**SmartFlow – Sistema de Gestión y Reservas (DAS) Documento Arquitectura Sistema**

**Versión 1.0**

**Identificación de Documento**

|  |  |
| --- | --- |
| **Identificación** | Dashboard (SmartFlow) |
| **Proyecto** | SmartFlow – Sistema de Gestión y Reservas |
| **Versión** | 1.0 |

|  |  |
| --- | --- |
| **Documento mantenido por** | Juan Castillo Loyola / Mallely Calfilaf Aguayo |
| **Fecha de ultima revisión** | 16/10/2025 |
| **Fecha de próxima revisión** | 23/10/2025 |

|  |  |
| --- | --- |
| **Documento aprobado por** | Duoc UC – Representante Académico |
| **Fecha de última aprobación** |  |

**Historia de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 16/10/2025 | 1.0 | Se comienza documento | Mallely Calfilaf |
| 16/10/2025 | 1.0 | Se cargan documentos | Juan Castillo |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**1**](#_heading=h.q4mxcjq48nyf) **Introducción 3**

[1.1](#_heading=h.y4l3oqz5ffos) Contexto del Problema 3

[1.2](#_heading=h.qw7jeif81zq) Propósito 3

[1.3](#_heading=h.fudt9fc1urqy) Ámbito 3

[1.4](#_heading=h.i8q9hh6kw251) Definiciones, acrónimos y abreviaciones 3

[1.5](#_heading=h.x4ygtxgvt0ll) Referencias 3

[1.6](#_heading=h.927ki2ww3xrs) Resumen ejecutivo 3

[1.7](#_heading=h.3foxvwe17p2d) Representación 4

[**2**](#_heading=h.u5pdn1bdf3j5) **Metas y Restricciones de la Arquitectura 5**

[2.1](#_heading=h.3wt0ksmpqegv) Metas de la arquitectura 5

[2.2](#_heading=h.wwlk4s5bvjma) Restricciones de la Arquitectura 5

[2.3](#_heading=h.82yjmnthf13l) Otros antecedentes y consideraciones 5

**3** **Vista de Escenarios 6**

[3.1](#_heading=h.vzrh9u3gbkzr) Modelo de Casos de Uso 6

[3.2](#_heading=h.6uqcsp6o1nl) Casos de Usos Extendidos 6

[3.3](#_heading=h.2le2xca0vm2q) Especificación de los Escenarios de Calidad Relevantes 7

[**4**](#_heading=h.91cds59f4d5j) **Vista de Procesos 8**

[**5**](#_heading=h.xkzwf4u4b83w) **Vista Lógica 9**

[5.1](#_heading=h.r81wnxcueq2i) Parte Estructural ( Diagrama de Clases y Diagrama Relacional) 9

[*5.1.1*](#_heading=h.omsda9vjqi26) *Descripción de Clases 9*

[*5.1.2*](#_heading=h.yj8st7mwnn1q) *Descripción de Tablas 10*

[5.2](#_heading=h.b9rr7c97u7e2) Parte Dinámica (Diagrama de Secuencias) 11

[**6**](#_heading=h.xk0gbxtb8u6f) **Vista de Desarrollo o Despliegue 12**

[**7**](#_heading=h.25tj2iakg4wq) **Vista Fisica 13**

[**8**](#_heading=h.ys5wxme2nnnh) **Decisiones de Diseño y Selección de Alternativas 14**

[**9**](#_heading=h.ldff5jbkaker) **Análisis de Reutilización 15**

1. **Introducción**
   1. **Contexto del Problema**

El presente documento describe el diseño, estructura funcional y visual del Dashboard (DAS) perteneciente al sistema SmartFlow – Sistema de Gestión y Reservas.

Su propósito es centralizar las métricas, indicadores y accesos directos del sistema, facilitando el monitoreo general por parte del Administrador y los usuarios con privilegios avanzados.

El Dashboard constituye el punto de entrada principal al sistema luego de la autenticación, y refleja información relevante en tiempo real sobre:

1. Reservas activas y completadas.
2. Usuarios registrados y roles asignados.
3. Servicios disponibles y su nivel de ocupación.
4. Notificaciones automáticas generadas.
5. Reportes y auditorías recientes.

Este documento forma parte de la fase de desarrollo del sistema (Sprint 4) y servirá de guía tanto para el diseño UI/UX como para la implementación técnica entro del entorno .NET Core 8 con Razor Pages y SQL Server.

**Propósito**

Definir el diseño funcional y visual del Dashboard (DAS) del sistema SmartFlow – Sistema de Gestión y Reservas, especificando su estructura, objetivos, métricas presentadas y criterios de usabilidad.

Este documento busca establecer los lineamientos necesarios para la correcta implementación del Dashboard dentro del entorno de desarrollo .NET Core 8 (Razor Pages), asegurando la coherencia con los requerimientos funcionales del sistema y el estándar de documentación de Duoc UC.

El Dashboard permitirá al Administrador acceder a información resumida y visual del sistema, optimizando la toma de decisiones mediante indicadores de control, reportes dinámicos y accesos directos a los módulos principales.

**Ámbito**

El Dashboard contempla la visualización de datos en tiempo real provenientes de los módulos:

Usuarios: cantidad total, activos/inactivos y roles asignados.

Reservas: número de reservas activas, completadas y canceladas.

Servicios: estado de disponibilidad y demanda por servicio.

Notificaciones: alertas automáticas enviadas por el sistema.

Auditoría: registro de acciones críticas del sistema (solo para administradores).

* 1. **Definiciones, acrónimos y abreviaciones**

|  |  |
| --- | --- |
| **ACRONIMO** | **DESCRIPCION** |
| *MVC* | Modelo-Vista-Controlador |
| *CRUD* | Create,Read,Update,Delete |
| *ORM* | Object Relational Mapping |
| *DAS* | Dashboard – Panel principal del sistema SmartFlow donde se visualizan métricas, reportes y accesos directos a módulos. |
| *ERS* | Especificación de Requerimientos de Software, documento base que define los requisitos funcionales y no funcionales del sistema. |
| *UI/UX* | Interfaz y Experiencia de Usuario. Diseño visual e interacción del sistema. |
| *SQL Server* | Sistema de gestión de bases de datos relacional utilizado en SmartFlow para almacenamiento y consultas. |
| *.NET Core* | Framework de desarrollo backend usado para implementar la lógica de negocio del sistema. |
| *Razor Pages* | Tecnología de desarrollo web usada para construir las vistas del sistema con .NET Core. |
| *EF Core* | Entity Framework Core. ORM utilizado para la comunicación entre la base de datos y el código C#. |
| *KPI* | Key Performance Indicator. Indicador clave de rendimiento utilizado en el Dashboard para medir resultados. |
| *PDF/XLS* | Formatos de exportación de reportes soportados por el sistema. |

* 1. **Referencias**

A continuación, se listan las referencias a otros documentos :

* **ERS**
* **Documento de Casos de uso Extendido**
* **Planilla de Requerimientos Funcionales y No Funcionales**

* 1. **Resumen ejecutivo**

El sistema “Dulce Arte” fue diseñado bajo una arquitectura modular, escalable y orientada a servicios, implementada en Python/Django y desplegada sobre un entorno web multiusuario.

La arquitectura permite integrar módulos de ventas, productos y logística con seguridad y trazabilidad, priorizando la mantenibilidad, reutilización de componentes y rendimiento.

* 1. **Representación**

La arquitectura del sistema <<Nombre del proyecto y/o Aplicación>> está representada siguiendo el enfoque de del framework 4+1 y las recomendaciones del proceso unificado. Las vistas incluidas en esta versión del documento son:

* **Vista de Escenarios**: 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 Procesos**: Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
* **Vista Lógica**: Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.
* **Vista de Desarrollo o Despliegue**: Describe los componentes de deployment construidos y sus dependencias.
* **Vista Fisica**: Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.

1. **Metas y Restricciones de la Arquitectura**
   1. **Metas de la arquitectura**

|  |  |
| --- | --- |
| Atributo de calidad | Descripción |
| Desempeño | Respuesta óptima en tiempo real durante operaciones CRUD y checkout. |
| Escalabilidad | Posibilidad de integrar modulos futuros (Api Movil, dashboards, analitica). |
| Seguridad | Control de acceso por roles, autenticacion django, https, y manejo seguro de datos personales. |
| Mantenibilidad | Codigo modular, basado en ORM y plantillas reutilizables. |
| Disponibilidad | Sistema Accesible 24/7 con respaldo de base de datos y logs de errores. |
| Usabilidad | Interfaz intuitiva y diseño responsive con Bootstrap |

* 1. **Restricciones de la Arquitectura**

Existen restricciones que han sido levantadas con los stakeholders, las cuales se presentan a continuación:

* Tiempo de construcción: Plazo estimado de 12 semanas.
* Infraestructura:
* Lenguaje y framework: Python 3.12 + Django 5.0.
* Sin licencias propietarias: se prioriza software libre.
  1. **Otros antecedentes y consideraciones**

El sistema se apoya en el framework Django, el cual:

* Implementa un ORM robusto para el manejo relacional.
* Incluye sistema de autenticación nativo.
* Soporta modularización y separación de responsabilidades.

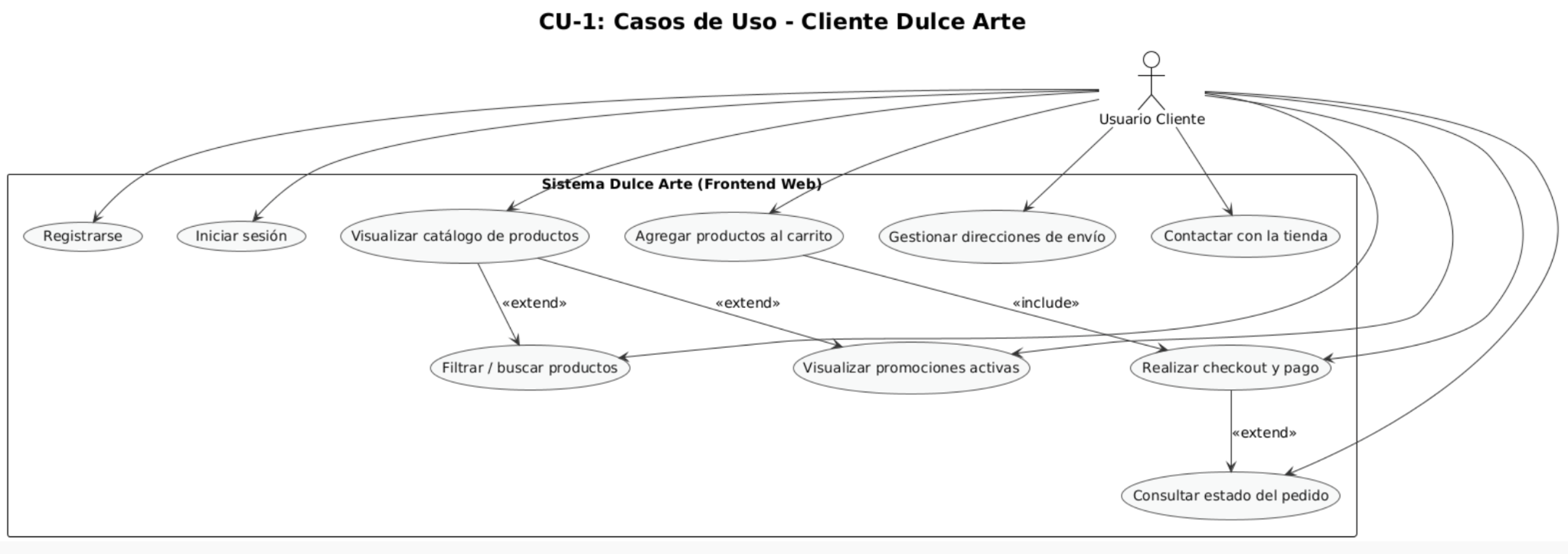
1. **Vista de Escenarios**

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.

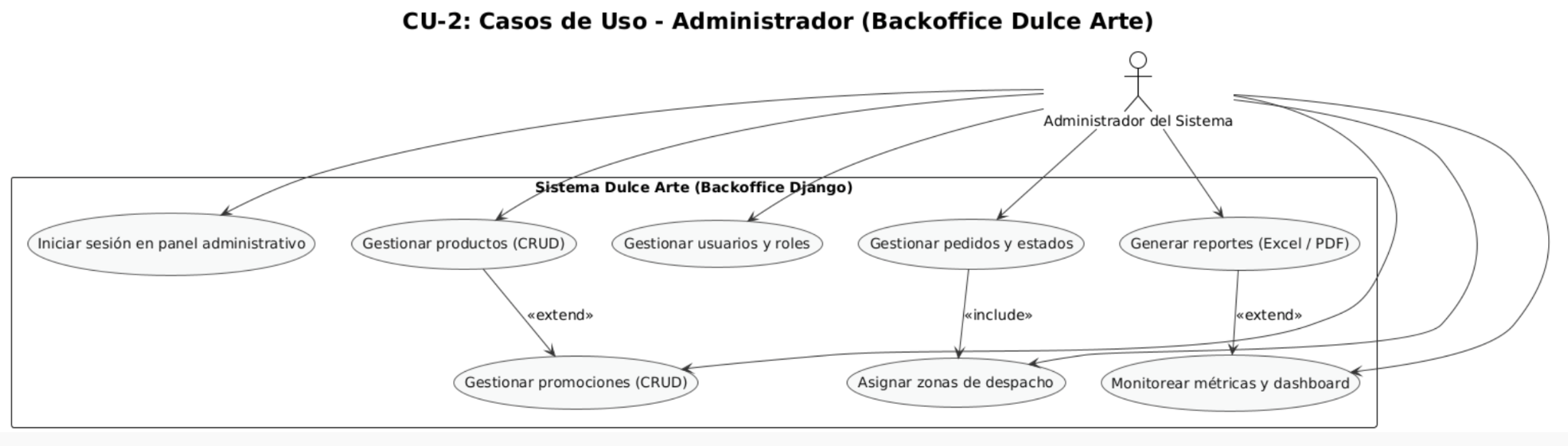
* 1. **Modelo de Casos de Uso**

Agregar el modelo de caso uso general del sistema

**Ilustración 1: Diagrama de Caso Uso General del Sistema Cliente**

****

**Ilustración 2: Diagrama de Caso Uso Administración del sistema**

****

* 1. **Casos de Usos Extendidos**

Los casos de uso considerados son los más relevantes para el desarrollo de la arquitectura. Se adjunta el documento o planilla caso uso.

**Adjuntar la planilla caso uso extendido**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Nombre** | **Actores** | **Prioridad** |
| CU-001 | Registrar usuario en el sistema | Cliente | Media |
| CU-002 | Autenticación de usuario | Todos los usuarios |  |
| CU-003 | Asignación de roles a usuario | Administrador |  |
| CU-004 | Visualizar catálogo de productos | Cliente, Administrador |  |
| CU-005 | Filtrar productos por categoría | Cliente |  |
| CU-006 | Agregar productos al carrito de compras | Cliente |  |
| CU-007 | Gestionar cantidades en el carrito | Cliente |  |
| CU-008 | Gestionar pedidos realizados | Cliente, Administrador |  |
| CU-009 | Crear pedido personalizado | Cliente |  |
| CU-010 | Realizar pedido con selección de zona de despacho | Cliente |  |
| CU-011 | Seguimiento del estado del pedido | Cliente |  |
| CU-012 | Administrar sistema desde backoffice | Administrador |  |
| CU-004 | Generar reportes y exportar a PDF/Excel | Administrador |  |
| CU-004 | Enviar notificaciones por correo electrónico | Sistema (disparado por eventos); Administrador (configuración) |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* 1. **Especificación de los Escenarios de Calidad Relevantes**

Los siguientes escenarios de calidad fueron priorizados tras el análisis con los stakeholders del proyecto Dulce Arte, considerando los atributos más críticos: rendimiento, seguridad y disponibilidad.

|  |  |  |
| --- | --- | --- |
| Identificador: EC-01 | | |
| Escenario(s): | | El cliente agregar varios productos al carrito y procede al checkout simultáneamente con otros usuarios. |
| Atributos de Calidad relevantes: | | Rendimiento/Desempeño - Escalabilidad |
| Componentes del Escenario | Estímulos: | Múltiples usuarios realizan operaciones de compra y confirmación de pedido de forma concurrente. |
| Fuente del estimulo | Cliente web (usuario final) y procesos internos del servidor Django. |
| Ambiente: | Producción, bajo carga promedio-alta (campaña promocional). |
| Artefacto: | Módulo de carrito, pedido y zonaDespacho dentro del servidor de aplicación Django. |
| Respuesta: | El sistema procesa las transacciones concurrentes sin bloqueos ni errores, recalculando subtotales y zonas en tiempo real. |
| Medida de Respuesta | Tiempo promedio de respuesta menor a 2 segundos por solicitud HTTP; tolerancia máxima de 5% de peticiones con demora mayor a 3 segundos. |

|  |  |  |
| --- | --- | --- |
| Identificador: EC-02 | | |
| Escenario(s): | | Un usuario intenta acceder a la cuenta de otro cliente mediante un ataque de fuerza bruta o reutilización de sesión. |
| Atributos de Calidad relevantes: | | Seguridad-Confidencialidad-Integridad de datos |
| Componentes del Escenario | Estímulos: | Intentos consecutivos de autenticación con credenciales incorrectas o manipulación de sesión. |
| Fuente del estímulo | Usuario malintencionado externo. |
| Ambiente: | Plataforma desplegada en entorno web https con autenticación de django. |
| Artefacto: | Modulo de usuarios (Login/logout, middleware de seguridad, token CSRF). |
| Respuesta: | El sistema bloquea temporalmente la cuenta tras 3 intentos fallidos, registra la IP y emite notificación al usuario legitimo. La sesión no reutilizable expira automáticamente. |
| Medida de Respuesta | Tasa de éxito de mitigación de ataque menor al 99%; sin pérdida ni exposición de datos personales; auditoría registrada en log de eventos. |

|  |  |  |
| --- | --- | --- |
| Identificador: EC-03 | | |
| Escenario(s): | | Fallo temporal del servidor de base de datos durante la operación de pedido o despacho. |
| Atributos de Calidad relevantes: | | Disponibilidad, tolerancia a fallos, recuperabilidad |
| Componentes del Escenario | Estímulos: | Interrupción del servicio de base de datos o conexión perdida entre aplicación y motor PostgreSQL. |
| Fuente del estímulo | Error de red, reinicio de servidor o mantenimiento no planificado. |
| Ambiente: | Producción con clientes activos realizando pedidos o consultas. |
| Artefacto: | Backend Django (ORM, capa de persistencia y controladores de pedidos). |
| Respuesta: | El sistema detecta la falla, pausa la operación crítica, muestra mensaje de “servicio temporalmente no disponible” y reintenta conexión automáticamente. Una vez restablecido, sincroniza las transacciones pendientes. |
| Medida de Respuesta | Tiempo máximo de recuperación menor o igual a 5 minutos; pérdida de transacción menor o igual a 0; disponibilidad global mayor o igual 99%. |

|  |  |  |
| --- | --- | --- |
| Identificador: EC-04 | | |
| Escenario(s): | | Un cliente accede desde distintos dispositivos (móvil, tablet, notebook) para realizar un pedido y consultar el estado de entrega. |
| Atributos de Calidad relevantes: | | Usabilidad - Portabilidad - Accesibilidad |
| Componentes del Escenario | Estímulos: | Navegación simultánea desde dispositivos con distintos tamaños de pantalla y navegadores. |
| Fuente del estímulo | Cliente final que busca una experiencia fluida e intuitiva. |
| Ambiente: | Entorno de producción (página publica HTTPS) |
| Artefacto: | Frontend Bootstrap 5, plantilla django y hojas css responsive. |
| Respuesta: | El sistema ajusta automáticamente la interfaz, mantiene legibilidad y permite completar acciones sin errores de formato ni desbordes. |
| Medida de Respuesta | Tasa de tareas completadas sin error mayor o igual a 95%; tiempo medio de compra desde móvil menor o igual a 3 minutos. |

|  |  |  |
| --- | --- | --- |
| Identificador: EC-05 | | |
| Escenario(s): | | El equipo de desarrollo debe incorporar un nuevo módulo de “Programación de repartos” sin afectar las funciones ya operativas. |
|  | |  |
| Atributos de Calidad relevantes: | | Mantenibilidad - Modificabilidad - Reusabilidad |
| Componentes del Escenario | Estímulos: | Solicitud de ampliación de funcionalidad por parte del cliente (gerencia Dulce Arte). |
| Fuente del estímulo | Equipo de desarrollo interno. |
| Ambiente: | Entorno de desarrollo controlado (usando git, entorno visual y django modular). |
| Artefacto: | Código fuente del proyecto django (Módulos pedidos/ y carrito/), modelo de datos normalizado y API REST. |
| Respuesta: | El nuevo módulo se integra usando el ORM y las vistas genéricas, sin afectar otras funcionalidades ni producir errores en migraciones o deploys. |
| Medida de Respuesta | Tiempo maximo de implementación menor o igual a 3 días;  número de incidencias post-deploy =0; cobertura de pruebas unitarias mayor o igual a 90%. |

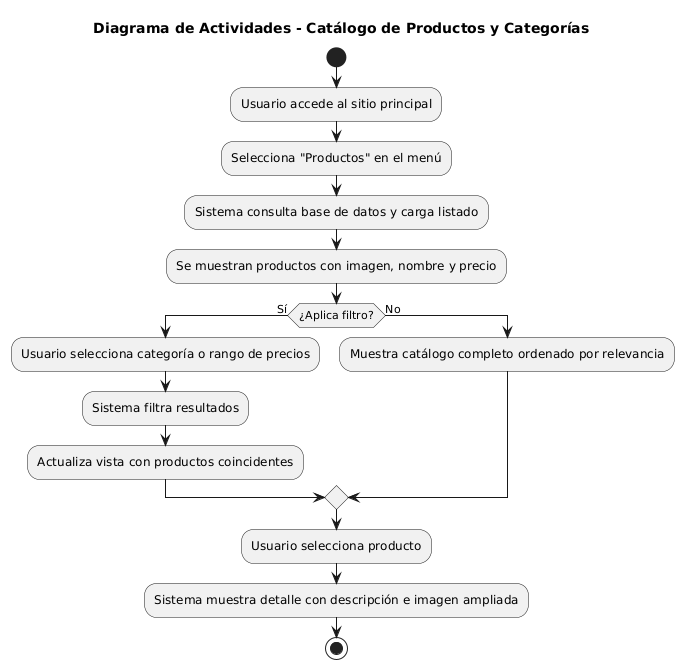
1. **Vista de Procesos**

**Ilustración 2: Diagramas de Actividades**

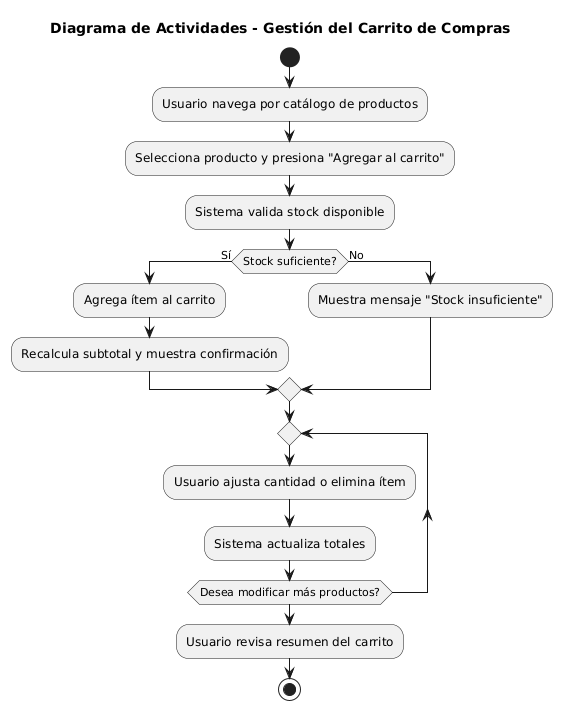
1. **Registro y autenticación de usuario.**



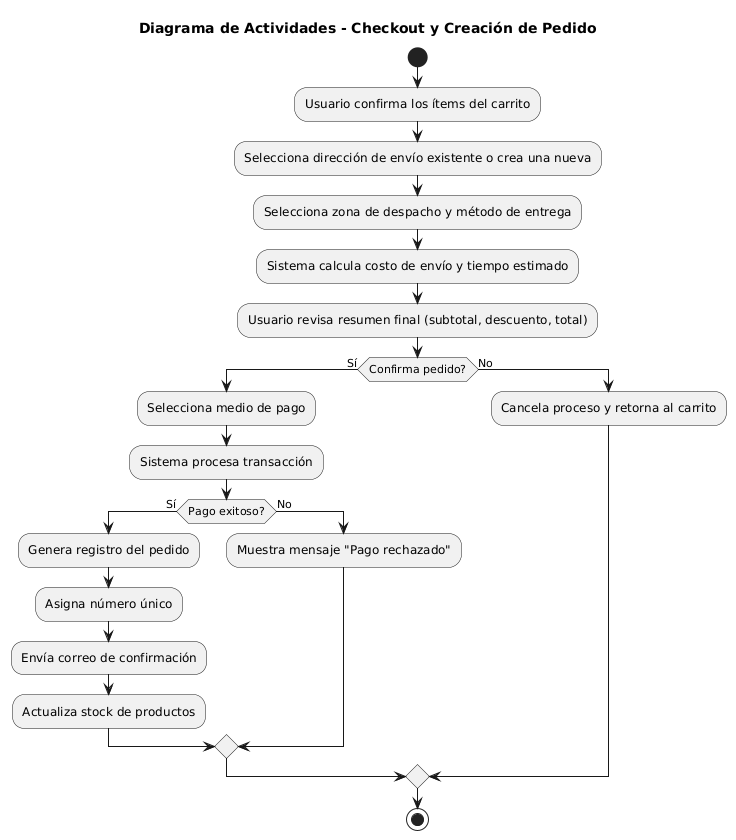
1. Catalogo de productos y categorias



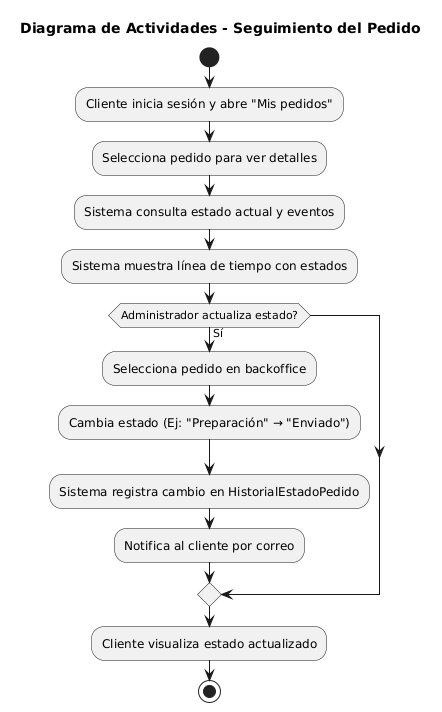
1. gestión del carrito de compra



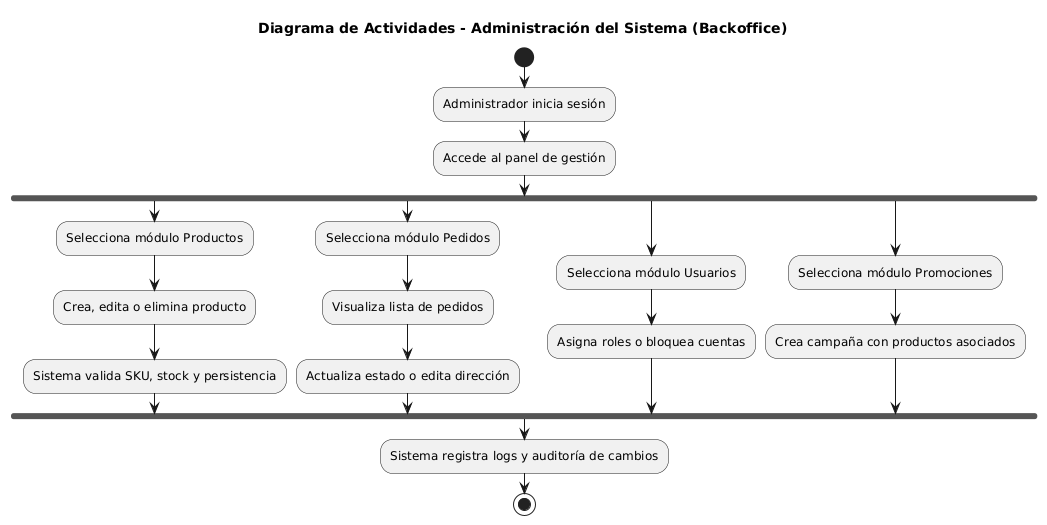
1. Checkout y creación del pedido.



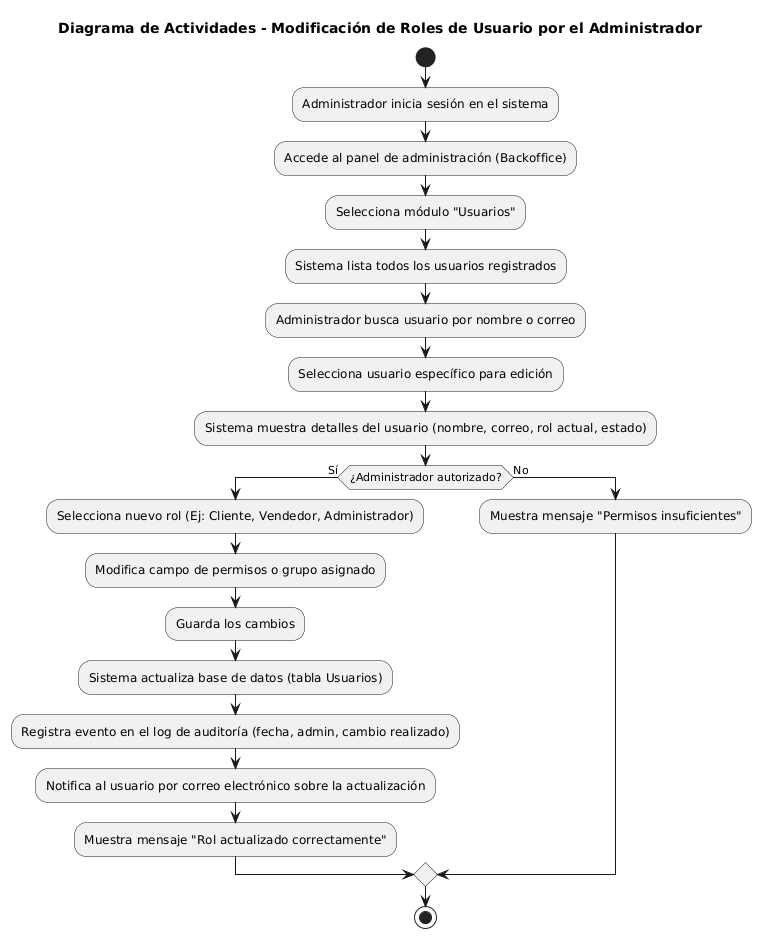
1. Seguimiento del pedido.

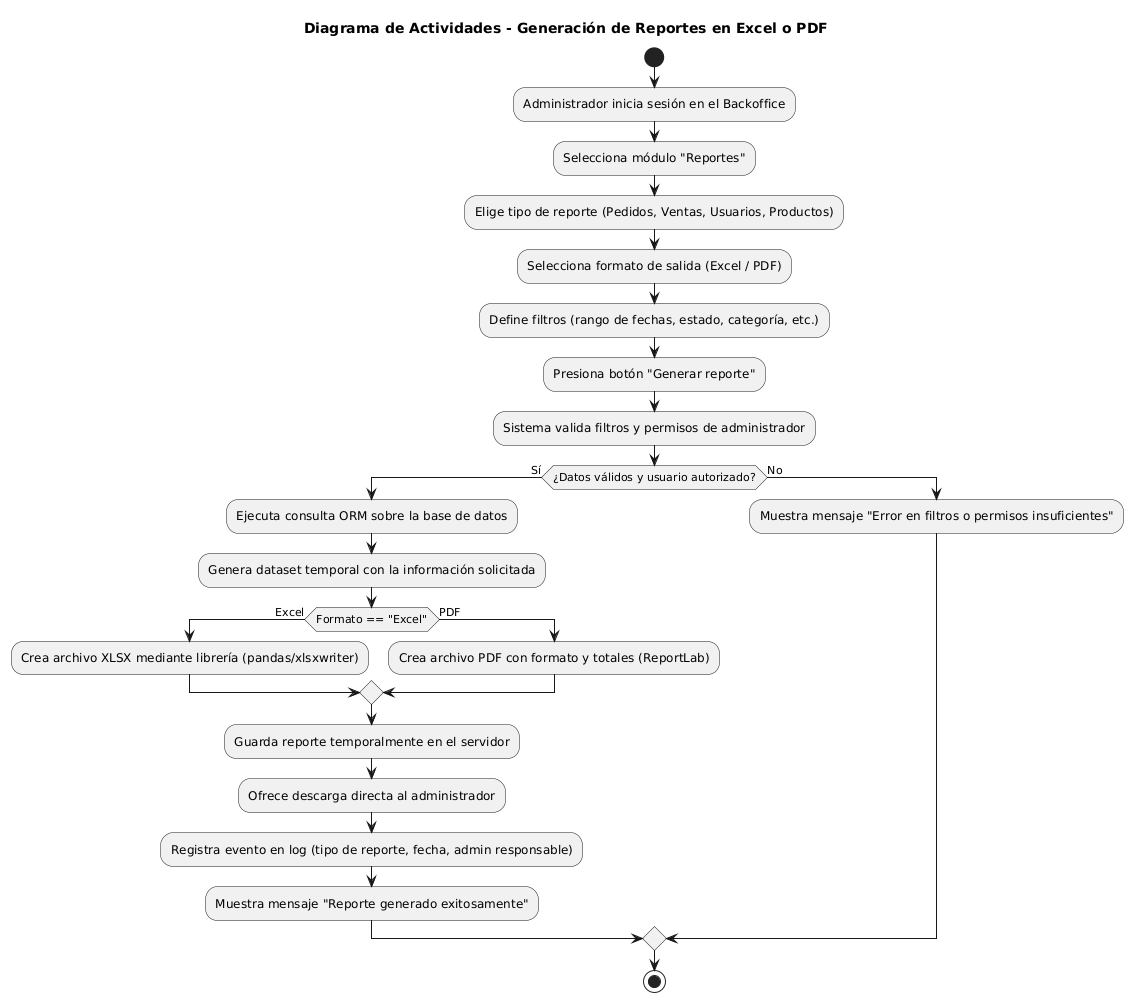


1. Administración del sistema (Back office)



1. Modificación de roles de usuario (Back office)



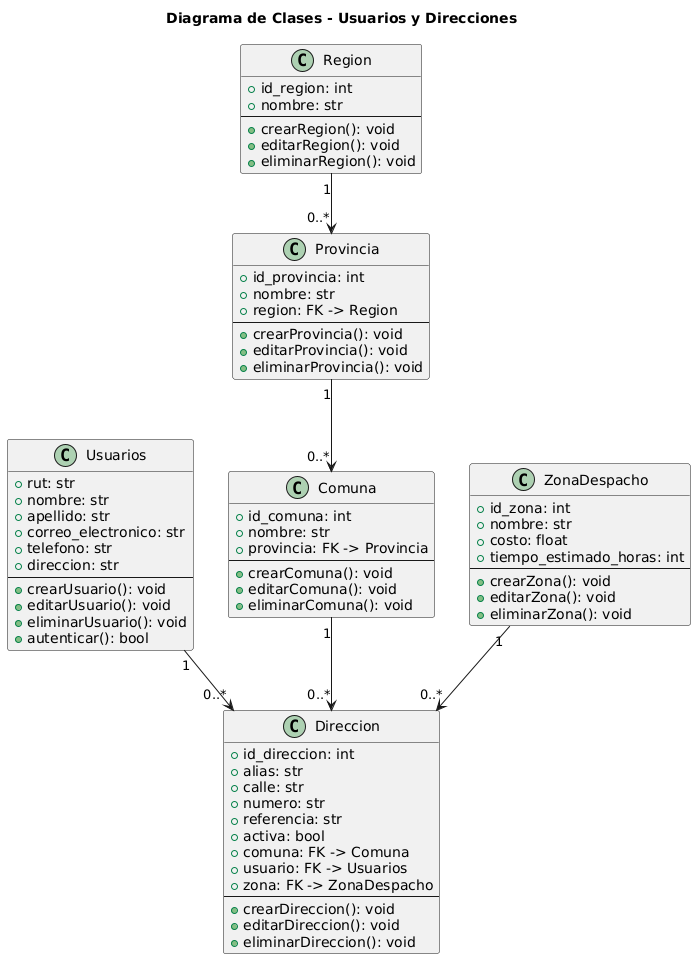
1. generación de reportes (excel/pdf)
2. 
3. **Vista Lógica**

A continuación, se presenta una vista lógica de la aplicación expresado en tres diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (clases) y a la base de datos (modelo relacional).otra vista que representa la parte dinámica (secuencias).

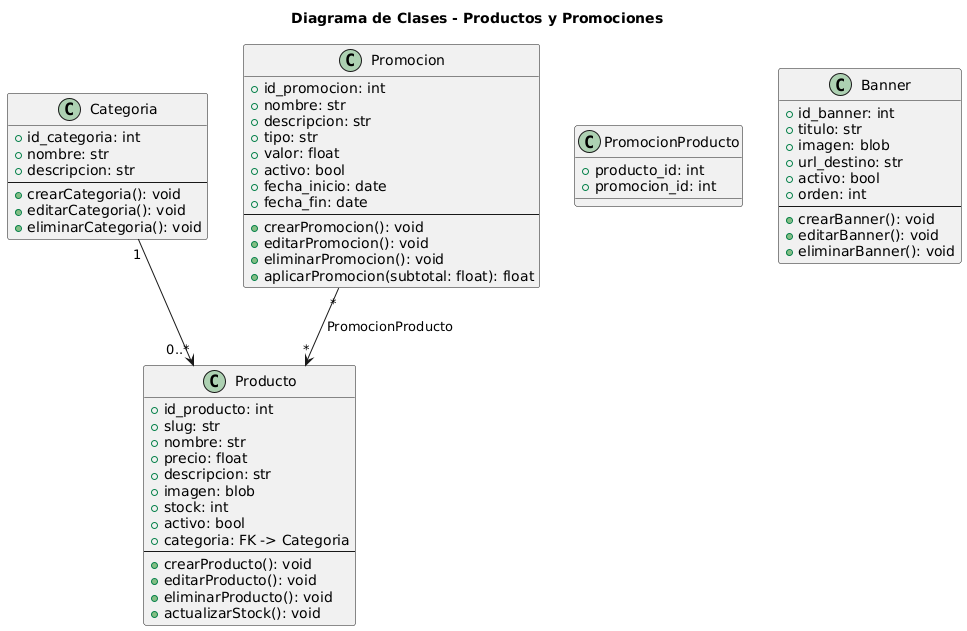
* 1. **Parte Estructural ( Diagrama de Clases y Diagrama Relacional)**

**Ilustración 3: Diagrama de Clases**

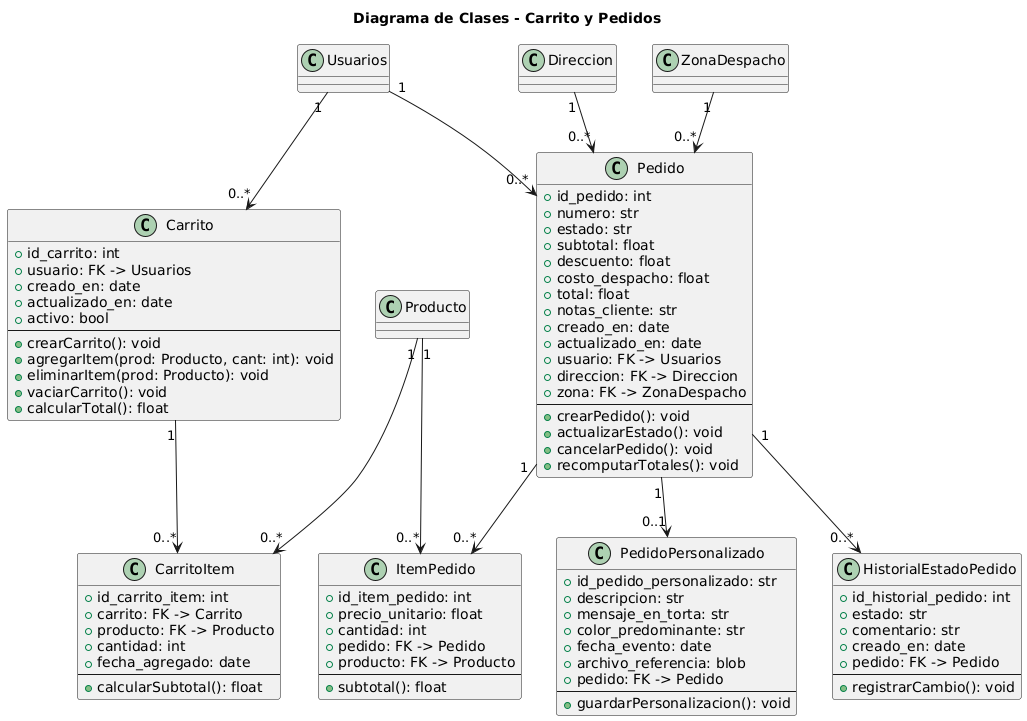
1. **Ilustración X.1 – Diagrama de Clases: Usuarios y Direcciones**

****

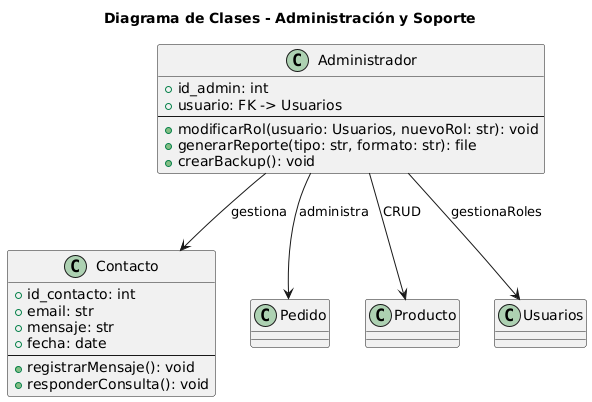
1. Ilustración X.2 – Diagrama de Clases: Productos y Promociones



1. Ilustración X.3 – Diagrama de Clases: Carrito y Pedidos



1. Ilustración X.4 – Diagrama de Clases: Soporte y Pedidos.

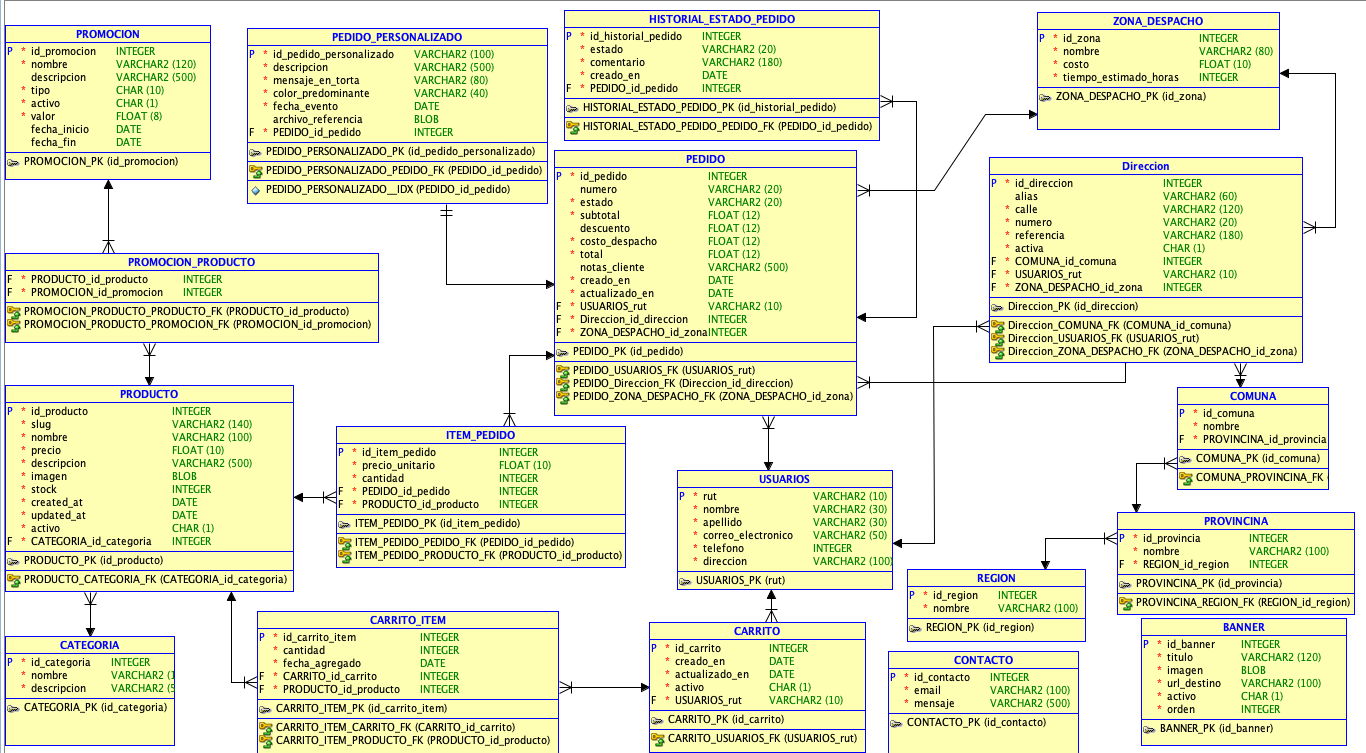


### Descripción de Clases

En la siguiente tabla se describen las principales clases identificadas en el modelo de dominio del sistema “Dulce Arte”. Cada clase contiene atributos y comportamientos asociados a las funcionalidades del sistema, representadas en los diagramas de clases UML.

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| CL-001 | Usuarios | Representa a los usuarios del sistema (clientes y administradores). Gestiona credenciales, datos personales, direcciones y roles de acceso. Contiene métodos para autenticación, registro y actualización de perfil. |
| CL-002 | Dirección | Almacena las direcciones de envío del usuario, asociadas a comuna, provincia, región y zona de despacho. Incluye alias, calle, número, referencias y estado de activación. |
| CL-003 | Región/Provincia/Comuna | Clases jerárquicas que modelan la ubicación geográfica nacional, permitiendo asociar cada dirección a su territorio correspondiente. |
| CL-004 | Zona Despacho | Define los costos y tiempos de entrega asociados a zonas específicas. Permite calcular el costo de envío según la ubicación del cliente. |
| CL-005 | Categoría | Agrupa los productos en conjuntos temáticos o funcionales. Permite mantener el catálogo ordenado y facilitar búsquedas y filtros. |
| CL-006 | Producto | Representa cada ítem disponible en la tienda. Contiene atributos como nombre, precio, descripción, imagen, stock, y su relación con una categoría. Incluye métodos de actualización de stock y control de disponibilidad. |
| CL-007 | Promoción | Modela las campañas promocionales aplicables a productos o categorías. Define tipo (porcentaje o monto fijo), valor, fechas y condiciones de aplicación. |
| CL-008 | Promoción Producto | Clase intermedia que vincula productos con promociones activas. Gestiona las relaciones mucho a muchos entre ambas entidades. |
| CL-009 | Banner | Clase encargada de gestionar los banners del sitio web. Contiene título, imagen, url de destino, orden y estado activo, utilizados en la página principal. |
| CL-010 | Carrito | Representa el carrito de compras persistente de cada usuario. Contiene la lista de productos seleccionados, su cantidad y subtotal calculado dinámicamente. Incluye métodos para agregar, eliminar y vaciar el carrito. |
| CL-011 | Carrito Item | Almacena los productos individuales dentro del carrito, con cantidad, fecha de agregado y subtotal. Se relaciona con la clase Carrito y Producto. |
| CL-012 | Pedido | Gestiona los pedidos realizados por los clientes, incluyendo subtotal, descuentos, despacho, total y estado (pendiente, enviado, entregado, cancelado). Se vincula con Usuario, Dirección, ZonaDespacho. |
| CL-013 | Item Pedido | Representa cada producto contenido en un pedido, incluyendo cantidad, nombre, precio unitario y subtotal. Permite auditar la compra en detalle. |
| CL-014 | Pedido Personalizado | Registra los pedidos esenciales solicitados por el cliente permitiendo agregar mensajes en tortas, colores predominantes o archivos de referencia. |
| CL-015 | HistorialEstadoPedido | Guarda el registro cronológico de cambios de estados de un pedido. Permite seguimiento y auditoría de la trazabilidad del proceso de entrega. |
| CL-016 | Administrador | Clase que representa al usuario con privilegios de gestión del sistema. Puede modificar roles, generar reportes y administrar productos, pedidos y promociones. |
| CL-017 | Reporte excel y PDF | Clase encargada de la generación de informes en distintos formatos (Excel, PDF) sobre ventas, pedidos, usuarios y puntos. Permite seleccionar filtros y descargar el archivo resultante. |
| CL-018 | Contacto | Registra los mensajes de contacto o consultas de los usuarios desde el formulario del sitio. Permite al administrador visualizar y responder solicitudes. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Ilustración 4: Diagrama de Base Datos (Relacional)**

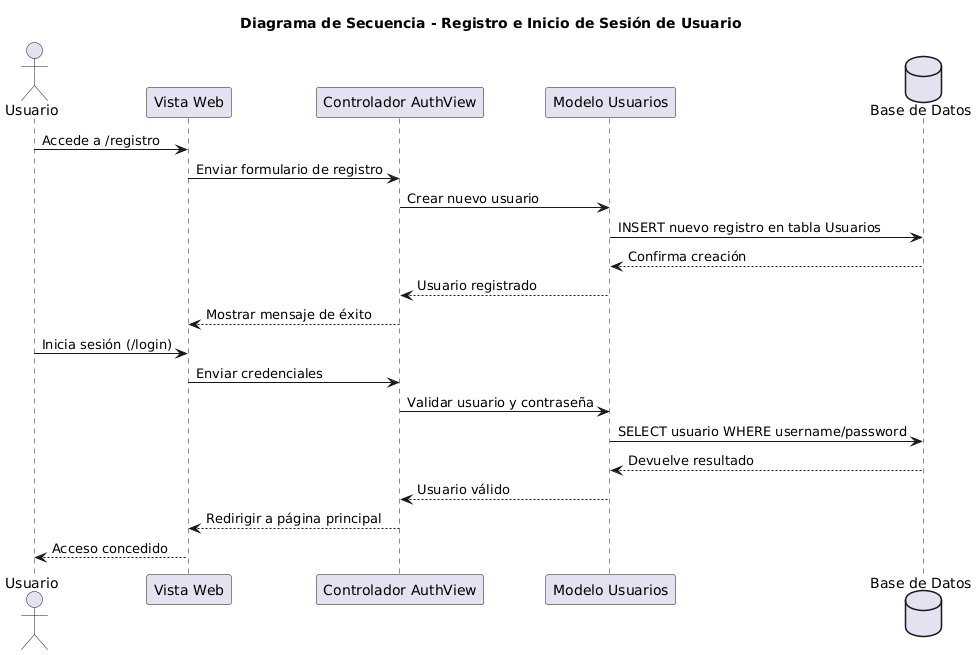


### Descripción de Tablas

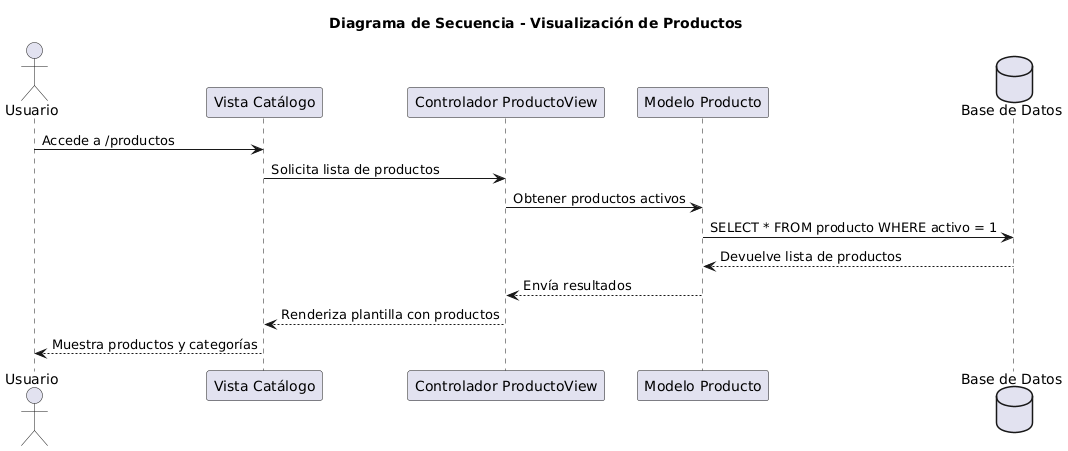
|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| TB-001 | CLIENTE |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* 1. **Parte Dinámica (Diagrama de Secuencias)**

**Ilustración 1: Registro e Inicio de Sesión de Usuario**



**Ilustración 2: Navegación y visualización de productos**



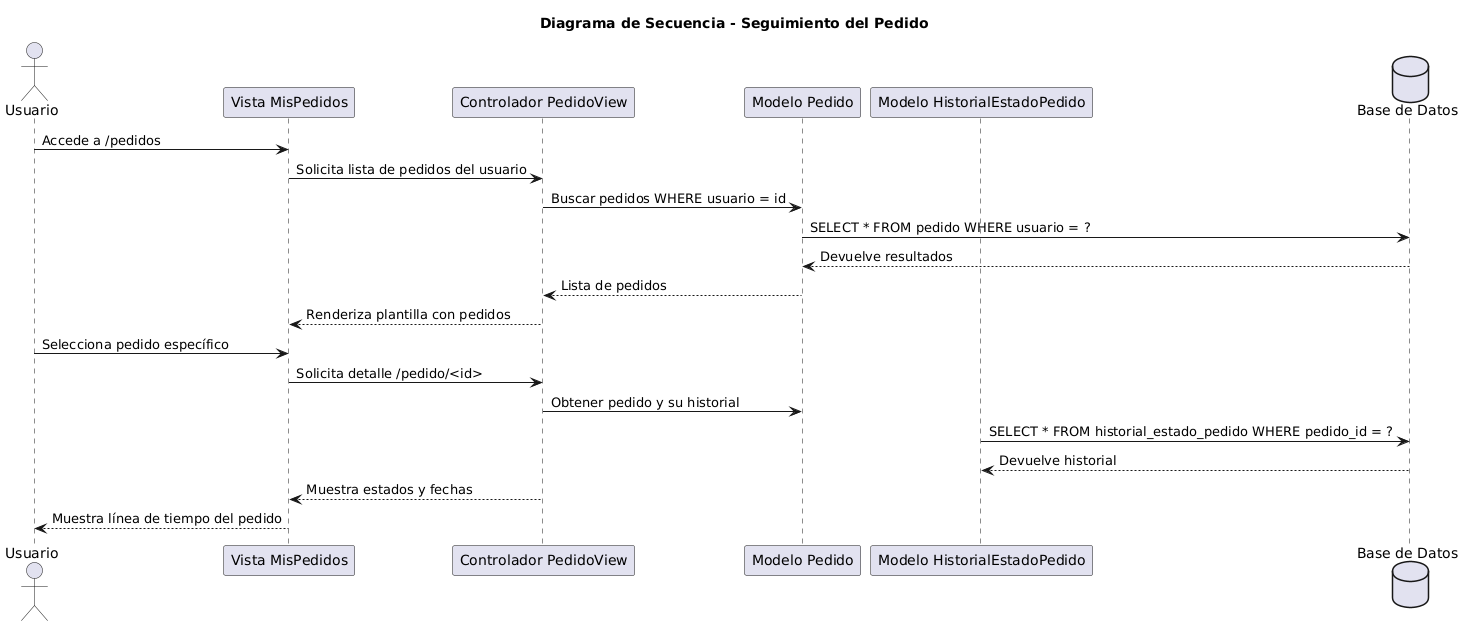
**Ilustración 3: Agregar productos al carrito**

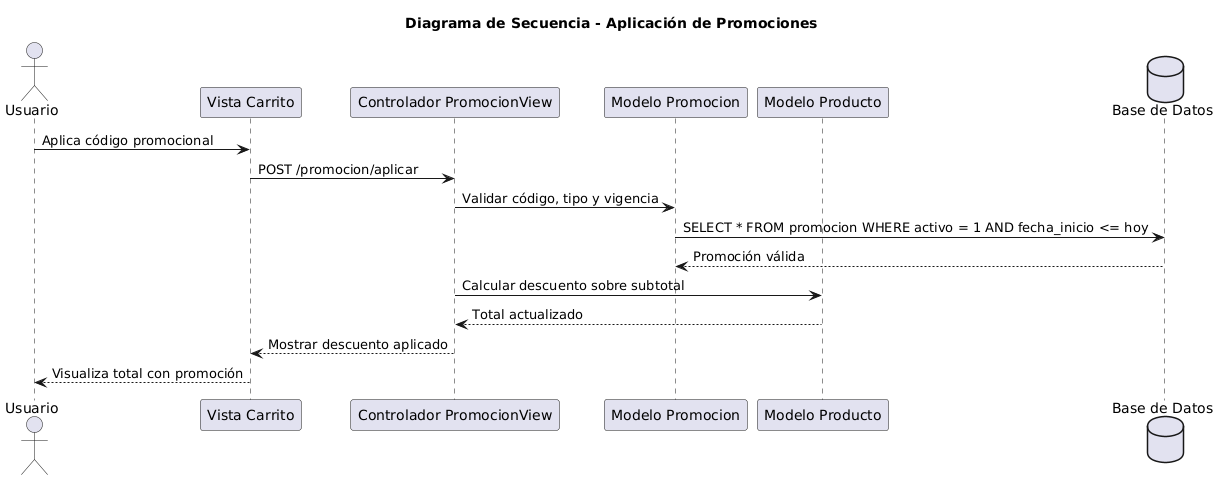


**Ilustración 4: Checkout y creación de pedido**

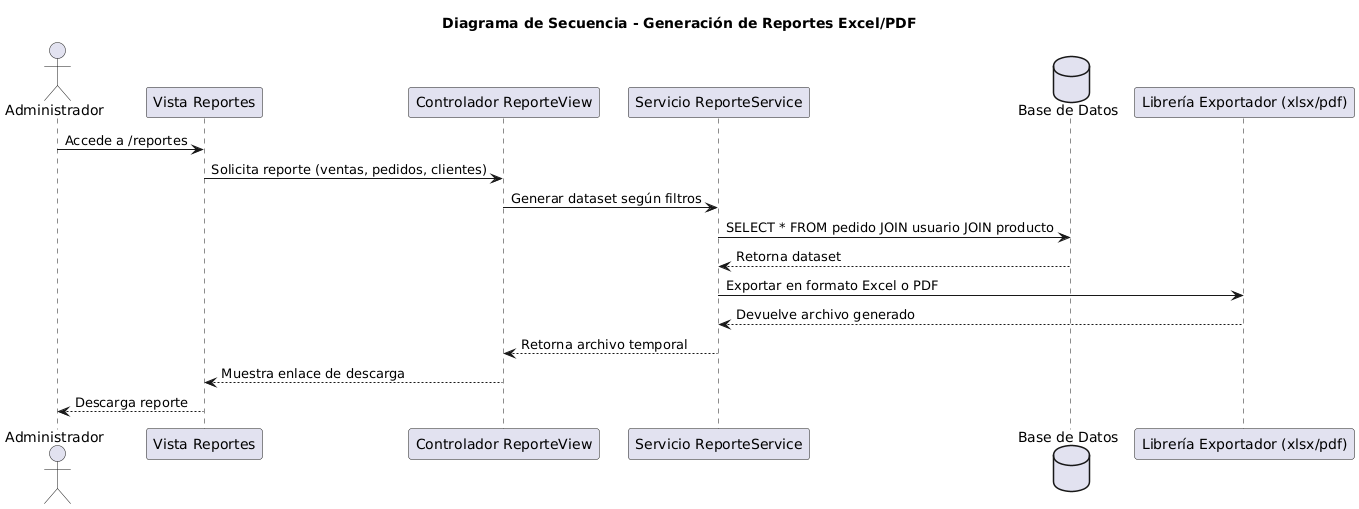


**Ilustración 5: Seguimiento del Pedido**

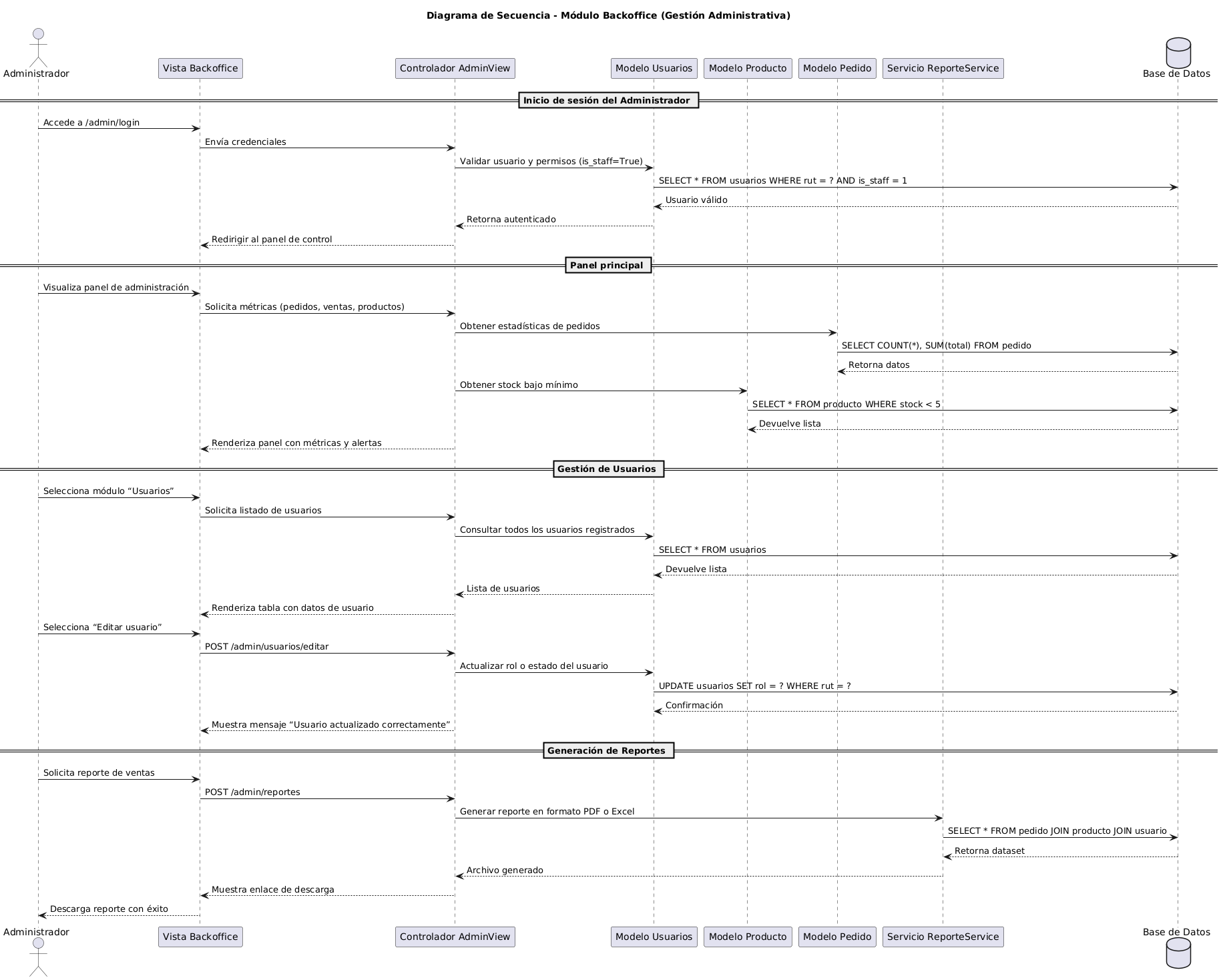
**Ilustración 6: Aplicación de Promociones**



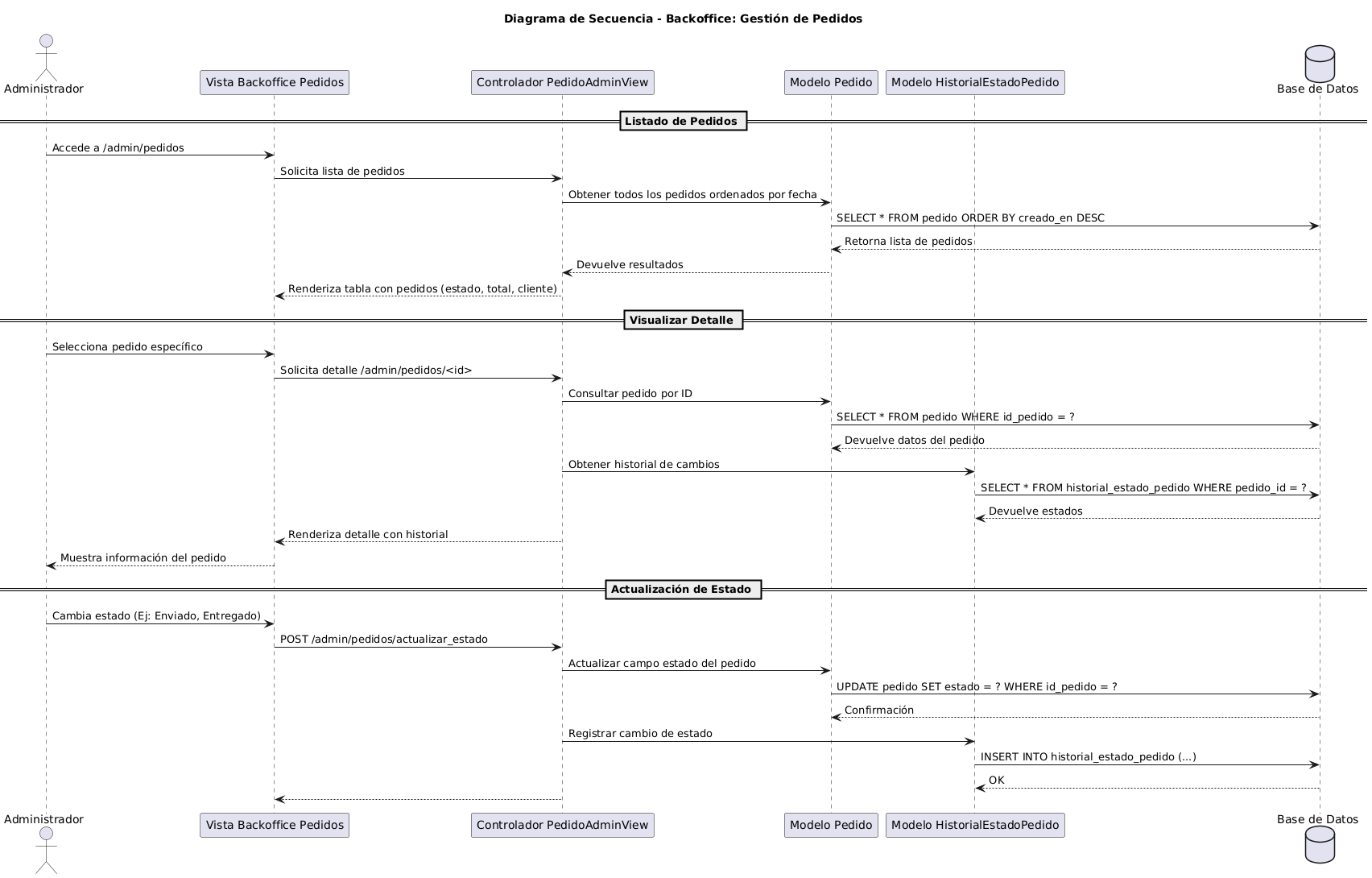
**Ilustración 7 : Generación de reportes (Excel/PDF)**



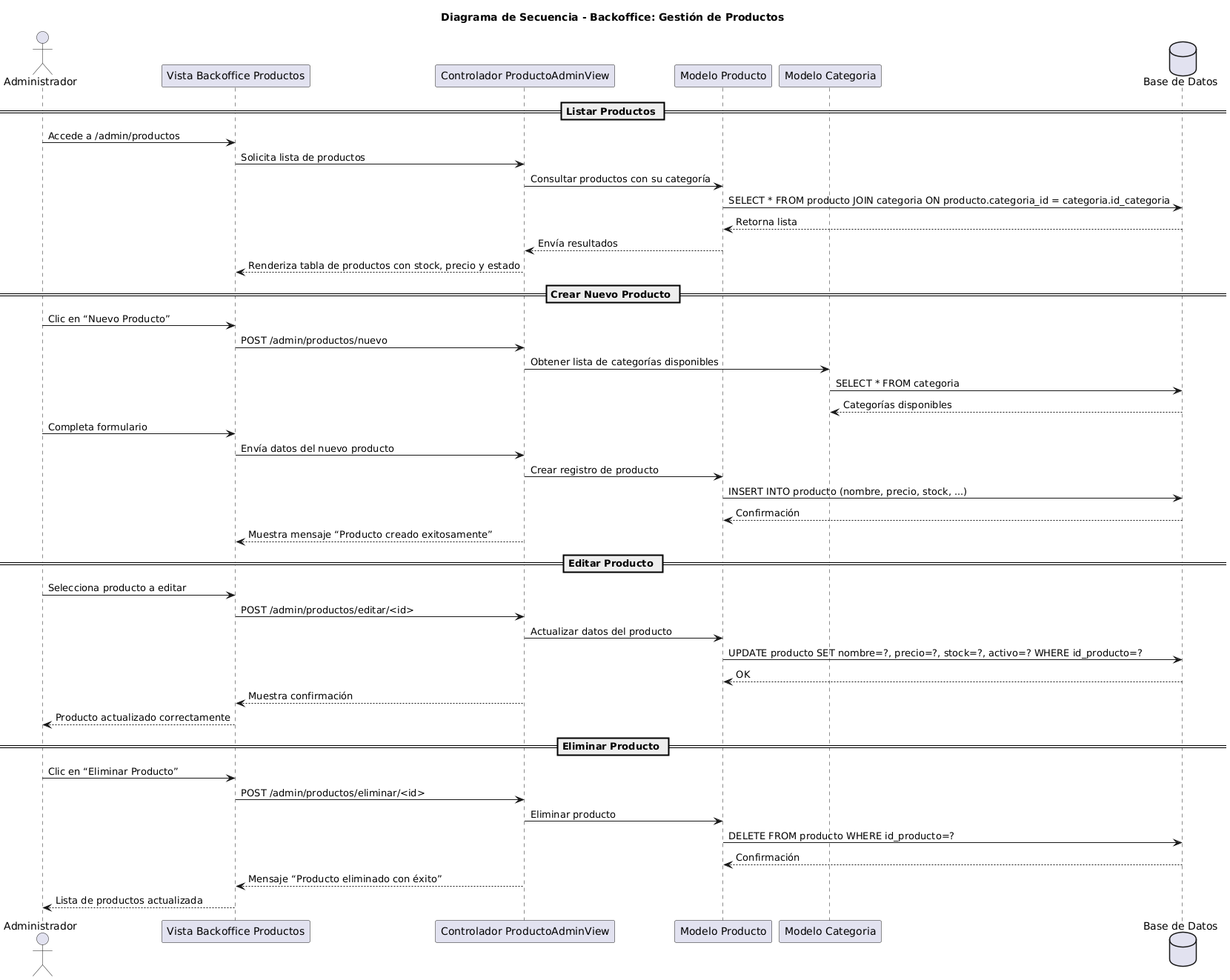
**Ilustración 8: Modulo Back Office (Gestión usuarios y reportes)**



**Ilustración 8.1 Backoffice: Gestión de pedidos**



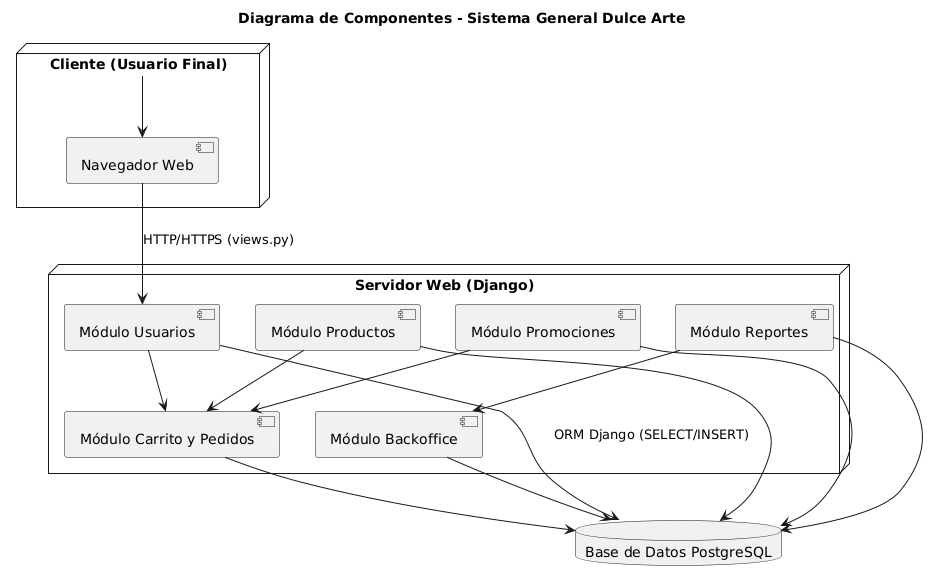
**Ilustración 8.2: Back office: gestión de productos**



1. **Vista de Desarrollo o Despliegue**

En esta vista se describen las componente o modulos en las cuales se dividirá o implementará el sistema

**Ilustración 1: Diagrama de componentes general del sistema**

****

**Ilustración 2: Diagrama de componentes modulo usuarios y direcciones**

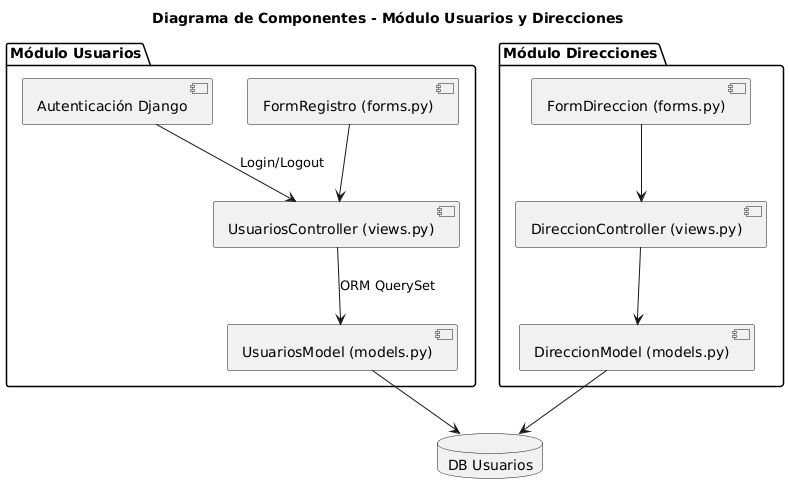
****

Ilustración 3: diagrama de componentes productos y promociones

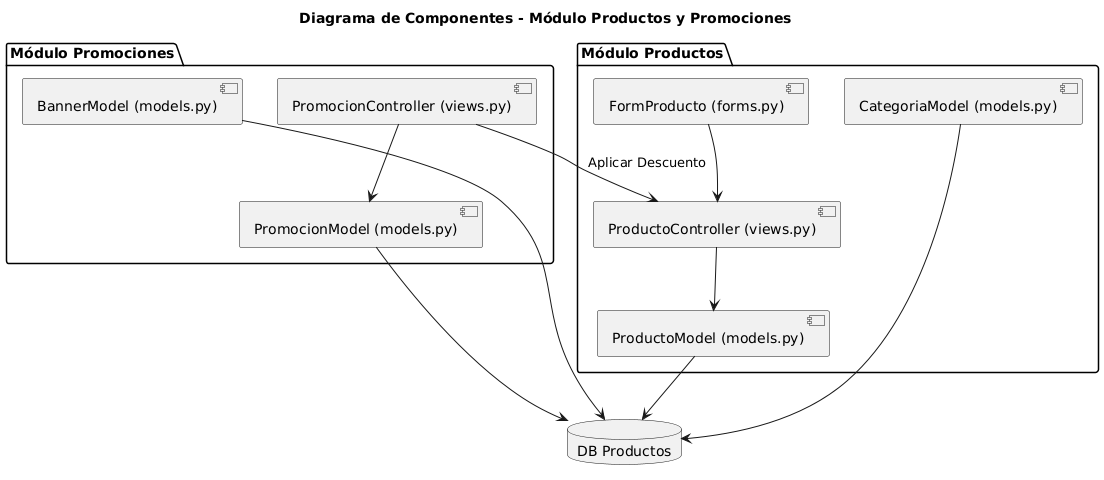
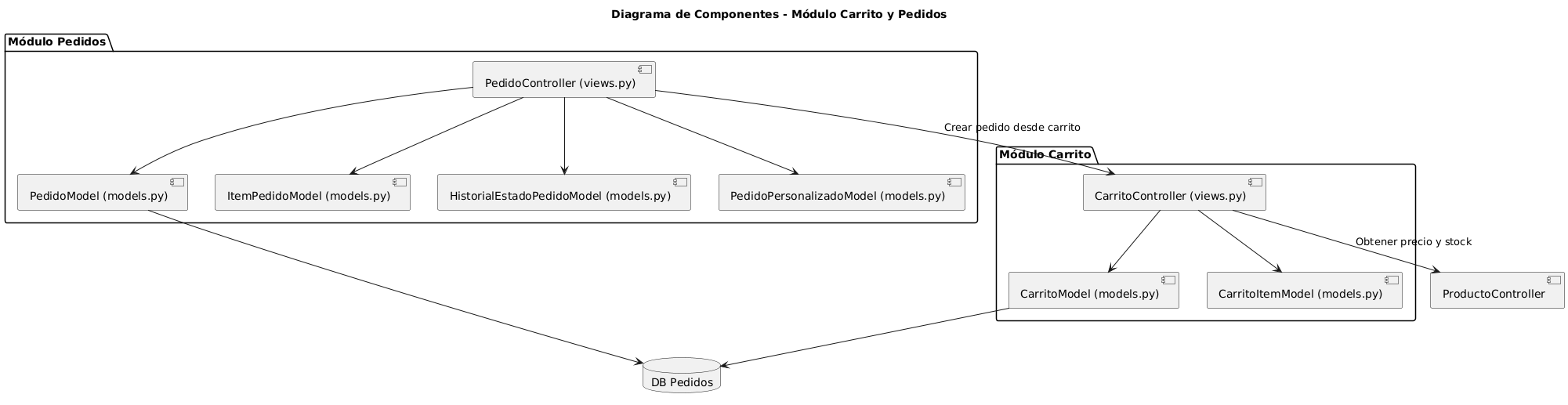
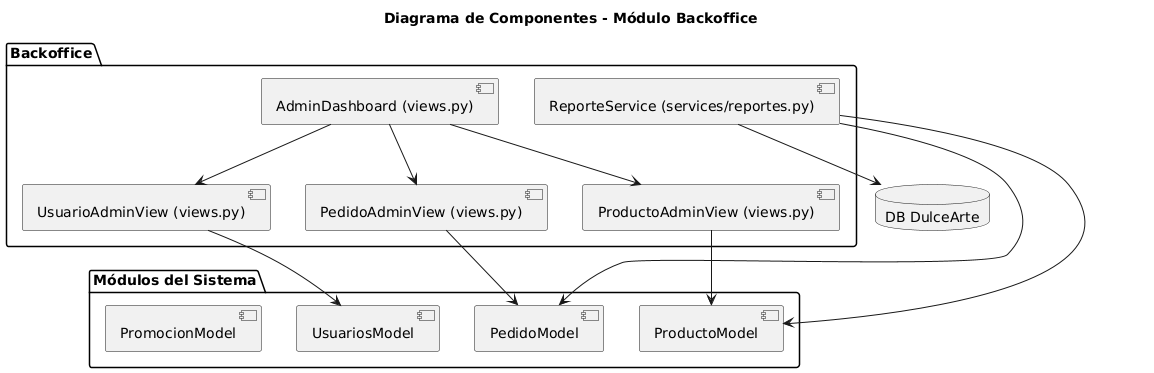


ilustración 4: diagrama de componentes carrito y pedidos

ilustración 5: diagrama de componentes modulo back office



**7. Vista Fisica**

En esta vista se despliegan los nodos que participan con el sistema.

Ilustración 1: diagrama despliegue general del sistema

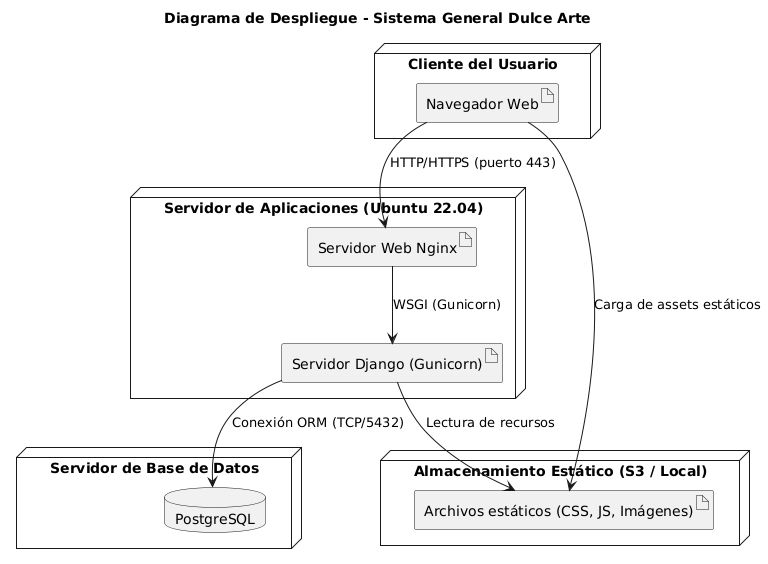


ilustración 2: diagrama de despliegue módulo de usuarios y direcciones

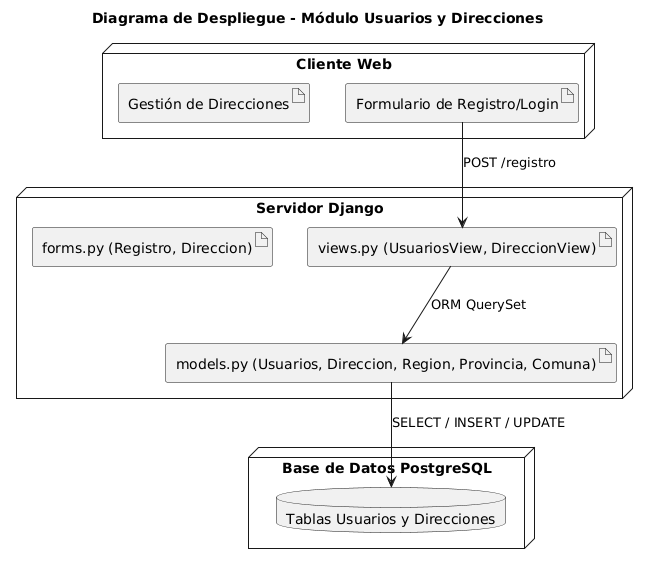


Ilustración 3: modulo productos y promociones

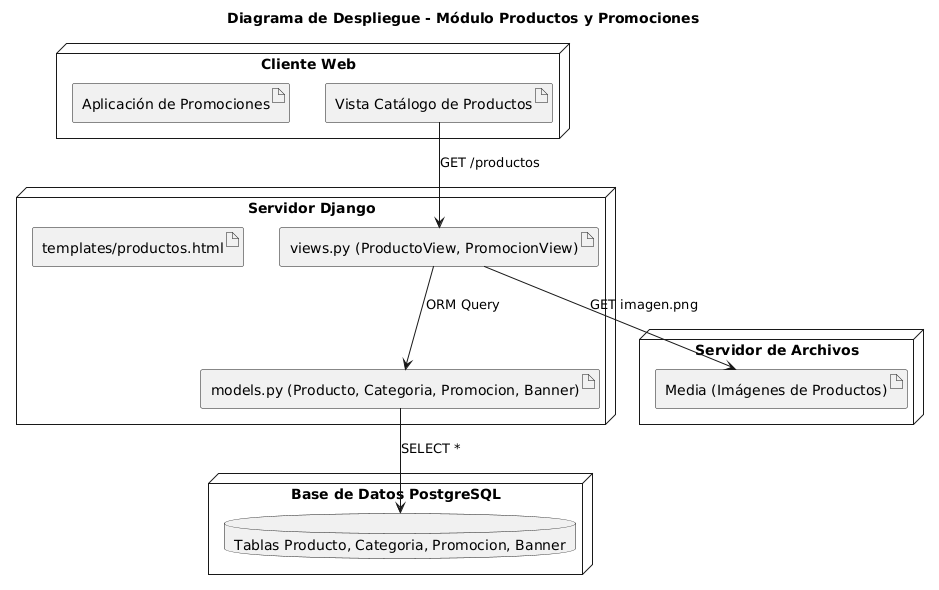


Ilustración 4: modulo carrito y pedidos

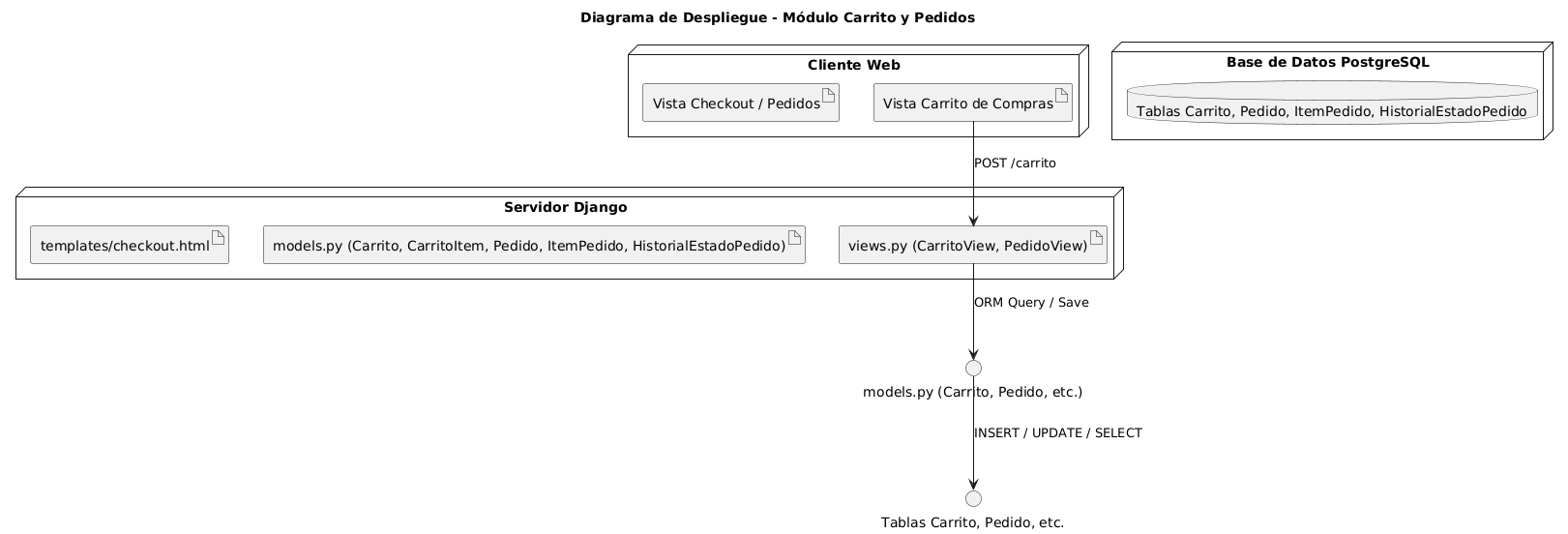


ilustración 5: modulo reportes excel pdf

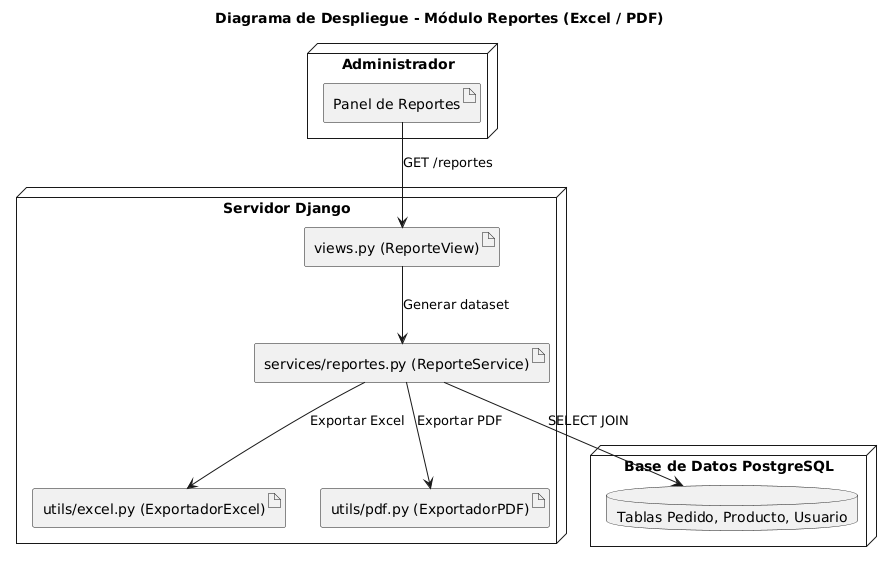
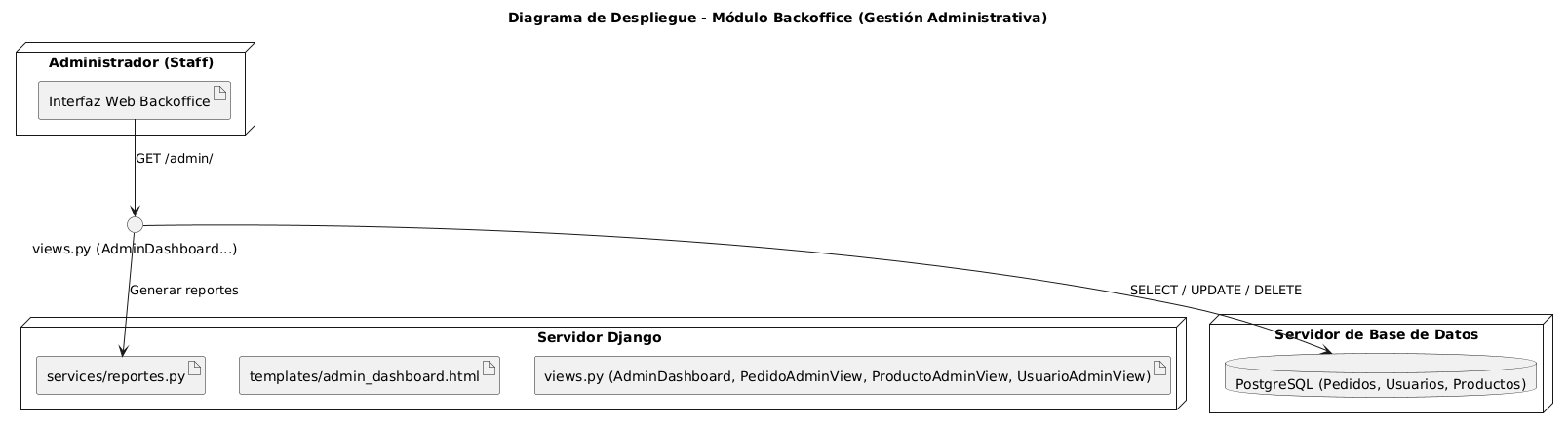


Ilustración 6: modulo back office



**8. Decisiones de Diseño y Selección de Alternativas**

Las principales decisiones arquitectónicas se tomaron en consideración de la restricción Tiempo de Construcción. Dado que el proyecto debe implementarse en un tiempo ajustado y sin holguras, se privilegió la adopción de una arquitectura conocida y que presente un bajo riesgo en su implementación.

Asimismo, la arquitectura se modularizó y cada módulo serán testeables unitariamente de forma de asegurar que cada pieza tenga una baja tasa de fallas.

Un segundo proposito que fue considerado en la seleccion de alternativa son :

1. **Análisis de Reutilización**

* Ejemplo : Se utilizara el codigo del mantenedor de productos en clientes
* Ejemplo: Se utilizaran plantillas gráficas con los mismos colores y logotipos en el mantendor de clientes, productos, ventas.