

**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:

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

**Angel Alessandro Chino Rivera (2021069830)**

**Juan Brendon Luna Juarez (2020068762)**

**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 solución web desarrollada por estudiantes universitarios con el propósito de ayudar a organizar las actividades académicas y evitar colisiones de fechas límite. Mediante el uso de tecnologías modernas y procesamiento de lenguaje natural, este sistema facilita la planificación académica en un entorno intuitivo, accesible y gratuito. En este informe se detalla el proceso de desarrollo, análisis técnico, especificaciones funcionales y no funcionales, modelado y documentación del proyecto.

**I. GENERALIDADES DE LA EMPRESA**

**1. Nombre de la Empresa**

**Deadline Collision Labs (Nombre académico para el proyecto)**

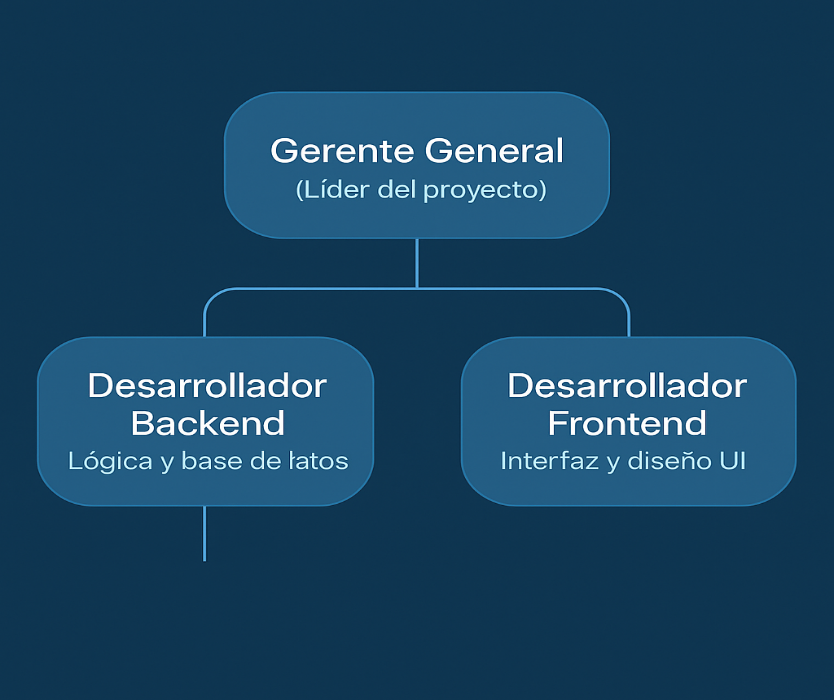
**2. Visión**

Ser una plataforma líder en la gestión inteligente de tareas académicas, reduciendo el estrés y mejorando la planificación de estudiantes universitarios a nivel nacional.

**3. Misión**

Brindar a los estudiantes una herramienta efectiva, intuitiva y automatizada que les permita registrar y organizar sus tareas académicas con facilidad, detectando conflictos de fechas y mejorando su rendimiento.

**4. Organigrama**

****

## ****II. VISIONAMIENTO DE LA EMPRESA****

### ****1. Descripción del Problema****

En el contexto académico actual, los estudiantes universitarios se enfrentan a una sobrecarga de tareas, actividades, trabajos y evaluaciones que muchas veces se superponen en fechas similares. Esta acumulación desorganizada de responsabilidades genera altos niveles de estrés, pérdida de control sobre los plazos y, en muchos casos, disminución del rendimiento académico. A pesar de que existen diversas herramientas de calendario o planificación, la mayoría requiere un registro manual y no ofrecen una detección inteligente de conflictos entre tareas.

Actualmente, no se cuenta con una solución que interprete entradas en lenguaje natural y, además, sea capaz de alertar automáticamente sobre posibles colisiones entre fechas de entrega o eventos académicos. Esta deficiencia genera una necesidad urgente de una plataforma que no solo centralice las actividades, sino que también anticipe los problemas de organización y brinde una experiencia intuitiva a los usuarios.

### ****2. Objetivos de Negocios****

El objetivo principal desde la perspectiva del negocio es brindar una solución tecnológica accesible, gratuita y eficiente que permita a los estudiantes planificar sus actividades de manera automática e inteligente. Con Deadline Collision Predictor se pretende posicionar una herramienta académica de valor, que sea útil, liviana, intuitiva y que logre ser adoptada por usuarios sin necesidad de capacitación compleja.

Desde una visión estratégica, este sistema puede sentar las bases para futuros desarrollos más avanzados que se integren a plataformas educativas institucionales, potenciando su alcance y adaptabilidad. Asimismo, se busca ofrecer un valor agregado no solo a los estudiantes, sino también a los docentes y centros educativos que deseen mejorar la gestión del tiempo y cumplimiento académico de su comunidad.

### ****3. Objetivos de Diseño****

En cuanto al diseño del sistema, se han definido los siguientes objetivos clave:

* Crear una interfaz web responsiva que se adapte a distintos dispositivos, permitiendo su uso desde computadoras, tabletas o smartphones.
* Incorporar un módulo de reconocimiento de lenguaje natural que permita a los usuarios registrar eventos con frases como “tengo examen de álgebra el viernes a las 10”.
* Ofrecer una visualización clara y dinámica de los eventos a través de un calendario interactivo.
* Detectar automáticamente colisiones entre fechas y mostrar advertencias visuales que faciliten la toma de decisiones.
* Mantener un entorno amigable, con un flujo de navegación simple y centrado en la experiencia del usuario.

### ****4. Alcance del Proyecto****

Deadline Collision Predictor ha sido diseñado como una plataforma web modular centrada en el usuario. El sistema contempla diversas funcionalidades, entre las cuales destacan:

* **Registro de eventos manualmente** mediante formularios.
* **Registro de eventos por voz o lenguaje natural** con ayuda de procesamiento del lenguaje natural (NLP).
* **Visualización de tareas y actividades** en un calendario dinámico.
* **Subida de horarios académicos** en formato imagen o PDF.
* **Detección de colisiones o conflictos de fecha**, alertando automáticamente al usuario.
* **Visualización de estadísticas académicas** básicas para el seguimiento del rendimiento y carga de tareas.

Estas funcionalidades están dirigidas principalmente a estudiantes universitarios, aunque su diseño permite adaptaciones a otros entornos académicos o laborales.

### ****5. Viabilidad del Sistema****

El proyecto ha demostrado ser viable desde diferentes enfoques:

* **Técnicamente**, se utilizó un stack tecnológico moderno basado en tecnologías como Node.js, React, PostgreSQL y herramientas de IA como spaCy/Dialogflow. Todo el desarrollo se realizó utilizando plataformas de código abierto.
* **Económicamente**, se optó por soluciones de despliegue gratuitas o de bajo costo como GitHub, Heroku o Vercel, minimizando los gastos.
* **Operativamente**, la plataforma fue pensada para funcionar en cualquier dispositivo con acceso a un navegador moderno, sin necesidad de instalaciones adicionales.

Además, el equipo de desarrollo cuenta con conocimientos suficientes para el mantenimiento del sistema y su escalabilidad futura.

### ****6. Información del Levantamiento de Información****

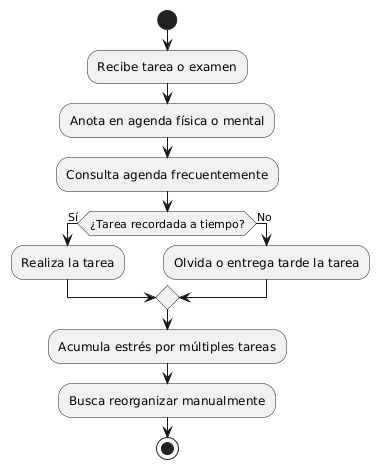
Para definir los requerimientos y características del sistema, se aplicaron las siguientes técnicas:

* **Entrevistas informales** a estudiantes de diferentes facultades para entender sus hábitos de organización académica.
* **Encuestas digitales** con preguntas relacionadas a los problemas comunes al gestionar tareas y fechas límite.
* **Análisis comparativo** de plataformas actuales como Google Calendar, Notion y Trello, identificando sus ventajas y limitaciones.

Los resultados revelaron una necesidad clara: herramientas más automatizadas, con menor carga manual, que entiendan el lenguaje humano y ayuden a evitar el colapso académico por mal manejo del tiempo. Deadline Collision Predictor nace como una respuesta directa a esta necesidad.

## ****III. ANÁLISIS DE PROCESOS****

### a) Diagrama del Proceso Actual – Actividades



### b) Diagrama del Proceso Propuesto – Actividades Iniciales

### 

## ****IV. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE****

### a) Cuadro de Requerimientos Funcionales Inicial

| **ID** | **Requerimiento Funcional** | **Descripción** |
| --- | --- | --- |
| RF1 | Registrar evento por texto | El sistema permite al usuario ingresar eventos manualmente. |
| RF2 | Consultar calendario | El usuario puede visualizar los eventos en un calendario. |
| RF3 | Mostrar advertencia de colisión | El sistema alerta al usuario si hay eventos que se superponen. |
| RF4 | Consultar estadísticas | El sistema ofrece información visual sobre las actividades. |

### b) Cuadro de Requerimientos No Funcionales

| **ID** | **Requerimiento No Funcional** | **Descripción** |
| --- | --- | --- |
| RNF1 | Compatibilidad con navegadores modernos | Debe funcionar correctamente en Chrome, Firefox y Edge. |
| RNF2 | Tiempo de respuesta | Las acciones deben ejecutarse en menos de 2 segundos. |
| RNF3 | Diseño responsivo | La interfaz debe adaptarse a dispositivos móviles y de escritorio. |
| RNF4 | Seguridad básica de datos | El sistema debe manejar contraseñas cifradas y control de sesión seguro. |

### c) Cuadro de Requerimientos Funcionales Final

| **ID** | **Requerimiento Funcional** | **Descripción** |
| --- | --- | --- |
| RF1 | Consultar estadísticas | El usuario visualiza el progreso y carga de trabajo. |
| RF2 | Subir horario (imagen/PDF) | Permite adjuntar horarios académicos como referencia. |
| RF3 | Detectar colisiones | El sistema evalúa si hay eventos solapados. |
| RF4 | Consultar calendarios | Se puede navegar entre fechas y visualizar eventos organizados. |
| RF5 | Registrar evento manual | El usuario ingresa tareas directamente con formularios. |
| RF6 | Registrar evento por IA (NLP) | El sistema reconoce frases como “tengo examen el lunes” y las convierte en eventos. |
| RF7 | Iniciar sesión | Los usuarios pueden acceder con credenciales seguras. |
| RF8 | Registrarse | Se permite la creación de nuevas cuentas de usuario. |

### d) Reglas de Negocio

| **ID** | **Regla de Negocio** |
| --- | --- |
| RN1 | Un usuario no puede registrar un evento si ya hay otro en la misma fecha y hora sin confirmar conflicto. |
| RN2 | Un evento debe contar obligatoriamente con fecha, nombre y tipo. |
| RN3 | Solo se aceptan eventos que estén dentro del calendario académico establecido. |

## ****V. FASE DE DESARROLLO****

### 1. Perfiles de Usuario

| Rol | Descripción |
| --- | --- |
| Estudiante | Es el usuario principal del sistema. Tiene la responsabilidad de registrar tareas, consultar el calendario académico, recibir alertas de colisión y revisar estadísticas. Utiliza tanto la entrada por texto como por voz para registrar eventos. |
| Sistema | Componente automatizado que procesa las entradas del usuario, verifica colisiones entre eventos, genera estadísticas visuales, sugiere cambios en horarios y administra las funciones del calendario. |

### 2. Modelo Conceptual

#### a) Diagrama de Paquetes

#### b) Diagrama de Casos de Uso

#### 

#### c) Escenarios de Caso de Uso (Narrativa)

**Caso de Uso: Registrar evento por IA**

**Actor principal:** Estudiante  
**Precondiciones:** El estudiante debe estar autenticado en el sistema.  
**Flujo principal:**

1. El estudiante accede a la función de "Registrar evento".
2. El estudiante dicta o escribe una frase natural como: “Tengo examen de matemática el lunes a las 8 am”.
3. El sistema interpreta la frase utilizando procesamiento de lenguaje natural (NLP).
4. El evento es analizado y registrado provisionalmente.
5. Se verifica si hay colisiones con otros eventos.
6. Si no hay conflictos, el evento se guarda definitivamente y se notifica al estudiante.
7. Si hay conflicto, se muestra una advertencia y se sugiere una acción.

**Caso de Uso: Consultar calendario**

**Actor principal:** Estudiante  
**Precondiciones:** Debe existir al menos un evento registrado.  
**Flujo principal:**

1. El estudiante accede a la vista de calendario.
2. El sistema muestra una vista mensual/semanal con los eventos registrados.
3. El usuario puede seleccionar un día para ver los detalles.
4. El sistema permite navegar entre meses y ver diferentes cargas de eventos.
5. El estudiante puede eliminar o editar eventos desde esta vista.

**Caso de Uso: Detectar colisiones**

**Actor principal:** Sistema  
**Precondiciones:** Se ha registrado o está por registrarse un nuevo evento.  
**Flujo principal:**

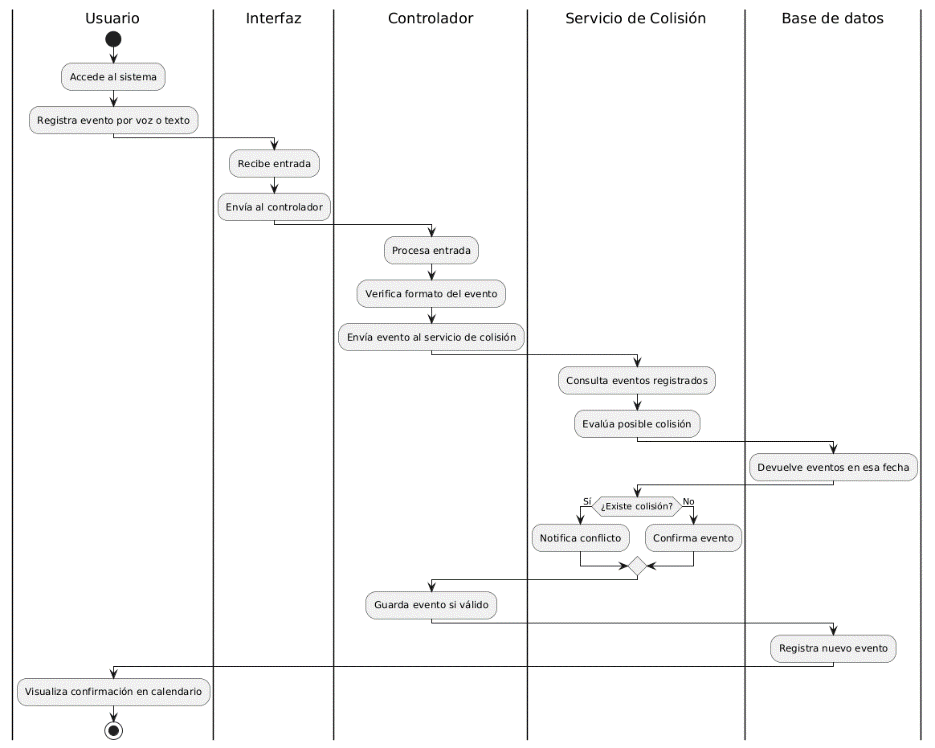
1. El sistema recibe un nuevo evento.
2. Revisa todos los eventos ya existentes en la misma fecha y hora.
3. Si detecta superposición, genera una advertencia visual para el usuario.
4. El sistema sugiere una nueva hora o día como alternativa.
5. El estudiante puede aceptar la sugerencia o confirmar la colisión manualmente.

### 3. Modelo Lógico

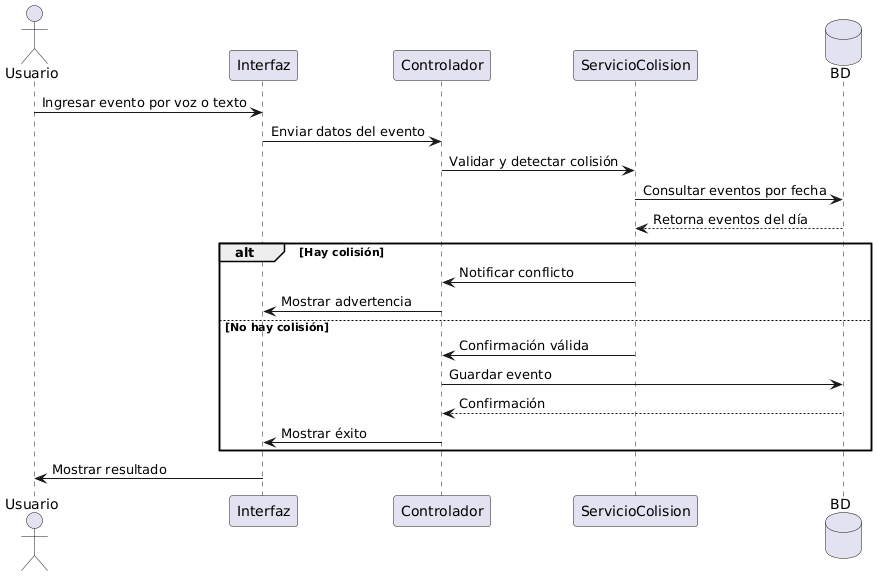
#### a) Análisis de Objetos

El sistema *Deadline Collision Predictor* se compone de cinco clases principales: **Usuario**, **Evento**, **Horario**, **Estadística** y **Sistema**. La clase *Usuario* almacena datos como nombre y correo, y permite acciones como registrarse, iniciar sesión y gestionar eventos. La clase *Evento* contiene información sobre tareas o exámenes, y puede validarse y comprobar colisiones. *Horario* permite subir archivos con cronogramas para analizarlos y convertirlos en eventos. *Estadística* resume la actividad del usuario, como eventos completados. Finalmente, la clase *Sistema* actúa como controlador general, integrando módulos como NLP, calendario y seguridad. Las relaciones entre objetos indican que un usuario puede tener múltiples eventos, horarios y estadísticas, mientras que el sistema coordina todas las clases y procesos.

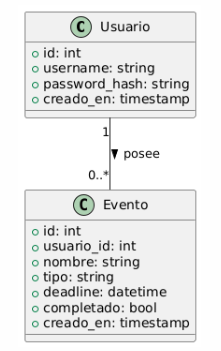
#### b) Diagrama de Actividades con Objetos



#### c) Diagrama de Secuencia



#### d) Diagrama de Clases



## ****CONCLUSIONES****

Deadline Collision Predictor es un proyecto viable y necesario. Su desarrollo permite automatizar la organización académica de los estudiantes. Ha sido concebido con tecnologías escalables y abiertas, y resuelve efectivamente un problema cotidiano. Su implementación futura en universidades podría representar una mejora significativa en la gestión del tiempo académico.

## ****RECOMENDACIONES****

* Escalar el proyecto e integrarlo con plataformas educativas como Moodle.
* Mejorar el reconocimiento de voz en ambientes ruidosos.
* Añadir alertas por correo o notificaciones push.
* Incorporar inteligencia artificial para sugerir tiempos de estudio según hábitos del usuario.