

Universidad San Carlos de Guatemala  
Facultad de ingeniería.  
Ingeniería en ciencias y sistemas  
Segundo semestre  
Escuela de vacaciones de diciembre



## Título del Proyecto:

# StudentGuard

**PONDERACIÓN: 35**

**Horas Aproximadas: 30**

# Resumen Ejecutivo

La deserción académica en instituciones educativas ha presentado un incremento constante y representa un riesgo significativo para docentes, autoridades universitarias y departamentos de bienestar estudiantil. Entre las causas se han identificado múltiples factores que incluyen bajo rendimiento, falta de asistencia, poca participación en clase y dificultades en el proceso de adaptación estudiantil. La detección temprana de estudiantes en riesgo continúa siendo limitada debido a la falta de herramientas que permitan anticipar comportamientos críticos antes de que estos se materialicen.

Ante esta problemática has decidido crear una solución tecnológica innovadora: **StudentGuard** para apoyar a las instituciones educativas en la identificación temprana de estudiantes con probabilidad de riesgo académico.

## Objetivos del Proyecto

### Objetivo General

Desarrollar una aplicación basada en aprendizaje supervisado que permita al estudiante aplicar los conocimientos adquiridos en el curso para entrenar, evaluar y ajustar modelos de predicción.

### Objetivos Específicos

- Implementar un modelo de aprendizaje supervisado utilizando los conocimientos adquiridos sobre recopilación, manipulación y preprocesamiento de datos.
- Medir, visualizar y ajustar hiperparámetros integrando los conceptos del proceso de entrenamiento y evaluación de modelos aprendidos en clase.
- Aplicar métodos de análisis y validación del desempeño del modelo para justificar la capacidad predictiva del sistema y reforzar los principios de evaluación y mejora continua estudiados durante el curso.

## Enunciado del Proyecto

A través de un enfoque basado en **aprendizaje supervisado**, StudentGuard utiliza variables clave como **promedio de calificaciones, asistencia, entrega de tareas, hábitos de estudio y participación en actividades**, para predecir si un estudiante presenta o no riesgo de abandono.

## Detalles del proyecto

### Carga Masiva

Dado que este problema es universal has decidido permitir que, bajo ciertos parámetros preestablecidos, las instituciones puedan utilizar sus propios datos para asegurar que las predicciones reflejan sus contextos. El proyecto debe permitir la carga masiva de archivos en formato csv con los datos de cada institución.

Los parámetros que StudentGuard evalúa para determinar el riesgo de un estudiante son:

- Promedio actual del estudiante (promedio\_actual).
- Porcentaje de asistencia a clases (asistencia\_clases).
- Porcentaje de tareas entregadas (tareas\_entregadas).
- Participación en clase (participacion\_clase).
- Horas dedicadas al estudio (horas\_estudio).
- Promedio en evaluaciones parciales (promedio\_evaluaciones).
- Cantidad de cursos reprobados o retirados (cursos\_reprobados).
- Participación en actividades extracurriculares (actividades\_extracurriculares).
- Cantidad de reportes disciplinarios (reportes\_disciplinarios).

### Limpieza de datos

Ya que los datos han sido cargados debe realizarse un proceso de limpieza que incremente la calidad de los mismos. Deben manejarse diversos escenarios como valores faltantes, estandarizaciones o inconsistencias.

### Entrenamiento

Se debe seleccionar el modelo que, a su criterio, se adapta a la problemática de mejor manera y entrenarlo con los datos preprocesados.

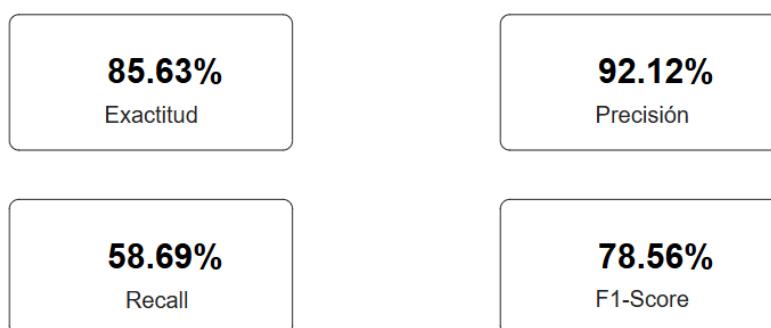


## Evaluación de rendimiento

StudentGuard cuenta con un panel que muestra las métricas de rendimiento del modelo que ha sido entrenado. Esto para que las instituciones puedan conocer la precisión y fiabilidad del modelo generado antes de utilizarlo.

---

### Evaluación de Rendimiento



## Ajuste de Hiperparámetros

También cuenta con un apartado para ajustar hiperparámetros con una interfaz fácil de utilizar, permitiendo mejorar la calidad de las predicciones según los resultados obtenidos.

---

### Ajuste de Hiperparámetros

El panel para ajustar hiperparámetros incluye los siguientes controles:

- Cantidad de árboles en el bosque
- Profundidad máxima del árbol
- Número máximo de hojas por árbol
- Botón Reentrenar el modelo

## Predicciones

Cuando se han alcanzado las métricas de rendimiento deseadas el sistema habilita la predicción individual, facilitando a las autoridades educativas tomar decisiones informadas y oportunas.

## Predicción de Riesgo

Promedio actual	Asistencia a clases
Tareas entregadas	
Participación en clase	
Horas de estudio	Actividades extracurriculares
Cursos reprobados	Reportes disciplinarios

**Resultado**

**Predecir**   **Cancelar**

El objetivo principal es ofrecer una herramienta intuitiva, accesible y basada en datos que apoye la gestión académica y contribuya a la retención estudiantil. StudentGuard no solo automatiza el proceso de análisis, sino que democratiza el uso del aprendizaje automático en el ámbito educativo, permitiendo que distintas instituciones puedan implementar soluciones de inteligencia artificial sin necesidad de conocimientos avanzados en ciencia de datos.

## Restricciones

- No se permite el uso de APIs externas (como la de chatgpt) o modelos preentrenados, el modelo a utilizarse debe ser de su completa creación.
- El lenguaje de programación para el entrenamiento del modelo y el desarrollo del backend debe ser Python.
- El lenguaje de programación para el frontend queda a elección del estudiante. Se recomienda React, Streamlit o Vue.
- La entrega se realizará por medio de Github. El nombre del repositorio debe ser OLC2\_2SEVD25\_ML\_#grupo, donde se creará una carpeta con el nombre Proyecto1.
- Por medio de UEDI se hará entrega del link del repositorio.
- Los repositorios deben mantenerse en privado para evitar copias de código entre estudiantes.
- Se debe realizar el manual técnico en el archivo README dentro de la carpeta de la respectiva, explicando el proceso de limpieza de datos, la selección del modelo y las decisiones de diseño tomadas.

- Todas las dudas durante el proceso de desarrollo deberán realizarse por medio de los foros de UEDI.
- Se recomienda realizar commits frecuentes en Github para demostrar el avance progresivo del proyecto.
- El código debe estar debidamente comentado y seguir buenas prácticas de programación.
- **Fecha de entrega 15/12/2025 a las 23:59 horas.**

## Penalizaciones

- Copias detectadas obtendrán una nota de 0 puntos y se reportará a la Escuela de Ciencias y Sistemas.
- No se permiten entregas tarde.