**[GreenMarket]**

**(SAD) Software Architecture Document**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | BugiFost |
| --- | --- |
| **Proyecto** | GreenMarket |
| **Versión** | 1.0 |

| **Documento mantenido por** | Juan Herrera |
| --- | --- |
| **Fecha de última revisión** | 16/09/2024 |
| **Fecha de próxima revisión** | 23/09/2024 |

| **Documento aprobado por** | Grupo de jardineros |
| --- | --- |
| **Fecha de última aprobación** |  |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 16/09/2024 | 1.0 | Se realizó la introducción, contexto del problema, propósito y ámbito | Juan Herrera |
| 23/09/2024 | 1.1 | Se realizaron las metas y restricciones |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[1 Introducción 3](#_heading=h.30j0zll)

[1.1 Contexto del Problema 3](#_heading=h.1fob9te)

[1.2 Propósito 3](#_heading=h.3znysh7)

[1.3 Ámbito 3](#_heading=h.2et92p0)

[1.4 Referencias 3](#_heading=h.3dy6vkm)

[1.5 Resumen ejecutivo 4](#_heading=h.1t3h5sf)

[1.6 Representación 4](#_heading=h.4d34og8)

[2 Metas y Restricciones de la Arquitectura 4](#_heading=h.2s8eyo1)

[2.1 Metas de la arquitectura 4](#_heading=h.3rdcrjn)

[Metas de la arquitectura de GreenMarket 4](#_heading=h.s85xprz35siy)

[2.2 Restricciones de la Arquitectura 6](#_heading=h.26in1rg)

[Restricciones de la Arquitectura 6](#_heading=h.g4wbfzdgjsmm)

[2.3 Otros antecedentes y consideraciones 6](#_heading=h.lnxbz9)

[3 Vista de Casos de Uso y Escenarios de Calidad 7](#_heading=h.35nkun2)

[3.1 Modelo de Casos de Uso 7](#_heading=h.1ksv4uv)

[3.2 Especificación de Casos de Uso Relevantes 7](#_heading=h.44sinio)

[3.3 Especificación de los Escenarios de Calidad Relevantes 8](#_heading=h.2jxsxqh)

[Vista Lógica 10](#_heading=h.z337ya)

[3.4 Parte Estructural 10](#_heading=h.3j2qqm3)

[3.5 Parte Dinámica 11](#_heading=h.1y810tw)

[4 Vista de Procesos 12](#_heading=h.4i7ojhp)

[5 Vista de Implementación 13](#_heading=h.2xcytpi)

[Vista de Despliegue 16](#_heading=h.1ci93xb)

[6 Decisiones de Diseño y Selección de Alternativas 16](#_heading=h.3whwml4)

[7 Análisis de Reutilización 17](#_heading=h.x6gm1xbexfjf)

# Introducción

En este documento, se hablarán de los aspectos de la arquitectura de software. Las especificaciones planteadas en este archivo fueron desarrolladas siguiendo la metodología tradicional Espirall;

Además en este documento realizamos el análisis 4+1 planteando las vistas pertinentes para el desarrollo e integración de la arquitectura del proyecto GreenMarket

## Contexto del Problema

* El contexto es que la creación de la plataforma GreenMarket, busca dar un espacio a los jardineros y sus clientes, en donde estos puedan mejorar su canales de venta y a su vez atraer a nuevo público.

## Propósito

* El propósito de GreenMarket es ofrecer la oportunidad a los jardineros de ofrecer sus productos a través de otros medios

## Ámbito

* En este proyecto, se desarrollarán diversas funcionalidad, la más destacable de estas es la implementación de una ia que tendrá el nombre de JardenBot, el cual sería un asistente virtual, al cual se podrán hacer diversas preguntas, pero solamente relacionadas con la jardinería, esta incluirá el análisis de imágenes con las cuales los usuarios de podrán aumentar su conocimientos y resolver sus dudas.
* Cosas que no realizara la plataforma sería que el chatbot no permitirá audio, no se podrá consultar de cosas ajenas a la jardinería, además la plataforma consta con funciones que buscan hacer más sencilla el uso de esta a personas del rango de edad acorde a los usuarios que se provee tendrá la plataforma que rondan los 35 a 55 años, con lo que se obito por no tener un inicio de session, si no que este solo está destinado a la gente que quiere vender sus productos.

Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *GM* | GreenMarket |
| *JB* | JardenBot |
|  |  |
|  |  |

## Referencias

A continuación se listan las referencias a otros documentos:

* Casos de Uso
* Casos de uso extendido
* Diagrama Dinámicos
  + Diagrama de actividades y secuencias
* Diagrama Estáticos
  + Diagrama de Clases, componente, paquete, despliegue .
* Diagrama de Base de Datos lógico y relacional.
* Informe ERS.

## Resumen ejecutivo

## Representación

La arquitectura del sistema GreenMarket está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Casos de Uso y Escenarios de Calidad**: Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
* **Vista de Metas y Restricciones**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.
* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Clases para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.(Diagrama de clases y componentes)
* **Vista de Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización. (Diagrama de actividades y secuencias)
* **Vista de Implementación**: Describe los componentes de deployment construidos y sus dependencias.

# **Metas y Restricciones de la Arquitectura**

A continuación se revisan las metas y restricciones de la arquitectura.

## Metas de la arquitectura

De acuerdo a las reuniones y al análisis de los requerimientos, se listan los principales conductores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

### **Metas de la arquitectura de GreenMarket**

#### **Desempeño:**

* **Objetivo:** Asegurar que la plataforma responda de manera eficiente bajo condiciones normales y de carga, proporcionando una experiencia de usuario fluida.
* **Meta 1:** El sistema debe responder a las solicitudes de los usuarios en un tiempo inferior a 200 ms en el 85% de los casos.
* **Meta 2:** Durante los picos de carga (más de 1000 usuarios simultáneos), el tiempo de respuesta no debe superar los 500 ms para evitar la degradación del rendimiento.
* **Meta 3:** La latencia entre el frontend (aplicación web/móvil) y el backend debe mantenerse bajo 100 ms en conexiones estándar, permitiendo tiempos de respuesta rápidos en la interacción del usuario con el sistema.

#### **Tolerancia a Fallos:**

* **Objetivo:** El sistema debe ser resistente y capaz de recuperarse ante fallos sin afectar significativamente la experiencia del usuario o la disponibilidad del servicio.
* **Meta 1:** Implementar mecanismos de redundancia para los componentes críticos del sistema, de modo que cualquier fallo en el hardware o software sea mitigado sin interrumpir el servicio.
* **Meta 2:** El sistema debe poder manejar caídas parciales del servicio con una capacidad de recuperación automática en menos de 15 - 20 minutos.
* **Meta 3:** El sistema debe contar con respaldos de la información, ya sea a nivel de código en caso de que algún riesgo invalide el sistema y a nivel de información se guardería los datos de los involucrados en el sistema

#### **Seguridad:**

* **Objetivo de Calidad General:** Garantizar que la plataforma "GreenMarket" ofrezca una experiencia segura para todos los usuarios, protegiendo su información personal y financiera de posibles vulnerabilidades.
* **Meta 1:** El sistema debe pasar el 90% de las pruebas de penetración y auditorías de seguridad sin vulnerabilidades críticas.
* **Meta 2:** Implementar autenticación de doble factor (2FA) para administradores y vendedores, asegurando la integridad de las cuentas sensibles.
* **Meta 3:** El sistema debe detectar e informar automáticamente cualquier intento de acceso no autorizado o actividad sospechosa en tiempo real.

#### **Modificabilidad/Reuso:**

* **Objetivo:** El sistema debe ser modular y fácilmente modificable para facilitar la evolución futura del sistema y su reusabilidad en otros proyectos.
* **Meta 1:** Los componentes deben diseñarse de manera modular, permitiendo su reutilización en otros proyectos o módulos adicionales de la plataforma.
* **Meta 2:** El 80% de los cambios o mejoras en el sistema deben poder implementarse sin afectar el núcleo del sistema, asegurando una fácil modificación y mantenimiento.

#### **Operatividad:**

* **Objetivo:** El sistema debe ser fácil de operar, con un bajo requerimiento de intervención manual para mantener su disponibilidad y rendimiento.
* **Meta 1:** Implementar monitoreo automatizado y alertas en tiempo real para detectar problemas de rendimiento, fallos en el sistema, o errores críticos.
* **Meta 2:** El sistema debe tener una capacidad de autodiagnóstico y autorreparación en caso de fallos menores, minimizando la intervención manual del equipo técnico.
* **Meta 3:** El equipo de soporte debe poder resolver cualquier problema crítico en el sistema en un tiempo promedio inferior a 72 horas hábiles.

#### **Satisfacción del Cliente:**

* **Objetivo:** Obtener retroalimentación continua de los usuarios y mejorar la experiencia de compra y venta en la plataforma.
* **Meta 1:** Realizar encuestas periódicas a los usuarios, con un objetivo de que al menos el 85% de los usuarios califiquen su experiencia como "buena" o "excelente".
* **Meta 2:** Mantener un índice de satisfacción de clientes del 80% en los primeros 6 meses, a través de mejoras continuas basadas en el feedback.

## Restricciones de la Arquitectura

1. **Tiempo de Construcción:**
   * **Descripción:** Se cuenta con un plazo muy ajustado debido a que son funcionalidades ambiciosas respecto a lo que se quiere ofrecer en el producto final, lo que nos tomará tiempo y puedes que no nos alcance el tiempo estimado para lograr concretar todo
   * **Impacto:** Este tiempo limitado impone una restricción en la profundidad y el alcance de las funcionalidades que pueden ser desarrolladas en cada iteración. Se priorizan las características esenciales y se implementarán mejoras o expansiones en fases posteriores del proyecto.
2. **Infraestructura:**
   * **Descripción:** El sistema contará con servidores de baja exigencia, pero nos tomaría algo de tiempo ya que se tendrá que montar todo, incluyendo los respaldos.
   * **Impacto:** La arquitectura debe estar optimizada para aprovechar esta infraestructura distribuida, lo que implica diseñar componentes que soporten la concurrencia, alta disponibilidad, y escalabilidad.
3. **Otros Componentes de Software:**
   * **Descripción:** No se considera la adquisición ni el licenciamiento de otros componentes de software comerciales o propietarios adicionales al stack de tecnologías preseleccionadas (por ejemplo, Angular, Ionic, Python/Django, MySQL).
   * **Impacto:** Se deben utilizar exclusivamente herramientas y tecnologías de código abierto o ya disponibles sin costo adicional para el proyecto. Esto limita la flexibilidad para introducir nuevas herramientas a medida que el sistema evoluciona, por lo que la selección de herramientas debe ser cuidadosamente optimizada desde el inicio.

## Otros antecedentes y consideraciones

1- Capacitación:

Es necesario contemplar programas de capacitación para el personal que operará el sistema, siendo este: formación en el uso de la interfaz de usuario, interpretación de datos generados y la forma de que estos puedan asistir a los clientes que quieran acceder a la plataforma.

2-Consideraciones financieras:

El uso del financiamiento debe ser gestionado de la manera más óptima posible, para evitar sobre costos

# **Vista de Casos de Uso y Escenarios de Calidad**

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

## Modelo de Casos de Uso

El modelo de casos de uso puede ser encontrado en la carpeta de “Casos de Uso”.

*1) Diagrama Casos de uso – Google Drive*. (s. f.). [1) Diagrama Casos de uso](https://drive.google.com/drive/u/1/folders/1s0IAmFvgGuJQ8v9fiNu-oiUrMRLTd6cy)

Y su especificación en esta carpeta:

*2) Especificación – Google Drive*. (s. f.). [2) Especificación](https://drive.google.com/drive/u/1/folders/1ktlcBmQfEfYV7ai_nuuU_xWEQ7RxILXt)

## Especificación de Casos de Uso Relevantes

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

* Su impacto en los usuarios.
* Su frecuencia de uso.
* Seguridad y Confianza.
* Funcionalidad central del negocio.
* Requisitos para el escalado.

A continuación se listan los casos de uso relevantes, los cuales pueden ser encontrados con su especificación detallada en el documento “Casos de Uso”.

| **Código** | **Nombre** | **Actores** | **Prioridad** |
| --- | --- | --- | --- |
| RF-10 | Registro de Usuario | Proveedor | Alta |
| RF-11 | Inicio de sesión | Proveedor, Administrador | Alta |
| RF-12 | Autenticación doble factor | Proveedor, Administrador | Media |
| RF-13 | Búsqueda de Producto | Cliente | Alta |
| RF-19 | Carro de compra | Cliente | Alta |
| RF-29 | Integración de sistemas | Cliente | Alta |
| RF-32 | Catálogo de producto | Cliente | Alta |
| RF-27 | Sistema de calificación a productos y vendedores | Cliente | Media |
| RF-33 | Creación de reportes inteligentes | Proveedor, Administrador | Baja |

## Especificación de los Escenarios de Calidad Relevantes

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

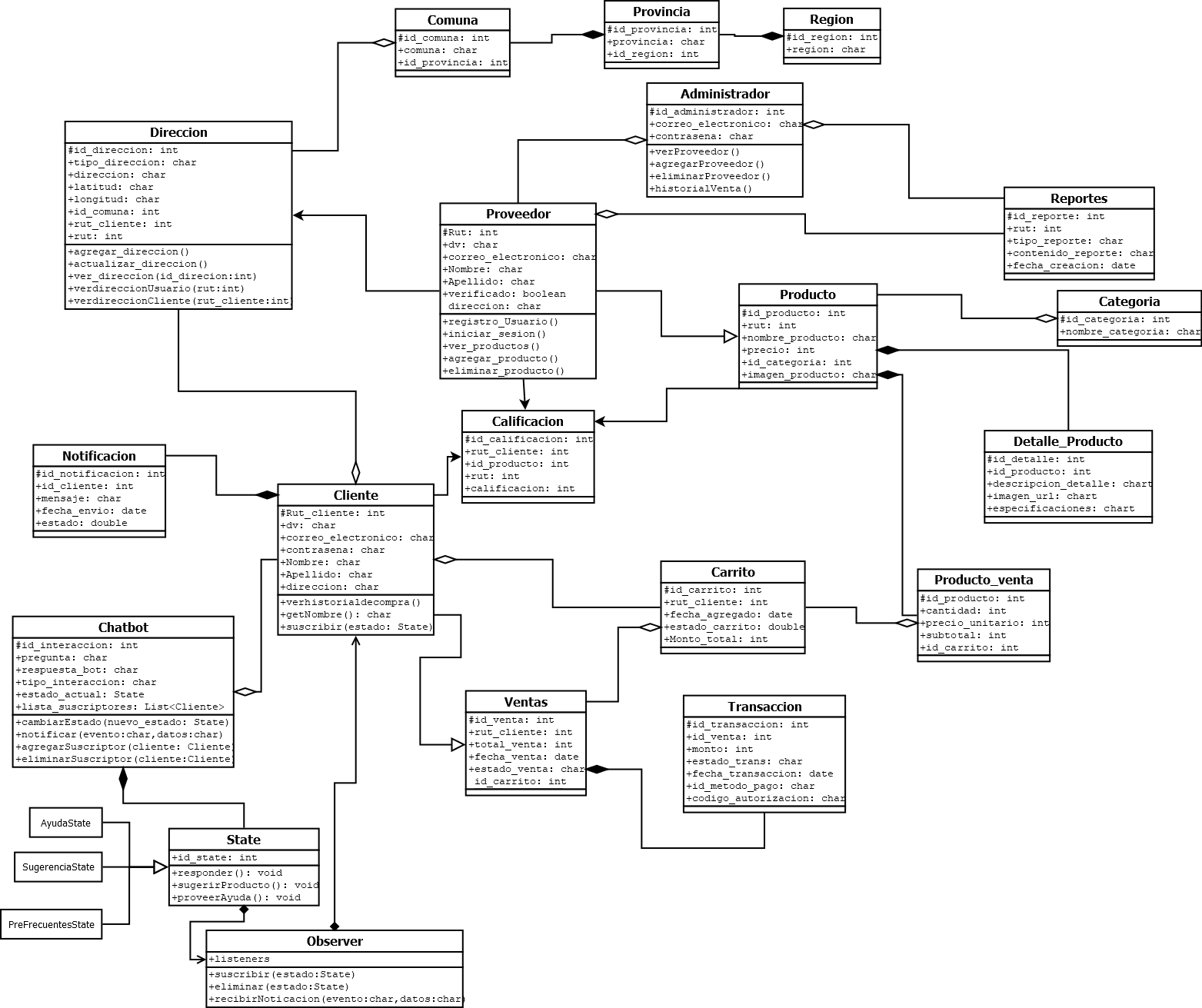
1. **Rendimiento**:
   1. **Escenario**: Tiempo de respuesta en la búsqueda de productos
      1. **Contexto**: Un cliente busca productos utilizando palabras clave o filtros en el catálogo.
      2. **Estímulo**: El cliente realiza una búsqueda en horas pico (alta carga en el sistema).
      3. **Respuesta**: El sistema debe mostrar los resultados en menos de 2 segundos.
      4. **Métrica**: Tiempo de respuesta 2 segundos para el 85% de las búsquedas bajo carga normal y alta.
   2. **Escenario**: Procesamiento del pago
      1. **Contexto**: El cliente procede a pagar después de agregar productos al carro de compras.
      2. **Estímulo**: Se activa el sistema de pago integrado con Transbank.
      3. **Respuesta**: El pago debe ser procesado en menos de 3 segundos.
      4. **Métrica**: Tiempo de procesamiento del pago ≤ 3 segundos en el 80% de las transacciones.
2. **Seguridad:**
   1. **Escenario**: Autenticación doble factor (2FA)
      1. **Contexto**: Un cliente o proveedor inicia sesión con doble autenticación activada.
      2. **Estímulo**: El usuario introduce sus credenciales y activa la autenticación de segundo factor.
      3. **Respuesta**: El sistema valida el segundo factor de autenticación en menos de 5 segundos.
      4. **Métrica**: Validación de 2FA exitosa en menos de 5 segundos en el 80% de los casos.
   2. **Escenario**: Protección de datos personales
      1. **Contexto**: Un administrador accede al sistema para gestionar cuentas o datos sensibles.
      2. **Estímulo**: El administrador solicita modificar datos de usuarios o proveedores.
      3. **Respuesta**: El acceso y modificación de datos sensibles solo se permiten con permisos válidos.
      4. **Métrica**: 80% de las operaciones con datos sensibles deben requerir autenticación y autorización válidas.
3. **Usabilidad**:
   1. **Escenario:** Facilidad de uso en el carro de compras
      1. **Contexto:** Un cliente usa el sistema móvil para agregar productos al carro y proceder al pago.
      2. **Estímulo:** El cliente interactúa con la interfaz de usuario móvil.
      3. **Respuesta:** El cliente debe poder completar el proceso de compra en menos de 5 pasos.
      4. **Métrica:** Completar el proceso de compra en ≤ 5 pasos para el 85% de los usuarios.
   2. **Escenario: Soporte multilingüe**
      1. **Contexto:** Un cliente extranjero accede a la plataforma y selecciona otro idioma en la opción de idiomas.
      2. **Estímulo**: El cliente cambia el idioma del sistema.
      3. **Respuesta**: La interfaz y las respuestas automáticas del chatbot deben cambiar de idioma sin retrasos perceptibles.
      4. **Métrica**: El cambio de idioma debe realizarse en ≤ 2 segundos, con soporte para el 80% del contenido en los idiomas habilitados.
4. **Escalabilidad**:
   1. **Escenario**: Soporte para múltiples usuarios concurrentes
      1. **Contexto**: Durante una oferta especial, la plataforma recibe un alto volumen de tráfico.
      2. **Estímulo**: Se conectan 1,000 usuarios concurrentes buscando productos y procesando pagos.
      3. **Respuesta**: El sistema debe manejar la carga sin tiempos de respuesta elevados ni caídas.
      4. **Métrica**: Soporte para 1,000 usuarios concurrentes con un tiempo de respuesta ≤ 3 segundos en el 85% de las operaciones.
   2. **Escenario**: Expansión de proveedores y productos
      1. **Contexto**: El número de proveedores y productos en el catálogo se duplica.
      2. **Estímulo**: Un incremento masivo de productos y proveedores en la plataforma.
      3. **Respuesta**: La plataforma debe gestionar el aumento sin ralentización perceptible en la navegación y búsqueda.
      4. **Métrica**: El tiempo de búsqueda de productos debe mantenerse en ≤ 2 segundos para el 85% de las consultas, con un catálogo duplicado.
5. **Mantenibilidad**:
   1. **Escenario**: Facilidad de actualización del catálogo de productos
      1. **Contexto**: Un proveedor actualiza o elimina productos desde su cuenta en la plataforma.
      2. **Estímulo**: El proveedor agrega, modifica o elimina un producto.
      3. **Respuesta**: Los cambios deben reflejarse en tiempo real en la interfaz del cliente.
      4. **Métrica**: Actualizaciones reflejadas en ≤ 1 segundo para el 80% de los productos modificados.
   2. **Escenario**: Diagnóstico y resolución de errores
      1. **Contexto**: El equipo de desarrollo detecta un error en el procesamiento de pagos.
      2. **Estímulo**: Se genera un reporte de error a través del sistema de monitoreo.
      3. **Respuesta**: El equipo de soporte debe identificar y resolver el error en menos de 1 hora.
      4. **Métrica**: Resolución de errores críticos en ≤ 1 hora en el 80% de los incidentes.
6. **Disponibilidad**:
   1. **Escenario**: Alta disponibilidad del sistema
      1. **Contexto**: El sistema debe estar disponible para clientes, proveedores y administradores las 24 horas del día.
      2. **Estímulo**: Se produce una falla de hardware o red en los servidores.
      3. **Respuesta**: El sistema debe recuperarse o redirigir las operaciones a servidores de respaldo en menos de 5 minutos.
      4. **Métrica**: Tiempo de inactividad ≤ 5 minutos con una disponibilidad del 85% anual.

# Vista Lógica

A continuación se presenta una vista lógica de la aplicación expresado en dos diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (Clases), y otra vista que representa la parte dinámica (componentes y conectores).

## Parte Estructural

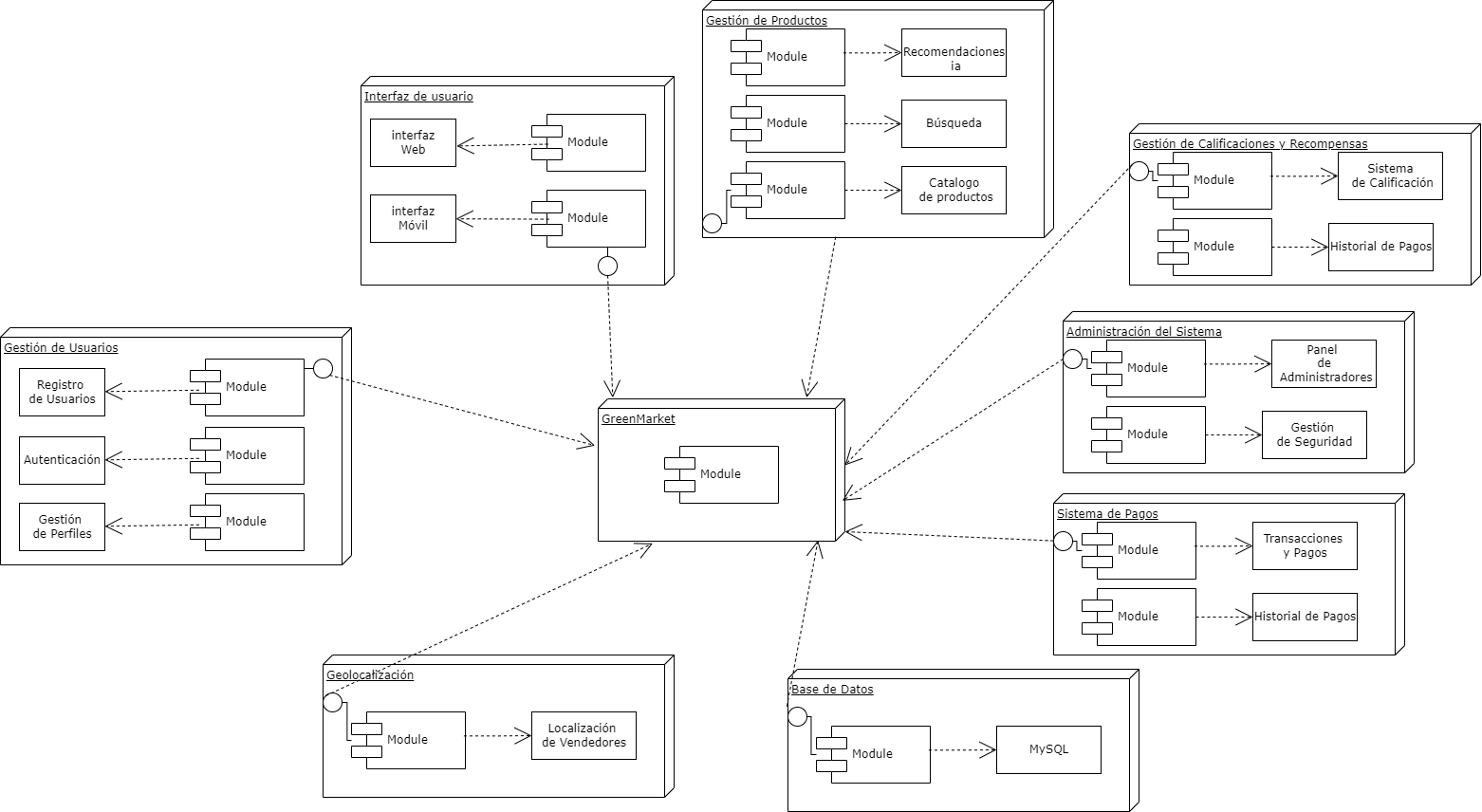
En el siguiente diagrama de Clases se observa los principales módulos

**

**Ilustración 1: Diagrama de Clases**

## Parte Dinámica

La parte dinámica



**Ilustración 2: Diagrama de Componentes y Conectores**

# Vista de Procesos

A continuación se muestra una vista de proceso que seguirá el proyecto:

Estas se encontraran en la siguiente carpeta para mantener un orden:

*3) Diagrama de actividad – Google Drive*. (s. f.). [3) Diagrama de Actividad](https://drive.google.com/drive/u/1/folders/1EOfxRYtU2vxwehJJPlsSL1oj5stxSNk3)

Además de poder ver la secuencia que tendrá cada actividad, esta se encontraran en la siguiente carpeta:

*4) Diagrama de secuencia – Google Drive*. (s. f.). [4) Diagrama de secuencia](https://drive.google.com/drive/u/1/folders/1eefrkQNKf3uA1Lxk5pjlQrd_4-ymE74U)

**Ilustración 3: Diagrama de Procesos(Diagramas Actividad y de secuencia)**

# Vista de Implementación

En este apartado se verán diferentes diagramas de la vista de desarrollo:

Mockups desarrollados:

*Untitled*. (s. f.). Figma. <https://www.figma.com/proto/V7PPcEcNJWSZT4TbkcUvZK/Untitled?node-id=39-204&t=KMvdpnL9PmE5reVz-1&starting-point-node-id=39%3A204>

Diagrama de Paquete:

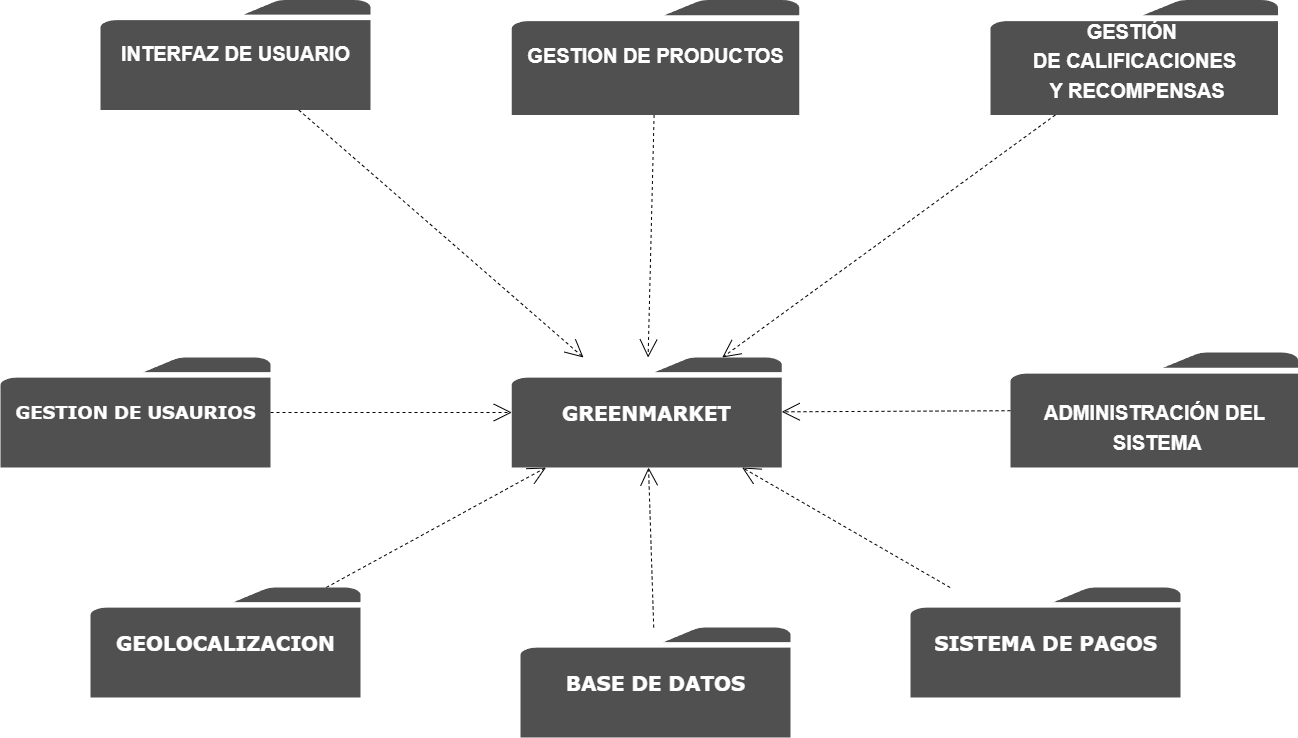


Diagrama de componente:

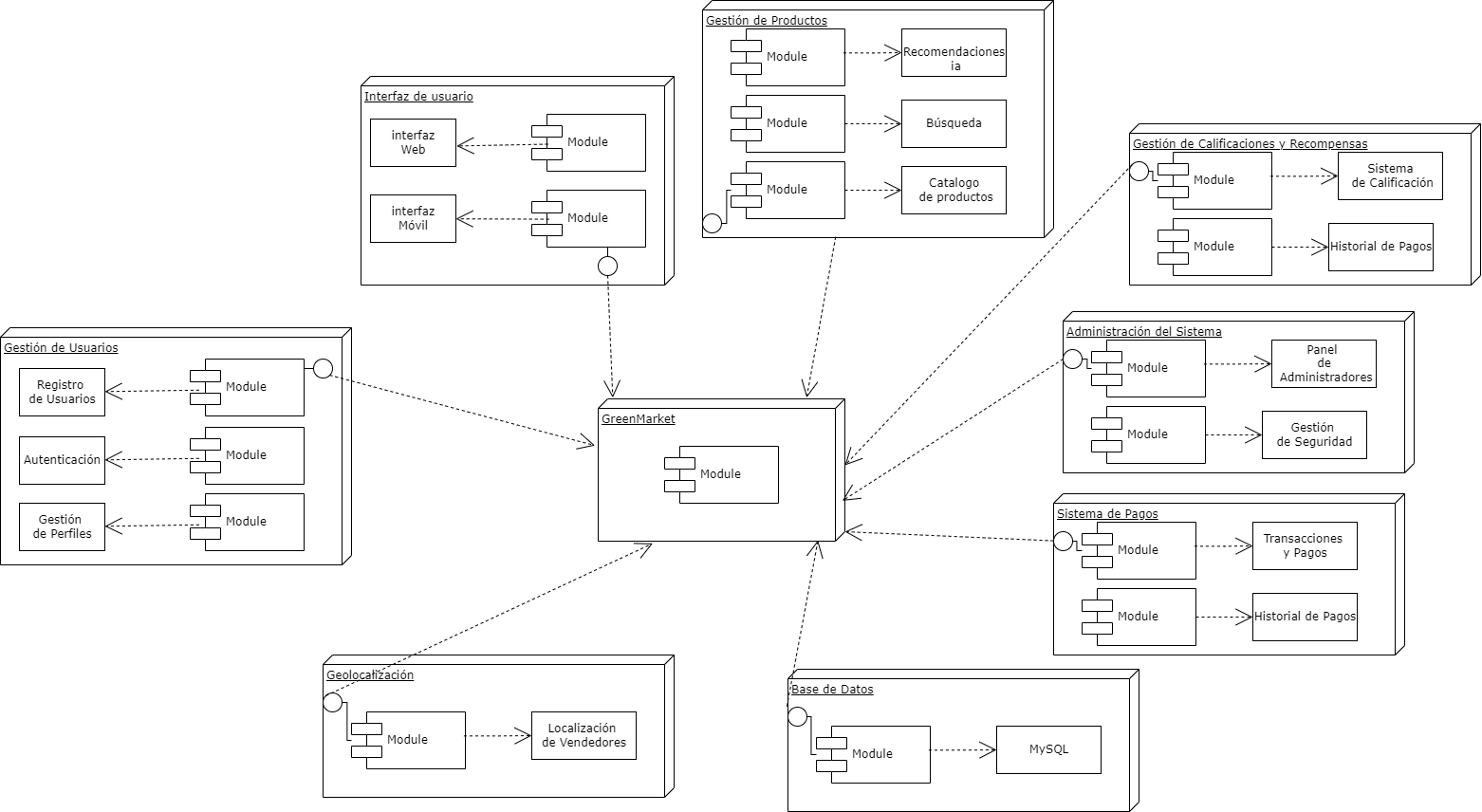


Diagrama de Clase:

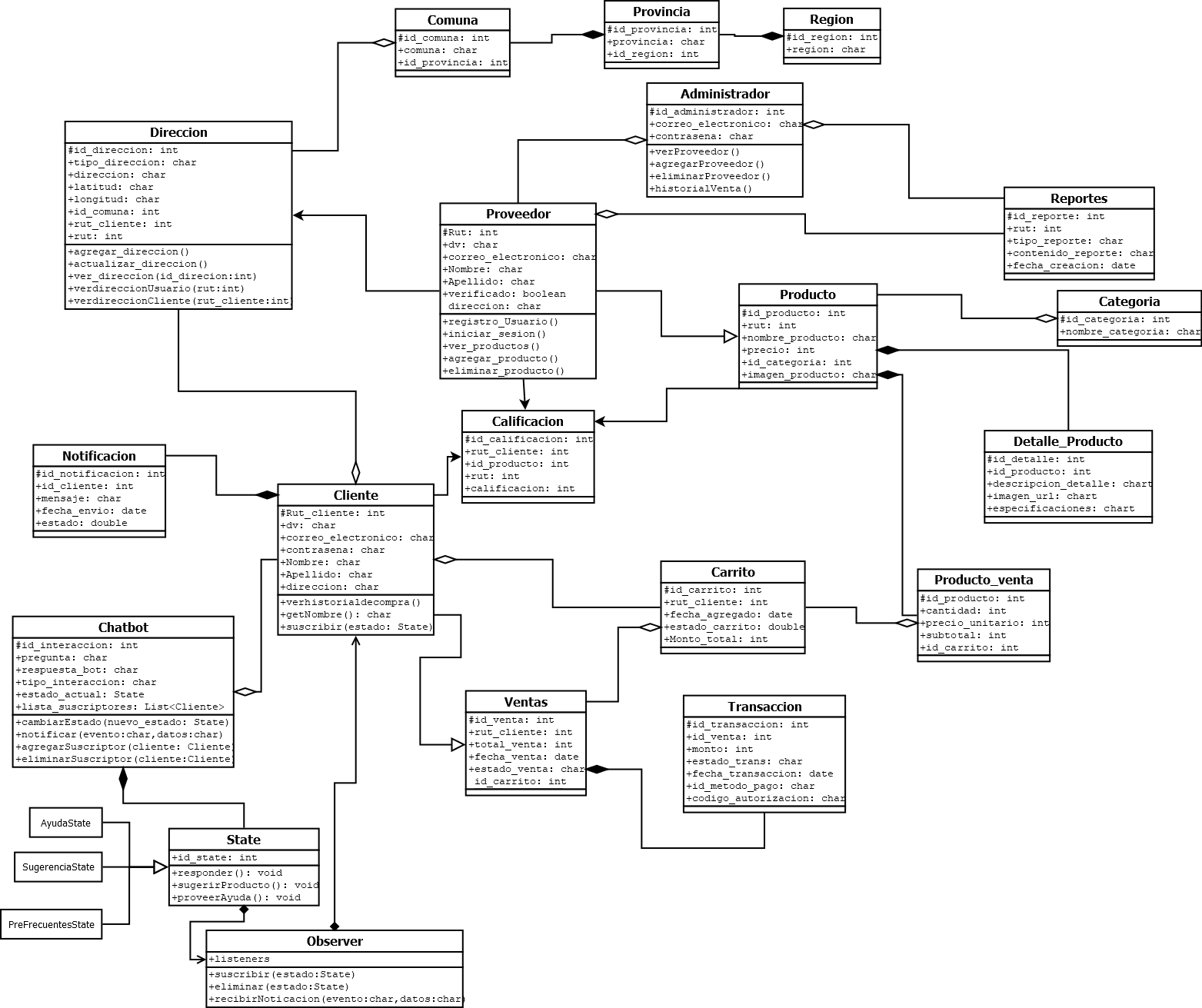
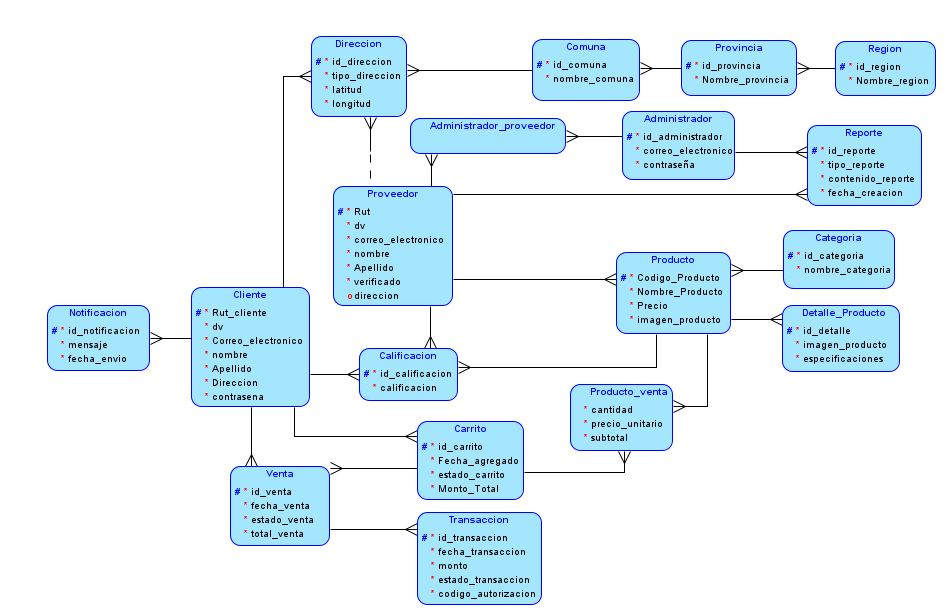
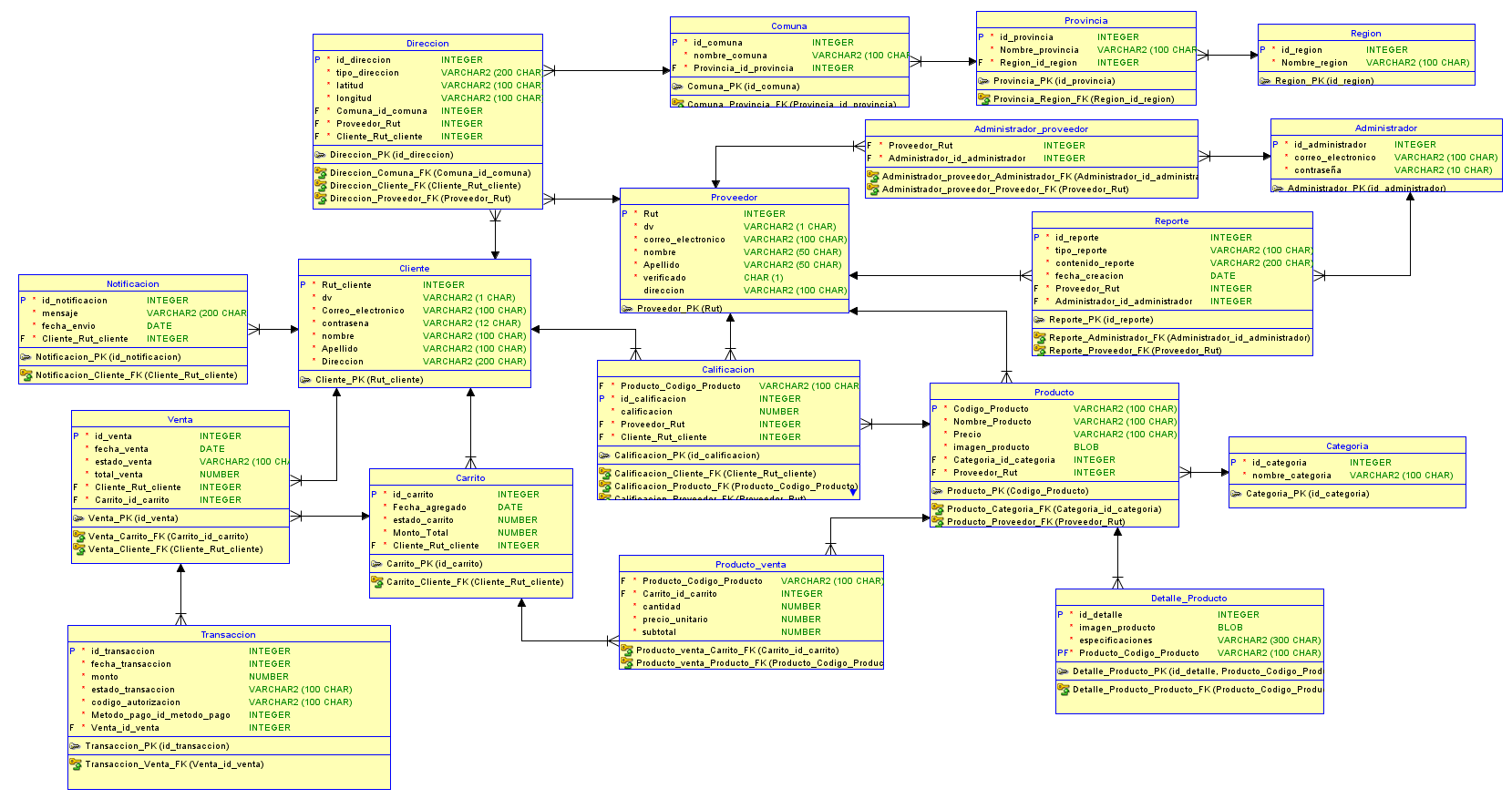


Diagrama de BBDD:

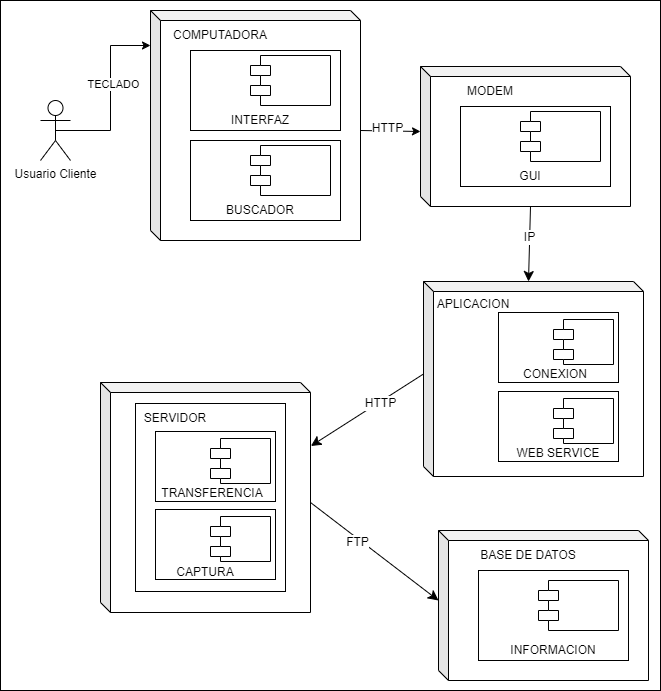




**Ilustración 4: Vista de Implementación**

# Vista de Despliegue

En esta vista se despliegan los nodos que participan con el sistema. Los nodos principales son los nodos Servidor de Integración. Características a continuación:

****

**Ilustración 6: Diagrama de Despliegue**

# **Decisiones de Diseño y Selección de Alternativas**

Decisiones de Diseño y Selección de Alternativas para GreenMarket

Las principales decisiones arquitectónicas para **GreenMarket** está teniendo en cuenta la restricción de **Tiempo de Construcción**. se prioriza la adopción de una arquitectura bien conocida y de bajo riesgo. Esta elección asegura un desarrollo rápido y facilita el mantenimiento durante y después de la ejecución del proyecto.

La arquitectura se modularizar con dos objetivos principales:

1. **Separación de Responsabilidades**: Al dividir el sistema en módulos distintos, se permite la paralelización en el desarrollo de cada componente, lo que asegura que diferentes equipos o miembros puedan trabajar simultáneamente en diversas partes del sistema sin generar dependencias críticas entre ellos.
2. **Cumplimiento de la Restricción de Infraestructura**: Dado que la aplicación debe ejecutarse en servidores replicados con balanceo de carga, y combinando esto con el escenario de **Tolerancia a Fallos**, se optó por una estructura de aplicación web.

# **Análisis de Reutilización**

En este caso el proyectos se realizará desde cero, usando material de anteriores proyectos para crear las funcionalidades nuevas y reciclar algunos documentos para poder tener una mejor idea de como poder crear un proyecto asegurando mejor calidad ya que se aprenderá de lo que se hizo anteriormente para poder mejorarlo.

* Uso de proyecto de integración de plataformas
* Uso de Mockups de tiendas de bicicletas y gestión ágil
* Diccionario de riesgos de Gestión de riesgos
* Proyecto de programación de aplicaciones móviles, creación de aplicación para tomar asistencia usando un QR
* Uso de patron de diseño Observer y State aprendido en la clase de arquitectura

Uso de frameworks conocidos como angular y ionic, que ayudarán a facilitar la creación de la plataforma, debido a que ya se maneja de mejor manera y se dispone de mucha información con la cual poder desarrollar las funciones de la aplicaion.

Uso de python para la creación del JB, la cual ayudará a que sea más fácil de programar al ser un lenguaje rápido y sencillo de aprender.

Se reutilizaron materiales de proyecto pasados para realizar, funcionalidades del nuevo proyecto, además se creará la funcionalidades acordes a la aplicación móvil, como lo es la geolocalización