

# **1. Modelo Esencial**

## **1.1 Modelo Ambiental**

El sistema de la plataforma web institucional se encuentra rodeado por un conjunto de actores externos que conforman su ambiente. Dichos actores incluyen a los visitantes, alumnos, docentes, coordinadores, gestores de contenido, administradores y servicios externos como Google Classroom y SIGED. Además, el ambiente comprende los dispositivos y navegadores utilizados por los usuarios, así como la infraestructura técnica de servidores, redes y bases de datos.

Las interacciones entre sistema y ambiente ocurren mediante accesos de usuarios, intercambio de información académica y administrativa, comunicación interna y la integración con plataformas externas. De esta forma, el sistema funciona como un nodo central que organiza, procesa y distribuye datos dentro de la comunidad educativa y hacia el exterior.

## **1.2 Propósito del Sistema**

El propósito de la plataforma es centralizar y modernizar la gestión institucional del liceo, ofreciendo un sitio web dinámico, accesible y seguro. El sistema está diseñado para diferenciar funciones según el rol de cada usuario: los visitantes acceden a información pública, los alumnos gestionan tareas y comunicación con docentes, los profesores administran actividades académicas, los coordinadores supervisan y organizan el flujo educativo, los gestores de contenido mantienen la página actualizada y los administradores controlan la operativa global.

De esta manera, la plataforma actúa como un medio de integración que facilita la comunicación, agiliza la gestión académica y refuerza la transparencia institucional, respondiendo tanto a las necesidades internas de la comunidad educativa como a las demandas externas de información pública.

## **1.3 Lista de Eventos**

Las interacciones entre el sistema y el ambiente pueden representarse en una serie de eventos clave:

- Un visitante accede al sitio web para consultar información pública, eventos o noticias institucionales.

- Un alumno inicia sesión con su correo institucional para visualizar tareas, subir entregas y revisar el calendario de actividades.
- Un docente se conecta al sistema para registrar tareas, calificar entregas, enviar mensajes a sus estudiantes y actualizar el calendario de su curso.
- Un coordinador administra usuarios, revisa horarios académicos, consulta información de alumnos y docentes y comunica novedades a la comunidad.
- Un gestor de contenido actualiza secciones de la página web institucional, edita textos, reemplaza imágenes y publica noticias o testimonios.
- Un administrador gestiona usuarios, contenidos globales, comunicaciones institucionales, configuraciones del sistema y accede a estadísticas de uso.
- Servicios externos como Google Classroom y SIGED se integran con la plataforma para facilitar enlaces y sincronización de información académica.
- El servidor ejecuta procesos automáticos como copias de seguridad, registro de accesos y generación de reportes de estadísticas.

## **1.4 Comportamiento del Sistema**

El sistema se apoya en una base de datos relacional donde se organizan los principales elementos: usuarios con sus roles y permisos, contenidos del sitio web, actividades académicas (tareas, calificaciones, calendarios) y registros de comunicación. Cada acción de los usuarios se traduce en operaciones sobre estas estructuras de datos, garantizando coherencia, seguridad y trazabilidad. Además, el sistema mantiene bitácoras de actividad y copias de respaldo automáticas para asegurar la integridad de la información.

## 2. Informe: Justificación de la Arquitectura Modelo-Vista-Controlador (MVC)

### 2.1 Introducción

En el diseño de aplicaciones web modernas resulta fundamental seleccionar una arquitectura que garantice organización, escalabilidad y mantenimiento del sistema. Para el desarrollo de la plataforma web institucional del liceo, se propone la utilización del **modelo Vista-Controlador (MVC)** como patrón arquitectónico central.

### 2.2 Fundamentación

El patrón MVC divide la aplicación en tres componentes principales que interactúan entre sí, cada uno con responsabilidades claramente definidas:

- **Modelo:** Gestiona la lógica de negocio y la interacción con la base de datos. En este caso, almacenará y procesará información sobre usuarios, tareas, calificaciones, calendarios y contenidos institucionales.
- **Vista:** Es la capa encargada de mostrar la información al usuario de forma amigable e intuitiva. Incluye las interfaces que utilizarán los visitantes, alumnos, docentes, gestores de contenido y administradores.
- **Controlador:** Actúa como intermediario entre el Modelo y la Vista, procesando las solicitudes de los usuarios y devolviendo las respuestas adecuadas.

### 2.3 Ventajas del uso de MVC en el proyecto

1. **Separación de responsabilidades:** Permite mantener el código más ordenado y modular, lo que facilita que diferentes equipos trabajen en paralelo (por ejemplo, programadores en la lógica y diseñadores en la interfaz).
2. **Escalabilidad:** El sistema podrá crecer en funcionalidades sin comprometer la estructura existente, ya que los cambios en un módulo no afectan directamente a los otros.
3. **Reutilización de componentes:** Al separar la lógica de negocio de la interfaz, es posible reutilizar el mismo modelo de datos en diferentes vistas (por ejemplo, web y aplicaciones móviles).

4. **Facilidad de mantenimiento:** Los errores se pueden identificar y corregir más rápidamente al estar cada funcionalidad bien delimitada en su capa correspondiente.
5. **Seguridad mejorada:** El controlador actúa como filtro entre las vistas y el modelo, evitando accesos directos a la base de datos y reduciendo riesgos de inyección de código o accesos indebidos.
6. **Compatibilidad con metodologías ágiles:** El patrón MVC se adapta bien a iteraciones cortas de desarrollo, lo que permite entregar avances funcionales de manera progresiva.

## 2.4 Conclusión

Adoptar la arquitectura **Modelo-Vista-Controlador (MVC)** para la aplicación web del liceo asegura una estructura robusta, flexible y escalable. Este patrón facilita el mantenimiento, potencia la seguridad y permite evolucionar el sistema de acuerdo a las necesidades futuras de la institución. Gracias a la separación clara de roles, se optimiza tanto el desarrollo inicial como la gestión de nuevas funcionalidades, consolidando la plataforma como una herramienta confiable y de largo plazo.