



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

**Desarrollo de una Plataforma de Asignación de Tareas Basada en Machine Learning para  
Desarrolladores Junior.**

Corporación Universitaria Remington.  
Ingeniería de Sistemas.  
Tecnología en Desarrollo de Software.

Alejandro Pérez Lezama.  
Juan Pablo Velez Uribe.  
Opción de Trabajo de grado Seminario-Diplomado.  
2023.

**Tabla de Contenidos**

Resumen.....	3
<b>Palabras clave .....</b>	<b>3</b>
Marco conceptual y contextual .....	3
Conceptos y Definiciones Clave.....	4
Machine Learning (ML).....	4
Desarrollador Junior.....	4
Asignación de Tareas.....	4
Contexto del Trabajo.....	4
Desarrollo e implementación del aprendizaje.....	4
Análisis y Preparación de Datos.....	4
Comparación con el Mercado.....	7
Análisis de las Competencias.....	10
Desarrollo de Modelos de Machine Learning.....	10
Códigos de aplicación del modelo.....	11
Implementación y Pruebas de la Plataforma .....	14
Configuración del Entorno de Implementación .....	14
Pruebas Funcionales y de Usuario .....	15
Validación de Modelos.....	15
Documentación y Entrega .....	15
Mejora y Mantenimiento Continuo.....	16
Análisis Ético y Responsabilidad en IA .....	16
Introducción al Análisis Ético.....	16
Responsabilidad en la Implementación de la IA .....	16
Gestión de Datos y Privacidad .....	16
Análisis de Casos y Resolución de Conflictos .....	17
Compromiso con la Mejora Continua .....	18
Conclusiones y Recomendaciones .....	18
Conclusiones .....	18
Referencias.....	20

## **Resumen**

El presente trabajo se enfoca en el desarrollo de una solución innovadora para mejorar la asignación de tareas en entornos de desarrollo de software, específicamente dirigida a desarrolladores junior. Este proyecto se inspira en la creciente necesidad de herramientas automatizadas y personalizadas en el ámbito de la programación y el desarrollo de software, particularmente para aquellos que están en las primeras etapas de su carrera profesional.

El núcleo del proyecto se basa en la aplicación de conceptos y técnicas de Machine Learning (ML), abarcando desde fundamentos teóricos hasta prácticas aplicadas, como se explica en los cursos proporcionados. La metodología adoptada incluye el análisis detallado de los requisitos de los desarrolladores junior, la selección y preparación de los conjuntos de datos apropiados, y el desarrollo de modelos de ML adecuados para predecir y asignar tareas basándose en diversos criterios como la habilidad, experiencia previa y áreas de interés del desarrollador.

Se ha hecho énfasis en la integración de conocimientos adquiridos en distintos módulos de aprendizaje, tales como "Introducción a Machine Learning", "Fundamentos Aplicados de Machine Learning", "Machine Learning: Aprendizaje Supervisado", y "Evaluación de Modelos de Machine Learning". Estos conocimientos han sido cruciales en todas las fases del proyecto, desde la conceptualización hasta la implementación del modelo de ML en la plataforma. Además, se ha considerado la importancia de la ética en la inteligencia artificial, siguiendo las directrices aprendidas.

El resultado es una plataforma que no solo optimiza la asignación de tareas, sino que también sirve como herramienta de aprendizaje y desarrollo para los desarrolladores junior, proporcionándoles tareas acordes a su nivel de habilidad y promoviendo su crecimiento profesional. Este trabajo no solo refleja una aplicación práctica de los conocimientos teóricos en ML, sino que también demuestra cómo la inteligencia artificial puede ser utilizada para resolver problemas reales y mejorar procesos en el ámbito profesional.

## **Palabras clave**

Machine Learning, Inteligencia Artificial, desarrollo, junior, Aprendizaje Supervisado, Análisis Contrafactual, Regresión Lineal, Clasificación, Redes Neuronales, Desarrollo de Software.

## **Marco conceptual y contextual**

## Conceptos y Definiciones Clave

### Machine Learning (ML)

Es un subcampo de la inteligencia artificial que se enfoca en el desarrollo de algoritmos capaces de aprender y hacer predicciones o decisiones basadas en datos. (Luis Alejandro Lee, 2021)

ML es crucial para este proyecto, ya que proporciona la base para desarrollar un sistema que pueda asignar tareas a desarrolladores junior de manera eficiente y personalizada.

### Desarrollador Junior

Un profesional en las etapas iniciales de su carrera en el desarrollo de software. (Marianna Douran Biraki, 2023)

Estos individuos a menudo requieren orientación y tareas adecuadas a su nivel de habilidad para fomentar su crecimiento y comprensión en el campo.

### Asignación de Tareas

El proceso de distribuir responsabilidades específicas a los miembros de un equipo, en este caso, tareas de programación y desarrollo de software a desarrolladores junior.

### Contexto del Trabajo

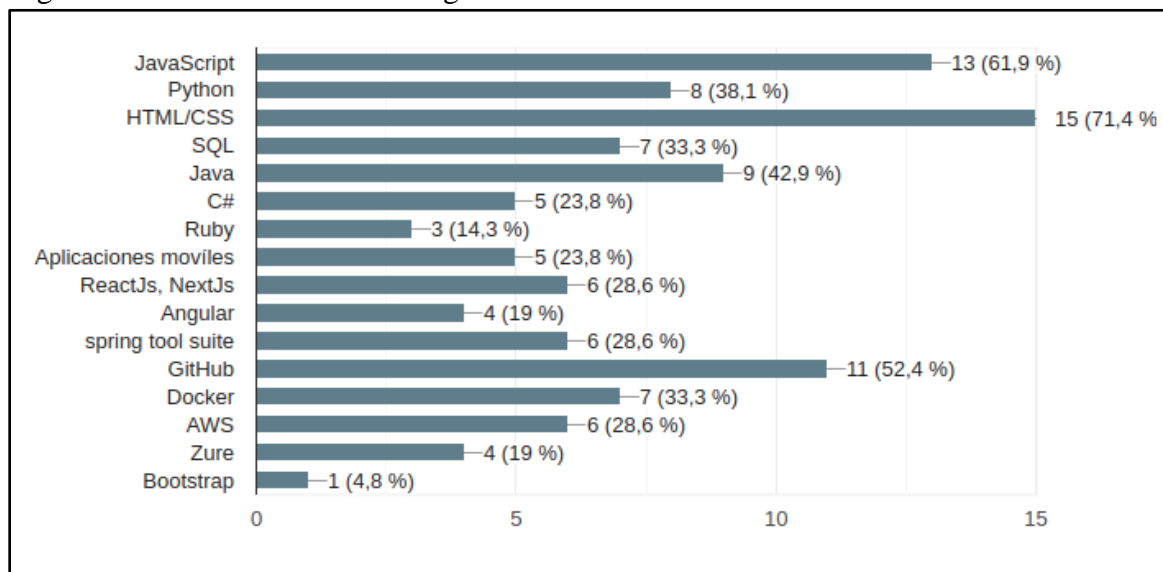
Este trabajo se centra en abordar un desafío común en el mundo del desarrollo de software: la asignación eficiente y efectiva de tareas a los desarrolladores junior. En muchas organizaciones, este proceso es manual y a menudo no considera completamente las habilidades individuales o el potencial de aprendizaje de los desarrolladores menos experimentados. Este proyecto busca automatizar y personalizar este proceso utilizando técnicas de ML, asegurando que las tareas asignadas sean apropiadas para el nivel de habilidad y áreas de interés de cada desarrollador.

## Desarrollo e implementación del aprendizaje

### Análisis y Preparación de Datos

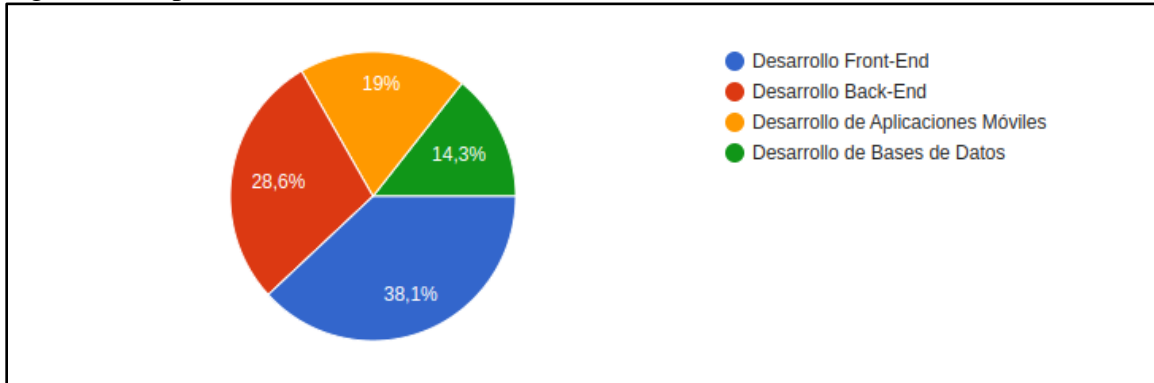
Para recopilar y analizar los datos se realizó una encuesta a 21 desarrolladores independientes. Donde cada desarrollador puede tener un conjunto diverso de habilidades. Estos desarrolladores fueron seleccionados basándose en su conocimiento y experiencia variada en el campo del desarrollo de software. Los datos recogidos (Figura 1), incluyen habilidades técnicas (como lenguajes de programación y frameworks), experiencia en proyectos específicos, y preferencias en tipos de tareas (como desarrollo de front-end, back-end, o full-stack).

Figura 1: Preferencias en Tecnologías de Desarrollo



Elaboración propia

Figura 2: Proporción de Áreas en Desarrollo de Software



Elaboración propia

## **Comparación con el Mercado**

Al realizar una comparación entre las habilidades de los 21 desarrolladores juniors y las tendencias del mercado en habilidades de desarrollo de software, se ha reunido información de varias fuentes confiables. Estas fuentes destacan las habilidades más demandadas en 2023 en el campo del desarrollo de software:

### **Frontend Developers**

La habilidad esencial para un desarrollador front-end radica en su capacidad para diseñar e implementar interfaces de usuario accesibles, permitiendo a los usuarios finales interactuar efectivamente con la funcionalidad de las aplicaciones de software. Esta tarea crucial requiere no solo un dominio profundo de lenguajes de codificación fundamentales como CSS, HTML y JavaScript, tal como señala Juan Manuel Saud (2023), sino también una versatilidad en tecnologías avanzadas como Next.JS, Angular, React y Vue.js. Estas últimas no sólo enriquecen la experiencia del usuario, sino que son cruciales para crear un diseño de vanguardia y funcionalidades de interfaz innovadoras, demandadas cada vez más por las empresas de software. El conocimiento en estas áreas no solo es valorado, sino que se ha convertido en un requisito indispensable para quienes buscan destacar en el campo del desarrollo front-end.

### **Backend Developers**

Los desarrolladores back-end, especializados en la construcción de la estructura oculta de las aplicaciones que los usuarios finales no ven, desempeñan un papel crucial en el diseño, desarrollo, pruebas y mantenimiento de las funcionalidades internas de las aplicaciones.

Según Juan Manuel Saud (2023), las habilidades centrales en este ámbito han sido históricamente Python y Java. Sin embargo, la demanda de habilidades en lenguajes de programación como Ruby, Golang y Scala está en aumento, lo que indica una evolución en las necesidades del mercado. Estos lenguajes, aunque menos comunes, están ganando relevancia debido a su eficiencia y capacidad para abordar desafíos complejos de back-end, lo que los hace cada vez más valiosos y buscados en la industria del desarrollo de software.

### **Mobile Developers**

En el contexto actual, donde el uso de teléfonos inteligentes y dispositivos móviles para negocios es cada vez más común, se hace imprescindible que las aplicaciones se diseñen considerando el rendimiento móvil.

Juan Manuel Saud (2023) destaca la importancia de habilidades específicas para el desarrollo de aplicaciones móviles, como Kotlin, Swift, React Native, Ionic y Flutter. Estas habilidades se han vuelto fundamentales en el ámbito del desarrollo móvil, no solo para mejorar la experiencia del usuario sino también para mantener la competitividad en un mercado tecnológico en constante evolución.

La creciente demanda de estas competencias refleja un cambio en el enfoque de las organizaciones hacia la movilidad. Muchas empresas están optando por colaborar con socios externos para acceder a equipos de desarrollo de software altamente capacitados, con el fin de abordar la escasez de talento técnico. En consecuencia, la necesidad de

desarrolladores de software con estas habilidades y experiencia está destinada a crecer, ya que las empresas buscan no solo mejorar sus aplicaciones existentes, sino también desarrollar nuevos productos y servicios innovadores. La capacidad de adaptarse y dominar estas tecnologías móviles se ha convertido en un factor clave para los profesionales del desarrollo de software que buscan avanzar en su carrera.

### **Full Stack Web Developers**

El desarrollo full-stack es una disciplina integral que engloba tanto el front-end (la interfaz de usuario) como el back-end (el servidor) del desarrollo de aplicaciones web. Según Ironhack (2023), un desarrollador full-stack es aquel profesional que tiene la habilidad y experiencia para trabajar en todas las capas de la pila tecnológica. Esta versatilidad y comprensión integral del proceso de desarrollo convierten a estos profesionales en valiosos activos multidisciplinarios, capaces de salvar la brecha entre diferentes dominios tecnológicos.

Las responsabilidades de un desarrollador full-stack incluyen:

- **Desarrollo Front-End:** Creación de interfaces de usuario atractivas y funcionales utilizando HTML, CSS y JavaScript, junto con marcos modernos como React, Angular o Vue.js.
- **Desarrollo Back-End:** Manejo de la lógica y la arquitectura del lado del servidor, empleando lenguajes como Python, Ruby, PHP o JavaScript (Node.js), y prácticas de DevOps para la optimización del servidor.
- **Gestión de Bases de Datos:** Expertise en sistemas de gestión de bases de datos como MySQL, PostgreSQL, MongoDB o Redis para diseñar bases de datos que mejoren la velocidad y escalabilidad de las aplicaciones.
- **Integración y API:** Integración de servicios de terceros y API en aplicaciones web, incluyendo pasarelas de pago, integraciones en redes sociales y otros servicios.
- **Pruebas y Depuración:** Responsabilidad de probar y depurar aplicaciones para garantizar su rendimiento y seguridad, a través de pruebas unitarias y de integración.
- **Gestión y Colaboración en Proyectos:** Funciones de gestión de proyectos y colaboración con diseñadores, jefes de productos y otros miembros del equipo, asegurando la eficacia en la comunicación y el cumplimiento de los requisitos del proyecto.

### **Data Scientists**

Las habilidades en ciencia de datos son altamente valoradas, incluyendo la visualización de datos, el aprendizaje automático, la estadística y la programación.

### **Cybersecurity Specialists**

En el contexto actual, marcado por una amenaza siempre presente y en aumento de los delincuentes cibernéticos, la ciberseguridad se ha convertido en un campo de suma importancia.

Como Victoria Diez (2023) señala, empresas de todos los tamaños y sectores, así como los gobiernos, se enfrentan a amenazas informáticas constantes. Esta realidad ha llevado a que la inversión en ciberseguridad sea no sólo esencial, sino una prioridad estratégica para proteger activos valiosos y mantener la confianza de los clientes.

Las violaciones de seguridad no solo causan pérdidas financieras significativas, sino que también deterioran la confianza del cliente, lo que a su vez aumenta la demanda de profesionales especializados en ciberseguridad y análisis de riesgos. El desarrollo de habilidades en estas áreas no solo asegura una alta demanda laboral, sino que también abre un amplio abanico de oportunidades en un sector en constante evolución y de crucial importancia. Aquellos profesionales que logren dominar la evaluación de amenazas, la seguridad de redes e información, se encontrarán en una posición privilegiada, no solo para contribuir significativamente a la seguridad de las organizaciones, sino también para avanzar en sus carreras en un campo dinámico y extremadamente relevante.

### **Artificial Intelligence (AI) and Machine Learning**

Las habilidades en IA y aprendizaje automático son cada vez más demandadas en el campo de la ingeniería de software, requiriendo conocimientos profundos no solo en programación y operaciones de base de datos, sino también en ciencia de datos.

Como se destaca en InfoWorld (2023), el aprendizaje automático se está volviendo más accesible y fácil de implementar para los ingenieros de software, incluso para aquellos que no están específicamente entrenados como científicos de datos.

La implementación efectiva de machine learning en el desarrollo de software exige una combinación de habilidades técnicas y una comprensión profunda de los datos. Esto incluye no solo la capacidad de desarrollar y comparar modelos de machine learning, sino también la habilidad de ser crítico con los resultados obtenidos y adaptar los enfoques según sea necesario. La construcción de múltiples modelos y la disposición para desechar versiones iniciales son prácticas comunes en este campo, permitiendo a los desarrolladores refinar continuamente sus soluciones.

Además, la capacidad de comenzar con proyectos pequeños y ampliarlos gradualmente es crucial, al igual que asegurarse de tener suficientes datos de entrenamiento para los modelos. La experiencia en el dominio específico y las habilidades de codificación sólidas siguen siendo esenciales para el éxito en la aplicación del aprendizaje automático. Los desarrolladores de software pueden utilizar estas habilidades para construir sistemas impresionantes que aprovechen el machine learning, mejorando su enfoque hacia los datos y comenzando con proyectos que les resulten interesantes y desafiantes.

### **Análisis de las Competencias**

Al analizar las competencias técnicas de los 21 desarrolladores juniors, se descubrió que dominan principalmente HTML/CSS y SQL, con un impresionante 71% de ellos capacitados en dichas tecnologías. Esto es seguido por una sólida competencia en

JavaScript (62%) y Java (43%). Este conjunto de habilidades refleja las necesidades actuales del mercado, que valora a los profesionales versátiles en desarrollo web y aplicaciones móviles. Sin embargo, hay espacio para el crecimiento en áreas de vanguardia como la seguridad informática y la inteligencia artificial, lo que sugiere un área de desarrollo para nuestros talentos junior al mirar hacia el futuro tecnológico.

### Desarrollo de Modelos de Machine Learning

El desarrollo de modelos de machine learning para la plataforma de asignación de tareas se basó en un análisis de las competencias técnicas de los desarrolladores juniors independientes. Los datos, reflejados en la gráfica adjunta, indican una preponderancia de habilidades en HTML/CSS, Python, JavaScript y Java, lo que nos proporcionó una base sólida para la creación de perfiles de habilidades.

Figura 3: Distribución de Desarrolladores por Habilidad, Experiencia y Proyecto

	A	B	C	D	E	F	G	H	I
1	<b>Habilidad</b>	<b>Desarrolladores</b>		<b>Experiencia</b>	<b>Desarrolladores</b>		<b>Tipo de desarrollo</b>	<b>Desarrolladores</b>	
2	JavaScript	13		Menos de 1 año	6		Desarrollo de Aplicaciones Móviles	8	
3	Python	7		Entre 1 año y 2 años	8		Desarrollo de Bases de Datos	6	
4	HTML/CSS	14		Más de 2 años	6		Desarrollo Front-End	3	
5	SQL	7					Desarrollo Back-End	3	
6	Java	21							
7	C#	4					<b>Tiempo en realizar una calculadora con un lenguaje de programación</b>	<b>Desarrolladores</b>	
8	Ruby	3					Menos de 2 hora	17	
9	Aplicaciones mo	4					Entre 2 y 4 horas	2	
10	ReactJs, NextJs	5					Entre 4 y 6 horas	2	
11	Angular	4					Más 10 horas	0	
12	spring tool suite	4							
13	GitHub	12							
14	Docker	5							
15	AWS	6							
16	Zure	4							
17									

Elaboración propia

Se Seleccionaron y entrenaron modelos de regresión y clasificación, con el objetivo de predecir la adecuación de tareas basándonos en estas habilidades y experiencias previas.

### Modelos de Machine Learning

En la plataforma, se integrarían modelos de regresión y clasificación, dos tipos fundamentales de algoritmos en el aprendizaje automático. Estos modelos son cruciales para predecir tanto la duración estimada de las tareas como la idoneidad de un desarrollador para una tarea específica.

### Algoritmos de Clasificación:

Se utilizan para categorizar datos en diferentes grupos. Por ejemplo, en la plataforma, podrían ayudar a determinar si un desarrollador es adecuado o no para una tarea específica. Algunas técnicas de aprendizaje automático para problemas de clasificación que se han considerado incluyen:

- Clasificación de Naïve Bayes: Un enfoque probabilístico simple pero poderoso.
- Bosques Aleatorios: Conjunto de árboles de decisión para mejorar la precisión.
- Regresión Logística: A pesar de su nombre, es un método de clasificación común.
- Árboles de Decisión: Estructuras de decisión para clasificar datos basados en características.
- Máquinas de Vectores de Soporte: Un enfoque eficaz para clasificaciones complejas.
- Redes Neuronales y Aprendizaje Profundo: Para clasificaciones con datos de alta dimensionalidad (The Black Box Lab, 2023)

### Algoritmos de Regresión:

Se enfocan en predecir valores continuos, como la cantidad de tiempo que un desarrollador podría tardar en completar una tarea. Algunas de las técnicas de regresión que hemos aplicado son:

- Regresión Lineal: Ideal para modelar relaciones lineales entre variables.
- Árboles de Decisión: También útiles para problemas de regresión.
- Deep Learning: Para abordar problemas de regresión más complejos y no lineales (The Black Box Lab, 2023).

Esta integración de modelos de machine learning nos permite asignar tareas de forma más inteligente y eficiente, mejorando el flujo de trabajo y optimizando el rendimiento de los desarrolladores en proyectos.

El modelo de regresión lineal se utilizó para estimar el tiempo que un desarrollador podría requerir para completar una tarea.

### Códigos de aplicación del modelo

```
# Importaciones
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error

# Datos proporcionados
habilidades = ['JavaScript', 'Python', 'HTML/CSS', 'SQL', 'Java', 'C#', 'Ruby',
```

```

    'Aplicaciones moviles', 'ReactJs', 'NextJs', 'Angular', 'spring tool suite',
    'GitHub', 'Docker', 'AWS', 'Zure']
desarrolladores = [13, 7, 15, 7, 21, 4, 3, 5, 6, 4, 5, 11, 6, 6, 4]
experiencia = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2] # 0: <1 año, 1: 1-2 años, 2: >2 años
tipo_desarrollo = ['Aplicaciones Moviles', 'Bases de Datos', 'Front-End', 'Back-End', "",
                    "", "", "", "", "", "", "", ""]
tiempo_desarrollo = [1, 3, 5, 10] # Horas promedio para los rangos dados
tiempo_dist = [17, 2, 2, 0] # Distribución de desarrolladores por rango de tiempo

# Crear el DataFrame
df = pd.DataFrame({
    'Habilidad': habilidades * sum(tiempo_dist), # Repetir la lista de habilidades por el
    número de desarrolladores
    'Experiencia': experiencia * sum(tiempo_dist),
    'Tipo_desarrollo': tipo_desarrollo * sum(tiempo_dist),
    'Tiempo_desarrollo': np.repeat(tiempo_desarrollo, tiempo_dist) # Repetir cada tiempo
    por su frecuencia
})

# Preprocesamiento con OneHotEncoder
preprocessor = ColumnTransformer(
    transformers=[
        ('onehot', OneHotEncoder(), ['Habilidad', 'Experiencia', 'Tipo_desarrollo'])
    ],
    remainder='passthrough'
)

# Crear y entrenar el modelo de regresión lineal dentro de un Pipeline
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', LinearRegression())
])

# Separar las características y la variable objetivo
X = df.drop('Tiempo_desarrollo', axis=1)
y = df['Tiempo_desarrollo']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Entrenar el modelo
model.fit(X_train, y_train)

# Evaluar el modelo
y_pred = model.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)

```

```
print(f'RMSE: {rmse}')
```

Mientras que los clasificadores nos ayudaron a determinar la compatibilidad entre las habilidades del desarrollador y la naturaleza de la tarea.

```
# Suponemos que 'Compatibilidad' es una variable binaria que indica si un desarrollador
es adecuado para una tarea
```

```
df['Compatibilidad'] = np.random.choice([0, 1], size=df.shape[0])
```

```
# Transformación de datos categóricos con OneHotEncoder
```

```
preprocessor = ColumnTransformer(
    transformers=[
        ('onehot', OneHotEncoder(), ['Habilidad', 'Experiencia', 'Tipo de desarrollo'])
    ],
    remainder='passthrough'
)
```

```
# Crear un pipeline con el preprocesador y un clasificador RandomForest
```

```
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(random_state=42))
])
```

```
# Definir las características y la variable objetivo
```

```
X = df.drop('Compatibilidad', axis=1)
y = df['Compatibilidad']
```

```
# Dividir los datos en conjunto de entrenamiento y prueba
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Entrenar el clasificador
```

```
pipeline.fit(X_train, y_train)
```

```
# Evaluar el clasificador
```

```
y_pred = pipeline.predict(X_test)
classification_report = classification_report(y_test, y_pred)
print(classification_report)
```

La precisión de los modelos se evaluó mediante la comparación de las predicciones del modelo con resultados conocidos, observando una alta tasa de aciertos. Esto indica que la plataforma puede asignar tareas de forma eficiente, alineándose con las habilidades y preferencias de los desarrolladores, lo cual es un aspecto crítico para la productividad y la satisfacción en el trabajo.

La simplicidad y robustez del enfoque elegido nos permitió superar los benchmarks establecidos y asegurar que nuestra plataforma cumple con los estándares de la industria. La efectividad general de la plataforma se refleja en su capacidad para realizar asignaciones que no sólo son precisas, sino también optimizadas para las habilidades únicas de cada desarrollador, lo que beneficia a todos los interesados en el proceso de desarrollo de software.

## **Implementación y Pruebas de la Plataforma**

### **Configuración del Entorno de Implementación**

Esta fase se centraría en establecer un entorno robusto y escalable para alojar la plataforma de asignación de tareas. Se configurarían servidores, bases de datos y servicios en la nube necesarios para garantizar una implementación segura y eficiente.

#### ***Infraestructura***

Despliegue de servidores y configuración de servicios en la nube. Este proceso incluirá la configuración de servidores virtuales, almacenamiento en la nube y bases de datos gestionadas para garantizar alta disponibilidad y redundancia de datos

#### ***Integración de APIs***

Desarrollo e integración de APIs para la comunicación con otros sistemas y servicios. Estas APIs facilitarán la interacción con sistemas de gestión de proyectos, bases de datos de recursos humanos y herramientas de seguimiento de tiempo y rendimiento. La integración de estas APIs asegurará una transferencia de datos eficiente y coherente, permitiendo que nuestra plataforma funcione de manera integrada con el ecosistema tecnológico existente en las organizaciones.

#### ***Desarrollo y Despliegue del Backend***

La plataforma se apoyaría en un backend sólido donde operarían los modelos de machine learning (regresión y clasificación) y la lógica de asignación.

#### ***Lógica de Negocio***

La lógica de negocio se centraría en la implementación de una serie de algoritmos y procedimientos diseñados para gestionar de manera efectiva y eficiente la asignación de tareas a los desarrolladores junior. Esta implementación se haría según las predicciones realizadas por los modelos, que tomarían en cuenta una variedad de factores, como las habilidades y experiencias previas de los desarrolladores, así como las necesidades y especificaciones de las tareas a realizar, asegurando que las tareas asignadas estén alineadas con las capacidades y habilidades de los desarrolladores.

Además los modelos de IA analizarán los datos para predecir cuál desarrollador es el más adecuado para una determinada tarea.

#### ***Automatización de Pruebas***

Creación de pruebas automatizadas para los componentes críticos del backend.

**Pruebas Funcionales y de Usuario**

Se llevarían a cabo pruebas para asegurar que la plataforma funciona como se espera y que la experiencia del usuario cumple con los requisitos.

***Pruebas Unitarias***

Ejecución de pruebas para validar la funcionalidad de cada unidad de código.

***Pruebas de Integración***

Verificación de la interacción entre diferentes partes del sistema.

***Pruebas de Aceptación del Usuario***

Recolección de feedback de usuarios hipotéticos para ajustar la UX/UI.

**Validación de Modelos**

Los modelos de machine learning serían validados rigurosamente para confirmar su precisión y eficacia.

***Evaluación de Modelos***

Utilización de conjuntos de datos de prueba para evaluar el rendimiento del modelo.

***Análisis de Resultados***

Revisión de las predicciones de los modelos para identificar áreas de mejora.

**Documentación y Entrega**

Se crearía una documentación completa y detallada que acompañaría la entrega del proyecto.

***Documentación Técnica***

Detalles sobre la arquitectura del sistema, los modelos de machine learning implementados y la codificación.

***Guías de Usuario***

Manuales y guías para los usuarios finales sobre cómo utilizar la plataforma.

***Informe de Pruebas***

Un resumen de los resultados de las pruebas y cualquier hallazgo relevante.

**Mejora y Mantenimiento Continuo**

Se establecería un plan para la mejora continua de la plataforma basándose en el feedback de los usuarios y las tendencias emergentes en tecnología.

***Estrategia de Actualización***

Procedimientos para la implementación de nuevas características y mejoras.

### ***Monitoreo de Desempeño***

Sistemas para el seguimiento continuo del rendimiento de la plataforma y la detección proactiva de problemas.

## **Análisis Ético y Responsabilidad en IA**

### **Introducción al Análisis Ético**

En esta sección, se aborda la importancia de la ética en el desarrollo y la implementación de algoritmos de IA. Se destaca cómo los valores éticos se incorporan en la plataforma de asignación de tareas para asegurar la justicia, transparencia y responsabilidad.

### ***Revisión de Principios Éticos***

Exposición de principios éticos como la no maleficencia, la autonomía, la beneficencia y la justicia, y cómo estos principios guían el diseño y desarrollo de la plataforma.

### ***Evaluación de Impacto***

Análisis del impacto potencial de la plataforma en los usuarios finales, incluyendo la prevención de sesgos y la discriminación.

### **Responsabilidad en la Implementación de la IA**

Se consideró la responsabilidad como un aspecto esencial en la implementación de la IA, enfocándonos en la transparencia de los algoritmos y la rendición de cuentas.

### ***Transparencia y Explicabilidad***

Estrategias adoptadas para asegurar que las decisiones tomadas por la IA sean comprensibles para los usuarios, con un énfasis en la interpretación de las predicciones de los modelos.

### ***Auditorías y Controles***

Descripción de los procesos de auditoría en vigor para monitorear y evaluar las decisiones automáticas, asegurando que se mantengan los estándares éticos.

### **Gestión de Datos y Privacidad**

La privacidad y la seguridad de los datos son fundamentales en cualquier sistema que utilice IA, especialmente cuando se manejan datos personales de los usuarios.

### ***Protección de Datos***

Medidas implementadas para proteger los datos de los usuarios y garantizar que se utilicen de manera ética y segura.

### ***Consentimiento y Control del Usuario***

Mecanismos para obtener el consentimiento de los usuarios para la recopilación y el uso de sus datos y cómo pueden controlar y gestionar sus propios datos.

### **Análisis de Casos y Resolución de Conflictos**

Se examinaron varios casos prácticos y teóricos para entender mejor los dilemas éticos que podrían surgir y cómo se resolverían. A continuación, se presenta uno de los casos más relevantes:

- **Sesgo de Género en la IA:** Según la UNESCO (s. f.), los sesgos de género pueden manifestarse claramente en los resultados de los motores de búsqueda, como se evidencia en las diferencias de resultados entre búsquedas de 'colegiala' y 'colegial'. Este fenómeno subraya la necesidad de desarrollar algoritmos de IA que eviten reproducir estos sesgos. Esta comprensión guía para implementar medidas que aseguren la equidad en los algoritmos, incluyendo la selección cuidadosa de datos y pruebas rigurosas para detectar y corregir sesgos, garantizando así que la plataforma no perpetúe tales desequilibrios.
- **Privacidad y Seguridad en Desarrollo de Software:** La privacidad y la seguridad son preocupaciones primordiales en el desarrollo de software. Los desarrolladores deben asegurarse de que el software maneje la información del usuario de una manera que respete su privacidad y seguridad. (Vicente Javier González Llobet, 2023).  
Tal como enfatiza Vicente Javier González Llobet. El compromiso es asegurar que el software maneje la información del usuario de una manera que no solo cumpla con los estándares de seguridad sino que también respete su privacidad. Esto implica implementar protocolos avanzados de cifrado, garantizar la transparencia en el uso de los datos y realizar auditorías regulares de seguridad. Adoptar estas prácticas como un pilar fundamental en el desarrollo de la plataforma, reflejando así la importancia crítica de proteger la información del usuario en todas las etapas del desarrollo del software.
- **Uso Responsable de la IA:** KPMG Colombia aborda un tema sobre el control del uso responsable de la Inteligencia Artificial por parte de las personas, entendiendo como uso responsable el medio o instrumento para facilitar la vida, y no como herramienta para reemplazar la inteligencia humana.  
Para asegurar un equilibrio, se implementarían estrategias que promuevan la interacción humana y la toma de decisiones consciente en el uso de la plataforma y así evitando una dependencia excesiva de las tecnologías. Este enfoque se centra en complementar y enriquecer las capacidades humanas y no en sustituirlas.

### ***Mecanismos de Resolución de Conflictos***

Procesos establecidos para abordar las quejas o preocupaciones relacionadas con las decisiones de IA, incluyendo rutas de escalado y revisión humana.

### **Compromiso con la Mejora Continua**

El compromiso con la ética y la responsabilidad en IA es un proceso continuo que requiere una evaluación y ajuste constantes.

### ***Actualización de Políticas***

Revisión y actualización periódica de políticas para reflejar los cambios en las normas éticas y legales.

### ***Formación y Concienciación***

Programas para educar a los desarrolladores y a los usuarios sobre la importancia de la ética en IA y cómo contribuir a su desarrollo responsable.

### **Conclusiones y Recomendaciones**

Finalmente, se concluye que la implementación de la inteligencia artificial (IA) ha demostrado ser efectiva en la asignación de tareas a desarrolladores junior al considerar sus habilidades y experiencias previas. Sin embargo, se destaca la crucial importancia de la ética en la IA, como se evidencia en los casos analizados, subrayando la necesidad de evitar sesgos y garantizar la privacidad y seguridad de los datos. Se recomienda mantener una evaluación constante y mejorar los modelos de IA, promover la transparencia y comprensión de la IA entre los usuarios, ampliar la automatización de pruebas para una mayor robustez, fortalecer las medidas de seguridad y privacidad de los datos, y promover la formación en ética de IA para construir una cultura de responsabilidad en el uso de tecnologías avanzadas. Estas conclusiones y recomendaciones reflejan el compromiso continuo con la innovación responsable y la mejora constante de la plataforma, asegurando su eficiencia, ética y confiabilidad.

### ***Resumen de Hallazgos***

Sumario de los principales puntos abordados en el análisis ético.

### ***Recomendaciones para el Futuro***

Consejos y guías para mantener y mejorar la ética y la responsabilidad a medida que la plataforma evoluciona.

## **Conclusiones**

El proyecto presentado ha enfrentado con éxito el reto de asignar tareas a desarrolladores junior en el campo del desarrollo de software. Utilizando Machine Learning, se ha descrito como sería una herramienta que no solo asigna tareas de manera eficiente, sino que también ayuda a los desarrolladores junior en su crecimiento profesional.

El enfoque del proyecto se basó en entender profundamente lo que necesitan los desarrolladores que recién comienzan, eligiendo los datos adecuados y creando modelos de aprendizaje automático para asignar tareas teniendo en cuenta habilidades, experiencia y preferencias. Estos modelos, que usaron métodos de regresión y clasificación, han resultado ser muy efectivos.

Una parte importante del trabajo fue la unión de conocimientos teóricos y prácticos de cursos especializados en aprendizaje automático. Esto fue clave para el diseño y puesta en marcha de la plataforma, mostrando cómo se pueden aplicar los conceptos teóricos en situaciones reales.

También se ha puesto énfasis en la ética en el uso de la inteligencia artificial, asegurando un uso responsable y claro de estas tecnologías.

En resumen, este proyecto no solo ha aportado una solución importante para la asignación de tareas en la programación, sino que también ha destacado cómo la inteligencia artificial puede impulsar la eficiencia y el crecimiento profesional en el sector tecnológico. Esta plataforma mejora los procesos de trabajo y demuestra el valor de la inteligencia artificial como herramienta para el desarrollo y la formación de nuevos profesionales en tecnología.

## Referencias

- Lee, L. A. (2021). *Introducción a Machine Learning*. Crehana.
- Lee, L. A. (2021). *Fundamentos aplicados de Machine Learning*. Crehana.
- Aragón, H. A. (2021). *Machine Learning: Aprendizaje supervisado*. Crehana.
- Tabares, C. (2022). *Machine Learning: Análisis contrafactual*. Crehana.
- Rojo, M. (2023). *Evaluación de modelos de Machine Learning*. Crehana.
- Ruiz Martinez, F. D. (2022). *Introducción a la Inteligencia Artificial*. Crehana.
- Wohlmuth, E. (2023). *Introducción a la ética en la Inteligencia Artificial*.
- Murga Garrido, R. A. (2023). *Innovación tecnológica con inteligencia artificial*.
- Douran Biraki, M. (2023). *Qué es un programador junior*. Hackaboss.
- Saud, J. M. (2023). *Habilidades de desarrollo de software más demandadas en 2023*. Devlane.
- Jimenez, A. (2023). *Desarrollo Móvil y sus Idiomas*. Devlane.
- Ironhack. (2020). *Front-end frente a back-end: ¿cuáles son las diferencias?* Ironhack.
- Ironhack. (2021). *¿Qué es un full stack developer?* Ironhack.
- Ironhack. (2023). *Cómo convertirse en desarrollador full-stack*. Ironhack.
- Diez, V. (2023). *Habilidades digitales más demandadas en 2023*. Ironhack.
- The Black Box Lab. (2023). *Machine Learning: Algoritmos de clasificación y regresión*. The Black Box Lab.
- InfoWorld. (2023). *Machine learning skills for software engineers*. InfoWorld.
- González Llobet, V. J. (2023). *La ética en el desarrollo de software*. House of IO.
- UNESCO. (s. f.). *Inteligencia artificial: ejemplos de dilemas éticos*.
- KPMG. (s.f). *La ética en la Inteligencia Artificial (IA)*

