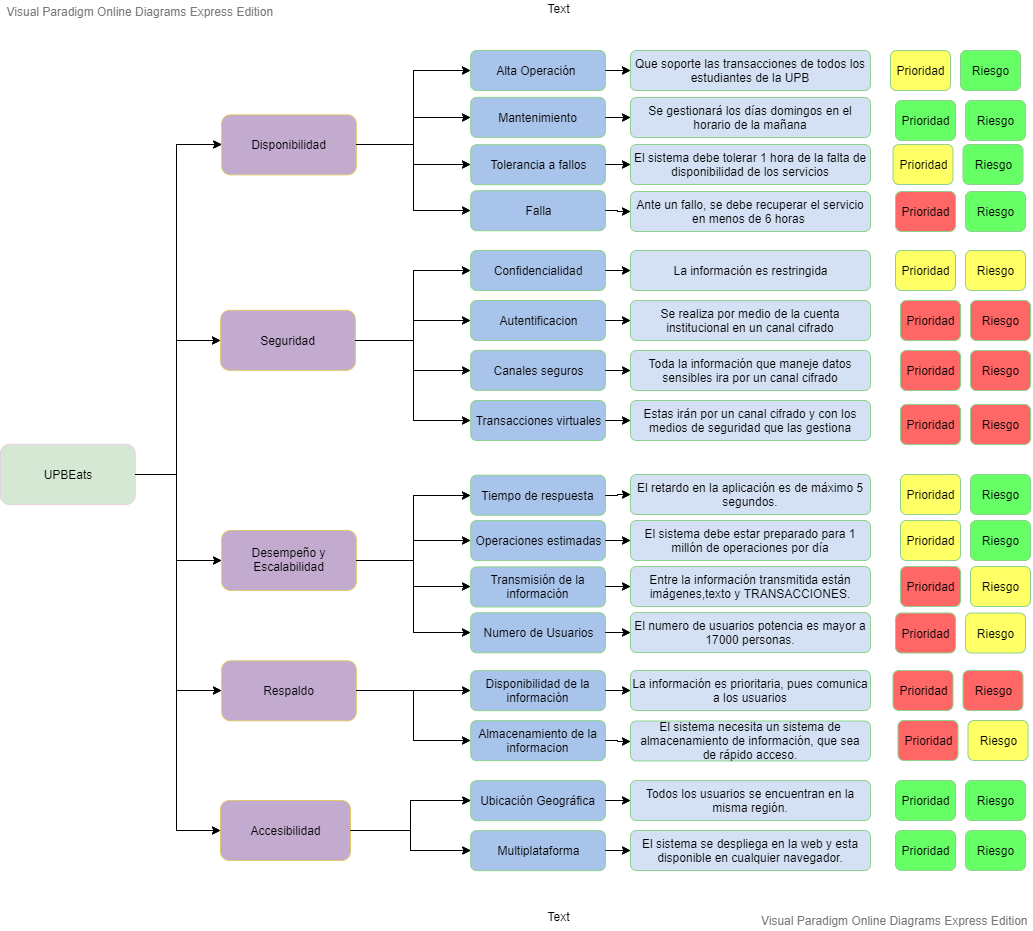
## Interfaces y operaciones de la capa de Datos

Esta será la capa de persistencia de datos para almacenar toda la información de los usuarios que posteriormente serán objeto de consulta:

* Base de Datos UPBEats: Se almacena y procesa toda la información de los usuarios de la plataforma, para poder realizar todas las funcionalidades.
* Base de Datos Universidad: este componente es vital para verificar la identificación de los usuarios como estudiantes de la universidad.

# Calidad

## Árbol de utilidad



Los atributos de calidad relevantes son:

* Seguridad: especial cuidado ya que UPBEats cuenta con transacciones virtuales, por lo cual la seguridad se convierte en una tributo no funcional clave.
* Desempeño y Escalabilidad
* Respaldo