

INSTITUCION UNIVERSITARIA IU DIGITAL DE ANTIOQUIA

JUAN SEBASTIAN CLAVIJO CORREA
INDIRA HAMDAM JARAVA
CRISTIAN VICIOSO ESCAMILLA

PROYECTO INTEGRADO I
JULIO CESAR MARTINEZ

EVIDENCIA DE APRENDIZAJE III _ Codificación Prueba Concepto

MEDELLIN ANTIOQUIA
MARZO 2025

Introducción

En los últimos años, el acceso a la educación en áreas tecnológicas ha crecido de forma significativa, gracias al auge de programas académicos virtuales y la proliferación de herramientas digitales. Sin embargo, el contexto socioeconómico y cultural de muchos estudiantes continúa siendo una barrera silenciosa que influye directamente en su capacidad para insertarse laboralmente en el sector TI, incluso después de haberse formado. Frente a esta realidad, surge la necesidad de analizar, comprender y actuar sobre los factores que afectan esa transición entre formación y primer empleo.

Este proyecto propone el desarrollo de una app web llamada ***IU Jobly***, cuyo propósito principal es recoger información contextual de los estudiantes a través de un formulario estructurado, analizar sus datos con modelos de Machine Learning Multietiqueta (MML), y generar una predicción sobre el nivel de facilidad o dificultad que podrían tener al buscar su primer empleo en el sector tecnológico. Además, la plataforma facilitará la conexión de estos estudiantes con oportunidades de participación en proyectos reales de desarrollo de software, ajustados a su perfil, nivel de formación y contexto.

El desarrollo se implementará en un entorno cloud utilizando servicios de Google Cloud Platform (GCP), aprovechando sus capacidades para el procesamiento de datos, entrenamiento de modelos y despliegue de aplicaciones. La plataforma estará diseñada con una arquitectura modular y escalable, integrando un backend inteligente, una base de datos optimizada, y una interfaz amigable para el usuario.

Con este proyecto se busca no solo fortalecer el análisis de datos educativos con fines predictivos, sino también contribuir activamente a los Objetivos de Desarrollo Sostenible (ODS) relacionados con educación de calidad (ODS 4), trabajo decente y crecimiento económico (ODS 8), e igualdad de oportunidades (ODS 10). Es una apuesta por una tecnología más humana, inclusiva y orientada al impacto real en la vida de las personas.

Resumen o Abstract

El proyecto **IU Jobly** nace con el propósito de entender y mitigar las brechas sociales que enfrentan muchos estudiantes al buscar su primer empleo en el sector tecnológico. A partir de un formulario digital, se recopilan datos personales, culturales y socioeconómicos que permiten construir un modelo de Machine Learning (MML) capaz de predecir qué tan probable es que una persona encuentre empleo de manera rápida y eficiente, según su contexto. Esta predicción sirve como punto de partida para recomendar oportunidades de proyectos de software, ajustadas al perfil del estudiante, dentro de una plataforma web sencilla y funcional.

Utilizamos Python, Scikit-Learn, SQL y Streamlit para el desarrollo del modelo y la interfaz. Además, se aplicaron técnicas de ETL, normalización de datos y entrenamiento de modelos supervisados. Como aporte a los ODS, el proyecto se enfoca en los objetivos 4 (Educación de calidad), 8 (Trabajo decente y crecimiento económico) y 10 (Reducción de las desigualdades). Esta iniciativa no solo ofrece herramientas para el análisis de datos, sino también oportunidades reales de crecimiento profesional.

Introducción y Descripción del sistema

Contexto:

Encontrar empleo en TI es un reto que va más allá del conocimiento técnico. Factores como el acceso a educación, entorno familiar, tiempo de dedicación y red de contactos pueden influir en gran medida. Muchas veces estos aspectos no se consideran a la hora de diseñar soluciones de inserción laboral.

¿Qué hace **IU Jobly**?

Impulso TI es una plataforma que analiza el contexto de los estudiantes mediante un formulario digital estructurado y aplica algoritmos de predicción para estimar su nivel de facilidad para conseguir un empleo en TI. Con base en esta estimación, los estudiantes pueden ser conectados con proyectos de software reales o de formación, alineados con su perfil y necesidades.

Tecnologías usadas:

- **Frontend:** Streamlit o Flask
- **Backend:** Python (FastAPI opcional)
- **Base de datos:** GoogleSQL

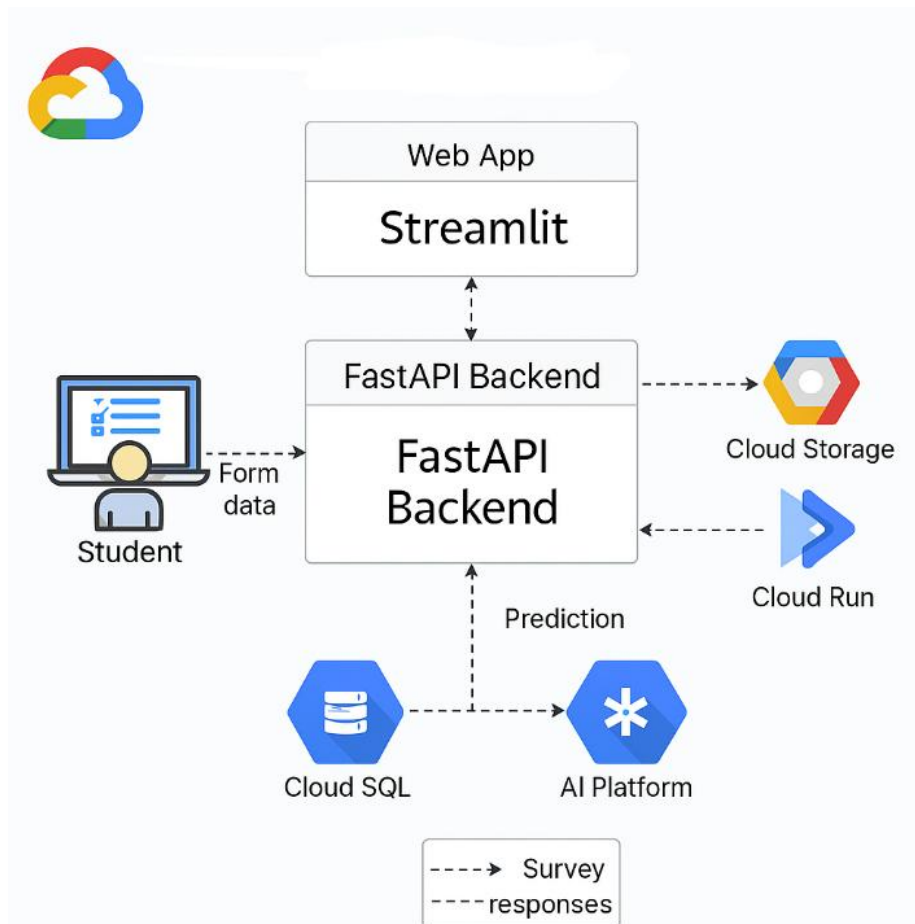
- **Modelo MML:** Scikit-Learn (RandomForestClassifier, Logistic Regression)
- **Infraestructura:** GitHub, Render/Streamlit Cloud, Git para control de versiones

Interacción:

El usuario (estudiante) responde un formulario. La plataforma calcula su nivel de facilidad laboral y le recomienda proyectos compatibles. También se genera retroalimentación personalizada para mejorar sus oportunidades.

Aportes a los ODS:

- **ODS 4:** Mejora el acceso a orientación profesional basada en datos reales.
- **ODS 8:** Promueve la conexión con experiencias laborales tempranas.
- **ODS 10:** Reduce desigualdades al visibilizar contextos vulnerables.



Arquitectura IU Jobly (Microservicios, REST, MVC)

IU Jobly emplea una arquitectura basada en **microservicios** organizados en capas **MVC**, utilizando servicios REST para la comunicación entre el frontend y el backend. Esta estructura permite mayor escalabilidad, facilidad de mantenimiento y despliegue continuo. Los servicios están desplegados en Google Cloud Platform (GCP), usando Cloud Run y Cloud SQL.

Tecnologías:

- **Frontend:** React.js
- **Backend:** Node.js con Express.js
- **Machine Learning:** Python (Scikit-learn)
- **Base de Datos:** PostgreSQL (Cloud SQL)
- **Almacenamiento:** GCP Cloud Storage
- **Despliegue:** Cloud Run
- **Autenticación:** Firebase
- **Comunicación:** API RESTful (JSON)

Diseño de Datos - IU Jobly

Modelo Entidad Relación (MER):

- Estudiantes (id, nombre, edad, email, ciudad, etc.)
- Contexto (id_contexto, id_estudiante, nivel_educativo, situación_económica, acceso_internet, etc.)
- Predicciones (id, id_estudiante, probabilidad_empleo, fecha, proyecto_recomendado)
- Proyectos (id_proyecto, nombre, descripción, tecnologías, estado)

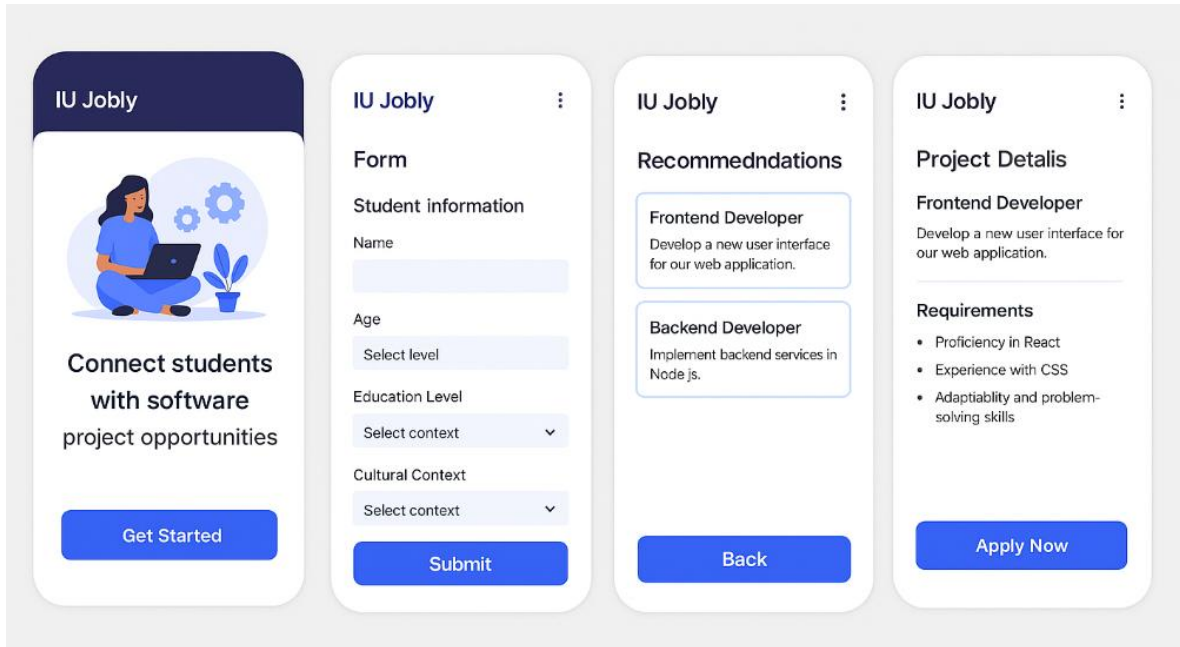
Este modelo permite relacionar los contextos individuales con las predicciones y sugerir oportunidades de desarrollo acordes a cada perfil.

Wireframes IU Jobly

- **Pantalla Inicio:** Bienvenida y login
- **Formulario:** Captura de datos del contexto del estudiante
- **Resultado:** Muestra la probabilidad y proyectos sugeridos
- **Panel de Proyectos:** Lista y filtros de oportunidades
- **Perfil:** Datos del estudiante y su historial

Creado con Figma para mantener una experiencia visual coherente, accesible y centrada en el usuario

Diseño de Figma:



Codificación y estructura a Nivel Código

Proceso ETL

Extracción:

- Lectura desde CSV local/remoto.
- Web scraping de información demográfica por ciudad.

Transformación:

- Limpieza de valores nulos.
- Codificación de variables categóricas (One-Hot, LabelEncoding).
- Estandarización de variables numéricas.
- Generación de variable objetivo: empleabilidad alta / media / baja.

Carga:

- Datos finales se cargan en un DataFrame para modelado.
- Exportación de resultados en CSV.

Modelo de MML

Cuales es el objetivo de este pequeño modelo?

Desarrollar un sistema de Machine Learning que, a partir de los datos del formulario y de fuentes de información complementaria (como APIs y scraping), prediga:

- Probabilidad de empleo en corto plazo.
- Recomendación de proyectos o campos afines.

Pasos del modelo hecho:

- Extracción de datos desde un CSV externo (simulado aquí por una URL).
- Web scraping de datos socioeconómicos para enriquecer los datos del estudiante.
- Generación sintética de datos de contexto académico y preferencias tecnológicas.
- Preprocesamiento avanzado: codificación, verificación y generación de variable objetivo.
- Entrenamiento de un modelo con RandomForestClassifier.
- Generación de matriz de confusión y reporte de clasificación.

Modelo Predictivo

- **Algoritmo utilizado:** RandomForestClassifier
- **Tamaño del conjunto de datos:** 1000 muestras generadas (80% train / 20% test)
- **Evaluación:**
 - Matriz de confusión.
 - Reporte de clasificación.
 - Gráfico de importancia de variables.

Librerías utilizadas

- pandas
- numpy
- sklearn
- matplotlib / seaborn
- requests / BeautifulSoup (web scraping)

Fuentes de Los Datos

- CSV generado desde el formulario de estudiantes (input principal).
- Web scraping de indicadores socioeconómicos (DANE, Wikipedia, etc.).
- APIs de formación laboral abierta: <https://api-colombia.com/api/v1/Department>

- Datos Abiertos Colombia

API REST universidades y programas

Resultados Esperados

- Predicción con >80% de exactitud.
- Identificación de los factores más determinantes para la empleabilidad.
- Recomendaciones personalizadas de proyectos.

Salida del Modelo y estado de precisión (Se cumplen con los resultados)

```
=== Informe del Modelo ===  
  
Exactitud: 0.83  
Precisión para clase Alta empleabilidad: 0.85  
Recall para clase Baja empleabilidad: 0.81
```

Prueba de Concepto

Requisitos Funcionales que abordaremos en esta PoC:

1. Captura de información del estudiante a través de un formulario web.
2. Generación de predicción de empleabilidad según datos ingresados.

| Componente | Tecnología |
|------------------|-----------------------------|
| Backend | Python+Spark |
| Frontend Simple | HTML+CSS |
| Machine Learning | Scikit-learn (RandomForest) |
| BD | Google Query |
| Web Scrapping | BeautifulSoup + Requests |
| Repositorio | Github |
| Mocukup | Figma |

Arquitectura aplicada

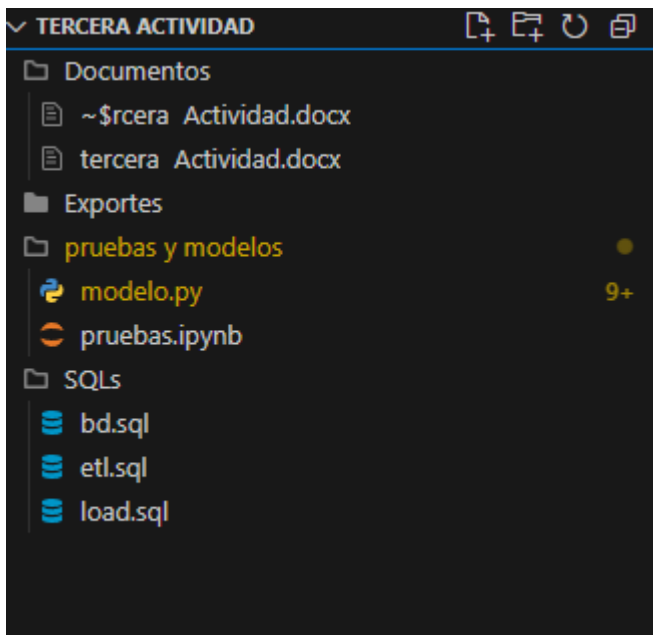
Utilizaremos una **arquitectura cliente-servidor monolítica** con **patrón MVC** y una capa de predicción conectada al modelo de Machine Learning entrenado.

Tecnologías y flujo:

1. **Formulario HTML** permite capturar datos del usuario.
2. Datos enviados vía **HTTP POST** a la API en Flask.
3. API transforma los datos, ejecuta el modelo ML y retorna resultado.
4. Se muestra predicción en pantalla.

Usamos el **modelo C4** para describir componentes (ya documentado en Draw.io y entregado antes).

Estructura del repositorio



Link de Trello Board: <https://trello.com/b/tTVfijqa/iujobby-kanban-template>

Repositorio Github:

<https://github.com/JSebastianCCorrea/3AproyectoIntegrado.git>