Un dibujo de una cara feliz

El contenido generado por IA puede ser incorrecto.MANUAL DE USUARIO – [Eduvia-Casa Estudio Chaminade]

Texto, Icono

El contenido generado por IA puede ser incorrecto.

Versión: 1.0

Fecha: [15/06/2025]

Elaborado por: Isaac Josue Sotelo Bravo y Maximiliano Gomez Cisterna

Destinatario: Analista Programador y equipo de Casa Estudio Chaminade

Contenido

[1. Introducción 3](#_Toc201212324)

[2. Objetivo del sistema 3](#_Toc201212325)

[3. Alcances y limitaciones 3](#_Toc201212326)

[4. Requisitos del sistema 3](#_Toc201212327)

[4.1 Hardware 3](#_Toc201212328)

[4.2 Software 3](#_Toc201212329)

[4.3 Usuarios del sistema 3](#_Toc201212330)

[5. Instalación del sistema 3](#_Toc201212331)

[6. Arquitectura del sistema 3](#_Toc201212332)

[7. Descripción general de módulos 3](#_Toc201212333)

[8. Manual de uso (Frontend y Backend) 3](#_Toc201212334)

[9. Flujos de procesos principales 3](#_Toc201212335)

[10. Mantenimiento del sistema 4](#_Toc201212336)

[11. Respaldo y recuperación 4](#_Toc201212337)

[12. Seguridad del sistema 4](#_Toc201212338)

[13. Buenas prácticas recomendadas 4](#_Toc201212339)

[14. Glosario de términos 4](#_Toc201212340)

[15. Anexos 4](#_Toc201212341)

## Introducción

El presente manual de usuario describe el funcionamiento del sistema web desarrollado para Casa Estudio Chaminade, una plataforma diseñada específicamente para apoyar la gestión académica de la institución.

Este sistema está orientado principalmente a los profesores, quienes podrán registrar de forma digital y segura la asistencia y las notas de sus alumnos. Además, contempla la existencia de superusuarios con permisos avanzados, encargados de la administración general del sistema. Estos usuarios tienen la capacidad de visualizar, modificar y gestionar integralmente los registros académicos de todos los estudiantes, asegurando el correcto funcionamiento y control del entorno educativo digital.

El manual está dirigido a analistas programadores y contiene toda la información necesaria para comprender la estructura, el uso y el mantenimiento de la plataforma.

## Objetivo del Sistema

El propósito principal del sistema es digitalizar y centralizar la gestión académica de Casa Estudio Chaminade, facilitando el trabajo docente y administrativo. A través de esta plataforma, los profesores pueden registrar de manera eficiente la asistencia diaria y las calificaciones de los alumnos, asegurando una trazabilidad precisa y actualizada del rendimiento estudiantil.

Además, el sistema permite a los superusuarios supervisar y gestionar toda la información académica, manteniendo un control integral y en tiempo real sobre los datos registrados. Esto contribuye a una administración educativa más transparente, ordenada y accesible para los responsables del seguimiento académico.

## Alcances y limitaciones

**Alcances actuales del Sistema**

* El sistema web desarrollado para Casa Estudio Chaminade permite actualmente:
* El registro de asistencia por parte de los profesores.
* La gestión y carga de notas académicas de los estudiantes.
* El acceso a la plataforma mediante diferentes roles, diferenciando claramente entre profesores y superusuarios.
* La gestión centralizada de datos académicos, permitiendo un control administrativo completo por parte de los superusuarios.

**Limitaciones actuales**

* El sistema está enfocado exclusivamente en la gestión de asistencia y notas, y no contempla por ahora otras funciones complementarias como informes disciplinarios, comunicaciones internas o carga de contenidos pedagógicos.
* No está disponible aún el acceso para estudiantes ni apoderados, por lo tanto, los alumnos no pueden consultar sus calificaciones de manera directa a través del sistema.
* El uso está limitado a profesores y superusuarios, sin acceso aún para otros funcionarios del establecimiento.

**Proyección de expansion**

* Aunque en esta primera versión el sistema está limitado a las funciones mencionadas, se proyecta su expansión futura para incluir:
* Acceso para otros funcionarios como el equipo PIE (Programa de Integración Escolar), área de psicología, entre otros equipos de apoyo.
* Habilitación de una vista para estudiantes, donde puedan revisar sus calificaciones, asistencias y otros registros relevantes de forma digital y segura.

## 4. Requisitos del sistema

|  |  |
| --- | --- |
| Categoría | Requisito |
| Python | Compatible desde la versión **3.8** hasta **3.12** |
| Django | Versión recomendada: **4.2.x** |
| pip | Última versión disponible (recomendado) |
| Base de datos | - Por defecto: **SQLite**- Compatibles: **PostgreSQL**, **MySQL** (*requiere configuración adicional*) Actualmente: **PostgreSQL** |
| Sistemas operativos | - **Windows**- **Linux** (Ubuntu, Arch, etc.)- **macOS** |
| Soporte técnico | Entorno recomendado: VS Code, Git, navegador actualizado (Chrome o Firefox) |

## 4.1 Hardware

***Para desarrollo / entorno local***

* **CPU:**  
  2 núcleos (Intel i3 / AMD Ryzen 3 o superior)  
  *Recomendado:* Procesador superior a i3/Ryzen 3
* **RAM:**  
  Mínimo: 2 GB  
  Recomendado: 4 GB o más
* **Almacenamiento:**  
  Al menos 1 GB libre  
  *Nota:* Puede aumentar si se manejan archivos multimedia
* **Sistema operativo:**  
  Compatible con Windows, Linux y macOS  
  Recomendado: Usar la última versión estable disponible

***Requisitos para Usuarios Finales (Navegador Web)***

**Dispositivo**

* PC de escritorio, notebook, Chromebook, tablet o smartphone moderno

**CPU**

* Procesador dual-core de los últimos 10 años o superior  
  (Ejemplos: Intel i3, AMD A4, ARM Cortex-A7)

**RAM**

* Mínimo 2 GB
* Recomendado 4 GB para mejor experiencia

**Almacenamiento**

* No requiere instalación local, solo espacio suficiente para el sistema operativo y navegador

**Pantalla**

* Resolución mínima: 1024 x 600 píxeles
* Recomendado: 1366 x 768 píxeles o superior

**Navegadores compatibles**

* Google Chrome (última versión)
* Mozilla Firefox (última versión)
* Microsoft Edge (última versión)
* Safari (en Mac y iOS)
* Navegadores móviles modernos en Android e iOS

**Conexión a Internet**

* Mínimo 1 Mbps por usuario
* Recomendado 5 Mbps para mejor fluidez y experiencia

## 4.3 Usuarios del sistema

**Profesores:**

Registran asistencia y notas de sus alumnos; acceso limitado a sus cursos.

**Superusuarios (Administradores):**

Gestionan completamente notas, asistencia, usuarios y configuraciones del sistema.

**Usuarios futuros:**

Equipos de apoyo (PIE, psicología) y estudiantes, con acceso planificado para futuras versiones.

## 5. Instalación del sistema

* entra a la carpeta del proyecto.
* Crea y activa un entorno virtual (venv).
* Instala las dependencias con pip install -r requirements.txt.
* Ejecuta las migraciones con python manage.py migrate.
* Crea un superusuario con python manage.py createsuperuser.
* Inicia el servidor con python manage.py runserver.

## 6. Arquitectura del sistema

El sistema sigue una arquitectura cliente-servidor:

**Frontend:** Interfaz web accesible desde navegadores modernos.

**Backend:** Servidor con Django que maneja la lógica y datos.

**Base de datos:** SQLite para desarrollo; PostgreSQL o MySQL en producción.

**Comunicación:** A través de solicitudes HTTP/HTTPS.

Seguridad: Control de acceso por roles y autenticación segura.

Esta estructura permite fácil mantenimiento y futuras expansiones.

## 7. Descripción general de módulos

**Usuarios:**

* Superusuarios pueden crear, editar y eliminar usuarios.
* Profesores solo gestionan usuarios dentro de su grupo de estudiantes.

**Asistencia:**

* Profesores pueden crear, editar y eliminar registros de asistencia solo para sus estudiantes.
* Superusuarios tienen acceso completo.

**Notas:**

* Profesores gestionan notas solo de sus alumnos.
* Superusuarios gestionan todas las notas.

**Años Lectivos:**

* Solo superusuarios pueden crear y administrar años lectivos para organizar el trabajo académico.

**Exportación:**

* Superusuarios pueden exportar datos e informes para análisis o respaldos.

**Autenticación:**

* Control de acceso, inicio de sesión seguro y gestión de roles.

## 8. Manual de uso (Frontend y Backend)

**Frontend (Interfaz de usuario)**

Acceso mediante navegador web compatible.

**Profesores pueden:**

Registrar asistencia y notas de sus estudiantes.

Editar o eliminar registros propios.

**Superusuarios pueden:**

Gestionar usuarios, años lectivos, asistencia y notas de todos los alumnos.

Exportar reportes y realizar configuraciones avanzadas.

**Backend (Panel Administrativo)**

Acceso restringido para superusuarios.

**Funciones disponibles:**

Administración completa de usuarios y roles.

Configuración del sistema y parámetros generales.

Gestión de años lectivos para organizar el ciclo académico.

Control total sobre registros académicos y asistencias.

## 9. Flujos de procesos principals

**Inicio de sesión**

El usuario ingresa RUT y contraseña.

El sistema valida:

* Si es correcto → entra al panel correspondiente (profesor o superusuario).
* Si es incorrecto → muestra error.

**Registro de asistencia (profesor)**

Accede al panel.

* Selecciona curso y fecha.
* Marca asistencia por alumno.
* Guarda los datos.

**Ingreso de notas (profesor)**

* Selecciona el curso y asignatura.
* Ingresa o edita notas por estudiante.
* Guarda los cambios.

Gestión administrativa (superusuario)

Accede al panel administrativo.

Puede:

* Crear usuarios y años lectivos.
* Editar asistencia y notas de cualquier curso.
* Exportar datos si es necesario.

## 10. Mantenimiento del sistema

**Actualizaciones**

Dependencias:

* pip list --outdated
* pip install --upgrade [paquete]

Django y Python:

* Verificar compatibilidad antes de actualizar versiones mayores.

**Limpieza**

Base de datos:

* Eliminar registros antiguos o pruebas (opcional según política del establecimiento).

## 11. Respaldo y recuperación

Recomendación para desarrolladores

Es altamente recomendable mantener el código fuente del sistema en un repositorio privado (por ejemplo, en GitHub), especialmente si se realizan personalizaciones o mejoras.

Esto permite:

* Control de versiones y cambios.
* Colaboración entre desarrolladores.
* Rápida restauración en caso de fallos.

**En caso de falla o pérdida total**

Si no es posible recuperar el sistema desde respaldos o repositorios, se recomienda **contactar directamente con los desarrolladores del sistema** para asistencia técnica y recuperación especializada.

## 12. Seguridad del Sistema

**Autenticación de usuarios**

* Acceso mediante **RUT** y **contraseña segura**.
* Solo usuarios registrados pueden ingresar.
* Contraseñas encriptadas mediante el sistema de autenticación de Django.

**Control de acceso por roles**

* **Superusuarios**: acceso completo a todas las funciones del sistema.
* **Profesores**: acceso restringido solo a sus propios estudiantes y datos asignados.
* Los permisos están gestionados en el backend según el rol asignado.

**Protección contra amenazas comunes**

* Protección integrada contra:
  + **Inyecciones SQL** (ORM de Django).
  + **Cross-Site Request Forgery (CSRF)**.
  + **Cross-Site Scripting (XSS)**.
* Sesiones seguras y expiración automática según configuración.

**Registro y seguimiento**

* Acciones importantes pueden ser registradas en logs del sistema.
* Revisión manual recomendada tras cambios o incidentes.

**Recomendaciones adicionales**

* Cambiar contraseñas periódicamente.
* Usar HTTPS en entornos productivos.
* Mantener actualizado Django y dependencias para evitar vulnerabilidades.

## 13. Buenas prácticas recomendadas

**Uso responsable del sistema**

* Iniciar sesión solo con credenciales propias.
* Cerrar sesión al terminar el trabajo.
* No compartir contraseñas ni acceder desde dispositivos no seguros.

**Respaldo y control de versiones**

* **Ya se ha mencionado previamente**, pero se refuerza que es **altamente recomendable** usar un **repositorio privado en GitHub** u otro sistema de control de versiones.
  + Permite llevar control de cambios.
  + Facilita la colaboración entre desarrolladores.
  + Es útil para recuperación ante errores.

**Mantenimiento regular**

* Revisar periódicamente logs del sistema.
* Mantener dependencias y Django actualizados.
* Realizar respaldos frecuentes y almacenarlos en lugares seguros.

**Seguridad y configuración**

* Cambiar contraseñas periódicamente. (Opcional)

**Comunicación**

* En caso de errores graves, pérdida de datos o problemas fuera del alcance del usuario, **contactar directamente al equipo de desarrollo** para recibir soporte.

## 14. Glosario de términos

Entendido, aquí tienes un glosario **solo con términos más técnicos o específicos que podrían ser menos conocidos**, basado en tu manual:

* **Superusuario:** Usuario con permisos administrativos totales para gestionar el sistema.
* **Repositorio (GitHub):** Plataforma para almacenar y controlar versiones del código fuente, facilitando la colaboración y recuperación.
* **Entorno virtual (venv):** Entorno aislado para instalar y gestionar dependencias de Python sin afectar el sistema operativo.
* **Backup (respaldo):** Copia de seguridad de datos y archivos para prevenir pérdida de información.
* **Restauración:** Proceso de devolver el sistema a un estado previo utilizando respaldos.
* **Logs:** Archivos que registran eventos y actividades del sistema para monitoreo y diagnóstico.
* **Frontend:** Interfaz de usuario visible en el navegador.
* **Backend:** Lógica y procesamiento del sistema ejecutados en el servidor.