



**Universidad Autónoma del Estado de México**

Centro Universitario UAEM Valle de Chalco

**DESARROLLO DE UN ALGORITMO EVOLUTIVO PARA  
LA PERFILACIÓN GEOGRÁFICA CRIMINAL**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE**

**DOCTOR EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A**

MTRO. PABLO VERA GONZÁLEZ

**TUTORA ACADÉMICA**

DRA. CRISTINA JUÁREZ LANDÍN

TUTOR ADJUNTO

DR. JOSÉ SERGIO RUIZ CASTILLA

TUTOR ADJUNTO

DR. ROGELIO REYES REYES

**VALLE DE CHALCO SOLIDARIDAD, MÉXICO OCTUBRE 2021.**



**CUVCH**



# **DESARROLLO DE UN ALGORITMO EVOLUTIVO PARA LA PERFILACIÓN GEOGRÁFICA CRIMINAL**

## Resumen

Esta tesis aplicó un algoritmo evolutivo desarrollado para la elaboración de perfiles geográficos delictivos. El algoritmo se basó en algoritmos genéticos, programación evolutiva y estrategias evolutivas. Las técnicas anteriores se aplicaron sobre un algoritmo para la elaboración de perfiles geográficos delictivos. El objetivo fue seleccionar individuos con características similares. El algoritmo evolutivo considera un cambio en las soluciones en caso de ingresar mínimos o máximos locales. Además, la selección de nuevos individuos de la población en caso de salir de los parámetros de evaluación. Las pruebas y la evaluación del algoritmo propuesto se obtuvieron heurísticamente. Los hechos considerados coincidieron con las características de la mayoría de la población inicial y con estos resultados. El algoritmo se puede utilizar como una herramienta adicional en la prevención del delito. Las pruebas y la evaluación del algoritmo propuesto se obtuvieron heurísticamente. Los hechos considerados coincidieron con las características de la mayoría de la población inicial y con estos resultados. El algoritmo se puede utilizar como una herramienta adicional en la prevención del delito.



## **Abstract**

This thesis applied an evolutionary algorithm developed for the criminal geographic profiling. The algorithm was based on genetic algorithms, evolutionary programming and evolutive strategies. The previous techniques were applied on an algorithm for the criminal geographic profiling. The objective was to select individuals with similar characteristics. The evolutionary algorithm considers a change in the solutions in case of entering local minimums or maximums. In addition to, the selection of new individuals from the population in case of leaving the evaluation parameters. The tests and evaluation of the proposed algorithm were obtained heuristically. The events considered coincided with the characteristics of the majority of the initial population and with these results. The algorithm can be used as an additional tool in crime prevention. The tests and evaluation of the proposed algorithm were obtained heuristically. The events considered coincided with the characteristics of the majority of the initial population and with these results. The algorithm can be used as an additional tool in crime prevention.

## Índice general

Resumen .....	IV
Abstract.....	V
Índice de figuras y gráficas .....	VIII
Capítulo 1. Introducción .....	2
<b>1.1 Motivación.....</b>	<b>3</b>
<b>1.2 Planteamiento del problema .....</b>	<b>4</b>
<b>1.3 Justificación .....</b>	<b>5</b>
<b>1.4 Hipótesis .....</b>	<b>8</b>
<b>1.5 Objetivos.....</b>	<b>9</b>
<b>1.5.1 Objetivo General.....</b>	<b>9</b>
1.5.2 Objetivos específicos .....	9
Capítulo 2. Antecedentes .....	10
<b>2.1 Uso de técnicas de inteligencia artificial en perfilación geográfica criminal .....</b>	<b>10</b>
<b>2.2 Perfilación geográfica criminal .....</b>	<b>11</b>
<b>2.3 Perfilaciones Generales.....</b>	<b>13</b>
Capítulo 3. Marco teórico y conceptual .....	14
<b>3.1 Delitos.....</b>	<b>14</b>
3.1.1 Clasificación del delito .....	14
3.1.2 Tipos de Prevención del delito .....	16
<b>3.2 Inteligencia Artificial .....</b>	<b>21</b>
3.2.1 Tipos de inteligencia artificial .....	21
<b>3.2.2 Áreas de la inteligencia artificial .....</b>	<b>23</b>
<b>3.3 Algoritmos.....</b>	<b>28</b>
3.3.1 Clasificación de los algoritmos.....	30
<b>3.4 Introducción al cómputo evolutivo .....</b>	<b>31</b>
3.4.1 Programación evolutiva .....	36
3.4.2 Estrategias evolutivas .....	37
3.4.3 Algoritmos Genéticos .....	38
Capítulo 4. Procedimiento y metodología.....	39
<b>4.1 Metodología de investigación con cómputo evolutivo para la perfilación geográfica criminal .....</b>	<b>39</b>
4.1.1 Tipo de investigación empleada .....	45
<b>4.2 Descripción del algoritmo evolutivo para la perfilación geográfica criminal .....</b>	<b>46</b>

- 4.2.1 Análisis general del problema y determinación de variables .....48
- 4.2.2 Algoritmo evolutivo para la perfilación geográfica criminal .....49
- 4.2.3 Representación de la población.....53
- 4.2.4 Proceso de Mutación .....56
- 4.2.5 Selección de individuos.....57
- 4.2.6 Recombinación en la población .....58
- 4.2.7 Evaluación de aptitud..... 60**
- Capítulo 5. Resultados .....62
- Capítulo 6. Conclusiones..... 71
- 6.1 Recomendaciones..... 72**
- 6.2 Trabajos futuros ..... 73**
- Apéndice ..... 74
- A.1 Código fuente del algoritmo en Python ..... 74**
- A.1.1 Clase para cargar la información de un archivo de texto y crear la matriz principal de los eventos delictivos ..... 74**
- A.1.2 Clase para preparar la población inicial y optimizarla..... 76**
- A.1.3 Heurística y análisis de datos ..... 83**
- A.1.4 Mutación..... 96**
- A.2 Participaciones y publicaciones..... 98**
- A.2.1 Correo de aceptación de artículo para publicación en revista..... 98**
- A.2.3 Presentación de artículo “Implementation of an evolutionary algorithm in the creation of criminal geographic profiles”, en MICAI, octubre 2021..... 100**
- A.2.2 Participación en el 22 seminario de investigación en la Universidad Autónoma de Aguascalientes, julio 2021. .... 101**
- A.2.3 Entrevista en TV Azteca, ADN40 “Perfilación sobre delitos cibernéticos”, mayo 2020. .... 102**
- A.2.1 Publicación de libro “Inteligencia artificial como herramienta en la prevención de delitos” ..... 103**
- Referencias .....104

## Índice de figuras y gráficas

Figura 1.3.1 Macro localización de la zona de estudio, Estado de México (Google, 2021). .....	6
Figura 1.3.2 Micro localización de la zona de estudio, Valle de Chalco (Google, 2021). .....	6
Figura 3.1 Representación de cromosoma, gen y alelo (Coello, 2019) .....	34
Figura 3.2 Representación conceptual de programación evolutiva (Coello, 2019) .....	36
Figura 3.3 Representación conceptual de estrategias evolutivas (Coello, 2019) .....	37
Figura 3.4 Representación conceptual de AG (Hernández et al., 2018) .....	38
Figura 4.1 Esquema general de la investigación empleada para el desarrollo del algoritmo evolutivo para la perfilación geográfica criminal (propuesta por el autor) .....	45
Figura 4.2 Algoritmo general .....	46
Figura 4.3 Representación conceptual de algoritmo evolutivo (Propuesta por el autor) .....	46
Figura 4.4 Bloque 1 de la representación de datos .....	53
Figura 4.5 Bloque 2 de la representación de datos .....	53
Figura 4.6 Representación binaria de los eventos y sus variables de la población .....	54
Gráfica 4.1 Población inicial .....	54
Gráfica 5.1 Población inicial, eventos sin la posición geográfica .....	62
Gráfica 5.2 Población inicial, eventos con la posición geográfica .....	62
Gráfica 5.3 Días de las semanas y horas de los eventos delictivos .....	63
Gráfica 5.4 Regresión lineal entre los días y horas de los eventos delictivos .....	66
Gráfica 5.5 Día con más eventos delictivos .....	66
Gráfica 5.6 Hora con más incidencia delictiva .....	67
Gráfica 5.7 Mes con mayor índice delictivo .....	67
Gráfica 5.8 Día del mes con alta recurrencia delictiva .....	68
Gráfica 5.9 El robo es el delito de mayor incidencia, representado por la línea naranja .....	69

## **Capítulo 1. Introducción**

A lo largo del tiempo, el desarrollo de nuevas tecnologías de la información y la computación, han evolucionado de forma tal que permiten la simulación y la creación de sistemas informáticos, capaces de procesar datos a grandes escalas y resolver problemas con técnicas computacionales que basan sus procesos lógicos y matemáticos en los modelos naturales. Las ciencias de la computación usan y modelan distintos paradigmas de la naturaleza, por mencionar algunos, usan el comportamiento, la evolución, adaptación y supervivencia de las especies para la solución de problemas o representación de los mismos, la aplicación de estos modelos para la solución de problemas, es llamado cómputo evolutivo.

El cómputo evolutivo toma los procedimientos intrínsecos de varios comportamientos o eventos naturales y simula tales procesos mediante el uso de tecnología, esta simulación y modelado, están enfocadas en la solución de diferentes problemas. La presente investigación, está enfocada en la aplicación de modelos naturales y de representación genética como: los algoritmos genéticos, estrategias evolutivas y programación evolutiva, para el tratamiento de información y determinación de los eventos delictivos que ocurren en una zona geográfica. El punto de este análisis de datos y procesamiento de los mismos, es determinar mediante el uso de un algoritmo evolutivo, si un punto dentro de una zona geográfica analizada, es susceptible de la comisión de algún tipo de delito y de este modo, aplicar técnicas policíacas correspondientes y lograr una buena prevención del delito.

## **1.1 Motivación**

La seguridad es un tema de alta relevancia para toda la población y existen distintos organismos de gobierno que se encargan de mantener la misma. Desafortunadamente, estos organismos de seguridad, se ven sobrepasados por las grandes cantidades de eventos delictivos de una zona. La cantidad y tipos de delitos que se cometen dentro de un espacio geográfico responden a las características del mismo, es decir, la comisión de un evento delictivo, muchas veces está vinculada a los rasgos y patrones particulares de un lugar.

Partiendo de esto, es necesario el análisis de los datos sobre los incidentes delictivos de una zona, para este proceso, es necesario aplicar técnicas de cómputo evolutivo que permitan emular eventos naturales en procesos de acciones delictivas, para que, mediante heurísticas computacionales, se logre la determinación y clasificación de nuevos puntos o posibles puntos delictivos.

Aunque existen distintas aplicaciones de software que permiten hacer una predicción delictiva como lo es Predpol, (Software de vigilancia predictiva) usado en Estados Unidos de América, DragNet, etc., se desconoce el funcionamiento lógico de los mismos y no son de uso gratuito para las organizaciones policíacas, se requiere de un algoritmo general que permita hacer una perfilación geográfica criminal y que se pueda ajustar a cualquier zona con el solo hecho de usar los datos característicos de dicha zona.

## 1.2 Planteamiento del problema

Actualmente, la seguridad en México es una problemática prioritaria, ya que independientemente de las políticas gubernamentales, ésta problemática ha prevalecido en todo momento. Dentro del país se han señalado zonas con mayor índice delictivo, colonias, municipios e incluso estados completos, donde se ven afectados por delitos de alto impacto como: homicidio, secuestro, extorsión, diferentes formas de robos y violación (Rivas, 2017).

Las conductas humanas varían o se ven alteradas por detonadores de cambio como: espacio geográfico, entorno familiar, situación económica, estados emocionales, etc. Al existir un cambio en las conductas, la probabilidad de incurrir en algún tipo de delito es mayor, que puede ser aislado o recurrente, de bajo o de alto impacto.

Debido a la magnitud del problema, las acciones gubernamentales actuales a nivel nacional terminan siendo insuficientes para una prevención del delito efectiva. Considerando el municipio de Valle de Chalco como una zona de alto índice delictivo, para este trabajo, se pretende tomar como referencia de caso por diversas razones. La primera, porque es una zona de incidencia e impacto para el Centro Universitario, en segundo lugar, porque se cuenta con la información y contactos con los cuales se pudo establecer un plan a seguir y por medio de la propuesta de algoritmos propuestos, llevar a cabo experimentación, contando con datos de acontecimientos delictivos reales. Por último, y no menos importante, es que es una zona geográfica estratégica para ser considerada con el objetivo de delinquir por la cercanía con la Ciudad de México y el aislamiento del Municipio tanto de comunicaciones como de vialidades accesibles.

Por lo anteriormente expuesto, surgen las siguientes preguntas de investigación:

- ¿Cómo identificar de forma anticipada las zonas delictivas en Valle de Chalco mediante un algoritmo evolutivo?
- ¿Cuáles de los algoritmos evolutivos serán más efectivos para identificar zonas delictivas?
- ¿Cuáles son los aspectos que toma en cuenta un algoritmo evolutivo enfocado a la perfilación criminal?
- ¿Cuáles son las características de la perfilación geográfica criminal que favorecen el desempeño de un algoritmo evolutivo?

### 1.3 Justificación

México cuenta con una plataforma (Plataforma México), que es un concepto tecnológico avanzado de telecomunicaciones y sistemas de información, que integra todas las bases de datos relativas a la seguridad pública, con la finalidad de contar con todos los elementos de información, para que las instancias policiales y de procuración de justicia de todo el país, lleven a cabo las actividades de prevención y combate al delito, mediante metodologías y sistemas homologados (SSP, 2018).

Sin embargo, la inseguridad sigue siendo un tema que afecta a todas las personas, en México los niveles de inseguridad se han incrementado, a tal modo que en el país existen zonas con mayor índice de violencia, Valle de Chalco en el estado de México es un municipio con alto índice delictivo, con delitos de alto impacto como: robo de vehículos, homicidio y robo a casa habitación (Rivas, 2017).

Aunque se tiene Plataforma México, esta funciona como medio estadístico, como sistema de comunicación, registro, consulta e intercambio de información relativa a la seguridad pública a nivel nacional. Actualmente, la plataforma no cuenta con un algoritmo que permita la identificación anticipada de zonas geográficas delictivas, y las aplicaciones predictivas como PredPol que apoya la perfilación geográfica criminal no es de acceso libre y el algoritmo bajo el que opera está restringido para todo tipo de usuarios.

Debido a la diversidad social con la que cuenta Valle de Chalco, complica una perfilación psicológica general que ayude en la prevención del delito, sin embargo, la zona geográfica es el común denominador para todo habitante de Valle de Chalco, por lo cual una perfilación geográfica criminal permite el acercamiento a los datos de un individuo y su mapa cognitivo, sin la necesidad de hacer un análisis sobre el individuo.

La importancia de esta investigación radica en desarrollar un algoritmo evolutivo que ayude en la perfilación geográfica criminal, haciendo un análisis del espacio geográfico donde se cometieron delitos de alto impacto y sobre los hechos, determinando de forma anticipada las nuevas posibles zonas delictivas, en la figura 1.3.1 se muestra la macro localización de la zona de estudio, Estado de México.

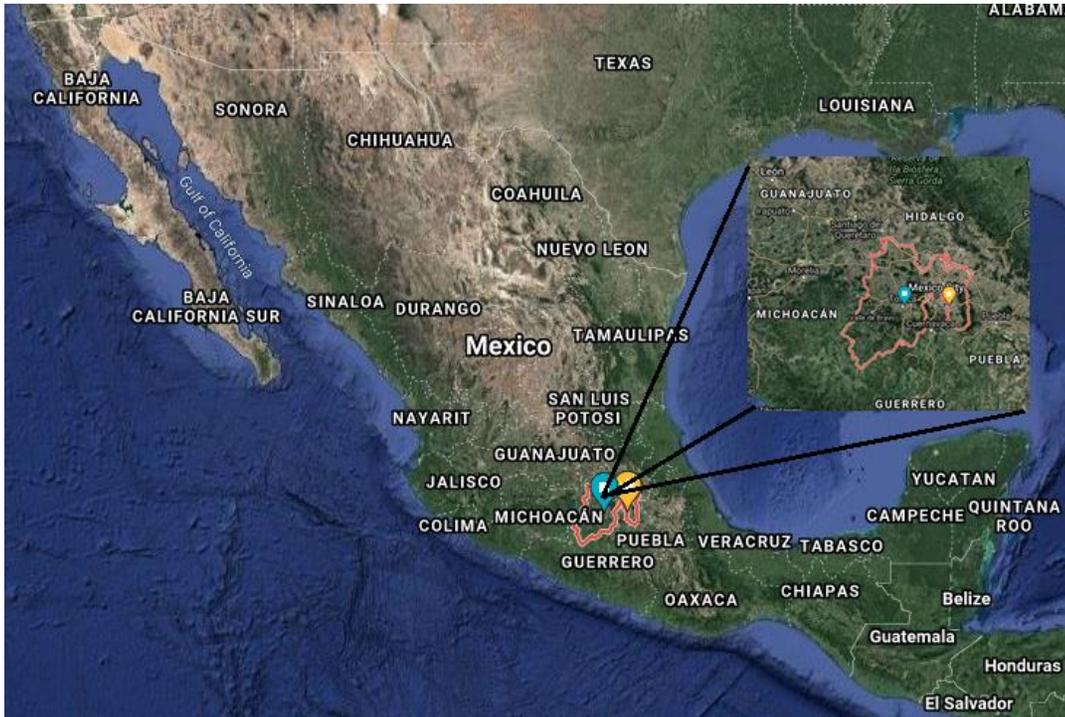


Figura 1.3.1 Macro localización de la zona de estudio, Estado de México (Google, 2021).

En la figura 1.3.2 se resalta Valle de Chalco en el Estado de México, micro localización.

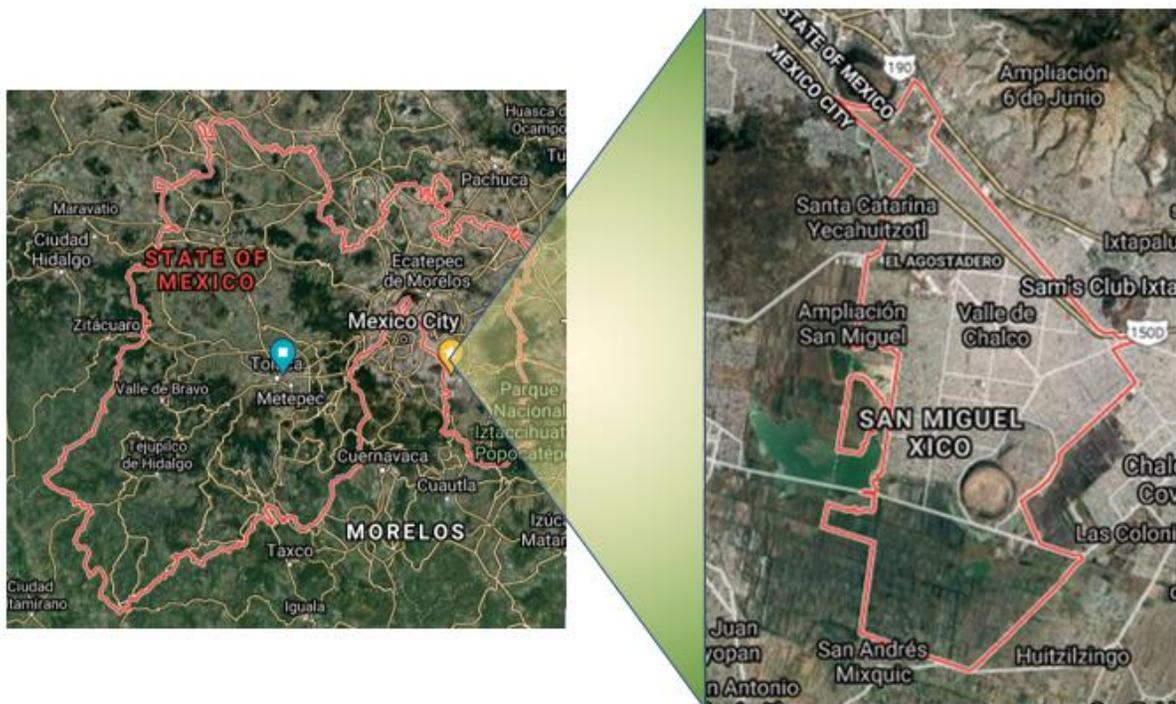


Figura 1.3.2 Micro localización de la zona de estudio, Valle de Chalco (Google, 2021).

Con la aplicación de los algoritmos evolutivos, se pueden modelar los sucesos delictivos cometidos y con técnicas de aprendizaje se pueden predecir zonas de riesgo. Mediante la aplicación de las estrategias policiales convenientes, se puede prevenir y reducir el nivel delictivo en Valle de Chalco.

## **1.4 Hipótesis**

Considerando que los delitos de alto impacto tienen mayor prevalencia en México y que a su vez contribuyen a otro tipo de delitos de mayor incidencia y afectación a la población, es apropiado tener en cuenta el uso de técnicas computacionales específicas como un algoritmo evolutivo enfocado a la perfilación geográfica criminal de los delitos de alto impacto, de tal forma que permita la identificación y clasificación anticipada de las zonas delictivas y así propiciar la toma de medidas preventivas en una región determinada para incidir directamente en la disminución de daños mayores.

## **1.5 Objetivos**

El desarrollo de la presente investigación está basada y enfocada a lograr varios objetivos específicos que permitan lograr el objetivo general, mismos que se escriben a continuación

### **1.5.1 Objetivo General**

Desarrollar un algoritmo evolutivo, considerando diversas técnicas de diseño, enfocado a la perfilación geográfica criminal de los delitos de alto impacto, para la identificación temprana de las zonas delictivas, tomando como referencia de estudio el municipio de Valle de Chalco.

### **1.5.2 Objetivos específicos**

1. Buscar información en artículos, revistas científicas, tesis, internet, repositorios de bases de datos, casos de estudio relacionados a la perfilación geográfica criminal.
2. Analizar los algoritmos genéticos, programación evolutiva, estrategias evolutivas, programación genética y seleccionar el más apropiado para la perfilación geográfica criminal.
3. Elaborar un modelo de datos de los delitos de alto impacto en Valle de Chalco, que permita la perfilación geográfica criminal.
4. Desarrollar un algoritmo evolutivo, que auxilie en la perfilación geográfica criminal.
5. Probar el algoritmo desarrollado con el modelado de datos de los delitos de alto impacto.
6. Contrastar los resultados del algoritmo evolutivo con expertos del área policial, criminológica, psicológica, y computacional.

## **Capítulo 2. Antecedentes**

### **2.1 Uso de técnicas de inteligencia artificial en perfilación geográfica criminal**

La inteligencia artificial (IA) es un conjunto de técnicas computacionales teóricas y prácticas que se encargan de estudiar los agentes computacionales que tienen un comportamiento inteligente, aplica desde las teorías del pensamiento humano hasta la evolución de las especies, modelando y representando la información de manera simbólica o lógica, que, mediante un proceso determinado, ayuda a resolver problemas o tomar decisiones.

La Inteligencia Artificial, es una tecnología que data de mediados de los años 50, momento en que inicia la aparición de los ordenadores digitales principalmente en universidades. El término IA (Inteligencia Artificial) se usa por primera vez en la conferencia de Dartmouth en 1956, retomando de la conferencia original, en McCarthy et al. (2006) se propone un estudio de inteligencia artificial, mismo que debe proceder de la base de la conjetura de que todos los aspectos del aprendizaje o cualquier otra característica de la inteligencia pueden describirse con tal precisión que pueda hacerse una máquina para simularlo y con la capacidad de resolver problemas.

La aplicación de técnicas y estrategias en la solución de problemas, se han presentado y aplicado a lo largo del tiempo en distintas áreas del conocimiento, esto ha llevado a la inteligencia artificial a aplicarse en distintas áreas y la han subdividido, Searle (1980), estableció dos tipos de IA, la IA débil y la IA fuerte, quedando como referente importante para IA.

La IA débil, se usa para la realización de tareas específicas, por ejemplo, jugadas de ajedrez, donde la aplicación de algoritmo es aprender movimientos particulares del contrincante, de forma que se anticipen o se determine la acción a realizar dentro del tablero según los movimientos previos, en este sentido, la IA débil ha demostrado ser útil en casos concretos. La IA fuerte, trata de simular la inteligencia humana y su comportamiento. La simulación del razonamiento y habilidades humanas mediante la IA se encuentra un tanto distante a pesar de los avances actuales (CESEDEN, 2018).

## 2.2 Perfilación geográfica criminal

En ciencias sociales, una de las aplicaciones de la informática y computación ha sido la perfilación, que de acuerdo a la Real Academia Española “es un conjunto de rasgos peculiares que caracterizan a alguien o algo” y las técnicas de IA se han aplicado en la creación de perfiles sociales, geográficos, psicológicos, criminales, académicos, policiales, etc.

En el aspecto policial y criminológico, las herramientas y técnicas computacionales e informáticas, se han usado como apoyo en la creación del perfil geográfico criminal, que se define como una técnica de análisis de la actividad espacial o geográfica de los delincuentes, aplicada a la investigación criminal. El perfil geográfico es una técnica de análisis complementaria o paralela al perfil psicológico, que está definido por el conjunto de características que reúne un ser humano y que determinan su carácter, sus actitudes, aptitudes y determinados comportamientos frente a una situación particular o ante la sociedad como tal (Garrido, 2006).

Debido a los incrementos de delitos, se usa el perfil geográfico en investigaciones policiales y criminológicas para obtener un perfil del delincuente basado en su mapa mental, movimientos geográficos y como objetivo establecer una hipótesis acerca de su domicilio, prever movimientos y zona de actuación (Jiménez, 2018).

Algunas de las aplicaciones que ayudan en la perfilación geográfica criminal son: Dagnet, Rigel, Predator, Crimestat, que mediante la captura de datos geográficos y del crimen como: tipo de delito, hora de los hechos, lugar de los hechos, lugar de contacto entre agresor y la víctima, condición geográfica, rutas de entrada, rutas de salida, entre otros, y con la interacción con un sistema de información geográfico, se traza un círculo sobre los espacios de los delitos, se indica con códigos de colores rutas de acceso y escape, así como posibles ubicaciones del criminal, a este proceso se le conoce como hipótesis del círculo (Summers, 2015).

Una de las aplicaciones de software más recientes es Predictive Policing (PredPol) que predice el lugar donde ocurrirá un nuevo delito, actualmente en uso en California, Estados Unidos de América; PredPol usa un algoritmo de aprendizaje automático y requiere de una base de datos

de hechos de 2 a 5 años para calcular sus predicciones en nuevas ciudades. Este sistema requiere de solo 4 tipos de datos: tipo de delito, ubicación de delito, fecha y hora del delito (PredPol, 2018).

En la misma frecuencia de prevención delictiva, en Nuevo León, México, a partir del tercer trimestre del 2018, se está implementando una App que determina polígonos de altos índices delictivos, con lo cual crean estrategias de seguridad que permitan la prevención del delito.

La IA ha colocado a las ciencias sociales como área de estudio, centrándose en acciones de ámbito judicial, donde los avances recientes en el procesamiento del lenguaje natural, el aprendizaje automático y redes neuronales, proporcionan las herramientas para construir modelos predictivos que se pueden utilizar para descubrir patrones que impulsan las decisiones judiciales. Esto puede ser útil para abogados y jueces, como una herramienta de ayuda para identificar rápidamente casos y extraer patrones que conducen a ciertas decisiones.

Aletras N. et al. (2016), presenta el primer estudio sistemático sobre la predicción del resultado de casos juzgados por el Tribunal Europeo de Derechos Humanos basados únicamente en el contenido textual. Formulando una tarea de clasificación binaria donde la entrada de los clasificadores es el contenido textual extraído de un caso y el resultado objetivo es el juicio real sobre si hubo violación de un artículo de la convención de los derechos humanos. La información textual se representa usando secuencias de palabras contiguas, es decir, N-grams y temas. Los modelos pueden predecir las decisiones de la corte con una gran precisión (79% en promedio). El análisis empírico indica que los hechos formales de un caso son el factor predictivo más importante. Esto es consistente con la teoría del realismo jurídico que sugiere que la toma de decisiones judiciales se ven significativamente afectadas por el estímulo de los hechos.

## 2.3 Perfilaciones Generales

Varios de los procesos cotidianos han tratado de ser simulados mediante tecnologías y técnicas de inteligencia artificial, en una perfilación estudiantil, Lacave (2016) aplica las redes Bayesianas para identificar perfiles de estudiantes en un estudio del abandono de las titulaciones de informática en la Universidad de Castilla-La Mancha, esto con el objetivo de definir acciones orientadas a reducir la tasa de deserción, se aborda el problema mediante algoritmos de aprendizaje de redes bayesianas. Los resultados del trabajo no son concluyentes debido a las restricciones de la base de datos utilizada, pero la descripción del estudio realizado pone en valor el interés de la técnica empleada y sienta las bases para mejorar el alcance de la investigación en trabajos futuros relacionados con la extracción de perfiles de estudiantes.

Del Alcázar (2014), con el objetivo de apoyar el desempeño estudiantil, aplicó un estudio que permitió cualificar y cuantificar la asociación entre las posibles causas de deserción de estudiantes de carreras técnicas mediante una red neuronal. Para generar un perfil de riesgo, se ejecutó el análisis de asociación entre las variables mediante red neuronal artificial multicapa en la información histórica sociodemográfica y académica de 396 estudiantes de primer semestre de Ingeniería Automotriz de una institución de educación superior. La red neuronal que modela la deserción se construyó a través del ajuste de los pesos de las variables mediante el algoritmo de aprendizaje. Para el desarrollo del modelo de red se usó el programa estadístico SPSS v21. Las variables usadas fueron: materias reprobadas en el primer semestre, materias aprobadas en el primer semestre, nota del colegio categorizada, nota de admisión categorizada, edad, número de materias tomadas en el primer semestre, tipo de colegio y sexo. En este estudio se encontró que el modelo generado permite identificar a los estudiantes en riesgo de desertar y apoya la toma de decisiones para prevenir la deserción estudiantil en el primer año de ingeniería Automotriz.

A nivel global, varias empresas de distintos giros han estado implementando técnicas de inteligencia artificial para la perfilación desde distintos puntos, gustos personales, necesidades empresariales e individuales, medidas y restricciones para la ejecución de tareas, asignación de actividades, etc.

## **Capítulo 3. Marco teórico y conceptual**

### **3.1 Delitos**

Desde el inicio de los tiempos se han creado normas, reglas, procedimientos y restricciones necesarias para poder llevar un orden dentro de un grupo o bloque social, cada una de estas restricciones de convivencia social han sido usadas para tratar de mantener armonía la sociedad. Es así que, a lo largo de la historia, las normas, reglas y sanciones, se han venido actualizando y adaptando a cada una de las épocas; Estas modificaciones y actualizaciones, han dejado acciones y reacciones bien definidas en un marco de ley, por lo tanto, el diferente tipo de acciones que quebrantan la ley en al menos una forma, se le conoce como delito.

#### **3.1.1 Clasificación del delito**

Los delitos pueden dividirse por: su gravedad, la conducta del agente, el resultado, por el daño que causan, por su duración, por la culpabilidad, simples y compuestos.

En función de la materia, los delitos pueden clasificarse en:

Comunes: constituyen la regla general y son aquellos que se formulan en las leyes dictadas por las legislaturas locales.

Federales: son aquellos que se formulan por las leyes expedidas por el congreso de la unión.

Oficiales: los que comente un empleado o funcionario público en el ejercicio de sus funciones (abuso de ellas), incluyendo altos funcionarios de la Federación.

Militares: afecta a la disciplina del ejército. La constitución Mexicana prohíbe a los tribunales militares extender su jurisdicción sobre personas no militares.

Políticos: Todos los hechos que de algún modo afecten la organización del estado en sí misma, en sus órganos o representantes (leyderecho, 2019)

Otra clasificación de los delitos que se pueden encontrar, son los “delitos informáticos”; Es decir, todos aquellos eventos donde se involucra la tecnología de la información y comunicación. Estos delitos pueden subdividirse en dos vertientes o modalidades:

Cuando las tecnologías son el medio en la comisión de un delito: Cuando se le da uso o está involucrada de forma directa o indirecta cualquier tipo de tecnología de la información en cualquier evento delictivo.

Cuando las tecnologías son el objeto de la comisión de un delito: Cuando las tecnologías de la información y comunicación son el objetivo de cualquier delito, por mencionar algunos, robos de información digital, fraudes bancarios, clonación de sitios web (pishing), etc.

Todo proceso o evento delictivo, se basa en un algoritmo determinado y detallado que permite la ejecución del mismo. La ejecución repetitiva de procesos lleva a la mejora, a la innovación, a la búsqueda de nuevas y mejores estrategias que permitan la ejecución de tales procesos con un menor índice de errores. La aplicación cíclica de las mismas técnicas delictivas, la convierten en un proceso predictivo, es decir, se mantienen patrones que son usados durante un lapso de tiempo largo o durante un grupo de acciones. La ejecución repetida de bloques de procesos, lo convierten en un proceso especializado y predecible a la vez.

Cuando se tienen procesos, acciones o eventos con características similares, estos pueden ser modelados, simulados y evaluados mediante técnicas inteligentes de computación.

### **3.1.2 Tipos de Prevención del delito**

La prevención del delito o prevención delictiva, según García-Pablos (1992), es un efecto disuasorio mediato, indirecto, perseguido a través de instrumentos no penales, que alteran el escenario criminal modificando algunos de los factores o elementos del mismo (espacio físico, diseño arquitectónico y urbanístico, actitudes de las víctimas, efectividad, etc.). Pretendiendo así, poner obstáculos de todo tipo al autor de ejecución del plan criminal mediante una intervención selectiva en el escenario del crimen.

La naturaleza de actos delictivos, requieren de múltiples acciones a seguir o aplicar para lograr disminuirlos, la Organización Mundial de la Salud (2002), da nueve recomendaciones para la prevención de la violencia

1. Crear, aplicar y supervisar un plan nacional de acción para prevenir la violencia: Este plan debería aplicarse a nivel local, nacional e internacional.
2. Aumentar la capacidad de recolectar datos sobre la violencia.
3. Definir las prioridades y apoyar la investigación de las causas, las consecuencias, los costos y la prevención de la violencia.
4. Promover respuestas de prevención primaria.
5. Reforzar las respuestas a las víctimas de la violencia.
6. Integrar la prevención de la violencia en las políticas sociales y educativas, y promover así la igualdad social y entre los sexos.
7. Incrementar la colaboración y el intercambio de información sobre la prevención de la violencia.
8. Promover y supervisar el cumplimiento de los tratados internacionales y la legislación y otros mecanismos de protección de los derechos humanos.
9. Buscar respuestas prácticas y consensuadas a nivel internacional al tráfico mundial de drogas y de armas.

Las recomendaciones anteriores son pensadas para aplicarse a nivel nacional, estatal y municipal. Las comisiones coinciden en la importancia de los municipios, porque tienen una capacidad mayor en la organización de estrategias aplicadas en los factores de riesgo que causan los delitos. Adicionalmente, para tener éxito, se necesitan establecer alrededor de un

diagnóstico de la situación, un trabajo en conjunto con instituciones claves como escuelas, creación de empleos, servicios sociales, albergues y aplicación de la ley.

Otras medidas de prevención de delito de acuerdo a Williams (2014) y CAF (2014) son:

Crear asociaciones culturales legales para acoger a miembros de pandillas ilegales.

Formar a la policía en prevención eficaz de la violencia y el delito.

Ofrecer actividades sanas a horas avanzadas

Fortalecer la nutrición y estimulación temprana de los niños y la promoción de ambientes familiares proactivos y no conflictivos.

### ***Tipos de prevención***

Existen diversas causas que hacen que un delito ocurra, Redondo et al. (2007) consideran 5 proposiciones:

La delincuencia se aprende.

Existen rasgos y características individuales que predisponen al delito.

Los delitos constituyen reacciones a vivencias individuales de estrés y tensión.

La implicación en actividades delictivas es el resultado de la ruptura de los vínculos sociales.

El inicio y mantenimiento de la carrera delictiva se relacionan con el desarrollo del individuo, especialmente en la infancia y la adolescencia.

Sin embargo, se pueden desarrollar competencias (tipos de prevenciones) en los individuos que son imprescindibles para la vida social, Redondo et al. (2007), menciona las siguientes:

Desarrollo de nuevas habilidades: Lograr que los sujetos adquieran habilidades y hábitos de comunicación no violenta de responsabilidad laboral y familiar, de motivación de logro personal, etc.

Desarrollo del pensamiento: Hacer que los sujetos logren una solución cognitiva de problemas interpersonales, que incluye patrones de reconocimiento y definición de problemas

Regulación emocional y control de ira: Interpretación de situaciones, comunicación asertiva, terapias, comunicación en grupos.

Prevención de recaídas: Promover actividades terapéuticas a los contextos habituales del sujeto y facilitar el mantenimiento de dichas mejoras a lo largo del tiempo

### ***Prevención situacional del delito***

La prevención situacional es un conjunto de teorías que tiene al crimen como opción racional, utilitario, instrumental, y selectivo. Los objetivos principales de la prevención situacional de acuerdo a Soto (2016) son:

Reducir las oportunidades para cometer delitos.

Disminuir la percepción de inseguridad de la población en espacios públicos concretos.

Promover el control social de la población sobre espacios urbanos comunes.

### ***Prevención situacional del delito en informática***

En el área de informática, la prevención del delito radica en la capacidad de infraestructura tecnológica de las organizaciones, la buena administración, uso y aplicación de las tecnologías de la información y comunicación TIC.

El diseño lógico (funcionamiento) y físico (estructura) de un sistema de tecnologías de información, es una parte fundamental en la prevención de delitos, el uso de computadoras (servidores) con características particulares que ayudan a controlar procesos, tareas o brindar servicios, como almacenamiento de información o alojamiento de sitios web; Permiten el control total o parcial de accesos a la información, dependiendo de las técnicas aplicadas para compartir la información.

La concentración de información en servidores, facilita el acceso a la misma, sin embargo, un delito cibernético es una amenaza latente. Todas las empresas o particulares que hacen uso de las TIC para el procesamiento y almacenamiento de la información, necesitan aplicar técnicas para su protección. El delito de intrusión a los sistemas de información es un negocio bastante grande, un experto en el área, puede vulnerar los sistemas de seguridad informáticos y apoderarse de la información dentro de los servidores, cambiando contraseñas, modificando información, encriptando información (una forma de alteración de los datos), para posteriormente solicitar algún tipo de remuneración para la liberación o recuperación de la información extraída.

A pesar de varios años de investigación e inversión en el área de seguridad informática, hay informes frecuentes sobre intrusiones cibernéticas a los sistemas informáticos más protegidos

del orbe. En estos casos, las empresas, deben considerar las siguientes estrategias de prevención situacional del ciberdelito.

Aumentar el esfuerzo: Restricción de accesos, uso de cortafuegos, encriptación de datos, cambio frecuente de contraseñas de accesos a sistemas, filtros de navegación en internet.

Aumentar el riesgo que corre el delincuente: Crear historial de accesos, reforzar procesos de acceso a sistemas, uso de keyloggers (software para registrar pulsaciones en el teclado), uso de cámaras en puntos particulares, doble autenticación en compras por internet.

Reducir los beneficios de cometer el delito: Control de sitios de internet, manejo de correo basura, legislar delitos cibernéticos, cambio o cancelación de servicios al detectar un delito informático.

Reducir las provocaciones y las excusas: Mantener las buenas relaciones entre los empleados, hacer conciencia del uso responsable de la informática (leyderecho, 2019).

### ***El fraude y la prevención situacional del delito***

En el área informática, todos los sistemas de información son pensados y modelados para satisfacer necesidades de los clientes, además de incluir dentro de estos, mecanismos de seguridad a nivel software (programas) y a nivel hardware (dispositivos físicos). La ejecución de todo bloque de proceso dentro de los sistemas deben ser registradas, esto permite tener el historial de actividades y en caso de presentarse alguna situación que altere el funcionamiento o la estructura de alguna de las partes del sistema, identificar de forma clara los puntos vulnerados, además de bloquear o dejar fuera las conexiones que se han establecido fraudulentamente con el sistema de información.

La prevención del fraude engloba varias actividades como: alertas tempranas, señales, patrones de comportamiento en fraudes, perfiles de usuarios y actividades de los mismos (historial), buen nivel en seguridad informática y lograr las satisfacciones de los clientes; Sin embargo, la gran cantidad de información almacenada y procesada, la aparición de nuevas técnicas de fraude, entre otras, hacen que la prevención en delitos informáticos sea compleja.

En Rivero (2014) se señalan las técnicas implementadas en la detección del fraude, pueden ser: estadísticas y de inteligencia artificial.

### **Las técnicas estadísticas**

- Creación del conjunto de datos, agrupación y clasificación.
- Construcción de atributos o características de eventos.
- Reducción o procesamiento de datos para su validación.

### **Las técnicas de inteligencia artificial**

- Minería de datos: categorización, agrupación y análisis de datos.
- Reconocimiento de patrones: usado para identificar comportamientos de forma automática.
- Redes Neuronales: aprenden patrones para su detección posterior.

## **3.2 Inteligencia Artificial**

Como se ha mencionado líneas anteriores, la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría una persona. A diferencia de las personas, los equipos con inteligencia artificial no requieren tiempos de reposo y pueden analizar grandes bloques de datos de forma paralela. Así mismo, los márgenes de error se reducen significativamente en comparación con los humanos (Rouhiainen, 2018). Esta idea tiene su punto de partida con información que data de los años 50, en el artículo de Alan Turing “Computing Machinery and Intelligence” sobre el tema de inteligencia artificial, actualmente conocida como la prueba de Turing.

En el artículo escrito por Turing se plantea la siguiente cuestión ¿Pueden las máquinas pensar?, es un examen de la capacidad de una máquina para exhibir un comportamiento inteligente similar al de un ser humano.

### **3.2.1 Tipos de inteligencia artificial**

La inteligencia artificial se encuentra dentro de un conjunto vasto de aplicaciones, pero la característica de cada una de las aplicaciones está basada en los algoritmos, procesos empleados y en los objetivos perseguidos, en este sentido, se puede hablar de distintos tipos de inteligencia artificial.

#### ***Máquinas reactivas***

Los sistemas de inteligencia artificial más básicos son puramente reactivos. No tiene la capacidad de generar información o recuerdos y tampoco son capaces de usar las experiencias pasadas que le permitan y ayuden en la toma de decisiones actuales. Un ejemplo es Deep Blue, fue una supercomputadora creada por IBM, a finales de la década de 1990 fue capaz de vencer en juegos de ajedrez al maestro internacional Garry Kasparov. Esta máquina puede identificar las piezas en un tablero de ajedrez y saber cómo se mueve cada una. Puede realizar predicciones sobre los mejores movimientos y elegir la mejor de todas las posibilidades. Esta máquina no tenía información del pasado, tampoco poseía recuerdos de lo que había sucedido antes. Aparte de una regla de ajedrez, Deep Blue ignora todo antes del momento presente. Todo lo que hace es enfocar y analizar las piezas del tablero en tiempo real, de la máquina y del oponente, eligiendo el movimiento siguiente más estratégico.

### ***Memoria limitada***

Los sistemas de inteligencia artificial de memoria limitada tienen la capacidad de mirar hacia el pasado, un ejemplo son los vehículos autónomos que hacen algo parecido, pueden observar la velocidad y dirección de otros autos, esta información es programada en la memoria de estos vehículos, añadiendo experiencias que ayudan a decidir el momento en que debe cambiar de carril para evitar interrumpir a otro conductor o ser colisionado por un automóvil cercano, aclarando que estas observaciones no se almacenan permanentemente.

### ***Teoría de la mente***

Es una expresión usada en psicología, implica que cuando un sujeto inteligente está dotado de la teoría de la mente tiene la capacidad de comprender y reflexionar respecto al estado mental de sí mismo y del prójimo que puede tener un sujeto. Esto es crucial en el momento que los humanos forman sociedades, debido a que permite la interacción social. Por lo tanto, este tipo de inteligencia artificial es el más deseado, las máquinas con este tipo de Inteligencia son mucho más avanzadas, no solo se pretende formen representaciones sobre el mundo, sino sobre agentes o entidades. Lo que se persigue es que las máquinas interactúen de forma directa con la sociedad, por lo cual deberán tener una comprensión sobre los pensamientos, sentimientos y como deben ser tratadas las personas, en consecuencia, tendrán que ajustar su comportamiento.

### ***Autoconciencia***

Los seres conscientes de sí mismos conocen sus estados internos y pueden predecir o entender los sentimientos de los demás, por lo tanto uno de los principales objetivos de la inteligencia artificial es crear sistemas que puedan formar representaciones de sí mismos. No solo se trata de tener conciencia, sino de construir máquinas que la tengan.

### **3.2.2 Áreas de la inteligencia artificial**

La IA tiene varias áreas de estudio y cada una de estas áreas puede ser aplicada y usada sin límites, todo en función del problema y objetivo perseguido, para entenderlas de forma general, se da una breve descripción de cada una.

#### **Aprendizaje automático (Machine Learning)**

El aprendizaje automático o machine learning es un tipo de inteligencia artificial que le da a las computadoras la capacidad de aprender. Se basa en el análisis de datos, a través de los cuáles puede encontrar nuevos patrones que le permiten modificar su comportamiento. Es decir, analiza y procesa la información, describiendo y descubriendo patrones que permiten la ejecución de nuevas acciones que están en función del resultado del procesamiento de información.

#### **Ingeniería del conocimiento**

Esta se dedica a estudiar, diseñar y desarrollar los sistemas expertos. Sistemas creados para solucionar problemas emulando la mente de una persona experta en un campo en particular, para ello, aprende reglas de negocio, y técnicas como arboles de decisión para emular el conocimiento de un experto en un campo. Con un árbol de decisión se puede hacer una representación esquemática de un problema que facilita la toma de decisiones al representar visualmente las posibilidades que existen ante tal escenario, además de las consecuencias que cada escenario podría tener.

#### **Lógica difusa**

Es una lógica que permite representar matemáticamente la incertidumbre y la indeterminación, proporcionando herramientas formales para su tratamiento. Esta técnica se basa en el uso de términos cualitativos (“Pedro es MUY alto”), a los cuáles se les otorga un valor cuantitativo (“¿cómo de alto debería ser Pedro para ser considerado MUY alto?”). Gracias a este tipo de lógica, la inteligencia artificial puede establecer patrones en base a términos no cuantitativos.

## **Redes Neuronales Artificiales**

Esta técnica se inspira en el funcionamiento de las redes neuronales humanas. Al igual que en el ser humano, son sistemas independientes que están interconectados entre sí. Cada una de estas neuronas artificiales recibe un número determinado de entradas, a las cuáles les otorga un “peso” determinado. En función de la cantidad de entradas y su peso, recibirá un determinado “impulso nervioso”, lo cual se traduce en un valor de salida. Este tipo de redes se organiza por capas, de forma que la primera capa recibe los datos reales, en bruto, y los va transfiriendo al resto de capas, las cuáles las van procesando para emitir un valor final de salida.

## **Sistemas Multiagente**

Un agente inteligente es capaz de percibir el entorno, analizar sus variables y actuar en consecuencia. Pues bien, los sistemas multiagente son una combinación de los mismos cuyo objetivo es realizar tareas que un agente individual es incapaz de afrontar. Se utilizan sobre todo para la resolución de problemas o el desarrollo de entornos colaborativos adaptables.

## **Sistemas basados en reglas**

Este tipo de sistemas funcionan aplicando diferentes reglas para una determinada situación, y comparando después los resultados obtenidos. Es posible llevar a cabo esta tarea por diferentes métodos. Por un lado, pueden partir de una evidencia o situación inicial y hallar su posible solución; por otro, pueden partir desde hipótesis con posibles soluciones y realizar el recorrido inverso para hallar la premisa o evidencia.

## **Razonamiento basado en casos**

Es un paradigma que busca la solución de problemas inspirado en el razonamiento humano y el uso de la memoria. Mediante este razonamiento, se pretende la solución de problemas nuevos tomando como referencia la solución de problemas anteriores.

## **Sistemas Expertos**

Los sistemas expertos son programas dedicados a solucionar problemas como lo haría un ser humano. El añadido, en este caso, es que el ser humano cuya mente se emula sería un experto en un tema en particular, es decir, que tendría un conocimiento especializado en ella. De ahí su nombre. El sistema experto, está programado con conocimientos iniciales especializados en una materia, y utiliza esos conocimientos cuando se le plantea un problema. Por tanto, un Sistema Experto está dividido, fundamentalmente, en:

- 1) Base del conocimiento: los conocimientos iniciales con los que trabaja. Redes semánticas, casos prácticos, reglas, predicados, toda la información inicial sobre una materia en concreto.
- 2) Motor de inferencia: el motor que le proporciona la capacidad para razonar, el que le lleva a seleccionar y combinar datos, contrastar los hechos, hacer hipótesis. Es emplear los conocimientos iniciales con base en el problema que se plantea, para encontrar una solución.

## **Vida Artificial**

La vida artificial trata de comprender mejor a los seres vivos construyendo y simulando sistemas que exhiben propiedades similares a las de los organismos biológicos. A su vez, esta comprensión permite construir sistemas artificiales que pueden adaptarse, evolucionar, aprender y autoorganizarse, tal y como los sistemas vivos.

## **Lingüística computacional**

Campo multidisciplinar de la Lingüística y la Informática que utiliza la última para estudiar y tratar el lenguaje humano. Para lograrlo, intenta modelar de forma lógica el lenguaje natural desde un punto de vista computacional. Dicho modelado no se centra en ninguna de las áreas de la lingüística en particular, sino que es un campo interdisciplinar, en el que participan lingüistas, informáticos especializados en inteligencia artificial, psicólogos cognoscitivos y expertos en lógica, entre otros. A la Lingüística Computacional se le llama a veces Procesamiento del Lenguaje Natural (PLN).

Esta puede ser dividida en a) lingüística teórica y b) lingüística aplicada

a) El objetivo es desarrollar teorías lingüísticas computables, es decir, que puedan ser aplicadas sutilmente a las computadoras. Dado que las teorías formales existentes no dan cuenta de todos los fenómenos lingüísticos posibles, la lingüística computacional teórica se perfila como un incentivo para la comprensión formal de los procesos lingüísticos, así como un medio para su demostración práctica.

b) Se centra en la aplicación de técnicas computacionales para la simulación de la conducta lingüística, buscando con ello, la creación de productos informáticos que incorporen algún componente en el que intervenga el lenguaje oral o escrito.

### **Procesamiento de lenguaje natural**

Se encarga de investigar la forma en que las máquinas puedan comunicarse con las personas mediante el uso de lenguaje natural como el español inglés, etc. Para lograrlo, se necesita aplicar modelos lógicos y probabilísticos, para los modelos lógicos, se deben escribir las reglas de reconocimiento de patrones estructurales, empleando un formalismo gramatical concreto. Para el aspecto probabilístico, se deben recoger ejemplos y datos y a partir de estos, calcular las frecuencias de las distintas unidades lingüísticas que van desde las letras, la unión de estas (palabras) y las oraciones y la probabilidad de aparecer en algún contexto determinado.

### **Minería de datos**

Es el proceso de detectar cierta información que se puede procesar de un conjunto grande de datos. Se apoya de un análisis matemático para identificar patrones y tendencias que existen en los datos. Para la creación de un modelo de minería de datos se requiere de un proceso mayor que incluye desde la generación de preguntas acerca de los datos, el modelado de datos para responder tales preguntas y la implementación en un entorno de trabajo.

### **Cómputo Evolutivo**

Esta área de la inteligencia artificial se encarga de la aplicación de procesos evolutivos de las especies mediante la aplicación de técnicas computacionales y modelaciones matemáticas que ayuden en la solución de problemas. Los algoritmos genéticos, la programación evolutiva

y las estrategias evolutivas, son algunas de las principales técnicas del cómputo evolutivo. La presente investigación está apoyada en el cómputo evolutivo para la perfilación geográfica criminal

### 3.3 Algoritmos

Al conjunto de instrucciones que están definidas, llevan un orden de ejecución y que son acotadas para la solución de un problema, se les llama algoritmo. Los algoritmos se encuentran en todos los procesos que se llevan a cabo por los humanos y por la propia naturaleza. Evolucionan de forma natural o artificial con el propósito de resolver un problema en particular de manera óptima. Por definición, suelen ser diseños estrictamente lógicos, como las operaciones matemáticas, que han demostrado ser la mejor opción para resolver determinados problemas. Básicamente, un algoritmo es la solución principal a un problema específico.

Los elementos principales de un algoritmo son: a) entrada: La entrada son los datos que debe tener al algoritmo antes de su ejecución. b) proceso: conjunto de acciones que se llevan a cabo para la solución o ejecución de la tarea. c) salida: datos o información que se obtiene después del procesamiento del mismo.

Durante el diseño y ejecución de un algoritmo informático, se tienen las siguientes fases

1. Análisis del problema

Esta etapa requiere una definición clara, donde se pueda considerar con precisión las operaciones que debe realizar el programa y los resultados o soluciones requeridos. A medida que se busca una solución, se requieren especificaciones de entrada y salida.

2. Diseño del algoritmo

El método utilizado en el proceso de diseño se basa en la premisa de divide y vencerás. En otras palabras, al descomponer el problema en subproblemas y luego descomponer estos subproblemas en otros problemas de nivel inferior, los problemas complejos se pueden resolver con la implementación de una computadora.

3. Codificación

Es una representación algorítmica escrita en un lenguaje de programación. Dado que el diseño del algoritmo no tiene nada que ver con el lenguaje de programación utilizado en la implementación, el código se puede escribir fácilmente en uno o más lenguajes.

4. Depuración

Es el proceso de ejecutar un programa usando una variedad de datos de entrada. Estos datos de entrada se denominan prueba o datos de prueba, tales como: valores de entrada normales, límites de programas de verificación de valores de entrada externos

y aspectos especiales de los programas de verificación de valores de entrada. Estos determinarán si el programa contiene errores.

#### 5. Compilación

Una vez que se tiene el algoritmo escrito en un lenguaje de programación, este se almacena y se genera el archivo ejecutable que la computadora entiende.

#### 6. Mantenimiento

Después de la implementación del algoritmo, es necesario mantenerlo en constante supervisión para detectar fallas en el mismo y poder hacer los cambios pertinentes.

En concreto, estos deben ser: finitos, eficientes, legibles, flexibles a modificaciones, estar contruidos por módulos, contar con un inicio y un final.

### 3.3.1 Clasificación de los algoritmos

La diversidad de tareas que existen, hace que se tenga una amplia clasificación de algoritmos, a continuación, se mencionan algunos de forma muy general.

- Algoritmo computacional: Por la complejidad de estos y por la cantidad de subprocesos, requieren de equipos de cómputo para su implementación. Se pueden llamar algoritmos cuantitativos que se optimizan.
- Algoritmo no computacional: Son los algoritmos que son embebidos en dispositivos electrónicos pequeños y su ejecución se lleva a cabo en ellos.
- Algoritmo cuantitativo: Usado para el procesamiento de números y cálculos matemáticos.
- Algoritmo probabilístico: Es usado para el análisis de datos y poder determinar el comportamiento de los mismos. Mediante la teoría de la probabilidad, se obtienen datos que permiten formar hipótesis en función de los resultados.
- Algoritmos heurísticos: Se encargan de encontrar soluciones de entre un conjunto de ellas, estas posibles soluciones a un problema, son obtenidas a partir de un grupo de datos bien identificados, por lo cual, las soluciones obtenidas pueden ser la mejor solución o la más próxima.
- Algoritmo voraz: Es un algoritmo usado para problemas de optimización, donde se encuentran soluciones globales a problemas particulares.
- Algoritmos genéticos: Estos algoritmos basan su estructura en la genética natural, por lo tanto, requiere de una representación estructurada de la información de los casos y pasar por un proceso de mutación, combinación, selección o evaluación y soluciones.

### 3.4 Introducción al cómputo evolutivo

Desde la década de los 40 en la que nació y comenzó a desarrollarse la informática, el modelo neuronal la ha acompañado, de hecho, la aparición de las computadoras digitales y el desarrollo de las teorías modernas acerca del aprendizaje y del procesamiento neuronal se produjeron aproximadamente al mismo tiempo, a finales de los años cuarenta, (Barceló, 2008).

Desde entonces y hasta la fecha, la investigación neurofisiológica y el estudio de sistemas neuronales artificiales ANS (por sus siglas en inglés, Artificial Neural Systems) han ido de la mano. Sin embargo, los modelos de ANS no se centran en la investigación neurológica, si no que toma conceptos e ideas del campo de las ciencias naturales para aplicarlos en la resolución de problemas pertenecientes a otras ramas de las ciencias y la ingeniería. Se puede decir que, la tecnología ANS incluye modelos inspirados por la comprensión del cerebro, pero que no tienen por qué ajustarse exactamente a los modelos derivados de dicho entendimiento.

El estudio de sistemas neuronales artificiales pertenece a la Inteligencia Artificial (IA), y ésta se define como una disciplina que estudia, imita, modela y analiza los agentes computacionales que tienen una conducta inteligente (Poole and Mackworth, 2010). Según (Whitby, 2009), la IA se define como: “El estudio del comportamiento inteligente (en los seres humanos, animales y máquinas) y el intento de encontrar la manera en que podría tal comportamiento ser incluido en cualquier tipo de artefacto”.

Uno de los campos de estudio de la inteligencia artificial, es el aprendizaje automático, que tiene que ver con la construcción de algoritmos o programas que mejoren con la experiencia de forma automática. Para el diseño lógico de un proceso de aprendizaje automático, es necesario determinar de forma correcta la información para el entrenamiento, la función objetivo que debe ser aprendida, la representación para esta función y un algoritmo para aprender la función objetivo a partir de la información de entrenamiento.

El aprendizaje implica búsquedas en un espacio de soluciones posibles a un problema para identificar la hipótesis que mejor se adapte a los ejemplos de entrenamiento disponibles y a otras restricciones a priori. Se pueden usar algunos tipos de aprendizaje que buscan dentro

de un conjunto de hipótesis; pueden ser reglas simbólicas, árboles de decisión redes neuronales artificiales, entre otros.

Uno de los problemas clásicos dentro de Inteligencia artificial, es predecir comportamientos a partir de un conjunto de datos que modela tales comportamientos, estos datos están en espacio de  $(n + 1)$  donde  $n$  es el número de variables del problema, por lo cual, las variables del problema serán  $x_1, x_2, \dots, x_n$  el valor a predecir será  $y$ , el objetivo es obtener la función como se aprecia en la ecuación 1.

$$f(x_1, x_2, \dots, x_n) = y \quad (1)$$

Y así poder obtener valores de  $y$  para cualquier valor posible de las variables.

Los algoritmos evolutivos constituyen un método de optimización de una función objetivo  $f$ , este proceso consiste en dados los parámetros  $x_1, x_2, \dots, x_n$  de la función objetivo, encontrar los valores  $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$  tales que hacen mínimo (o máximo, según la función objetivo) el valor de la función en dicho punto  $f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ .

La computación evolutiva (EC por sus siglas en inglés, Evolutionary Computing) tiene sus raíces en los conceptos fundamentales de la vida, tomando como modelo el principio de selección natural. La EC imita los procesos de evolución natural basándose en métodos de adaptación para resolver problemas de búsqueda y optimización, mediante la utilización de algoritmos que tienen la capacidad de evolucionar para encontrar la mejor solución (Smolinski et al., 2008).

De acuerdo a Bronwlee (2011), algunos de los algoritmos derivados en cómputo evolutivo son: algoritmos genéticos, programación genética, estrategias evolutivas y evolución diferencial, por otro lado, Engelbrecht (2007) integra los algoritmos de Programación Evolutiva, Evolución Cultural y Coevolución.

Todas las clases de algoritmos evolutivos desarrollados, se han creado bajo el concepto de simular la evolución a través de los procesos de selección, mutación y reproducción

(Rutkowski, 2005). Algunos ejemplos de ellos según (Bronwlee, 2011), (Jones, 2008) y (Engelbrecht, 2007) son:

1. Algoritmos que modelan la evolución genética.
2. Programación genética que se basa en algoritmos genéticos, pero los individuos son programas (representados como árboles).
3. Programación evolutiva que se deriva de la simulación del comportamiento adaptativo en evolución (evolución fenotípica).
4. Estrategias de evolución orientadas a modelar los parámetros de la estrategia que controlan la variación en la evolución, es decir, la evolución de la evolución.
5. Evolución diferencial, que es similar a los algoritmos genéticos, que difieren en el mecanismo de reproducción utilizado.
6. Evolución cultural que modela la evolución de la cultura de una población y como la cultura influye en la evolución genética y fenotípica de los individuos.
7. Coevolución donde los individuos inicialmente “mudos” evolucionan a través de la cooperación, o en competencia unos con otros, adquiriendo las características necesarias para sobrevivir.

El principio fundamental de los algoritmos evolutivos se basa en los esquemas propuestos por Charles Darwin en la década de 1850, con su teoría de la evolución y selección natural. Estos algoritmos implican la selección natural, así como la evolución, y se centran en encontrar las mejores soluciones a los problemas y transmitirlos a generaciones futuras. El factor importante es la supervivencia que se logra mediante la reproducción (Engelbrecht, 2007). Los individuos nacen y mueren, pero por medio de la reproducción pasan material genético a las siguientes generaciones, los que heredan las mejores características sobreviven y los que heredan las malas, se consideran soluciones poco significativas y se desechan (Jones, 2008).

Los algoritmos evolutivos utilizan una población de individuos, en la figura 3.1 se representa a un individuo de una población, donde un individuo se conoce como cromosoma. Un cromosoma define las características de los individuos de la población. Cada característica se conoce como gen. El valor de un gen se denomina alelo. Para cada generación, los individuos

compiten para reproducir su descendencia. Aquellos individuos con las mejores capacidades de supervivencia tienen la mejor oportunidad de reproducirse.

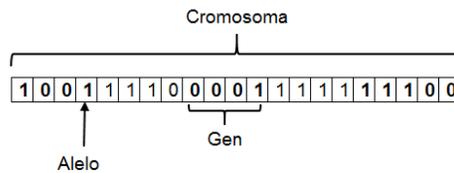


Figura 3.1 Representación de cromosoma, gen y alelo (Coello, 2019)

Los descendientes se generan combinando partes de los padres en un proceso conocido como cruce. Cada individuo en la población también puede sufrir una mutación que altera algunos de los alelos del cromosoma. La fuerza de supervivencia de un individuo, se mide usando una función de aptitud, que refleja los objetivos y las restricciones del problema a ser resuelto. Después de cada generación, los individuos pueden someterse a sacrificios, o los individuos pueden sobrevivir a la siguiente generación, proceso que se le conoce como elitismo.

Las características del comportamiento (como encapsulado de los fenotipos) se pueden utilizar para repercutir en el proceso evolutivo de dos maneras: 1) que los fenotipos influyan en los cambios genéticos, 2) que las características de comportamiento evolucionen por separado (Engelbrecht, 2007).

Los algoritmos evolutivos tratan de imitar los procesos de la evolución natural, principalmente la selección y la generación de nuevos genotipos descendientes de los individuos de la población.

A pesar de la inspiración biológica y los logros de rendimiento en las técnicas de inteligencia artificial aplicada, los sistemas difieren de la inteligencia humana de maneras cruciales. Revisando el avance en la ciencia cognitiva, se ha sugerido que las máquinas verdaderamente humanas de aprendizaje y pensamiento tendrán que ir más allá de las tendencias actuales de la ingeniería tanto en lo que aprenden y en cómo lo aprenden. Específicamente las máquinas, algoritmos y sistemas deben:

1. Construir modelos causales del mundo que apoyen la explicación y la comprensión, en lugar de simplemente resolver problemas de reconocimiento de patrones.
2. Aterrizar las teorías intuitivas de la física y la psicología, para apoyar y enriquecer el conocimiento que se adquiere.
3. Aprovechar la composición y aprender a aprender para adquirir y generalizar rápidamente conocimientos a nuevas tareas y situaciones.

Se sugieren desafíos concretos y rutas prometedoras hacia estos objetivos que pueden combinar las fortalezas de los recientes avances en redes neuronales con modelos cognitivos más estructurados (Lake et al., 2016).

En términos generales, para simular el proceso evolutivo en una computadora se requiere:

- Codificar las estructuras que se replicarán.
- Operaciones que afecten a los “individuos”.
- Una función de aptitud.
- Un mecanismo de selección.

De acuerdo a la dificultad para distinguir las diferencias entre los distintos tipos de algoritmos evolutivos y razones históricas, los tres paradigmas principales son: programación evolutiva, estrategias evolutivas y algoritmos genéticos.

### 3.4.1 Programación evolutiva

Lawrence J. Fogel propuso en la década de los 60s una técnica denominada “programación evolutiva”, en la cual la inteligencia se ve como un comportamiento adaptativo. La programación evolutiva enfatiza los nexos de comportamiento entre padres e hijos, en vez de buscar emular operadores genéticos específicos (como en el caso de los algoritmos genéticos), una representación general de programación evolutiva se muestra en la figura 3.2.

El algoritmo básico de la programación evolutiva es el siguiente:

1. Generar aleatoriamente una población inicial.
2. Se aplica la mutación.
3. Se calcula la aptitud de cada hijo y se usa un proceso de selección mediante torneo (selección estocástica) para determinar cuáles serán las soluciones que se retendrán.

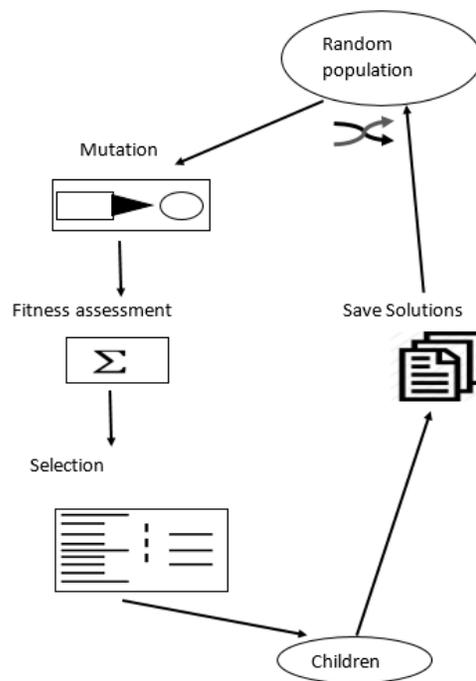


Figura 3.2 Representación conceptual de programación evolutiva (Coello, 2019)

### 3.4.2 Estrategias evolutivas

Las estrategias evolutivas fueron desarrolladas en 1964 en Alemania para resolver problemas hidrodinámicos de alto grado de complejidad por un grupo de estudiantes de ingeniería encabezado por Ingo Rechenberg.

La versión original (1+1)-EE usaba un solo padre y con él se generaba un solo hijo. Este hijo se mantenía si era mejor que el padre, o de lo contrario se eliminaba (a este tipo de selección se le llama extintiva, porque los peores individuos obtienen una probabilidad de ser seleccionado de cero).

En la (1+1)-EE, un individuo nuevo es generado usando la ecuación 2:

$$x^{-t+1} = x^{-t} + N(0, \bar{\sigma}) \quad (2)$$

donde  $t$  se refiere a la generación (o iteración) en la que se encuentra el proceso, y  $N(0, \bar{\sigma})$  es un vector de números Gaussianos independientes con una media de cero y desviaciones estándar  $\bar{\sigma}$  (Coello, 2018).

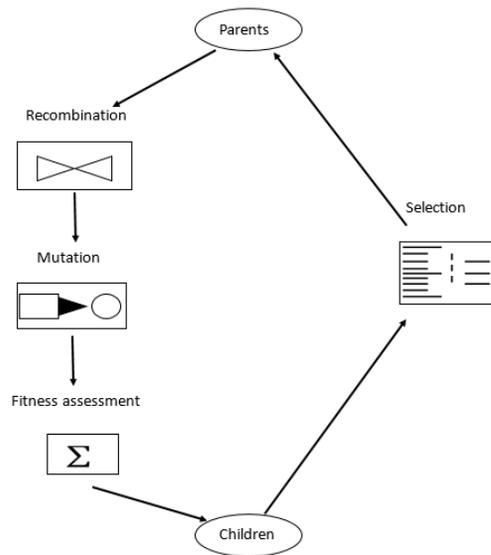


Figura 3.3 Representación conceptual de estrategias evolutivas (Coello, 2019)

### 3.4.3 Algoritmos Genéticos

El algoritmo genético enfatiza la importancia de la cruce sexual (operador principal) sobre la mutación (operador secundario) y usa selección probabilística. El algoritmo básico es el siguiente:

1. Generar aleatoriamente una población inicial.
2. Calcular la aptitud de cada individuo.
3. Seleccionar (probabilísticamente) con base en la aptitud.
4. Aplicar operadores genéticos (cruce y mutación) para generar la siguiente población
5. Ciclar hasta que la condición se satisfaga.

En la figura 3.4, se conceptualizan los algoritmos genéticos.

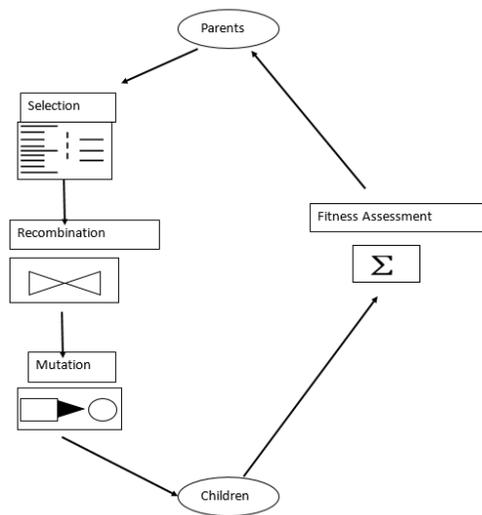


Figura 3.4 Representación conceptual de AG (Hernández et al., 2018)

## Capítulo 4. Procedimiento y metodología

### 4.1 Metodología de investigación con cómputo evolutivo para la perfilación geográfica criminal

En un proceso de investigación, la metodología es una de las fases con que cuenta un proyecto, en esta, el investigador o investigadores deciden el conjunto de técnicas y métodos que se emplearan para ejecutar las tareas vinculadas a la investigación.

En la metodología usada para la perfilación geográfica criminal mediante cómputo evolutivo, es necesario resaltar la importancia de la participación de expertos en temas de seguridad pública, prevención delictiva e informáticos.

El papel de los expertos en esta metodología radica en la aplicación de sus habilidades de investigación, formación, conocimiento general, trabajo en equipo y adaptación de sus habilidades para la solución de problemas.

Los métodos empleados en la investigación se describen a continuación:

**Cuantitativo:** Este método consiste en el conteo de los eventos delictivos registrados (denunciados) en un periodo de tiempo que puede ser día, semana, mes y años. Así mismo, la base de datos de estos registros se amplía con la investigación en distintas fuentes sobre los delitos no denunciados, pero que ocurrieron dentro de las demarcaciones del espacio geográfico a estudiar.

En la adquisición de estos datos, es importante la participación de instituciones de seguridad pública, ya que son ellos quienes llevan el registro de denuncias y en esos registros las características de los eventos delictivos.

Los expertos implicados en este método son la policía preventiva y policía de investigación.

**Cualitativo:** Apoyados en el análisis y estudio de la información de los eventos delictivos, se determina que eventos son recurrentes y peculiaridades de cada uno como: ubicación del

evento, tipo de delito, hora, fecha, día mes, día semana, mes y datos clave sobre un perímetro. Los datos clave sobre el perímetro son designados por los expertos y deben ser ajustados al entorno socioeconómico de la zona, estos pueden ser escuelas, bancos, cualquier tipo de comercio.

Además, la clasificación de los delitos y su impacto es determinada por la secretaria de seguridad pública de la entidad, con esa clasificación se hace la selección correcta de las variables que conforman un evento, estos eventos pueden ser: tipos de robo, homicidio, secuestro, violación, extorsión entre otros.

Los expertos vinculados en este método son: criminólogos, policía preventiva, policía de investigación.

### **Inductivo**

En este método de investigación, participan distintos personajes como: criminólogos, policía de investigación, policía preventiva e informáticos investigadores. El rol de cada uno de ellos permite la visualización de panoramas de solución distintos.

### **Criminólogo**

- Observación: Analiza sistemáticamente el espacio geográfico de un delito, aplicando la teoría del círculo.
- Observar un patrón: Analiza sistemáticamente las variables de un delito y su interpretación estadística, identifica y clasifica peculiaridades de la zona con otras.
- Desarrollar una teoría: Después del análisis de las zonas geográficas de los delitos, brinda un conjunto de sugerencias y recomendaciones que ayudan en la prevención. Propone la implementación de cámaras, alumbrado, análisis de tránsito peatonal, vehicular, conteo de escuelas, número de tiendas, etc. Estas recomendaciones dan partida a la determinación de las variables a considerar para el estudio de los casos.

## **Policía preventiva, Policía de investigación**

- Observación: Levantamiento de las evidencias y registro de las características de los delitos denunciados, fecha, hora, ubicación, semana, mes, características de la ubicación y alrededores.
- Observar un patrón: Identificación de las similitudes de los delitos y las zonas. La policía preventiva, recorre de forma preventiva las calles de la zona e identifica rasgos entre estas. La policía de investigación hace un análisis específico sobre un área y sobre un delito o grupo de delitos en particular.
- Desarrollar una teoría: Las teorías de estos dos personajes dentro de la investigación, se basa principalmente en la experiencia que han adquirido al paso de los años en el mismo trabajo.

### **Informático investigador:**

- Observación: en este proceso, se encarga de hacer un modelado de datos de los eventos delictivos.
- Observar un patrón: Con el análisis estadístico de los eventos delictivos y con la aplicación de algoritmos genéticos, estrategias evolutivas y programación evolutiva, se crean datos heurísticos basados en los eventos previos.
- Desarrollar una teoría: Con el estudio de los datos generados heurísticamente, se hace una perfilación geográfica criminal.

### **Deductivo**

En la implementación de este método, se toman las premisas de los expertos como punto de partida. Por mencionar algunas:

Criminólogo: “La ampliación e instalación de alumbrado público, reducen el índice delictivo”; Esta premisa ayuda a determinar si el alumbrado público es un factor que frena o no la comisión de un delito.

Policía preventiva: “Los lugares con mayor afluencia peatonal, es más susceptible para la comisión de un delito”.

Policía de investigación: “La mayoría de los robos se cometen en fines de semana”.

Informático: “La representación binaria de los eventos delictivos, favorece el procesado de datos”.

## **Comparativo**

Se analizan los datos recabados de las denuncias para hacer una clasificación de delitos y zonas geográficas afectadas. La comparación debe ser cuantitativa y cualitativa. Todos los expertos participan en este proceso.

### **Método lógico-deductivo:**

Policía preventiva: Basándose en su experiencia, es el encargado de determinar las zonas geográficas y su índice delictivo.

Policía de investigación: Se encarga del estudio de campo, aplicando su experiencia para determinar las variables de los delitos.

Criminólogo: Genera un conjunto de estrategias para la prevención del delito, cada estrategia se determina por las características geográficas.

Informático: Determina la representación óptima de las variables de cada evento, creando un modelo de datos, que adaptativo a las técnicas generales de cómputo evolutivo.

Las técnicas a aplicar en una metodología para la perfilación geográfica criminal, son:

#### Documentales.

- Análisis de los datos registrados: Estudio del historial delictivo de la zona geográfica.
- Selección de la técnica de cómputo evolutivo: Dependiendo del tipo de evento delictivo a perfilar, se opta por la selección de la técnica, sea algoritmo genético, programación evolutiva, estrategias evolutivas o una hibridación de éstas.

#### De campo

- Entrevista: Se aplica a la población, esto permitirá tener otra perspectiva del problema, lo que ayudará en la selección y determinación de variables.
- Observación: Consiste en la visita a las ubicaciones donde se cometió el delito y sus alrededores, para determinar variables adicionales que favorecen o limitan los tipos de delitos. Es aquí donde se considera el análisis de las variables como: bancos, escuelas, tiendas, iglesias, centros nocturnos, alumbrado público, avenidas de entrada y salida, entre otras. La determinación de estas variables se debe ajustar a la zona geográfica en cuestión, población y tipo de delitos.

#### Experimentales

- Modelado de datos: Cada uno de los eventos delictivos, son tomados formados por variables, cada evento es un cromosoma, y cada una de las variables es un gen del cromosoma; para la aplicación de los algoritmos genéticos, es recomendable usar un modelo binario, para las estrategias evolutivas, se usa una representación con números reales, la programación evolutiva usa autómatas finitos. La aplicación de estas tres técnicas, ayudan en la solución del problema y en distintos puntos de esta solución, se aplica cada una de las técnicas.
- Manipulación de información: Este proceso se lleva a cabo dentro del procesamiento de los algoritmos genéticos, estrategias evolutivas y programación evolutiva.

- Simulación de eventos: a heurística generada por el procesamiento de datos, da las nuevas ubicaciones de los eventos y sus características geográficas; esto es la perfilación geográfica.

### 4.1.1 Tipo de investigación empleada

Desde el planteamiento del problema se inicia con la identificación de los elementos necesarios para la determinación de estrategias y métodos requeridos, uno de estos requerimientos es la necesidad de trabajar de forma colaborativa con expertos en temas de seguridad pública.

La atención a las llamadas ciudadanas, el desplazamiento a las zonas de conflicto o al punto del delito, el registro detallado de los sucesos, el análisis del espacio geográfico, el patrullaje de la zona, características con las que se cometió el evento delictivo, detalles de los alrededores donde ocurrió el evento delictivo, fechas, hora, son de las principales características que se consideraron para el desarrollo del proyecto.

En conjunto con los colaboradores, se plantó el siguiente esquema general de investigación para el desarrollo del algoritmo evolutivo para la perfilación geográfica criminal, en la figura 4.1 se muestra el esquema.

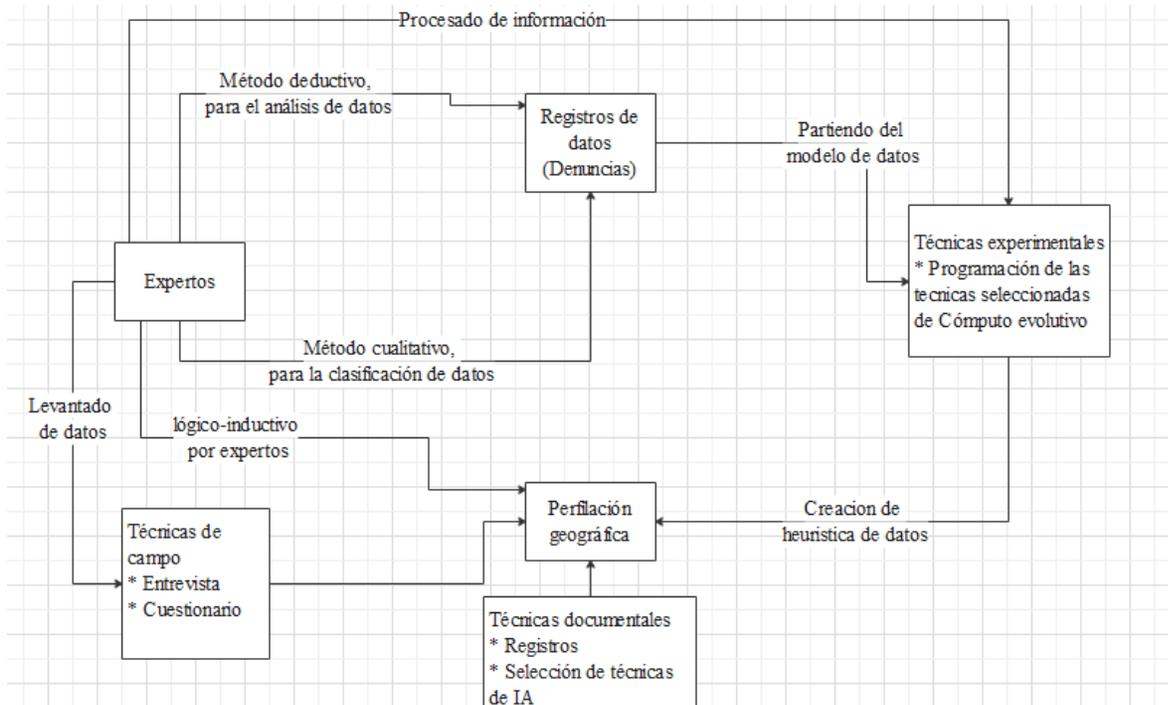


Figura 4.1 Esquema general de la investigación empleada para el desarrollo del algoritmo evolutivo para la perfilación geográfica criminal (propuesta por el autor)

## 4.2 Descripción del algoritmo evolutivo para la perfilación geográfica criminal

El algoritmo general propuesto para la perfilación geográfica criminal, se muestra a continuación, figura 4.2

```
1  Inicio
2  Determinar variables
3  Modelar variables
4  representar eventos
5  cargar datos de eventos
6  mientras (datos != "")
7      Crear población inicial
8      Leer padres
9      Mutación
10     Selección
11     Recombinación
12     Evaluación de aptitud
13     Selección de soluciones
14 fin mientras
15 Fin
```

Figura 4.2 Algoritmo general

Usando elementos de los algoritmos genéticos, las estrategias evolutivas y la programación evolutiva, se elabora una propuesta híbrida de algoritmo evolutivo, en la figura 4.3 se tiene una representación conceptual.

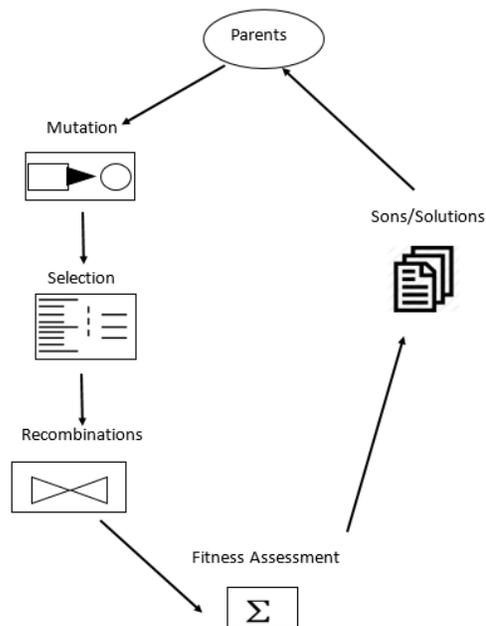


Figura 4.3 Representación conceptual de algoritmo evolutivo (Propuesta por el autor)

## Descripción general del algoritmo

- Determinación de variables: Se limitan a la ubicación geográfica, fecha y hora del evento, bancos, iglesias, tiendas, escuelas.
- Modelado de variables: Se determina la relación de los eventos con las características geográficas y elementos del entorno cercano (5 calles).
- Representación de eventos: Cada una de las variables son ajustadas para ser representadas en código binario.
- Se aplica proceso estocástico, se crea la población inicial, se identifican los padres, se mutan, se seleccionan, se recombinan, las posibles soluciones son elegidas en función de la fecha actual del sistema. Este proceso se repite hasta un valor asignado de forma inicial o hasta que el usuario decida detener la ejecución.

#### **4.2.1 Análisis general del problema y determinación de variables**

La inseguridad que se tiene en la zona de estudio, ha sido estigmatizada por los eventos delictivos y el impacto que estos han dejado en la población. Las diferentes formas de robo y el homicidio, se han mantenido como los principales delitos de la zona y han sido clasificados como delitos de alto impacto, esta peculiaridad se debe precisamente al impacto que genera, marcándola como zona roja o de alta peligrosidad.

La selección y estudio de estos dos tipos de eventos, permite la determinación de las variables que son usadas para el procesamiento de datos.

## 4.2.2 Algoritmo evolutivo para la perfilación geográfica criminal

El algoritmo evolutivo para la perfilación geográfica criminal propuesto, se muestra a continuación:

```
1 Inicio
2 nFile=0
3 numRun=0
4 mientras (F1.press=False or numRun<=300)
5     numRun++
6     Leer archivonFile.txt
7     eventos[]=eventos.txt
8     //cada evento
9     Cada evento evento[i]={diaSem, diaMes, mesActual ...}
10    n=len(eventos)//largo de eventos[]
11    aleatorio=[0]
12    Pi=[0] //población inicial
13    Pim = [0] //población inicial mutada
14    i=0
15
16    //preparando la población inicial
17    invoca Function rndEventos (Num.Eventos)
18    i=0
19    mientras (len(aleatorio)<n)
20        Pi[i]=eventos[aleatorio[i]] //Población inicial
21        i=i+1
22    Fin mientras
23
24    //Mutación uniforme aleatoria
25    probMutacion=1/largo(Evento[0]) // 1 entre 110
26    Para x=0 hasta n
27        num=random(0, largo (eventos[0]))
28        Si eventos[x][num]==0 entonces
29            eventos[x][num]=1
30            Pim[x]=eventos[x]
31        En caso contrario
32            eventos[x][num]=0
33            Pim[x]=eventos[x]
34
35    //selección aleatoria, se logra diversidad
36    invoca function rdnEventos(#eventos)
37        eventSelected[]=aleatorio[]
38
39
40    //recombinación PComb población combinada
```

```

41 //cruce en un punto
42 mientras (i<=(#eventos)*0.5)
43     ptsCruce[i]=aleatorio(0,largo(eventos[0]))
44     i++
45 fin mientras
46
47 para x=0 hasta x<= #eventos
48     //plev1 parte 1 de evento 1
49     plev1[]=0
50     p2ev1[]= 0
51     plev2[]=0
52     p2ev2[]= 0
53     cont=0
54     cont2=0
55     mientras (cont<=ptsCruce[x])
56         plev1.append(eventos[0][cont])
57         plev2.append(eventos[1][cont])
58         cont++
59     fin mientras
60     cont2=ptsCruce[x]
61     mientras (cont2<=largo(eventos[x]))
62         p2ev1.append(eventos[0][cont2])
63         p2ev2.append(eventos[1][cont2])
64         cont2++
65     fin mientras
66     //nueva población combinada pComb
67     pComb[x].append=plev1+p2ev2
68     pComb[x+1].append=plev2+p2ev1
69     x=x+2
70 Fin para
71
72 //Evaluación de aptitud
73 invocar calcMedia(pComb)
74 invocar calcModa(pComb)
75 invocar calcMediana(pComb)
76 invocar calcdesEstandar(pComb)
77 invocar calcVarianza(pComb)
78 //datos por fecha actual del Sistema
79 diaSemana=0 //[[1-7] valores que puede tomar
80 diaMes =0 //[[1-31] valores que puede tomar
81 mesActual=0 //[[1-12] valores que puede tomar
82
83 para x=0 hasta tamaño(pComb)
84     si pComb[x].diaSem>=diaSemana and
85     pComb[x].diaMes>=diaMes and
86     pComb[x].mesActual=mesActual entonces
87         hijos[x]= pComb[x]

```

```

88         en caso contrario si pComb[x].mesActual=
89         mesActual.moda entonces
90             hijos[x]=pComb[x]
91         en caso contrario si pComb[x].diaSem=diaSem.moda
92         entonces
93             hijos[x]=pComb[x]
94         en caso contrario si pComb[x].diaSem= diaSem.media
95         entonces
96             hijos[x]=pComb[x]
97         fin si
98     fin para
99
100     Guardar ArchivonFile.txt=hijos.
101     nFile=nFile+1
102 fin mientras
103
104 //FUNCIONES
105 // Función que Genera números aleatorios entre 0 y número de //
106 eventos
107 Function rndEventos (#eventos)
108     n=0
109     Mientras n<#eventos
110         aleatorio[n]=random(0, #eventos)
111         n=n+1
112     Fin Mientras
113     regresa aleatorio[]
114 Fin Function rndEventos
115
116 Function calcMedia (eventos)
117     media = media(eventos) //Pim población inicial mutada
118     regresa media
119 Fin function calcMedia
120
121 Function calcMediana (eventos)
122     mediana=mediana(eventos)
123     regresa mediana
124 Fin Function calcMedia
125
126 Function calcModa (eventos)
127     moda= moda(eventos)
128     regresa moda
129 Fin function calcModa
130
131 Function calcdesvEst(eventos)
132     desvEstandar=desvEstandar(eventos)
133     regresa desvEstandar
134 Fin function calcdesvEst
135

```

```
136 Function calcCorrelacion(x,y)
137     correlaciones[x]=correlacion(x+1)
138     regresa correlaciones
139 Fin function calcCorrelación ()
140
141 Function calcVarianza(eventos)
142     Varianzas=var(eventos)
143     Regresa Varianza
144 Fin function calcVarianza
145 Fin
```

### 4.2.3 Representación de la población

Una vez realizado el levantamiento de datos y análisis de los mismos en compañía de distintos expertos en seguridad pública y criminólogos, se determinan las variables que formarán y representarán cada uno de los delitos, en las figuras 4.4 y 4.5 se muestra la representación de las variables.

Día de la semana	Día de la semana número	Día de la semana binario	Día de la semana binario (3 dígitos) TODOS	Día del mes	Día del mes, binario	Día del mes, binario (5 dígitos)	Mes	Mes en binario	Mes en binario (4 dígitos)	Hora	hora en binario	Hora en binario (5 dígitos)	Coordenadas (Latitud, Longitud) <b>1. Separa coordenadas</b>
LUNES	1 1	001	4	100	00100	1	1	0001	4	100	00100	19.291264, -98.930971	
MIÉRCOLES	3 11	011	6	110	00110	1	1	0001	2	10	00010	19.295907, -98.953367	
JUEVES	4 100	100	7	111	00111	1	1	0001	15	1111	01111	19.286818, -98.940359	
DOMINGO	7 111	111	10	1010	01010	1	1	0001	14	1110	01110	19.280806, -98.943228	
MARTES	2 10	010	12	1100	01100	1	1	0001	15	1111	01111	19.306241, -98.947869	
MARTES	2 10	010	12	1100	01100	1	1	0001	17	10001	10001	19.324517, -98.950771	
VIERNES	5 101	101	15	1111	01111	1	1	0001	18	10010	10010	19.274153, -98.936121	
VIERNES	5 101	101	22	10110	10110	1	1	0001	13	1101	01101	19.283347, -98.928590	
SABADO	6 110	110	23	10111	10111	1	1	0001	11	1011	01011	19.282286, -98.944894	
DOMINGO	7 111	111	24	11000	11000	1	1	0001	0	0	00000	19.286470, -98.924038	
MARTES	2 10	010	26	11010	11010	1	1	0001	7	111	00111	19.284856, -98.949333	
VIERNES	5 101	101	31	11111	11111	1	1	0001	22	10110	10110	19.301632, -98.948587	
LUNES	1 1	001	1	1	00001	2	10	0010	2	10	00010	19.288267, -98.931200	
MIÉRCOLES	3 11	011	3	11	00011	2	10	0010	21	10101	10101	19.284856, -98.949333	
VIERNES	5 101	101	5	101	00101	2	10	0010	18	10010	10010	19.280830, -98.943249	
MARTES	2 10	010	9	1001	01001	2	10	0010	4	100	00100	19.303676, -98.939845	
JUEVES	4 100	100	11	1011	01011	2	10	0010	11	1011	01011	19.300627, -98.949945	

Figura 4.4 Bloque 1 de la representación de datos

Escuelas	Escuelas binario	escuelas binario (5 dígitos)	Bares	bares binario	bares binario (5 dígitos)	Iglesias	iglesias binario	iglesias binario (5 dígitos)	Número de tiendas	tiendas binario	tiendas binario (5 dígitos)	Bancos	bancos binario	bancos binario (5 dígitos)	Delito homicidio: 01 Robo: 10
2 10	00010	1 1	00001	2 10	00010	3 1001	01001	1 1	00001	10					
5 101	00101	0 0	00000	2 10	00010	6 110	00110	0 0	00000	10					
2 10	00010	0 0	00000	0 0	00000	9 1001	01001	0 0	00000	10					
2 10	00010	1 1	00001	3 11	00011	8 1000	01000	4 100	00100	10					
6 110	00110	0 0	00000	4 100	00100	7 111	00111	0 0	00000	10					
1 1	00001	1 1	00001	4 100	00100	5 101	00101	0 0	00000	10					
3 11	00011	0 0	00000	2 10	00010	4 100	00100	0 0	00000	10					
2 10	00010	2 10	00010	2 10	00010	4 100	00100	0 0	00000	10					
4 100	00100	1 1	00001	4 100	00100	8 1000	01000	4 100	00100	10					
1 1	00001	0 0	00000	4 100	00100	7 111	00111	0 0	00000	10					
1 1	00001	0 0	00000	1 1	00001	8 1000	01000	0 0	00000	10					
3 11	00011	2 10	00010	3 11	00011	4 100	00100	3 11	00011	10					
3 11	00011	0 0	00000	3 11	00011	4 100	00100	0 0	00000	10					
2 10	00010	0 0	00000	1 1	00001	3 11	00011	0 0	00000	10					
2 10	00010	1 1	00001	3 11	00011	8 1000	01000	4 100	00100	10					
5 101	00101	2 10	00010	3 11	00011	3 1001	01001	1 1	00001	10					
3 11	00011	2 10	00010	3 11	00011	4 100	00100	3 11	00011	10					

Figura 4.5 Bloque 2 de la representación de datos

Las variables de las figuras 4.4 y 4.5 se representan de forma decimal y se hace una conversión a binario para para el procesamiento de estas. Cada una de las variables usa un numero determinado de bits que corresponde a la conversión binaria del decimal

correspondiente, posteriormente, se hace la agrupación de eventos delictivos en código binario, la representación final de los eventos se muestra en la figura 4.6.

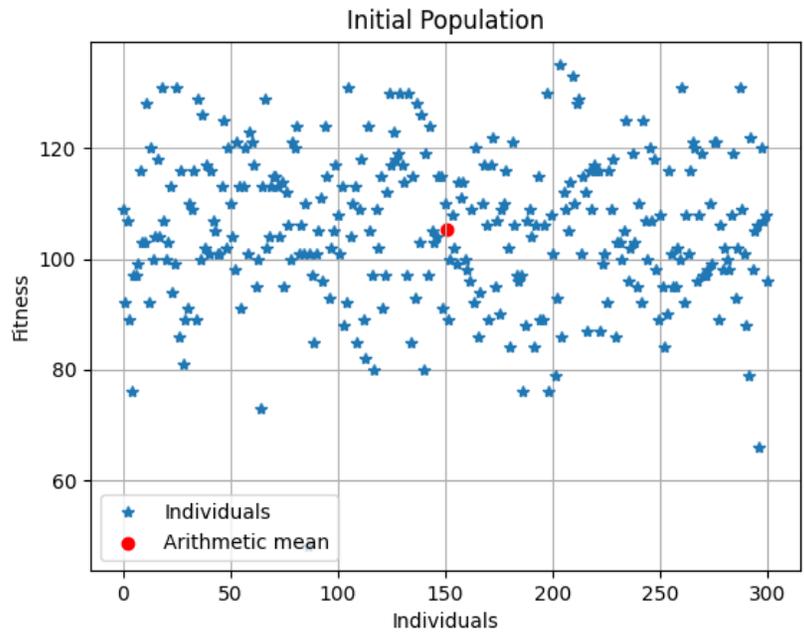
```

001 00100 0001 00100 00110001 00111001 01100000 00110010 00111001 00110001 00110010 00110110 00110100 00111101
011 00110 0001 00010 00110001 00111001 01100000 00110010 00111001 00110101 00111001 00110000 00110111 00111101
100 00111 0001 01111 00110001 00111001 01100000 00110010 00111000 00110110 00111000 00110001 00111000 00111101
111 01010 0001 01110 00110001 00111001 01100000 00110010 00111000 00110000 00111000 00110000 00110110 00111101
010 01100 0001 01111 00110001 00111001 01100000 00110011 00110000 00110110 00110010 00110100 00110001 00111101
010 01100 0001 10001 00110001 00111001 01100000 00110011 00110010 00110100 00110101 00110001 00110111 00111101
101 01111 0001 10010 00110001 00111001 01100000 00110010 00110111 00110100 00110001 00110101 00110011 00111101
101 10110 0001 01101 00110001 00111001 01100000 00110010 00111000 00110011 00110011 00110100 00110111 00111101
110 10111 0001 01011 00110001 00111001 01100000 00110010 00111000 00110010 00110010 00111000 00110110 00111101
111 11000 0001 00000 00110001 00111001 01100000 00110010 00111000 00110110 00110100 00110111 00110000 00111101
010 11010 0001 00111 00110001 00111001 01100000 00110010 00111000 00110100 00111000 00110101 00110110 00111101
101 11111 0001 10110 00110001 00111001 01100000 00110011 00110000 00110001 00110110 00110011 00110010 00111101
001 00001 0010 00010 00110001 00111001 01100000 00110010 00111000 00111000 00110010 00110110 00110111 00111101

```

Figura 4.6 Representación binaria de los eventos y sus variables de la población

Una vez agrupados los datos en su forma binaria, se hace una reducción de caracteres repetidos, en las variables latitud y longitud se quitan los dos y tres primeros dígitos respectivamente, ya que estos corresponden en todo momento a la zona de estudio y el cromosoma que originalmente se forma con 204 bits, se representa con solo 110 bits. Se hace una representación decimal única de cada individuo y la población inicial se muestra en la gráfica 4.1.



Gráfica 4.1 Población inicial

## El bloque del algoritmo

```
Leer archivonnFile.txt
eventos[]=eventos.txt
//cada evento
Cada evento evento[i]={diaSem, diaMes,mesActual ...}
n=len(eventos)//largo de eventos[]
aleatorio=[0]
Pi=[0] //población inicial
Pim = [0] //población inicial mutada
i=0

//preparando la población inicial
invoca Function rndEventos (Num.Eventos)
i=0
mientras (len(aleatorio)<n)
    Pi[i]=eventos[aleatorio[i]] //Población inicial
    i=i+1
Fin mientras
```

#### 4.2.4 Proceso de Mutación

El proceso de mutación es aleatorio por inversión de bits, en la ecuación 3 se representa la probabilidad de mutación de un alelo de cada evento es

$$p(a) = 1/na \quad (3)$$

sustituyendo,

$$p(a) = 1/300=0.009090.$$

Para un evento:

```
00110010011101001110011110000110011110111110110110101110011110100110001111
000110001000000000000000001100000010
```

Se tiene una probabilidad de mutación  $p(a)$ , haciendo la inversión de bits, el individuo anterior es modificado en un alelo:

```
001100100111010011100111100001100111101111010110110101110011110100110001111
000110001000000000000000001100000010
```

Este proceso de mutación es durante cada una de las iteraciones de ejecución del algoritmo:

```
//Mutación uniforme aleatoria
probMutacion=1/largo(Evento[0]) // 1 entre 110
Para x=0 hasta n
    num=random(0, largo(eventos[0]))
    Si eventos[x][num]==0 entonces
        eventos[x][num]=1
        Pim[x]=eventos[x]
    En caso contrario
        eventos[x][num]=0
        Pim[x]=eventos[x]

//selección aleatoria, se logra diversidad
invoca function rdnEventos(#eventos)
    eventSelected[]=aleatorio[]
```

#### 4.2.5 Selección de individuos

Partiendo de la población inicial, la selección de los individuos para la creación de nuevas poblaciones, es estocástica, esta selección de individuos evoluciona en función de las poblaciones previas, se hace una selección generacional, donde los hijos seleccionados en cada generación, se convierten en la generación siguiente.

```
//selección aleatoria, se logra diversidad
invoca function rndEventos(#eventos)
    eventSelected[]=aleatorio[]
```

```
Function rndEventos (#eventos)
    n=0
    Mientras n<#eventos
        aleatorio[n]=random(0,#eventos)
        n=n+1
    Fin Mientras
    regresa aleatorio[]
Fin Function rndEventos
```

## 4.2.6 Recombinación en la población

Parala recombinación de individuos, se hace la cruza aleatoria en un punto y se generan los nuevos individuos

```
P1=00101001010111000110010110111110101111001110100110000110010110101110110  
110010111001001000000000011001010000001
```

```
P2=00110010011101001110011110000110011110111110110110101110011110100110001  
111000110001000000000000000001100000010
```

Creando los nuevos individuos

H1:

```
00101001011101001110011110000110011110111110110110101110011110100110001111  
0001100010000000000000000001100000010
```

H2:

```
00110010010111000110010110111110101111001110100110000110010110101110110110  
010111001001000000000011001010000001
```

Este proceso de recombinación se aplica desde n=0 hasta n=300, que es la población inicial y número de poblaciones

```
//recombinación PComb población combinada  
//cruce en un punto  
mientras (i<=#eventos)*0.5)  
    ptsCruce[i]=aleatorio(0,largo(eventos[0]))  
    i++  
fin mientras  
  
para x=0 hasta x<= #eventos  
    //plev1 parte 1 de evento 1  
    plev1[]=0  
    p2ev1[]= 0  
    plev2[]=0  
    p2ev2[]= 0  
    cont=0
```

```

cont2=0
mientras (cont<=ptsCruce[x])
    p1ev1.append(eventos[0][cont])
    p1ev2.append(eventos[1][cont])
    cont++
fin mientras
cont2=ptsCruce[x]
mientras (cont2<=largo(eventos[x]))
    p2ev1.append(eventos[0][cont2])
    p2ev2.append(eventos[1][cont2])
    cont2++
fin mientras
//nueva población combinada pComb
pComb[x].append=p1ev1+p2ev2
pComb[x+1].append=p1ev2+p2ev1
x=x+2
Fin para

```

#### 4.2.7 Evaluación de aptitud

La determinación de los individuos más aptos, se calcula por la distancia más corta que existe con la media de la población, la media se determina a cada población y a cada variable del evento, ver ecuación 4,

$$\mu = \frac{1}{n} \sum x_i \quad (4)$$

para identificar la dispersión de cada individuo, en la ecuación 5, se tiene que

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n-1}} \quad (5)$$

Por lo que la función de aptitud será dada por la ecuación 6:

$$f(x) = (e_i - (\mu P_{i+1})) \quad (6)$$

donde  $e_i$ , son las variables de cada individuo, y  $p_i$ , son las poblaciones que van generando. La menor diferencia que exista entre individuos y medias de la población de cada una de las variables independientes, hará que se considere a un individuo como apto.

Segmento del algoritmo para la evaluación de la aptitud

```
//Evaluación de aptitud
invocar calcMedia (pComb)
invocar calcModa (pComb)
invocar calcMediana (pComb)
invocar calcdesEstandar (pComb)
invocar calcVarianza (pComb)
//datos por fecha actual del Sistema
diaSemana=0 // [1-7] valores que puede tomar
diaMes =0 // [1-31] valores que puede tomar
mesActual=0 // [1-12] valores que puede tomar

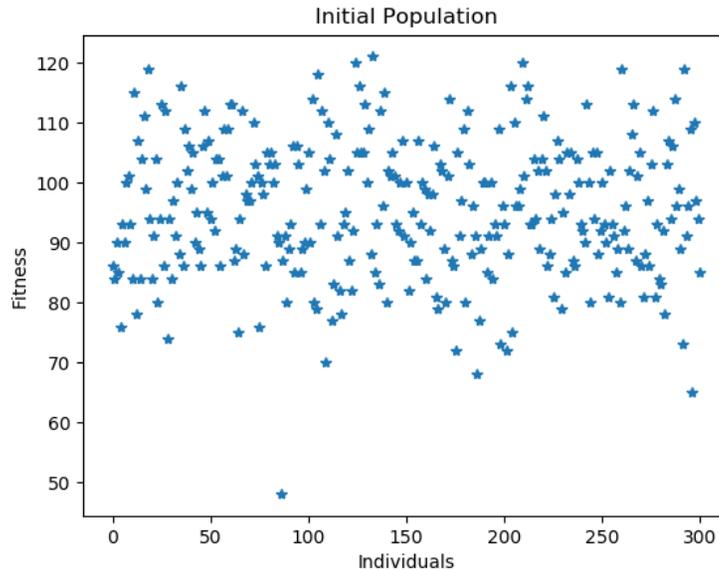
para x=0 hasta tamaño (pComb)
  si pComb[x].diaSem>=diaSemana and pComb[x].diaMes>=diaMes
  and pComb[x].mesActual=mesActual entonces
    hijos[x]= pComb[x]
```

```
en caso contrario si pComb[x].mesActual= mesActual.moda
entonces
    hijos[x]=pComb[x]
en caso contrario si pComb[x].diaSem=diaSem.moda entonces
    hijos[x]=pComb[x]
en caso contrario si pComb[x].diaSem= diaSem.media entonces
    hijos[x]=pComb[x]
fin si
fin para

Guardar ArchivosFile.txt=hijos.
nFile=nFile+1
fin mientras
```

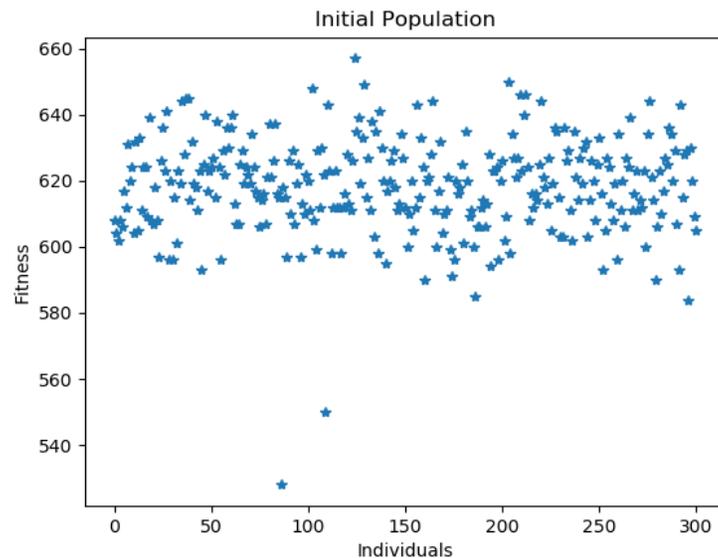
## Capítulo 5. Resultados

La población inicial que se trabajó se muestra en la gráfica 5.1 y 5.2, es importante destacar que en esta gráfica se consideran todos los elementos que forman a un individuo de la población inicial.



Gráfica 5.1 Población inicial, eventos sin la posición geográfica

En la gráfica 5.2, a la población inicial, se le agregan los datos que representan las coordenadas de los eventos delictivos, manteniéndose la relación gráfica de cada uno de los eventos.

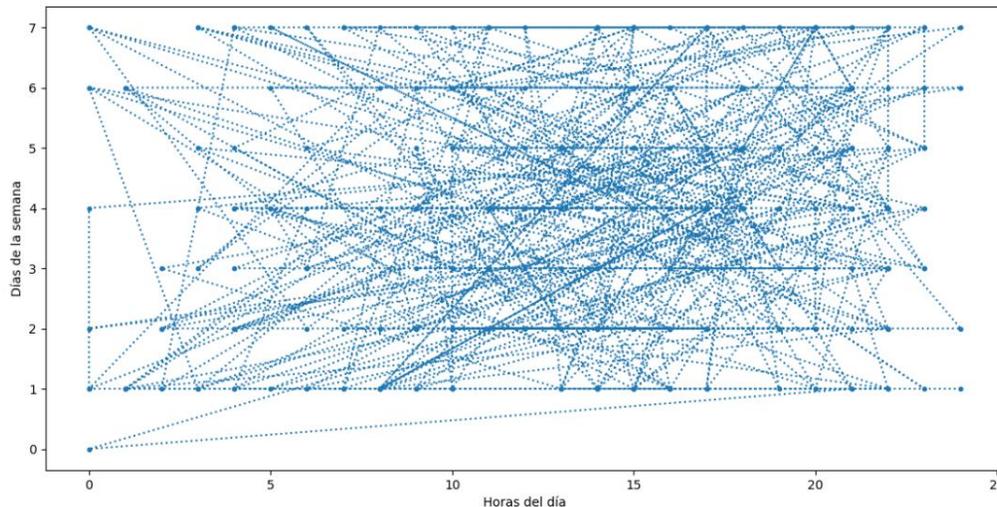


Gráfica 5.2 Población inicial, eventos con la posición geográfica

En la ecuación 7, se muestra la correlación entre los días y horas de la semana:

$$m = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (7)$$

Los valores a analizar son días y horas de la semana, se muestran en la gráfica 5.3,



Gráfica 5.3 Días de las semanas y horas de los eventos delictivos

Los valores enteros de los días y horas de las semanas, se muestran a continuación

Días de la semana y :

[1, 1, 7, 6, 3, 7, 2, 6, 6, 4, 1, 4, 1, 7, 3, 7, 7, 5, 5, 5, 1, 6, 4, 4, 2, 2, 1, 4, 2, 6, 3, 6, 5, 7, 3, 3, 1, 4, 4, 3, 7, 4, 2, 6, 1, 1, 4, 4, 1, 4, 5, 7, 2, 6, 4, 7, 1, 4, 4, 7, 7, 6, 7, 3, 2, 4, 3, 3, 3, 4, 2, 6, 3, 7, 3, 3, 7, 5, 1, 1, 5, 1, 4, 3, 5, 1, 0, 3, 1, 4, 2, 4, 4, 7, 4, 6, 7, 1, 7, 4, 7, 2, 7, 3, 1, 7, 4, 7, 4, 1, 4, 5, 1, 3, 5, 2, 4, 1, 5, 1, 4, 4, 1, 1, 6, 2, 5, 5, 3, 3, 4, 5, 1, 6, 2, 2, 7, 4, 2, 5, 2, 4, 7, 4, 1, 3, 2, 2, 2, 1, 7, 1, 7, 4, 6, 2, 1, 4, 1, 4, 2, 5, 4, 6, 3, 4, 2, 7, 7, 2, 4, 7, 7, 2, 1, 3, 6, 2, 3, 7, 2, 6, 6, 1, 1, 6, 2, 2, 5, 7, 7, 5, 1, 7, 6, 7, 2, 1, 1, 6, 4, 2, 1, 3, 3, 2, 5, 3, 6, 7, 6, 5, 5, 5, 1, 5, 7, 2, 2, 4, 6, 4, 5, 7, 4, 2, 4, 7, 7, 7, 5, 6, 6, 4, 2, 4, 6, 4, 1, 4, 5, 2, 5, 1, 2, 5, 3, 4, 4, 7, 7, 1, 6, 2, 2, 2, 5, 2, 6, 3, 7, 7, 6, 2, 2, 1, 4, 4, 7, 4, 4, 3, 6, 7, 5, 1, 6, 1, 2, 2, 3, 4, 3, 5, 5, 1, 4, 5, 7, 6, 1, 1, 6, 4, 6, 7, 1, 6, 5, 4, 3]

Horas del día x :

[24, 9, 24, 10, 3, 11, 9, 5, 21, 14, 20, 17, 15, 20, 19, 7, 14, 10, 17, 18, 17, 18, 13, 18, 7, 20, 1, 8, 9, 1, 10, 19, 23, 23, 4, 16, 15, 21, 4, 14, 9, 15, 19, 16, 13, 16, 11, 17, 8, 17, 21, 11, 8, 23, 13, 12, 13, 15, 4, 21, 15, 10, 15, 14, 0, 23, 20, 17, 9, 20, 20, 21, 6, 8, 16, 22, 19, 13, 15, 17, 22, 20, 5, 6, 15, 21, 0, 17, 7, 9, 14, 16, 9, 18, 22, 18, 15, 14, 22, 22, 10, 13, 6, 11, 14, 20, 15, 5, 15, 16, 3, 19, 13, 2, 21, 16, 19, 3, 21, 8, 17, 8, 16, 21, 16, 14, 12, 18, 17, 20, 21, 10, 10, 15, 2, 24, 17, 20, 9, 16, 2, 21, 3, 23, 5, 13, 14, 17, 10, 1, 17, 8, 17, 17, 21, 14, 8, 22, 15, 5, 16, 3, 21, 0, 17, 9, 17, 14, 22, 19, 13, 23, 15, 10, 22, 21, 10, 21, 22, 0, 6, 15, 9, 10, 1, 12, 10, 13, 23, 16, 11, 4, 16, 22, 11, 3, 17, 22, 4, 18, 12, 9, 6, 22, 14, 4, 18, 12, 21, 20, 15, 19, 18, 13, 23, 13, 0, 16, 22, 17, 11, 18, 6, 18, 11, 13, 22, 9, 21, 14, 13, 24, 1, 11, 22, 13, 22, 19, 4, 6, 23, 4, 17, 8, 22, 20, 16, 17, 14, 4, 15, 3, 0, 11, 16, 10, 9, 12, 19, 23, 19, 7, 18, 14, 16, 14, 11, 13, 14, 18, 15, 19, 16, 20, 18, 19, 15, 9, 15, 16, 22, 12, 23, 10, 17, 0, 0, 22, 20, 8, 0, 7, 9, 11, 15, 4, 2, 16, 15, 18, 14]

### Elementos $x^2$

[576, 81, 576, 100, 9, 121, 81, 25, 441, 196, 400, 289, 225, 400, 361, 49, 196, 100, 289, 324, 289, 324, 169, 324, 49, 400, 1, 64, 81, 1, 100, 361, 529, 529, 16, 256, 225, 441, 16, 196, 81, 225, 361, 256, 169, 256, 121, 289, 64, 289, 441, 121, 64, 529, 169, 144, 169, 225, 16, 441, 225, 100, 225, 196, 0, 529, 400, 289, 81, 400, 400, 441, 36, 64, 256, 484, 361, 169, 225, 289, 484, 400, 25, 36, 225, 441, 0, 289, 49, 81, 196, 256, 81, 324, 484, 324, 225, 196, 484, 484, 100, 169, 36, 121, 196, 400, 225, 25, 225, 256, 9, 361, 169, 4, 441, 256, 361, 9, 441, 64, 289, 64, 256, 441, 256, 196, 144, 324, 289, 400, 441, 100, 100, 225, 4, 576, 289, 400, 81, 256, 4, 441, 9, 529, 25, 169, 196, 289, 100, 1, 289, 64, 289, 289, 441, 196, 64, 484, 225, 25, 256, 9, 441, 0, 289, 81, 289, 196, 484, 361, 169, 529, 225, 100, 484, 441, 100, 441, 484, 0, 36, 225, 81, 100, 1, 144, 100, 169, 529, 256, 121, 16, 256, 484, 121, 9, 289, 484, 16, 324, 144, 81, 36, 484, 196, 16, 324, 144, 441, 400, 225, 361, 324, 169, 529, 169, 0, 256, 484, 289, 121, 324, 36, 324, 121, 169, 484, 81, 441, 196, 169, 576, 1, 121, 484, 169, 484, 361, 16, 36, 529, 16, 289, 64, 484, 400, 256, 289, 196, 16, 225, 9, 0, 121, 256, 100, 81, 144, 361, 529, 361, 49, 324, 196, 256, 196, 121, 169, 196, 324, 225, 361, 256, 400, 324, 361, 225, 81, 225, 256, 484, 144, 529, 100, 289, 0, 0, 484, 400, 64, 0, 49, 81, 121, 225, 16, 4, 256, 225, 324, 196]

### Elementos $xy$

[24, 9, 168, 60, 9, 77, 18, 30, 126, 56, 20, 68, 15, 140, 57, 49, 98, 50, 85, 90, 17, 108, 52, 72, 14, 40, 1, 32, 18, 6, 30, 114, 115, 161, 12, 48, 15, 84, 16, 42, 63, 60, 38, 96, 13, 16, 44, 68, 8,

68, 105, 77, 16, 138, 52, 84, 13, 60, 16, 147, 105, 60, 105, 42, 0, 92, 60, 51, 27, 80, 40, 126, 18, 56, 48, 66, 133, 65, 15, 17, 110, 20, 20, 18, 75, 21, 0, 51, 7, 36, 28, 64, 36, 126, 88, 108, 105, 14, 154, 88, 70, 26, 42, 33, 14, 140, 60, 35, 60, 16, 12, 95, 13, 6, 105, 32, 76, 3, 105, 8, 68, 32, 16, 21, 96, 28, 60, 90, 51, 60, 84, 50, 10, 90, 4, 48, 119, 80, 18, 80, 4, 84, 21, 92, 5, 39, 28, 34, 20, 1, 119, 8, 119, 68, 126, 28, 8, 88, 15, 20, 32, 15, 84, 0, 51, 36, 34, 98, 154, 38, 52, 161, 105, 20, 22, 63, 60, 42, 66, 0, 12, 90, 54, 10, 1, 72, 20, 26, 115, 112, 77, 20, 16, 154, 66, 21, 34, 22, 4, 108, 48, 18, 6, 66, 42, 8, 90, 36, 126, 140, 90, 95, 90, 65, 23, 65, 0, 32, 44, 68, 66, 72, 30, 126, 44, 26, 88, 63, 147, 98, 65, 144, 6, 44, 44, 52, 132, 76, 4, 24, 115, 8, 85, 8, 44, 100, 48, 68, 56, 28, 105, 3, 0, 22, 32, 20, 45, 24, 114, 69, 133, 49, 108, 28, 32, 14, 44, 52, 98, 72, 60, 57, 96, 140, 90, 19, 90, 9, 30, 32, 66, 48, 69, 50, 85, 0, 0, 110, 140, 48, 0, 7, 54, 44, 90, 28, 2, 96, 75, 72, 42]

Para la correlación entre los días y horas de la semana, se usa la ecuación 8:

$$m = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (8)$$

Sustituyendo

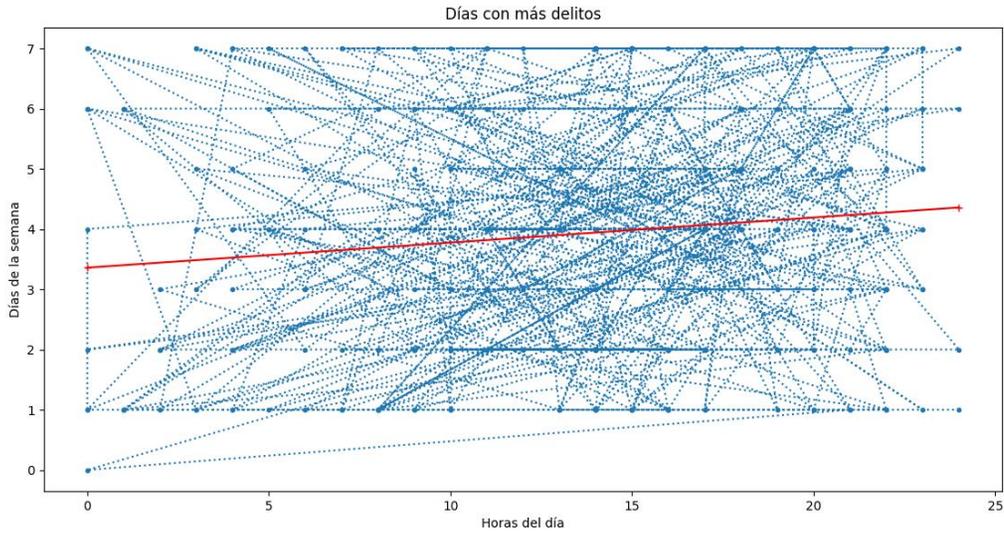
$$m = \frac{300(16787) - (4138)(1181)}{300(69014) - (4138)^2} = 0.04162443301143913$$

Para el eje de las y, trazar la recta y revisar la relación de forma gráfica, para  $x = 0$  en y

$$b = \bar{y} - m\bar{x} = 3.35938656$$

Punto  $p_1$  (0, 3.35938656),

Para  $x=24$ , queda  $y=(0.0416)(24)+3.3593 =4.357$ , quedando  $p_2(24,4.357)$ , en la gráfica 5.4, se traza la recta junto a días y horas.



Gráfica 5.4 Regresión lineal entre los días y horas de los eventos delictivos

Al interpretar la gráfica, se puede apreciar que los días con mayor incidencia delictiva, son los días miércoles y jueves, siendo este último el que tiene mayor recurrencia delictiva, se puede hacer un contraste con la gráfica 5.5.



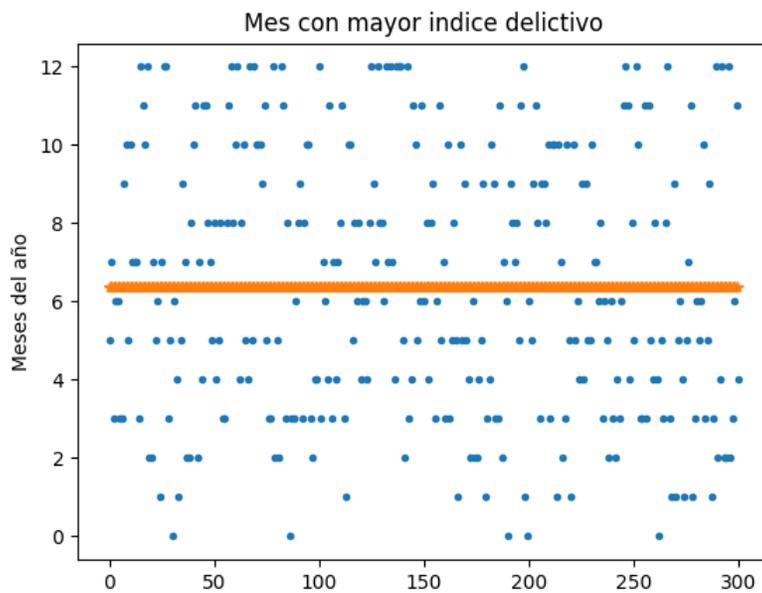
Gráfica 5.5 Día con más eventos delictivos

En otros resultados, se puede apreciar la hora en que se cometen más delitos, en la gráfica 5.6 se señala mediante una x, que la hora con mayor recurrencia delictiva, son las 17 hrs.



Gráfica 5.6 Hora con más incidencia delictiva

En la gráfica 5.7 se puede apreciar el mes con mayor índice delictivo.



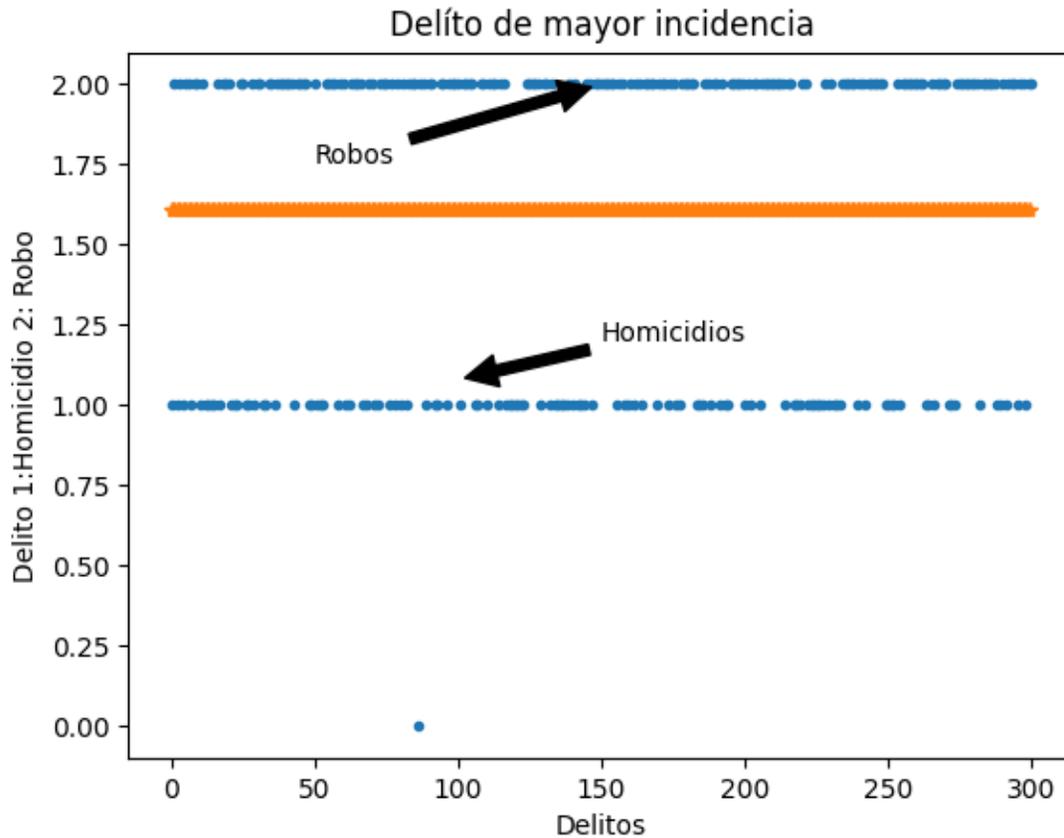
Gráfica 5.7 Mes con mayor índice delictivo

En la gráfica 5.8, se muestra el día del mes con mayor registro de delitos.



Gráfica 5.8 Día del mes con alta recurrencia delictiva

El delito de mayor recurrencia es el robo en sus distintas modalidades, se aprecia en la gráfica 5.9, la línea en color naranja. El desplazamiento de la línea naranja hacia los robos a lo largo de las ejecuciones del algoritmo, indican el delito más susceptible de ocurrencia.



Gráfica 5.9 El robo es el delito de mayor incidencia, representado por la línea naranja

Con los individuos de prueba

01100100010110110110011110000111001110100111001110101110100110000111001110  
111110010000010000000010001100000010

1111010100100000011001011100111001111011110000110101110001110100110  
100110001000010000000000001010000010

0010111100101000111001011100111100011010011000011000011001111010111001111  
001110010000110000100011001100000010

00100111101110000110010111000110110110010110000110000110100110010110001110  
010110110000110000000000001110000010

10011010010101101110010111000110001110011110000110001110001110001110101110  
010110010000110000000011001110000101

11010100100010111110010111001110010111000110000111000110101111001110100110  
11111001000000000000001001010000001

1000101010110111110011110000110001110110110100110010110100111000110110110  
000111001001100000000100001110000010

00101000100001000110010111000110101110011110110110010110010110011110011110  
100110001000010000000100001110000110

11110111101010100110011110000110001110110110111110011110100111000110101111  
001111000000110001000011001000001110

01010110001101110110010111000110000111000110100110111110010111000110010110  
100110001000010000100010000110000101

11001000101010101110011110000110001110110110100110010110100111000110101110  
110110110000110001000011001000001110

11011011100010000110010111001111001110001110110110100110100111000111001110  
101111000000110001000011001000001110

00101110001100111110010111000111001111001110011110100110101110000110100110  
000111000000100000100001010100000010

10111110100101100110010111001110000110001110100111000110001111000110001111  
001110110000110000000010001100000010

01010110001101110110010111000110000111000110100110111110010111000110010110  
100110001000010000100010000110000101

01010000001010011110010111001111001110100110101110111110101110000110110110  
001110011000110001000011001000001110

01111110101110110110010111000110001111001111000111000110100110001111001110  
100110110000010001000010001010001110

11110001010010100110010111000110100110110110110110000110001111000110011111  
000110101000000000000011001010000001

10001010101101111110011110000110001110110110100110010110100111000110110110  
000111001001100000000100001110000010

01100001011001011110010111000110110111001110001110010110101110100110111110  
100110001001000000100001010100000010

## Capítulo 6. Conclusiones

El uso de los algoritmos evolutivos en la solución de problemas con alto índice de complejidad, los han convertido en una opción viable, principalmente en la solución de problemas de optimización y predicción.

En este trabajo se presenta el desarrollo de un algoritmo evolutivo para la perfilación geográfica criminal, que se hace una combinación de técnicas de algoritmos genéticos, estrategias evolutivas y programación evolutiva para el proceso de predicción. Se puede usar como una herramienta adicional en la prevención del delito y mediante el ajuste de variables, la implementación del mismo en distintas áreas geográficas, es posible. Estas deben ajustarse al tipo de delito y al espacio geográfico en cuestión. Las variables seleccionadas para el análisis, fueron determinadas de forma multidisciplinaria al igual que los eventos seleccionados. De una Base de datos con 3005 registros, fueron considerados 300 registros, ya que fueron clasificados como de alto impacto.

Se realizó la ejecución del algoritmo para 300 generaciones y en los resultados de la ejecución e investigación desarrollada, se obtuvo información sobre los días, horas, meses y posibles ubicaciones donde puede ocurrir un evento delictivo. Con éste estudio y análisis de los datos, se ha permitido la identificación de varios parámetros que son de ayuda para seguridad pública. Por lo anterior, se concluye que el algoritmo evolutivo planteado en este trabajo y en conjunto con la metodología de investigación aplicada, se puede usar como una herramienta y estrategia en la prevención del delito.

## 6.1 Recomendaciones

En este apartado se hacen algunas recomendaciones para dar una idea general del ajuste del algoritmo en caso de su aplicación en otros espacios geográficos

- Este algoritmo fue diseñado para trabajar con 12 variables, mismas que fueron seleccionadas por un grupo multidisciplinario y en función del delito y de la zona de estudio. De esa manera, se deberá ajustar el número de variables de acuerdo a las necesidades propias de la investigación.
- El algoritmo fue programado para ejecutarse una cantidad de veces igual al tamaño de la población (300), si se desea cambiar este valor, se debe asignar un valor a la variable *nReg*, de la clase *crearMatrizCls*.
- Los resultados que arroja al final de la ejecución, son graficas de análisis de los registros e información textual, se debe hacer la interpretación de los mismos.

## 6.2 Trabajos futuros

La realización de esta investigación ha dejado vertientes para el desarrollo de nuevos elementos que permitan optimizar los procesos de seguridad pública.

Varios de los procesos de investigación que se llevan a cabo por los expertos en seguridad, muestran características para ser abordados desde un punto de vista computacional, desde el desarrollo de algoritmos que optimicen sus actividades convencionales de seguridad, hasta el planteamiento de nuevas técnicas de monitoreo, registro y control de evidencias.

- El planteamiento principal futuro de esta investigación, es poder determinar los puntos geográficos que presentan o pueden presentar algún evento delictivo, directamente en un mapa de la zona.
- Desarrollar la interfaz para la implementación del algoritmo a problemas de perfilaciones generales.
- Actualización del algoritmo de forma tal que permita la selección de la técnica informática e inteligencia artificial a utilizar.

# Apéndice

## A.1 Código fuente del algoritmo en Python

### A.1.1 Clase para cargar la información de un archivo de texto y crear la matriz principal de los eventos delictivos

```
import numpy as np
2
3 class creaMatrizCls():
4     nReg = 0 # almacenará el número de registros
5     diaSem = '0' # 3 dígitos
6     diaMes = 0 # 5 dígitos
7     mes = 0 # 4 dígitos
8     hora = 0 # 5 dígitos
9     # latitud y longitud 8 dígitos por variable
10    latA = 0
11    latB = 0
12    latC = 0
13    latD = 0
14    latE = 0
15    latF = 0
16    latG = 0
17    latH = 0
18    latI = 0
19    coma = 0 # debe valer siempre 00101100
20    menos = 0
21    longA = 0
22    longB = 0
23    longC = 0
24    longD = 0
25    longE = 0
26    longF = 0
27    longG = 0
28    longH = 0
29    longI = 0
30    escuela = 0 # 5 dígitos
31    bar = 0 # 5 dígitos
32    iglesia = 0 # 5 dígitos
33    tienda = 0 # 5 dígitos
34    banco = 0 # 5 dígitos
35    tipoDelito = 0 # 2 dígitos
36    Matriz = "" # almacenará la información a procesar, la matriz de todos
los datos
37
38    individuos = ""
39
40    @classmethod
41    def crearMatrizMetod(cls): # es un método de clase
42        # archivo="códigos RobosHomicidios.txt" #valores ordenados en el
archivo
43        archivo = "PoblacionAleatoriaSep10.txt" # valores ordenados
ALEATORIAMENTE en el archivo
44        creaMatrizCls.individuos = str(np.loadtxt(archivo))
```

```

45
46         # se crea una matriz de 300 x30 300 registros de 30 datos cada uno
47         creaMatrizCls.diaSem, creaMatrizCls.diaMes, creaMatrizCls.mes,
creaMatrizCls.hora, \
48         creaMatrizCls.latA, creaMatrizCls.latB, creaMatrizCls.latC,
creaMatrizCls.latD, creaMatrizCls.latE, \
49         creaMatrizCls.latF, creaMatrizCls.latG, creaMatrizCls.latH,
creaMatrizCls.latI, creaMatrizCls.coma, \
50         creaMatrizCls.menos, creaMatrizCls.longA, creaMatrizCls.longB,
creaMatrizCls.longC, \
51         creaMatrizCls.longD, creaMatrizCls.longE, creaMatrizCls.longF,
creaMatrizCls.longG, creaMatrizCls.longH, \
52         creaMatrizCls.longI, creaMatrizCls.escuela, creaMatrizCls.bar,
creaMatrizCls.iglesia, \
53         creaMatrizCls.tienda, creaMatrizCls.banco, creaMatrizCls.tipoDelito =
np.loadtxt(archivo, dtype='str').T
54
55         # Matriz, es una matriz de 300x30
56
57         creaMatrizCls.Matriz = [creaMatrizCls.diaSem, creaMatrizCls.diaMes,
creaMatrizCls.mes, creaMatrizCls.hora,
58                                 creaMatrizCls.latA,
59                                 creaMatrizCls.latB, creaMatrizCls.latC,
creaMatrizCls.latD, creaMatrizCls.latE,
60                                 creaMatrizCls.latF, creaMatrizCls.latG,
creaMatrizCls.latH, creaMatrizCls.latI,
61                                 creaMatrizCls.coma, creaMatrizCls.menos,
62                                 creaMatrizCls.longA, creaMatrizCls.longB,
creaMatrizCls.longC, creaMatrizCls.longD,
63                                 creaMatrizCls.longE, creaMatrizCls.longF,
64                                 creaMatrizCls.longG, creaMatrizCls.longH,
creaMatrizCls.longI, creaMatrizCls.escuela,
65                                 creaMatrizCls.bar, creaMatrizCls.iglesia,
66                                 creaMatrizCls.tienda, creaMatrizCls.banco,
creaMatrizCls.tipoDelito]
67
68         creaMatrizCls.nReg = len(creaMatrizCls.diaSem) # obteniendo la
cantidad de registros de la matriz
69

```

## A.1.2 Clase para preparar la población inicial y optimizarla

```
1  # SEGUNDA
2  # en esta clase, se crean los individuos cortos
3  import creaMatrizCls as creaMatriz #importando la clase que crea la matriz de
datos y creando el objeto de la clase
4  import numpy as np
5
6  class GeneraPoblacion ():
7      largoCromosoma=0
8      poblacionInicial=[]
9      individuo=[] #individuo largo, sin latA,latB,latC, sin
longA,longB,longC,longD (longitud total 132)
10     individuoOptimo=[] # individuo corto, a variables de latitud y longitud,
quito los indices 0,1,2, quedando solo de 5 dígitos (3,4,5,6,7)
11     indOptimoCorto=[] # a latitudes y longitudes, se les recortan los primeros
dos dígitos, para optimizar, solo queda del 2 al 7 (longitud total 110)
12     # debido a que las latitudes y longitudes se repiten los dos primeros
dígitos, al cruzarlos, si los dos primeros cambian, habrá inconsistencias en el
13     # resultado final, los dos primeros dígitos son constantes, PARA ESTE
PROBLEMA
14     indOptimoCortoEnt=[] # el individuo óptimo en decimal, matriz con todos
los valores
15     latDcorta=[]
16     latEcorta = []
17     latFcorta = []
18     latGcorta = []
19     latHcorta = []
20     latIcorta = []
21     longEcorta = []
22     longFcorta = []
23     longGcorta = []
24     longHcorta = []
25     longIcorta = []
26     latPrueba2 = []
27     diaSemEnt=[]
28     diaMesEnt=[]
29     mesEnt=[]
30     horaEnt=[]
31     latDcortaEnt=[]
32     latEcortaEnt = []
33     latFcortaEnt = []
34     latGcortaEnt = []
35     latHcortaEnt = []
36     latIcortaEnt = []
37     longEcortaEnt = []
38     longFcortaEnt = []
39     longGcortaEnt = []
40     longHcortaEnt = []
41     longIcortaEnt = []
42     escuelaEnt=[]
43     barEnt=[]
44     iglesiaEnt=[]
45     tiendaEnt=[]
46     bancoEnt=[]
47     tipodelitoEnt=[]
48     numRegistros=0
```

```

49     #variables para almacenar los cálculos estadísticos de cada una de las
poblaciones
50     #mediaArtemetica
51
52
53     #El método generaIndividuos, genera una población completa, donde los
individuos tiene 132 genes, a latitud y longitud no se la ha quitado ningún Gen
54     @classmethod
55     def generaIndividuos(cls):
56
57         #cargando los valores del método creamatriz
58         creaMatriz.creaMatrizCls.crearMatrizMetod()
59         #obteniendo el número de registros del arreglo
60         nReg=len(creaMatriz.creaMatrizCls.diaSem)
61         GeneraPoblacion.numRegistros=len(creaMatriz.creaMatrizCls.diaSem) #
número de registros de la población
62         # desde 0 hasta el tamaño del arreglo o vectores en el archivo de
texto
63         #generando individuos independientes
64         # valores de latitud descartados
65         # latitudes (latA,latB,latC, latD(solo puede ser 2 o 3))
66         #valores de longitud descartados
67         # longitudes longA,longB,longC,longD
68         #longE, solo toma valores (0-6)
69
70         #latD, solo puede ser 2 o 3,
71         #si latD=2, latE, ira de 0 a 9
72         #si latD=3, latE sera 0 o 1
73         for x in range(0, nReg):
74
75             GeneraPoblacion.individuo.append(creaMatriz.creaMatrizCls.diaSem[x]+creaMatriz.crea
MatrizCls.diaMes[x]+creaMatriz.creaMatrizCls.mes[x]+
76             creaMatriz.creaMatrizCls.hora[x]+creaMatriz.creaMatrizCls.latD[x]+creaMatriz.creaMa
trizCls.latE[x]+
77             creaMatriz.creaMatrizCls.latF[x]+creaMatriz.creaMatrizCls.latG[x]+creaMatriz.creaMa
trizCls.latH[x]+
78             creaMatriz.creaMatrizCls.latI[x] +
79             creaMatriz.creaMatrizCls.longE[x]+creaMatriz.creaMatrizCls.longF[x]+
80             creaMatriz.creaMatrizCls.longG[x]+creaMatriz.creaMatrizCls.longH[x]+creaMatriz.crea
MatrizCls.longI[x]+
81             creaMatriz.creaMatrizCls.escuela[x] +
82             creaMatriz.creaMatrizCls.bar[x]+creaMatriz.creaMatrizCls.iglesia[x]+
83             creaMatriz.creaMatrizCls.tienda[x]+creaMatriz.creaMatrizCls.banco[x]+creaMatriz.crea
MatrizCls.tipoDelito[x])
84
85
86         #*****MÉTODO DE CLASE*****
87         #en este método se creará una población con menos genes, para optimizar
tiempo de ejecución
88
89         #este método, recorta las variables de latitud (latD a latI) y longitud
(longE a longI), dejándolas de 8 bits en solo 6 bits, para optimizar el
procesamiento de datos
90         # los primeros dígitos de las variables son 00

```

```

88     @classmethod
89     def creaIndOptimoCorto(cls):
90         creaMatriz.creaMatrizCls.crearMatrizMetod()
91         # obteniendo el número de registros del arreglo
92         nReg = len(creaMatriz.creaMatrizCls.diaSem)
93         GeneraPoblacion.numRegistros = len(creaMatriz.creaMatrizCls.diaSem) #
número de registros de la poblacion
94         creaMatriz.creaMatrizCls.crearMatrizMetod() #cargando los valores del
método de la clase creaMatriz
95         a=0
96         b=0
97         z=0
98         # haciendo cortas las latitudes y longitudes
99         for a in range(0, nReg):
100             #Creando las matrices de datos, cortos, los demás se quedan igual
101             GeneraPoblacion.latDcorta.append([])
102             GeneraPoblacion.latEcorta.append([])
103             GeneraPoblacion.latFcorta.append([])
104             GeneraPoblacion.latGcorta.append([])
105             GeneraPoblacion.latHcorta.append([])
106             GeneraPoblacion.latIcorta.append([])
107             GeneraPoblacion.longEcorta.append([])
108             GeneraPoblacion.longFcorta.append([])
109             GeneraPoblacion.longGcorta.append([])
110             GeneraPoblacion.longHcorta.append([])
111             GeneraPoblacion.longIcorta.append([])
112
113             for b in range(2, 8):
114                 #cargando valores a los vectores solo 6 valores del índice 2
al 7
115                 # de las variable latitud y longitud de a-i y de e-i
116
117             GeneraPoblacion.latDcorta[a].append(creaMatriz.creaMatrizCls.latD[a][b])
118             GeneraPoblacion.latEcorta[a].append(creaMatriz.creaMatrizCls.latE[a][b])
119             GeneraPoblacion.latFcorta[a].append(creaMatriz.creaMatrizCls.latF[a][b])
120             GeneraPoblacion.latGcorta[a].append(creaMatriz.creaMatrizCls.latG[a][b])
121             GeneraPoblacion.latHcorta[a].append(creaMatriz.creaMatrizCls.latH[a][b])
122             GeneraPoblacion.latIcorta[a].append(creaMatriz.creaMatrizCls.latI[a][b])
123             GeneraPoblacion.longEcorta[a].append(creaMatriz.creaMatrizCls.longE[a][b])
124             GeneraPoblacion.longFcorta[a].append(creaMatriz.creaMatrizCls.longF[a][b])
125             GeneraPoblacion.longGcorta[a].append(creaMatriz.creaMatrizCls.longG[a][b])
126             GeneraPoblacion.longHcorta[a].append(creaMatriz.creaMatrizCls.longH[a][b])
127             GeneraPoblacion.longIcorta[a].append(creaMatriz.creaMatrizCls.longI[a][b])
128
129             #se usa para recortar la longitud de las variables latitud y longitud
130             # a cada variable se le quitan dos dígitos
131             for y in range (0, nReg):

```

```

132         for z in range (0,6):
133             GeneraPoblacion.indOptimoCorto.append(
134                 creaMatriz.creaMatrizCls.diaSem[y] +
creaMatriz.creaMatrizCls.diaMes[y] +
135                 creaMatriz.creaMatrizCls.mes[y] +
creaMatriz.creaMatrizCls.hora[y] + GeneraPoblacion.latDcorta[y][z] +
136                 GeneraPoblacion.latEcorta[y][z] +
GeneraPoblacion.latFcorta[y][z] + GeneraPoblacion.latGcorta[y][z] +
137                 GeneraPoblacion.latHcorta[y][z] +
GeneraPoblacion.latIcorta[y][z] + GeneraPoblacion.longEcorta[y][z]+
138                 GeneraPoblacion.longFcorta[y][z] +
GeneraPoblacion.longGcorta[y][z] + GeneraPoblacion.longHcorta[y][z] +
139                 GeneraPoblacion.longIcorta[y][z] +
creaMatriz.creaMatrizCls.escuela[y] + creaMatriz.creaMatrizCls.bar[y] +
140                 creaMatriz.creaMatrizCls.iglesia[y] +
creaMatriz.creaMatrizCls.tienda[y] + creaMatriz.creaMatrizCls.banco[y] +
141                 creaMatriz.creaMatrizCls.tipoDelito[y])
142             #guardando la población inicial
143             GeneraPoblacion.poblacionInicial.append(
144                 creaMatriz.creaMatrizCls.diaSem[y] +
creaMatriz.creaMatrizCls.diaMes[y] +
145                 creaMatriz.creaMatrizCls.mes[y] +
creaMatriz.creaMatrizCls.hora[y] + GeneraPoblacion.latDcorta[y][
146                 z] +
147                 GeneraPoblacion.latEcorta[y][z] +
GeneraPoblacion.latFcorta[y][z] + GeneraPoblacion.latGcorta[y][
148                 z] +
149                 GeneraPoblacion.latHcorta[y][z] +
GeneraPoblacion.latIcorta[y][z] + GeneraPoblacion.longEcorta[y][
150                 z] +
151                 GeneraPoblacion.longFcorta[y][z] +
GeneraPoblacion.longGcorta[y][z] + GeneraPoblacion.longHcorta[y][
152                 z] +
153                 GeneraPoblacion.longIcorta[y][z] +
creaMatriz.creaMatrizCls.escuela[y] +
154                 creaMatriz.creaMatrizCls.bar[y] +
155                 creaMatriz.creaMatrizCls.iglesia[y] +
creaMatriz.creaMatrizCls.tienda[y] +
156                 creaMatriz.creaMatrizCls.banco[y] +
157                 creaMatriz.creaMatrizCls.tipoDelito[y])
158
159
160         #método que crea la población corta con 300 individuos ORIGINALES Y con
solo 110 genes
161         #CORRECTO, indOptimoCorto, tiene todos los registros originales
162         @classmethod
163         def metodPoblacionCorta(cls):
164             creaMatriz.creaMatrizCls.crearMatrizMetod()
165             GeneraPoblacion.creaIndOptimoCorto() # INVOCA EL MÉTODO QUE RECORTA
LOS CROMOSOMAS
166             GeneraPoblacion.indOptimoCorto.append([])
167             GeneraPoblacion.poblacionInicial.append([]) #población inicial
168             nReg = len(creaMatriz.creaMatrizCls.diaSem) # cuenta el número de
registros
169             GeneraPoblacion.numRegistros = len(creaMatriz.creaMatrizCls.diaSem) #
número de registros de la población
170             f = open('PoblacionInicial.txt', 'w') # se abre el archivo para
almacenar ahí los datos

```

```

171         for x in range (0,nReg):
172             for y in range (0,6): #creando los individuos cortos óptimos
CORRECTO
173 GeneraPoblacion.indOptimoCorto[x]=creaMatriz.creaMatrizCls.diaSem[x]+
creaMatriz.creaMatrizCls.diaMes[x] + \
174             creaMatriz.creaMatrizCls.mes[x] +
creaMatriz.creaMatrizCls.hora[x]+ """.join( GeneraPoblacion.latDcorta[x])+ """.join(
GeneraPoblacion.latEcorta[x]) + \
175             """.join( GeneraPoblacion.latFcorta[x])+ """.join(
GeneraPoblacion.latGcorta[x]) + """.join( GeneraPoblacion.latHcorta[x])+ """.join(
GeneraPoblacion.latIcorta[x])+\
176             """.join( GeneraPoblacion.longEcorta[x])+ """.join(
GeneraPoblacion.longFcorta[x])+ """.join( GeneraPoblacion.longGcorta[x])+ """.join(
GeneraPoblacion.longHcorta[x])+\
177             """.join(GeneraPoblacion.longIcorta[x])+"".join(creaMatriz.creaMatrizCls.escuela[x])
+"".join(creaMatriz.creaMatrizCls.bar[x])+"".join(creaMatriz.creaMatrizCls.iglesia[
x])+\
178             """.join(creaMatriz.creaMatrizCls.tienda[x])+"".join(creaMatriz.creaMatrizCls.banco[
x])+"".join(creaMatriz.creaMatrizCls.tipoDelito[x])
179
180             #población inicial, la misma pero en otra variable
181             GeneraPoblacion.poblacionInicial[x] =
creaMatriz.creaMatrizCls.diaSem[x] + \
182             creaMatriz.creaMatrizCls.diaMes[x] + \
183             creaMatriz.creaMatrizCls.mes[x] + creaMatriz.creaMatrizCls.hora[
184                                     x] +
185             """.join(GeneraPoblacion.latDcorta[x]) + """.join(
186                 GeneraPoblacion.latEcorta[x]) + \
187             """.join(GeneraPoblacion.latFcorta[x]) + """.join(
188                 GeneraPoblacion.latGcorta[x]) +
189             """.join(GeneraPoblacion.latHcorta[x]) + """.join(
190                 GeneraPoblacion.latIcorta[x]) + \
191             """.join(GeneraPoblacion.longEcorta[x]) + """.join(
192                 GeneraPoblacion.longFcorta[x]) +
193             """.join(GeneraPoblacion.longGcorta[x]) + """.join(
194                 GeneraPoblacion.longHcorta[x]) + \
195             """.join(GeneraPoblacion.longIcorta[x]) + """.join(
196                 creaMatriz.creaMatrizCls.escuela[x]) +
197             """.join(creaMatriz.creaMatrizCls.bar[x]) + """.join(
198                 creaMatriz.creaMatrizCls.iglesia[x]) + \
199             """.join(creaMatriz.creaMatrizCls.tienda[x]) + """.join(
200                 creaMatriz.creaMatrizCls.banco[x]) +
201             """.join(creaMatriz.creaMatrizCls.tipoDelito[x])
202             #GUARDANDO DATOS EN ARCHIVO DE TEXTO
203             #f = open('ARCHIVOTEXTO.txt', 'w')
204             f.write( creaMatriz.creaMatrizCls.diaSem[x] +
creaMatriz.creaMatrizCls.diaMes[x] + \
205             creaMatriz.creaMatrizCls.mes[x] +
creaMatriz.creaMatrizCls.hora[x] + """.join(GeneraPoblacion.latDcorta[x]) + """.join(

```

```

201             GeneraPoblacion.latEcorta[x]) +
""" .join(GeneraPoblacion.latFcorta[x]) + """ .join(GeneraPoblacion.latGcorta[x]) +
""" .join(GeneraPoblacion.latHcorta[x]) + """ .join(
202             GeneraPoblacion.latIcorta[x]) +
""" .join(GeneraPoblacion.longEcorta[x]) + """ .join(
203             GeneraPoblacion.longFcorta[x]) +
""" .join(GeneraPoblacion.longGcorta[x]) + """ .join(
204             GeneraPoblacion.longHcorta[x]) +
""" .join(GeneraPoblacion.longIcorta[x]) + """ .join(
205             creaMatriz.creaMatrizCls.escuela[x]) +
""" .join(creaMatriz.creaMatrizCls.bar[x]) + """ .join(
206             creaMatriz.creaMatrizCls.iglesia[x]) +
""" .join(creaMatriz.creaMatrizCls.tienda[x]) + """ .join(
207             creaMatriz.creaMatrizCls.banco[x]) +
""" .join(creaMatriz.creaMatrizCls.tipoDelito[x]) + '\n' )
208             #f.close() # cerrando el archivo
209             GeneraPoblacion.largoCromosoma =
len(GeneraPoblacion.indOptimoCorto[0]) # almaceno el largo del cromosoma
210             f.close() #cerrando archivo, poblacioninicial.txt
211
212             print("individuo corto 0: ",GeneraPoblacion.indOptimoCorto[0])
213             print("individuo corto 1: ", GeneraPoblacion.indOptimoCorto[1])
214             print("individuo corto 2: ", GeneraPoblacion.indOptimoCorto[2])
215             print("individuo corto 3: ", GeneraPoblacion.indOptimoCorto[3])
216
217
*****
218             #CREANDO UNA POBLACIÓN DE ENTEROS,
219
220             # LA MATRIZ BINARIA DE INDIVIDUOS CORTOS, SE PASA A NÚMEROS REALES, TODOS
ENTEROS
221
222
223             @classmethod
224             def poblacionNumEnteros(cls):# se cambió el nombre de "poblacionDecimales"
a "poblacionNumEnteros"
225                 creaMatriz.creaMatrizCls.crearMatrizMetod()
226                 GeneraPoblacion.creaIndOptimoCorto()
227
228                 nReg=len(creaMatriz.creaMatrizCls.diaSem) # número de registros
229                 for x in range (0,nReg):
230                     # Creando las matrices de datos en enteros
231
GeneraPoblacion.diaSemEnt.append(int(creaMatriz.creaMatrizCls.diaSem[x],2))
232
GeneraPoblacion.diaMesEnt.append(int(creaMatriz.creaMatrizCls.diaMes[x],2))
233
GeneraPoblacion.mesEnt.append(int(creaMatriz.creaMatrizCls.mes[x],2))
234
GeneraPoblacion.horaEnt.append(int(creaMatriz.creaMatrizCls.hora[x],2))
235
GeneraPoblacion.latDcortaEnt.append(int(creaMatriz.creaMatrizCls.latD[x], 2)) #
revisar
236
GeneraPoblacion.latEcortaEnt.append(int(creaMatriz.creaMatrizCls.latE[x], 2))
237
GeneraPoblacion.latFcortaEnt.append(int(creaMatriz.creaMatrizCls.latF[x], 2))

```

```

238
GeneraPoblacion.latGcortaEnt.append(int(creaMatriz.creaMatrizCls.latG[x], 2))
239
GeneraPoblacion.latHcortaEnt.append(int(creaMatriz.creaMatrizCls.latH[x], 2))
240
GeneraPoblacion.latIcortaEnt.append(int(creaMatriz.creaMatrizCls.latI[x], 2))
241
GeneraPoblacion.longEcortaEnt.append(int(creaMatriz.creaMatrizCls.longE[x], 2))
242
GeneraPoblacion.longFcortaEnt.append(int(creaMatriz.creaMatrizCls.longF[x], 2))
243
GeneraPoblacion.longGcortaEnt.append(int(creaMatriz.creaMatrizCls.longG[x], 2))
244
GeneraPoblacion.longHcortaEnt.append(int(creaMatriz.creaMatrizCls.longH[x], 2))
245
GeneraPoblacion.longIcortaEnt.append(int(creaMatriz.creaMatrizCls.longI[x], 2))
246
GeneraPoblacion.escuelaEnt.append(int(creaMatriz.creaMatrizCls.escuela[x], 2))
247
GeneraPoblacion.barEnt.append(int(creaMatriz.creaMatrizCls.bar[x], 2))
248
GeneraPoblacion.iglesiaEnt.append(int(creaMatriz.creaMatrizCls.iglesia[x], 2))
249
GeneraPoblacion.tiendaEnt.append(int(creaMatriz.creaMatrizCls.tienda[x], 2))
250
GeneraPoblacion.bancoEnt.append(int(creaMatriz.creaMatrizCls.banco[x], 2))
251
GeneraPoblacion.tipodelitoEnt.append(int(creaMatriz.creaMatrizCls.tipoDelito[x], 2))
252
253     print ("semana", GeneraPoblacion.diaSemEnt) #imprimiendo valores enteros
de diaSem, de toda la población
254     a=GeneraPoblacion.diaSemEnt[1]+GeneraPoblacion.diaSemEnt[2] #prueba
para sumar valores de la matriz
255     print ("la suma es:", a, "número de registros
es:", GeneraPoblacion.numRegistros) #imprimiendo la suma de los valores de la
matriz, solo de dos índices
256
257 GeneraPoblacion.metodPoblacionCorta() # recorta las longitudes y latitudes de
8 a 6

```

### A.1.3 Heurística y análisis de datos

```
1  # TERCERA
2  import seaborn as sb
3  from sklearn import linear_model
4  import GeneraPoblacion as generaPoblacion
5  import statistics as stat
6  import numpy as np
7  import matplotlib.pyplot as plt
8  import matplotlib.animation as animation
9  import scipy as scipy
10 import random as rnd
11
12
13 class estadistica:
14     # orden de las variables, como se almacenaran en la matriz
15     diaSem,diaMes,mes,hora,latD,latE,latF,latG,latH,latI,
16     # longE,longF,logG,longH,longI,escuela,bar,iglesia,tienda,banco,tipoDelito
17     0      1      2      3      4      5      6
18     # 7      8      9      10     11     12     13     14     15     16     17     18
19     20
20     # VARIABLES DE LOS CALCULOS ESTADISTICOS
21     # de toda la poblacion
22     pobTemp=[0] #poblacion temporal
23     poblacTemp=[0] #poblaciones temporales
24     evPoblacion=[0]
25     mediaAritIndEnt = [0]
26     modaIndEnt = [0]
27     varianzaIndEnt = [0]
28     correlacionIndEnt = [0]
29     medianaIndEnt = [0]
30     desEstandarIndEnt = [0]
31
32     generaciones = 0
33     mediaAritDiaSem = [0]
34     modaDiaSem = [0]
35     varianzaDiaSem = [0]
36     correlacionDiaSem = [0]
37     medianaDiaSem = [0]
38     desvEstandarDiaSem = [0]
39
40     mediaAritDiaMes = [0]
41     modaDiaMes = [0]
42     varianzaDiaMes = [0]
43     correlacionDiaMes = [0]
44     medianaDiaMes = [0]
45     desvEstandarDiaMes = [0]
46
47     mediaAritMes = [0]
48     modaMes = [0]
49     varianzaMes = [0]
50     correlacionMes = [0]
51     medianaMes = [0]
52     desvEstandarMes = [0]
```

```
53     varianzaHora = [0]
54     correlacionHora = [0]
55     medianaHora = [0]
56     desvEstandarHora = [0]
57
58     mediaAritLatD = [0]
59     modaLatD = [0]
60     varianzaLatD = [0]
61     correlacionLatD = [0]
62     medianaLatD = [0]
63     desvEstandarLatD = [0]
64
65     mediaAritLatE = [0]
66     modaLatE = [0]
67     varianzaLatE = [0]
68     correlacionLatE = [0]
69     medianaLatE = [0]
70     desvEstandarLatE = [0]
71
72     mediaAritLatF = [0]
73     modaLatF = [0]
74     varianzaLatF = [0]
75     correlacionLatF = [0]
76     medianaLatF = [0]
77     desvEstandarLatF = [0]
78
79     mediaAritLatG = [0]
80     modaLatG = [0]
81     varianzaLatG = [0]
82     correlacionLatG = [0]
83     medianaLatG = [0]
84     desvEstandarLatG = [0]
85
86     mediaAritLatH = [0]
87     modaLatH = [0]
88     varianzaLatH = [0]
89     correlacionLatH = [0]
90     medianaLatH = [0]
91     desvEstandarLatH = [0]
92
93     mediaAritLatI = [0]
94     modaLatI = [0]
95     varianzaLatI = [0]
96     correlacionLatI = [0]
97     medianaLatI = [0]
98     desvEstandarLatI = [0]
99
100    mediaAritLongE = [0]
101    modaLongE = [0]
102    varianzaLongE = [0]
103    correlacionLongE = [0]
104    medianaLongE = [0]
105    desvEstandarLongE = [0]
106
107    mediaAritLongF = [0]
108    modaLongF = [0]
109    varianzaLongF = [0]
110    correlacionLongF = [0]
```

```
111     medianaLongF = [0]
112     desvEstandarLongF = [0]
113
114     mediaAritLongG = [0]
115     modaLongG = [0]
116     varianzaLongG = [0]
117     correlacionLongG = [0]
118     medianaLongG = [0]
119     desvEstandarLongG = [0]
120
121     mediaAritLongH = [0]
122     modaLongH = [0]
123     varianzaLongH = [0]
124     correlacionLongH = [0]
125     medianaLongH = [0]
126     desvEstandarLongH = [0]
127
128     mediaAritLongI = [0]
129     modaLongI = [0]
130     varianzaLongI = [0]
131     correlacionLongI = [0]
132     medianaLongI = [0]
133     desvEstandarLongI = [0]
134
135     mediaAritEscuela = [0]
136     modaEscuela = [0]
137     varianzaEscuela = [0]
138     correlacionEscuela = [0]
139     medianaEscuela = [0]
140     desvEstandarEscuela = [0]
141
142     mediaAritBar = [0]
143     modaBar = [0]
144     varianzaBar = [0]
145     correlacionBar = [0]
146     medianaBar = [0]
147     desvEstandarBar = [0]
148
149     mediaAritIglesia = [0]
150     modaIglesia = [0]
151     varianzaIglesia = [0]
152     correlacionIglesia = [0]
153     medianaIglesia = [0]
154     desvEstandarIglesia = [0]
155
156     mediaAritTienda = [0]
157     modaTienda = [0]
158     varianzaTienda = [0]
159     correlacionTienda = [0]
160     medianaTienda = [0]
161     desvEstandarTienda = [0]
162
163     mediaAritBanco = [0]
164     modaBanco = [0]
165     varianzaBanco = [0]
166     correlacionBanco = [0]
167     medianaBanco = [0]
168     desvEstandarBanco = [0]
```

```

169
170     mediaAritTipoDelito = [0]
171     modaTipoDelito = [0]
172     varianzaTipoDelito = [0]
173     correlacionTipoDelito = [0]
174     medianaTipoDelito = [0]
175     desvEstandarTipoDelito = [0]
176     matIndEnt = [] # almacena la representación de los individuos en
decimales
177     mat2IndEnt = [] # solo las variables generales, sin tomar en cuenta las
ubicaciones
178     x=[]
179     y=[]
180     xy=[]
181     x2=[]
182     sumax=0
183     sumay=0
184     sumax2=0
185     sumaxy=0
186     temp=0
187     m=0
188     b=0
189     ply=0
190     p2y=0
191     plx=0
192     p2x=0
193
194     nregistros=generaPoblacion.GeneraPoblacion.numRegistros
195     # VARIABLES DE LOS CÁLCULOS ESTADÍSTICOS
196
197     # Representación los individuos con un solo número
198     @classmethod
199     def representarIndEntero(cls):
200
201         # a=generaPoblacion.GeneraPoblacion.diaMesDec[0]
202         for y in range(0, generaPoblacion.GeneraPoblacion.numRegistros): # de
cero hasta el tamaño del vector
203             estadistica.matIndEnt.append(
204                 generaPoblacion.GeneraPoblacion.diaSemEnt[y] +
generaPoblacion.GeneraPoblacion.diaMesEnt[y] +
205                 generaPoblacion.GeneraPoblacion.mesEnt[y] +
generaPoblacion.GeneraPoblacion.horaEnt[y] +
206                 generaPoblacion.GeneraPoblacion.latDcortaEnt[y] +
generaPoblacion.GeneraPoblacion.latEcortaEnt[y] +
207                 generaPoblacion.GeneraPoblacion.latFcortaEnt[y] +
generaPoblacion.GeneraPoblacion.latGcortaEnt[y] +
208                 generaPoblacion.GeneraPoblacion.latHcortaEnt[y] +
generaPoblacion.GeneraPoblacion.latIcortaEnt[y] +
209                 generaPoblacion.GeneraPoblacion.longEcortaEnt[y] +
generaPoblacion.GeneraPoblacion.longFcortaEnt[y] +
210                 generaPoblacion.GeneraPoblacion.longGcortaEnt[y] +
generaPoblacion.GeneraPoblacion.longHcortaEnt[y] +
211                 generaPoblacion.GeneraPoblacion.longIcortaEnt[y] +
generaPoblacion.GeneraPoblacion.escuelaEnt[y] +
212                 generaPoblacion.GeneraPoblacion.barEnt[y] +
generaPoblacion.GeneraPoblacion.iglesiaEnt[y] +
213                 generaPoblacion.GeneraPoblacion.tiendaEnt[y] +
generaPoblacion.GeneraPoblacion.bancoEnt[y] +

```

```

214         generaPoblacion.GeneraPoblacion.tipodelitoEnt[y])
215
216         # solo para variables principales
217         estadistica.mat2IndEnt.append(
218             generaPoblacion.GeneraPoblacion.diaSemEnt[y] +
generaPoblacion.GeneraPoblacion.diaMesEnt[y] +
219             generaPoblacion.GeneraPoblacion.mesEnt[y] +
generaPoblacion.GeneraPoblacion.horaEnt[y] +
220             generaPoblacion.GeneraPoblacion.longIcortaEnt[y] +
generaPoblacion.GeneraPoblacion.escuelaEnt[y] +
221             generaPoblacion.GeneraPoblacion.barEnt[y] +
generaPoblacion.GeneraPoblacion.iglesiaEnt[y] +
222             generaPoblacion.GeneraPoblacion.tiendaEnt[y] +
generaPoblacion.GeneraPoblacion.bancoEnt[y] +
223             generaPoblacion.GeneraPoblacion.tipodelitoEnt[y])
224
225         estadistica.evPoblacion=estadistica.mat2IndEnt
226         # print("la suma es:",y,":",estadistica.matIndDec[y])
227
228         # MÉTODO PARA SACAR LA ESTADÍSTICA DE LOS DATOS ORIGINALES
229         @classmethod
230         def metodoGraficar(cls):
231             print("xxxdiasemana:",
len(generaPoblacion.GeneraPoblacion.escuelaEnt))
232
233             varianza = np.var(generaPoblacion.GeneraPoblacion.iglesiaEnt)
234             media = np.mean(generaPoblacion.GeneraPoblacion.diaSemEnt)
235             desvEstandar = np.std(generaPoblacion.GeneraPoblacion.diaSemEnt)
236
237             plt.figure ("Initial population")
238             plt.plot(estadistica.mat2IndEnt, "*", label="Individuals Population ",
239                 linestyle='dashed') # se puede cambiar el tipo de símbolo de
la gráfica
240
241             plt.plot(estadistica.mediaAritIndEnt)
242             plt.scatter(len(estadistica.matIndEnt) / 2,
estadistica.mediaAritIndEnt, label="Arithmetical means", color='r')
243             # anotacion en la grafica
244             #plt.annotate('Arithmetical
mean',xy=(100,100),xytext=(50,120),arrowprops=dict(facecolor= 'black'))
245             plt.legend(loc="lower left") # leyenda
246             plt.grid(True) # cuadrícula
247
248
249             plt.xlabel("Individuals") # etiqueta de x
250             plt.ylabel("Integer") # etiqueta de y
251             plt.title("Initial population") # título de grafico
252             plt.show()
253
254             plt.figure("Day with the highest crime rate")
255             plt.plot(estadistica.mediaAritDiaSem, "x") # media día de la semana
256             plt.plot(estadistica.modaDiaSem * 300, "<") # moda día semana
257             plt.ylabel("Weekdays")
258             plt.xlabel("Events")
259             plt.title("Day with the highest crime rate")
260             plt.plot(generaPoblacion.GeneraPoblacion.diaSemEnt, ".") # semana
261             plt.show()
262             # graficando las horas con más delitos

```

```

263     plt.figure("Time with the highest crime rate")
264     plt.ylabel("Time of the day")
265     plt.xlabel("Events")
266     plt.plot(generaPoblacion.GeneraPoblacion.horaEnt, ".")
267     plt.plot(estadistica.modaHora*300, "x",lineStyle="solid")
268     plt.title("Time with the highest crime rate")
269     plt.show()
270
271     plt.figure("Months of the year")
272     plt.ylabel("Months of the year")
273     plt.xlabel("Events")
274     plt.plot(generaPoblacion.GeneraPoblacion.mesEnt, ".")
275     plt.plot(estadistica.mediaAritMes*300,"*",lineStyle="solid")
276     plt.title("Month with the highest crime rate")
277     plt.show()
278
279     plt.ylabel("Days of the month")
280     plt.xlabel("Events")
281     plt.plot(generaPoblacion.GeneraPoblacion.diaMesEnt, ".")
282     plt.plot(estadistica.mediaAritDiaMes * 300, "*", lineStyle="solid")
283     plt.title ("Days of the month with the highest crime rate")
284     plt.show()
285
286
287     @classmethod # cálculo de media, moda, varianza, desv estandar,
coefCorrelacion
288     def metCalculos(cls):
289         # se calcula la media aritmética y se agrega a la matriz de datos de
medias
290         estadistica.mediaAritDiaSem[0] =
np.mean(generaPoblacion.GeneraPoblacion.diaSemEnt)
291         estadistica.mediaAritDiaMes[0] =
np.mean(generaPoblacion.GeneraPoblacion.diaMesEnt)
292         estadistica.mediaAritMes[0] =
np.mean(generaPoblacion.GeneraPoblacion.mesEnt)
293         estadistica.mediaAritHora[0] =
np.mean(generaPoblacion.GeneraPoblacion.horaEnt)
294         estadistica.mediaAritLatD[0] =
np.mean(generaPoblacion.GeneraPoblacion.latDcortaEnt)
295         estadistica.mediaAritLatE[0] =
np.mean(generaPoblacion.GeneraPoblacion.latEcortaEnt)
296         estadistica.mediaAritLatF[0] =
np.mean(generaPoblacion.GeneraPoblacion.latFcortaEnt)
297         estadistica.mediaAritLatG[0] =
np.mean(generaPoblacion.GeneraPoblacion.latGcortaEnt)
298         estadistica.mediaAritLatH[0] =
np.mean(generaPoblacion.GeneraPoblacion.latHcortaEnt)
299         estadistica.mediaAritLatI[0] =
np.mean(generaPoblacion.GeneraPoblacion.latIcortaEnt)
300         estadistica.mediaAritLongE[0] =
np.mean(generaPoblacion.GeneraPoblacion.longEcortaEnt)
301         estadistica.mediaAritLongF[0] =
np.mean(generaPoblacion.GeneraPoblacion.longFcortaEnt)
302         estadistica.mediaAritLongG[0] =
np.mean(generaPoblacion.GeneraPoblacion.longGcortaEnt)
303         estadistica.mediaAritLongH[0] =
np.mean(generaPoblacion.GeneraPoblacion.longHcortaEnt)

```

```

304     estadistica.mediaAritLongI[0] =
np.mean(generaPoblacion.GeneraPoblacion.longIcortaEnt)
305     estadistica.mediaAritEscuela[0] =
np.mean(generaPoblacion.GeneraPoblacion.escuelaEnt)
306     estadistica.mediaAritBar[0] =
np.mean(generaPoblacion.GeneraPoblacion.barEnt)
307     estadistica.mediaAritIglesia[0] =
np.mean(generaPoblacion.GeneraPoblacion.iglesiaEnt)
308     estadistica.mediaAritTienda[0] =
np.mean(generaPoblacion.GeneraPoblacion.tiendaEnt)
309     estadistica.mediaAritTipoDelito[0] =
np.mean(generaPoblacion.GeneraPoblacion.tipodelitoEnt)
310     estadistica.mediaAritIndEnt[0] = np.mean(estadistica.mat2IndEnt)
311     # DESVIACIÓN ESTANDAR
312     estadistica.desvEstandarDiaSem[0] =
np.std(generaPoblacion.GeneraPoblacion.diaSemEnt)
313     estadistica.desvEstandarDiaMes[0] =
np.std(generaPoblacion.GeneraPoblacion.diaMesEnt)
314     estadistica.desvEstandarMes[0] =
np.std(generaPoblacion.GeneraPoblacion.mesEnt)
315     estadistica.desvEstandarHora[0] =
np.std(generaPoblacion.GeneraPoblacion.horaEnt)
316     estadistica.desvEstandarLatD[0] =
np.std(generaPoblacion.GeneraPoblacion.latDcortaEnt)
317     estadistica.desvEstandarLatE[0] =
np.std(generaPoblacion.GeneraPoblacion.latEcortaEnt)
318     estadistica.desvEstandarLatF[0] =
np.std(generaPoblacion.GeneraPoblacion.latFcortaEnt)
319     estadistica.desvEstandarLatG[0] =
np.std(generaPoblacion.GeneraPoblacion.latGcortaEnt)
320     estadistica.desvEstandarLatH[0] =
np.std(generaPoblacion.GeneraPoblacion.lathcortaEnt)
321     estadistica.desvEstandarLatI[0] =
np.std(generaPoblacion.GeneraPoblacion.latIcortaEnt)
322     estadistica.desvEstandarLongE[0] =
np.std(generaPoblacion.GeneraPoblacion.longEcortaEnt)
323     estadistica.desvEstandarLongF[0] =
np.std(generaPoblacion.GeneraPoblacion.longFcortaEnt)
324     estadistica.desvEstandarLongG[0] =
np.std(generaPoblacion.GeneraPoblacion.longGcortaEnt)
325     estadistica.desvEstandarLongH[0] =
np.std(generaPoblacion.GeneraPoblacion.longHcortaEnt)
326     estadistica.desvEstandarLongI[0] =
np.std(generaPoblacion.GeneraPoblacion.longIcortaEnt)
327     estadistica.desvEstandarEscuela[0] =
np.std(generaPoblacion.GeneraPoblacion.escuelaEnt)
328     estadistica.desvEstandarBar[0] =
np.std(generaPoblacion.GeneraPoblacion.barEnt)
329     estadistica.desvEstandarIglesia[0] =
np.std(generaPoblacion.GeneraPoblacion.iglesiaEnt)
330     estadistica.desvEstandarTienda[0] =
np.std(generaPoblacion.GeneraPoblacion.tiendaEnt)
331     estadistica.desvEstandarTipoDelito[0] =
np.std(generaPoblacion.GeneraPoblacion.tipodelitoEnt)
332     estadistica.desEstandarIndEnt[0] = np.std(estadistica.mat2IndEnt)
333     # CALCULANDO LA VARIANZA
334     estadistica.varianzaDiaSem[0] =
np.var(generaPoblacion.GeneraPoblacion.diaSemEnt)

```

```

335     estadistica.varianzaDiaMes[0] =
np.var(generaPoblacion.GeneraPoblacion.diaMesEnt)
336     estadistica.varianzaMes[0] =
np.var(generaPoblacion.GeneraPoblacion.mesEnt)
337     estadistica.varianzaHora[0] =
np.var(generaPoblacion.GeneraPoblacion.horaEnt)
338     estadistica.varianzaLatD[0] =
np.var(generaPoblacion.GeneraPoblacion.latDcortaEnt)
339     estadistica.varianzaLatE[0] =
np.var(generaPoblacion.GeneraPoblacion.latEcortaEnt)
340     estadistica.varianzaLatF[0] =
np.var(generaPoblacion.GeneraPoblacion.latFcortaEnt)
341     estadistica.varianzaLatG[0] =
np.var(generaPoblacion.GeneraPoblacion.latGcortaEnt)
342     estadistica.varianzaLatH[0] =
np.var(generaPoblacion.GeneraPoblacion.latHcortaEnt)
343     estadistica.varianzaLatI[0] =
np.var(generaPoblacion.GeneraPoblacion.latIcortaEnt)
344     estadistica.varianzaLongE[0] =
np.var(generaPoblacion.GeneraPoblacion.longEcortaEnt)
345     estadistica.varianzaLongF[0] =
np.var(generaPoblacion.GeneraPoblacion.longFcortaEnt)
346     estadistica.varianzaLongG[0] =
np.var(generaPoblacion.GeneraPoblacion.longGcortaEnt)
347     estadistica.varianzaLongH[0] =
np.var(generaPoblacion.GeneraPoblacion.longHcortaEnt)
348     estadistica.varianzaLongI[0] =
np.var(generaPoblacion.GeneraPoblacion.longIcortaEnt)
349     estadistica.varianzaEscuela[0] =
np.var(generaPoblacion.GeneraPoblacion.escuelaEnt)
350     estadistica.varianzaBar[0] =
np.var(generaPoblacion.GeneraPoblacion.barEnt)
351     estadistica.varianzaIglesia[0] =
np.var(generaPoblacion.GeneraPoblacion.iglesiaEnt)
352     estadistica.varianzaTienda[0] =
np.var(generaPoblacion.GeneraPoblacion.tiendaEnt)
353     estadistica.varianzaTipoDelito[0] =
np.var(generaPoblacion.GeneraPoblacion.tipodelitoEnt)
354     estadistica.varianzaIndEnt[0] = np.var(estadistica.mat2IndEnt)
355
356     # calculando la correlación
357     estadistica.correlacionDiaSem[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.diaSemEnt)
358     estadistica.correlacionDiaMes[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.diaMesEnt)
359     estadistica.correlacionMes[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.mesEnt)
360     estadistica.correlacionHora[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.horaEnt)
361     estadistica.correlacionLatD[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.latDcortaEnt)
362     estadistica.correlacionLatE[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.latEcortaEnt)
363     estadistica.correlacionLatF[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.latFcortaEnt)
364     estadistica.correlacionLatG[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.latGcortaEnt)

```

```

365     estadistica.correlacionIatH[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.lathcortaEnt)
366     estadistica.correlacionLatI[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.latIcortaEnt)
367     estadistica.correlacionLongE[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.longEcortaEnt)
368     estadistica.correlacionLongF[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.longFcortaEnt)
369     estadistica.correlacionLongG[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.longGcortaEnt)
370     estadistica.correlacionLongH[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.longHcortaEnt)
371     estadistica.correlacionLongI[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.longIcortaEnt)
372     estadistica.correlacionEscuela[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.escuelaEnt)
373     estadistica.correlacionBar[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.barEnt)
374     estadistica.correlacionIglesia[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.iglesiaEnt)
375     estadistica.correlacionTienda[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.tiendaEnt)
376     estadistica.correlacionTipoDelito[0] =
np.corrcoef(generaPoblacion.GeneraPoblacion.tipodelitoEnt)
377     estadistica.correlacionIndEnt[0] = np.corrcoef(estadistica.mat2IndEnt)
378
379     # calculando la MODA, manejando posibles errores en caso que existan
dos o más modas
380     try:
381         estadistica.modaDiaSem[0] =
stat.mode(generaPoblacion.GeneraPoblacion.diaSemEnt)
382     except:
383         nTemp = rnd.randint(1, 7)
384         estadistica.modaDiaSem[0] = nTemp
385
386     # marca error en día mes, porque hay dos modas
387     try: # si existen dos modas, manejar el error
388         estadistica.modaDiaMes[0] =
stat.mode(generaPoblacion.GeneraPoblacion.diaMesEnt)
389     except:
390         nTemp = rnd.randint(0, 31) # crear un numero aleatorio
391         estadistica.modaDiaMes[0] = nTemp # asignado el numero a la
variable
392
393     try:
394         estadistica.modaMes[0] =
stat.mode(generaPoblacion.GeneraPoblacion.mesEnt)
395     except:
396         nTemp = rnd.randint(1, 12)
397         estadistica.modaMes[0] = nTemp
398
399     try:
400         estadistica.modaHora[0] =
stat.mode(generaPoblacion.GeneraPoblacion.horaEnt)
401     except:
402         nTemp = rnd.randint(1, 24)
403         estadistica.modaHora[0] = nTemp
404

```

```

405         # para latD, aleatorio 50 y 51, porque el digito puede ser 2 0 3, en
406         # caso contrario sale del espacio geográfico
407         try:
408             estadistica.modaLatD[0] =
stat.mode(generaPoblacion.GeneraPoblacion.latDcortaEnt) # entre 50 y 51
409         except:
410             nTemp == rnd.randint(50, 51)
411             estadistica.modaLatD[0] = nTemp
412         try:
413             estadistica.modaLatE[0] =
stat.mode(generaPoblacion.GeneraPoblacion.latEcortaEnt)
414         except:
415             nTemp = rnd.randint(48, 57)
416             estadistica.modaLatF[0] = nTemp
417         try:
418             estadistica.modaLatG[0] =
stat.mode(generaPoblacion.GeneraPoblacion.latGcortaEnt)
419         except:
420             nTemp = rnd.randint(48, 57)
421             estadistica.modaLatG[0] = nTemp
422         try:
423             estadistica.modaLatH[0] =
stat.mode(generaPoblacion.GeneraPoblacion.latHcortaEnt)
424         except:
425             nTemp = rnd.randint(48, 57)
426             estadistica.modaLatH[0] = nTemp
427         try:
428             estadistica.modaLatI[0] =
stat.mode(generaPoblacion.GeneraPoblacion.latIcortaEnt)
429         except:
430             nTemp = rnd.randint(48, 57)
431             estadistica.modaLatI[0] = nTemp
432         try:
433             estadistica.modaLongE[0] =
stat.mode(generaPoblacion.GeneraPoblacion.longEcortaEnt)
434         except:
435             nTemp = rnd.randint(48, 57)
436             estadistica.modaLongE[0] = nTemp
437         try:
438             estadistica.modaLongF[0] =
stat.mode(generaPoblacion.GeneraPoblacion.longFcortaEnt)
439         except:
440             nTemp = rnd.randint(48, 57)
441             estadistica.modaLongF[0] = nTemp
442         try:
443             estadistica.modaLongG[0] =
stat.mode(generaPoblacion.GeneraPoblacion.longGcortaEnt)
444         except:
445             nTemp = rnd.randint(48, 57)
446             estadistica.modaLongG[0] = nTemp
447         try:
448             estadistica.modaLongH[0] =
stat.mode(generaPoblacion.GeneraPoblacion.longHcortaEnt)
449         except:
450             nTemp = rnd.randint(48, 57)
451             estadistica.modaLongH[0] = nTemp
452         try:

```

```

453             estadistica.modaLongI[0] =
stat.mode(generaPoblacion.GeneraPoblacion.longIcortaEnt)
454         except:
455             nTemp = rnd.randint(48, 57)
456             estadistica.modaLongI[0] = nTemp
457         try:
458             estadistica.modaEscuela[0] =
stat.mode(generaPoblacion.GeneraPoblacion.escuelaEnt)
459         except:
460             nTemp = rnd.randint(0, 10)
461             estadistica.modaEscuela[0] = nTemp
462         try:
463             estadistica.modaBar[0] =
stat.mode(generaPoblacion.GeneraPoblacion.barEnt)
464         except:
465             nTemp = rnd.randint(0, 10)
466             estadistica.modaBar[0] = nTemp
467
468         try:
469             estadistica.modaIglesia[0] =
stat.mode(generaPoblacion.GeneraPoblacion.iglesiaEnt)
470         except:
471             nTemp = rnd.randint(0, 10)
472             estadistica.modaIglesia[0] = nTemp
473         try:
474             estadistica.modaTienda[0] =
stat.mode(generaPoblacion.GeneraPoblacion.tiendaEnt)
475         except:
476             nTemp = rnd.randint(0, 10)
477             estadistica.modaTienda[0] = nTemp
478         try:
479             estadistica.modaTipoDelito[0] =
stat.mode(generaPoblacion.GeneraPoblacion.tipodelitoEnt)
480         except:
481             nTemp = rnd.randint(0, 1)
482             estadistica.modaTipoDelito[0] = nTemp
483         try:
484             estadistica.modaIndEnt[0] = stat.mode(estadistica.mat2IndEnt)
485         except:
486             nTemp = rnd.randint(65, 120)
487             estadistica.modaIndEnt[0] = nTemp
488
489         print("imprimeindo las variables estadisticas deia Semana")
490         print("Media aritmetica", estadistica.mediaAritDiaSem)
491         print("Desv Estandar", estadistica.desvEstandarDiaSem)
492         print("varianza", estadistica.varianzaDiaSem)
493         print("correlacion", estadistica.correlacionDiaSem)
494         print("Moda", estadistica.modaDiaSem)
495
496
497     @classmethod # evolución de la población
498     def evolucionPoblacion(cls):
499
500         pobTemp=[0]
501         print("poblacion original",estadistica.evPoblacion)
502         for i in range (0,300):
503             estadistica.poblacTemp.append([])
504             nRnd=rnd.randint(0,150)

```

```

505         estadistica.pobTemp.append(nRnd)
506     print("poblacion Temporal",estadistica.pobTemp)
507
508
509     #graficando la población siguiente, primer gráfica
510     plt.figure("Last population")
511     plt.plot(estadistica.pobTemp, "*", label="Individuals Population",
512             linestyle='dashed') # se puede cambiar el tipo de símbolo de
la gráfica
513     plt.plot(estadistica.mediaAritIndEnt * 300)
514     plt.scatter(len(estadistica.matIndEnt)/2 ,
estadistica.mediaAritIndEnt, label="Arithmetic means", color='r')
515     plt.plot(estadistica.modaIndEnt* 300)
516     plt.scatter(len(estadistica.matIndEnt) , estadistica.mediaAritIndEnt,
label="Arithmetic means", color='black')
517     plt.legend(loc="lower left") # leyenda
518     plt.grid(False) # cuadrícula
519
520     # plt.annotate("ssss")
521     plt.xlabel("Individuals") # etiqueta de x
522     plt.ylabel("Integer events") # etiqueta de y
523     plt.title("Last Population ") # titulo de gráfico
524     plt.show()
525
526
527     @classmethod # regresión lineal entre los días de la semana y las horas
de los eventos
528     def regreDiaHora(cls):
529         # X sera para Horas
530         # Y dias de la semana
531         for i in range (0,estadistica.nregistros):
532             #calculado X cuadrado
533
estadistica.x2.append(generaPoblacion.GeneraPoblacion.horaEnt[i]*generaPoblacion.Ge
neraPoblacion.horaEnt[i])
534
estadistica.xy.append(generaPoblacion.GeneraPoblacion.horaEnt[i]*generaPoblacion.Ge
neraPoblacion.diaSemEnt[i])
535         estadistica.temp=generaPoblacion.GeneraPoblacion.horaEnt[i]
536         #estadistica.sumax=estadistica.temp+estadistica.sumax
537
estadistica.sumax=float(generaPoblacion.GeneraPoblacion.horaEnt[i]+estadistica.suma
x)
538
estadistica.sumay=float(generaPoblacion.GeneraPoblacion.diaSemEnt[i]+estadistica.su
may)
539         estadistica.sumaxy=float(estadistica.xy[i]+estadistica.sumaxy)
540         estadistica.sumax2=float(estadistica.x2[i]+estadistica.sumax2)
541
542     print ("suma x:",estadistica.sumax)
543     print ("suma y:",estadistica.sumay)
544     print("suma x2:", estadistica.sumax2)
545     print("suma xy:", estadistica.sumaxy)
546     #REgresion lineneal
547     #Sustituyendo valores en la formula
548     #CALCULADO A M

```

```

549         estadistica.m=((float(estadistica.nregistros)*(estadistica.sumaxy)-
((estadistica.sumax)*(estadistica.sumay)))/((estadistica.nregistros*estadistica.sum
ax2)-(estadistica.sumax*estadistica.sumax))
550         #CALCULANDO A B
551
552         estadistica.b = (estadistica.mediaAritDiaSem) - ((estadistica.m) *
(estadistica.mediaAritHora[0]))
553
554         estadistica.plx=0
555         estadistica.ply=estadistica.b
556         estadistica.p2x=max(generaPoblacion.GeneraPoblacion.horaEnt)
557         estadistica.p2y=(estadistica.m*24)+estadistica.b
558
559         print("Valor de m formula=", estadistica.m)
560         print("Valor de B formula=", estadistica.b)
561         print ("promedio horaSem",estadistica.mediaAritHora)
562         print("promedio DiaSem",estadistica.mediaAritDiaSem)
563         print("plx",estadistica.plx)
564         print("ply", estadistica.ply)
565         print("p2x", estadistica.p2x)
566         print("p2y", estadistica.p2y)
567
568         @classmethod # regresión lineal entre los días de la semana y las horas
de los eventos
569         def plotRegreDiaHora(cls):
570             p1=[estadistica.plx,estadistica.ply]
571             p2=[estadistica.p2x,estadistica.p2y]
572             plt.figure("Días con mayor índice delictivo")
573
plt.plot(generaPoblacion.GeneraPoblacion.horaEnt,generaPoblacion.GeneraPoblacion.di
aSemEnt,"r.",linestyle='dotted',label="Días de la semana")
574             plt.xlabel("Horas del dia")
575             plt.ylabel("Días de la semana")
576             plt.title("Días con mayor incidencia delictiva")
577             plt.plot((estadistica.plx, estadistica.p2x), (estadistica.ply,
estadistica.p2y), "b+", linestyle='solid',label ="Días y horas")
578             plt.legend(loc='upper left')
579             plt.show()

```

## A.1.4 Mutación

```
1
2 import creaMatrizCls as creaMatriz
3 import Padres as Padres
4 import GeneraPoblacion as GeneraPoblacion
5 import random as rnd
6 import numpy as np
7
8 class Mutacion ():
9
10     Temporal=[]
11     listaAleatorios=[]
12     nAleatorio=0 #almacenará un aleatorio entre 0 y el numero e registros
13     poblacionMutada=[] #guardara las poblaciones mutadas
14
15     #metodo para la mutación, de forma aleatoria en un punto
16     @classmethod
17     def mutar (cls):
18         Padres.Padres.metMostrarPadresCromosomas() #invocando el método de la
19         #Padres.Padres.atribClase()
20         numCromosomas=creaMatriz.creaMatrizCls.nReg #cantidad de cromosomas o
21         #registro
22         longCromosoma=len(GeneraPoblacion.GeneraPoblacion.indOptimoCorto[0])
23         #largo del cromosoma
24         i=0
25         for i in range (0,numCromosomas): # desde 0 hasta el numero de
26         #registros, cromosomas o casos, elementos de la población 300
27         Mutacion.nAleatorio = rnd.randint(0,longCromosoma - 1) #
28         #aleatorio entre 0 y numero de elementos del cromosoma
29         Temp=GeneraPoblacion.GeneraPoblacion.poblacionInicial[i] # paso la
30         #cadena a otra variable
31         Temp=list(Temp) #convierto en lista la cadena de caracteres
32         if (Temp[Mutacion.nAleatorio]=='0'):# si el valor del cromosoma en
33         #la posición nAleatorio =0, cambiar por 1, en caso contrario por 0
34         Temp[Mutacion.nAleatorio]=1 # se cambia 0 por 1
35         Mutacion.poblacionMutada.append(Temp)
36         else:
37         Temp[Mutacion.nAleatorio] = 0 # se hace la mutaciones cambia 1
38         #por 0
39         Mutacion.poblacionMutada.append(Temp)
40
41         print("clase mutar: elemento de la poblacion
42         inicial",GeneraPoblacion.GeneraPoblacion.poblacionInicial[300])#original
43         print("clase mutar elemento MUTADO",Mutacion.poblacionMutada[0])
44         #mutado
45
46     @classmethod
47     def crearAleatorio (cls):
48         creaMatriz.creaMatrizCls.crearMatrizMetod() # se ejecuta el método
49         #para poder darle valor a la variable
50         #las siguientes líneas generan un ciclo aleatorio que se detiene al
51         #cumplirse una restricción
52         i=0 #inicializando la variable i
53         while (i>=0):
```

```
43         Mutacion.nAleatorio = rnd.randint(0,
creaMatriz.creaMatrizCls.nReg) # crear un aleatorio entre 0 y el número de
registros que existen en la matriz inicial TODOS LOS DATOS
44
45         Mutacion.listaAleatorios.append(Mutacion.nAleatorio)
46
47         if (i==300): # condición de parada
48             break # se detiene el bucle cuando se llega a la restricción
49             i+=1 #incrementa a i
50
51     for i in range (0,300):
52         Mutacion.crearAleatorio()
53         print ("Repetidos",Mutacion.listaAleatorios.sort())
54         print (Mutacion.listaAleatorios)
55         Mutacion.mutar()
```

## A.2 Participaciones y publicaciones

### A.2.1 Correo de aceptación de artículo para publicación en revista

Correo: Pablo Vera - Outlook

#### MICAI 2021 acceptance notification, paper 113 -- clarification

MICAI 2021 <micai2021@easychair.org>

Lun 30/08/2021 01:40 AM

Para: Pablo Vera <pablillomx83@hotmail.com>

Dear Professor Pablo Vera,

You have previously received an acceptance notification on your paper

113: "Implementation of an evolutionary algorithm in the creation of criminal geographic profiles"

accepted for a short presentation at MICAI 2021 and for publication in a journal issue derived from the conference.

In particular, please note that your paper will NOT be included in the Springer LNAI volume of the conference proceedings. Instead, it will be included in an issue of the journal.

Please also note that, by the policy of the journal, the journal will NOT explicitly refer to the MICAI conference.

We will contact you separately on the modality of your presentation at the conference.

Sincerely,  
MICAI 2021 Program Chairs

## MICAI 2021 acceptance notification, paper 113 (R)

MICAI 2021 <micai2021@easychair.org>

Lun 30/08/2021 12:56 AM

Para: Pablo Vera <pablillomx83@hotmail.com>

Dear Professor Pablo Vera,

We are happy to notify you that your paper

113: "Implementation of an evolutionary algorithm in the creation of criminal geographic profiles"

has been ACCEPTED for a short presentation at MICAI 2021, 20th Mexican International Conference on Artificial Intelligence, [www.MICAI.org](http://www.MICAI.org), to be held on October 25-30 in Mexico City, Mexico, and for publication in a journal volume derived from the conference.

Namely, a number of selected papers, yours among them, are invited to a special issue of the journal Research in Computing Science, [www.rcs.cic.ipn.mx](http://www.rcs.cic.ipn.mx).

The invitation is final; the paper will not be subject to any additional review or selection process and no additional fees (apart from the MICAI participation fee) will apply.

CAMERA-READY VERSION:

Please find below the review results. Please carefully take them into account when preparing your camera-ready version.

We will contact you separately on the details of your presentation, possible travel (optional), and camera-ready instructions.

The fee payment deadline is August 31 (please contact us if you need more time).

The camera-ready deadline will be later; we will contact you separately on this.

Sincerely,  
MICAI 2021 Program Chairs

**A.2.3 Presentación de artículo “Implementation of an evolutionary algorithm in the creation of criminal geographic profiles”, en MICA, octubre 2021.**



The Mexican Society for Artificial Intelligence (SMIA)  
and the Centro de Investigación en Computación del Instituto Politécnico Nacional



Award this certificate to

**Pablo Vera, Cristina Juárez, Sergio Ruiz and Rogelio Reyes**

for presentation of the paper entitled

**Implementation of an evolutionary algorithm in the creation of criminal geographic profiles**

at the 20<sup>th</sup> Mexican International Conference on Artificial Intelligence, MICA 2021.  
Mexico City, Mexico, October 25 - 30, 2021.



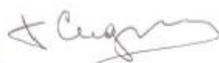
**Dr. Félix Castro Espinoza**  
SMIA President



**Dr. Ildar Batyrshin**  
Program Chair



**Dr. Alexander Gelbukh**  
Program Chair



**Dr. Grigori Sidorov**  
Program Chair



**A.2.2 Participación en el 22 seminario de investigación en la Universidad Autónoma de Aguascalientes, julio 2021.**

 UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

La Universidad Autónoma de Aguascalientes a través de la Dirección General de Investigación y Posgrado otorga la presente:

 DIRECCIÓN GENERAL DE INVESTIGACIÓN Y POSGRADO

# CONSTANCIA

a:

**Pablo Vera, Cristina Juárez Landín, Rogelio Reyes Reyes  
José Sergio Ruiz Castilla**

Por su participación con la **ponencia:**  
**Algoritmo evolutivo para la perfilación geográfica criminal**

En la mesa de: **Ingenierías y Tecnologías**  
Disciplina: **Inteligencia Artificial**

En el marco del

 SEMINARIO DE INVESTIGACIÓN 2021

*Se lumen proferre*  
Aguascalientes, Ags., Julio de 2021

  
Dr. en C. Francisco Javier Avelar González  
*Rector*

  
Mtra. en C.O. Elizabeth Casillas Casillas  
*Directora General de Investigación y Posgrado*

**A.2.3 Entrevista en TV Azteca, ADN40 “Perfilación sobre delitos cibernéticos”, mayo 2020.**



Enlace de entrevista

<https://fb.watch/982uvsfJBk/>

<https://www.facebook.com/OficialAMCCO/videos/194811984918875/>

## A.2.1 Publicación de libro “Inteligencia artificial como herramienta en la prevención de delitos”

P. Vera. Inteligencia artificial como herramienta en la prevención de delitos, ISBN 978-9962-721-23-9, Editorial: Seguridad y Defensa Corp. S.A., diciembre 1, 2019. Panamá.

Editorial: Seguridad y Defensa Corp. S.A.  
Panamá.

Materia: Inteligencia artificial

Público objetivo: General

Publicado:2019-12-01

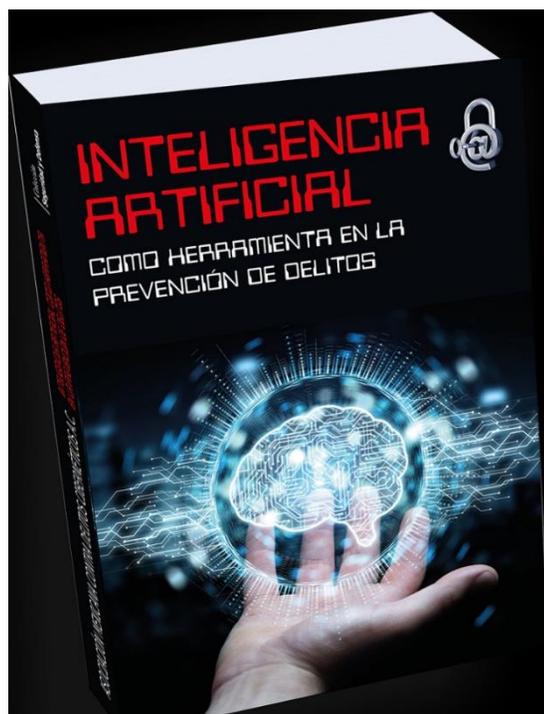
Número de edición:1

Número de páginas:96

Tamaño 14x20cm.

Encuadernación: Tapa blanda o rústica

Soporte: Impreso



### Reseña

La evolución natural de las especies ha motivado a los investigadores a desarrollar técnicas basadas en la teoría evolutiva, las que pueden ser aplicadas en la solución de diferentes tipos de problemas y como herramientas para la prevención de delitos. Los diferentes tipos que ocurren hoy; pueden mostrar características, rasgos o patrones similares que pueden ser estudiados y modelados bajo los principios de la inteligencia artificial. Se mencionan y describen diferentes técnicas de Inteligencia artificial y se hace una comparación de estas, con el objetivo de visualizar el alcance que estas pueden tener el momento de usarse como herramientas en la prevención del delito, problemas de optimización, predicción de datos, entre otras.

### Link

<http://isbn.binal.ac.pa/risbn53/catalogo.php?mode=detalle&nt=19825>

## Referencias

Aletras N, Tsarapatsanis D, Preoțiu-Pietro D, Lampos V. (2016). Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. PeerJ Computer Science 2:e93 <https://doi.org/10.7717/peerj-cs.93>

Barceló, M. (2008). Una historia de la Informática. Barcelona: Universidad Abierta de Cataluña. ISBN: 9788497887090

Brownlee, J. (2012). Clever algorithms: nature-inspired programming recipes. Recuperado de <http://www.cleveralgorithms.com/>. ISBN: 1446785068 9781446785065

CAF (Banco de Desarrollo de América Latina). (2014). Por una América Latina más segura: Una nueva perspectiva para prevenir y controlar el delito. Bogotá: CAF. 2014. ISBN 980-6810-01-5.

CESEDEN. (2018). La inteligencia artificial aplicada a la defensa. España. IEEE.

Coello, C. (2019). Computación Evolutiva. México: AMEXCOMP.

Del Alcázar L. D., Silva E., Fernández X. O. (2014). Sistema inteligente para perfilar la deserción en estudiantes universitarios de carreras técnicas. Cuadernos del Contrato Social por la Educación. No. 10. Contrato Social por la Educación. Ecuador. Recuperado de <https://uide.edu.ec>.

Engelbrecht, A. P. (2007). Computational intelligence: an introduction. Sudáfrica: John Wiley. Segunda Edición.

García Pablos A. (1992). La prevención del delito en un estado social y democrático de derecho. Estudios Penales y Criminológicos, vol. XV. Cursos e Congresos nº 71 Servizo de Publicacións da Universidade de Santiago de Compostela. ISBN 84-7191-866-8, pp. 80-97.

Garrido, V., López, P. (2006). El rastro del asesino. El perfil psicológico de los criminales en la investigación policial. Barcelona, España: Ariel. Primera edición. ISBN:978-84-344-4492-8.

Google. (2021). Valle de Chaco, Estado de México. Recuperado el 20 de octubre de 2021, de <https://www.google.com/maps/>

Jiménez S. J. (2018). El perfil geográfico criminal. Obtenido de <https://www.psicologia-online.com>. 1 de abril 2018.

Jones, M. T. (2008). Artificial intelligence: A systems approach. Hingham, Massachusetts: Infinity Science press llc. Recuperado de <https://www.researchgate.net>. ISBN:978-0-9778582-3-1.

Lacave C., Molina A. I., Redondo M. A., Ortega M. (2016), Redes bayesianas para identificar perfiles de estudiante. Aplicación al estudio del abandono de las titulaciones de Informática en la Universidad de Castilla-La Mancha. Actas de las XXII Jenui. Almería. ISBN: 978-84-16642-30-4. Páginas: 85-92

Lake., B. M., Ullman., T. D., Tenenbaum., J. B., & Gershman., S. J. (2016). Building Machines That Learn and Think Like People. Arxive.

Leyderecho. (2019). Clases de delitos. Consulta 10 octubre, 2019. Disponible en: [mexico.leyderecho.org](http://mexico.leyderecho.org).

OMS. (2002). Informe mundial sobre la violencia y la salud: Resumen. Washington DC. ISBN 92-4 154-562-3.

PredPol (2018). Predictive Policing: orientación sobre dónde y cuándo patrullar. Los Ángeles, California. Recuperado de <http://www.predpol.com>.

Poole, D. L. and Mackworth, A. K. (2010). *Artificial Intelligence: foundations of computational agents*. Vancouver, Canadá: Cambridge University Press.

Redondo Illescas, Santiago, & Pueyo, Antonio Andrés (2007). La psicología de la delincuencia. *Papeles del Psicólogo*, 28(3). Consulta 20 de octubre de 2019. ISSN: 0214-7823. Disponible en: <https://www.redalyc.org/articulo.oa?id=778/77828302>

Rivas F. J., R. e. (2017). Reporte sobre delitos de alto impacto. Observatorio Nacional Ciudadano. México. Recuperado de <http://onc.org.mx>.

Rivero Pérez, Jorge Luis. (2014). Técnicas de aprendizaje automático para la detección de intrusos en redes de computadoras. *Revista Cubana de Ciencias Informáticas*, 8(4), 52-73. Consulta 20 de octubre de 2019, disponible en: [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2227-18992014000400003&lng=es&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992014000400003&lng=es&tlng=es)

Rouhiainen, L., (2018). *Inteligencia Artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Barcelona, España: Editorial Planeta. ISBN: 978-84-17568-08-5

Rutkowski, L. (2005). *Computational intelligence: methods and techniques*. Częstochowa, slaskie, Poland: Springer.

Searle, J. R., (1980). Minds, Brains, and Program, *Behavioral and Brain Science*, Cambridge (UK), pp.417-457.

Smolinski, T. G., Milanova, M. G., and Hassanien, A.-E. (2008). *Computational Intelligence in Biomedicine and Bioinformatics: Current trends and applications*, Chennai, India: Springer.

Soto, Carles. (2016). La prevención situacional: bases teóricas de fundamento criminológico. *Iter Criminis. Revista de Ciencias Penales*. 15. 85-110.

SSP (2018). Secretaria de Seguridad Pública. México. Recuperado de <https://www.gob.mx/>.

Summers L., Rossmo K. (2015). El perfil geográfico en la investigación criminal. *International E-journal of Criminal Sciences*. Artículo 3. No. 9. Recuperado de <http://www.ehu.eus/ojs/index.php/inecs>. ISSN: 1988-7949

Whitby, B. (2009). *Artificial intelligence*. New York, EUA: Rosen Publishing.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, Agosto 31, 1955. *AI Magazine*, 27(4), 12. <https://doi.org/10.1609/aimag.v27i4.1904>

Williams, Naomi (2014). 100 promising practices on safer cities. *The Global Network on Safer Cities*.