TRABAJO PRÁCTICO 3: MATEMÁTICA DISCRETA

CURSO 2022-2023

**Memoria**

**GPS**

Autores:

**Lucía Prado Fernández-Vega**

**Lara Ocón Madrid**

**GP02**

Madrid, enero, 2023

# Introducción

Este documento expone la memoria del programa criptochat.py, chat cifrado de extremo a extremo. El programa permite generar un par de claves para la librería RSA, o introducir unas claves preexistentes tanto para el origen como para el destinatario de los mensajes, pudiendo mantener una conversación cifrada entre dos participantes. Para ello, implementamos la librería rsa.py con la funcionalidad básica de un sistema RSA. Contiene las funciones necesarias para crear claves RSA y utilizarlas para cifrar y descifrar mensajes numéricos o de texto empleando un esquema de padding básico. También contiene funciones que permiten realizar ataques básicos sobre un sistema RSA pequeño o sin padding.

.

# Implementación del TAD en grafo.py

A la hora de implementar el TAD generado en la librería grafo.py hemos creado dos grafos. Ambos guardan las mismas aristas y vértices con la misma información pero con una pequeña diferencia, en uno consideramos el peso de las aristas como la distancia de un vértice a otro, y en el segundo, consideramos que el peso es el tiempo que se tarda en recorrer dicha arista (longitud de la arista partido de la velocidad máxima de circulación permitida en dicha arista).

De esta forma a la hora de comenzar a recorrer el csv de cruces, hemos creado dos grafos:

# Funcion iniciar ( ) gps.py

AQUí explicamos como hemos reconstruido el grafo a partir de los csvs.

## Funciones…

# Estructura gps.py (resto de funciones)

Este es el programa principal del chat encriptado...

## Class Usuario

…

## Generar par claves

# Búsqueda de la ruta óptima

Aquí explicamos las funciones que buscan los vértices origen y destino a partir de unos inputs, y como calculamos la ruta optima.

Despues explicamos como a partir de los vértices obtenidos en ruta optima reconsturimos el camino y las direcciones (gira a la derecha/a la izquierda/ durante cuantos metros permanecer en x calle…) y como imprimimos la ruta usando network x.

# Bibliografía

* [Cipolla's algorithm - Wikipedia](https://en.wikipedia.org/wiki/Cipolla%27s_algorithm#:~:text=Cipolla%27s%20algorithm%20In%20computational%20number%20theory%2C%20Cipolla%27s%20algorithm,of%20x%2C%20and%20where%20is%20an%20odd%20prime.)
* [Algoritmo de Cipolla AlgoritmoyEjemplo (hmong.es)](https://hmong.es/wiki/Cipolla%27s_algorithm)
* [▷Cómo resolver esta ecuación de congruencia cuadrática ✔️ Foro Ayuda 【 2022 】](https://foroayuda.es/como-resolver-esta-ecuacion-de-congruencia-cuadratica/)
* <https://mathworld.wolfram.com/LegendreSymbol.html>
* <https://www.youtube.com/watch?v=d7ZFCf95MAQ&t=1171s>