

# Openhire: Portal de Empleo Inclusivo

Raul Elizalde Breton, Victor Manuel Zamora Ramos

*CENTRO UNIVERSITARIO DE CIENCIAS  
EXACTAS E INGENIERÍAS, (CUCEI, UDG)*

raul.elizalde3473@alumnos.udg.mx  
victor.zamora@academicos.udg.mx

**Abstract—** OpenHire es un portal de empleo inclusivo dedicado a crear un espacio de oportunidades específicamente diseñado para promover la diversidad y la inclusión en el mercado laboral. Su objetivo principal es conectar a empleadores comprometidos con la inclusión con candidatos diversos, facilitando un entorno de trabajo equitativo y accesible para todos. Para optimizar la búsqueda de empleo, OpenHire utiliza un algoritmo de filtrado colaborativo que analiza las preferencias de los aplicantes para recomendarles las oportunidades laborales más adecuadas. Este enfoque personalizado mejora la experiencia del usuario, aumentando las posibilidades de encontrar un empleo que se ajuste perfectamente a sus capacidades y expectativas. OpenHire se posiciona como un recurso esencial tanto para empresas que buscan enriquecer su equipo con talento diverso, como para profesionales que desean encontrar un entorno laboral inclusivo y respetuoso.

**Palabras claves –** Proyecto modular, Portal de empleo inclusivo, Diversidad, Inclusión, Oportunidades laborales, Algoritmo de filtrado colaborativo, Recomendación de empleos, Preferencias y habilidades, Experiencia del usuario, Empresas inclusivas, Talento diverso.

## I. INTRODUCCIÓN

La inclusión laboral de personas con discapacidad sigue siendo un desafío significativo en México, donde la falta de oportunidades y condiciones laborales justas persiste. A pesar de los avances en otros ámbitos, el mercado laboral aún muestra una considerable brecha que necesita ser abordada de manera eficaz y sostenible.

OpenHire es un portal de empleo inclusivo que aborda la problemática de la baja participación laboral de personas con discapacidad en México. Mediante el uso de un algoritmo de filtrado colaborativo, OpenHire recomienda empleos basados en las habilidades y preferencias de los candidatos, mejorando sus posibilidades de empleo. Este proyecto es esencial tanto para las empresas que buscan diversificar sus equipos como para los profesionales con discapacidad que desean un entorno laboral inclusivo y accesible. Con OpenHire, se busca no solo incrementar la participación de personas con discapacidad en el mercado laboral, sino también asegurar condiciones de trabajo justas y dignas para todos.

### A. Planteamiento del problema

En México, un millón de personas viven con discapacidad, pero de manera alarmante, 7 de cada 10 de ellas están sin trabajo, lo que implica que solo el 30% participa activamente en el mercado laboral. Esta estadística revela una realidad preocupante: la falta de oportunidades laborales inclusivas y accesibles. Además, entre aquellos que logran encontrar empleo, cinco de cada diez no cuentan con prestaciones, y el 15% no recibe ninguna remuneración. Estas cifras subrayan la necesidad urgente de iniciativas que promuevan la integración laboral de las personas con discapacidad, asegurando condiciones de trabajo justas y equitativas.

### B. Justificación

OpenHire surge como una respuesta innovadora a esta problemática. Este portal de empleo inclusivo está dedicado a crear un espacio de oportunidades específicamente diseñado para personas con discapacidad, promoviendo la diversidad y la inclusión en el mercado laboral. OpenHire tiene como objetivo conectar a empleadores comprometidos con la inclusión con candidatos diversos, facilitando un entorno de trabajo equitativo y accesible para todos.

Para optimizar la búsqueda de empleo, OpenHire utiliza un algoritmo de filtrado colaborativo. Este algoritmo analiza las preferencias y habilidades de los aplicantes para recomendarles las oportunidades laborales más adecuadas. Este enfoque personalizado no solo mejora la experiencia del usuario, sino que también incrementa significativamente las posibilidades de que los candidatos encuentren un empleo que se ajuste perfectamente a sus capacidades y expectativas.

El proyecto OpenHire no solo busca aumentar la participación de personas con discapacidad en el mercado laboral, sino también garantizar que estas oportunidades sean dignas y justas. Al conectar a los empleadores con candidatos calificados y diversos, OpenHire ayuda a las empresas a enriquecer sus equipos con talento variado y a fomentar un entorno de trabajo inclusivo y respetuoso.

## II. TRABAJOS RELACIONADOS

El desarrollo de OpenHire se inspiró en el éxito y las innovaciones de otras aplicaciones relacionadas que han demostrado la eficacia de

enfoques inclusivos y tecnológicos en el mercado laboral. Plataformas como LinkedIn y Glassdoor, aunque no específicas para personas con discapacidad, han revolucionado la forma en que las personas buscan empleo y se conectan con empleadores, utilizando algoritmos avanzados para emparejar candidatos con oportunidades laborales adecuadas. Estas herramientas mostraron el potencial de la tecnología para personalizar la búsqueda de empleo y mejorar la accesibilidad de la información.

Asimismo, aplicaciones especializadas como Incluyeme.com, que se centra en la inclusión laboral de personas con discapacidad en América Latina, sirvieron como un modelo directo. Incluyeme.com demostró la importancia de crear un espacio dedicado exclusivamente a este grupo, adaptando las funcionalidades del portal a sus necesidades específicas. El éxito de estas plataformas en conectar a personas con discapacidad con empleadores inclusivos subrayó la demanda y viabilidad de un servicio similar en México.

Inspirándose en estas aplicaciones, OpenHire adoptó un enfoque centrado en el usuario, incorporando un algoritmo de filtrado colaborativo avanzado para recomendar empleos basados en las habilidades y preferencias de los candidatos. Además, se integraron características de accesibilidad para garantizar que el portal fuera fácil de usar para todos. Al aprender de las mejores prácticas y éxitos de otras plataformas, OpenHire se posiciona como una herramienta necesaria para promover la inclusión laboral en México.

### III. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO MODULAR

#### A. *Hipótesis*

La implementación de un portal de empleo inclusivo, apoyado por un avanzado algoritmo de filtrado colaborativo, aumentará significativamente la tasa de empleo y mejorará las condiciones laborales de las personas con discapacidad en México. Al facilitar la conexión entre candidatos diversos y empleadores comprometidos con la inclusión, se espera reducir la brecha laboral existente y fomentar un entorno de trabajo más equitativo y accesible.

#### B. *Metodología*

Primeramente se creó una labor de investigación sobre la problemática del proyecto. Posteriormente, se definieron los objetivos y el establecimiento de metas.

El método de cascada se eligió para el desarrollo del proyecto debido a varias razones que se alinean con las necesidades y características del proyecto. En primer lugar, dado que OpenHire es un proyecto con objetivos bien definidos desde el principio, como crear un portal de empleo inclusivo con funcionalidades específicas, el método de cascada es ideal. Este método se basa en una fase inicial de planificación detallada y definición de requerimientos, lo cual es crucial para un proyecto con un enfoque claro y concreto.

Además, se requiere un enfoque estructurado y ordenado para garantizar la calidad y coherencia del producto final. Cada etapa del desarrollo se completó antes de pasar a la siguiente, lo que facilitó la identificación y mitigación temprana de posibles problemas o desviaciones del plan inicial.

Una vez definido los objetivos y los requerimientos del proyecto se eligieron las siguientes herramientas tecnológicas acorde al presupuesto y las necesidades del mismo.

En el entorno de desarrollo del proyecto, decidí utilizar Ubuntu como sistema operativo por varias razones fundamentales que contribuyeron significativamente al éxito y eficiencia del proceso de desarrollo. En primer lugar, la estabilidad y confiabilidad de Ubuntu fueron aspectos cruciales que influyeron en mi decisión. Durante el desarrollo del proyecto, era esencial contar con un sistema operativo sólido y robusto que proporcione un entorno de trabajo estable y libre de errores. Ubuntu ha demostrado consistentemente ser una opción confiable en este sentido, con actualizaciones regulares y una comunidad activa que garantiza un rendimiento óptimo y una experiencia de usuario fluida.

Además, la flexibilidad y compatibilidad de Ubuntu con una amplia variedad de herramientas y tecnologías de desarrollo fueron factores clave en su elección. Al ser una distribución de Linux ampliamente utilizada en el ámbito del desarrollo de software, Ubuntu ofrece una amplia gama de herramientas de desarrollo y bibliotecas de software que facilitó el proceso de desarrollo. Desde la instalación de paquetes y dependencias hasta la configuración de entornos de desarrollo específicos, Ubuntu proporcionó una plataforma versátil y adaptable que se ajustaba perfectamente a las necesidades del proyecto.

Se utilizó Django como el framework principal para desarrollar el backend del portal. Django proporcionó un entorno robusto y seguro para manejar la lógica de negocio y la administración de usuarios. El lenguaje de programación Python se empleó para escribir la lógica del servidor y las funcionalidades del algoritmo de filtrado colaborativo.

Se diseñó la interfaz de usuario utilizando HTML y CSS, asegurando que fuera accesible y fácil de usar para computadoras y móviles.

Se utilizó Render para el despliegue y alojamiento del portal y la base de datos, facilitando un entorno escalable y confiable para la aplicación.

Amazon S3 se empleó para el almacenamiento seguro y eficiente de archivos estáticos del proyecto, proporcionando alta disponibilidad y durabilidad.

Se utilizaron diversas bibliotecas de Python especializadas en machine learning y análisis de datos (scikit learn, pandas, seaborn y numpy para desarrollar el algoritmo de filtrado colaborativo.

#### C. *Diseño del Proyecto Modular*

1) *Arquitectura del Sistema.* El sistema de "OpenHire" se diseñó siguiendo una arquitectura que separa claramente el frontend del backend, asegurando modularidad y escalabilidad. La aplicación utiliza tecnologías robustas y bien establecidas para garantizar un rendimiento óptimo y una experiencia de usuario satisfactoria.

El frontend está construido utilizando HTML, CSS y Bootstrap, proporcionando una interfaz de usuario interactiva y responsiva. El backend se desarrolló en Python utilizando el framework Django, que gestiona la lógica del proyecto, las solicitudes del usuario y la

interacción con la base de datos. Para el almacenamiento de datos, se empleó una base de datos de modelo relacional desplegada en la web.

2) *Estructura del Proyecto.* El proyecto está organizado en varias aplicaciones dentro del framework Django, cada una responsable de diferentes aspectos del sistema:

La aplicación de applicants se encarga de gestionar los datos y operaciones relacionados con los solicitantes de empleo, permitiendo el registro y seguimiento de sus aplicaciones.

La aplicación de employers gestiona los datos de los empleadores y sus ofertas de trabajo, facilitando la creación, edición y publicación de ofertas de empleo.

La aplicación de job\_applications administra las solicitudes de empleo recibidas, permitiendo a los empleadores revisar y gestionar las aplicaciones de los candidatos.

La aplicación de job\_offers maneja la creación y gestión de las ofertas de trabajo, asegurando que las ofertas se presenten de manera clara y accesible para los solicitantes.

La aplicación de resumes almacena y gestiona los curriculums de los candidatos, facilitando su acceso y revisión por parte de los empleadores.

La aplicación de users gestiona la autenticación y autorización de los usuarios, asegurando que solo los usuarios autorizados puedan acceder a ciertas funcionalidades del sistema.

TABLA I  
CARACTERÍSTICAS DE LOS EMPLEADORES

Tipo de Usuario	Empleadores
Formación	Principalmente Recursos Humanos
Habilidades	<div><input type="checkbox"/> Habilidades de gestión,</div> <div><input type="checkbox"/> Toma de decisiones.</div> <div><input type="checkbox"/> Comunicación efectiva.</div> <div><input type="checkbox"/> Experiencia en reclutamiento.</div>
Actividades	<div><div>▪ Crear y completar perfil</div><div>▪ Publicar ofertas de empleo.</div><div>▪ Revisar y gestionar solicitudes de candidatos.</div></div>

TABLA II  
CARACTERÍSTICAS DE LOS CANDIDATOS

Tipo de Usuario	Candidatos
Formación	Diversa variedad profesional
Habilidades	Diversa variedad de habilidades
Actividades	<div><div>▪ Crear y completar perfil.</div><div>▪ Buscar ofertas de empleo según sus habilidades.</div><div>▪ Aplicar a ofertas de</div></div>

	<div><div>empleo.</div><div>▪ Gestionar aplicaciones de empleo.</div></div>
--	---

3) *Componentes Clave.* El sistema incluye varios componentes clave diseñados para mejorar la eficiencia y efectividad del proceso de contratación:

- Los paneles de control proporcionan dashboards personalizados tanto para empleadores como para solicitantes, ofreciendo una vista general del estado de las contrataciones/solicitudes y facilitando la toma de decisiones informadas.
- Los módulos de gestión permiten la creación, edición y seguimiento de ofertas de trabajo y aplicaciones de empleo, proporcionando herramientas intuitivas y fáciles de usar para la administración de procesos de contratación.
- El sistema de autenticación asegura que solo los usuarios autenticados y autorizados puedan acceder a ciertas funcionalidades del sistema, garantizando la seguridad de la información y la privacidad de los datos.

D. *Fases del Proyecto Modular*

1) *Fase de Planificación y Diseño.* La fase de planificación comenzó con la definición de los requisitos del sistema, donde se identificaron las necesidades del proyecto y se documentaron los requisitos funcionales y no funcionales. Se establecieron los objetivos del proyecto y los criterios de éxito, asegurando una visión clara y compartida del producto final. Además, se realizó un análisis de mercado exhaustivo para investigar soluciones similares e identificar brechas y oportunidades de diferenciación.

En la fase de diseño, se creó la arquitectura del sistema mediante diagramas detallados que mostraban la interacción entre el frontend, backend y la base de datos. Se desarrollaron wireframes y prototipos de las interfaces de usuario en Canvas, asegurando una experiencia de usuario intuitiva y eficiente. El modelado de datos incluyó la estructura de datos necesaria para soportar las funcionalidades del sistema.

Durante el desarrollo del diseño de las interfaces para el proyecto, se optó por visualizar de manera holística y detallada cada aspecto del diseño. En lugar de depender de una estructura rígida y jerárquica.

El proceso comenzó con la identificación y comprensión profunda de los requisitos y objetivos del proyecto, así como las necesidades y expectativas de los usuarios finales. Esta etapa inicial fue fundamental para establecer un punto de partida claro y garantizar que el diseño final cumpliera con los criterios de éxito del proyecto.

A continuación, se procedió a la creación del lienzo Canvas, que sirvió como espacio de trabajo virtual para organizar y visualizar todos los elementos del diseño de la interfaz de usuario. En este lienzo, se incluyeron secciones dedicadas a diferentes aspectos del diseño, como la arquitectura de la información, la disposición visual, la navegación y la interacción del usuario.

En cada sección del lienzo, se exploraron múltiples ideas y conceptos de diseño, utilizando herramientas como esquemas, diagramas de flujo, wireframes y prototipos de baja fidelidad con el propósito de llevar a cabo la lluvia de ideas, lo que permitió una exploración creativa y una iteración rápida de los conceptos.

A medida que el diseño evolucionó, se realizaron revisiones y refinamientos continuos en el lienzo Canvas. Se recopilaban comentarios de los interesados y se realizaron ajustes según fuera necesario para mejorar la claridad, la usabilidad y la estética del diseño. Esta iteración constante garantizó que el diseño final reflejara con precisión los requisitos del proyecto y las necesidades de los usuarios finales.

La elección de utilizar el algoritmo de recomendación de manera extensa en el proyecto se basó en varias consideraciones fundamentales que buscaban garantizar la eficacia y la relevancia de las recomendaciones ofrecidas a los usuarios. En primer lugar, la implementación extensiva del algoritmo permitió aprovechar al máximo la riqueza de los datos históricos disponibles sobre las solicitudes de empleo. Al procesar una cantidad significativa de datos, el algoritmo pudo identificar patrones de comportamiento más sólidos y generar recomendaciones más precisas y relevantes.

Además, la utilización extensiva del algoritmo permitió una mayor personalización en las recomendaciones ofrecidas a los usuarios. Al analizar detenidamente las preferencias y los comportamientos individuales de cada usuario, el algoritmo pudo adaptar las recomendaciones según las necesidades y los intereses específicos de cada usuario, lo que mejoró significativamente la experiencia del usuario y aumentó la probabilidad de éxito en la búsqueda de empleo.

Otra justificación importante para el uso extensivo del algoritmo fue su capacidad para adaptarse y evolucionar con el tiempo. Al entrenarse con una gran cantidad de datos y recibir retroalimentación continua de los usuarios, el algoritmo pudo mejorar gradualmente su precisión y relevancia a medida que se recopilaban y procesaban más datos. Esto garantiza que las recomendaciones permanecieran actualizadas y reflejaran con precisión las tendencias y preferencias cambiantes en el mercado laboral.

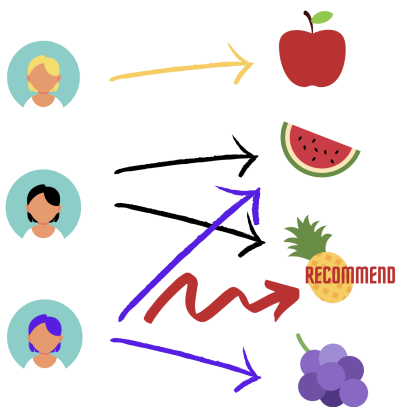


Fig. 1 Explicación gráfica del funcionamiento del algoritmo [3]

2) *Fase de Desarrollo.* La fase de desarrollo comenzó con la configuración del entorno de desarrollo, incluyendo la instalación de dependencias y la configuración de los entornos de desarrollo y prueba. Se identificaron y se instalaron todas las dependencias necesarias para el proyecto. Se preparó un archivo de requisitos que lista todas las dependencias, facilitando la instalación y configuración mediante herramientas de gestión de paquetes.

Una vez instaladas las dependencias, se configura el servidor de desarrollo de Django. Esto incluye la creación de una base de datos de prueba la cual fue postgresql y la configuración de las rutas necesarias para que la aplicación funcione localmente. También se establecen las directrices para el control de versiones utilizando Git y Github, lo que permitió mantener un historial detallado de los cambios realizados en el código.

El desarrollo del backend se centró en la implementación de la lógica del proyecto y la configuración de la base de datos utilizando Django. Primero, se crearon los modelos de datos que representan las entidades clave del sistema, como usuarios, empleadores, solicitantes, ofertas de trabajo y aplicaciones. Estos modelos se traducen en tablas en la base de datos, con relaciones definidas para asegurar la integridad de los datos.

Luego, se implementan las vistas y los serializadores necesarios para manejar las solicitudes HTTP y convertir las consultas de la base de datos en respuestas JSON. Se utiliza Django Rest Framework para facilitar la creación de APIs RESTful, que permiten la comunicación entre el frontend y el backend de manera eficiente. Las vistas manejan las solicitudes entrantes, realizan las operaciones necesarias y devuelven las respuestas apropiadas.

Finalmente, se configuraron las rutas del sistema, definiendo las URL que permiten acceder a las diferentes vistas y recursos de la aplicación. Esta configuración asegura que los usuarios puedan interactuar con la aplicación de manera intuitiva y acceder a las funcionalidades ofrecidas.

El desarrollo del frontend se centró en la creación de interfaces de usuario interactivas y responsivas. Se diseñaron plantillas HTML para las diversas páginas de la aplicación, asegurando que sean fáciles de usar y visualmente atractivas además de que tenga un diseño inclusivo para las computadoras de escritorio y dispositivos móviles. El uso de CSS ayudó a estilizar las páginas, mejorando la experiencia del usuario y asegurando que la aplicación sea accesible desde diferentes dispositivos y tamaños de pantalla.

Se desarrollaron formularios y componentes interactivos que permiten a los usuarios registrarse, iniciar sesión, crear y gestionar ofertas de trabajo, y aplicar a las mismas. Estos componentes interactúan con el backend a través de las APIs RESTful, enviando y recibiendo datos en formato JSON. Se asegura que la comunicación entre el frontend y el backend sea fluida y segura, proporcionando una experiencia de usuario coherente y confiable.

En el proyecto, se implementó la biblioteca Faker para generar datos ficticios con el objetivo de entrenar el algoritmo colaborativo de manera extensa y efectiva. Faker proporcionó una amplia gama de datos simulados, incluyendo perfiles de usuarios, ofertas de trabajo y solicitudes de empleo, que se utilizaron para crear un conjunto de datos representativo y diversificado. Esta estrategia permitió probar el algoritmo en una variedad de escenarios y optimizar su rendimiento

antes de aplicarlo a datos reales, asegurando así resultados más precisos y confiables en la recomendación de ofertas de trabajo.

El algoritmo implementado se basa en la similitud entre usuarios para recomendar ofertas de trabajo. Utiliza datos históricos de solicitudes de empleo para calcular la similitud entre usuarios y generar recomendaciones personalizadas.

Además, para implementar este algoritmo de recomendación basado en la similitud entre usuarios, se hizo uso extensivo de diversas bibliotecas y herramientas especializadas en análisis de datos y aprendizaje automático. Entre estas herramientas se incluyen bibliotecas populares como pandas, numpy y scikit-learn en el ecosistema de Python.

Pandas proporcionó funcionalidades clave para el preprocesamiento y manipulación de datos, permitiendo la carga y manipulación eficiente de los datos históricos de las solicitudes de empleo. Numpy, por otro lado, se utilizó para realizar cálculos numéricos y operaciones matriciales necesarias para calcular la similitud entre usuarios.

Scikit-learn, una biblioteca de aprendizaje automático en Python, fue esencial para implementar el cálculo de similitud entre usuarios utilizando algoritmos como la similitud coseno. Esta biblioteca ofrece una amplia variedad de algoritmos de aprendizaje automático y herramientas para el análisis de datos, facilitando la implementación del algoritmo de recomendación y su integración en el sistema.

En la siguiente sección, se explicará en detalle cómo funciona el algoritmo, desde el preprocesamiento de datos hasta la generación de recomendaciones.

- **Preprocesamiento de Datos:** Primero, se agruparon las solicitudes de empleo para calcular el número total de solicitudes recibidas por cada oferta de trabajo. Luego, esta información se combinó con la tabla de ofertas de trabajo, proporcionando una vista integral del número de aplicaciones por cada oferta específica.
- **Creación de la Matriz de Solicitudes:** Posteriormente, se combinaron los datos de solicitudes de empleo con las ofertas de trabajo para obtener los datos relevantes. Esta combinación permitió crear una matriz donde las filas representan a los candidatos y las columnas a las ofertas de trabajo, con valores que indican si una solicitud fue realizada.
- **Cálculo de Similitud entre Usuarios:** Para identificar usuarios con comportamientos similares, se utilizó la correlación de Pearson entre las aplicaciones realizadas por los candidatos. Además, se calculó la similitud coseno entre los usuarios, lo que permite medir cómo de similares son en términos de las ofertas a las que han aplicado.
- **Selección de Usuarios Similares:** Con la similitud calculada, se seleccionaron los usuarios más similares a un usuario específico (el usuario seleccionado para la recomendación) que superan un cierto umbral de similitud. Esta selección ayuda a enfocar la recomendación en usuarios con comportamientos de solicitud de empleo similares.

- **Filtrado de Ofertas de Trabajo Luego:** Se eliminaron las ofertas de trabajo a las que el usuario seleccionado ya había aplicado. A continuación, se extrajeron las aplicaciones realizadas por los usuarios similares, excluyendo aquellas que el usuario seleccionado ya había considerado.
- **Cálculo de Puntuaciones de Ofertas de Trabajo:** Para cada oferta de trabajo, se calculó una puntuación basada en las aplicaciones de los usuarios similares. La puntuación se determinó sumando la similitud del usuario con cada usuario similar, ponderada por si estos usuarios habían aplicado a la oferta de trabajo. Esto permitió generar una lista de ofertas ordenadas por relevancia.

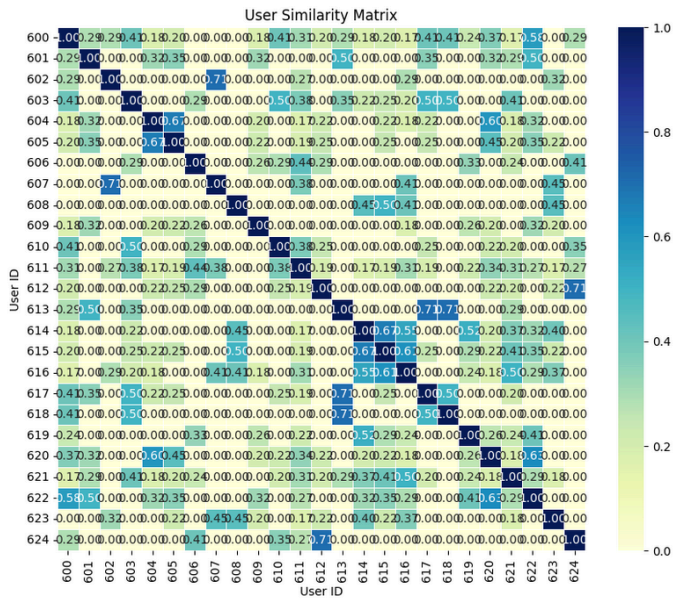


Fig. 2 Resultados del algoritmo dentro de una matriz de similitud de usuario.

Posteriormente, se integró el algoritmo en el backend de Django y se realizaron pruebas exhaustivas para asegurar su precisión y eficiencia en la recomendación de empleos.

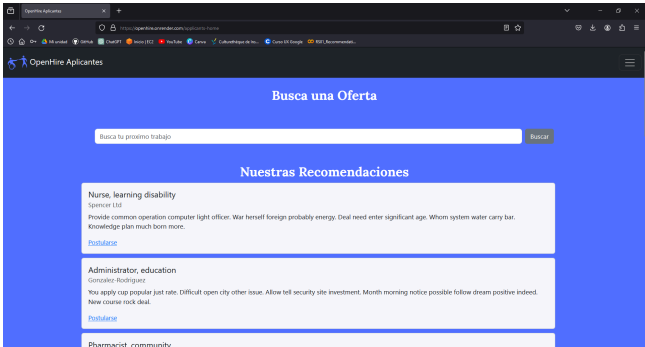


Fig. 3 Algoritmo aplicado ya al proyecto.

3) *Fase de Despliegue.* En la fase de despliegue, se preparó el entorno de producción y se realizaron pruebas finales para asegurar que todo funcionara correctamente. Después configuré una cuenta en AWS y accedí al servicio S3. Desde allí, procedí a crear un nuevo bucket, proporcionando un nombre único y seleccionando la región adecuada para su ubicación.

Una vez creado el bucket, configuré las políticas de acceso y permisos según los requisitos de seguridad del proyecto. Establecí permisos de acceso público o restringir el acceso solo a usuarios autorizados.

Por último, ingrese las credenciales del bucket y del usuario creado en IAM dentro de variables globales secretas que no se encuentran dentro del repositorio de Github por razones de seguridad para poder subir los archivos estáticos al bucket de Amazon.

Se lanzó el portal OpenHire en un entorno de producción utilizando Render junto con la base de datos que fue creada por la misma plataforma. Los datos de la base de datos que fue utilizado en el entorno de prueba fue migrada a la de producción asegurando que hubiera integridad y coherencia. Posteriormente se creó un servicio web en Render clonando el repositorio del proyecto al servidor, declarando las variables globales que se mantendrán encriptadas y poder desplegarlo, asegurando la estabilidad y escalabilidad del servicio.

El sistema se desplegó en un servidor de producción, asegurando que estuviera accesible para los usuarios finales y cumpliendo con los requisitos de rendimiento y seguridad.

Durante la fase de pruebas del proyecto, se llevaron a cabo diversas actividades para garantizar el funcionamiento correcto y la calidad del sistema desarrollado. Estas pruebas se realizaron en diferentes niveles, desde pruebas unitarias hasta pruebas de integración y aceptación.

En primer lugar, se realizaron pruebas unitarias para validar el funcionamiento individual de cada componente del sistema. Se hizo observación a casos de prueba que cubrían todas las funcionalidades y se ejecutaron para identificar posibles errores y asegurar que cada unidad funcionará según lo esperado.

Posteriormente, se llevaron a cabo pruebas de integración para verificar la interacción adecuada entre los diferentes módulos del sistema. Se probaron los flujos de datos y la comunicación entre los componentes para asegurarse de que no hubiera problemas internos del sistema.

Una vez completadas las pruebas de integración, se realizaron pruebas de sistema para evaluar el sistema en su conjunto. Se simuló escenarios de uso real y se evaluó el rendimiento, la estabilidad y la usabilidad del sistema en diferentes condiciones.

Basado en los comentarios de los usuarios y los resultados de las pruebas, se ajustaron y optimizaron las recomendaciones del algoritmo para mejorar la satisfacción y éxito de los usuarios en encontrar empleo.

## E. *Módulo I Gestión de la tecnología de información*

En el proyecto modular, la gestión de la tecnología de la información se abordó utilizando la metodología de cascada para la gestión del proyecto. Este enfoque permitió una planificación y ejecución estructurada, garantizando que cada fase del desarrollo se completará antes de pasar a la siguiente. Se utilizó GitHub para el versionamiento del código, facilitando la gestión y el seguimiento de cambios en el desarrollo del proyecto. El proyecto se desarrolló full stack, implementando tanto el frontend como el backend y siguiendo buenas prácticas de programación, incluyendo la escritura de código limpio y documentado. Además, se diseñó la aplicación de manera inclusiva, asegurando su compatibilidad y funcionalidad tanto en computadoras de escritorio como en dispositivos móviles.

## F. *Módulo II Sistemas robustos, paralelos y distribuidos*

El proyecto modular se centró en diseñar y desarrollar un sistema robusto y distribuido que pudiera manejar cargas de trabajo variables y simultáneas. Se utilizó Amazon S3 de AWS para el almacenamiento de archivos estáticos, lo que proporcionó una solución escalable y fiable para manejar los archivos del proyecto. La base de datos y la aplicación se alojaron en Render, un servicio de hospedaje en la nube que facilitó el despliegue y la gestión del proyecto en un entorno distribuido. Esta infraestructura en la nube permitió garantizar la disponibilidad, escalabilidad y rendimiento del sistema, manejando eficientemente las solicitudes concurrentes de los usuarios.

## G. *Módulo III Cómputo Flexible (Softcomputing)*

El proyecto modular incorporó el desarrollo del algoritmo de recomendación de ofertas de trabajo. Se implementó un enfoque basado en la similitud entre usuarios para generar recomendaciones personalizadas y relevantes. El algoritmo se adaptó de manera flexible a los cambios en los datos y en las preferencias de los usuarios, lo que permitió proporcionar recomendaciones dinámicas y precisas. Además, se realizaron pruebas exhaustivas para validar la efectividad y la eficacia del algoritmo en la generación de recomendaciones relevantes para los usuarios.

## IV. RESULTADOS OBTENIDOS DEL PROYECTO

El algoritmo de recomendación produjo una lista de ofertas de trabajo ordenadas por su relevancia para el usuario seleccionado. Los resultados se detallan a continuación:

- **Personalización de Recomendaciones:** Las recomendaciones son personalizadas para el usuario, basadas en las aplicaciones de otros usuarios con comportamientos similares. Esto asegura que las ofertas de trabajo sugeridas sean altamente relevantes para el candidato.
- **Aumento de la Eficiencia en la Búsqueda de Empleo:** Al proporcionar recomendaciones específicas, se reduce el tiempo que los candidatos necesitan invertir en buscar ofertas de trabajo. Esto hace que el proceso de búsqueda de empleo sea más eficiente y dirigido.
- **Diversidad en las Recomendaciones:** Aunque las recomendaciones están basadas en similitudes, el algoritmo

asegura que se presenten nuevas oportunidades que el usuario no había considerado previamente. Esto puede abrir nuevas posibilidades y perspectivas laborales para el candidato.

- **Mejora en la Satisfacción del Usuario:** Al recibir recomendaciones más pertinentes y personalizadas, se espera que los usuarios tengan una mejor experiencia en la plataforma, aumentando su satisfacción y posiblemente su tasa de conversión de solicitudes a empleos.
- **Impacto en las Ofertas de Trabajo:** Las empresas que publican ofertas de trabajo también se benefician de este sistema de recomendación, ya que sus vacantes son mostradas a candidatos que tienen una alta probabilidad de estar interesados y calificados para las posiciones, mejorando la calidad de los candidatos que aplican.

## V. CONCLUSIONES Y TRABAJO A FUTURO

El diseño y la fase de prototipado del proyecto proporcionaron una base sólida para la creación de una aplicación web de recomendación de ofertas de trabajo. La arquitectura del sistema permite una escalabilidad y un mantenimiento eficientes, mientras que el desarrollo full stack y la integración del algoritmo de recomendación garantizan una experiencia de usuario personalizada y relevante. El despliegue exitoso a la web marcó el hito final en la fase de prototipado, preparando el terreno para futuras iteraciones y mejoras en el proyecto.

Durante la exposición del proyecto hubo aceptación del concepto del proyecto, que fue un hito crucial que reflejó la comprensión y el respaldo de los interesados, así como el reconocimiento del valor y la viabilidad del proyecto en su conjunto. Esta aceptación no sólo validó la relevancia y la importancia del problema que el proyecto buscaba abordar.

En primer lugar, la aceptación del concepto del proyecto implicó una comunicación efectiva y una alineación de expectativas entre todos los interesados. Esto se logró mediante la presentación clara y convincente del concepto del proyecto, destacando sus objetivos, beneficios y el valor agregado que ofrecería a los usuarios finales y a la organización en general.

Además, la aceptación del concepto del proyecto se basó en una evaluación exhaustiva de su viabilidad técnica y social. Esto implicó un análisis detallado de los requisitos y restricciones del proyecto, así como una evaluación de los recursos necesarios y los riesgos potenciales asociados. La identificación y mitigación proactiva de los riesgos fueron aspectos clave que inspiraron confianza en la capacidad del proyecto para lograr sus objetivos de manera efectiva y oportuna.

Otro factor importante que contribuyó a la aceptación del concepto del proyecto fue la demostración de su valor y beneficios potenciales para las partes interesadas involucradas. Esto se logró mediante la presentación de casos de uso y escenarios de aplicación práctica, que ilustraron cómo el proyecto abordará las necesidades y desafíos específicos de la problemática y que proporcione soluciones tangibles y medibles.

Finalmente, la aceptación del concepto del proyecto se tradujo en un compromiso activo y un apoyo continuo por parte de los interesados, que respaldaron el proyecto en base a sus opiniones.

### A. Evolución previsible del sistema

- **Integración con Redes Sociales:** Evaluar la posibilidad de integrar la plataforma con redes sociales populares para facilitar la divulgación de ofertas de empleo y la inscripción de candidatos.
- **Funciones de Análisis de Datos:** Implementar herramientas de análisis de datos para ayudar a los empleadores a comprender mejor las tendencias en la contratación y a los candidatos a evaluar su idoneidad para ciertas ofertas de empleo.
- **Aplicación Móvil:** Considerar el desarrollo de una aplicación móvil complementaria que permita a los usuarios acceder a la plataforma de manera más conveniente desde dispositivos móviles.
- **Mejoras en la interfaz de usuario:** Continuar optimizando la interfaz de usuario para hacerla más intuitiva y atractiva, teniendo en cuenta las tendencias actuales de diseño y usabilidad.
- **Chat en Vivo y Soporte:** Implementar una función de chat en vivo para proporcionar soporte en tiempo real a los usuarios y resolver preguntas de manera eficiente.
- **Traducción y Localización:** Ofrecer soporte para múltiples idiomas y considerar la localización de la plataforma para adaptarse a diferentes regiones geográficas.
- **Programas de Capacitación:** Diseñar y ofrecer programas de capacitación en línea para empleadores y candidatos sobre cómo utilizar eficazmente la plataforma.
- **Mejoras en la Seguridad:** Mantenerse al tanto de las mejores prácticas de seguridad informática y realizar mejoras continuas para proteger la privacidad de los usuarios y los datos.
- **Evaluación de Impacto Social:** Realizar evaluaciones periódicas del impacto social de la plataforma en la inclusión laboral y considerar iniciativas para medir y mejorar este impacto.

## RECONOCIMIENTOS

Quiero expresar mi más sincero agradecimiento al asesor Víctor Manuel Zamora Ramos por su invaluable apoyo y orientación en el desarrollo de este proyecto. Su expertise y dedicación fueron fundamentales para la elaboración de la documentación y el éxito del proyecto en general. Agradecemos profundamente su disponibilidad y el tiempo que nos dedicó, siempre dispuesto a ofrecer su asesoría y guía en cada etapa del proceso.

## REFERENCIAS

- [1] Solís, A. (2016, March 2). 7 de cada 10 personas con discapacidad, sin trabajo. Forbes México. Retrieved February 14, 2024, from <https://www.forbes.com.mx/7-de-cada-10-personas-con-discapacidad-sin-trabajo/>
- [2] Saavedra, J. (2017, October 24). Sistemas de recomendación, parte 2: filtrado colaborativo | by Juan Saavedra. Medium. Retrieved February 14, 2024, from <https://medium.com/@eng.saavedra/sistemas-de-recomendaci%C3%B3n-parte-2-b8a5dc9dc730>
- [3] Grab N Go Info. (2022, January 22). User-Based Collaborative Filtering In Python | Machine Learning. YouTube. Retrieved May 15, 2024, from <https://www.youtube.com/watch?v=cxcFi3RDrEw>
- [4] PyConES-2019-data/files/sistemas-de-recomendacion-con-surprise/pycones-surprise-algorithms.ipynb at master · python-spain/PyConES-2019-data. (n.d.). GitHub. Retrieved February 14, 2024, from <https://github.com/python-spain/PyConES-2019-data/blob/master/files/sistemas-de-recomendacion-con-surprise/pycones-surprise-algorithms.ipynb>
- [5] Glass, E. (2020, August 19). Cómo configurar Django con Postgres, Nginx y Unicorn en Ubuntu 20.04. DigitalOcean. Retrieved February 14, 2024, from <https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-unicorn-on-ubuntu-20-04-es>
- [6] Cómo subir tu proyecto de Django a AWS. (n.d.). Platzi. Retrieved February 14, 2024, from <https://platzi.com/blog/django-aws/>