****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *Deadline Collision Predictor***

Curso: Calidad y Pruebas de Software

Docente: Patrick Cuadros Quiroga

Integrantes:

***Luna Juarez, Juan Brendon (2020068762)***

***Vargas Gutierrez, Angel Jose (2020066922)***

***Chino Rivera, Angel Alessandro (2021069830)***

**Tacna – Perú**

***2025***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema *Deadline Collision Predictor*

Documento de Especificación de Requerimientos de Software

Versión *{1.0}*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

**INDICE GENERAL**

[INTRODUCCION](#_Toc394513795) 4

[I. Generalidades de la Empresa](#_Toc394513799) 5

[1. Nombre de la Empresa 5](#_Toc394513800)

[2. Vision 5](#_Toc394513800)

[3. Mision 5](#_Toc394513800)

[4. Organigrama 5](#_Toc394513800)

[II. Visionamiento de la Empresa](#_Toc394513799) 5

[1. Descripcion del Problema 5](#_Toc394513800)

[2. Objetivos de Negocios 5](#_Toc394513800)

[3. Objetivos de Diseño 5](#_Toc394513800)

[4. Alcance del proyecto 5](#_Toc394513800)

[5. Viabilidad del Sistema 5](#_Toc394513800)

[6. Informacion obtenida del Levantamiento de Informacion](#_Toc394513800) 6

[III. Análisis de Procesos](#_Toc394513799) 6

[a) Diagrama del Proceso Actual – Diagrama de actividades](#_Toc394513800) 6

[b) Diagrama del Proceso Propuesto – Diagrama de actividades Inicial](#_Toc394513800) 7

[IV Especificacion de Requerimientos de Software](#_Toc394513799) 7

[a) Cuadro de Requerimientos funcionales Inicial](#_Toc394513800) 7

[b) Cuadro de Requerimientos No funcionales](#_Toc394513800) 7

[c) Cuadro de Requerimientos funcionales Final](#_Toc394513800) 8

[d) Reglas de Negocio](#_Toc394513800) 9

[V Fase de Desarrollo](#_Toc394513799) 12

[1. Perfiles de Usuario](#_Toc394513800) 12

[2. Modelo Conceptual 5](#_Toc394513800)

[a) Diagrama de Paquetes 5](#_Toc394513800)

[b) Diagrama de Casos de Uso](#_Toc394513800) 12

[c) Escenarios de Caso de Uso (narrativa)](#_Toc394513800) 14

[3. Modelo Logico](#_Toc394513799) 23

[a) Analisis de Objetos](#_Toc394513800) 23

[b) Diagrama de Actividades con objetos](#_Toc394513800) 32

[c) Diagrama de Secuencia](#_Toc394513800) 37

[d) Diagrama de Clases](#_Toc394513800) 42

[CONCLUSIONES](#_Toc394513803) 46

[RECOMENDACIONES](#_Toc394513804) 46

[BIBLIOGRAFIA](#_Toc394513805) 46

[WEBGRAFIA](#_Toc394513806) 46

**INTRODUCCIÓN**

Deadline Collision Predictor es una plataforma web diseñada para asistir a estudiantes en la gestión eficiente de sus tareas académicas. Mediante una interfaz intuitiva y un asistente inteligente, permite registrar actividades mediante lenguaje natural, visualizar eventos en un calendario interactivo y detectar conflictos entre fechas límite. El objetivo principal del sistema es reducir el estrés académico y mejorar la organización personal, alertando sobre posibles colisiones entre exámenes, entregas y otros compromisos.

**I. Generalidades de la Empresa**

**1. Nombre de la Empresa**

Deadline Tech S.A.C.

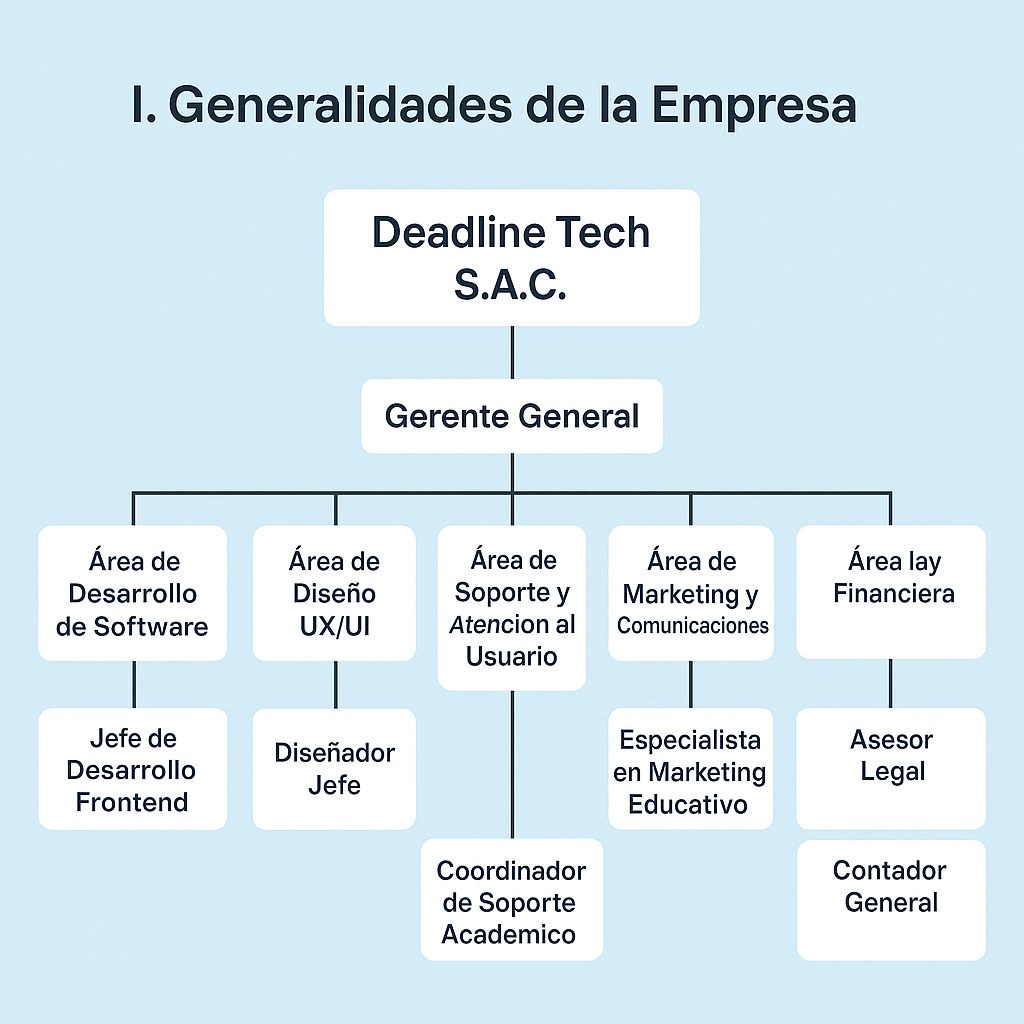
**2. Visión**

Ser la plataforma líder en Latinoamérica en apoyo académico digital, ayudando a estudiantes a gestionar eficazmente sus actividades y reducir el estrés mediante tecnología inteligente.

**3. Misión**

Brindar a los estudiantes una herramienta web intuitiva e inteligente que les permita organizar sus tareas, visualizar compromisos académicos y anticipar colisiones entre fechas importantes, mejorando así su rendimiento académico y bienestar personal.

**4. Organigrama**



**II. Visionamiento de la Empresa  
1. Descripción del Problema**

Los estudiantes universitarios frecuentemente enfrentan múltiples tareas, entregas y evaluaciones con fechas cercanas o incluso superpuestas. Esta sobrecarga y desorganización provoca altos niveles de estrés, baja productividad y, en muchos casos, un desempeño académico deficiente. La falta de herramientas que gestionen de manera proactiva los conflictos entre fechas límite agrava este problema.

**2. Objetivos de Negocios**

* Ofrecer una solución tecnológica que ayude a los estudiantes a gestionar sus actividades académicas.
* Reducir el estrés asociado a la mala planificación y sobrecarga de tareas.
* Posicionar a Deadline Tech S.A.C. como líder en herramientas digitales de apoyo educativo en Latinoamérica.
* Generar una base de usuarios activa y fiel mediante una experiencia útil, intuitiva y diferenciada.

**3. Objetivos de Diseño**

* Desarrollar una plataforma web con una interfaz amigable e intuitiva para estudiantes.
* Integrar un asistente inteligente capaz de registrar tareas mediante lenguaje natural.
* Implementar un calendario visual que muestre claramente eventos académicos.
* Crear un sistema de detección de colisiones que alerte al usuario de conflictos de fechas.
* Proporcionar un dashboard que resuma tareas completadas y pendientes.

**4. Alcance del Proyecto**

Deadline Collision Predictor permitirá a los usuarios:

* Registrar tareas académicas usando lenguaje natural.
* Visualizar sus actividades en un calendario dinámico.
* Recibir alertas automáticas ante posibles colisiones de fechas límite.
* Consultar un panel de control personalizado con indicadores de progreso.  
  Este sistema estará orientado principalmente a estudiantes universitarios y será accesible desde navegadores modernos.

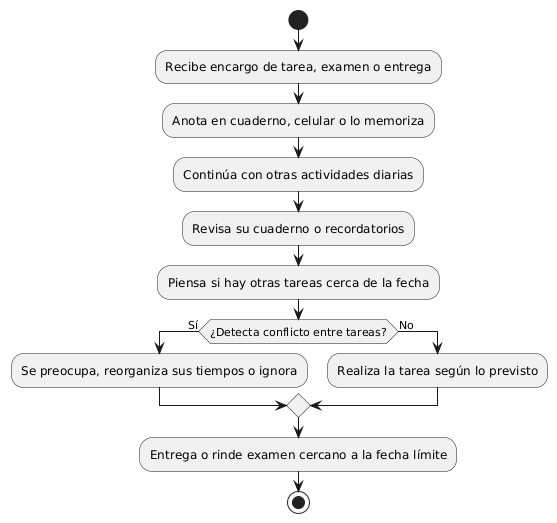
**5. Viabilidad del Sistema**

* Técnica: El sistema es viable utilizando tecnologías web actuales (frameworks front-end/back-end, NLP para lenguaje natural, bases de datos relacionales).
* Económica: La inversión inicial es moderada y puede ser financiada en fases. El modelo freemium permitirá atraer usuarios sin una gran barrera de entrada.
* Operativa: La interfaz será diseñada para que cualquier estudiante pueda usarla sin entrenamiento previo.
* Legal: No maneja información sensible ni viola regulaciones de protección de datos al mantenerse dentro de los límites éticos de privacidad académica.

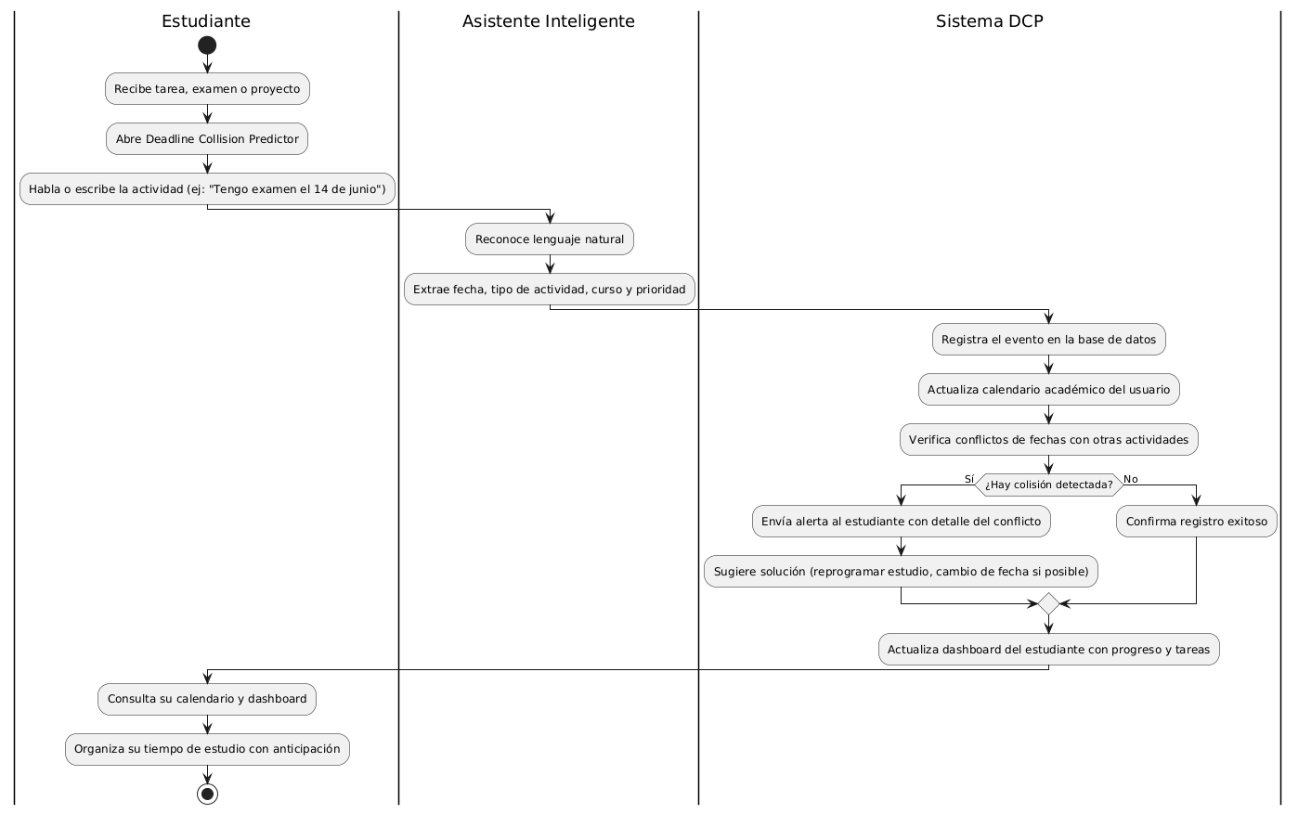
**6. Información obtenida del Levantamiento de Información**

A partir de la observación directa en el entorno universitario, se ha identificado que muchos estudiantes, de diversos ciclos académicos, enfrentan dificultades constantes para gestionar adecuadamente sus tareas, entregas y exámenes. Esta problemática se manifiesta en situaciones de estrés, retrasos y superposición de actividades académicas. Aunque no se han realizado encuestas formales hasta el momento, la experiencia compartida por numerosos estudiantes en distintos cursos y carreras evidencia una necesidad latente de una herramienta que facilite la organización académica. Este contexto motivó el desarrollo de Deadline Collision Predictor, una plataforma web pensada para resolver estos problemas de forma práctica e inteligente.

**III. Análisis de Procesos  
 a) Diagrama del Proceso Actual – Diagrama de actividades**



**b) Diagrama del Proceso Propuesto – Diagrama de actividades inicial**



**IV. Especificación de Requerimientos de Software  
 a) Cuadro de Requerimientos Funcionales Inicial**

| **ID** | **Requerimiento Funcional** | **Prioridad** |
| --- | --- | --- |
| **RF1** | **Registrar nuevas actividades mediante texto o voz** | **Alta** |
| **RF2** | **Almacenar las actividades con fecha y tipo** | **Alta** |
| **RF3** | **Visualizar las tareas en un calendario interactivo** | **Alta** |
| **RF4** | **Detectar colisiones entre fechas de actividades** | **Alta** |
| **RF5** | **Enviar alertas sobre conflictos detectados** | **Media** |
| **RF6** | **Mostrar panel (dashboard) con resumen de actividades** | **Media** |
| **RF7** | **Permitir editar o eliminar actividades** | **Baja** |

**b) Cuadro de Requerimientos No Funcionales**

| **ID** | **Requerimiento No Funcional** | **Prioridad** |
| --- | --- | --- |
| **RNF1** | **El sistema debe ser accesible desde navegadores web modernos** | **Alta** |
| **RNF2** | **La interfaz debe ser responsiva y usable en dispositivos móviles** | **Alta** |
| **RNF3** | **Las respuestas del sistema deben generarse en menos de 2 segundos** | **Media** |
| **RNF4** | **Debe garantizar la privacidad de los datos académicos del usuario** | **Alta** |
| **RNF5** | **El sistema debe estar disponible al menos 99% del tiempo mensual** | **Media** |
| **RNF6** | **Soporte de lenguaje natural en español para entradas de voz/texto** | **Alta** |

**c) Cuadro de Requerimientos Funcionales Final**

| **ID** | **Requerimiento Funcional** | **Prioridad** |
| --- | --- | --- |
| **RF1** | **Registrar actividades mediante entrada de texto o voz** | **Alta** |
| **RF2** | **Guardar actividades con atributos: tipo, fecha, curso, prioridad** | **Alta** |
| **RF3** | **Mostrar calendario interactivo con todas las actividades** | **Alta** |
| **RF4** | **Detectar y mostrar conflictos entre actividades próximas** | **Alta** |
| **RF5** | **Enviar alertas al usuario vía notificaciones visuales** | **Media** |
| **RF6** | **Generar dashboard con tareas pendientes, completadas y alertas** | **Alta** |
| **RF7** | **Permitir edición y eliminación de eventos registrados** | **Media** |
| **RF8** | **Filtrar actividades por tipo, curso, y prioridad** | **Baja** |
| **RF9** | **Reconocer lenguaje natural para facilitar la creación de tareas** | **Alta** |

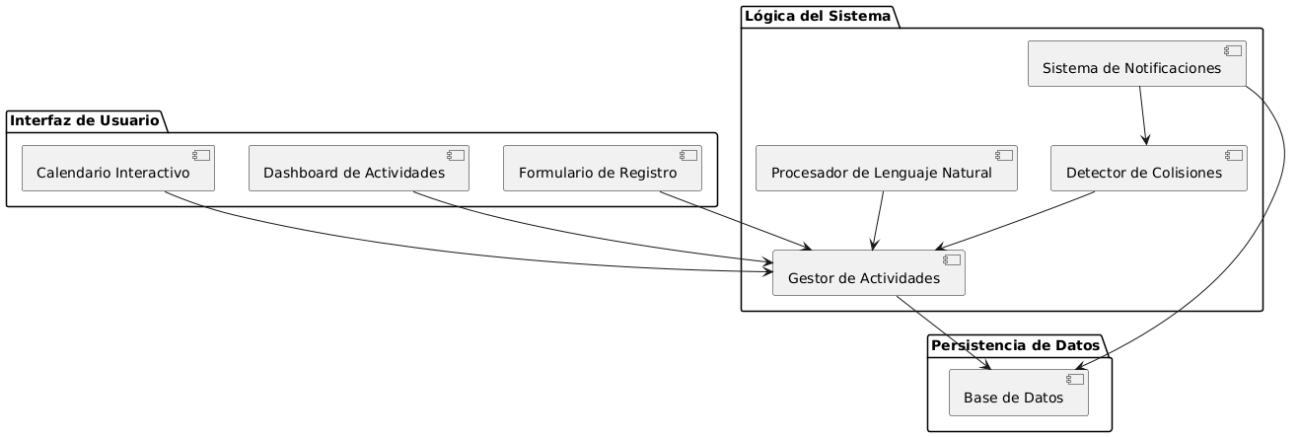
**d) Reglas de Negocio**

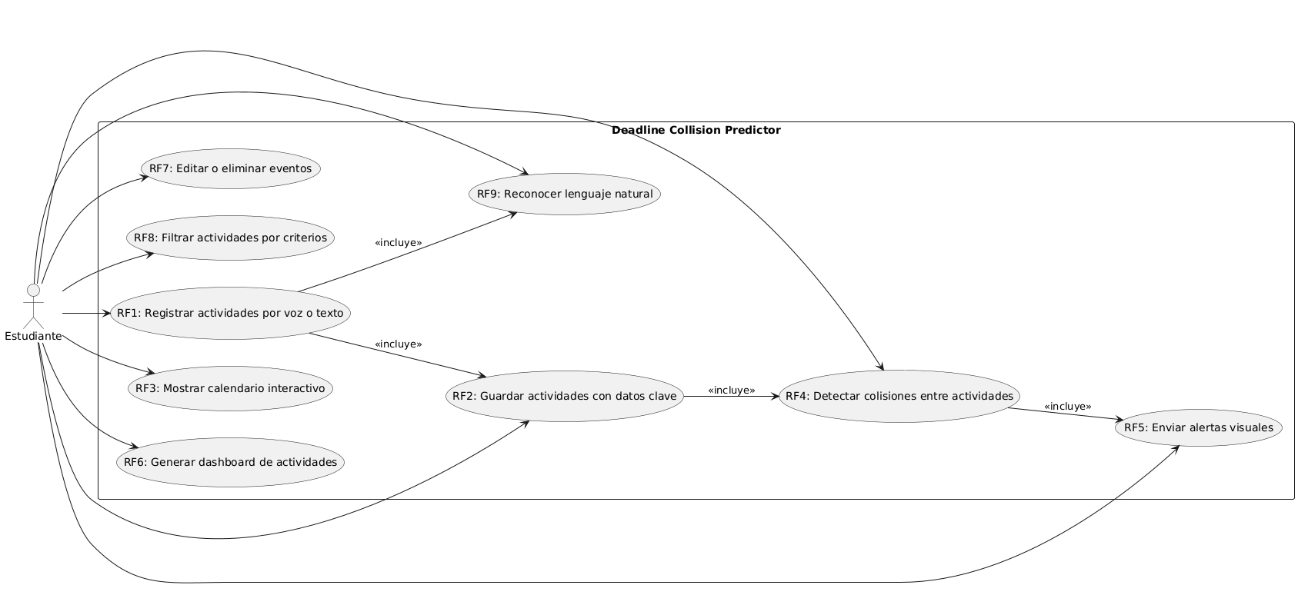
| **ID** | **Regla de Negocio** |
| --- | --- |
| **RN1** | **Toda tarea debe tener al menos una fecha asignada para ser registrada** |
| **RN2** | **No se puede registrar una actividad con fecha pasada sin confirmación del usuario** |
| **RN3** | **Si dos o más actividades coinciden en fecha y hora, se considera colisión** |
| **RN4** | **El sistema debe verificar automáticamente colisiones al registrar una actividad** |
| **RN5** | **Solo el usuario propietario de las actividades puede modificarlas o eliminarlas** |
| **RN6** | **Las actividades se clasifican como "pendientes", "en curso" o "completadas"** |
| **RN7** | **El usuario puede recibir una advertencia si el número de colisiones es alto** |
| **RN8** | **El asistente debe validar la estructura del mensaje antes de crear el evento** |

**V. Fase de Desarrollo  
 1. Perfiles de Usuario**

| **Perfil de Usuario** | **Descripción** | **Funcionalidades Principales** |
| --- | --- | --- |
| **Estudiante** | **Usuario principal del sistema. Estudiantes universitarios de distintos ciclos que desean organizar sus tareas, exámenes y entregas.** | **- Registro de actividades con texto o voz- Visualización en calendario- Alertas- Acceso al dashboard- Edición y eliminación de tareas** |
| **Administrador del Sistema** | **Encargado de la configuración técnica, mantenimiento y control general del sistema.** | **- Gestión de usuarios- Supervisión de registros- Acceso a estadísticas de uso- Control de disponibilidad del sistema** |
| **Asistente Virtual (IA)** | **Módulo del sistema con capacidad de procesamiento de lenguaje natural que interpreta lo que el estudiante dice o escribe.** | **- Interpretar comandos de voz/texto- Registrar actividades- Detectar fechas- Alertar sobre colisiones** |
| **Diseñador/Desarrollador (Equipo interno)** | **Perfil de usuario técnico que trabaja en el desarrollo y evolución del sistema.** | **- Implementación de nuevas funcionalidades- Corrección de errores- Mantenimiento del sistema** |

**2. Modelo Conceptual  
  a) Diagrama de Paquetes**

 **b) Diagrama de Casos de Uso**



**c) Escenarios de Caso de Uso (narrativa)**

**Caso de Uso: RF1 – Registrar actividades mediante entrada de texto o voz**

* Actor Principal: Estudiante
* Precondición: El usuario ha iniciado sesión.
* Flujo principal:
  1. El estudiante accede a la sección de registrar actividad.
  2. Ingresa una tarea usando texto o voz: “Tengo examen de calidad el lunes”.
  3. El sistema interpreta el mensaje con procesamiento de lenguaje natural.
  4. Se identifican datos: tipo = examen, curso = calidad, fecha = próximo lunes.
  5. El sistema solicita confirmación.
  6. El estudiante confirma y la actividad se guarda.

**Caso de Uso: RF2 – Guardar actividades con atributos: tipo, fecha, curso, prioridad**

* **Actor Principal:** Estudiante
* **Precondición:** El estudiante ha iniciado sesión y desea registrar una actividad.
* **Flujo principal:**
  1. El estudiante abre el formulario de registro.
  2. Ingresa manualmente o por voz los siguientes datos:
     + Tipo de actividad: tarea
     + Curso: matemáticas
     + Fecha: 20 de junio
     + Prioridad: alta
  3. El sistema valida los campos.
  4. Se guarda la información en la base de datos.
  5. Se muestra mensaje de confirmación.

**Caso de Uso: RF3 – Mostrar calendario interactivo**

* Actor Principal: Estudiante
* Precondición: El sistema contiene actividades del usuario.
* Flujo principal:
  1. El estudiante abre el calendario desde el dashboard.
  2. El sistema carga las actividades según la fecha actual.
  3. Se muestran actividades por día con etiquetas visuales.
  4. El usuario puede hacer clic sobre un evento para editar o eliminar.

**Caso de Uso: RF4 – Detectar y mostrar conflictos**

* Actor Principal: Sistema
* Precondición: Existen dos o más eventos en la misma fecha y hora aproximada.
* Flujo principal:
  1. El sistema analiza nuevas actividades registradas.
  2. Se detecta superposición con otra tarea del mismo día.
  3. El sistema lanza una alerta visual y sonora al usuario.
  4. Se propone reprogramar alguna de las actividades.

**Caso de Uso: RF5 – Enviar alertas al usuario vía notificaciones visuales**

* **Actor Principal:** Sistema
* **Precondición:** Existen eventos próximos o colisiones.
* **Flujo principal:**
  1. El sistema revisa diariamente las actividades próximas (dentro de 48h).
  2. Si se encuentra una colisión o evento urgente:
     + Muestra un pop-up en la interfaz.
     + Destaca el evento en el dashboard y el calendario.
  3. Se le indica al estudiante tomar acción.
  4. Las alertas se marcan como leídas si el usuario interactúa con ellas.

**Caso de Uso: RF6 – Generar dashboard con tareas pendientes, completadas y alertas**

* **Actor Principal:** Estudiante
* **Precondición:** El usuario tiene eventos registrados.
* **Flujo principal:**
  1. El estudiante accede al dashboard desde el menú principal.
  2. El sistema muestra:
     + Lista de tareas pendientes ordenadas por prioridad.
     + Historial de tareas completadas.
     + Alertas recientes.
  3. El estudiante puede hacer clic en una tarea para actualizar su estado.
  4. El dashboard se actualiza dinámicamente.

**Caso de Uso: RF7 – Permitir edición y eliminación de eventos registrados**

* **Actor Principal:** Estudiante
* **Precondición:** El evento ya existe en el sistema.
* **Flujo principal:**
  1. El estudiante abre el calendario o el dashboard.
  2. Selecciona una actividad previamente registrada.
  3. Elige entre las opciones “Editar” o “Eliminar”.
  4. Si edita, cambia datos como la fecha o el curso.
  5. El sistema actualiza la base de datos.
  6. Si elimina, se pide confirmación y se borra el evento.

**Caso de Uso: RF8 – Filtrar actividades por tipo, curso y prioridad**

* **Actor Principal:** Estudiante
* **Precondición:** El usuario tiene varias actividades registradas.
* **Flujo principal:**
  1. El estudiante accede al calendario o al dashboard.
  2. Abre el panel de filtros.
  3. Selecciona un filtro:
     + Tipo: Examen
     + Curso: Calidad
     + Prioridad: Alta
  4. El sistema muestra solo las actividades que coinciden.
  5. El usuario puede quitar los filtros para ver todo nuevamente.

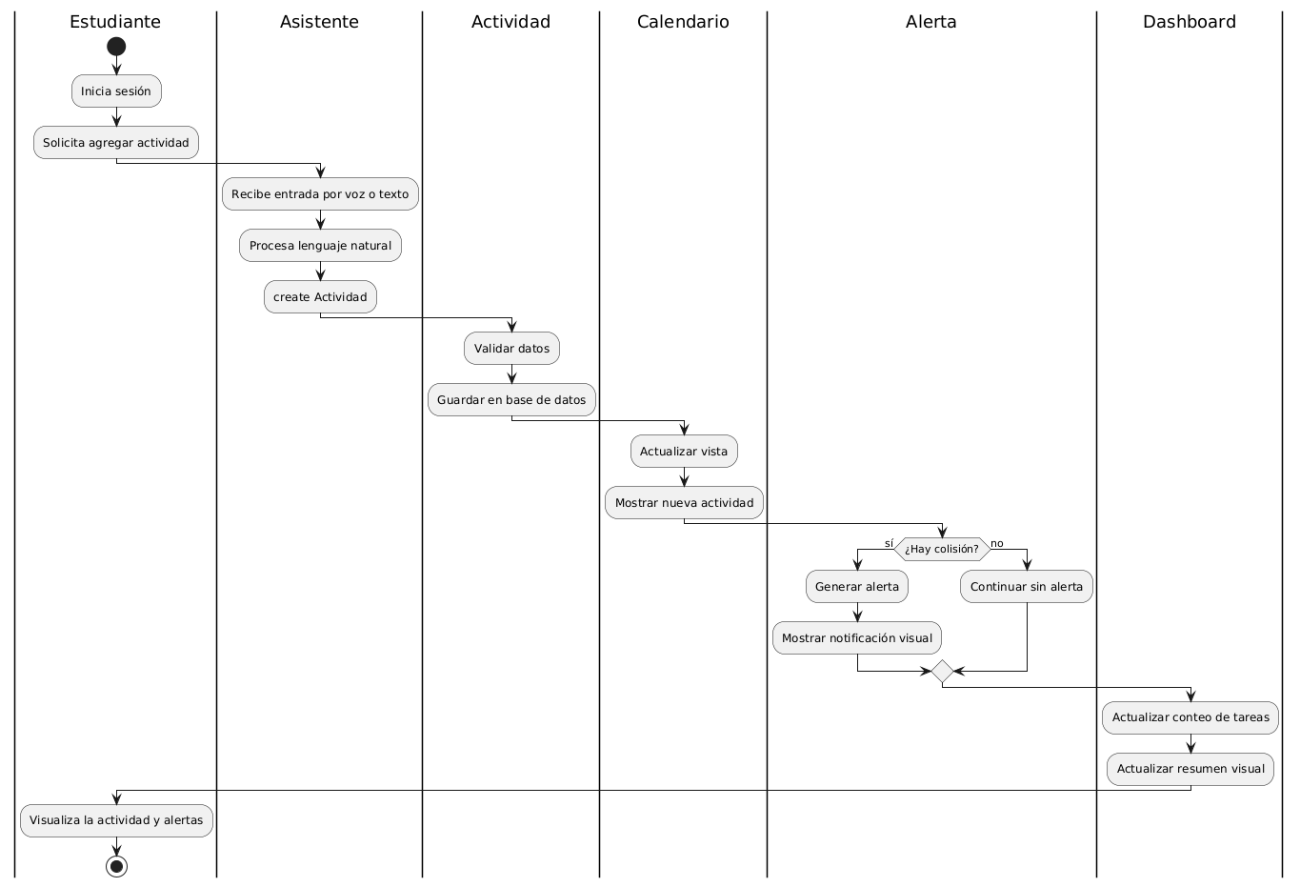
**Caso de Uso: RF9 – Reconocer lenguaje natural para facilitar la creación de tareas**

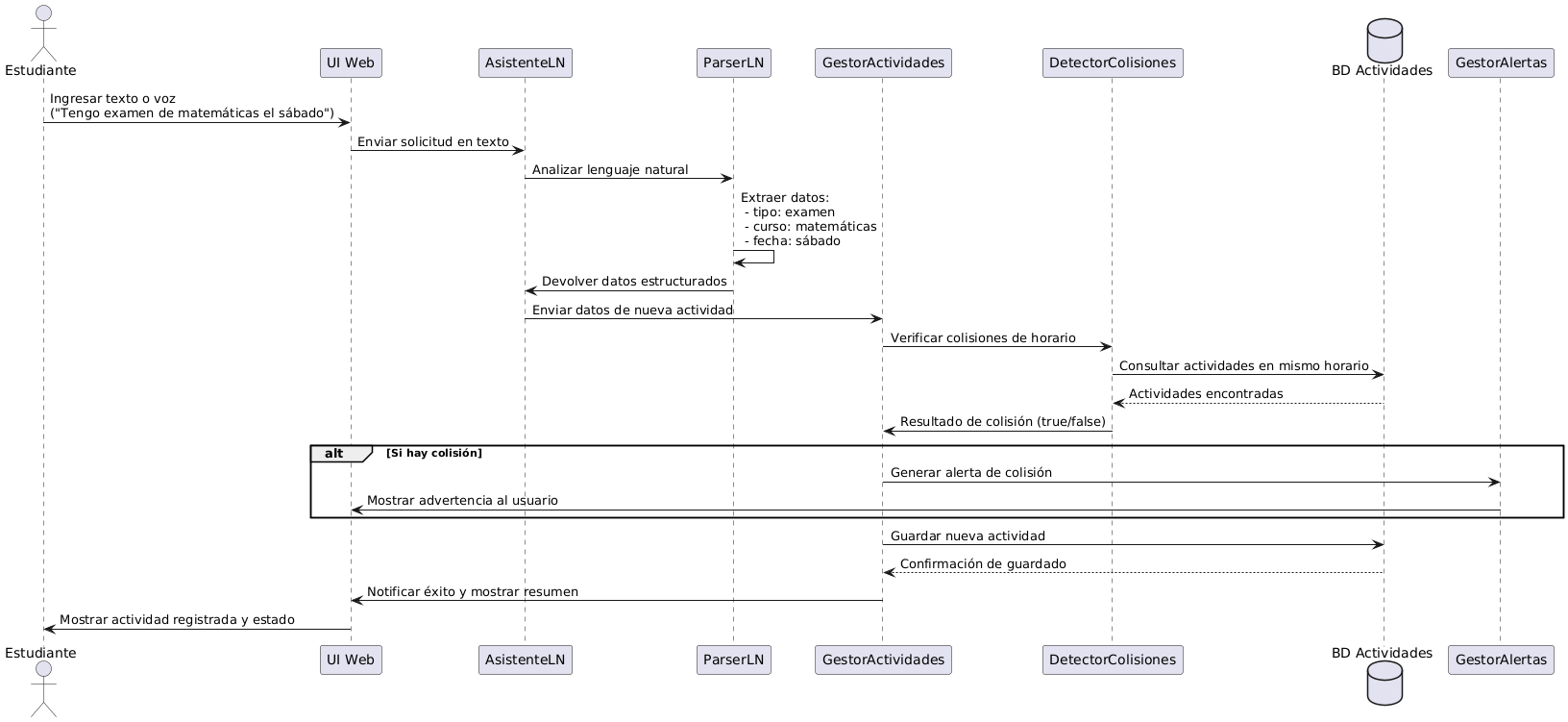
* **Actor Principal:** Estudiante
* **Precondición:** El estudiante tiene habilitado el micrófono o entrada de texto.
* **Flujo principal:**
  1. El estudiante activa la opción de “Registrar por voz”.
  2. Dice: “Tengo exposición de ética el viernes a las 10 a.m.”
  3. El sistema interpreta:
     + Tipo: exposición
     + Curso: ética
     + Fecha: viernes
     + Hora: 10:00 a.m.
  4. Se genera una vista previa con los datos extraídos.
  5. El estudiante confirma y se guarda la tarea.

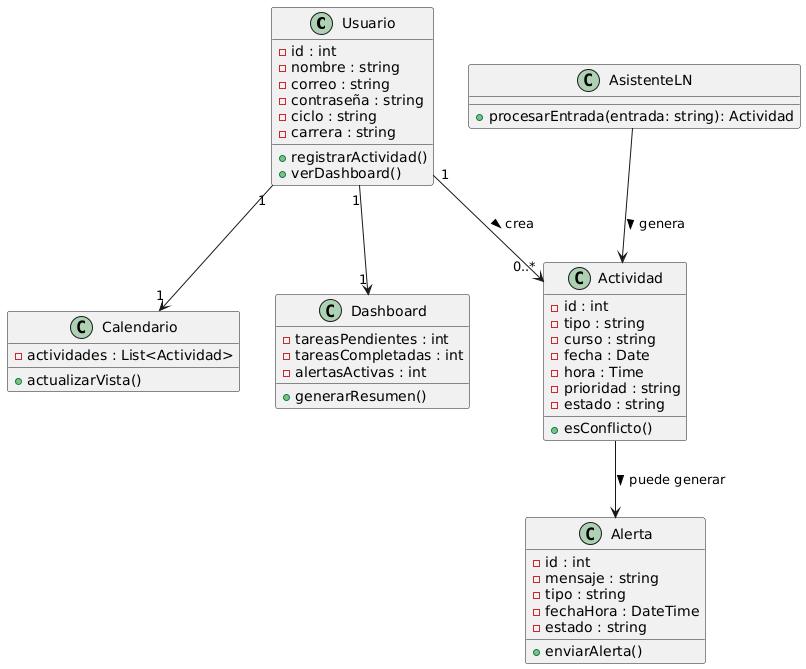
**3. Modelo Lógico  
  a) Análisis de Objetos**

| Objeto | Descripción | Atributos |
| --- | --- | --- |
| Usuario | Representa a los estudiantes registrados. | ID, nombre, correo, contraseña, ciclo, carrera |
| Actividad | Representa una tarea, examen, entrega u otro evento académico. | ID, tipo, curso, fecha, hora, prioridad, estado, ID\_usuario |
| Asistente | Agente que interpreta lenguaje natural y genera actividades automáticamente. | ID, entrada\_voz\_texto, actividad\_generada, fecha\_hora, ID\_usuario |
| Alerta | Notificación enviada al estudiante ante colisiones o eventos cercanos. | ID, mensaje, tipo, fecha\_hora, estado, ID\_usuario |
| Calendario | Representación visual de las actividades registradas. | ID, ID\_usuario, lista\_actividades |
| Dashboard | Resumen visual de actividades completadas, pendientes y alertas. | ID, ID\_usuario, resumen\_pendientes, resumen\_completadas, resumen\_alertas |

**b) Diagrama de Actividades con Objetos**

 **c) Diagrama de Secuencia**

 **d) Diagrama de Clases**



**CONCLUSIONES  
RECOMENDACIONES  
BIBLIOGRAFÍA  
WEBGRAFÍA**