



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**GII 21.13 Sistema de
recomendación de tecnologías
para desarrolladores basado en
la encuesta de Stack Overflow**



Presentado por Enrique Camarero Alonso
en Universidad de Burgos
13 de junio de 2022
Tutores: José Ignacio Santos Martín
Virginia Ahedo García



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. José Ignacio Santos Martín, profesor del departamento de Ingeniería de Organización, área de Organización de Empresas. Y D.^a Virginia Ahedo García, profesora del departamento de Ingeniería de Organización, área de Organización de Empresas.

Expone:

Que el alumno D. Enrique Camarero Alonso, con DNI 71307018-A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado sistema de recomendación de tecnologías para desarrolladores basado en la encuesta de Stack Overflow.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección de los que suscriben, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de junio de 2022

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

El objetivo de este TFG es diseñar e implementar un sistema de recomendación de tecnologías para desarrolladores (lenguaje de programación, base de datos, plataformas, etc.) utilizando la información recogidas en la encuesta que Stack Overflow realiza todos los años entre desarrolladores. Los datos de esta encuesta, que se encuentran disponibles bajo licencia ODbL son accesibles desde [su propia página](#).

Descriptores

Sistemas de Recomendación, Stack Overflow, Teoría de Redes, Python, HTML, CSS, Heroku, Aplicación web, SCRUM, Metodología ágil, Trabajo de Fin de Grado (TFG), frontend, GitHub

Abstract

The objective of this FDP is to design and implement a technology recommendation system for developers (programming language, database, platforms, etc.) using the information collected in the survey that Stack Overflow conducts every year among developers. which is available under the ODbL license available under ODbL license are accessible from [its own page](#)..

Keywords

Recommendation Systems, Stack Overflow, Network Theory, Python, HTML, CSS, Heroku, Web Application, SCRUM, Agile Methodology, Final Degree Project (FDP), frontend, GitHub

Índice general

| | |
|---|-----|
| Índice general | iii |
| Índice de figuras | v |
| Índice de tablas | vi |
| Introducción | 1 |
| Objetivos del proyecto | 3 |
| 2.1. Objetivos generales | 3 |
| 2.2. Objetivos de carácter técnico | 3 |
| 2.3. Objetivos de carácter personal | 4 |
| Conceptos teóricos | 5 |
| 3.1. Sistema de recomendación | 5 |
| 3.2. Teoría de redes | 6 |
| 3.3. Frontend | 9 |
| 3.4. Framework | 10 |
| 3.5. Metodologías ágiles | 10 |
| 3.6. SCRUM | 10 |
| 3.7. Dataframe | 11 |
| Técnicas y herramientas | 13 |
| 4.1. Lenguajes de programación | 13 |
| 4.2. Entornos de desarrollo | 14 |
| 4.3. Herramientas de documentación | 15 |
| 4.4. Plataformas | 15 |

| | |
|---|-----------|
| 4.5. Forma de presentación | 16 |
| Aspectos relevantes del desarrollo del proyecto | 19 |
| 5.1. Primeros pasos | 19 |
| 5.2. Primeros objetivos | 20 |
| 5.3. Creación frontend | 21 |
| 5.4. Creación opción de cambio de umbral | 22 |
| 5.5. Creación del sistema de recomendación de lenguajes | 23 |
| 5.6. Heroku | 24 |
| 5.7. Creación de las nuevas páginas con las tecnologías restantes | 24 |
| 5.8. Mejoras del código | 25 |
| Trabajos relacionados | 27 |
| Conclusiones y Líneas de trabajo futuras | 29 |
| 7.1. Conclusiones | 29 |
| 7.2. Líneas de trabajo futuras | 29 |
| Bibliografía | 31 |

Índice de figuras

Índice de tablas

Introducción

El proyecto se basa en diseñar e implementar en una aplicación web, un sistema de recomendación de tecnologías para desarrolladores (lenguajes de programación, bases de datos, plataformas, etc.) a partir de la información proporcionada en la encuesta que Stack Overflow realiza todos los años entre desarrolladores, la cual podemos [descargar](#) y [consultar los resultados](#) en su página oficial.

Para su realización, de entre los distintos sistemas de recomendación que se pueden emplear (filtros colaborativos basados en usuarios, en productos, etc.), se ha optado por el enfoque de redes.

La encuesta está formada por las respuestas a preguntas de diversos tipos con el fin de mejorar la comunidad y la plataforma que los atiende. Los datos se dividen en 5 bloques:

- *Perfil de los desarrolladores*: está formado por la media de la localización geográfica, años de experiencia de trabajo, tipo de trabajo, educación y edad de los encuestados.
- *Tecnologías*: compuesto por los lenguajes, entornos de desarrollo, marcos de trabajo, bases de datos, herramientas y plataformas utilizadas por los encuestados en el último año y las tecnologías que quieren utilizar en el siguiente año.
- *Trabajo*: este bloque se basa en conocer información relacionada con el trabajo de cada encuestado como si es a tiempo parcial o completo, si es estudiante, autónomo, jubilado, etc. También se pregunta acerca del salario de cada uno y el tamaño de las empresas en las que trabajan.

- *Comunidad*: es el apartado más importante. Se basa en conocer el uso que le da cada usuario a Stack Overflow, la participación en la página, las visitas diarias, semanales y mensuales aproximadas a la plataforma, si se sienten parte de Stack Overflow, entre otras preguntas.
- *Metodología*: es el resumen y la valoración que dan los encargados de la encuesta a los resultados obtenidos.

De estos apartados para el proyecto se ha utilizado el de tecnologías.

Este proyecto en términos de visualización y presentación de los datos, contiene una red en la que las tecnologías representan los nodos y los enlaces el número de desarrolladores que las utilizan simultáneamente, un histograma que muestra las propiedades de la red y un gráfico de sectores creado a partir del porcentaje de relaciones de cada nodo.

En cada pestaña se han añadido elementos de interacción para el usuario como la posibilidad de cambiar el umbral de poda del grafo, para que cada usuario adapte la recomendación de tecnologías a su gusto, la búsqueda manual de las tecnologías en las distintas tablas ofrecidas y un grafo que ofrece información al posicionarse en los distintos nodos.

Para la creación de la aplicación web se ha desplegado en la plataforma Heroku el código Python junto a su correspondiente frontend, el cual está formado por códigos en Python, HTML, CSS y el material necesario para la presentación, como imágenes, grafos y tablas proporcionados a partir de la ejecución del código.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos generales, los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto y los objetivos de carácter personal.

2.1. Objetivos generales

- Implementación de un sistema de recomendación basado en la encuesta de Stack Overflow.
- Emisión de recomendaciones de: bases de datos, entornos de desarrollo, herramientas, lenguajes, marcos de trabajo y plataformas.
- Creación de una aplicación web que pueda ser utilizada libremente por cualquier usuario sin necesidad de descargarse nada.

2.2. Objetivos de carácter técnico

- Visualización: diseñar una aplicación visualmente atractiva y fácil de comprender.
- Online: ejecutar la aplicación en un servidor web para que pueda ser utilizada por cualquier usuario sin necesidad de descargar nada.
- Interacción: permitir a los usuarios interaccionar con la aplicación para así mejorar la experiencia de éste.

- Renovación: la aplicación permite fácilmente actualizarse anualmente para tener los datos más recientes.
- Disponibilidad: permite acceder a la aplicación en cualquier momento siempre que se tenga una conexión a internet.
- Usabilidad: la aplicación tiene una interfaz muy clara y sencilla para permitir a cualquier usuario utilizarla y entenderla sin ningún problema.
- Asequibilidad: es gratuita, por lo que no será necesario pagar para poder utilizarla.
- SCRUM: metodología que controla la planificación del proyecto.

2.3. Objetivos de carácter personal

- Aprender a desplegar aplicaciones en Heroku ya que hasta este momento no había utilizado esta herramienta.
- Aumentar mis conocimientos sobre creación de aplicaciones web creadas a partir de HTML y CSS.
- Aprender sobre la Teoría de Redes.

Conceptos teóricos

Se van a explicar una serie de conceptos los cuales ayudarán a los usuarios a comprender mejor el proyecto.

3.1. Sistema de recomendación

Un sistema de recomendación [10] es una herramienta que ofrece al usuario recomendaciones sobre algún tipo de información (en este proyecto tecnologías de desarrollo) construidas como predicciones basadas generalmente en los datos ofrecidos por otros usuarios.

Tipos de sistemas de recomendación [6]

- *Sistemas de popularidad*: se basan en tomar como referencia el prestigio de un objeto a estudiar por una variable principal, la cual puede ser cualquier característica del objeto. Las recomendaciones son las mismas para todos los usuarios, ya que se trata de recomendar aquellos productos que tienen más visitas, mejores valoraciones, más compras, etc.
- *Sistemas de contenido*: se basan en sugerir a los usuarios productos acordes a sus gustos y sus búsquedas. Esto es posible tras encontrar patrones al analizar las características de los usuarios y el contenido de los productos. Este tipo de recomendación no tiene en cuenta las opiniones y valoraciones que otros usuarios puedan tener sobre el producto. Además, al crear perfiles para los distintos usuarios no se va a producir ningún problema con los nuevos productos, ya que tan solo se deberá comprobar si encaja con el perfil del usuario para de esta forma poder recomendárselo. Por otro lado, así como con los nuevos productos no hay problemas, sí se

produce si se crea un nuevo usuario, ya que al no tener un perfil definido no se le podrán realizar recomendaciones.

- *Filtrado colaborativo*: se basan en identificar y agrupar perfiles similares dependiendo de la información del perfil del usuario. Resuelve los problemas que puede crear el uso de información en grandes cantidades a los usuarios. Se incrementa la calidad de las recomendaciones. Pueden ser de 2 tipos:
 - Basados en memoria: utilizan la información para definir similitudes entre usuarios y productos que serán utilizados para construir las recomendaciones. Si estas similitudes se realizan entre usuarios, se van a hacer recomendaciones a partir de los gustos y compras de otros usuarios con perfiles similares, mientras que si las similitudes se realizan entre productos, las recomendaciones se realizarán entre productos similares a los previamente buscados. Estas técnicas de recomendación se llaman técnicas de los k-vecinos.
 - Basados en modelos: utilizan la información para ajustar modelos que permitan en un futuro proponer recomendaciones. Esto se realiza creando distintos modelos a través de algoritmos de *machine learning*.
- *Basados en redes*: se apoya en la teoría de redes [16]. Podemos encontrar 3 tipos de redes para generar recomendaciones:
 - De productos: los nodos corresponden con los productos y los enlaces muestran la relación que tienen los nodos.
 - De usuarios: los nodos corresponden con los usuarios y los enlaces muestran la relación que tienen los nodos.
 - De productos-usuarios: hay dos tipos de nodos: productos y usuarios. Los enlaces muestran que un usuario se ha relacionado con ese producto, ya sea porque lo ha buscado, visitado, comprado, etc.

En este tipo de sistemas de recomendación se pueden hacer recomendaciones de múltiples maneras como pueden ser los vecinos más cercanos o en comunidades en las que se encuentran los nodos.

3.2. Teoría de redes

Esta teoría se origina en el siglo XVIII con el problema de los puentes de Königsberg, en donde se quería demostrar que no era posible cruzar los 7 puentes de la ciudad de Prusia sin cruzar uno de ellos en 2 ocasiones.

La teoría de redes [25] se basa en el estudio de redes complejas, las cuales se encuentran formadas por nodos y se relacionan a través de enlaces.

Los nodos pueden ser de una única clase (grafos unimodales), de dos clases (grafos bimodales), o de muchas clases (grafos multimodales).

Los enlaces pueden formar redes dirigidas o no dirigidas, dependiendo si estos tienen dirección o no. También pueden ser binarias (solo se informa de la existencia del enlace) o pesadas (el enlace tiene un valor) y pueden formar ciclos (red cíclica).

Dado una red binaria (N,E) formada por un conjunto de nodos N y un conjunto de enlaces E, se define la matriz nxn de adyacencia como el conjunto de 1 o 0, siendo 1 si hay un enlace entre los nodos i,j y 0 sino.

Propiedades de la red

Existen múltiples métricas con las que se pueden describir las redes. Entre todas estas las más importantes son las siguientes:

- *Tamaño*: se calcula como el número de nodos (N) o de enlaces (E) que contiene la red. Si la red contiene menos de 10 nodos se considera como una red pequeña, mientras que si tiene más de 1000 nodos se considera una red grande.
- *Densidad*: se calcula como el número de enlaces que tiene la red sobre el número máximo de enlaces que pueden existir para el número de nodos que forman la red. Se dice que la red es poco densa (la matriz de adyacencia es dispersa) si el número de enlaces E es del mismo orden que el número de nodos N ($E=N$). Por el contrario si $E \gg N$ se dice que es densa.
- *Grado*: en redes no dirigidas, se conoce como el grado de un nodo al número de enlaces que este tiene con otros nodos. En redes dirigidas el grado se puede calcular como el número de enlaces que entran en el nodo (*in-degree*), el número de enlaces que salen del nodo (*out-degree*) o como la suma de ambos (*total-degree*).
- *Coeficiente de clustering*: es la probabilidad que tienen los nodos que se enlazan con otro de conectarse entre ellos.

$$C_i = \frac{E_i}{k_i(k_i-1)/2}$$

- *Distribución de grado*: se refiere a la probabilidad de que un al escoger aleatoriamente un nodo N , este tenga k enlaces.

$$p_k = N_k/N$$

Detección de comunidades

Una comunidad de nodos está formada por grupos de nodos densamente conectados entre sí. Para detectar todas estas comunidades, las cuales ayudan a identificar y describir la estructura de la red, existen diferentes algoritmos:

- *Girvan-Newman* [23]: detecta las distintas comunidades tras ir eliminando los enlaces que forman la red original. Esto se realiza siguiendo los siguientes pasos:
 1. Lo primero que se calcula es la intermediación de todos los bordes existentes en la red.
 2. Se eliminan los bordes con la mayor distinción.
 3. Se recalcula la intermediación de todos los bordes afectados por la eliminación.
 4. Los pasos 2 y 3 se repiten hasta que no quedan bordes.
- *Louvain* [11]: estudia y evalúa los conjuntos de datos para determinar la relación de contraste con la densidad de las aristas que se encuentra en el conjunto de datos, tanto en el interior como en el exterior.
- *Leiden* [5]: este algoritmo mejora el de Louvain, ya que puede detectar comunidades mal conectadas. Para garantizar esto, este algoritmo se basa en el algoritmo de movimiento local inteligente. El algoritmo se ejecuta siguiendo los siguientes pasos:
 1. Movimiento local de los nodos.
 2. Depuración de la partición.
 3. Adición de la red basada en la depuración del paso anterior, usando la partición no depurada para crear una partición inicial para la red añadida.
- *Kernighan-Lin* [15]: trata de mejorar una partición inicial utilizando otros métodos como el de la bisección. Se trata de optimizar la distinción de los enlaces dentro de las particiones y entre particiones.

k-vecinos más próximos

Este algoritmo [14] de clasificación es uno de los más básicos y esenciales que existen. A pesar de que existan muchos otros algoritmos más complejos como los basados en comunidades, este ha sido el elegido para la realización de las recomendaciones que se realizan en el proyecto.

Este método tiene como objetivo clasificar los nodos dependiendo de la proximidad con el resto o predecir sobre la agrupación de los nodos.

Por ello, este algoritmo ha sido utilizado en la realización del proyecto, ya que para recomendar tecnologías se va a tener en cuenta la proximidad que tienen unas tecnologías con otras, así como el número de enlaces que tiene cada una.

Este algoritmo funciona de manera que:

1. Se calcula la distancia que tiene el nodo escogido con los nodos a recomendar. Todos ellos forman la red.
2. Se escogen las tecnologías que tengan distancia igual a 1 con nodo escogido.
3. Entre las tecnologías escogidas, se recomiendan las 3 que posean mayor número de enlaces.

El valor de k va a tomar un valor muy importante en el algoritmo, ya que dependiendo de este un nodo puede ser escogido por una clase u otra.

3.3. Frontend

Front End [19] es la parte de una aplicación, conocida como el lado de los clientes, ya que es la que interactúa con los usuarios. Podría decirse que es aquello que somos capaces de ver en la pantalla al acceder a una aplicación o sitio web. Todo este conjunto crea la experiencia del usuario.

Los principales lenguajes que se utilizan son:

- HTML
- CSS
- JavaScript

Los dos primeros son lenguajes de etiquetas, mientras que JavaScript es un lenguaje de programación.

3.4. Framework

Un framework o marco de trabajo [8] es una estructura a modo de plantilla, que sirve de punto de partida a los desarrolladores para elaborar un proyecto con ciertos objetivos y simplificar la elaboración de éste.

El uso de frameworks pueden ayudar a los programadores simplificando las tareas y sus procesos, acelerando el trabajo, reduciendo fallos y obteniendo mejores resultados.

3.5. Metodologías ágiles

Las metodologías ágiles [18] son aquellas que adaptan los proyectos a sus condiciones y las del entorno consiguiendo flexibilidad de esta forma.

Ventajas de la utilización de metodologías ágiles en los proyectos:

- Mayor calidad del proyecto.
- Mejora la satisfacción del cliente.
- Trabajo cooperativo.
- Aumento de la productividad.
- Mejora el control del proyecto.
- Menores costes.

3.6. SCRUM

SCRUM [1] se trata de un framework utilizado dentro de los proyectos que tratan una metodología ágil. Su finalidad es la de realizar a través de *sprints*¹ entregas parciales del producto final para así agilizar el proceso de entrega del proyecto.

Eventos de SCRUM:

- *Planificación del sprint*: determina las tareas que se desarrollaran a lo largo del sprint. Además, se asigna cada tarea a una persona.

¹Intervalo de tiempo corto en el que se lleva a cabo una parte del proyecto para incrementar el valor del producto el cual se está construyendo. [17]

- *Seguimiento del sprint*: a través de una reunión cada persona explica el trabajo que está realizando y si necesita algo para poder llevarla a cabo.
- *Revisión del sprint*: se muestra el avance que se ha conseguido tras la realización del sprint a todas las personas implicadas en el proyecto.

3.7. Dataframe

Un DataFrame [7] es una estructura de datos similar a una hoja de cálculo, en la cual podemos guardar información de distintos tipos en filas y columnas. Todas las filas del dataframe están numeradas empezando en 0.

Técnicas y herramientas

Se van a explicar una serie de técnicas y herramientas las cuales han sido utilizadas para realizar el proyecto y el por qué se han escogido.

4.1. Lenguajes de programación

Python

Python [21] es un lenguaje de programación de alto nivel el cual no necesita de proceso de compilación para ejecutar las aplicaciones puesto que se ejecutan directamente utilizando un programa denominado interpretador.

Es el lenguaje en el que se ha programado el proyecto.

Se ha elegido Python porque es el lenguaje con el que me siento más cómodo y familiarizado. Además, Python ofrece diferentes librerías para la programación de redes y sistemas de recomendación con las cuales he trabajado en diferentes asignaturas del grado, lo cual me ha permitido aplicar todos mis conocimientos acerca del uso de estas librerías para el desarrollo del proyecto.

HTML

El lenguaje de marcado de hipertexto (HTML) [3] es un lenguaje con el que se codifican las aplicaciones y las páginas web.

Es el lenguaje que se ha utilizado para crear la aplicación web en el proyecto.

A pesar de no tener mucho conocimiento sobre este lenguaje, he trabajado con HTML en alguna ocasión, lo que me ha permitido aprender más rápido el lenguaje y su uso. El haber trabajado con el lenguaje sumado al potencial que

ofrece HTML a la hora de crear páginas y aplicaciones web es lo que me ha llevado a elegir el uso de este lenguaje.

CSS

CSS [22] es el lenguaje que usamos para diseñar un documento HTML.

Es el editor de estilos que se ha utilizado para diseñar las distintas pestañas del proyecto.

Al igual que con HTML, he trabajado con CSS en alguna ocasión, ya que normalmente se suele utilizar la combinación de ambos lenguajes para la creación de páginas web. He elegido este lenguaje no solo por su anterior uso y la posibilidad de poder diseñar las distintas pestañas del proyecto libremente, sino por su buena comunicación con HTML.

4.2. Entornos de desarrollo

Jupyter Notebook

Jupyter Notebook [12] es una aplicación web interactiva que permite crear documentos en diferentes lenguajes como Julia, Python y R.

En el proyecto se ha utilizado para programar el código Python antes de introducirlo en la aplicación web.

He elegido Jupyter Notebook porque es el entorno más utilizado por mi parte cuando necesito programar en Python. Además, ofrece la posibilidad de guardar los códigos con la extensión .py lo cual es lo que necesitaba para la realización del proyecto.

Visual Studio

Visual Studio [9] es un entorno de desarrollo integrado o IDE (Integrated Development Environment) que proporciona diferentes servicios que facilitan a los desarrolladores la creación de software como aplicaciones web eficaces y de alto rendimiento.

Es el entorno en el que se ha creado la página web a partir de los lenguajes antes mencionados.

Elegí utilizar Visual Studio por ser el entorno de desarrollo que utilicé cuando aprendí a programar páginas web con HTML y CSS.

4.3. Herramientas de documentación

LaTeX

LaTeX [24] es un procesador gratuito de textos el cual se utiliza para la creación de documentos con una alta calidad tipográfica como tesis, artículos o libros científicos.

En el proyecto se ha utilizado para realizar la memoria y los anexos.

La documentación se ha llevado a cabo con este procesador ya que era una de las opciones que se ofrecían junto a OpenOffice para realizar la documentación. Entre estas herramientas escogí LaTeX ya que es la que menos conocía y quería aprender más acerca de ella.

Overleaf

Overleaf [20] es un software que permite redactar, editar y publicar fácilmente textos de manera online. Overleaf contiene un editor de textos en LaTeX fácil de usar y permite en tiempo real ver la salida perfectamente compilada.

Se ha utilizado el editor de textos en LaTeX de Overleaf para realizar la memoria y los anexos.

Se ha escogido este editor ya que es el mejor editor de LaTeX de manera online. Además, permite ver en formato PDF el resultado del documento en tiempo real, lo que permite realizar cambios si el resultado no era de mi agrado.

4.4. Plataformas

GitHub

GitHub [2] es una plataforma de organización y gestión de repositorios gratuito online. Es decir, permite a los usuarios editar, gestionar y guardar sus proyectos en la nube de manera pública o privada. También permite descargar a los usuarios proyectos almacenados en la plataforma que sean públicos.

En este caso, GitHub almacena el proyecto desde su creación y lo gestiona mediante sprints.

Personalmente pienso que GitHub es la mejor plataforma para gestionar proyectos de manera ágil. Es por esta razón que la he escogido.

GitHub Desktop

GitHub Desktop es la versión escritorio de GitHub.

En el proyecto se ha utilizado para ir subiendo al repositorio de GitHub del proyecto todas aquellas actualizaciones que se han ido realizando.

Heroku

Heroku [4] es una Plataforma como Servicio (PaaS) en la nube la cual permite a través de contenedores mantener y ejecutar aplicaciones. Estos contenedores son escalables bajo demanda de los usuarios.

Heroku soporta distintos lenguajes entre los que destacan Clojure, Go, Java, Node.js, PHP, Python, Ruby y Scala.

Esta plataforma es la encargada de almacenar y ejecutar el proyecto de manera online.

A pesar de no tener conocimiento sobre la plataforma al comienzo del proyecto, se escogió ya que además de ser gratuita, permite cargar la aplicación web en sus servidores y que cualquier persona pueda acceder mediante un enlace. Además, Heroku tiene la posibilidad de cargar el código del proyecto directamente desde GitHub lo que facilita su despliegue.

4.5. Forma de presentación

Para la presentación de este proyecto se barajaron 2 opciones diferentes: una aplicación local o web.

Aplicación local

Podía ser de dos maneras diferentes:

- La primera se trataría de un Jupyter Notebook en el cual se crearía una página web programada en HTML al ejecutar el código.
- En la segunda el código Python se ejecutaría junto con código HTML y CSS desde el frontend de una aplicación. Para ver la página web se necesitaría ejecutar la aplicación desde el terminal e ir a la página `http://127.0.0.1:5000`.

Aplicación web

A partir del frontend creado para la aplicación local, el proyecto se subiría a la plataforma Heroku, en donde se podría acceder al proyecto a través de un enlace.

Aplicación local vs Aplicación web

Desde el principio del proyecto se pensó en la realización de una aplicación ejecutándola desde el terminal desarrollada por frontend ya que tanto HTML como CSS permiten un mejor diseño web.

Una vez creada, los tutores propusieron ejecutar el programa en Heroku para que de esta forma, los usuarios tuvieran acceso al proyecto sin necesidad de instalar ningún programa, por lo que me decidí por la aplicación web.

Aspectos relevantes del desarrollo del proyecto

En este apartado se va a comentar el desarrollo del proyecto, el cual ha sido gestionado mediante metodología ágil a través de SCRUM.

5.1. Primeros pasos

Antes de empezar con el proyecto se creó un repositorio en [GitHub](#). De esta forma a través de sprints se iría elaborando el proyecto y se podría realizar un seguimiento del mismo.

La creación del código Python fue lo primero con lo que se empezó en el proyecto. Se comenzó con la creación de un notebook de Jupyter y la visualización del archivo csv, en el que se encuentran los resultados de la encuesta, y la creación de redes sencillas para de esta manera irme familiarizando con los datos con los que se trabaja en este proyecto. De las diferentes columnas que forman la encuesta se decidió usar las siguientes:

- Lenguajes
- Bases de datos
- Plataformas en la nube
- Marcos de trabajo web
- Herramientas
- Entornos de desarrollo

- Otros marcos de trabajo

Aunque la columna de marcos de trabajo y otros marcos de trabajo se juntaron en una sola.

Se empezó por la programación de la página de lenguajes, mostrando diferentes características de los nodos que formaban parte del grafo y creando histogramas.

Se quería crear un grafo interactivo en el que se mostrara los datos de cada nodo en ese instante, pero esto no podía realizarse con las librerías que conocía. Es por ello que se tuvo que buscar y probar diferentes librerías que permitieran mostrar información en tiempo real al posicionarse encima de los nodos. Este problema se solventó al encontrar Bokeh.

Tras informarme sobre ella se creó la red, la cual es un grafo de relaciones que está creado de la siguiente manera:

- Se recorre los resultados de la encuesta.
- En los lenguajes con los que dice haber trabajado se relacionan todos entre ellos de manera que si esa pareja de lenguajes ya está en el grafo se aumenta en 1 su peso, sino se crea la relación con un peso igual a 1.

Además, se crearon diferentes gráficos que representaban distintas propiedades de la red.

5.2. Primeros objetivos

Uno de los objetivos que se tenía desde el principio era el de poder variar los datos de la red al gusto del usuario. Se crearon e implementaron distintas funciones:

- `poda(G, umbral)`: permite podar la red G a partir de un umbral interpuesto por el usuario. Devuelve una nueva red H con el nuevo grafo.
- `comunidades(G)`: divide los nodos del grafo en distintas comunidades.
- `gradosNodos(G)`: crea y guarda un gráfico de sectores a partir de los grados de los nodos del grafo G .

- `propiedadesRed(G)`: crea y guarda un histograma que contiene el porcentaje de densidad, transitividad y promedio de agrupación de los nodos del grafo G .
- `calculaModularidad(G)`: calcula la modularidad de cada nodo del grafo G .
- `grafoInteractivo(G)`: crea a través de la herramienta Bokeh un grafo interactivo que muestra distintos datos de los nodos al posicionarse encima de cada uno.
- `crearHTML(G)`: crea una página web programada en HTML.
- `ejecutar(umbral)`: ejecuta la aplicación creando un grafo correspondiente al umbral introducido por el usuario.

La primera versión de la aplicación consistía en una función en el propio código Python que construía una página web en HTML con los datos creados anteriormente. A pesar de que esto permitía crear la aplicación web, no cumplía todos los objetivos ya que no permitía cambiar el diseño de la página ni añadir nuevas pestañas ni variar el umbral de la poda.

El principal problema que se encontró fue el de cómo poder crear un HTML desde Python, al cual se pudieran pasar los datos del grafo y lo creara. Se encontró primero una solución parcial que consistía en utilizar los métodos que ofrece la librería *webbrowser*. Esta solución permitía crear páginas web desde Python utilizando HTML, pero por otro lado no permitía que se le pudieran pasar datos por parámetros para crear la red. Esto se solventó guardando la red como un archivo HTML y cargando este archivo en la página web.

Por otro lado, la página no permitía cambiar el umbral de la poda desde la propia página sino que había que hacerlo desde el código. Esto no se pudo solventar hasta el cambio a Visual Studio.

5.3. Creación frontend

Tras la creación de una versión poco práctica y estática de la aplicación web, se empezó a desarrollar una nueva versión en el entorno de Visual Studio ayudado de los lenguajes HTML y CSS para crear el diseño. Esto hizo que se eliminara el método *crearHTML(G)* del código Python.

Se empezó creando una página web sencilla en HTML a partir de la aplicación anterior pero con la diferencia que esta nueva se ejecutaba desde el

terminal y sí iba a poder ser diseñada gracias a CSS. Una vez se probó su correcto funcionamiento se empezó a añadir diseños a la página y se le añadió una pestaña de información del proyecto, la cual solo contenía el nombre de la aplicación, el del autor (Enrique Camarero Alonso) y el de los tutores (José Ignacio Santos Martín y Virginia Ahedo García).

El principal problema ocurrió a la hora de comprobar si funcionaba la aplicación, ya que aunque tenía Python3 instalado no disponía del entorno virtual de este lenguaje, el cual es necesario para ejecutarlo en local. Tras diversas búsquedas por Internet, se descubrió la forma de poder ejecutarlo.

5.4. Creación opción de cambio de umbral

El umbral es el resultado que se obtiene de multiplicar el valor dado por el usuario en tanto por ciento y el máximo peso que tiene una pareja de enlaces en una tecnología. Una vez se tiene este valor se poda el grafo, en donde se eliminan todas aquellas relaciones que tengan menos peso que el del umbral.

Con el frontend ya creado se buscó la posibilidad de que se pudiera añadir una opción que permitiera cambiar el umbral de la poda, el cual era un objetivo principal del proyecto. Para ello se creó una nueva pestaña con un formulario en su interior que permitía añadir números enteros entre 0 y 100 inclusive. Este valor se guardaría en una variable de sesión y sería esta la variable que se pasaría al método ejecutar del código Python.

Una vez se probó la total funcionalidad del cambio de umbral desde una pestaña nueva se quiso juntar la pestaña principal y la del cambio de umbral en una, ya que de esta forma se ganaría en usabilidad al poder cambiar el umbral en la misma página en la que te encuentras. Además, se añadió un letrero informativo con el umbral que está ejecutado en ese momento.

Se encontraron diversos problemas a la hora de programar el cambio de umbral los cuales originaban errores internos del sistema. Estos errores se originaban cuando:

- Se introducía un valor no numérico.
- Se introducía un número menor que 0.
- Se introducía un número mayor que 100.
- Se introducía un número con decimales.

5.5. CREACIÓN DEL SISTEMA DE RECOMENDACIÓN DE LENGUAJES

Estos errores se solucionaron haciendo que el formulario solo aceptara números enteros comprendidos entre 0 y 100, ambos inclusive.

5.5. Creación del sistema de recomendación de lenguajes

Creada ya una página funcional se empezó la programación del sistema de recomendación de lenguajes. Éste está formado por los lenguajes que hay en el grafo y sus recomendaciones, los cuales son los enlaces con los que se relaciona. Estas relaciones se van a ordenar de mayor a menor recomendación dependiendo del porcentaje de relación que tengan entre ellos esos lenguajes. Esto se calcula de forma que cuanto mayor sea el peso de la relación entre 2 lenguajes mejor recomendado estará.

La tabla de recomendaciones se exportó a una hoja de cálculo y ésta a un dataframe. De todas las columnas tan sólo nos quedaríamos con el lenguaje y un máximo de 3 recomendados dependiendo el número de columnas de la tabla. Este dataframe resultante lo convertiríamos a un HTML para poder utilizarlo en la aplicación y permitir así que se le añadieran estilos.

Como al podar la red habría nodos que no aparecerían en el grafo ni en el sistema de recomendaciones decidí crear una tabla, de la misma forma que la del sistema de recomendación, que estuviera formada por esos lenguajes que desaparecen tras la poda.

Además, para facilitar el trabajo de búsqueda se añadió una barra buscadora para encontrar rápidamente el lenguaje que estamos buscando.

Para que funcionara debía de estar tanto el código del buscador como la tabla en el mismo HTML, por lo que me encontré con un problema, el cual conseguí resolver rescatando y modificando el método *crearHTML(G)*. El código de la tabla-buscador está formado por 3 partes de las cuales la primera y la tercera parte son estáticas, por lo cual eso se escribiría como código HTML directamente. La segunda parte, la de la tabla, se introdujo leyendo línea por línea el código HTML creado del dataframe y escribiéndolo en el HTML de la tabla-buscador. De esta forma conseguiríamos tener todas partes juntas y que funcionara exitosamente.

5.6. Heroku

Con la primera página y tecnología acabada se creó una aplicación en los servidores de Heroku, tras darme de alta en la plataforma, y se enlazó con el proyecto. Este enlace se puede realizar o bien a través de GitHub o bien a través de un del terminal del ordenador.

El problema llegó en este momento, ya que al no tener conocimiento previo sobre Heroku, tuve que informarme acerca de la manera de desplegar la aplicación y cuál debía ser el contenido de los archivos necesarios para su correcto funcionamiento.

Para que la aplicación funcione correctamente en Heroku se necesitan los siguientes archivos:

- Archivo Procfile: en este archivo se especifica el tipo de servidor a crear y el nombre del archivo ejecutable.
- Archivo ejecutable: este archivo principal del proyecto ya que es el que contiene la ejecución de la aplicación.
- requirements.txt: este archivo de texto está formado por todas las librerías que necesita el proyecto para su correcta ejecución.

Una vez creado todo se despliega el proyecto en Heroku, y si no ha surgido ningún problema podemos ejecutarlo y utilizar la aplicación de manera web a través de un enlace que nos facilita la plataforma.

5.7. Creación de las nuevas páginas con las tecnologías restantes

Para la creación del resto de tecnologías se ha creado una pestaña para cada una y se ha replicado el código de la pestaña de lenguajes y su código Python, modificando la columna a utilizar para esa tecnología en el código para que funcione correctamente para cada una, salvo para los marcos de trabajo, los cuales se encontraban divididos en dos columnas y se tenían que relacionar entre ambas.

5.8. Mejoras del código

Una vez se tenía una versión completa de la aplicación se corrigieron ciertos fallos que se encontraron al utilizar la aplicación Python y al revisar el código Python. Entre estos errores podemos encontrar:

- Parámetros en las variables de los métodos innecesarios ya que no se necesitan usar tras el despliegue de la aplicación en Visual Studio y la posibilidad de cambiar el umbral.
- Un fallo que hacía que la página diera error si no se escribe un umbral y ejecutas la aplicación.
- Creación de una página de *Inicio*.
- Finalización de la página *Acerca de*.

Trabajos relacionados

A pesar de que este proyecto no se relacione directamente con ningún otro, existen otros trabajos acerca de sistemas de recomendación basados en redes en muchos campos, desde recomendaciones de películas por ejemplo en Netflix, de música en Spotify o bien de productos en Amazon [13].

Tal y como se cuenta en el artículo *Network-based recommendation algorithms: A review* [26], los sistemas de recomendación son vitales en nuestro día a día ya que nos ayudan a resolver el problema que supone la sobrecarga de información, y es por ello por lo que las empresas lo utilizan para ayudarnos en el día a día.

Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

Una vez se ha finalizado el proyecto puedo sacar mis conclusiones, y ver tanto la evolución del mismo como las herramientas empleadas para programarlo.

El proyecto ha sido controlado mediante la metodología ágil de SCRUM a través de GitHub, lo que me ha permitido tener un mayor control sobre el proyecto, permitiéndome conseguir un mejor resultado en el mismo tiempo. Esto se debe a que al realizar el proyecto en sprints puedo ir viendo la evolución y corrigiendo los pequeños fallos que iba encontrando en el siguiente sprint.

Al tener que utilizar una encuesta desconocida para mí implica que al principio del proyecto me llevó un tiempo el estudio de su contenido, el pensar y decidir que utilizar y de que manera ya que se disponía de bastante información.

El trabajo ha supuesto un reto para mí, ya que además de utilizar y reforzar los diferentes conocimientos aprendidos durante el grado, he tenido que utilizar nuevas herramientas como pueden ser Heroku o Bokeh, las cuales supusieron un tiempo de comprensión y aprendizaje para así poder utilizarlas. Gracias a estas herramientas ha sido posible crear una aplicación web la cual se ejecuta desde un servidor en internet y es accesible a todas las personas.

7.2. Líneas de trabajo futuras

Algunas de las futuras líneas de trabajo que se podrían realizar serían:

- Añadir la posibilidad de cambiar el idioma de la aplicación.

- Incluir una opción para que no solo se pueda utilizar los datos de la encuesta del año en el que se ha programado, sino que el usuario pueda escoger entre todos los resultados de encuestas que haya hasta ese momento.
- Crear una opción que permita cambiar la aplicación a modo oscuro o a colores mejor adaptados para personas que sufren de daltonismo.
- Además del umbral, permitir más tipos de filtros para el sistema de recomendación como elegir el rango de experiencia de los encuestados, filtrar por países, por el tipo de trabajo que desempeña cada uno, entre otros.
- Página que tenga todas las tecnologías unidas en una sola red.
- Explotar la información disponible en la base de datos que no ha sido empleada para la realización de este TFG, como por ejemplo el trabajo y el perfil del desarrollador.

Bibliografía

- [1] Encarna Abellán. Scrum. <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>, 2022. [Internet; Accedido 26-abril-2022].
- [2] Gustavo B. Github. https://www.hostinger.es/tutoriales/que-es-github#%C2%BFQue_es_GitHub_y_por_que_es_tan_popular, 2022. [Internet; Accedido 27-abril-2022].
- [3] Gustavo B. Html. <https://www.hostinger.es/tutoriales/que-es-html>, 2022. [Internet; Accedido 27-abril-2022].
- [4] Salesforce blog. Heroku. <https://www.nts-solutions.com/blog/heroku-que-es.html>, 2022. [Internet; Accedido 27-abril-2022].
- [5] David Marquinez Burgos. Algoritmo de leiden. https://oa.upm.es/68317/1/TFG_DAVID_MARQUINEZ_BURGOS.pdf, 2021. [Internet; Accedido 7-junio-2022].
- [6] Daniel Ciruelo. Introducción a sistemas de recomendación. <https://blog.bi-geek.com/introduccion-a-sistemas-de-recomendacion/>, 2020. [Internet; Accedido 6-junio-2022].
- [7] Datacarpentry. Dataframe. <https://datacarpentry.org/python-ecology-lesson-es/02-starting-with-data/#:~:text=Un%20DataFrame%20es%2C>, 2022. [Internet; Accedido 28-abril-2022].
- [8] Edix. Framework. <https://www.edix.com/es/instituto/framework/>, 2022. [Internet; Accedido 26-abril-2022].

- [9] Kilo Tapias Peralta Escobar. Visual studio. <https://www.espaciohonduras.net/microsoft-visual-studio-concepto-y-que-es-y-para-que-sirve-microsoft-visual-studio>, 2022. [Internet; Accedido 27-abril-2022].
- [10] Graph everywhere. Sistemas de recomendación. <https://www.grapheverywhere.com/sistemas-de-recomendacion-que-son-tipos-y-ejemplos/#:~:text=Un%20sistema%20de%20recomendaci%C3%B3n%20es,o%20valor%20para%20el%20usuario.>, 2022. [Internet; Accedido 26-abril-2022].
- [11] Graph Everywhere. Algoritmo de louvain. <https://www.grapheverywhere.com/algoritmo-de-louvain/#:~:text=El%20algoritmo%20de%20Louvain%20esta,que%20pertenecen%20a%20una%20red.>, 2022. [Internet; Accedido 7-junio-2022].
- [12] Santiago Figueiras. Jupyter notebook. <https://ceupe.mx/blog/conoces-jupyter-notebook.html>, 2022. [Internet; Accedido 27-abril-2022].
- [13] Jennifer A. Jacobi Gregory D. Linden and Eric A. Benson. Patente sistema de recomendación de amazon. <https://patents.google.com/patent/US6266649>, 1998. [Internet; Accedido 1-junio-2022].
- [14] Juan Ignacio. Algoritmo k-nearest neighbor. <https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/>, 2018. [Internet; Accedido 8-junio-2022].
- [15] Juan A. Zúñiga Verdugo M. Paz Contreras Warz. Algoritmo de kernighan-lin. http://opac.pucv.cl/pucv_txt/txt-5000/ucd5101_01.pdf, 2015. [Internet; Accedido 7-junio-2022].
- [16] Marta Caro Martínez. Sistemas de recomendación y explicaciones basados en grafos de interacción. <https://eprints.ucm.es/id/eprint/71516/1/T43129.pdf>, 2022. [Internet; Accedido 6-junio-2022].
- [17] Hector Hurtado Ruesga. Sprint. <https://www.bbva.com/es/metodologia-scrum-que-es-un-sprint/>, 2022. [Internet; Accedido 26-abril-2022].
- [18] Sandra Garrido Sotomayor. Metodologías ágiles. <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>, 2022. [Internet; Accedido 26-abril-2022].

- [19] Piotr Stefaniak. Frontend. <https://descubrecomunicacion.com/que-es-backend-y-frontend/>, 2022. [Internet; Accedido 26-abril-2022].
- [20] UC3M. Overleaf. <https://www.uc3m.es/sdic/servicios/overleaf>, 2022. [Internet; Accedido 27-abril-2022].
- [21] Santander Universidades. Python. <https://www.becas-santander.com/es/blog/python-que-es.html>, 2022. [Internet; Accedido 27-abril-2022].
- [22] W3schools. Css. <https://www.w3schools.com/css/>, 2022. [Internet; Accedido 27-abril-2022].
- [23] Wikipedia. Algoritmo de girvan-newmann. https://en.wikipedia.org/w/index.php?title=Girvan%E2%80%93Newman_algorithm&oldid=1034946281, 2022. [Internet; Accedido 7-junio-2022].
- [24] Wikipedia. Latex. <https://es.wikipedia.org/w/index.php?title=LaTeX&oldid=142156268>, 2022. [Internet; Accedido 27-abril-2022].
- [25] Wikipedia. Teoria de redes. https://es.wikipedia.org/w/index.php?title=Ciencia_de_redes&oldid=141496321, 2022. [Internet; Accedido 01-junio-2022].
- [26] Fei Yu, An Zeng, Sebastien Gillard, and Medo Matus. Network-based recommendation algorithms: A review. *CoRR*, abs/1511.06252, 2015.

