**[RecasteSoft]**

**(DAS) Documento Arquitectura Sistema**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | DAS-RescateSoft-001 |
| --- | --- |
| **Proyecto** | RescateSoft |
| **Versión** | 1.0 |

| **Documento mantenido por** | Luis Valenzuela |
| --- | --- |
| **Fecha de última revisión** | 09/10/2024 |
| **Fecha de próxima revisión** | 07/10/2024 |

| **Documento aprobado por** | Victor Reyes |
| --- | --- |
| **Fecha de última aprobación** | 07/10/2024 |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 11/09/2024 | 1.0 | Comienzo de documento | Luis Valenzuela |
| 23/09/2024 | 1.1 | Modificación de documento | Felipe Vega |
| 25/09/2024 | 1.2 | Modificación de documento | Victor Reyes |
| 29/09/2024 | 1.3 | Modificación de documento | Felipe Vega |
| 04/10/2024 | 1.4 | Modificación de documento | Felipe Vega |
| 07/10/2024 | 1.5 | Finalización del documento | Luis Valenzuela |

**Tabla de Contenidos**

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

[1.1](#_heading=h.30j0zll) Contexto del Problema 3

[1.2](#_heading=h.1fob9te) Propósito 3

[1.3](#_heading=h.3znysh7) Ámbito 3

[1.4](#_heading=h.2et92p0) Definiciones, acrónimos y abreviaciones 4

[1.5](#_heading=h.tyjcwt) Referencias 4

[1.6](#_heading=h.3dy6vkm) Resumen ejecutivo 4

[1.7](#_heading=h.1t3h5sf) Representación 5

[**2**](#_heading=h.4d34og8) **Metas y Restricciones de la Arquitectura 6**

[2.1](#_heading=h.2s8eyo1) Metas de la arquitectura 6

[2.2](#_heading=h.17dp8vu) Restricciones de la Arquitectura 6

[2.3](#_heading=h.3rdcrjn) Otros antecedentes y consideraciones 6

[**3**](#_heading=h.26in1rg) **Vista de Escenarios 7**

[3.1](#_heading=h.lnxbz9) Modelo de Casos de Uso 7

[3.2](#_heading=h.35nkun2) Casos de Usos Extendidos 8

[3.3](#_heading=h.1ksv4uv) Especificación de los Escenarios de Calidad Relevantes 9

[**4**](#_heading=h.44sinio) **Vista de Procesos 12**

[**5**](#_heading=h.2jxsxqh) **Vista Lógica 13**

[5.1](#_heading=h.z337ya) Parte Estructural ( Diagrama de Clases y Diagrama Relacional) 13

[*5.1.1*](#_heading=h.3j2qqm3) *Descripción de Clases* 13

[*5.1.2*](#_heading=h.1y810tw) *Descripción de Tablas 1*5

[5.2](#_heading=h.4i7ojhp) Parte Dinámica (Diagrama de Secuencias) 16

[**6**](#_heading=h.2xcytpi) **Vista de Desarrollo o Despliegue 21**

[**7**](#_heading=h.1ci93xb) **Vista Fisica 22**

[**8**](#_heading=h.3whwml4) **Decisiones de Diseño y Selección de Alternativas 23**

[**9**](#_heading=h.2bn6wsx) **Análisis de Reutilización 24**

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

El cuerpo de bomberos actualmente enfrenta un reto significativo en la gestión eficiente de los incidentes a los que asiste. El proceso de recopilación de datos es manual, mediante formularios en papel, lo cual genera múltiples inconvenientes: la información es difícil de organizar, consultar y analizar, y la posibilidad de cometer errores en el registro es alta. Además, la falta de acceso rápido a datos históricos afecta directamente la toma de decisiones estratégicas y operativas, lo que ralentiza la capacidad de respuesta del cuerpo de bomberos ante emergencias recurrentes o emergencias con patrones similares.

El sistema **RescateSoft** ha sido diseñado para abordar estos problemas mediante la digitalización y centralización de la información relacionada con los incidentes. El sistema no solo facilitará el registro de datos en tiempo real desde cualquier dispositivo conectado, sino que también permitirá análisis avanzados para identificar patrones y mejorar la planificación. La solución propuesta incluye un dashboard interactivo que mostrará métricas clave, y un sistema de minería de datos que ayudará en la predicción de la ocurrencia de incidentes futuros.

* 1. **Propósito**

El propósito de este documento es proporcionar una descripción detallada de la arquitectura técnica de **RescateSoft**. En él, se explicará cómo los componentes del sistema interactúan entre sí, cómo están estructurados y cómo cada uno contribuye a resolver los problemas planteados en el contexto. Este documento servirá como una guía tanto para el equipo de desarrollo como para los administradores del sistema, detallando las tecnologías clave, los patrones de diseño utilizados y las decisiones técnicas adoptadas.

Además, este documento será fundamental para las fases posteriores del ciclo de vida del sistema, ya que sentará las bases para futuras expansiones, integraciones con otros sistemas y la evolución natural del software a lo largo del tiempo. La arquitectura del sistema está diseñada con un enfoque modular, lo que garantiza que pueda adaptarse y escalar según las necesidades de la organización.

* 1. **Ámbito**

Este documento de arquitectura técnica cubre todos los aspectos relacionados con la estructuración y diseño del sistema **RescateSoft**. Los principales componentes que se describirán incluyen:

* **Frontend (Interfaz de usuario)**: El diseño y arquitectura de las interfaces web en las que los usuarios interactúan directamente, asegurando que sean intuitivas, responsivas y accesibles desde diferentes dispositivos.
* **Backend (Lógica del servidor)**: Los servicios y controladores que manejan las operaciones de negocio, validaciones, y procesamiento de datos. Aquí se incluye la integración con la base de datos y la lógica para gestionar los flujos de trabajo de los incidentes.
* **Base de datos**: La estructura de las tablas, relaciones y restricciones que aseguran la integridad de los datos almacenados, así como su capacidad de consulta rápida y eficiente.
* **Seguridad y acceso**: Los mecanismos de autenticación, autorización y protección de datos sensibles para cumplir con las normativas de seguridad y garantizar la privacidad de la información gestionada por el sistema.
* **Escalabilidad y rendimiento**: Las consideraciones técnicas para que el sistema soporte un crecimiento en la cantidad de usuarios y datos sin degradar el rendimiento o comprometer la seguridad.
  1. **Definiciones, acrónimos y abreviaciones**

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *CU* | Caso de Uso |
| *EC* | Escenario de Calidad |
| *DS* | Diagrama de secuencia |
| *MVC* | Modelo Vista Controlador |
| *MVT* | Modelo Vista Template |

* 1. **Referencias**

A continuación, se listan las referencias a otros documentos :

* **Casos de Uso**
* **ERS**
* **Acta de constitución**
* **Diagrama de Clases**
* **Diagrama de Componentes**
* **Diagrama de Despliegue**
* **Diagramas Actividad**
* **Diagramas Secuencia** 
  1. **Resumen ejecutivo**

El presente documento ofrece una visión general de la arquitectura técnica del sistema **RescateSoft**. Este sistema ha sido diseñado para resolver problemas clave de gestión de datos en el contexto de emergencias atendidas por los bomberos. La arquitectura está basada en el patrón **Modelo Vista Controlador (MVC)**, separando claramente las responsabilidades entre la capa de presentación (frontend), la lógica de negocio (backend) y el manejo de datos (base de datos).

El frontend del sistema será desarrollado utilizando tecnologías modernas como **HTML5**, **CSS3**, y **JavaScript**, con frameworks como **React** o **Angular** para garantizar una experiencia de usuario fluida y responsiva. El backend estará construido utilizando un lenguaje robusto como **Python** o **Node.js**, con un enfoque en APIs RESTful para gestionar las interacciones entre el cliente y el servidor.

La base de datos relacional, utilizando un **DBMS** como **MySQL** o **PostgreSQL**, almacenará toda la información relacionada con los incidentes, usuarios, y operaciones del sistema. La arquitectura del sistema también contempla mecanismos de seguridad avanzados, incluyendo el uso de **cifrado** para proteger los datos tanto en tránsito como en reposo, además de medidas de **control de acceso** basado en roles (RBAC).

El sistema está diseñado para ser modular, escalable y adaptable, permitiendo futuras expansiones y la integración con otros sistemas de emergencia. Las decisiones de diseño y las tecnologías empleadas garantizan que **RescateSoft** sea eficiente, seguro, y capaz de gestionar grandes volúmenes de datos de manera efectiva.

* 1. **Representación**

La arquitectura del sistema **RescateSoft** 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 Escenarios:**Describe los casos de uso más significativos del sistema **RescateSoft**, como la **gestión de incidentes**, la **generación de predicciones** y la **creación de informes**. Presenta los actores clave (por ejemplo, bomberos, administradores, analistas) y una descripción detallada de los escenarios de uso asociados, considerando las interacciones más relevantes para la arquitectura.
* **Vista de Procesos:**Representa los **procesos involucrados en la operación** del sistema, como el **registro de incidentes**, la **sincronización de datos** entre los diferentes módulos y la **comunicación entre servicios web** y la base de datos. Describe también cómo estos procesos se comunican y cómo se manejan las dependencias entre ellos.
* **Vista Lógica:**Expone la **arquitectura lógica** de **RescateSoft**, desglosando los módulos principales como la **gestión de usuarios**, la **gestión de notificaciones**, el **motor de análisis de datos** y los componentes del **sistema de predicciones avanzadas**. Además, detalla cómo estos módulos se interconectan y las responsabilidades de cada uno, definiendo claramente las dependencias entre ellos.
* **Vista de Desarrollo o Despliegue:**Detalla los **componentes de deployment** del sistema, incluyendo las **instancias web**, los **servidores de datos** y las **bases de datos**. Describe cómo estos componentes se distribuyen en el entorno de producción, así como las dependencias y configuraciones necesarias para asegurar la correcta integración de los módulos.
* **Vista Física:**Define las **restricciones tecnológicas y normativas** que influyen sobre el diseño y despliegue de **RescateSoft**. Esto incluye las **tecnologías utilizadas** (como bases de datos SQL, servicios web, servidores en la nube), los **estándares de seguridad** aplicados (encriptación de datos, autenticación), y las **limitaciones de hardware y red** que condicionan el rendimiento del sistema.

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

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

* 1. **Metas de la arquitectura**

De acuerdo con las reuniones y análisis de los requerimientos, se han identificado los principales conductores iniciales de la arquitectura, que corresponden a las metas arquitectónicas iniciales (atributos de calidad) para **RescateSoft**. Estos atributos son fundamentales para garantizar un desempeño óptimo del sistema en situaciones críticas de rescate y emergencia. Algunos de estos son:

* **Desempeño:** El sistema debe ofrecer tiempos de respuesta rápidos en situaciones de emergencia.
* **Tolerancia a fallos:** Debe mantenerse operativo incluso ante fallos de componentes individuales.
* **Seguridad:** Garantizar la protección de datos sensibles, como información de incidentes y usuarios.
* **Modificabilidad/Reuso:** La arquitectura debe ser flexible y adaptable a futuras modificaciones.
* **Operatividad:** Soportar un funcionamiento continuo, 24/7.
* **Adaptabilidad:** Facilitar la implementación de nuevas funcionalidades de análisis de datos y predicciones.
* **Escalabilidad:** Soportar un número creciente de usuarios y transacciones.
  1. **Restricciones de la Arquitectura**

A partir de las reuniones con los stakeholders, se han identificado las siguientes restricciones que influyen en el desarrollo y despliegue de la arquitectura:

* **Tiempo de construcción:** Se cuenta con un plazo de **X semanas** para la construcción completa del sistema, con entregas parciales para validaciones.
* **Infraestructura:** El sistema se desplegará en servidores replicados y una base de datos SQL con **alta disponibilidad** para garantizar la operatividad continua.

**Otros componentes de software:** No se considera la adquisición de componentes de software adicionales que no sean open-source o que no sean compatibles con la plataforma actual.

* 1. **Otros antecedentes y consideraciones**

La empresa encargada del desarrollo de **RescateSoft** cuenta con un **framework** que contempla los siguientes componentes arquitectónicos que permiten satisfacer los requerimientos establecidos:

* **Framework Angular y Node JS:** Soporta la modularización y encapsulación de componentes, lo que facilita la mantenibilidad del sistema y mejora el rendimiento en tiempo de ejecución. Este framework es conocido por su alta fiabilidad en aplicaciones críticas.
* **Framework de seguridad:** Implementado para cumplir con los estándares de seguridad y proteger datos sensibles contra accesos no autorizados.
* **Normativas y Estándares:** El sistema cumplirá con los estándares de calidad establecidos por las normativas internacionales para sistemas de gestión de emergencias.

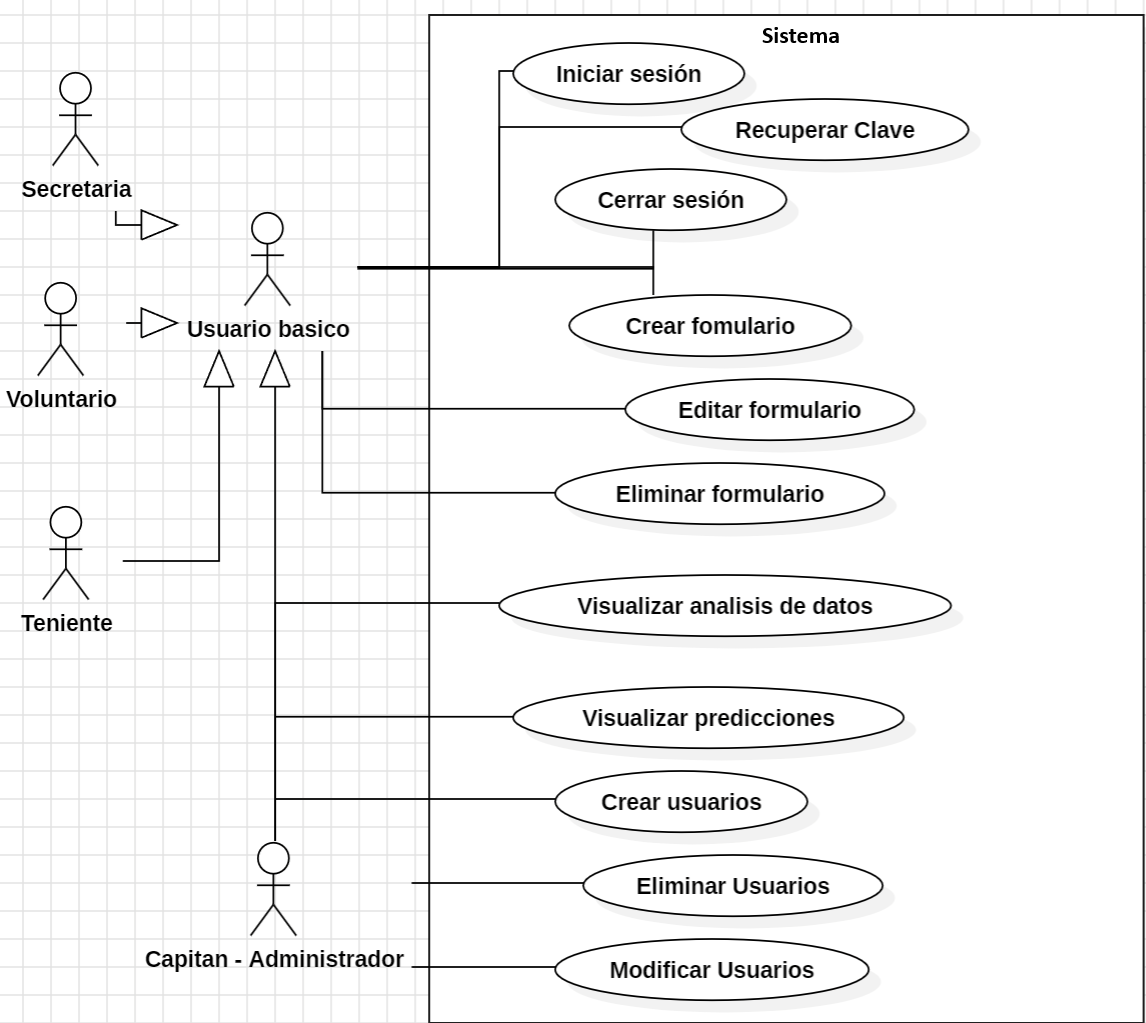
1. **Vista de Escenarios**

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**



* 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.

[**Planilla caso uso extendido**](https://docs.google.com/document/d/1mEBTfVs868T1PCEuRkIjsALUWtnGjVjl/edit?usp=drive_link&ouid=111123434177942630745&rtpof=true&sd=true)

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** |
| --- | --- | --- | --- |
| CU001 | Iniciar Sesión | Todos los usuarios | Alta |
| CU002 | Recuperar Clave | Todos los usuarios | Alta |
| CU003 | Cerrar Sesión | Todos los usuarios | Alta |
| CU004 | Crear Formulario | Todos los usuarios | Alta |
| CU005 | Editar Formulario | Todos los usuarios | Alta |
| CU006 | Eliminar Formulario | Todos los usuarios | Alta |
| CU007 | Visualizar Análisis de Datos | Capitán (Administrador) | Alta |
| CU008 | Visualizar Predicciones | Capitán (Administrador) | Alta |
| CU009 | Crear Usuarios | Capitán (Administrador) | Alta |
| CU010 | Eliminar Usuarios | Capitán (Administrador) | Alta |
| CU011 | Modifica Usuarios | Capitán (Administrador) | Alta |

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

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

| Identificador: EC01 | | |
| --- | --- | --- |
| Escenario(s): | | Alta concurrencia durante una emergencia significativa |
| Atributos de Calidad relevantes: | | Desempeño y escalabilidad |
| Componentes del Escenario | Estímulos: | Una gran cantidad de usuarios acceden al sistema simultáneamente para registrar y consultar datos |
| Fuente del estímulo | Usuarios conectados a la plataforma |
| Ambiente: | Producción, con todos los módulos funcionando |
| Artefacto: | Plataforma RescateSoft |
| Respuesta: | El sistema debe gestionar el acceso concurrente sin degradar el tiempo de respuesta |
| Medida de Respuesta | El tiempo de respuesta no debe superar los 2 segundos |

| Identificador: EC02 | | |
| --- | --- | --- |
| Escenario(s): | | Ataque de seguridad |
| Atributos de Calidad relevantes: | | Seguridad |
| Componentes del Escenario | Estímulos: | Intento de acceder al sistema con credenciales inválidas |
| Fuente del estímulo | Usuario externo no autorizado |
| Ambiente: | Producción |
| Artefacto: | SIstema de autenticación de RescateSoft |
| Respuesta: | El sistema debe bloquear el acceso y notificar a los administradores |
| Medida de Respuesta | El intento de acceso no autorizado debe ser bloqueado al tercer intento y generar una alerta automática |

| Identificador: EC03 | | |
| --- | --- | --- |
| Escenario(s): | | Pérdida de conexión durante el registro de un incidente |
| Atributos de Calidad relevantes: | | Tolerancia a fallos |
| Componentes del Escenario | Estímulos: | Se pierde la conexión a internet mientras se registra un incidente |
| Fuente del estímulo | Fallo de red |
| Ambiente: | Producción |
| Artefacto: | Plataforma RescateSoft (Modulo de registro de incidentes) |
| Respuesta: | El sistema debe guardar los datos temporalmente y permitir la sincronización cuando se restablezca la conexión |
| Medida de Respuesta | Los datos deben ser almacenados localmente en menos de 2 segundos y sincronizados automáticamente en menos de 5 segundos tras recuperar la conexión |

| Identificador: EC04 | | |
| --- | --- | --- |
| Escenario(s): | | Actualización de la plataforma sin interrumpir el servicio |
| Atributos de Calidad relevantes: | | Modificabilidad |
| Componentes del Escenario | Estímulos: | Se requiere actualizar una funcionalidad crítica del sistema |
| Fuente del estímulo | Equipo de desarrollo |
| Ambiente: | Producción |
| Artefacto: | Backend del sistema |
| Respuesta: | La actualización debe implementarse sin afectar la disponibilidad del sistema para los usuarios |
| Medida de Respuesta | El tiempo de inactividad percibido por los usuarios debe ser de 0 segundos |

| Identificador: EC05 | | |
| --- | --- | --- |
| Escenario(s): | | Recuperación de contraseña fallida |
| Atributos de Calidad relevantes: | | Seguridad y usabilidad |
| Componentes del Escenario | Estímulos: | Un usuario intenta recuperar contraseña, pero ingresa un código incorrecto o no puede completar el proceso |
| Fuente del estímulo | Usuario registrado en el sistema |
| Ambiente: | Producción |
| Artefacto: | Sistema de autenticación y recuperación de contraseñas de RescateSoft |
| Respuesta: | El sistema debe permitir un máximo de tres intentos antes de bloquear la recuperación de contraseña y requerir intervención manual por parte de soporte |
| Medida de Respuesta | El sistema debe bloquear el intento de recuperación de contraseña tras tres intentos fallidos y notificar el usuario del bloqueo |

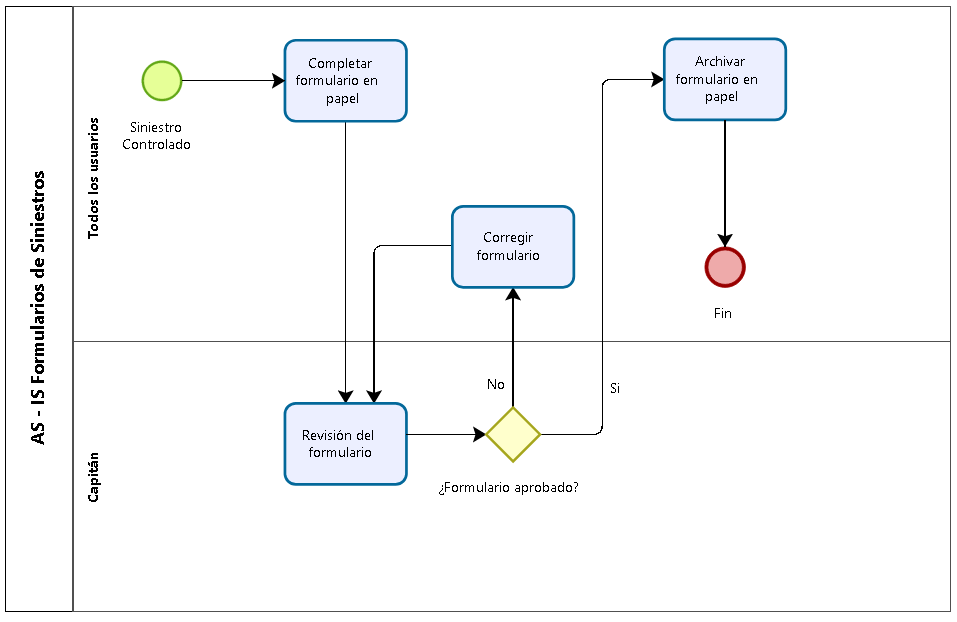
| Identificador: EC06 | | |
| --- | --- | --- |
| Escenario(s): | | Escalabilidad durante el análisis de grandes volúmenes de datos históricos |
| Atributos de Calidad relevantes: | | Desempeño y escalabilidad |
| Componentes del Escenario | Estímulos: | Se solicita generar un reporte de predicciones basado en un gran volumen de datos históricos |
| Fuente del estímulo | Administrador del sistema o capitán |
| Ambiente: | Producción |
| Artefacto: | Módulo de predicciones de RescateSoft |
| Respuesta: | El sistema debe generar el análisis sin ralentizar otros módulos del sistema, optimizando el uso de los recursos |
| Medida de Respuesta | El análisis debe completarse en menos de 10 segundos |

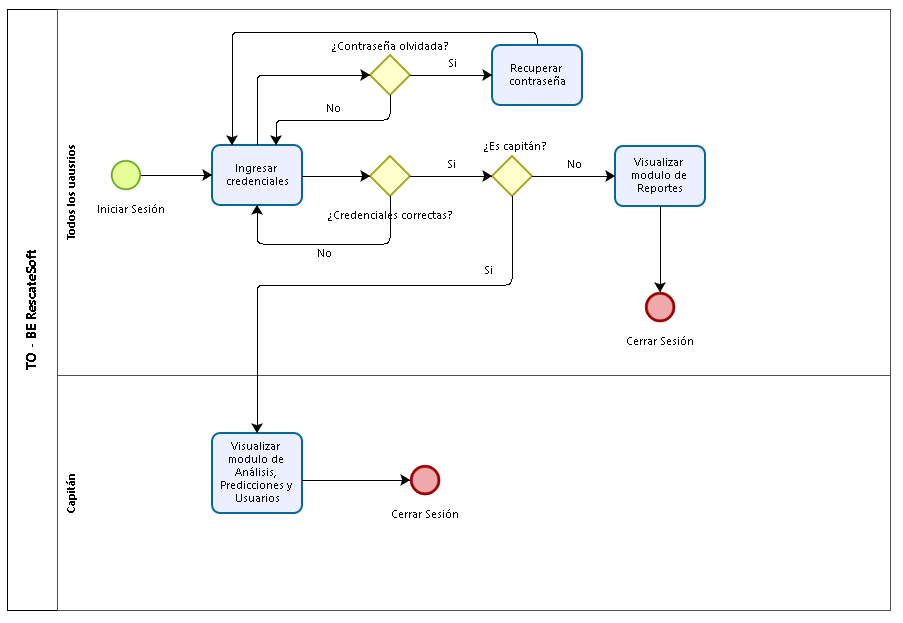
| Identificador: EC07 | | |
| --- | --- | --- |
| Escenario(s): | | Pérdida de datos durante apagón |
| Atributos de Calidad relevantes: | | Tolerancia a fallos y confiabilidad |
| Componentes del Escenario | Estímulos: | Se produce un apagón durante la entrada de datos en el sistema |
| Fuente del estímulo | Fallo de electricidad |
| Ambiente: | Producción |
| Artefacto: | Plataforma RescateSoft y su módulo de registro de incidentes |
| Respuesta: | El sistema debe guardar automáticamente los datos en un curso antes de que el apagón ocurra |
| Medida de Respuesta | Los datos en curso deben guardarse en menos de 1 segundo y restaurarse automáticamente al reanudar la conexión |

| Identificador: EC08 | | |
| --- | --- | --- |
| Escenario(s): | | Lentitud en la carga de formularios |
| Atributos de Calidad relevantes: | | Usabilidad y rendimiento |
| Componentes del Escenario | Estímulos: | Los usuarios reportan que los formularios tardan en cargarse cuando intentan editarlos o eliminarlos. |
| Fuente del estímulo | Usuarios conectados al sistema |
| Ambiente: | Producción |
| Artefacto: | Módulo de gestión de formularios de RescateSoft. |
| Respuesta: | El sistema debe mejorar la carga de formularios a través de optimización de consultas y caché. |
| Medida de Respuesta | El tiempo de carga de formularios debe reducirse a menos de 2 segundos con hasta 100 usuarios concurrentes. |

1. **Vista de Procesos**

**Ilustración 2: Diagramas de Actividades**



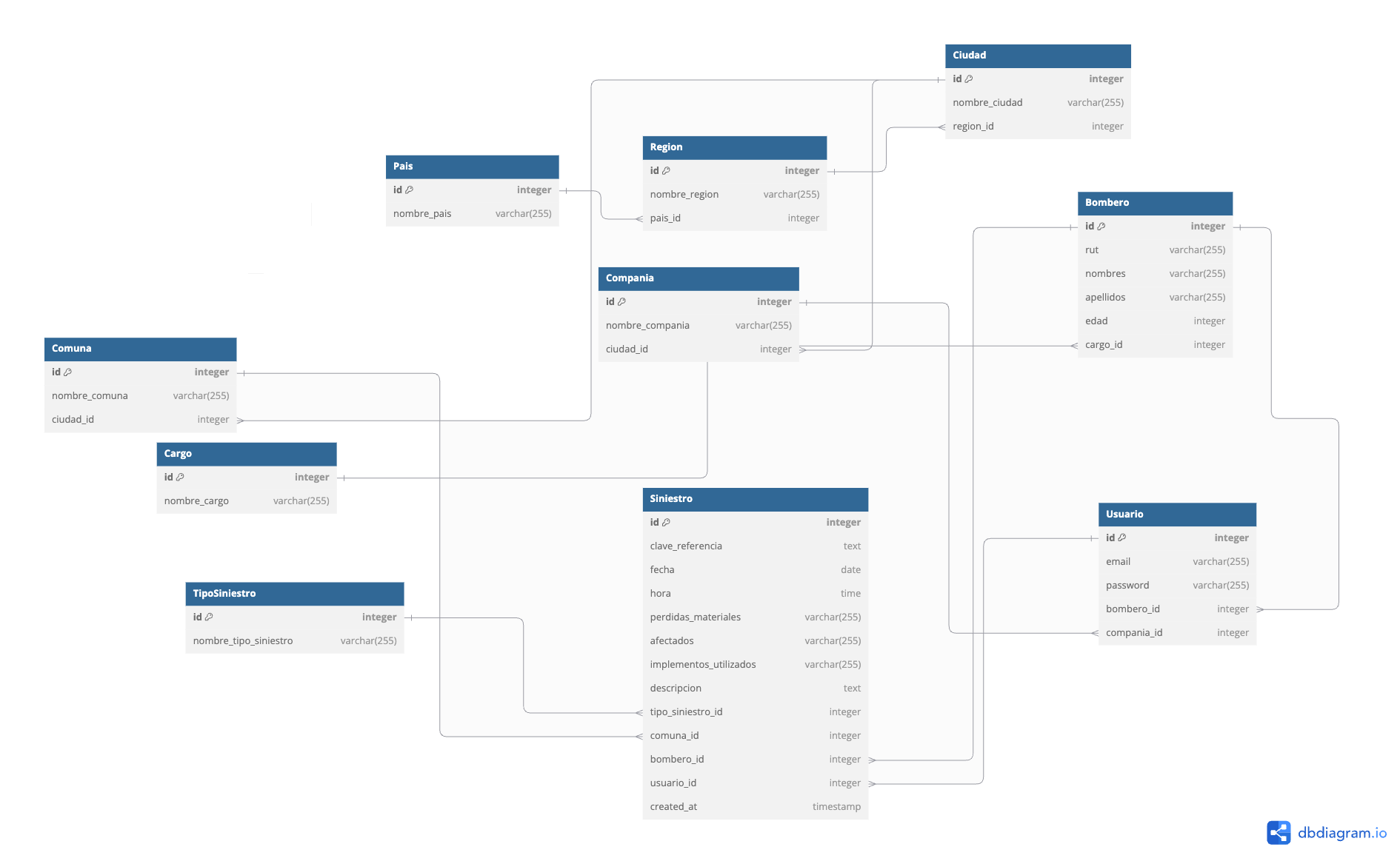


1. **Vista Lógica**

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)**

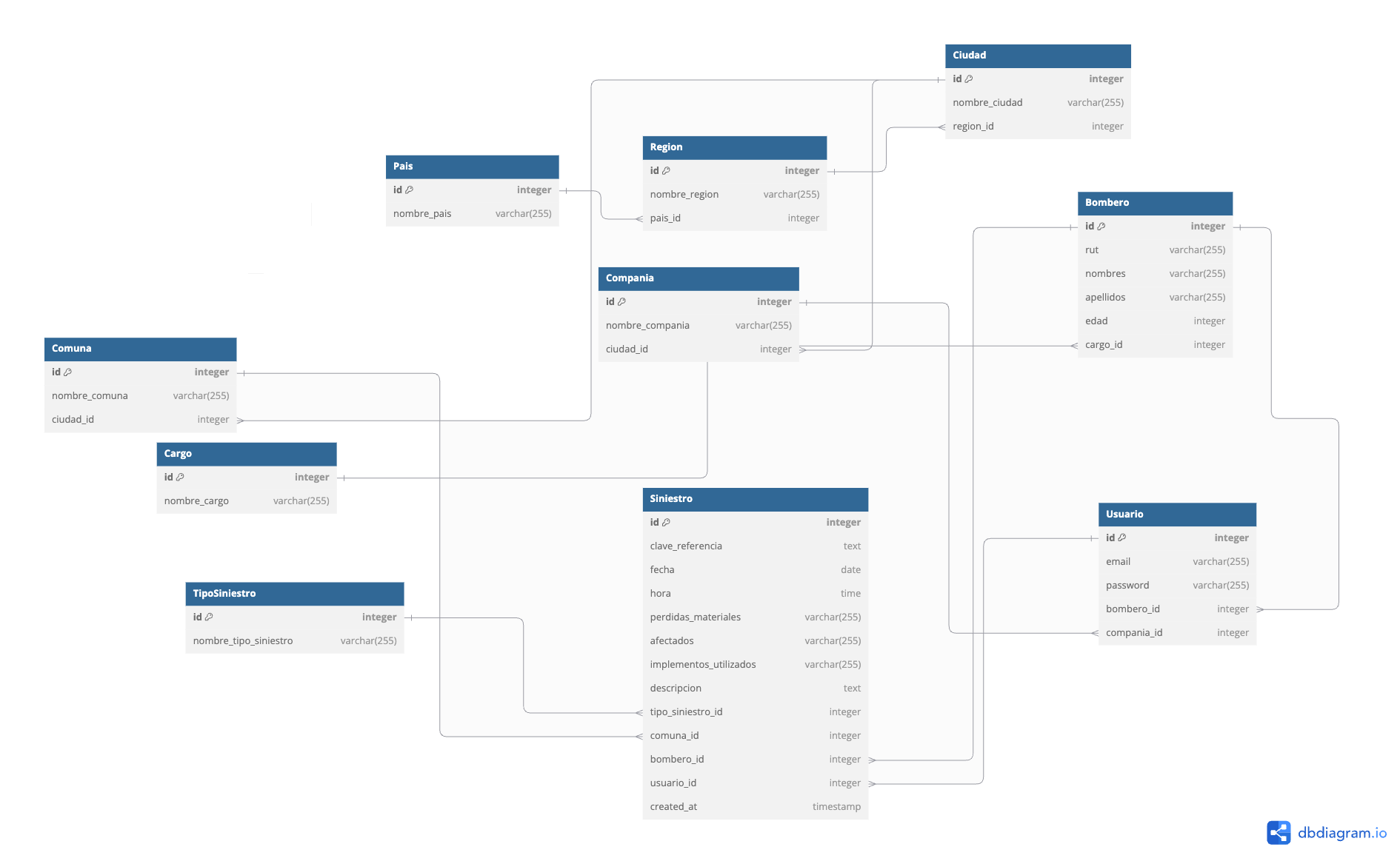
**Ilustración 3: Diagrama de Clases**



### Descripción de Clases

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| CL-01 | Bombero | Representa a los usuarios del cuerpo de bomberos. Contiene los datos personales y su cargo. Métodos principales: asignarSiniestro(), registrarIncidente(). |
| CL-02 | Siniestro | Representa los eventos de emergencia (incendios, accidentes, etc.). Métodos principales: registrarSiniestro(), asignarBombero(). |
| CL-03 | Usuario | La clase Usuario incluye los detalles de las personas que utilizan el sistema, ya sean bomberos o administradores. Los usuarios pueden iniciar sesión, gestionar formularios y reportes |
| CL-04 | Compañía | Representa una estación de bomberos o compañía en la que trabajan los bomberos. Una compañía tiene múltiples bomberos asociados. |
| CL-05 | Predicciones | La clase Predicciones es responsable de generar análisis predictivos basados en datos históricos de siniestros. |
| CL-06 | Reporte | Gestiona los reportes generados tras cada siniestro. Métodos principales: generarReporte(), verificarDatos(). |
| CL-07 | Formulario | Representa los formularios utilizados para registrar información relevante sobre los siniestros y usuarios. Métodos principales: crearFormulario(), validarInformacion(). |
| CL-08 | Cargo | Representa los diferentes cargos que pueden ocupar los bomberos en las compañías. Métodos principales: asignarCargo(), cambiarCargo(). |
| CL-09 | TipoSiniestro | Define los diferentes tipos de siniestros (incendios, rescates, accidentes, etc.). Métodos principales: definirTipo(), actualizarTipo(). |
| CL-10 | Localización | Contiene información sobre la ubicación específica del siniestro (país, región, ciudad, comuna). Métodos principales: registrarLocalizacion(), actualizarLocalizacion(). |
| CL-11 | SistemaDeAlertas | Se encarga de generar y gestionar las alertas para los bomberos y usuarios del sistema. Métodos principales: enviarAlerta(), configurarAlertas(). |
| CL-12 | Equipamiento | Gestiona los implementos utilizados durante un siniestro. Métodos principales: registrarEquipamiento(), actualizarEstadoEquipamiento(). |
| CL-13 | Historial | Mantiene un registro de las actividades y siniestros en los que ha participado cada bombero. Métodos principales: agregarRegistro(), consultarHistorial(). |

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



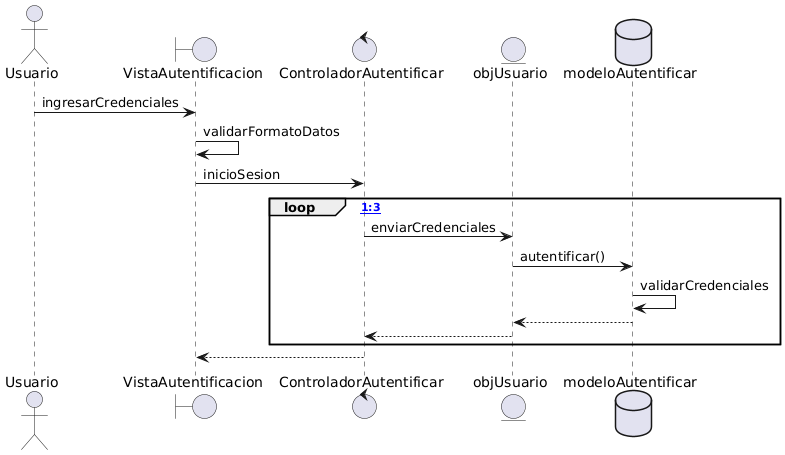
### Descripción de Tablas

| **Código** | **Nombre** | **Descripción** |
| --- | --- | --- |
| TB-01 | Bombero | Registro de los bomberos que incluye datos personales como RUT, nombres, apellidos, edad, y cargo asociado. |
| TB-02 | Usuario | Información de los usuarios del sistema, incluyendo email, contraseña, y la relación con bomberos y compañías. |
| TB-03 | Compañía | Registro de las compañías de bomberos, asociadas a una ciudad específica. |
| TB-04 | Siniestro | Detalles de los siniestros registrados, incluyendo fecha, hora, pérdidas materiales, afectados, implementos utilizados, y el bombero responsable. |
| TB-05 | TipoSiniestro | Clasificación de los siniestros en diferentes tipos para mejor categorización y análisis. |
| TB-06 | Cargo | Tipos de cargos que pueden tener los bomberos dentro de la organización. |
| TB-07 | País | Lista de países en los que se encuentran las regiones. |
| TB-08 | Región | División geográfica que agrupa a varias ciudades y está asociada a un país específico. |
| TB-09 | Ciudad | Información de las ciudades que forman parte de las regiones, con su nombre y referencia de región. |
| TB-10 | Comuna | Subdivisiones dentro de las ciudades para una mayor especificidad en la localización de siniestros. |

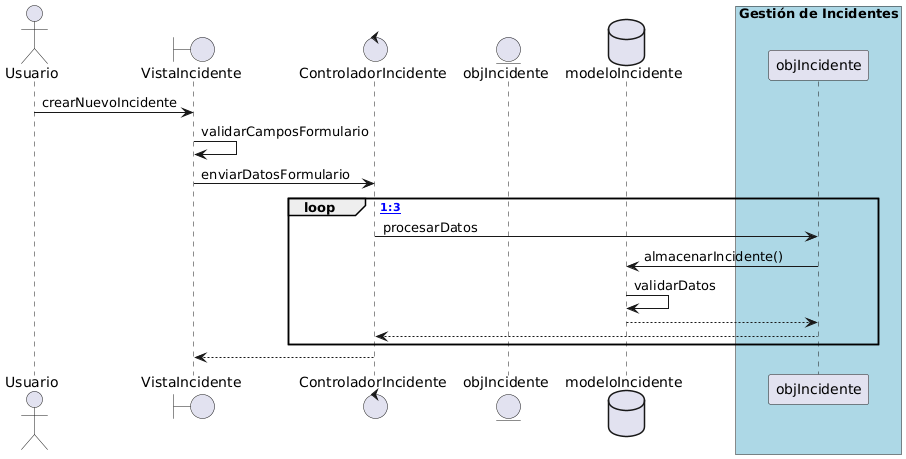
* 1. **Parte Dinámica (Diagrama de Secuencias)**

**Ilustración 5: Diagramas de Secuencias**

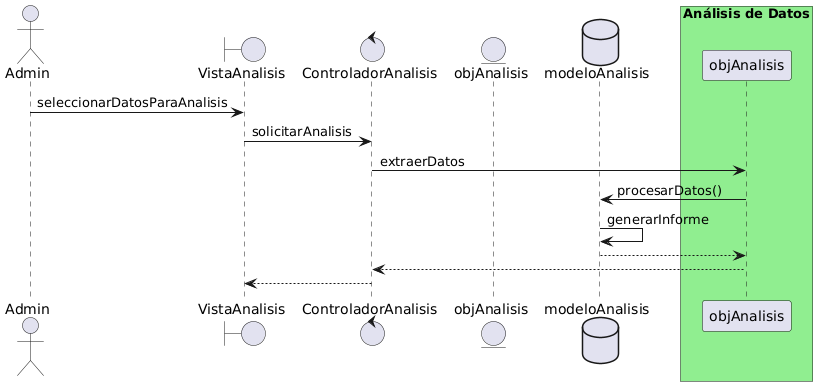
**DS-001: Login**



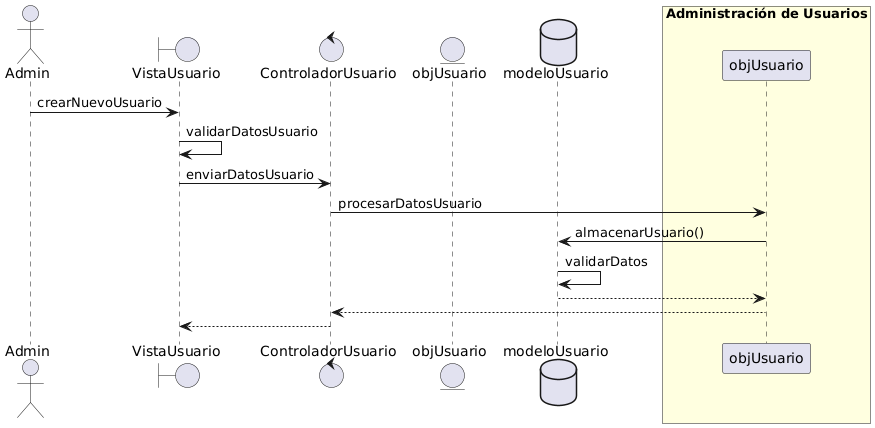
**DS-002: Incidentes**

****

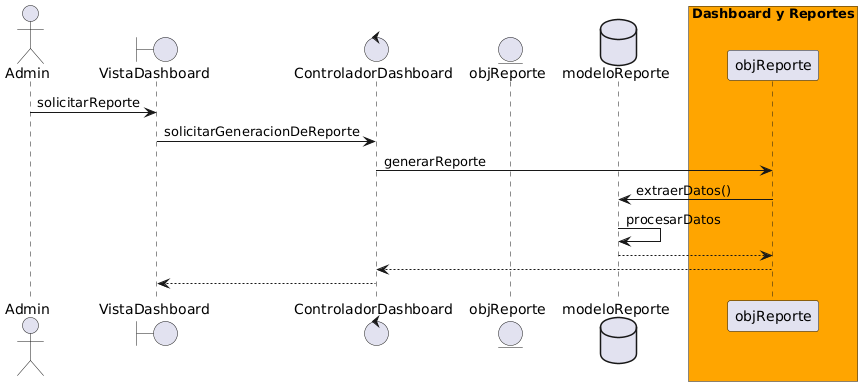
**DS-003: Analisis**

****

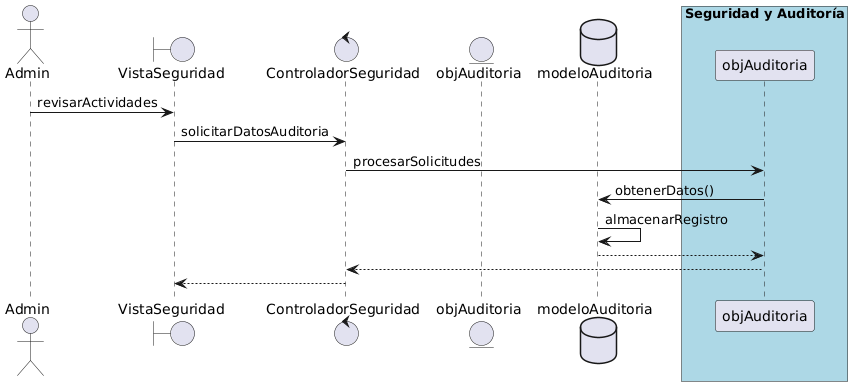
**DS-004: Creación Usuarios**

****

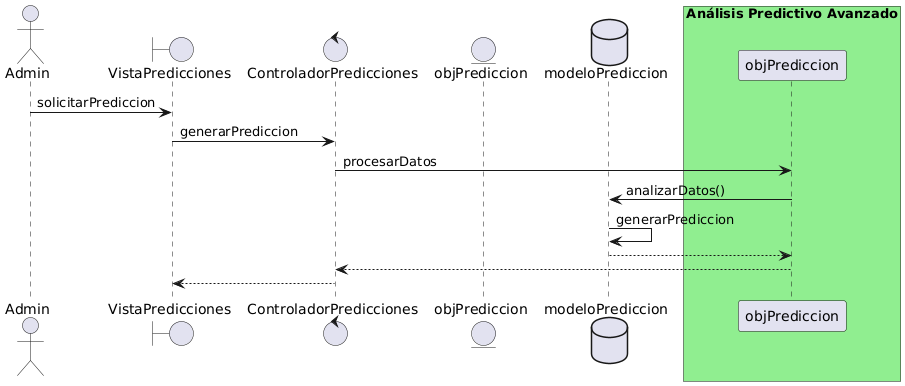
**DS-005: Dashboard**

****

**DS-007: Seguridad**

****

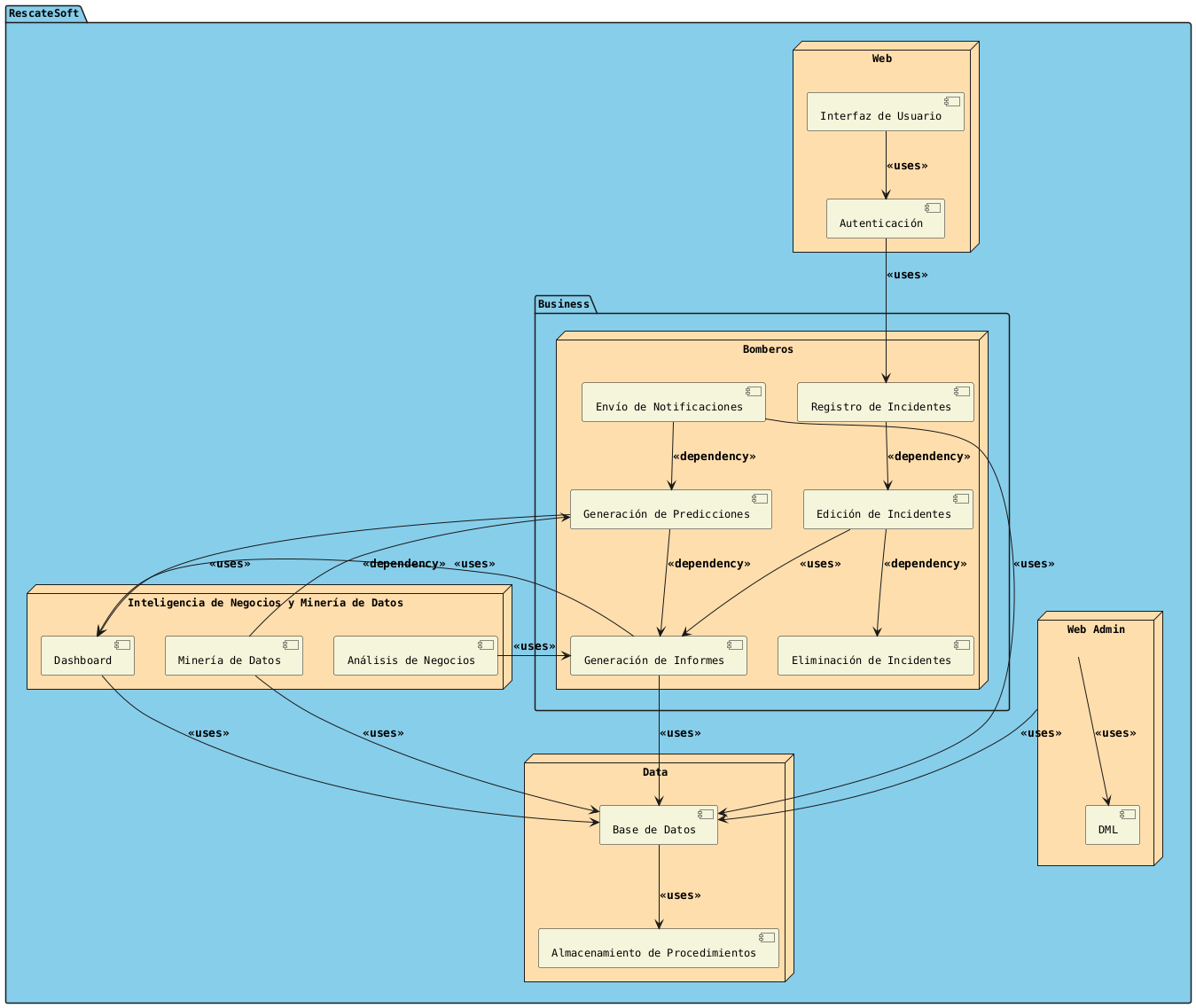
**DS-008: Mineria de datos**

****

1. **Vista de Desarrollo o Despliegue**

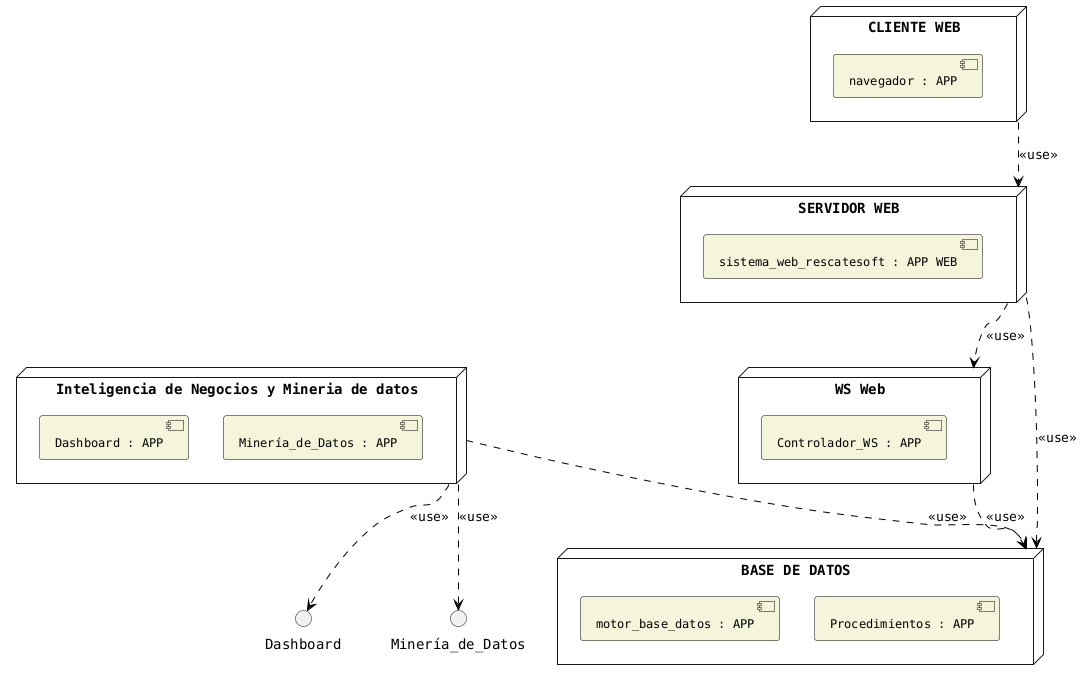
En esta vista se describen las componente o modulos en las cuales se dividirá o implementará el sistema

**Ilustración 6: Diagrama de componentes**

**7. Vista Física**

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

**Ilustración 7: Diagrama de Despliegue**

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

Las principales decisiones arquitectónicas se tomaron en consideración con base en las siguientes restricciones y necesidades del proyecto:

1. **Tiempo de Construcción:**Dado que el proyecto debe implementarse en un tiempo ajustado y sin márgenes de holgura, se decidió adoptar una **arquitectura conocida** que presente **bajo riesgo** en su implementación. Esta decisión permite reducir la curva de aprendizaje del equipo de desarrollo y mejorar los tiempos de entrega al usar tecnologías y patrones ya comprobados.
2. **Modularidad:**La arquitectura se **modularizó**, lo que significa que se dividió en diferentes módulos funcionales. Cada módulo es **testable unitariamente**, lo que facilita la identificación de errores y garantiza que cada componente tenga una **baja tasa de fallos**. La modularidad también permite la escalabilidad del sistema, ya que los módulos pueden ser ajustados o mejorados de forma independiente sin afectar el resto del sistema.
3. **Facilidad de Mantenimiento:**Al seguir un diseño modular, se asegura que cualquier mantenimiento futuro, ya sea para correcciones de errores o para mejoras, se pueda realizar con el menor impacto posible sobre el sistema. Además, se buscó que los módulos fueran altamente reutilizables, maximizando su flexibilidad.
4. **Selección de Alternativas:**Se analizaron diversas alternativas para los diferentes componentes del sistema, priorizando las siguientes características:
   * **Rendimiento:** Se eligieron tecnologías y componentes que garanticen un rendimiento adecuado para el manejo del volumen de datos y el número de usuarios esperados.
   * **Escalabilidad:** Se seleccionaron componentes que permitan una fácil escalabilidad horizontal y vertical para soportar el crecimiento del sistema en el futuro.
   * **Compatibilidad:** Se privilegiaron soluciones que sean compatibles con las tecnologías ya existentes en la infraestructura de la organización, facilitando la integración y evitando problemas de compatibilidad.

**9. Análisis de Reutilización**

En el desarrollo de **RescateSoft**, se identifican varias áreas clave en las que se aplicará la reutilización de código, plantillas y módulos con el fin de optimizar el tiempo de desarrollo, reducir errores y mantener la consistencia en el sistema. A continuación, se describen algunos ejemplos:

* **Servicios Comunes de Autenticación:**Se reutilizarán los **módulos de autenticación** ya existentes para todos los roles de usuario del sistema (administradores, bomberos, personal técnico). Esto incluye la verificación de credenciales y la gestión de sesiones. Esta reutilización asegura que las políticas de seguridad y control de acceso sean consistentes en todos los puntos de entrada del sistema.
* **Módulo de Notificaciones:**El **módulo de envío de notificaciones** será reutilizado en diferentes funcionalidades del sistema, como en la gestión de incidentes, auditorías y reportes. Este módulo centralizado permitirá notificar a los usuarios sobre incidentes, actualizaciones y reportes generados, sin necesidad de duplicar código en cada módulo.
* **Plantillas para Reportes y Dashboard:**Se utilizarán **plantillas predefinidas para la generación de reportes** y dashboards en los módulos de análisis de datos y auditoría. Estas plantillas compartirán una estructura visual unificada, facilitando el mantenimiento de los componentes gráficos y asegurando una presentación coherente de la información, independientemente del tipo de datos mostrado.
* **Algoritmos de Predicción Reutilizables:**Los **algoritmos de minería de datos y predicción**, que se utilizan en el módulo de predicciones avanzadas, serán reutilizados para el análisis de tendencias de incidentes, así como para la predicción de posibles futuros escenarios de emergencia. De esta forma, se evita desarrollar algoritmos redundantes en diferentes áreas del sistema.
* **Componente de Base de Datos:**Las consultas y procedimientos almacenados utilizados en el **motor de la base de datos** se reutilizarán en los diferentes módulos del sistema (gestión de incidentes, auditorías, generación de reportes), asegurando que todos los módulos trabajen con un conjunto de datos consistente y estructurado de forma unificada.
* **Validaciones de Entrada de Datos:**Las **rutinas de validación de datos** que se aplican en el ingreso de información en el módulo de gestión de incidentes, como la verificación de formatos y restricciones, serán reutilizadas en otros módulos del sistema (por ejemplo, en el registro de usuarios o en la carga de datos para los reportes). Esto garantiza que la validación de los datos sea coherente en toda la plataforma.
* **Interfaz de Usuario Unificada:**Se reutilizará el **diseño de interfaz de usuario** en todos los módulos del sistema, de manera que los formularios, menús y flujos de navegación mantengan un estilo coherente. Esto no solo mejora la experiencia del usuario, sino que también facilita la capacitación y el soporte técnico.