**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**
* **ERS**
* **Documento de Casos de uso Extendido**
* **Planilla de Requerimientos Funcionales y No Funcionales**
* Acta de Constitución del Proyecto
* Documento de Mockups
* Informe de Avance de Sprint 3
* Norma IEEE 830-1998

* 1. **Resumen ejecutivo**

El presente documento tiene como finalidad describir la planificación, diseño y alcance funcional del Dashboard (DAS) correspondiente al sistema SmartFlow – Sistema de Gestión y Reservas.

Este módulo constituye el núcleo visual de control administrativo del sistema, diseñado para proporcionar al Administrador una vista integral del estado operativo mediante indicadores clave de rendimiento (KPI), reportes dinámicos y notificaciones en tiempo real.

El Dashboard consolida la información proveniente de los distintos módulos —usuarios, reservas, servicios y notificaciones—, ofreciendo una interfaz intuitiva y visual que permite analizar la actividad del sistema de manera rápida y eficiente.

Su diseño se basa en la arquitectura establecida en el documento ERS, garantizando la integración con la base de datos SQL Server, la comunicación mediante Entity Framework Core y la visualización a través de Razor Pages con Bootstrap 5.

El desarrollo de este módulo forma parte del Sprint 4 del proyecto, correspondiente a la fase de implementación de los módulos principales.

Con el Dashboard se busca optimizar el proceso de monitoreo, reducir los tiempos de revisión manual de información y mejorar la capacidad de toma de decisiones del administrador, cumpliendo con los objetivos definidos en el Acta de Constitución y los estándares técnicos establecidos por Duoc UC.

Finalmente, el Dashboard servirá como punto de acceso central del sistema, reflejando el rendimiento global de SmartFlow y consolidando su funcionalidad como herramienta de gestión integral, segura y escalable.

* 1. **Representación**

1.5 Representación

La arquitectura del sistema SmartFlow – Sistema de Gestión y Reservas está representada siguiendo el enfoque del framework 4+1 y las recomendaciones del Proceso Unificado de Desarrollo (RUP).

Este enfoque permite visualizar la arquitectura desde diferentes perspectivas, facilitando la comprensión global del módulo Dashboard (DAS) dentro del sistema.

Las vistas incluidas en esta versión del documento son las siguientes:

● Vista de Escenarios:

Describe los casos de uso más significativos relacionados con el Dashboard, los actores que intervienen (Administrador, Usuario y Sistema) y la interacción entre ellos.

Esta vista se basa principalmente en los casos de uso CU004 (Generar Reporte de Reservas), CU011 (Exportar Reportes PDF/XLS) y CU012 (Monitoreo y Auditoría del Sistema), ya que son los más representativos para el control administrativo y visualización de métricas.

● Vista de Procesos:

Describe los procesos internos que intervienen en la ejecución del Dashboard, incluyendo la consulta de datos mediante Entity Framework Core, la carga dinámica de información en las tarjetas y gráficos y la actualización en tiempo real de los indicadores.

También se detalla la comunicación entre las capas Controlador (C#), Vista (Razor Pages) y Base de Datos (SQL Server).

● Vista Lógica:

Presenta la estructura lógica del sistema, compuesta por capas bien definidas:

Capa de Presentación: interfaces Razor Pages con componentes Bootstrap y gráficos Chart.js.

Capa de Negocio: controladores y servicios que gestionan la lógica del Dashboard.

Capa de Datos: contexto de Entity Framework Core y modelos que conectan con las tablas Usuarios, Reservas, Servicios, Notificaciones y Auditoría.

Estas capas garantizan modularidad, mantenibilidad y reutilización del código.

● Vista de Desarrollo o Despliegue:

Describe los componentes técnicos construidos y su distribución en el entorno de desarrollo.

El módulo Dashboard se despliega dentro del proyecto SmartFlow.Web, bajo la ruta

/Pages/Admin/Dashboard.cshtml, utilizando un layout compartido (\_Layout.cshtml) que mantiene la coherencia visual con el resto del sistema.

Los gráficos y métricas se generan dinámicamente con Chart.js y consultas LINQ a través de EF Core.

● Vista Física:

Define las restricciones tecnológicas y de infraestructura que influyen sobre las decisiones arquitectónicas.

El sistema se ejecuta sobre un servidor IIS o Kestrel, con base de datos SQL Server 2019/2022,

y se desarrolla bajo .NET Core 8.0.

El Dashboard debe mantener un tiempo de respuesta menor a 3 segundos y un nivel de disponibilidad superior al 95%.

Asimismo, cumple con los estándares de seguridad definidos en la norma IEEE 830 y las políticas de Duoc UC para proyectos de titulación.

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 del proyecto, las cuales se presentan a continuación:

Tiempo de construcción: El desarrollo total del sistema, incluyendo el módulo Dashboard, tiene un plazo estimado de 16 semanas, dividido en 5 sprints según la planificación establecida en la Carta Gantt.

Infraestructura:

El sistema se ejecutará sobre entornos Windows 10/11 o Windows Server, utilizando Internet Information Services (IIS) o Kestrel como servidores web.

El motor de base de datos será Microsoft SQL Server 2019 o superior.

Lenguaje y framework:

Se establece el uso exclusivo de C# con .NET Core 8.0, empleando Razor Pages para el desarrollo del frontend y Entity Framework Core como ORM para la comunicación con la base de datos.

Diseño y librerías:

Se prioriza el uso de Bootstrap 5 y librerías de código abierto como Chart.js para la representación gráfica de los indicadores del Dashboard.

No se autoriza el uso de librerías propietarias o externas que alteren la arquitectura base del sistema.

Seguridad:

El acceso al Dashboard será exclusivo para usuarios con rol Administrador.

Se aplicarán políticas de autenticación seguras (hash + salt), conexión cifrada mediante HTTPS, y trazabilidad mediante el módulo de auditoría interna.

Compatibilidad:

El sistema deberá ser completamente funcional en navegadores modernos (Google Chrome, Microsoft Edge y Mozilla Firefox) y contar con diseño responsivo para uso en dispositivos móviles y tablets.

Sin licencias propietarias:

Se prioriza el uso de software libre o de código abierto para todas las dependencias del proyecto, garantizando independencia tecnológica y libre distribución académica.

* 1. **Otros antecedentes y consideraciones**

El sistema se apoya en el framework .NET Core 8.0, el cual proporciona una base sólida y moderna para el desarrollo web de SmartFlow.

Este entorno fue seleccionado por su estabilidad, seguridad y compatibilidad con los requerimientos del proyecto, especialmente en el módulo Dashboard (DAS).

Implementa un ORM robusto mediante Entity Framework Core, que facilita el manejo relacional con SQL Server, optimizando la comunicación entre la base de datos y las capas de negocio.

Incluye un sistema de autenticación nativo, permitiendo gestionar roles, sesiones y permisos con encriptación de contraseñas (hash + salt).

Soporta modularización y separación de responsabilidades, gracias a su arquitectura en capas (Presentación, Negocio y Datos), lo que mejora la mantenibilidad del sistema.

Ofrece integración fluida con librerías de frontend, como Bootstrap 5 y Chart.js, utilizadas para la creación del Dashboard responsivo y visualmente atractivo.

Permite despliegue multiplataforma, ya que puede ejecutarse en entornos Windows, Linux o en la nube, aunque en esta versión el entorno oficial de pruebas es Windows Server + IIS/Kestrel.

Cumple con los lineamientos de la norma IEEE 830, asegurando consistencia, trazabilidad y calidad en la documentación y en la implementación del sistema.

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**

### Caso de Uso CU001 – Mantenedor datos de usuarios

|  |  |
| --- | --- |
| **Actores** | Administrador |
| **Precondición** | El administrador debe estar autenticado en el sistema. |
| **Req. No Funcional** | Interfaz amigable, validación de formularios. |
| **Descripción** | Este caso de uso se inicia cuando el administrador accede al módulo de usuarios para agregar, editar o eliminar datos de los usuarios registrados. |

### Caso de Uso CU002 – Cálculo de ingreso de matrícula

|  |  |
| --- | --- |
| **Actores** | Cajero |
| **Precondición** | El cajero debe iniciar sesión en el sistema. |
| **Req. No Funcional** | Exactitud en cálculos, tiempo de respuesta menor a 3 segundos. |
| **Descripción** | Este caso de uso se inicia cuando el cajero ingresa los datos de matrícula para calcular los ingresos asociados. |

### Caso de Uso CU003 – Reservar un servicio

|  |  |
| --- | --- |
| **Actores** | Usuario |
| **Precondición** | El usuario debe estar registrado y autenticado. |
| **Req. No Funcional** | Disponibilidad en tiempo real. |
| **Descripción** | Este caso de uso se inicia cuando el usuario selecciona un servicio y solicita una reserva en el calendario disponible. |

### Caso de Uso CU004 – Generar reporte de reservas

|  |  |
| --- | --- |
| **Actores** | Administrador |
| **Precondición** | El administrador debe tener permisos de generación de reportes. |
| **Req. No Funcional** | Exportación a PDF/XLS, seguridad en acceso. |
| **Descripción** | Este caso de uso se inicia cuando el administrador genera reportes detallados de reservas realizadas. |

### Caso de Uso CU005 – Envío de notificación automática

|  |  |
| --- | --- |
| **Actores** | Sistema |
| **Precondición** | Debe existir un evento programado |
| **Req. No Funcional** | Entrega confiable de mensajes vía correo/móvil. |
| **Descripción** | Este caso de uso se inicia cuando el sistema envía automáticamente una notificación a un usuario sobre una reserva o actualización. |

### Caso de Uso CU006 – Autenticación y gestión de sesiones

|  |  |
| --- | --- |
| **Actores** | Usuario, Administrador |
| **Precondición** | El usuario debe tener credenciales válidas. |
| **Req. No Funcional** | Encriptación de contraseñas, control de sesiones. |
| **Descripción** | Este caso de uso se inicia cuando un usuario o administrador ingresa sus credenciales para acceder al sistema. |

### Caso de Uso CU007 – Mantener datos de servicios

|  |  |
| --- | --- |
| **Actores** | Administrador |
| **Precondición** | El administrador debe estar autenticado en el sistema. |
| **Req. No Funcional** | Validación de entradas, consistencia en base de datos. |
| **Descripción** | Este caso de uso se inicia cuando el administrador agrega, edita o elimina información de los servicios ofrecidos. |

### Caso de Uso CU008 – Gestión de perfiles y roles

|  |  |
| --- | --- |
| **Actores** | Administrador |
| **Precondición** | El administrador debe tener permisos de configuración. |
| **Req. No Funcional** | Control de accesos, usabilidad en la asignación de roles. |
| **Descripción** | Este caso de uso se inicia cuando el administrador asigna, modifica o elimina roles y permisos de usuarios. |

### Caso de Uso CU009 – Cancelar reserva

|  |  |
| --- | --- |
| **Actores** | Usuario |
| **Precondición** | El usuario debe tener una reserva activa. |
| **Req. No Funcional** | Confirmación de cancelación, actualización inmediata de disponibilidad. |
| **Descripción** | Este caso de uso se inicia cuando el usuario cancela una reserva previamente realizada. |

### Caso de Uso CU010 – Consultar disponibilidad de horarios

|  |  |
| --- | --- |
| **Actores** | Usuario |
| **Precondición** | El usuario debe estar registrado en el sistema. |
| **Req. No Funcional** | Tiempo de respuesta inmediato, interfaz clara. |
| **Descripción** | Este caso de uso se inicia cuando el usuario consulta el calendario para verificar la disponibilidad de horarios de un servicio. |

Caso de Uso CU011 – Exportar reportes a PDF/XLS

|  |  |
| --- | --- |
| **Actores** | Administrador |
| **Precondición** | El administrador debe tener permisos de exportación. |
| **Req. No Funcional** | Compatibilidad con múltiples formatos, integridad de los datos. |
| **Descripción** | Este caso de uso se inicia cuando el administrador exporta un reporte generado a formato PDF o XLS. |

### Caso de Uso CU012 – Monitoreo y auditoría del sistema

|  |  |
| --- | --- |
| **Actores** | Administrador, Sistema |
| **Precondición** | El sistema debe estar en funcionamiento y con registros de actividad. |
| **Req. No Funcional** | Seguridad, trazabilidad, disponibilidad del 99%. |
| **Descripción** | Este caso de uso se inicia cuando el administrador revisa los registros de actividad para auditar el uso del sistema. |

* 1. **Casos de Usos Extendidos**

### Caso de Uso CU00X – Nombre Caso Uso

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actores** | | Usuario (actor primario)  Sistema (actor secundario)  Administrador  Cajero | | |
| **Objetivo** | | Permitir que el usuario seleccione un servicio, consulte disponibilidad de horarios y registre una reserva confirmada en el sistema. | | |
| **Precondición** | | El usuario debe estar registrado y autenticado en el sistema (CU06 – Autenticación y gestión de sesiones).  Deben existir servicios y horarios disponibles (CU10 – Consultar disponibilidad de horarios). | | |
| **PostCondición** | | La reserva queda almacenada en la base de datos con estado “Confirmada”.  Se programa el envío de una notificación automática al usuario (CU05 – Envío de notificación automática). | | |
| **Curso Normal** | | | | |
| **Paso** | **Acción del Actor** | | **Paso** | **Respuesta del Sistema** |
| 1 | |  | | --- | | El usuario accede al módulo de reservas. |  |  | | --- | |  | | | 1 | El sistema muestra la lista de servicios disponibles. |
| 2 | |  | | --- | | Selecciona un servicio de la lista. |  |  | | --- | |  | | | 2 | El sistema muestra los horarios disponibles para ese servicio. |
| 3 | |  | | --- | | Elige un horario y confirma la solicitud. |  |  | | --- | |  | | | 3 | |  | | --- | | El sistema valida que el horario siga libre. |  |  | | --- | |  | |
| 4 | |  | | --- | | Ingresa datos necesarios (ej.: comentarios, medio de contacto). |  |  | | --- | |  | | | 4 | |  | | --- | | El sistema valida el formulario (campos obligatorios, formato). |  |  | | --- | |  | |
| 5 | |  | | --- | | Confirma la reserva. |  |  | | --- | |  | | | 5 | |  | | --- | | El sistema registra la reserva en la BD, genera un número de reserva y muestra mensaje de éxito. |  |  | | --- | |  | |
| 6 | - | | 6 | El sistema programa el envío de notificación automática (correo/SMS). |
| **Cursos Alternativos** | | | | |
| **Paso** | **Actor** | | **Paso** | **Respuesta del Sistema** |
| 3A | Usuario | |  | |  | | --- | | Si el horario ya está ocupado, el sistema muestra mensaje de error y sugiere horarios alternativos. |  |  | | --- | |  | |
| 4A | Usuario | |  | Si los datos del formulario no son válidos, el sistema resalta los campos con error y solicita corrección. |
| 5A | Usuario | |  | Si ocurre un error al registrar la reserva, el sistema informa que no se pudo completar y pide reintentar. |

|

Diagrama

El contenido generado por IA puede ser incorrecto.

**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 | Estudiante, Administrador | Alta |
| CU-002 | Autenticación y gestión de sesiones | Todos los usuarios | Alta |
| CU-003 | Asignación de roles a usuario | Administrador | Media |
| CU-004 | Visualizar servicios disponibles | Estudiante | Alta |
| CU-005 | Consultar disponibilidad de horarios | Estudiante, Sistema | Alta |
| CU-006 | Crear solicitud | Estudiante | Alta |
| CU-007 | Reservar un servicio | Estudiante, Sistema | Alta |
| CU-008 | Aprobar/Rechazar solicitudes | Coordinador | Alta |
| CU-009 | Enviar notificación automática | Sistema Notificaciones | Alta |
| CU-010 | Visualizar historial de solicitudes | Estudiante, Coordinador | Media |
| CU-011 | Generar reportes en PDF/XLS | Administrador, Coordinador | Media |
| CU-012 | Administrar servicios | Administrador, Coordinador | Media |
| CU-013 | Administrar usuarios | Administrador | Media |
| CU-014 | Registrar auditoría | Sistema | Alta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

## EC-01

|  |  |
| --- | --- |
| Identificador | EC-01 |
| Escenario(s) | Múltiples usuarios intentan reservar el mismo horario o consultar disponibilidad simultáneamente. |
| Atributos de Calidad relevantes | Rendimiento – Desempeño – Escalabilidad |
| Componentes del Escenario | Estímulos: Consultas concurrentes al calendario y creación de reservas. Fuente del estímulo: Usuarios del sistema. Ambiente: Producción, alta demanda. Artefacto: Módulo de Reservas (SmartFlow), EF Core, SQL Server. Respuesta: Sistema responde sin bloqueos, evita reservas duplicadas. Medida: Tiempo < 2s; error < 2%; sin duplicidad. |

## EC-02

|  |  |
| --- | --- |
| Identificador | EC-02 |
| Escenario(s) | Un usuario externo intenta ingresar repetidamente con credenciales inválidas. |
| Atributos de Calidad relevantes | Seguridad – Confidencialidad – Integridad |
| Componentes del Escenario | Estímulos: Intentos de login fallidos. Fuente del estímulo: Usuario no autorizado. Ambiente: Sistema con HTTPS. Artefacto: Módulo de Autenticación. Respuesta: Bloqueo tras 3 intentos, registro de IP. Medida: 0% accesos indebidos; bloqueo automático. |

## EC-03

|  |  |
| --- | --- |
| Identificador | EC-03 |
| Escenario(s) | El sistema debe mostrar la disponibilidad de horarios bajo concurrencia. |
| Atributos de Calidad relevantes | Disponibilidad – Tolerancia a fallos |
| Componentes del Escenario | Estímulos: Usuarios consultan disponibilidad. Fuente del estímulo: Estudiantes. Ambiente: Producción. Artefacto: Calendario + consultas SQL. Respuesta: Sistema mantiene operación continua. Medida: Disponibilidad ≥ 99%; carga < 1.5s. |

## EC-04

|  |  |
| --- | --- |
| Identificador | EC-04 |
| Escenario(s) | El administrador genera un reporte filtrado sin errores. |
| Atributos de Calidad relevantes | Usabilidad – Eficiencia – Accesibilidad |
| Componentes del Escenario | Estímulos: Uso del módulo de reportes. Fuente del estímulo: Administrador. Ambiente: Dashboard. Artefacto: Módulo de Reportes. Respuesta: Filtros intuitivos, exportación correcta. Medida: 95% éxito; exportación < 3s. |

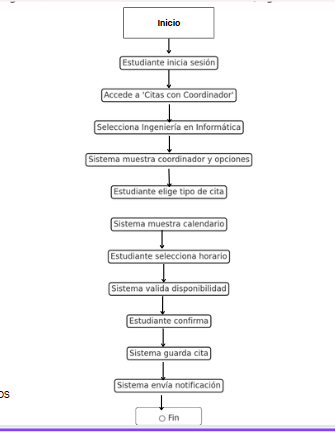
## EC-05

|  |  |
| --- | --- |
| Identificador | EC-05 |
| Escenario(s) | El administrador agrega un nuevo servicio sin afectar el sistema. |
| Atributos de Calidad relevantes | Mantenibilidad – Escalabilidad – Reusabilidad |
| Componentes del Escenario | Estímulos: Solicitud de agregar servicio. Fuente del estímulo: Administración. Ambiente: Desarrollo controlado. Artefacto: Módulo Servicios, EF Core. Respuesta: Integración sin errores. Medida: Implementación < 1 día; incidencias = 0. |

1. **Vista de Procesos**

**Ilustración 2: Diagramas de Actividades**

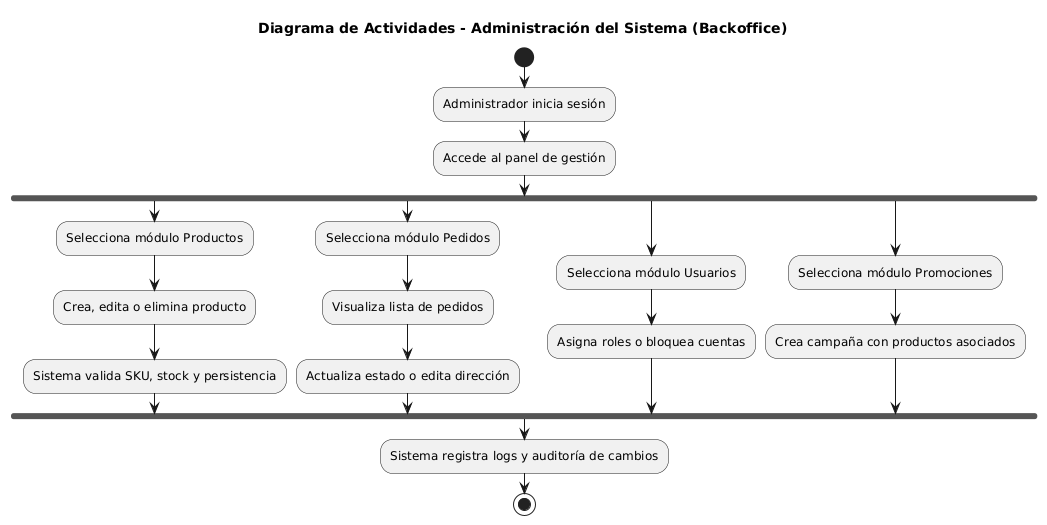
**1.Diagrama Agendar Cita con Coordinador de Carrera**



**2. Registro y autenticación de usuario.**



**3. Administracion de sistema**

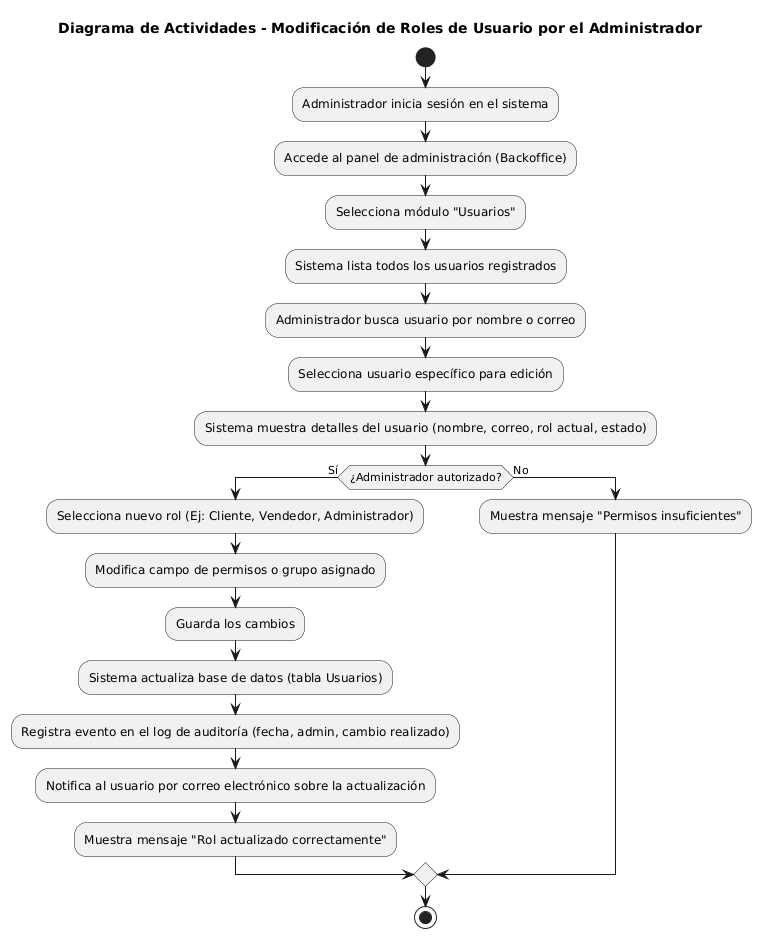


**4. Genera reportes**

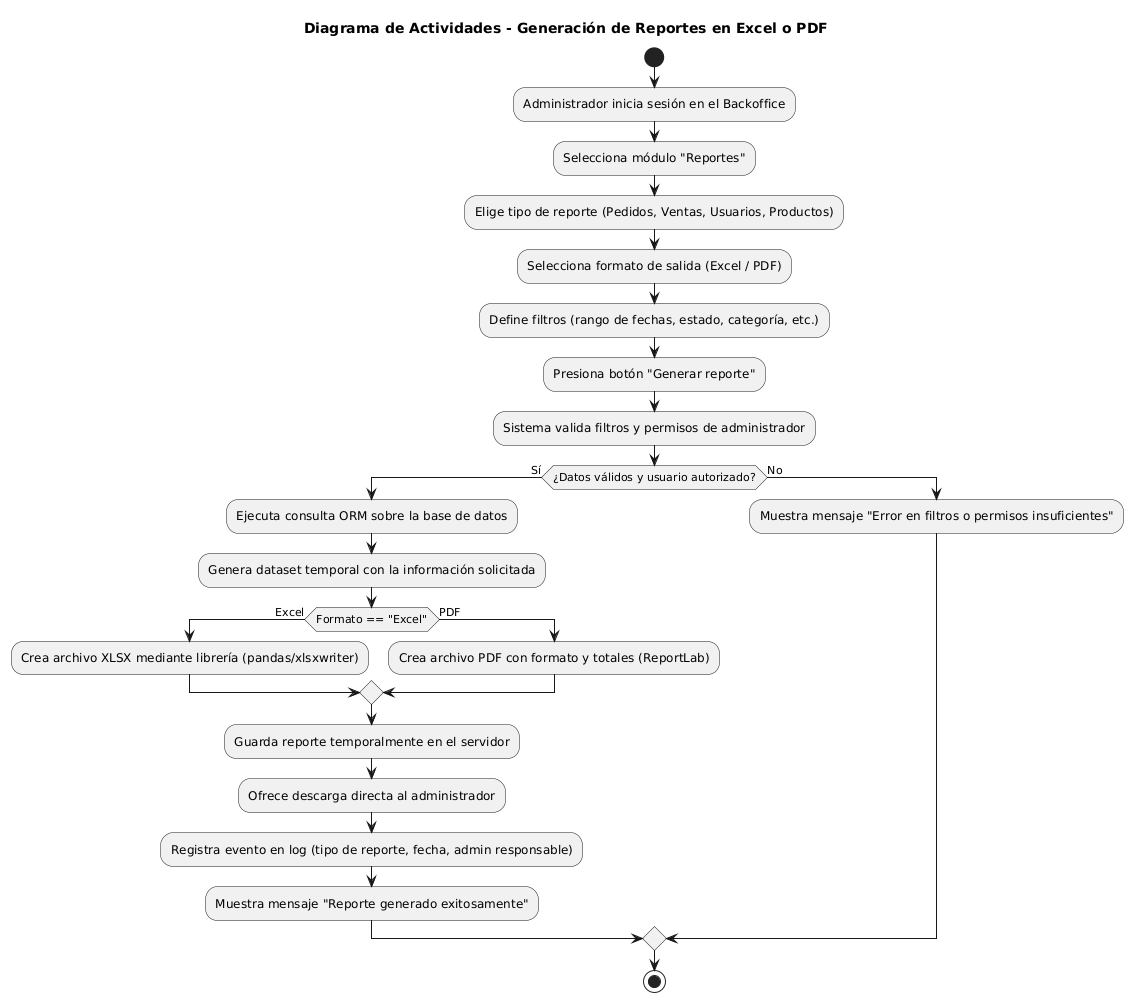
Tabla

El contenido generado por IA puede ser incorrecto.

**5. modificacion de roles**



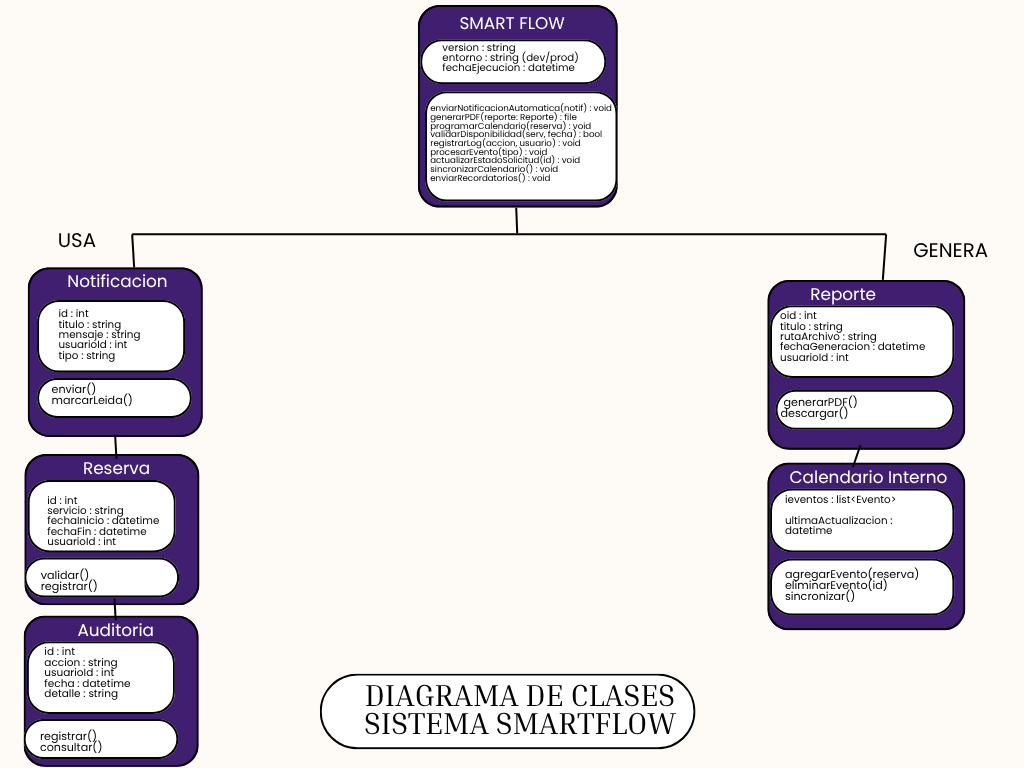
**6. generar excel o pdf**

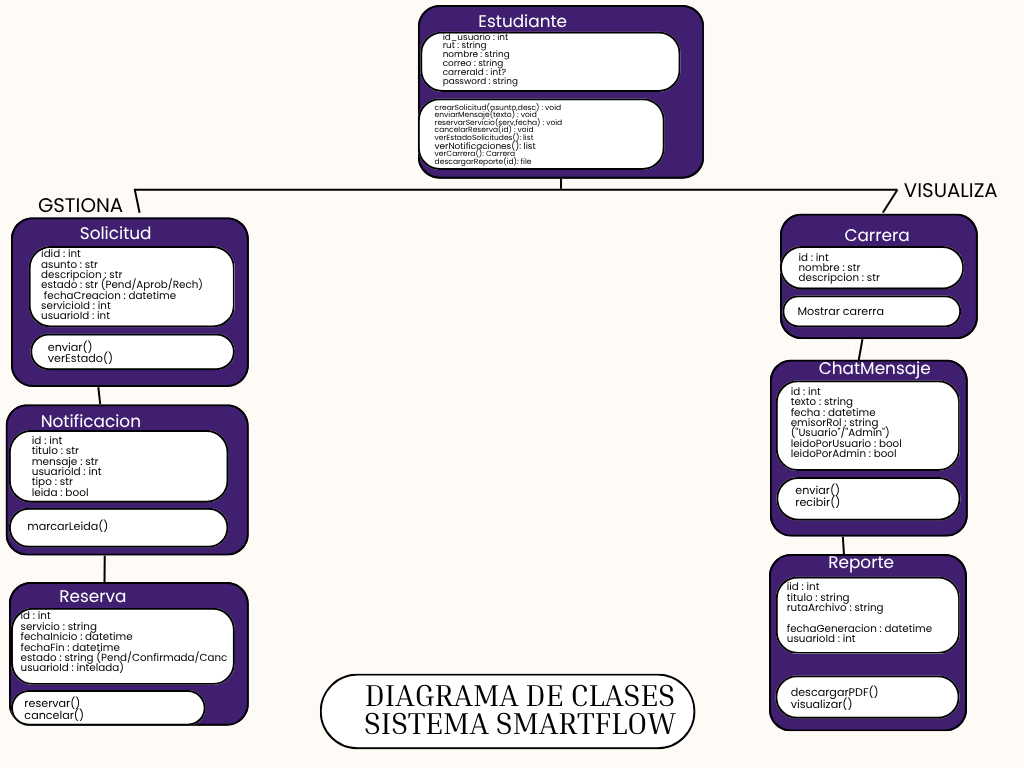
1. 

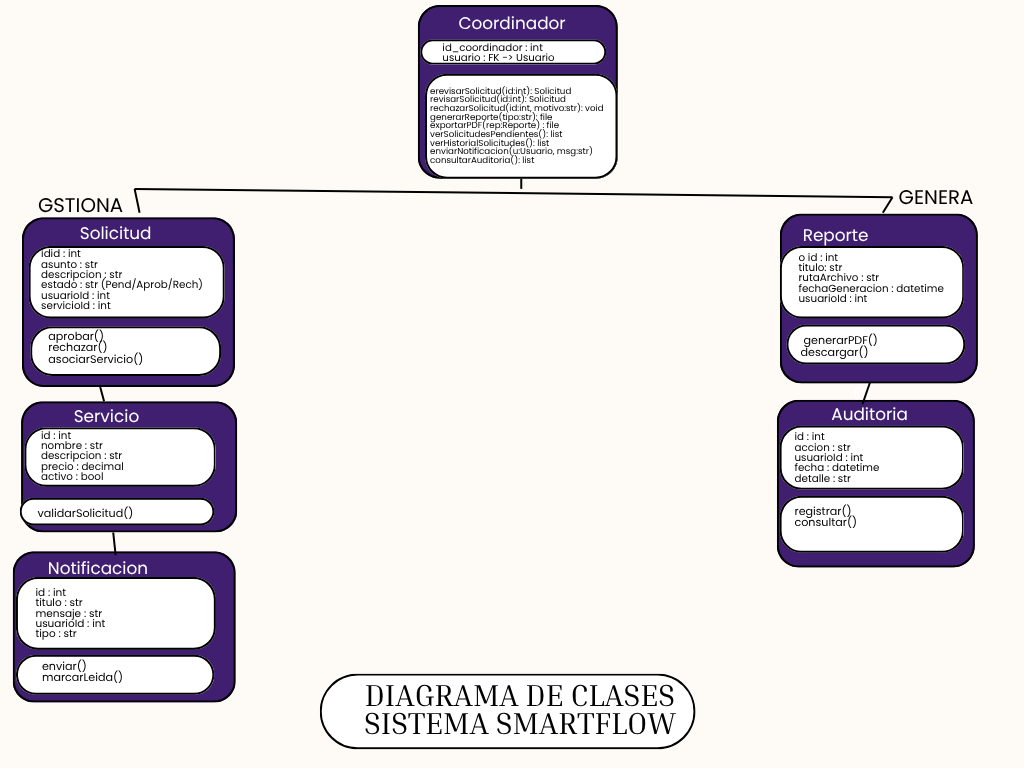
**5 Vista Lógica**

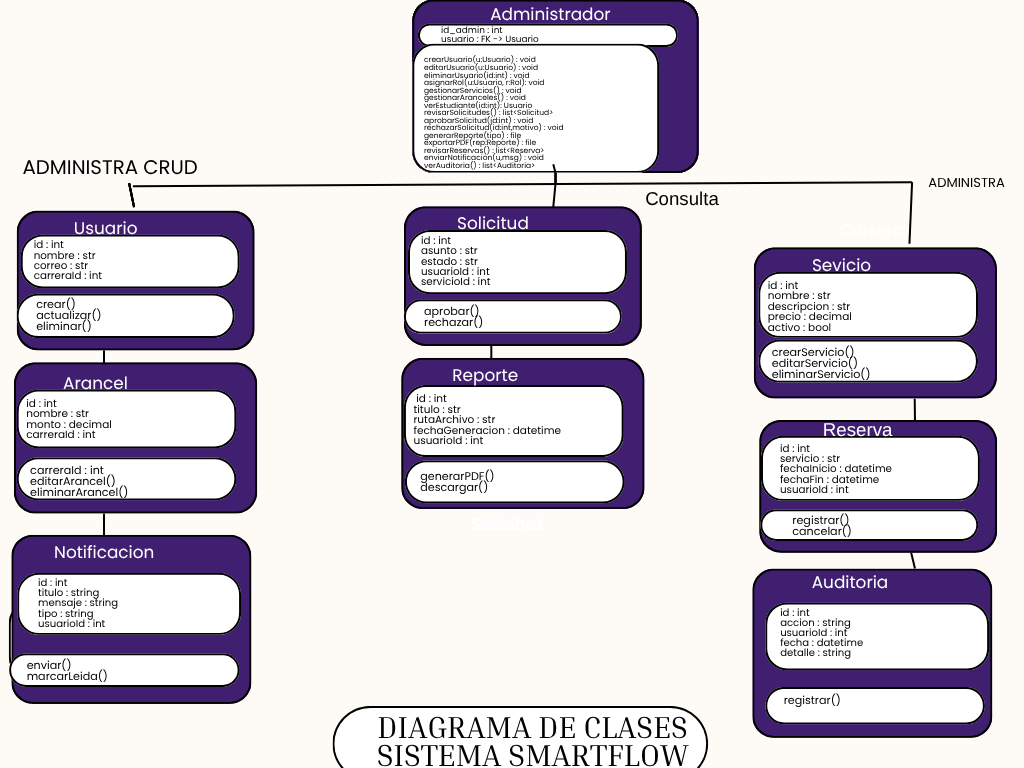
A continuación, se presenta una vista lógica de la aplicación expresado en tres diagramas, uno de ellos que muestra la parte estructural o estática de la aplicación (clases) y a la base de datos (modelo relacional).otra vista que representa la parte dinámica (secuencias).

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









### Descripción de Clases

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| CL-001 | Usuario | Representa a cualquier persona que utiliza el sistema (estudiantes, coordinadores, administradores). Contiene credenciales y datos básicos. |
| CL-002 | Estudiante | Usuario que puede crear solicitudes, reservar horas, visualizar notificaciones, comunicarse y descargar reportes. |
| CL-003 | Coordinador | Usuario encargado de aprobar/rechazar solicitudes, gestionar servicios, generar reportes y revisar auditorías. |
| CL-004 | Administrador | Usuario con privilegios altos para administrar CRUD de usuarios, servicios, aranceles, reservas y auditorías del sistema. |
| CL-005 | Solicitud | Representa una petición realizada por un estudiante para agendar una reunión o servicio. Registra estado, descripción, fecha y servicio asociado. |
| CL-006 | Reserva | Registra la reserva de una cita. Contiene fechas, servicio, estado (pendiente/confirmada/cancelada) y usuario asociado. |
| CL-007 | Notificación | Mensajes enviados al usuario sobre cambios de estado, confirmaciones, alertas o recordatorios del sistema. |
| CL-008 | Reporte | Documento generado automáticamente por el sistema con información de solicitudes, reservas o auditorías. Puede descargarse en PDF. |
| CL-009 | Auditoría | Registra eventos importantes del sistema (acciones críticas hechas por usuarios). Permite trazabilidad interna. |
| CL-010 | Servicio | Representa los servicios disponibles para agendamiento. Gestionado por coordinación y administración. |
| CL-011 | Carrera | Contiene información de la carrera del estudiante y se usa para contextualizar solicitudes y reportes. |
| CL-012 | CalendarioInterno | Módulo interno que sincroniza, agrega y elimina eventos relacionados a reservas y reuniones. |
| CL-013 | ChatMensaje | Representa los mensajes del canal de comunicación entre estudiante y administración/coordinación. |
| CL-014 | Arancel | Información de valores asociados a trámites o servicios internos. Gestionado por administrador. |
| CL-015 | SmartFlow (Core) | Clase principal del sistema. Coordina notificaciones, reportes, calendario, reservas y procesos automáticos. |
| CL-016 | EventoCalendario | Representa un evento individual dentro del calendario interno (bloques, horarios, reuniones). |
| CL-017 | HistorialEstadoSolicitud | Registra cambios de estado de una solicitud, manteniendo trazabilidad. |
| CL-018 | ConfiguraciónSistema | Contiene parámetros globales del sistema (entorno, políticas, rutas de archivo). Administrada por el Administrador. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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

**Diagrama

El contenido generado por IA puede ser incorrecto.**

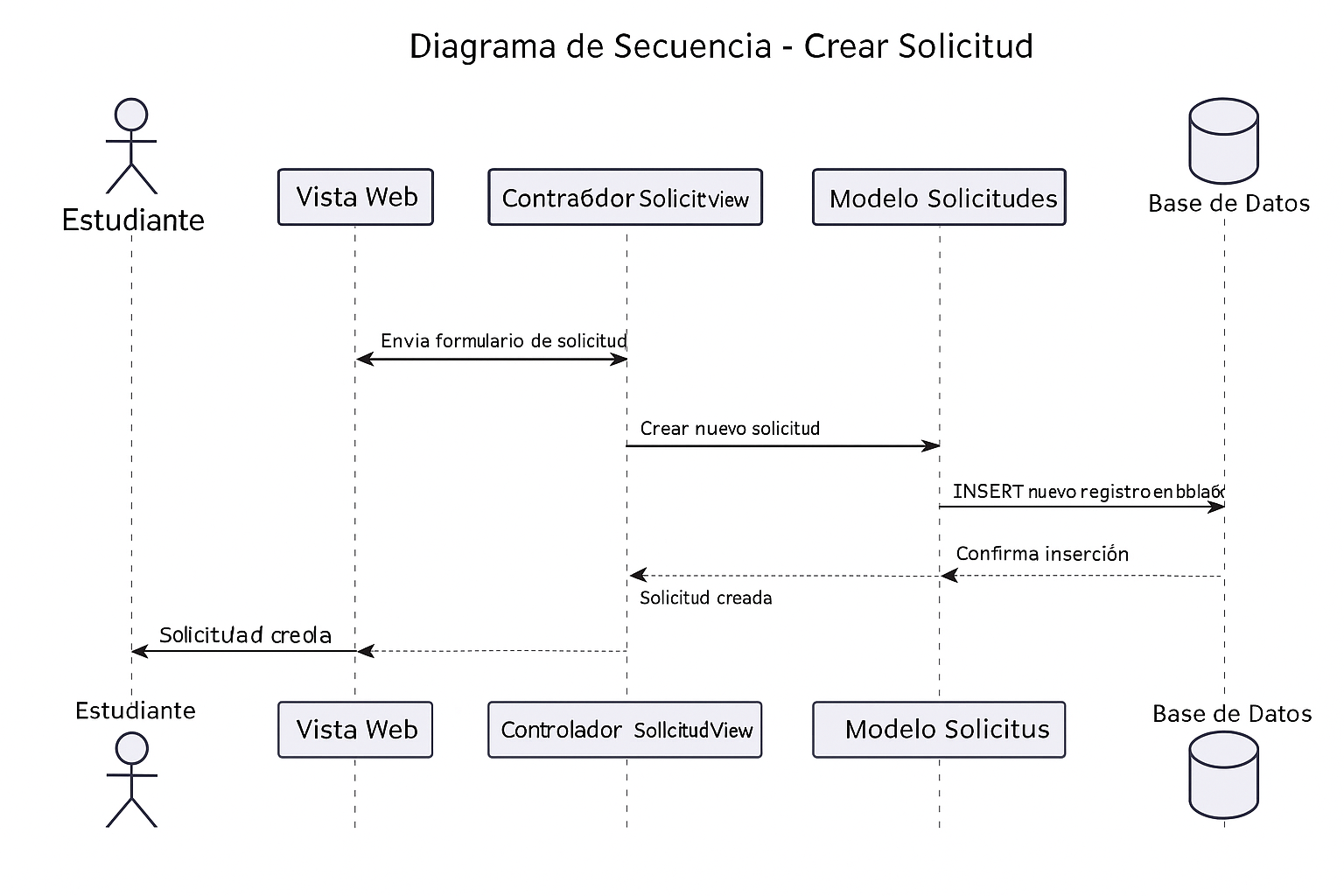
### Descripción de Tablas

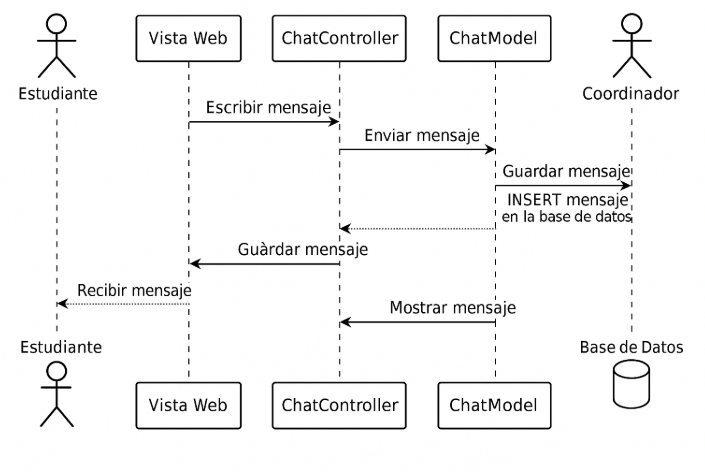
|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| TB-001 | USUARIO | Contiene la información base de cada usuario del sistema: credenciales, rol asignado, datos personales y fecha de registro. |
| TB-002 | ROL | Define los tipos de rol disponibles en la plataforma (Administrador, Coordinador, Estudiante). Se relaciona con la tabla Usuario. |
| TB-003 | ESTUDIANTE | Almacena información académica del estudiante y su relación con un usuario y una carrera. |
| TB-004 | CARRERA | Registra las carreras disponibles dentro de la institución. Se relaciona con Estudiante y Arancel. |
| TB-005 | ARANCEL | Define los aranceles asociados a cada carrera, incluyendo monto y descripción. |
| TB-006 | SERVICIO | Lista los servicios que pueden ser solicitados (consultas, trámites, certificados). Incluye duración y descripción. |
| TB-007 | SOLICITUD | Representa las solicitudes creadas por los estudiantes, indicando servicio requerido, estado, fecha y comentarios. |
| TB-008 | RESERVA | Registra la reserva generada desde una solicitud, incluyendo fecha, hora y estado de la reserva. |
| TB-009 | ESTADO\_SOLICITUD | Historial de cambios de estados para cada solicitud, registrando el estado anterior, nuevo y fecha del cambio. |
| TB-010 | NOTIFICACION | Guarda las notificaciones enviadas a los usuarios, su contenido y si fueron leídas. |
| TB-011 | AUDITORIA | Registra acciones importantes realizadas por los usuarios (login, creación, modificaciones), junto con fecha e IP. |
| TB-012 | CHATMENSAJE | Contiene mensajes enviados entre usuarios dentro del sistema, con emisor, receptor y contenido. |
| TB-013 | CALENDARIO\_EVENTO | Almacena los eventos del calendario relacionados a solicitudes o reservas, indicando fecha, tipo y referencia. |

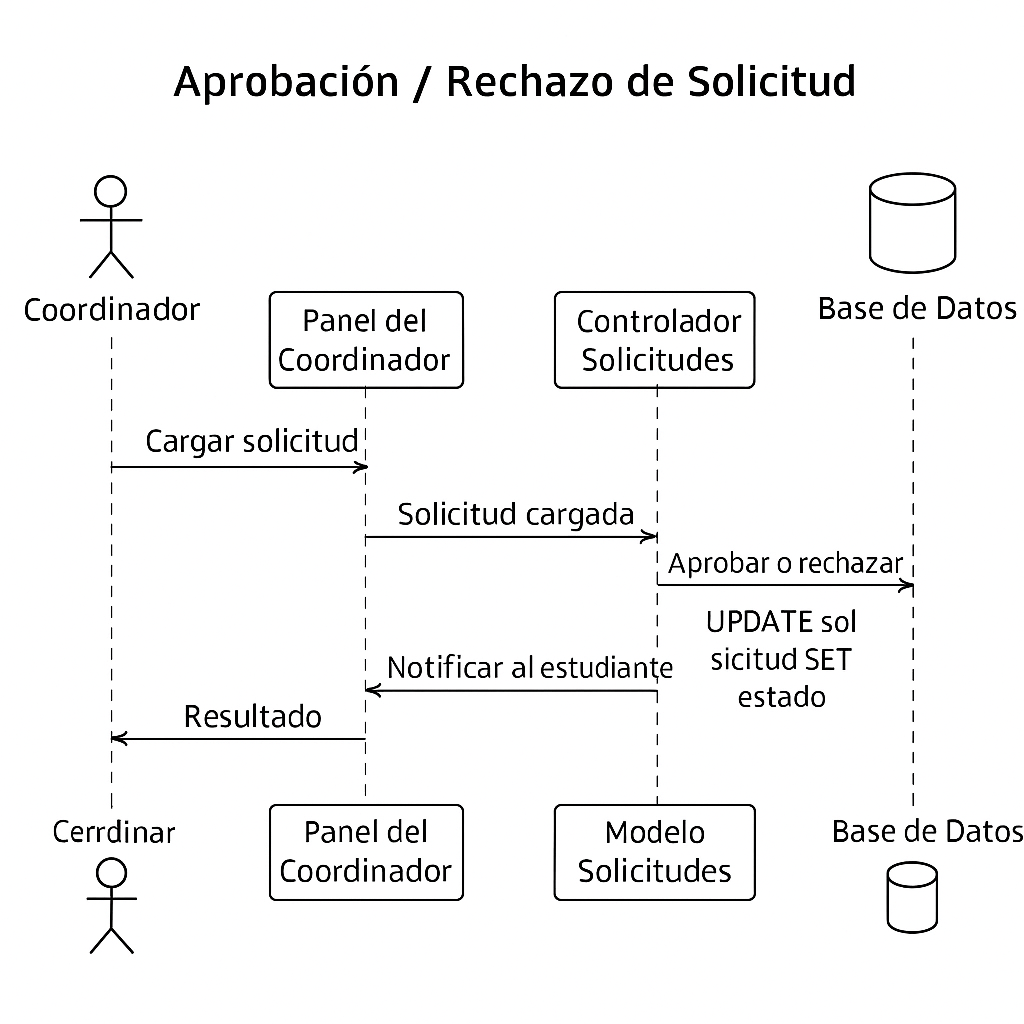
* 1. **Parte Dinámica (Diagrama de Secuencias)**

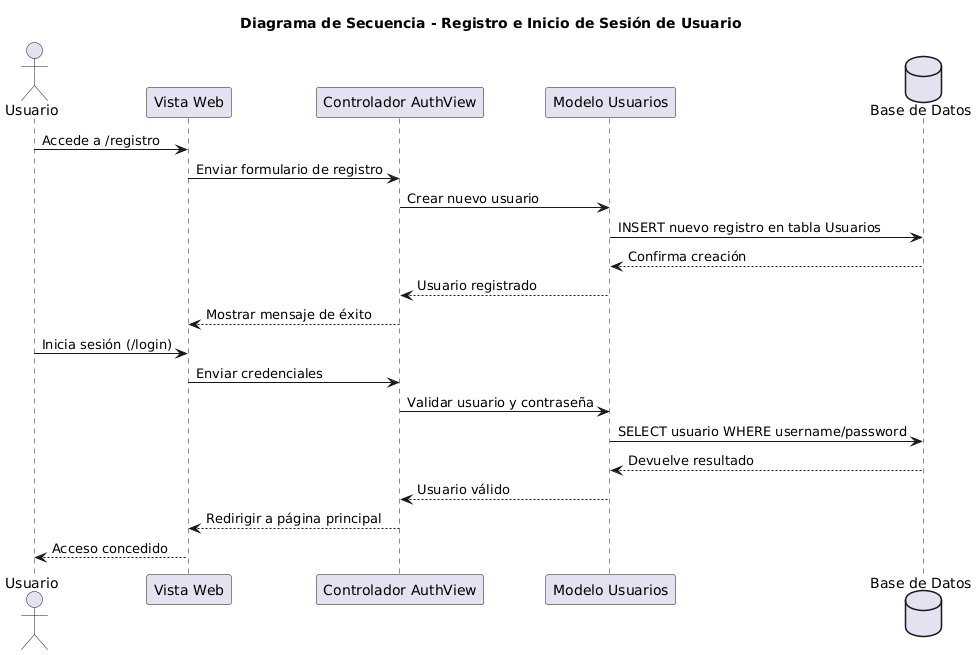
Diagrama

El contenido generado por IA puede ser incorrecto.

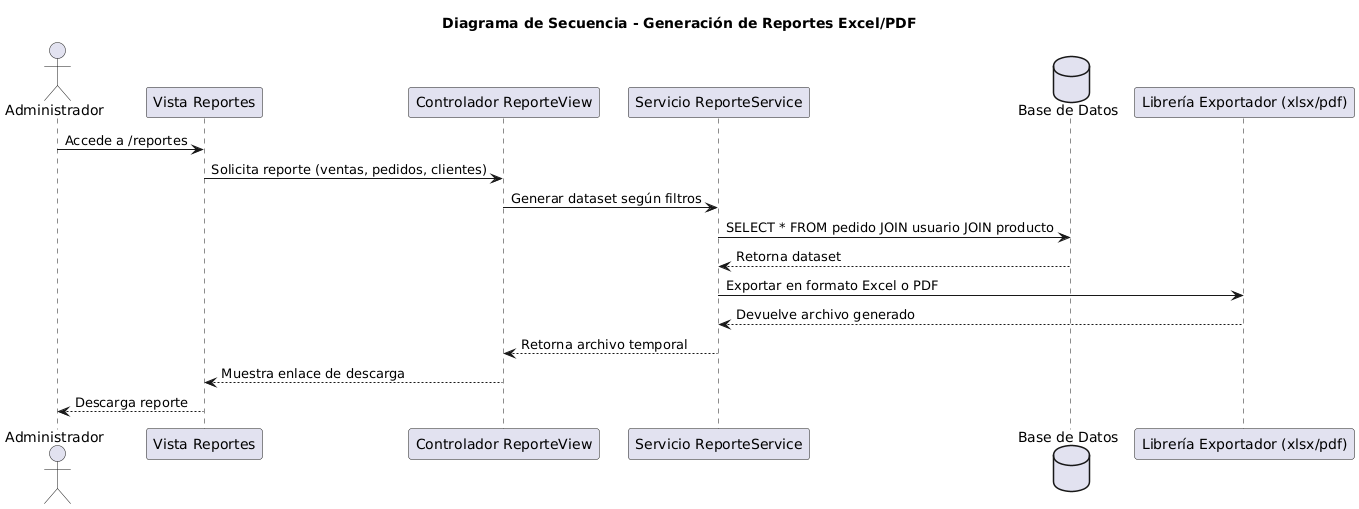






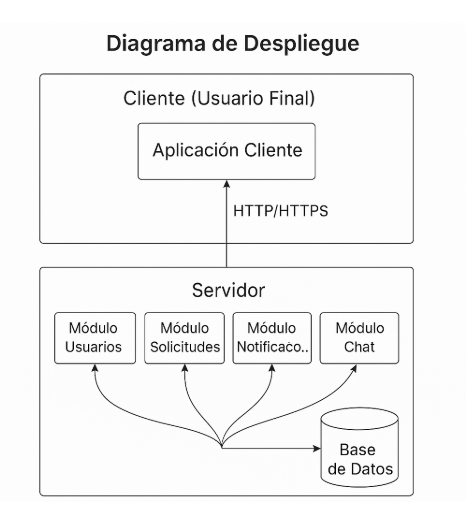


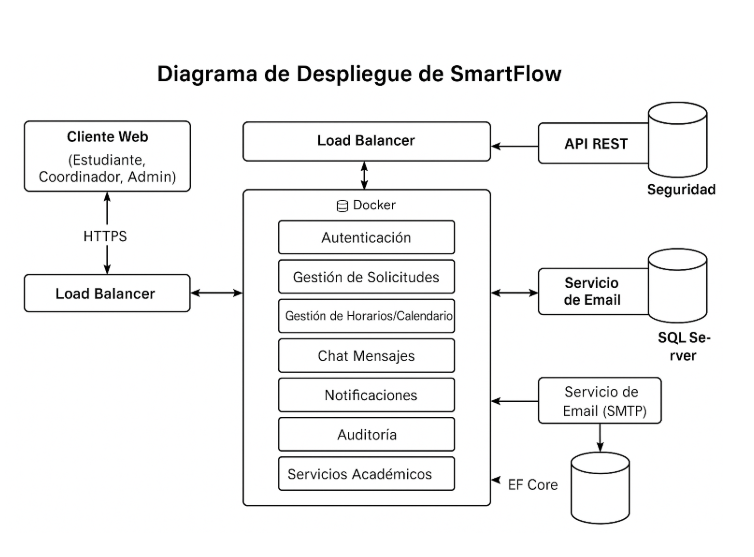
**Ilustración : Generación de reportes (Excel/PDF)**



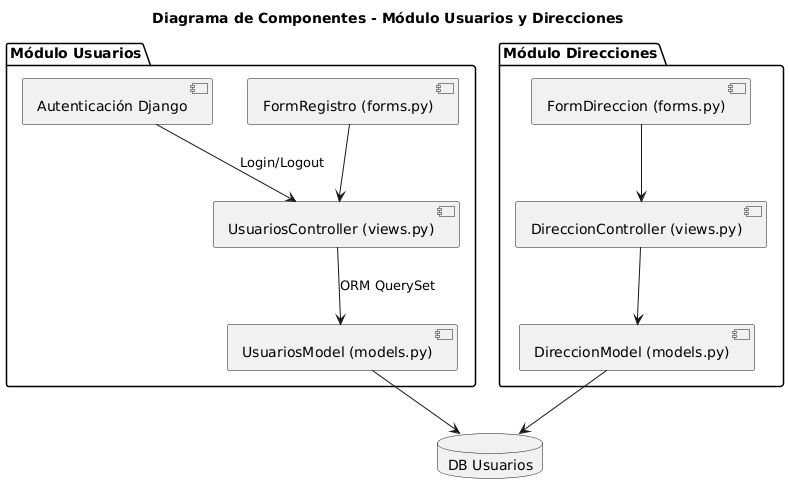
1. **Vista de Desarrollo o Despliegue**

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



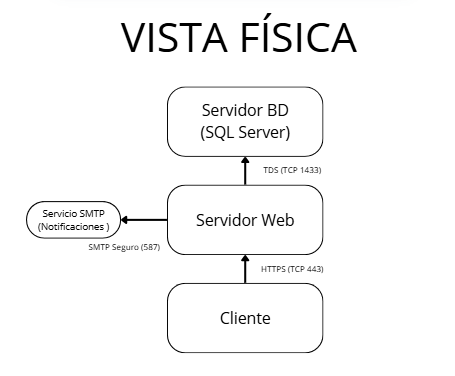


**Ilustración 2: Diagrama de componentes modulo usuarios y direcciones**

****

**7. Vista Fisica**

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



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

|  |  |
| --- | --- |
| Elemento Reutilizado | Descripción |
| Estructuras CRUD | Reutilización del mismo patrón CRUD en Usuarios, Servicios, Carreras y Aranceles, incluyendo validación, formularios y Razor Pages. |
| Layout General (\_Layout.cshtml) | Plantilla principal utilizada en todas las vistas: Panel Admin, Usuario, Reservas, Reportes y Notificaciones. |
| Componentes EF Core | Reutilización del SmartFlowContext y modelos como Usuario, Rol, Reserva, Servicio, Arancel, Solicitud y Notificación. |
| Módulo de Notificaciones | Un único componente para enviar alertas de reserva, cancelación, solicitudes y respuestas del coordinador. |
| Lógica de Disponibilidad | El mismo algoritmo para validar horarios libres se reutiliza en reservas y administración de servicios. |
| Plantillas Bootstrap | Reutilización de tarjetas, tablas, formularios y modales en todo el sistema para mantener coherencia visual. |
| Mecanismo de Auditoría | Un componente central registra cambios de creación, edición, eliminación y aprobaciones. |
| Control de Acceso y Roles | Una sola lógica administra permisos, dashboards y restricciones en todas las vistas según el rol del usuario. |

1. **Análisis de Reutilización**

Se reutilizará el código del mantenedor de Usuarios para implementar los módulos de Coordinadores y Administradores, dado que comparten atributos base como nombre, correo, credenciales, rol y estado.

Se reutilizarán las plantillas gráficas del sistema (paleta de colores, tipografía, componentes visuales y estructura de tarjetas) en los módulos de Solicitudes, Reservas y Calendario, asegurando coherencia visual en toda la aplicación.

Se reutilizará la lógica del sistema de notificaciones implementada en la creación de solicitudes para los eventos de Solicitud Aprobada, Solicitud Rechazada y Reserva Generada, ya que comparten el mismo flujo y estructura de datos.

Se reutilizará el módulo de Auditoría, debido a que funciona de manera transversal registrando acciones relevantes en todos los procesos del sistema, incluyendo Usuarios, Solicitudes, Reservas y Aranceles.

Se reutilizará el mecanismo de validación de usuario y roles, con el fin de controlar el acceso a las vistas del Panel Estudiante, Panel Coordinador y Panel Administrador, sin duplicar lógica de autenticación.

Se reutilizarán los componentes de carga de datos (DTO y ViewModels) diseñados para el módulo de Solicitudes, permitiendo estandarizar la forma en que la Vista, el Controlador y el Modelo intercambian información en todos los módulos.

Se reutilizará la estructura del controlador base, que gestiona respuestas estándar, manejo de errores y validaciones, garantizando consistencia en toda la API del sistema.