

REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD NACIONAL EXPERIMENTAL  
PARA LAS TELECOMUNICACIONES E INFORMÁTICA  
VICERRECTORADO ACADÉMICO  
PROGRAMA NACIONAL DE FORMACIÓN EN INGENIERÍA EN  
INFORMÁTICA

## **Documentación del Proyecto**

**José Ángel Álamo**

*Autores:*

Leandro Mulato C.I: 31.394.067

Daniel Romero CI: 31307643

Jaime Patiño C.I: 31.604.037

Néstor Girón CI: 30314018

Profesora: Yuly Delgado

Caracas, Julio de 2025



## Introducción

El presente Plan de Desarrollo de Software (PDS) formaliza la ejecución del proyecto de Plataforma Web para la Gestión Administrativa de la U.E.N. "José Ángel Álamo".

- **Propósito:** Migrar los procesos administrativos centrales (Inscripción, Asistencia y Solicitudes) de formatos manuales basados en papel a una solución digital centralizada y accesible vía web.
- **Alcance:** El sistema abarcará los módulos de: Autenticación y Roles, Gestión de Expedientes Estudiantiles, Inscripción en Línea, Carga de Asistencia y Notas (para docentes) y Generación Automática de Documentos (Constancias de Estudio). Se excluyen, por el momento, módulos complejos de contabilidad o nómina.
- **Tecnología Base (Arquitectura):** Se adoptará React para el Frontend y Django (Python) con PostgreSQL para el Backend y la Base de Datos.



## Plan de Desarrollo de Software (PDS)

### Metodología de Desarrollo: Ágil (Scrum Híbrido)

Se ha seleccionado el marco de trabajo Scrum debido a su enfoque en la adaptación continua y la entrega rápida de valor funcional. Dada la naturaleza cambiante del entorno educativo y la necesidad de validar flujos de negocio con el personal del liceo, Scrum es la metodología más adecuada.

- Justificación: Permite la entrega de incrementos de software funcionales cada dos semanas, facilitando la retroalimentación temprana del personal administrativo y minimizando el riesgo de desarrollar una solución que no se ajuste a sus procesos operativos diarios.
- Periodo de Iteración (Sprint): La duración estándar del Sprint será de dos (2) semanas.

### B. Ceremonias (Eventos Scrum)

La eficiencia del equipo se garantizará mediante la adhesión a las siguientes ceremonias:



Evento	Frecuencia	Duración Máx.	Participantes	Propósito y Resultado Esperado
Planificación del Sprint	Inicio de cada Sprint	4 Horas	PM, Devs, QA	Definir el Objetivo del Sprint (Sprint Goal) y seleccionar tareas del <i>Product Backlog</i> para crear el Sprint Backlog.
Daily Scrum (Stand-up)	Diario	15 Minutos	Devs, QA	Sincronización diaria: Revisar el progreso hacia el <i>Sprint Goal</i> , identificar impedimentos y planificar el trabajo de las próximas 24 horas.
Revisión del Sprint	Final de cada Sprint	2 Horas	Equipo y Stakeholders	Demostración formal del Incremento completado al docente/liceo para obtener la aprobación y validar la funcionalidad.
Retrospectiva del Sprint	Final de cada Sprint	1 Hora	Equipo	Evaluación interna del proceso: Analizar qué salió bien, qué no, y generar acciones de mejora para el siguiente Sprint. Liderada por el Project Manager.

Los siguientes artefactos Scrum se utilizarán y se mapearán directamente a los entregables de la documentación solicitada:

- **Product Backlog (Lista de Requisitos Priorizados):** Lista dinámica de todas las funcionalidades, bugs y mejoras. La priorización se basa en el valor de negocio (MoSCoW: Must have, Should have, Could have, Won't have).
- **Sprint Backlog (Lista de Tareas por Iteración):** Es la lista de tareas detalladas que el equipo se compromete a completar dentro del Sprint.
- **Incremento (Software Funcional):** La suma de todos los elementos completados en el Sprint. El resultado final del proyecto (30 de Enero) es la



suma de todos los Incrementos.

Se utilizará un flujo de trabajo GitFlow simplificado para garantizar la estabilidad del código:

1. **Rama main:** Contiene únicamente el código de producción listo para el despliegue final.
2. **Rama development:** Rama principal de integración donde se fusionan todas las funcionalidades completadas.
3. **Ramas feature/:** Creadas por los desarrolladores (Leandro y Jaime) para trabajar en funcionalidades específicas. Una vez completadas, pasan por una Revisión de Código (*Code Review*) y luego se fusionan con development.
4. **Herramienta de Gestión:** GitHub Projects se configura como el tablero Kanban donde las tareas fluyen de To Do -> In Progress -> In Review -> Done.

### Organización del Equipo y Roles

La estructura del equipo se ha reestructurado para asignar responsabilidades clave de forma clara tras la salida de un integrante:

Integrante	Cédula de Identidad	Rol Principal	Responsabilidades Clave
Jaime Patiño	31.604.037	Project Manager	Gestión: Máximo responsable.



Integrante	Cédula de Identidad	Rol Principal	Responsabilidades Clave
		& Frontend Lead	Liderazgo de Scrum, control del <i>Roadmap</i> y enlace. Desarrollo: Liderar la implementación del Frontend (React) y asegurar la experiencia de usuario (UX).
Leandro Mulato	31.394.067	Backend Developer	Desarrollo de la lógica de negocio en Django (Python). Implementación de la API REST, seguridad de <i>endpoints</i> y ejecución de Pruebas Unitarias.
Daniel Romero	31.307.643	DB Architect	Diseño, modelado y mantenimiento del modelo de datos en PostgreSQL. Responsable de la integridad referencial y optimización de las consultas.
Nestor Girón	30.314.018	QA Analyst & Technical Writer	Aseguramiento de Calidad (QA): Diseño y ejecución de Pruebas de Integración y Aceptación (UAT). Documentación: Redacción y mantenimiento de ERS, DES y Manuales de Usuario.

### Planes de Iteraciones (RoadMap)



Sprint (Entrega de Avance)	Foco Principal	Fechas Estimadas	Entregables del Sprint
Sprint 1	Gestión y Fundamentos Técnicos	13 Dic - 26 Dic	Diseño de Interfaz (UI), Sistema de Login/Home Page, Base de Datos Inicial
Sprint 2	Módulo de Inscripción	27 Dic - 09 Ene	Formulario de Inscripción Web, Carga de Documentos, Panel Administrativo de Solicitudes
Sprint 3	Módulos Críticos y Diseño	10 Ene - 23 Ene	Módulo Docente, Lógica de Promedios, Generador de Constancias
Sprint 4	Pruebas Intensivas y Automatizaci ón	24 Ene - 06 Feb	Demo de Flujo Completo, Reporte de Pruebas, Versión Beta Desplegada
Sprint 5	Cierre, Manuales y Versión Final	07 Feb - 10 Feb	Código Fuente Versión 1.0 (Release), Manuales de Usuario

### **Lista de Requisitos Priorizados del Usuario (Product Backlog)**

El *Product Backlog* es la lista única y priorizada de todas las funcionalidades y mejoras deseadas para el sistema, expresadas como Historias de Usuario. La priorización se basa en el valor de negocio para la Intitucion U.E.N. "José Ángel Álamo"



ID	Módulo	Historia de Usuario	Actor	Prioridad	Valor de Negocio
RF-01	Inscripción	Como posible Estudiante, quiero realizar mi solicitud de inscripción en línea, llenando un formulario digital para ahorrar tiempo y papel.	Estudiante	Crítica	Automatización de proceso central.
RF-02	Autenticación	Como Usuario (Admin, Docente o Estudiante), quiero iniciar sesión con mi cédula y clave para acceder solo a mis funciones.	Todos	Crítica	Seguridad y delimitación de roles.
RF-03	Administración	Como Administrativo, quiero validar los documentos y la planilla de inscripción cargada por el estudiante para aprobar o rechazar la matrícula.	Administrativo	Crítica	Control y verificación de expedientes.
RF-04	Académico	Como Docente, quiero cargar la nota final de mis estudiantes por asignatura para mantener el registro académico actualizado.	Docente	Crítica	Registro fundamental para la promoción.
RF-05	Datos	Como Sistema, necesito que todos los datos ingresados pasen por una Validación Automática para asegurar su integridad y formato correcto.	Sistema	Crítica	Calidad y consistencia de datos.

### Requisitos Importantes (Deseables)

ID	Módulo	Historia de Usuario	Actor	Prioridad	Valor de Negocio
RF-06	Solicitudes	Como Estudiante, quiero solicitar y descargar una Constancia de Estudio	Estudiante	Alta	Mejora el servicio al usuario.





ID	Módulo	Historia de Usuario	Actor	Prioridad	Valor de Negocio
		generada automáticamente en PDF para trámites externos.			
RF-07	Trazabilidad	Como Administrativo, quiero consultar el historial de las acciones realizadas sobre una Planilla de Inscripción para auditoría.	Administrativo	Alta	Gobernanza y seguimiento.

### Requisitos Secundarios (Opcionales)

ID	Módulo	Historia de Usuario	Actor	Prioridad	Valor de Negocio
RF-08	Notificaciones	Como Usuario, quiero recibir notificaciones (ej. por correo) cuando el estado de mi solicitud cambie.	Todos	Media	Comunicación fluida.

### Especificaciones del Sistema (DES)

Vista de Casos de Uso a Implementar:



El diagrama de Casos de Uso es la representación visual de los requisitos funcionales. Los tres actores principales interactúan con los módulos clave identificados en el modelado del dominio:

1. Gestión de Inscripción (RF-01, RF-03): Flujo principal que inicia el Estudiante y culmina con la Validación del Administrativo.
2. Gestión Académica (RF-04): Flujo de interacción exclusivo del Docente.
3. Servicios al Estudiante (RF-06): Flujo de solicitud de constancias y documentos.

## Modelo de Casos de Uso



Describe el comportamiento funcional del sistema para todos los actores  
definidos: Estudiante, Personal Administrativo y Docente.

ID Caso de Uso	Nombre del Caso de Uso	Actores Principales	Descripción Breve
CU-00	Autenticar Usuario	Estudiante, Docente, Administrativo	Permite el ingreso seguro al sistema mediante credenciales y redirige al usuario según su rol.
CU-01	Procesar Solicitud de Inscripción	Estudiante, Administrativo	Gestiona el envío del formulario de inscripción por parte del alumno y la validación/aprobación por parte de la administración.
CU-02	Cargar Notas y Asistencia	Docente	Permite al profesor registrar las calificaciones finales y el control de asistencia de su sección asignada.
CU-03	Generar Constancia de Estudio	Estudiante	Genera y descarga automáticamente un documento PDF que certifica la matrícula del alumno.
CU-04	Consultar Historial (Trazabilidad)	Administrativo	Permite auditar las acciones realizadas en el sistema (cambios de estado, registros) para seguridad y control.

#### Flujo Básico (Escenario Principal: Éxito)

1. El Estudiante accede a la sección "Inscripción" y selecciona el Periodo Escolar.
2. El Sistema precarga los datos personales si es un estudiante regular.



3. El Estudiante completa la información faltante (ej. nuevo Representante, documentos).
4. El Sistema ejecuta la Validación Automática (RF-05) de los datos.
5. El Estudiante confirma y envía la solicitud. El estado es "Enviada".
6. El Administrativo recibe la notificación de una nueva Planilla Pendiente.
7. El Administrativo revisa la Planilla y los Documentos Recaudados.
8. El Administrativo selecciona la opción "Aprobar" e indica el GradoSeccion asignado.
9. El Sistema actualiza el estado de la Planilla a "Aprobada" y notifica al Estudiante.

### **Flujos Alternos**

- FA 1.1: Datos Inválidos (Fallo en RF-05): En el paso 4, si la validación falla (ej. campo Cédula vacío), el sistema muestra un mensaje de error específico y regresa al paso 3 sin enviar la solicitud.
- FA 2.1: Rechazo Administrativo: En el paso 8, si los documentos están incompletos, el Administrativo selecciona "Rechazar" e introduce el motivo. El sistema notifica al Estudiante con el motivo del rechazo, permitiendo que el Estudiante corrija y reenvíe (vuelve al paso 1).

### **Vista de Casos de Uso a Implementar**

Para cumplir con los requisitos funcionales priorizados, se han diseñado los siguientes Casos de Uso (CU) que estructuran la interacción entre los actores y el sistema.



- **Actores del Sistema:**

- **Estudiante:** Actor principal que solicita servicios (inscripción, constancias).
- **Personal Administrativo:** Actor encargado de la gestión, validación y supervisión.
- **Docente:** Actor responsable de la carga de datos académicos.

## Descripción Detallada de la Arquitectura

El sistema implementa una Arquitectura de Tres Capas (Three-Tier Architecture) basada en el patrón Modelo-Vista-Controlador (MVC), adaptado a un entorno web moderno con separación de cliente y servidor.

### Capa de Presentación (Frontend)

- **Tecnología:** **React.js** (Librería JavaScript) + **Bootstrap** (Framework CSS).
- **Responsabilidad:** Proporcionar una interfaz de usuario interactiva y *responsive* (adaptable a móviles). Es responsable de capturar los datos del usuario (formularios) y presentarlos de forma amigable.
- **Comunicación:** Se conecta con el servidor mediante peticiones asíncronas HTTP (AJAX) utilizando la librería **Axios/Fetch**.

### Capa de Lógica de Negocio (Backend)

- **Tecnología:** **Python** con el framework **Django**.
- **Responsabilidad:** Contiene toda la lógica del sistema.
  - **API REST:** Expone *endpoints* (URLs) seguros para que el Frontend consuma datos.



- **Seguridad:** Maneja la autenticación de usuarios y la validación de permisos (RBAC).
- **Validación de Datos:** Asegura que la información recibida (ej. cédulas, fechas) cumpla con las reglas de negocio antes de guardarse.

### Capa de Persistencia (Base de Datos)

- **Tecnología:** PostgreSQL.
- **Responsabilidad:** Motor de base de datos relacional robusto, encargado de almacenar permanentemente la información crítica (expedientes, notas) garantizando la integridad referencial y la seguridad de los datos.



## Modelo de Datos y Diccionario de Datos

Este modelo define la estructura de almacenamiento de la información en la base de datos PostgreSQL, derivada estrictamente del Diagrama de Clases del dominio .

### Entidad: Estudiante

Representa al alumno regular de la institución.

- **cedula (PK):** *String*. Identificador único e irrepensible del estudiante.
- **nombres:** *String*. Nombres completos del alumno.
- **apellidos:** *String*. Apellidos completos del alumno.
- **fechaNacimiento:** *Date*. Fecha de nacimiento, utilizada para cálculos de edad.
- **direccion:** *String*. Dirección de habitación actual.
- **telefono:** *String*. Número de contacto personal.

### Entidad: Representante

Datos del padre, madre o tutor legal responsable.

- **cedula (PK):** *String*. Identificador único del representante.
- **nombres:** *String*. Nombres completos.
- **apellidos:** *String*. Apellidos completos.
- **telefono:** *String*. Número telefónico principal para emergencias.
- **parentesco:** *String*. Describe la relación con el estudiante (ej. "Madre", "Abuela").

### Entidad: PlanillaInscripcion

Documento central que formaliza el registro del alumno en un periodo específico.

- **id (PK):** *Int (Auto-incremental)*. Código único de la planilla generado por el sistema.
- **fecha:** *Date*. Fecha en la que se creó la solicitud (por defecto, la fecha actual).
- **estado:** *String*. Indica el estatus del trámite (Valores posibles: "Pendiente", "Aprobada", "Rechazada").



- **Relaciones (FK):**

- Vinculado a la tabla **Estudiante** (quien se inscribe).
- Vinculado a la tabla **PeriodoEscolar** (año académico).
- Vinculado a la tabla **GradoSeccion** (ubicación académica asignada).

### **Entidad: PeriodoEscolar**

Define el ciclo académico activo en el sistema.

- **id (PK):** *Int (Auto-incremental)*. Identificador del periodo.
- **anioInicio:** *Int*. Año calendario de inicio (ej. 2024).
- **anioFin:** *Int*. Año calendario de cierre (ej. 2025).
- **descripcion:** *String*. Texto opcional para detalles adicionales del lapso.

### **Entidad: GradoSeccion**

Unidad organizativa que agrupa a los estudiantes.

- **id (PK):** *Int (Auto-incremental)*. Identificador interno de la sección.
- **grado:** *String*. Año académico a cursar (ej. "1er Año", "4to Año").
- **seccion:** *String*. Letra identificadora del grupo (ej. "A", "B").
- **turno:** *String*. Horario asignado (ej. "Mañana", "Tarde").

### **Entidad: PersonalAdministrativo**

Usuarios con privilegios de gestión dentro del sistema.

- **id (PK):** *Int (Auto-incremental)*. Identificador interno del usuario.
- **usuario:** *String*. Nombre de usuario único para el inicio de sesión.
- **clave:** *String*. Contraseña de acceso (almacenada con encriptación hash).
- **cargo:** *String*. Rol funcional (ej. "Director", "Control de Estudios").





## Modelo de Diseño (Diagramas de Interacción)

### Diagrama de Secuencia: Solicitud de Constancia

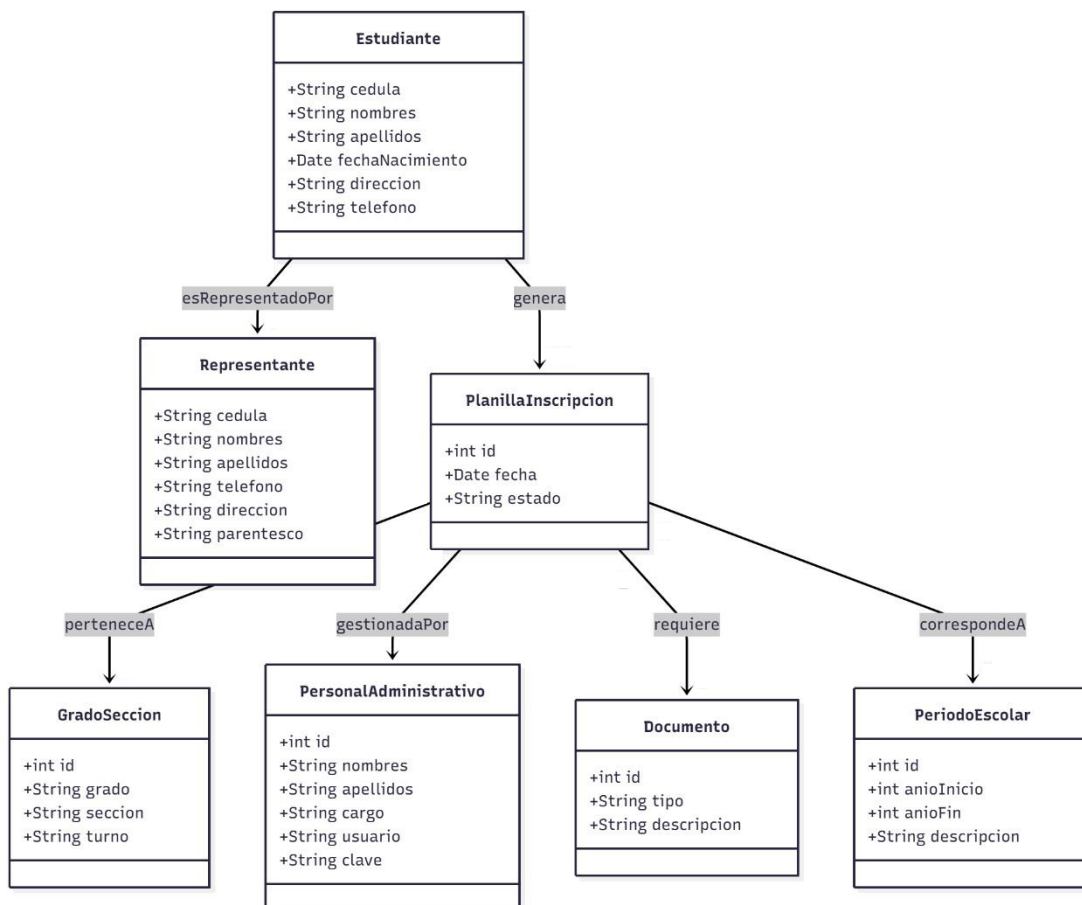
Este diagrama detalla el flujo técnico de la funcionalidad RF-06 (Solicitar Constancia) .

1. Estudiante (Actor): Inicia el evento "Solicitud de constancia" desde la interfaz.
2. Plataforma Web (Frontend): Envía una "Petición HTTP (Datos del estudiante)" a la API.
3. API (Django): Recibe la solicitud y ejecuta "Consulta registros académicos".
4. Base de Datos (PostgreSQL): Procesa la *query* y "Devuelve datos" históricos del alumno.
5. API (Django): Con la información recibida, "Genera el PDF con sus datos de la constancia" y lo envía de vuelta.
6. Plataforma Web (Frontend): Recibe el archivo y habilita la opción "Descargar la constancia en pdf" para el Estudiante.

### Diagrama de Clases (Estructura Estática)

El sistema se estructura bajo un modelo orientado a objetos donde:

- Un Estudiante *es representado por* un Representante.
- Un Estudiante *genera* una PlanillaInscripcion.
- La PlanillaInscripcion es el eje central que:
  - *Pertenece a* un GradoSeccion.
  - *Corresponde a* un PeriodoEscolar.
  - *Requiere* múltiples Documentos.
  - *Es gestionada por* el PersonalAdministrativo.





## Documento de Arquitectura del Sistema

### Diagrama de Paquetes (Estructura del Proyecto)

El sistema se organiza en dos grandes repositorios o directorios principales, desacoplando el cliente del servidor.

#### 1. Backend (Django Project: ue\_jose\_alamo\_backend)

- core/: Configuraciones globales (settings.py), URLs principales y gestión WSGI/ASGI.
- usuarios/: Aplicación encargada de la autenticación, roles (Admin, Docente, Estudiante) y gestión de perfiles.
- gestion\_academica/: Contiene la lógica de Inscripciones, Periodos Escolares y Grados/Secciones.
- reportes/: Módulo para la generación de PDFs (Constancias) y exportación de datos.

#### 2. Frontend (React Project: ue\_jose\_alamo\_frontend)

- src/components/: Elementos reutilizables (Botones, Inputs, Navbar, Footer).
- src/pages/: Vistas completas (Login, DashboardEstudiante, FormularioInscripcion).
- src/services/: Configuración de Axios para la comunicación con la API.
- src/context/: Gestión del estado global (Autenticación del usuario).



## Modelo de Despliegue (Infraestructura)

La arquitectura física para el despliegue en producción consta de los siguientes nodos:

1. Nodo Cliente: Navegador Web (Chrome/Firefox) en PC o Móvil del usuario (Estudiante/Administrativo) ejecutando el paquete de React.
2. Servidor Web: Servidor capaz de servir archivos estáticos del frontend y actuar como *Proxy Inverso* hacia el backend.
3. Servidor de Aplicaciones: Entorno Python ejecutando Gunicorn para servir el proyecto Django.
4. Servidor de Datos: Instancia de PostgreSQL alojando la base de datos relacional.

## Manual de Operaciones

Este documento está dirigido al personal técnico encargado del mantenimiento del sistema.

## Procedimiento de Respaldo (Backup)

Para prevenir la pérdida de datos críticos (Notas e Inscripciones), se debe ejecutar el siguiente script semanalmente en el servidor de base de datos:

- Comando: `pg_dump -U postgres -h localhost ue_jose_alamo_db > backup_fecha.sql`
- Almacenamiento: El archivo .sql resultante debe subirse a una nube segura o disco externo.

## Gestión de Logs y Errores

El sistema registra eventos críticos en archivos de texto para auditoría:

- Access Log: Registra quién ingresa al sistema. Ubicación: `/var/log/django/access.log`.
- Error Log: Registra fallos del servidor (ej. Error 500). Ubicación: `/var/log/django/errors.log`.



## Plan de Capacitación y Adopción

Dado que el diagnóstico del liceo evidenció una "ineficiencia de los métodos manuales" y una necesidad de modernización, el plan de capacitación se enfoca no solo en el uso de la herramienta, sino en la gestión del cambio cultural hacia lo digital.

### Estrategia de Formación

- Modalidad: Híbrida (Talleres presenciales en el laboratorio de informática del liceo + Video-tutoriales de soporte).
- Cronograma: Semana del 01 al 05 de Febrero (Previo al lanzamiento final).

### Contenido Programático de las Sesiones

Sesión	Público Objetivo	Duración	Temario / Contenido
S01: Inducción General	Todo el Personal	1 Hora	<ol style="list-style-type: none"> <li>1. Beneficios del sistema (Cero papel, rapidez).</li> <li>2. Normas de seguridad (Contraseñas seguras).</li> <li>3. Demostración general del flujo.</li> </ol>
S02: Gestión Administrativa	Personal Administrativo Secretaría/Dirección	4 Horas Práctica	<ol style="list-style-type: none"> <li>1. Creación de Periodos Escolares.</li> <li>2. Validación de Inscripciones: Cómo revisar PDFs y aprobar cupos.</li> <li>1. Generación de reportes y auditoría.</li> </ol>
S03: Gestión Académica	Docentes	2 Horas	<ol style="list-style-type: none"> <li>1. Acceso al módulo docente.</li> <li>2. Carga de calificaciones por corte.</li> <li>3. Registro de asistencia diaria/semanal.</li> </ol>
S04: Autogestión Estudiantil	Representantes y Voceros Estudiantiles	1 Hora (Charla)	<ol style="list-style-type: none"> <li>1. Cómo registrarse y recuperar clave.</li> <li>2. Pasos para llenar la planilla de inscripción.</li> <li>3. Descarga de constancias.</li> </ol>



## Manual de Usuarios

Esta sección sirve como guía de referencia rápida para las tareas más comunes.

### Módulo 1: Acceso al Sistema

1. Ingrese a la dirección web del liceo.
2. Escriba su Cédula de Identidad y Contraseña.
3. Haga clic en el botón **"Ingresar"**.
  - *Nota:* Si es su primera vez, debe hacer clic en "Registrarse" y validar su correo.

### Módulo 2: Inscripción (Para Estudiantes/Representantes)

El proceso clave para formalizar la matrícula .

1. En el menú principal, seleccione **"Inscripciones"**.
2. Verifique sus datos personales pre-cargados. Si hay errores, corríjanos.
3. Carga de Documentos: Haga clic en "Subir Archivo" para adjuntar la cédula escaneada y las notas certificadas (Formato PDF).
4. Seleccione el botón **"Enviar Solicitud"**.
5. El sistema mostrará un mensaje de éxito: *"Su solicitud ha sido enviada y está en espera de revisión administrativa"*.

*(Espacio reservado para: [Insertar Captura de Pantalla del Formulario de Inscripción])*

### Módulo 3: Validación de Solicitudes (Para Administrativos)

- Ingrese al panel de control y busque la sección **"Solicitudes Pendientes"**.
- Haga clic en el ícono de "Ojo" (Ver detalles) de un estudiante.
- Revise que los documentos PDF sean legibles.



**Acción:**

1. **Aprobar:** Asigne el Grado y Sección correspondiente y guarde.
2. **Rechazar:** Escriba el motivo del rechazo (ej. "Documento borroso") para notificar al usuario.

#### **Módulo 4: Solicitud de Constancias (Para Estudiantes)**

- Vaya al menú "**Mis Trámites**".
- Seleccione "**Constancia de Estudio**".
- El sistema verificará su inscripción activa.
- Haga clic en "**Descargar PDF**". El documento se guardará automáticamente en su dispositivo.