



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería Informática



TFG en Ingeniería Informática
Sire



Presentado por María Ojeda Ruiz
en Burgos el
Tutores académicos: Dr. José Manuel Galán Ordax y
Dra. Virginia Ahedo García

D. José Manuel Galán Ordax y Dña. Virginia Ahedo García, profesores del departamento de Ingeniería de Organización

Exponen:

Que el alumno D. María Ojeda Ruiz, con DNI 71297369Z, ha realizado el TFG en Ingeniería Informática titulado: *Sire*.

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 a 8 de julio de 2021

Tutores académicos:

Fdo. D. José Manuel Galán Ordax

Fdo. Dña. Virginia Ahedo García

Resumen

Actualmente, nos encontramos sumidos en una pandemia a nivel mundial, la pandemia del COVID-19, causada por el virus SARS-CoV-2. Hecho que ha generado nuevas necesidades, que han repercutido en cambios muy relevantes en la vida de las personas en los distintos ámbitos de su vida.

Este proyecto surge para dar solución a una de las necesidades encontradas a raíz de esta pandemia, el proceso de generación de clases en los colegios tratando de minimizar el riesgo de contagios. Anteriormente, se tenían en cuenta los hermanos encontrados en el mismo curso, tratando de unirlos en una misma clase puesto que su contacto además de en el colegio también se producía en el hogar. Pero ¿y los hermanos que no tienen la misma edad?

Gracias a la ayuda del algoritmo denominado recocido simulado y las redes complejas, hallamos una solución a esta organización de un modo más eficiente. Tenemos en cuenta todos los hermanos que se puedan encontrar en el colegio, generando enlaces entre las clases y tras ello buscamos la mejor combinación para minimizar tanto el riesgo de contagio individual como el riesgo por grupo.

Sire (Sibling-Rewiring) es por tanto una aplicación concebida para minimizar los riesgos de contagio de cualquier enfermedad transmitida por el aire o contacto cercano.

Descriptores

Generación de redes complejas, análisis de comunidades, pandemia, COVID-19, minimización de contagio, Python, Servidor Flask, MySQL.

Abstract

Nowadays, we are experiencing a global pandemic, COVID-19 pandemic produced by SARS-CoV-2 virus. This fact has generated new needs changing people lifestyle.

This project emerges to respond to one of those needs, the grouping of children in school. Previously they had thought of siblings with same age, getting them into the same class, because they also be together at home. But what happens with siblings with different age?

Thanks to the help of simulated annealing and complex networks we have achieved a solution to the problem. We consider all siblings in school to generate a network, where links are siblings and nodes classes of school. We mix that net looking for a similar one with a minimum risk of transmission.

Sire (Sibling-Rewiring) is therefore an application designed to minimize the risk of spread of any disease transmitted by air or close contact.

Keywords

Complex networks, community analysis, pandemic, COVID-19, risk minimization, Python, Flask server, MySQL.

Índice General

Índice General	4
Índice de Ilustraciones.....	6
A. Introducción	7
1.1 Estructura de la memoria	8
1.2 Enlaces de interés.....	8
B. Objetivos del proyecto.....	9
2.1 Objetivos generales.....	9
2.2 Objetivos técnicos.....	9
2.3 Objetivos personales	10
C. Conceptos teóricos.....	11
3.1 Recocido Simulado	11
<i>Estructura de implementación</i>	11
<i>Mecanismo de transición entre soluciones – Vecindad</i>	12
<i>Secuencia de enfriamiento</i>	12
3.2 Red	13
3.3 Conectividad	15
<i>Componentes conectados dentro de la red</i>	15
D. Técnicas y herramientas	17
4.1 Metodología ágil - SCRUM.....	17
4.2 Control de versiones	17
4.3 Gestión de tareas	17
4.4 Gestión del repositorio.....	18
4.5 Documentación	18
4.6 Referencias bibliográficas.....	18
4.7 Lenguaje de programación.....	18
4.8 Framework web	19
4.9 Herramienta para la interfaz.....	19
4.10 Generación de la red	19
4.11 Base de datos.....	19
4.12 Wiki.....	19
4.13 Creación de prototipo.....	19

Índice General

4.14	Generación de diagramas	20
E.	Aspectos relevantes del desarrollo del proyecto.....	21
5.1	Inicio del proyecto	21
5.2	Metodologías.....	21
5.3	Formación	21
	Flask	22
	Bootstrap	22
	Diseño UI/UX	22
	HTML y css	22
5.4	Desarrollo del proyecto.....	22
5.5	Problemas técnicos encontrados	24
	Problema de conexión con la base de datos	24
	Un hilo de ejecución secundario seguía activo tras finalizar el principal	25
	Heroku no permite alojar la aplicación	25
	Uso del framework deap	28
5.6	Wiki.....	28
F.	Trabajos relacionados	29
G.	Conclusiones y líneas de trabajo futuras	30
7.1	Conclusiones	30
7.2	Líneas de trabajo futuras.....	30
	Alojar la aplicación en web.....	30
	Implementar de una solución alternativa	30
	Mostrar el tiempo restante.....	30
	Realizar encriptados	31
	Internacionalizar.....	31
	Bibliografía	32

Índice de Ilustraciones

Ilustración 1- Red completa del colegio.....	14
Ilustración 2- Red de clases.....	15
Ilustración 3 - Conectividad de una red.	16
Ilustración 4 - Tablero Zenhub.....	17
Ilustración 5 - GitHub Desktop.....	18
Ilustración 6 - Conexión entre Flask y MySQL.	22
Ilustración 7 - Creación de tablas y procedimientos.	23
Ilustración 8 - Subida de archivos.	23
Ilustración 9 - Secuencias de enfriamiento.	24
Ilustración 10 - Importación Matplotlib.....	25
Ilustración 11 - Opciones de despliegue de Heroku.....	25
Ilustración 12 - Despliegue Manual con GitHub.	26
Ilustración 13 - Error de apertura de aplicación.....	26
Ilustración 14 - Login en Heroku.....	26
Ilustración 15 - Crear aplicación.....	27
Ilustración 16 - Web crashed.....	27
Ilustración 17 - Logs Heroku error H-10.	27
Ilustración 18 - Escalado de la web.....	28

A. Introducción

Actualmente, nos encontramos sumidos en una pandemia a nivel mundial, la pandemia del COVID-19, causada por el virus SARS-CoV-2. Los primeros casos surgieron en 2019 en una zona de China llamada Wuhan. Tras ello, fueron surgiendo más y más casos, hasta que el 11 de marzo de 2020 la Organización Mundial de la Salud lo dio consideración de pandemia [1].

Este hecho ha producido un gran cambio en nuestra sociedad en cuanto a estilo de vida y también a relaciones interpersonales. Ha surgido la necesidad del uso de mascarillas, distanciamientos sociales, cierre de aulas y trabajos. Por todos estos motivos, al igual que se han producido cambios han surgido necesidades.

Uno de los grandes cambios que ha sido reseñable es en el ámbito escolar. Tras el cierre de aulas aproximadamente durante 16 semanas, lo que hizo difícil tanto para profesores como alumnos mantener el ritmo y estilo de aprendizaje llevados hasta el momento, la vuelta a las aulas tampoco ha sido más sencilla. Al volver, se han encontrado problemas sobre cómo hacer que estos posibles contagios se redujeran.

Se creó un plan de vuelta a los colegios que implantaba los grupos burbuja. Estos grupos se han utilizado en las primeras etapas del aprendizaje, consistían en que los alumnos no debían mantener las distancias obligatorias entre sí, pero por el contrario no podían juntarse con alumnos de otras etapas o clases en los pasillos o el patio.

Aquí se comprobó un fallo en la eficiencia de este planteamiento, ya que se estaban teniendo en cuenta los alumnos que tuvieran hermanos en la misma etapa y curso, para juntarlos en una misma clase. Estos debían estar unidos ya que su probabilidad de contagiarse el uno al otro en casa también existía. Pero ¿y los hermanos que no estuvieran en la misma etapa o en el mismo curso?

La finalidad de Sire (acrónimo de Siblings Rewiring) no es otra que facilitar uno de este procedimiento de organización. Estamos trabajando con el cómputo total de los alumnos teniendo en cuenta posibles hermanos tanto de la misma edad como de distinta. En el proyecto, nos hemos centrado en disminuir al máximo posible la posibilidad de contagio que ocurre en los colegios, teniendo en cuenta tanto el contagio individual como el que puede aparecer por grupo. Así, somos capaces de identificar unos grupos burbuja más amplios, relacionados entre sí y teniendo en cuenta la interacción doméstica de alta intensidad que se produce entre hermanos convivientes y que nos permitieran mejorar la eficacia y eficiencia del proceso, comúnmente manual, que se realiza en los colegios para reorganizar las clases.

Además de la reorganización pura y dura, en el proyecto se ha creado una base de datos que nos permitirá poder realizar estadísticas sobre el funcionamiento de la misma aplicación. Estos datos serán valiosos ya que podremos comprobar con qué tipo de colegios o estructuras de la red es más eficiente el proyecto. Estos datos podrían terminar explotándose para poder realizar mejoras en un futuro donde tuviera mayores debilidades.

Para realizar este proceso, hemos utilizado el marco conceptual de redes complejas [2]. Una red está compuesta por un conjunto de nodos y sus interacciones, llamadas enlaces. En el proyecto tenemos dos redes principales, la primera incluye el conjunto total de los alumnos como nodos, a estos nodos se les dan los atributos de clase, etapa y curso, los enlaces de la red consisten en pertenencia a la misma clase o que haya una relación de hermanos entre dos nodos. Tras procesar esta red, creamos una red

A. Introducción

auxiliar donde los nodos son las distintas clases o aulas del colegio y los enlaces serían el número de hermanos entre clases, es decir, si un alumno de la clase A tiene un hermano en la clase B, habrá un enlace entre las clases A y B.

En nuestro caso realizamos un tratamiento de estas redes para comprobar las interacciones de los enlaces, dándoles unos pesos que después podemos cuantificar para realizar los análisis y comprobar cuál será la mejor forma de enlazar los nodos que tenemos para reducir los riesgos de contagio de los que he hablado anteriormente.

Este planteamiento de proyecto ha surgido a raíz de la pandemia del COVID – 19, pero se podría extrapolar a cualquier enfermedad contagiosa, ya sea por aire o contacto personal. Puesto que todos los años hay casos de gripe, varicela, piojos, entre otras, que podríamos intentar reducir utilizando esta aplicación.

Desde el punto de vista académico, ha supuesto una oportunidad para comprender mejor el funcionamiento de las redes y el análisis de estas. También ha sido posible en este proyecto trabajar con un caso real donde se pudieran aplicar los conocimientos adquiridos.

1.1 Estructura de la memoria

La memoria tiene la siguiente estructura:

- Introducción: Se realiza una descripción del proyecto y cómo se ha llegado a la conclusión de que este es necesario. También se explica la distribución de la memoria y los anexos y algunos enlaces de interés a la aplicación y al repositorio.
- Objetivos del proyecto: En este apartado se realiza la descripción de los objetivos que se busca alcanzar: los generales, técnicos y personales.
- Conceptos teóricos: En este apartado desarrollamos los conceptos considerados como necesarios para poder comprender el proyecto.
- Técnicas y herramientas: Se muestra un listado de las herramientas principales y destacables que se han utilizado en la realización del proyecto.
- Aspectos relevantes del desarrollo del proyecto: En el apartado se exponen hechos reseñables durante el desarrollo del proyecto.
- Trabajos relacionados: En este caso concreto se expone la falta de trabajos previos.
- Conclusiones y líneas de trabajo futuras: En este apartado se expondrán las posibles líneas de trabajo futuras, así como las conclusiones y conocimientos obtenidos tras la finalización del proyecto.

También tenemos la estructura que tiene el documento con los anexos:

1.2 Enlaces de interés

En este apartado mostraremos algunos enlaces que podrían resultar interesantes relativos al proyecto.

- Wiki de la aplicación: <https://github.com/Mariaojruiz/Sibling-Rewiring/wiki>

B. Objetivos del proyecto

En el siguiente apartado se exponen tanto los objetivos generales y técnicos como los personales que encontramos en este proyecto.

El proyecto fue propuesto por la universidad, a través de una idea de mi tutor José Manuel tras comprobar una necesidad muy actual.

2.1 Objetivos generales

En este apartado se describen los objetivos generales del proyecto, objetivos que se fijaron al inicio del proyecto. Los objetivos generales son los siguientes:

- Generar redes que procesen la información de los colegios.
- Registrar en una base de datos la información relativa a los usuarios.
- Generar registros de la creación de la red aleatoria, como son nodos, enlaces y los valores necesarios para resolver el recocido simulado. Estos datos son registrados para poder ser procesados posteriormente.
- Generar registros de los archivos que son cargados en la aplicación.
- Trabajar con el recocido simulado [3] como método de optimización de la red, así como todas las posibles secuencias de enfriamiento que se han estudiado, lineal, geométrica, logarítmica, de Cauchy y de Cauchy modificado.
- Crear una aplicación que permita procesar la información de la red para obtener el mejor resultado posible con la organización que genere menor número de contagios, equilibrado de riesgo y otros potenciales objetivos aislados o simultáneos.
- Integración de base de datos junto a servidor Flask.
- Escribir un manual de usuario accesible a través de una Wiki.
- Generar un apartado de loggeo dentro de la aplicación para poder controlar los usuarios que son registrados.

2.2 Objetivos técnicos

En este apartado se definen los objetivos técnicos del proyecto, los detalles de la implementación y herramientas usadas. Estos objetivos de carácter técnico son los siguientes:

- Crear una aplicación utilizando Python como lenguaje de desarrollo. En Python se realiza la creación de las redes, utilizando la librería NetworkX. Además, utilizaremos otras librerías de Python para el procesamiento de la red, Numpy, Pyplot, Matplotlib y pandas.
- Utilizar Flask para realizar el desarrollo de la web, usándolo como servidor.
- Trabajar siguiendo metodología ágil, SCRUM, para la gestión de proyectos, dado en la asignatura de Gestión de Proyectos.
- Utilizar GitHub como herramienta para el control de versiones.

B. Objetivos del proyecto

- Usar ZenHub para el seguimiento, planificación y generación de informes relativos al proceso seguido en GitHub.
- Generar una base de datos MySQL.
- Proporcionar una Wiki donde se encuentre el manual de usuario de la aplicación web.

2.3 Objetivos personales

En este apartado se exponen los objetivos de carácter personal. Estos objetivos son los siguientes:

- Utilizar la metodología SCRUM sobre un caso real para así entender y poner en práctica lo estudiado anteriormente, pudiendo comprender su utilidad.
- Aprender desarrollo web utilizando Flask.
- Aprender a trabajar con programación web en html.
- Poner en práctica los conocimientos adquiridos sobre procesamiento de redes.
- Aprender a trabajar con el lenguaje de bases de datos MySQL poniendo a su vez en práctica los conocimientos teóricos obtenidos en Bases de Datos.
- Comprender el funcionamiento de bases de datos mediante programación en Python.
- Aplicar los conocimientos adquiridos en asignaturas relacionadas con el desarrollo software como Análisis y Diseño de Sistemas, Interacción Hombre/Máquina, Diseño y Mantenimiento del Software...

C. Conceptos teóricos

Esta parte del documento está dedicada a explicar conceptos que nos van a permitir comprender de una forma más profunda el proyecto.

3.1 Recocido Simulado

El Recocido Simulado es un algoritmo basado en trayectorias que usa una metáfora basada en la industria siderúrgica, sobre la termodinámica que utilizan, para la aceptación de las posibles soluciones o su desecho [3].

En este algoritmo se realizan diversas iteraciones sobre un problema. En cada una de las iteraciones utilizamos un mecanismo para evitar atascarnos en óptimos locales, aceptando algunas soluciones peores, aunque este proceso se realiza de un modo controlado. En este algoritmo como se ha dicho previamente se utilizan símiles con la termodinámica, por lo que este parámetro de aceptación se denomina temperatura.

La solución final del problema se halla partiendo de una solución inicial y ejecutando un número fijado de iteraciones del algoritmo, en cada iteración generamos un vecino (cambio realizado sobre la solución previa), trabajando con este vecino, generamos una solución candidata, que puede resultar ser mejor o peor que la inicial. Si fuera mejor la tomamos como buena y sustituimos la previa por la candidata. Si esta fuera peor, podemos aceptarla en función de una probabilidad que depende de la temperatura actual (T) y de la degradación de la función objetivo (δ), ya que, si nunca escogiéramos soluciones peores, podríamos quedarnos en un entorno de vecinos que nos llevasen a descartar soluciones mejores por no probar.

La probabilidad de aceptación de la solución se modela utilizando distintas posibles distribuciones, siendo la más común la distribución de Boltzmann [4].

$$P_{aceptación} = e^{(-\frac{\delta}{T})}$$

Estructura de implementación

La estructura de implementación del problema consiste en los pasos a seguir para llegar a la solución del problema. Dividiéndose en la representación, la búsqueda de la solución inicial, el mecanismo de transición entre soluciones y la secuencia de enfriamiento.

Tal y como vemos en la definición, explicamos a continuación las cuatro partes en que se divide la estructura de implementación:

- **Representación:** Será la forma en que se irán guardando los valores dentro de la solución. En nuestro caso utilizaremos una codificación real, es decir, se guardarán los parámetros reales en la solución. En nuestro caso, guardamos la red.
- **Búsqueda de la solución inicial:** Se podrá llegar a la solución inicial por diversos caminos, tomando unos valores aleatorios del problema, con una solución previa hallada anteriormente con este mismo algoritmo o utilizando una solución obtenida a partir de otras heurísticas.

C. Conceptos teóricos

- **Mecanismo de transición entre soluciones:** en este caso utilizamos el concepto de vecindad.
- **Secuencia de enfriamiento**

Mecanismo de transición entre soluciones – Vecindad

Este mecanismo consistirá en la búsqueda de la posible nueva solución al problema. Nos encargaremos de generar una nueva solución utilizando el concepto de vecindad, tras ello, calcularemos cuál de las soluciones es mejor para el problema, si la actual o la que teníamos anteriormente y, por último, si esta fuera peor pasaremos a utilizar el criterio de aceptación.

Como describimos en la descripción del proceso, definiremos cada uno de los pasos a seguir.

- **Generación de una solución candidata:** Para generar una posible solución al problema utilizamos el concepto de vecindad, la vecindad será el conjunto de movimientos posibles. La estructura de vecindad define para cada solución existe un conjunto soluciones cercanas a esta en algún sentido. Una vez tenemos este conjunto vecindad se realiza la selección de un elemento de la misma, un vecino candidato del que obtendremos una posible solución candidata al problema.
- **Cálculo de la mejor solución:** Tras obtener el vecino, obtendremos una solución candidata utilizando este. Una vez tenemos las dos soluciones, la inicial y la candidata, obtenemos la diferencia entre estas, si esta diferencia fuera positiva, no modificamos la solución previa, pero si fuera negativa, tomaremos la solución candidata como la actual del problema.
- **Uso del criterio de aceptación:** Una vez hemos calculado la diferencia, si consideramos que la solución inicial es mejor que la candidata, vamos a utilizar un criterio de aceptación. Este criterio se utiliza para no quedarnos en óptimos locales y abrir la búsqueda de soluciones a otras partes del vecindario. Este criterio de aceptación consiste en comprobar si un número aleatorio en el conjunto de números entre [0-1] sea menor que $e^{(-\frac{diff}{T})}$, siendo *diff* la diferencia entre las dos soluciones y T la temperatura actual.

Secuencia de enfriamiento

La secuencia de enfriamiento se basa en cómo vamos reduciendo el valor de la Temperatura en cada una de las iteraciones para llegar de la que tomamos como temperatura inicial del problema a la final. Para realizar este proceso, necesitamos unos parámetros iniciales, que serán la temperatura inicial, L y la temperatura final.

- **Temperatura inicial:** Este valor lo obtendremos multiplicando el valor de la solución que tomamos como inicial por 0,4.
- **L:** Denominamos L al número total de iteraciones que realizamos para la búsqueda de la solución a nuestro problema.
- **Temperatura final:** La temperatura final es un valor muy próximo a cero, que será el valor al que llegaremos tras las diversas reducciones del parámetro temperatura.

C. Conceptos teóricos

Una vez tenemos estos valores iniciales procedemos a hacer las diversas iteraciones marcadas por el parámetro L . Estas iteraciones las realizaremos hasta que la temperatura actual sea mayor que la temperatura final escogida.

En cada iteración obtendremos una nueva temperatura, que hallaremos mediante una secuencia de enfriamiento entre las diversas opciones que hay. En nuestro caso hemos implementado cinco posibles secuencias [5]. Estas secuencias de enfriamiento son:

- **Secuencia de enfriamiento lineal:** Calcula la nueva temperatura de la iteración utilizando una función lineal. La nueva temperatura será la resta entre la actual y la iteración en que nos encontramos multiplicada por un parámetro beta.
- **Secuencia de enfriamiento geométrica:** Se calcula la nueva temperatura utilizando una función geométrica, la nueva temperatura se obtendrá multiplicando la actual por un parámetro pasado $Alpha$ (α).
- **Secuencia de enfriamiento logarítmica:** Calculamos la nueva temperatura utilizando una función logarítmica, que será la división de la temperatura inicial fijada entre $1 + (\alpha * \log(1 + iteraciónActual))$.
- **Secuencia de enfriamiento de Cauchy:** La nueva temperatura se calcula usando una progresión de Cauchy. Esta temperatura se hallará dividiendo la temperatura inicial fijada inicialmente entre $1 + iteraciónActual$.
- **Secuencia de enfriamiento de Cauchy Modificado:** Se calculará la temperatura con la progresión de Cauchy Modificado. Calcularemos por una parte un parámetro $\beta = \frac{temperaturaFinal}{(L-1) \times temperaturaInicial \times temperaturaFinal}$ y una vez tenemos este $Betha$, la temperatura actual será la previa entre $1 + \beta \times temperaturaActual$.

3.2 Red

En este apartado definiremos el concepto de red [2], para explicar qué es una red, debemos explicar sus partes. Una red es un conjunto de nodos y los enlaces que relacionan estos nodos. Cada red tendrá un tamaño determinado por el número de nodos de esta.

En lo relativo a los **nodos**, estos pueden ser de un solo tipo en toda la red o puede haber nodos de varios tipos en la misma red. Si estos nodos son de un solo tipo se denominan redes unimodales, como en nuestro caso. Las redes que encontramos en nuestra aplicación son redes unimodales, ya que los nodos son todos del mismo tipo, en la primera red, encontramos que todos los nodos son alumnos y en la procesada, cada uno de los nodos es una clase. [6]

Además, cada nodo de la red puede tener unos atributos. Estos atributos nos aportan información sobre el nodo con que se relacionan. Los atributos pueden ser de distintos tipos y se podrían crear diccionarios que recojan toda la información relativa a los nodos de la red.

Por otra parte, se encuentran los **enlaces**, los enlaces son el modo en que se relacionan los nodos de la red. Las redes se pueden dividir en dos tipos en función de los enlaces, redes dirigidas y redes no dirigidas. En las redes no dirigidas los enlaces muestran únicamente una relación entre dos nodos, este sería el caso de nuestra red, una clase se relaciona con otra si hubiera un hermano en cada una de ellas. Respecto a las redes dirigidas, los enlaces muestran un sentido, uno de los nodos será el padre y el que se

C. Conceptos teóricos

relaciona con este su hijo, habiendo una relación jerárquica entre ellos. Además, podrían existir auto-enlaces dentro de una red dirigida, lo que implicaría que el origen y el destino del enlace son el mismo nodo. [7]

Además, teniendo en cuenta el número de enlaces que encontramos en cada nodo, se puede distinguir entre redes simples o complejas. Las redes simples son aquellas en que por cada par de nodos hay un máximo de un enlace. Si se encontrase más de un enlace por par de nodos, se denominaría red compleja.

En nuestro proyecto encontramos dos redes diferenciadas. En la primera red, cada uno de los nodos representa a un alumno del colegio, estos nodos, tienen unos atributos relativos al curso, clase y etapa a la que pertenece el nodo. Los enlaces que encontramos en esta primera red son pertenencia a la misma clase, además de encontrarse enlaces que relacionan hermanos. Mostramos un ejemplo a continuación, hemos coloreado cada etapa en un tono diferente, siendo infantil el tono rosado, primaria el naranja y secundaria el azul.

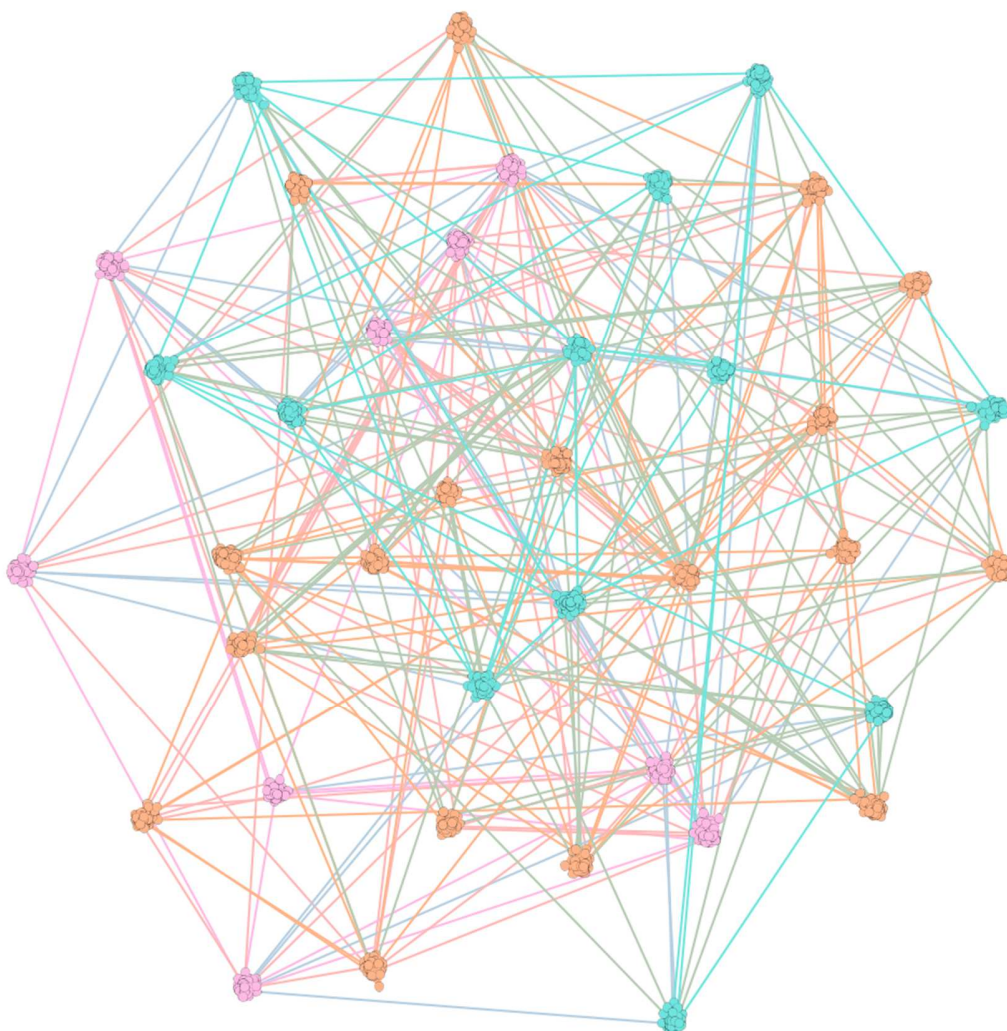


Ilustración 1- Red completa del colegio.

C. Conceptos teóricos

En segundo lugar, mostramos un ejemplo de la red de clases. Esta red es una simplificación de la anterior. En ella cada uno de los nodos es una de las clases del colegio y se encuentra un enlace entre dos nodos si hay un hermano en cada una de ellas. Por ejemplo, si hubiera un hermano en primero de primaria A y su hermano se encontrara en sexto de primaria A, habría un enlace entre los dos nodos primaria1A y primaria6A.

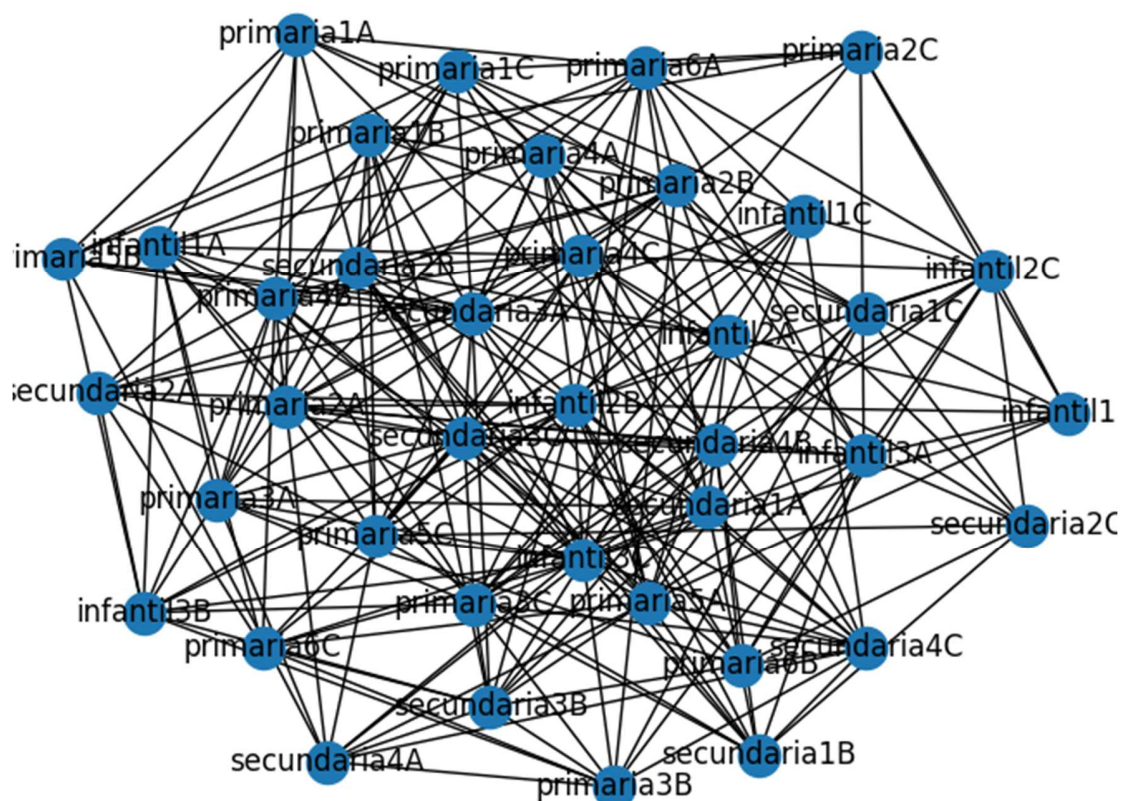


Ilustración 2- Red de clases.

3.3 Conectividad

Las redes se pueden diferenciar por cómo es su conectividad [8]. Las redes podrían estar **totalmente conectadas**, es decir, se permite la conexión directa entre los nodos de la red. Esto implicaría que hay enlaces que unirían todas las posibles parejas de nodos sin necesidad de pasar por otros nodos. En el caso contrario, se diría que la red está **parcialmente conectada**. Por otra parte, se dice que una red es **bloqueante** si no es posible conectar, en ningún caso, todas las posibles parejas de nodos que se puedan formar.

Componentes conectados dentro de la red

Un componente conectado de una red no dirigida como es nuestro caso es el número máximo de nodos donde cada par está conectado por un camino. Los componentes conectados forman una partición del conjunto de nodos de la red, la unión del total de los componentes de la red forma el conjunto de nodos de la red. De igual modo, podemos decir que la relación que hay entre dos nodos del componente es una relación reflexiva, simétrica y transitiva.

C. Conceptos teóricos

En las redes no dirigidas reales, lo más común es encontrar un componente de gran tamaño (el componente gigante) que conecta la gran parte de la red. Mientras esto ocurre, el resto de la red se divide en un gran número de pequeños componentes desconectados del resto. Como se puede observar en la siguiente imagen:

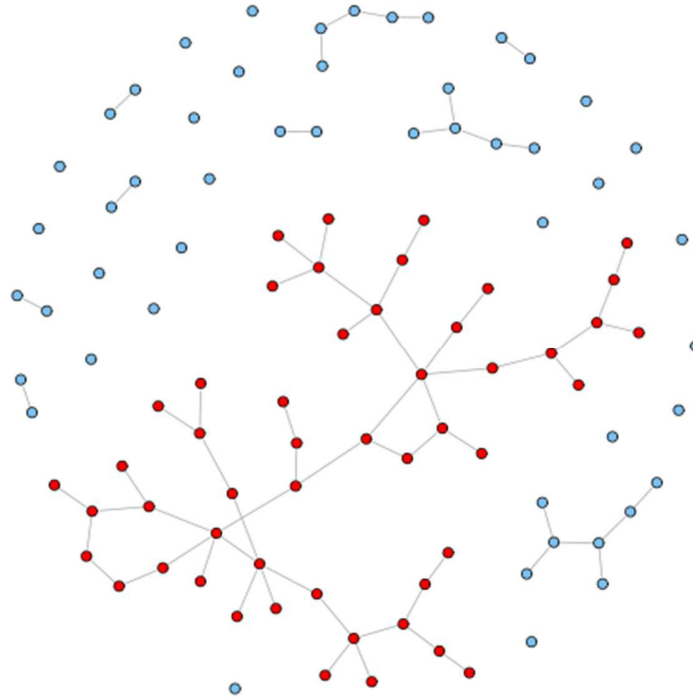


Ilustración 3 - Conectividad de una red.

En este proyecto, utilizamos los componentes conectados para hallar la probabilidad de contagio por cada componente siendo esta una de las partes para hallar la solución.

D. Técnicas y herramientas

En este apartado se explicarán las herramientas utilizadas en el desarrollo del proyecto, explicando cada una.

4.1 Metodología ágil - SCRUM

La metodología SCRUM es un modo de trabajo ágil basado en el principio de desarrollo iterativo e incremental. Esta especialmente indicado para proyectos complejos. Esta forma de trabajar favorece reducir la duración del proyecto. Los parámetros fundamentales son innovación, competitividad, flexibilidad y productividad. [9]

El total del trabajo se divide en períodos cortos llamados “sprints” que serán de duración variable según la necesidad de estos. Cada uno de estos Sprints proporcionará un resultado completo y funcional. Al inicio de cada Sprint se realiza una reunión para establecer el trabajo a realizar y al final del mismo se producirá otra para evaluar los resultados obtenidos. Este tipo de metodología se ha explicado y puesto en práctica gracias a la asignatura de Gestión de Proyectos.

4.2 Control de versiones

La herramienta empleada para el control de versiones es **Github** [10]. Es un portal que fue creado para que los desarrolladores suban el código de sus aplicaciones y herramientas. En esta plataforma podemos alojar los repositorios tanto de forma pública como privada, permitiéndonos usarla de modo colaborativo o como espacio personal.

Esta herramienta fue escogida puesto que fue la utilizada y enseñada en la asignatura de Gestión de Proyectos.

4.3 Gestión de tareas

La herramienta utilizada es **Zenhub** [11]. Esta es una extensión del navegador usada para poder gestionar nuestro repositorio de Github donde se encuentra el proyecto. Gracias a ella, podemos añadir Sprints, tareas y gestionar estos. En Zenhub se pueden crear tarjetas y realizar los pasos de nueva tarea, en progreso, *backup*, finalizado y cerrado, entre otras opciones personalizables. Además, nos permite obtener informes detallados.

Mostramos un ejemplo del tablero en la siguiente imagen.

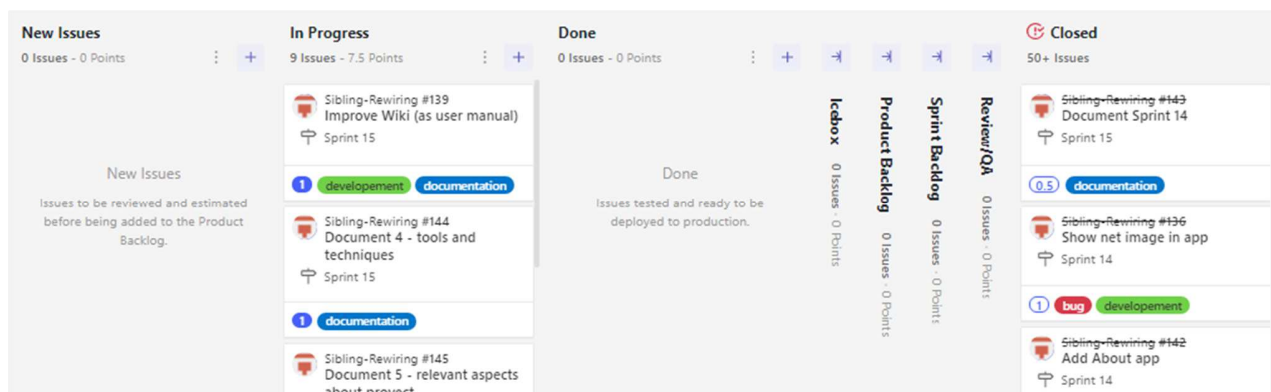


Ilustración 4 - Tablero Zenhub.

D. Técnicas y herramientas

Esta herramienta ha sido escogida ya que fue explicada y utilizada en la asignatura de gestión de proyectos.

4.4 Gestión del repositorio

Hemos utilizado **GitHub Desktop** [12], esta es una interfaz gráfica de usuario que se mimetiza con GitHub permitiéndonos realizar la gestión completa del repositorio desde nuestro ordenador. Desde la misma se pueden gestionar los commits de un modo sencillo, pudiendo ligarlos a los colaboradores.

En la herramienta, se muestran las diferencias encontradas en cada uno de los ficheros de un modo visual, mostrando resaltados cada una de estas modificaciones. Mostramos a continuación un ejemplo.

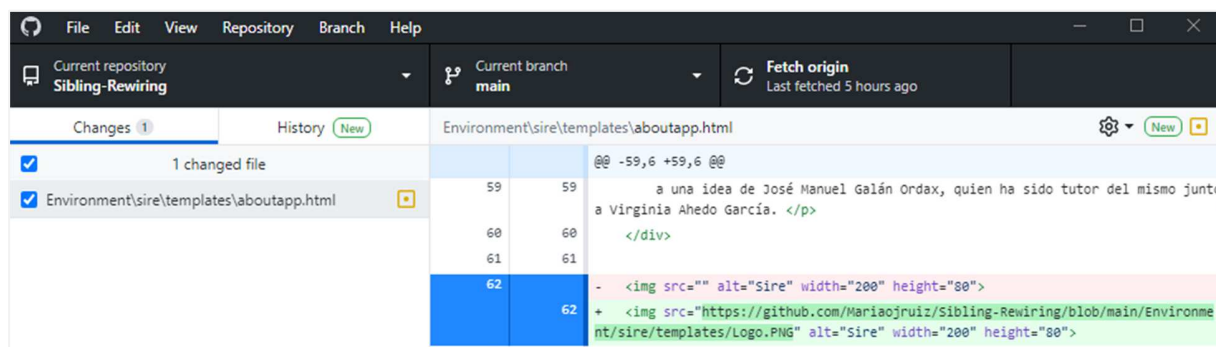


Ilustración 5 - GitHub Desktop.

4.5 Documentación

Para la documentación, la herramienta escogida ha sido **Microsoft Word** [13]. Esta herramienta es un editor de textos, que ofrece la posibilidad de crear documentos de un modo sencillo. Se eligió esta frente a las otras opciones planteadas, ya que poseo un mayor conocimiento sobre esta que sobre las alternativas.

Las posibles alternativas planteadas fueron LaTeX [14] junto a su editor online Overleaf, pero esta fue descartada ya que los conocimientos sobre la misma son escasos y está muy enfocada a la creación de documentos más técnicos, junto a una gran cantidad de fórmulas o expresiones matemáticas. Aparte, la otra opción que se tuvo en cuenta fue OpenOffice [15], pero esta se descartó ya que era similar a Microsoft Word y se prefería trabajar con la segunda.

4.6 Referencias bibliográficas

Se ha utilizado **Zotero** [16] como herramienta para este apartado. Zotero es un gestor de referencias libre que permite almacenar referencias de un modo sencillo y rápido. Posee una extensión de navegador, así como un complemento propio para Microsoft Word, este último nos permite realizar las citas y la bibliografía con gran eficacia.

4.7 Lenguaje de programación

Python [17] es el lenguaje utilizado para realizar el proyecto junto al *framework* web Flask, este se explicará a continuación. Este ha sido escogido por haberse adquirido grandes conocimientos y en especial porque se ha empleado para el tratamiento de redes en la carrera. Este lenguaje posee una librería llamada NetworkX, que es una de las más utilizadas en el mundo del análisis de redes.

También ha sido escogida por conocer librerías como ‘numpy’ o ‘pandas’ que son utilizadas en el tratamiento de datos.

4.8 Framework web

Flask [18] es un *framework* escrito en Python para poder desarrollar aplicaciones web de forma sencilla. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

4.9 Herramienta para la interfaz

Se ha utilizado **Bootstrap** [19] para realizar la interfaz. En lugar de desarrollar el código desde 0, se optó por implementar Bootstrap. Es una biblioteca multiplataforma de código abierto que permite a los desarrolladores diseñar sus sitios y aplicaciones web.

4.10 Generación de la red

Se ha utilizado la librería **NetworkX** [20] de Python ya que permite creación y análisis de redes complejas, además de ayudar con su procesamiento. Se utiliza para estudiar los grafos. Es un software libre publicado bajo la licencia BSD-new.

El motivo principal por el que se ha escogido esta librería es por el hecho de haberse empleado a lo largo de la carrera en algunas asignaturas por lo que ya se tenían conocimientos sobre la misma.

4.11 Base de datos

Para la base de datos hemos trabajado con **MySQL** [21]. Aparte, se ha utilizado MySQL WorkBench como gestor de la base de datos. No sería necesario utilizarlo, pero resulta más cómodo emplearlo.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra cuenta con una versión comercial acompañada por Oracle. Actualmente es de las más famosas y utilizadas mundialmente.

Además, se ha comprobado que Python permite la creación, modificación e inserción de elementos en tablas directamente. [22]

4.12 Wiki

Para realizar la Wiki, se ha utilizado la plataforma **Wikidot**. Esta permite realizar wikis propias de forma gratuita y realizarlo íntegramente desde el navegador. La versión gratuita nos limita el número de sitios que podemos realizar dándoles un tamaño máximo, pero suficiente para el tamaño necesitado para nuestro proyecto.

4.13 Creación de prototipo

Se ha utilizado **Pencil** para realizar el prototipo de la aplicación. Esta aplicación nos permite exportar el prototipo en imágenes, así que era una buena opción para después poder introducirlas en los anexos.

En este caso, se podrían haber utilizado diversas herramientas, pero se ha escogido Pencil ya que se trabajó con ella en la asignatura de Hombre Máquina. Entonces, al haber conseguido conocimientos de esta pareció la posibilidad más coherente.

4.14 Generación de diagramas

Para este proceso, se ha utilizado una herramienta llamada **Draw.io** accesible a través del siguiente [enlace](#). La herramienta permite realizar diagramas UML de modo sencillo, incorporando una serie de plantillas que pueden ayudar al crear estos.

Es una herramienta gratuita que no necesita realizar ninguna descarga ya que se trabaja desde web. Además, permite vincularlo con nuestro repositorio de GitHub, Google Drive o OneDrive, entre otras opciones.

E. Aspectos relevantes del desarrollo del proyecto

En este apartado se van a explicar las decisiones tomadas a medida que el proyecto se ha ido desarrollando, exponiendo los posibles problemas que han ido surgiendo.

5.1 Inicio del proyecto

La idea del proyecto surgió por una necesidad encontrada por el tutor del mismo José Manuel. Se planteó la posibilidad de realizar este proyecto y tras cursar la asignatura de “Nuevas Tecnologías” surgió interés por las redes complejas sus características y análisis de estas.

Por lo tanto, viendo el momento actual en que nos encontramos y siendo que estaba ligado a las redes complejas, me pareció una oportunidad de conseguir un aprendizaje amplio sobre este tema. Aparte, considero que podría ser un proyecto que mejorase la calidad de vida de las personas.

Para llevar a cabo la idea, se planteó realizar una aplicación que integrase tanto la posibilidad de realizar redes aleatorias como la subida de ficheros que contuvieran la red real del colegio. A partir de esto, ya se procesarán las redes.

Este fue el proceso que nos llevó a la creación de Sire (Sibling Rewiring).

5.2 Metodologías

Se ha seguido una metodología SCRUM, metodología aprendida en la asignatura de “Gestión de Proyectos”. Esta metodología se basa en la gestión ágil para realizar el proyecto. Aunque debido a haberse realizado el proyecto de manera individual, se pueden reconocer ciertas diferencias con la metodología SCRUM tal y como se conoce.

El proyecto se ha dividido en diferentes Sprints, cada uno de ellos se subdividía en distintas tareas a las cuáles se les daba un tiempo estimado. A lo largo del proyecto se han mantenido reuniones al inicio y fin de cada Sprint. Al inicio para establecer las tareas que se debían llevar a cabo y al final de estos Sprints para revisar los avances obtenidos. En las reuniones se comentaban posibles fallos encontrados y se buscaban soluciones a estos.

Cada uno de los Sprints ha tomado una duración estimada de 2 semanas, que puede haber variado ligeramente si se encontraba la necesidad de una reunión previa o si las tareas han llevado más tiempo de lo esperado inicialmente.

Con la ayuda de Zenhub, se ha ido realizando un tablero por Sprint, donde podía tener una visión clara de las tareas conseguidas y pendientes, así como posibles bugs. Permitiéndome así tener una auto-gestión de un modo sencillo y sobre todo visual.

5.3 Formación

El lenguaje empleado en la programación del proyecto ha sido Python, lenguaje sobre el que se tenían ciertos conocimientos al haberse empleado en algunas asignaturas de la carrera, aún así, ciertas herramientas relacionadas con el mismo no habían sido utilizadas nunca por mi parte, por lo que gran parte de la realización del proyecto ha consistido en investigación.

E. Aspectos relevantes del desarrollo del proyecto

A continuación, se exponen algunos de estos ejemplos.

Flask

Para la realización del servidor se ha utilizado el *framework* Flask [23]. Es un *framework* escrito en Python que permite crear aplicaciones web de modo sencillo. Flask ha sido un entorno muy cómodo, puesto que permitía probar lo desarrollado en un entorno local sin necesidad de subirlo a web. Se optó por esta opción y no otras como Django, ya que fue recomendada por los tutores y tras buscar información me pareció una buena opción ya que me permitía la conexión con una base de datos de una forma sencilla. [24]

Bootstrap

Ya que nunca se había trabajado con programación de entornos web, tras realizar un par de pruebas sin utilizar una plantilla, se llegó a la conclusión de que la mejor opción sería utilizar una. En este caso se trabajó con Bootstrap, que ofrece multitud de plantillas a nuestra disposición. Antes de comenzar a usarlo, fue necesario buscar información, ya que era algo completamente nuevo. [25]

Diseño UI/UX

Diseño interfaz de usuario – experiencia de usuario. En este caso, resultó un apartado de interés propio que estaba enteramente ligado a la realización de este proyecto, ya que la aplicación se basa en una interacción directa del usuario con la misma. [26]

HTML y css

Aunque la mayor parte del trabajo se haya realizado en lenguaje Python, estos dos lenguajes han formado una gran parte del desarrollo web. El ‘html’ para realizar las modificaciones necesarias para la plantilla escogida y css para proporcionar estilos comunes. Anteriormente, no había realizado un aprendizaje sobre los mismos, por lo que fue necesario investigar para poder llevar a término el proyecto.

5.4 Desarrollo del proyecto

Tras comenzar la realización del proyecto, nos topamos con la primera decisión importante, cuál sería la base de datos que se podría utilizar para trabajar con Flask y que no diera demasiados problemas de conexión. Tras investigar y buscar información sobre otros proyectos realizados con este servidor que necesitasen de una base de datos, se llegó a la conclusión de utilizar MySQL. Para la conexión de la base de datos con el servidor se siguieron los siguientes pasos:

En primer lugar, se buscó como realizar la conexión directa entre Flask y MySQL.

```
mydb = mysql.connector.connect(host="localhost", user="root", passwd="Password", database="sire")  
  
mycursor = mydb.cursor()
```

Ilustración 6 - Conexión entre Flask y MySQL.

Tras ello se plantearon dos opciones, trabajar con ficheros que realizasen la creación de las tablas y procedimientos necesarios en ficheros externos o hacerlo desde el propio fichero de la conexión. Se optó por realizar la segunda opción para tener condensado todo lo necesario con la base de datos en un mismo sitio.

E. Aspectos relevantes del desarrollo del proyecto

```
def create_tables():
    mycursor.execute("CREATE TABLE IF NOT EXISTS tbl_user(user_id INT, name VARCHAR(50))")
    mycursor.execute("CREATE TABLE IF NOT EXISTS tbl_network(net_id INT, parent_id INT, child_id INT)")
    mycursor.execute("CREATE TABLE IF NOT EXISTS tbl_file(file_id INT, file_name VARCHAR(100))")

def create_procedure():
    mycursor.execute("DROP PROCEDURE IF EXISTS sp_createUser")
    file = open('procedure.txt', 'r')
    query = file.read()
    mycursor.execute(query)
```

Ilustración 7 - Creación de tablas y procedimientos.

Tras terminar la estructura del proyecto. Fue el momento de decidir cómo se iba a procesar la red y qué algoritmo se iba a utilizar para procesar esta. En un primer momento se plantearon dos modos de llegar a obtener la red. Una forma aleatoria, donde nosotros creásemos una red desde cero y otro modo de obtenerla que implicase la introducción de un fichero que contuviera una red real de un colegio. Se han terminado implementando ambas opciones.

La primera fase, la de la creación aleatoria de la red no produjo demasiados problemas una vez se fijaron nodos y enlaces, así como atributos necesarios. Para la segunda implementación, encontramos una pequeña traba por parte de la administración **¿no sé si debería explicar esto?**, aún así, tenemos un prototipo realista, que nos permite introducir ficheros que nos lleven a la red que nosotros necesitamos. Para realizar este segundo apartado, fue necesario investigar sobre la subida y descarga de ficheros a la aplicación.




Ilustración 8 - Subida de archivos.

Tras tener la red, era el momento de procesarla para obtener los resultados que nosotros necesitábamos, surgieron distintas opciones, un algoritmo que implementase el recocido simulado o un algoritmo evolutivo, ambos multicriterio. Siendo más amplios mis conocimientos sobre el recocido simulado tras haber cursado la asignatura de ‘Nuevas Tecnologías y Empresa’, esta opción fue la escogida, decidiendo implementar todas las secuencias de enfriamiento estudiadas.

E. Aspectos relevantes del desarrollo del proyecto

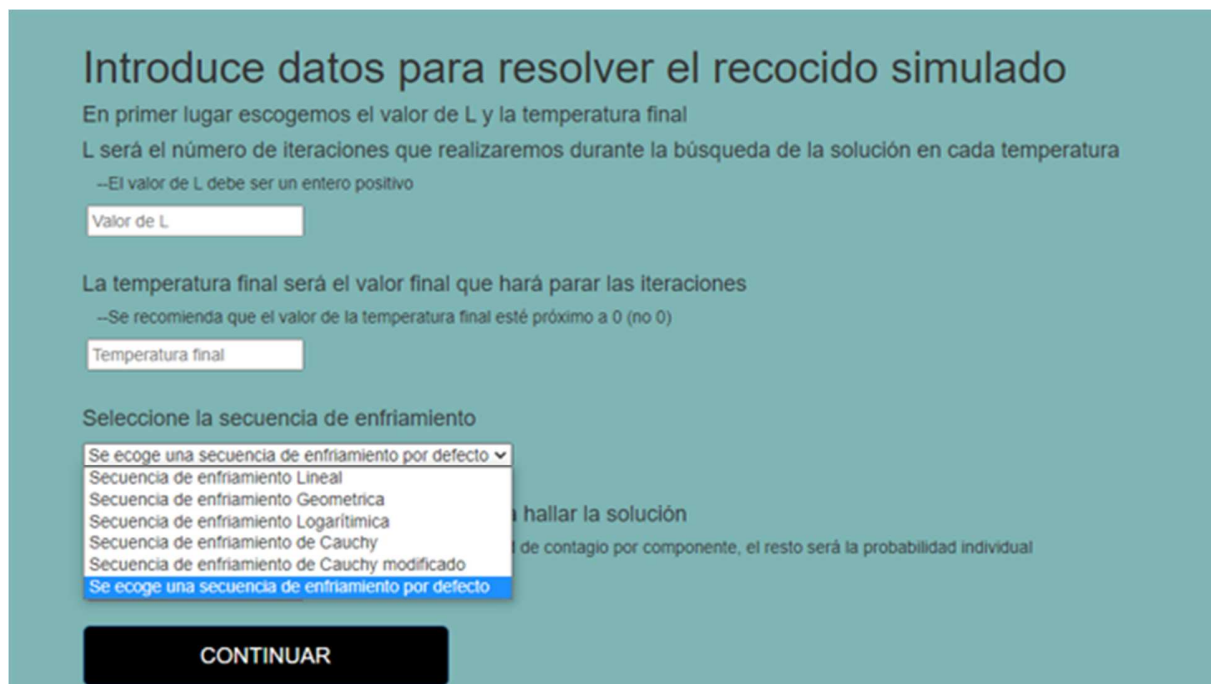


Ilustración 9 - Secuencias de enfriamiento.

Se ha tenido en cuenta que no todos los usuarios de nuestra aplicación podrían tener conocimientos suficientes sobre este algoritmo, por lo que se ha creado una versión que nos proporciona opciones por defecto.

Para terminar el proceso, nos muestra la mejor solución encontrada, con el cambio de alumnos y nos permite la posibilidad de descargar esta red.

5.5 Problemas técnicos encontrados

A lo largo del proyecto han ido surgiendo pequeños problemas, algunos siquiera reseñables, pero en cambio hay otros que merece la pena destacar por si pudieran servir para otras personas.

Problema de conexión con la base de datos

Una vez comenzamos con la conexión entre la base de datos y el servidor surgió un problema, ya que esta conexión no se establecía.

Fue específicamente al crear el procedimiento y tratar de ejecutar el mismo que surgía un error: “MySQL no se reconoce como comando interno o externo, programa o archivo por lotes ejecutable”. Tras recoger este problema, lo primero en lo que se pensó tras buscar información fue introducir la ruta de MySQL en el path [27]. Esto no funcionó.

Tras ello tratamos de comprobar si era un problema de permisos para el usuario en que nos encontrábamos, pero esto tampoco se correspondía con el error que estábamos encontrando.

Por último, vimos casos similares relacionados con MySQL connector [24], esta librería nos permite trabajar con MySQL desde Python. La solución relativa a esta nueva línea de acción consistió en desinstalar este, puesto que se podía haber instalado con una versión incompatible con nuestro entorno y tras ello, reinstalarlo. Esta fue la solución obtenida y la que dio resultado.

E. Aspectos relevantes del desarrollo del proyecto

Un hilo de ejecución secundario seguía activo tras finalizar el principal

Este problema surge del uso de pyplot para mostrar las redes en una ventana emergente [28]. En la ejecución del programa, se había implementado una opción de mostrar la red, en este caso, consistía en hacer un ‘matplotlib.pyplot.show()’ esto mostraba la red en una ventana, pero aunque esta fuera cerrada, no lo tomaba como tal y una vez se finalizaba la ejecución del programa principal, esta ventana seguía activa lo que daba problemas con volver a introducir una nueva red.

Para ello, se encontró una solución bastante sencilla. Consistía en hacer una importación de bibliotecas un poco diferente.

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
```

Ilustración 10 - Importación Matplotlib.

Esto, nos permite seguir utilizando la librería ‘matplotlib’ con la extensión ‘pyplot’ pero consiguiendo que no cree hilos adicionales, ya que no permite la apertura de elementos externos a la aplicación.

Heroku no permite alojar la aplicación

Se ha tratado de realizar un despliegue de nuestra aplicación en un entorno web, intentando usar Heroku [29] para alojarla.

Se ha observado que existen distintas opciones para realizar el despliegue, utilizando Git [30] o partiendo de un repositorio de Github. Se han intentado ambos procedimientos tanto desde la cmd como desde la aplicación web.

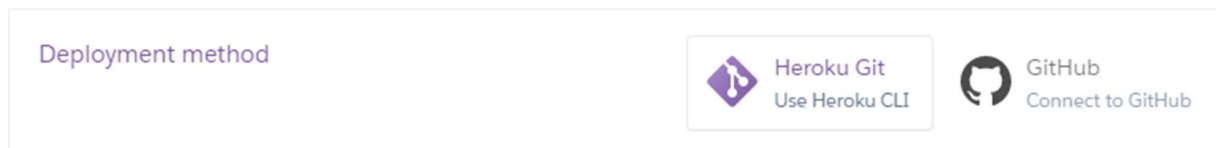
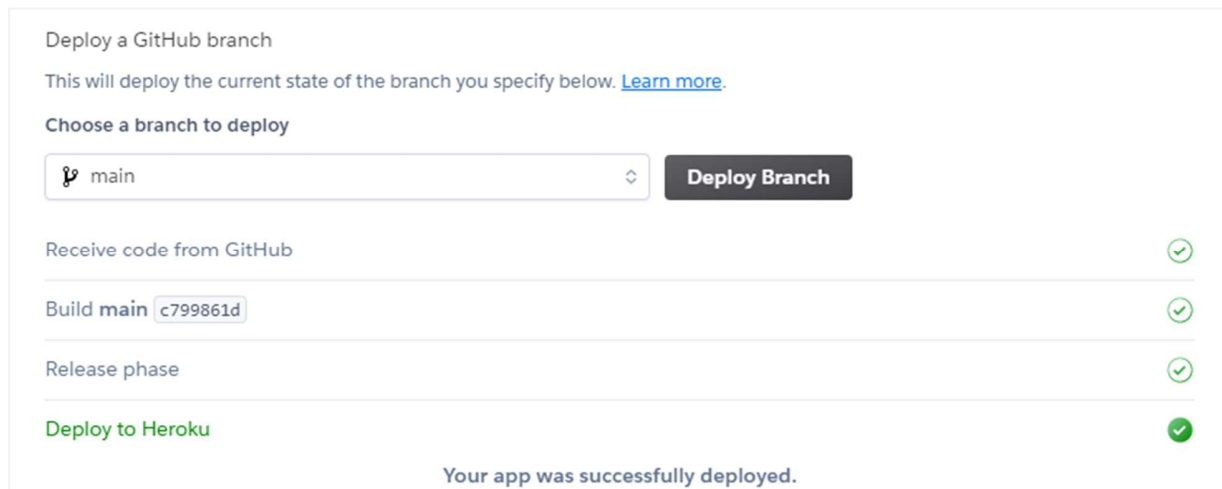


Ilustración 11 - Opciones de despliegue de Heroku.

Realizando ambos procedimientos, nos muestra que el proyecto se ha desplegado, aunque al tratar de abrir la aplicación, aparece que se ha producido un error. Mostramos a continuación los pasos seguidos y los errores que se han ido encontrando.

Utilizando el método de GitHub de forma manual, podemos observar lo siguiente:

E. Aspectos relevantes del desarrollo del proyecto



Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

Deploy Branch

Receive code from GitHub ✓

Build main c799861d ✓

Release phase ✓

Deploy to Heroku ✓

Your app was successfully deployed.

Ilustración 12 - Despliegue Manual con GitHub.

Pero una vez abrimos la aplicación, nos muestra que ha surgido un error.

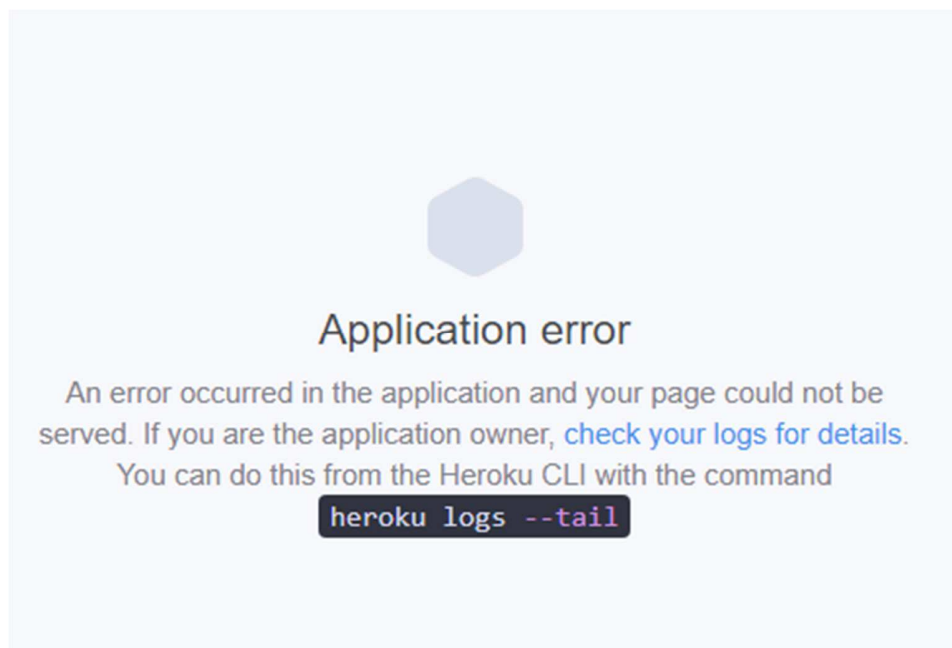


Ilustración 13 - Error de apertura de aplicación.

Y a continuación, mostraremos el proceso realizado desde la cmd. En primer lugar, nos conectaremos a heroku realizando un login.

```
E:\Users\sony\TFG\sire\Sibling-Rewiring\Environment\sire>heroku login
» Warning: heroku update available from 7.53.0 to 7.56.0.
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/6fece021-0305-4004-804d-00ee0b1fa057?requestor=SFMyNTY.g
2gDbQAAAA05MC4xNjcuMjU0LjcxYjYAJfBvd3oBYgABUYA.pm5PM2AgOCANxzAHWxz1cXk9j0DkktdDSqmvItSUyRc
Logging in... done
Logged in as mor1001@alu.ubu.es
```

Ilustración 14 - Login en Heroku.

E. Aspectos relevantes del desarrollo del proyecto

Tras ello, creamos el apartado git, carpeta contenedora de los procedimientos, desde donde procesaremos la información, obteniendo 'requirements.txt' y 'runtime.txt', ficheros necesarios para arrancar el procedimiento, a parte, 'Procfile' fichero que contiene el tipo de aplicación, en nuestro caso web y el main.

```
E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku create sibling_rewiring
» Warning: heroku update available from 7.53.0 to 7.56.0.
Creating sibling_rewiring... !
! Name must start with a letter, end with a letter or digit and can only contain lowercase letters, digits, and dashes.

E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku create siblingrewiring
» Warning: heroku update available from 7.53.0 to 7.56.0.
Creating siblingrewiring... done
https://siblingrewiring.herokuapp.com/ | https://git.heroku.com/siblingrewiring.git

E:\Users\sony\TFG\sire\Sibling-Rewiring>git remote -v
heroku https://git.heroku.com/siblingrewiring.git (fetch)
heroku https://git.heroku.com/siblingrewiring.git (push)

E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku git:remote -a siblingrewiring
» Warning: heroku update available from 7.53.0 to 7.56.0.
set git remote heroku to https://git.heroku.com/siblingrewiring.git
```

Ilustración 15 - Crear aplicación.

Tras realizar estos pasos, tendríamos nuestra aplicación, con un procedimiento web activo. El problema es que al intentar ejecutar el comando 'heroku open' llegamos al punto obtenido en la Ilustración 13, donde encontramos que la aplicación intenta arrancar, pero nos aparece que ha ocurrido un error, entonces, si comprobamos desde la cmd, podemos observar que el proceso web ha sido 'roto'.

```
E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku ps
» Warning: heroku update available from 7.53.0 to 7.56.0.
Free dyno hours quota remaining this month: 548h 33m (99%)
Free dyno usage for this app: 0h 0m (0%)
For more information on dyno sleeping and how to upgrade, see:
https://devcenter.heroku.com/articles/dyno-sleeping

=== web (Free): gunicorn app (1)
web.1: crashed 2021/07/05 18:57:17 +0200 (~ 5m ago)
```

Ilustración 16 - Web crashed.

Una vez llegamos a este punto, se produjo una búsqueda de información. Después de investigar se observó que sugerían revisar los logs de Heroku. En estos, se observa que la aplicación pasa de estar iniciada a estar 'rota' y tras ello, aparece que el error resultante es el H-10. [31]

```
2021-07-05T18:57:17.032452+00:00 heroku[web.1]: Process exited with status 1
2021-07-05T18:57:17.923765+00:00 heroku[web.1]: State changed from starting to crashed
2021-07-05T17:07:43.738447+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=siblingrewiring.herokuapp.com request_id=fb287e5f-cb5e-4
1d2-a9cd-183f47214560 fwd="84.78.240.143" dyno= connect= service= status=500 bytes= protocol=https
2021-07-05T17:07:44.360301+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/favicon.ico" host=siblingrewiring.herokuapp.com request_id=70c7
3147-20bf-4ed2-b92a-4b935df21eb5 fwd="84.78.240.143" dyno= connect= service= status=500 bytes= protocol=https
2021-07-05T17:07:49.940417+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/" host=siblingrewiring.herokuapp.com request_id=30a71b82-0313-4
837-b279-b6a23d74d16d fwd="84.78.240.143" dyno= connect= service= status=500 bytes= protocol=https
2021-07-05T17:07:50.212620+00:00 heroku[router]: at=error code=H10 desc="App crashed" method=GET path="/favicon.ico" host=siblingrewiring.herokuapp.com request_id=922a
c2ae-dc7b-4f8f-bab7-04b7ab954d48 fwd="84.78.240.143" dyno= connect= service= status=500 bytes= protocol=https
```

Ilustración 17 - Logs Heroku error H-10.

Si vamos al propio manual de códigos de error de Heroku, nos explica que este puede significar que la web se ha 'roto' o que se ha excedido el tiempo de espera de arranque de la web. Al utilizarse el servicio gratuito, encontramos que la única opción que podíamos probar sería reiniciar la aplicación y tratar de escalar de nuevo la web, pasando esta a un nivel inferior para después devolver a la inicial. La otra opción que se encontró fue volver a realizar el despliegue desde cero.

Entonces, se trataron de realizar estos escalados [32] como se muestra en la siguiente imagen.

E. Aspectos relevantes del desarrollo del proyecto

```
E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku restart
» Warning: heroku update available from 7.53.0 to 7.56.0.
Restarting dynos on ☐ siblingrewiring... done

E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku ps:scale web=0
» Warning: heroku update available from 7.53.0 to 7.56.0.
Scaling dynos... done, now running web at 0:Free

E:\Users\sony\TFG\sire\Sibling-Rewiring>heroku ps:scale web=1
» Warning: heroku update available from 7.53.0 to 7.56.0.
Scaling dynos... done, now running web at 1:Free
```

Ilustración 18 - Escalado de la web.

Aunque el resultado obtenido fue el mismo.

Uso del framework deap

Tras terminar la implementación de nuestro algoritmo, el recocido simulado, nos propusimos llevar a cabo una mejora, introduciendo la implementación de un algoritmo evolutivo multicriterio [33] [34]. Y para ello, se decidió implementar un framework, en este caso, deap. [35]

Una vez comenzamos a trabajar con deap, se decidió implementar el algoritmo SPEA [36] y como se comprobó que deap permitía su versión mejorada, SPEA2 [37], se optó por este último.

Tras comenzar a implementarlo, vimos que surgía un problema con nuestro proyecto, en nuestro caso particular, necesitamos que la población que obtengamos sean variaciones de una misma red de un modo acotado. La población deberá estar formada por individuos, cada uno de estos individuos será una red, un posible resultado a nuestro problema y, por lo tanto, la distribución de los hermanos debe ser equivalente, pudiendo así cambiar únicamente la clase a la que pertenecen, sin tocar el curso o la etapa.

Entonces, al entrar en materia y comenzar a buscar ejemplos sobre deap y la creación de las poblaciones [38], vimos que podríamos haber trabajado con una matriz, que sería equivalente a la red, pero nos obligaba a obtenerla de modo aleatorio, lo que podría conseguir un escenario irreal. Este modo aleatorio, lo que causaría, es que no podemos controlar los enlaces relativos a los hermanos explicados anteriormente.

Por estos motivos, se descartó la posibilidad de implementar este framework.

5.6 Wiki

Un aspecto reseñable de la aplicación sería la creación de una Wiki que funciona como manual de usuario. Ha supuesto una inversión de tiempo, pero nos ha parecido un accesorio muy útil para nuestra aplicación, ya que es algo accesible y que permite al usuario entender de un modo más simple el funcionamiento de la aplicación. [39]

Ha supuesto un aprendizaje, ya que nunca se había escrito nada similar. Además, esta hace que la aplicación sea más realista, ya que, si un usuario tiene problemas con el uso de la aplicación, lo más intuitivo será buscar esta información en la misma aplicación y es lo que nos permite la wiki.

Se puede acceder a esta a través del siguiente enlace: [wiki](#).

F. Trabajos relacionados

Al inicio del proyecto, se realizó una búsqueda previa sobre posibles trabajos similares, que nos pudieran ayudar a la hora de la toma de decisiones. Que mostraran el modo de implementar los distintos algoritmos, estilo de la página, base de datos, cómputo y análisis de los resultados...

Nuestra sorpresa fue que no se encontró ningún proyecto que tuviera una temática similar o, aunque tuviera una temática ligeramente distinta, su estilo de programación parecido.

Por lo tanto, podemos denotar como punto a favor en este trabajo su carácter innovador, ya que se ajusta perfectamente a la situación actual y trata un problema que está afectando a un amplio número de personas. Ha surgido la idea debido al alto número de contagios surgidos por la pandemia del COVID – 19, pero se podría exportar a cualquier enfermedad contagiosa por proximidad o por contacto en el aire e incluso, está pensado para llevarse a colegios, pero se podría ampliar a otros sectores.

G. Conclusiones y líneas de trabajo futuras

En este apartado expondremos las conclusiones junto al aprendizaje obtenidos tras finalizar este proyecto y las posibles líneas de trabajo futuras.

7.1 Conclusiones

Tras finalizar el proyecto, si echo la vista atrás, puedo afirmar que mis conocimientos han sido ampliamente desarrollados.

Desde la perspectiva de la gestión de un proyecto, ha supuesto un gran aprendizaje, ya que los trabajos de la universidad nunca habían sido a esta escala. En este caso, teniendo presente en todo momento el estilo de metodología SCRUM, que nos ha permitido una gestión controlada del tiempo y un buen modo de fijar objetivos.

El resultado de la aplicación cumple con creces los objetivos fijados inicialmente. En un primer momento, no se había tenido en cuenta realizar una base de datos para guardar nuestras redes y así después poder explotar los resultados obtenidos. También, se había considerado únicamente el procesamiento de ficheros, sin tener en cuenta la parte de la red aleatoria, que se terminó dejando para que usuarios externos al mundo escolar, pudieran probar e investigar el funcionamiento de nuestra aplicación.

En cuanto a mis objetivos personales, también los considero cumplidos, siempre había tenido inquietud sobre el desarrollo web y aunque no me ha resultado demasiado sencillo, ha supuesto un punto de partida. Además, he conseguido ampliar los conocimientos sobre las bases de datos reforzando así los conocimientos previos de la carrera. Al igual que con SQL, se ha conseguido fijar con mayor claridad los conocimientos obtenidos en ‘Nuevas Tecnologías y Empresa’ sobre las redes, grafos y sus componentes. Así como mejorar y profundizar con el lenguaje de programación Python.

En resumen, el balance es positivo y en mi opinión se ha creado una aplicación que podría explotarse y generar resultados beneficiosos para algo como es la minimización de contagio de enfermedades.

7.2 Líneas de trabajo futuras

Mientras se realizaba el proyecto, han surgido ciertas posibilidades de mejora que podrían enriquecer la aplicación ya creada. En este apartado se expondrán algunas de ellas.

Alojar la aplicación en web

Realizar un despliegue de la aplicación en un entorno web como podría ser heroku. Esto favorecería la accesibilidad de los posibles usuarios a la aplicación.

Implementar de una solución alternativa

Implementar un algoritmo evolutivo multicriterio que generase una posible solución alternativa a la conseguida gracias al recocido simulado. Para una vez generadas ambas, realizar una comparación de las mismas y seleccionar la más adecuada para el problema según lo que estemos priorizando.

Mostrar el tiempo restante

En el procesado de las redes en la aplicación, mostrar al usuario el tiempo restante mientras la aplicación lleva a cabo los cálculos. Esto haría que el usuario en todo momento conozca la situación.

G. Conclusiones y líneas de trabajo futuras

Realizar encriptados

A la hora de guardar cierta información en la base de datos, esta podría ser encriptada para generar una mayor seguridad al usuario.

Internacionalizar

Llevar a cabo la internacionalización de la aplicación para que sea accesible a un mayor número de usuarios, ya que este problema se encuentra a nivel global.

Bibliografía

- [1] Pandemia Coronavirus (OMS), (n.d.). <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019> (accessed June 29, 2021).
- [2] N. M. E., Newman, M. E. (2003). The structure and function of complex networks. SIAM review, 45(2), 167-256., in: Struct. Funct. Complex Netw., 2003: pp. 167–256.
- [3] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P., Optimization by simulated annealing., in: Optim. Simulated Annealing, 1983: pp. 671–680.
- [4] Santos, J.I., Pereda, M., Ahedo, V., Galán, J.M, Herramienta docente en NetLogo para ilustrar el proceso de enfriamiento en el recocido simulado mediante el modelo de Metrópolis. En Proceedings of the 15th International Conference on Industrial Engineering and Industrial Management and XXV Congreso de Ingeniería de Organización., (2021). <https://cioxxv.pressbooks.com/chapter/herramienta-docente-en-netlogo-para-ilustrar-el-proceso-de-enfriamiento-en-el-recocido-simulado-mediante-el-modelo-de-metropolis/>.
- [5] Talbi, E. G., Metaheuristics: from design to implementation (Vol. 74)., John Wiley & Sons., 2009.
- [6] O.C. García, Redes y Sistemas Complejos Cuarto Curso del Grado en Ingeniería Informática, (n.d.) 92.
- [7] M. Mocholí, Nodos círculos Arcos (flechas) sólo es posible desplazarse en un solo sentido Aristas (líneas) es posible desplazarse en ambos sentidos (2,3), (5,6), (n.d.) 25.
- [8] Conectividad (teoría de grafos) - Wikipedia, la enciclopedia libre, (n.d.). [https://es.wikipedia.org/wiki/Conectividad_\(teor%C3%ADa_de_grafos\)](https://es.wikipedia.org/wiki/Conectividad_(teor%C3%ADa_de_grafos)) (accessed June 29, 2021).
- [9] Juan Palacio, Claudia Ruata, Scrum Manager. Gestión de proyectos. Rev 1.4., (n.d.).
- [10] GitHub, GitHub. (n.d.). <https://github.com> (accessed June 30, 2021).
- [11] ZenHub - Agile Project Management for GitHub, (n.d.). <https://www.zenhub.com/> (accessed June 30, 2021).
- [12] GitHub Desktop, GitHub Deskt. (n.d.). <https://desktop.github.com/> (accessed June 30, 2021).
- [13] Microsoft Word: software de procesamiento de texto | Microsoft 365, (n.d.). <https://www.microsoft.com/es-es/microsoft-365/word> (accessed June 30, 2021).
- [14] Introduction to LaTeX, (n.d.). <https://www.latex-project.org/about/> (accessed November 13, 2020).
- [15] Por qué Apache OpenOffice, (n.d.). <https://www.openoffice.org/es/por-que/> (accessed June 30, 2021).
- [16] Zotero - Zotero y recursos CEPAL - Biblioguias at Biblioteca CEPAL, Naciones Unidas, (n.d.). <https://biblioguias.cepal.org/zotero> (accessed June 30, 2021).
- [17] Python, (n.d.). <https://www.python.org/> (accessed July 5, 2021).
- [18] Welcome to Flask — Flask Documentation (1.1.x), (n.d.). <https://flask.palletsprojects.com/en/1.1.x/> (accessed November 17, 2020).
- [19] M.O. contributors Jacob Thornton, and Bootstrap, Bootstrap, (n.d.). <https://getbootstrap.com/> (accessed June 30, 2021).
- [20] NetworkX — NetworkX documentation, (n.d.). <https://networkx.org/> (accessed June 30, 2021).
- [21] MySQL, (n.d.). <https://www.mysql.com/> (accessed June 30, 2021).
- [22] Python MySQL Create Table, (n.d.). https://www.w3schools.com/python/python_mysql_create_table.asp (accessed December 19, 2020).
- [23] Installation — Flask Documentation (1.1.x), (n.d.). <https://flask.palletsprojects.com/en/1.1.x/installation/> (accessed November 18, 2020).
- [24] B. Colen, flask-mysql-connector: Easy to use MySQL client for Flask apps., n.d. https://github.com/brandenc40/flask_mysql_connector (accessed November 27, 2020).
- [25] Flask Tutorial - Adding Bootstrap & Template Inheritance, Techwithtim.Net. (n.d.). <https://www.techwithtim.net/tutorials/flask/adding-bootstrap/> (accessed March 10, 2021).

Bibliografía

- [26] Crehana, Crehana, Recursos Gráficos, Crehana. (n.d.). <https://www.crehana.com/recursos/> (accessed July 2, 2021).
- [27] Como arreglar “mysql no se reconoce como un comando interno o externo, programa o archivo por lotes ejecutable,” Platzi. (n.d.). <https://platzi.com/tutoriales/1272-sql-mysql/2256-como-arreglar-mysql-no-se-reconoce-como-un-comando-interno-o-externo-programa-o-archivo-por-lotes-ejecutable/> (accessed December 4, 2020).
- [28] python matplotlib: Plotting to GUI in non-main thread, (n.d.). <http://5.9.10.113/65247422/can-i-set-main-thread-in-python> (accessed July 2, 2021).
- [29] Heroku - Una plataforma para la creación de aplicaciones - Esteban Romero, (n.d.). <https://estebanromero.com/herramientas-emprender-desarrollar-proyectos/heroku-una-plataforma-para-la-creacion-de-aplicaciones/> (accessed July 5, 2021).
- [30] Git, (n.d.). <https://git-scm.com/> (accessed July 5, 2021).
- [31] Heroku Error Codes | Heroku Dev Center, (n.d.). <https://devcenter.heroku.com/articles/error-codes#h10-app-crashed> (accessed July 5, 2021).
- [32] Scaling Your Dyno Formation | Heroku Dev Center, (n.d.). <https://devcenter.heroku.com/articles/scaling> (accessed July 5, 2021).
- [33] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Pub., Second Edition, 2007, 2002.
- [34] John Wiley & Sons, K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, 2001.
- [35] DEAP/deap: Distributed Evolutionary Algorithms in Python, (n.d.). <https://github.com/DEAP/deap> (accessed April 1, 2021).
- [36] Zitzler, E., Deb, K., Thiele, L., Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation Journal 8, in: 2000: pp. 125–148.
- [37] Eckart Zitzler, Marco Laumanns, Lothar Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm., n.d.
- [38] DEAP documentation — DEAP 1.3.1 documentation, (n.d.). <https://deap.readthedocs.io/en/master/> (accessed May 18, 2021).
- [39] 262588213843476, Matplotlib Backends #matplotlib #python, Gist. (n.d.). <https://gist.github.com/CMCDragonkai/4e9464d9f32f5893d837f3de2c43daa4> (accessed July 3, 2021).