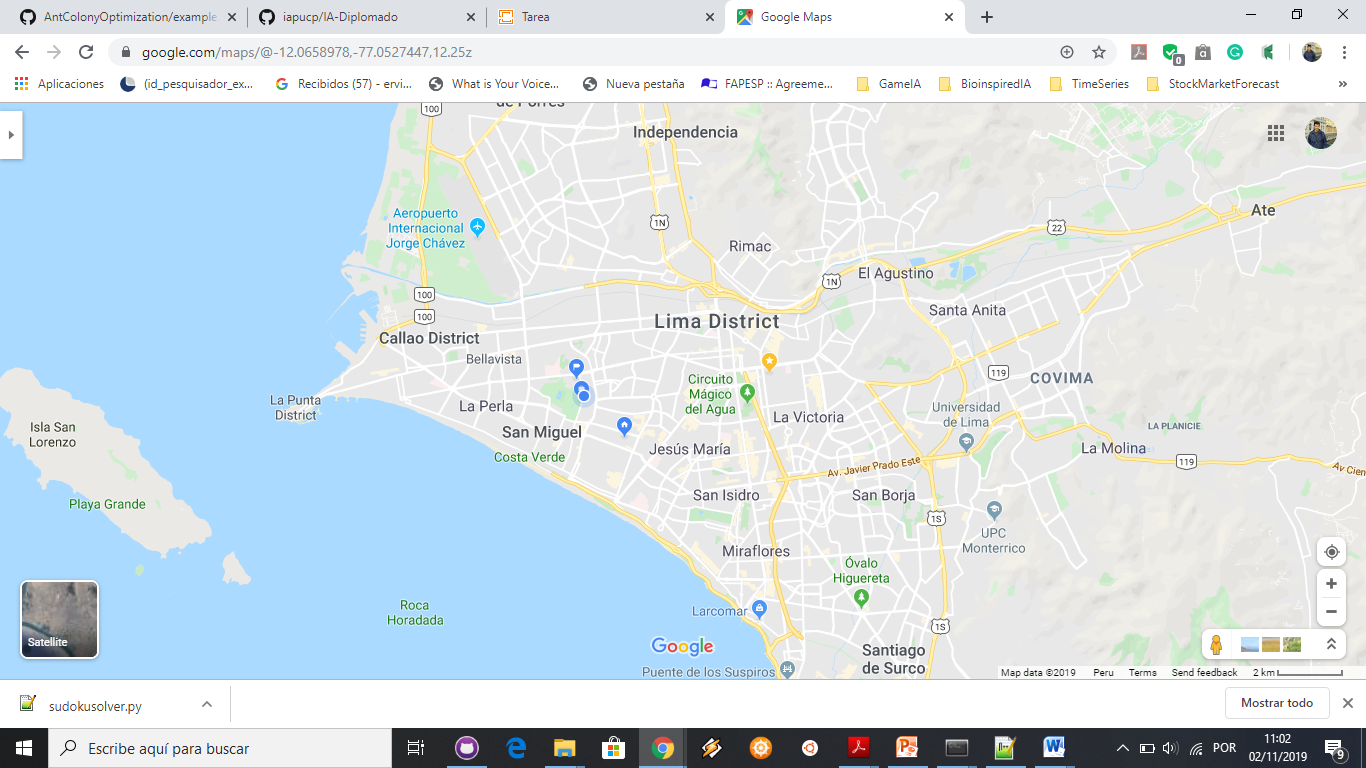
# Diplomatura de Especialización en Desarrollo de Aplicaciones con IA

**Optimizacion Industrial con Computacion Evolutiva**

**Proyecto Final: Usando computación evolutiva para encontrar localizaciones optimas de una cadena de supermercados**

Una conocida cadena de supermercados desea ingresar al mercado limeño. Usted coomo experto en Inteligencia Artificial ha sido contratado para proponer donde deben estar instalados los supermercados. Para ello, se le está proporcionando a usted un mapa de la ciudad con L posibles locales donde la cadena de supermercados podría localizar sus supermercados. La cadena tiene presupuesto para instalar hasta S <L supermercados. Se le proporciona también una tabla con las coordenadas de los L posibles localizaciones y la población estimada que hay en 500 metros a la redonda.

Su objetivo es localizar los S supermercados de manera que se maximice la población atendida que vive alrededor de cada supermercado y la suma de distancias entre los S supermercados.



**TIPS:**

* Se recomienda representar cada individuo con un cromosoma de 9 genes, cada gen representando un bloque diferente del tablero. Mantener siempre **alelos válidos** (9 números diferentes en el bloque)
* En el cruzamiento, los genes (bloques) pueden ser considerados indivisibles. Así, si se hace cruzamiento single point y la posición de cruzamiento es entre el 3er y 4to gen, entonces un individuo hijo heredaría los 3 primeros bloques de un padre y los otros 6 bloques del otro padre. El otro hijo seria creado de forma recíproca. Para cruzamiento uniforme se puede hacer de manera análoga
* La mutación de un individuo debe alterar solo un gen del individuo (escogido aleatoriamente) y debe producir un nuevo alelo válido (respetando no alterar los numeros del tablero original). Puede usar el método make\_neighborBoard() para ello.
* Los individuos iniciales pueden ser creados basándose en el método randomAssign()

**Bonus:**

* Implementar más de un operador de cruzamiento y/o mutación y comparar sus desempeños
* Implementar alguna forma más eficiente del algoritmo genético (demostrar su efectividad en los resultados). Aquí algunas ideas:
  + Ajuste dinámico de la taza de mutación de acuerdo al grado de convergencia que se tiene
  + Usar un pool de operadores de cruzamiento y mutación y aplicarlos de acuerdo a su eficacia en las generaciones anteriores;
  + Cruzamiento con más de dos padres

**Entregables:**

1. Archivos findroute.py y sudokusolver.py para verificar la obtención de los resultados
2. Informe conteniendo:

## Breve descripción de su algoritmo (representación de individuos, operadores de cruzamiento y mutación, etc). Si ha hecho desarrollos para ganar puntos bonus debe indicarlo explícitamente aquí como “Desarrollos para puntos bonus”

## Tablas de resultados para los 3 tableros dados. Si hay algún desarrollo de puntos bonus, extender las tablas de resultados con los resultados de dichos desarrollos.

## Análisis de resultados, comparando los desempeños de las diferentes configuraciones experimentadas.

## Conclusión del mejor método y configuración de parámetros a ser usada para resolver tableros de sudoku.