UNIDAD ACADÉMICA:

OFICINA DE POSGRADOS

TEMA:

IMPLEMENTACIÓN DE ALGORITMOS GENÉTICOS PARA EL ESTUDIO DE AFLUENCIA Y DESCONGESTIONAMIENTO DE USUARIOS EN EMPRESAS PÚBLICAS EN LA CIUDAD DE RIOBAMBA

Proyecto de Investigación y Desarrollo previo a la obtención del título de

Magister en Gerencia Informática

Línea de Investigación, Innovación y Desarrollo principal:

Inteligencia Artificial, Robótica, Domótica y/o Sistemas Expertos

Clasificación técnica del trabajo:

Desarrollo

Autor:

Víctor Manuel Oquendo Coronado

Director:

José Marcelo Balseca Manzano, Mg.

Ambato – Ecuador

Marzo 2019

Implementación de algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba

Informe de Trabajo de Titulación

Presentado ante la

Pontifica Universidad Católica del Ecuador

Sede Ambato

por

Víctor Manuel Oquendo Coronado

En cumplimiento parcial

de los requisitos para el Grado de

Magister en Gerencia Informática

Oficina de Posgrados

Marzo 2019

Implementación de algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba

Aprobado por:

María Fernanda San Lucas Solórzano, Mg
Presidente del Comité Calificador
Coordinadora de la oficina de posgrados

Darío Javier Robayo Jácome, Ing. Mg.
Miembro Calificador

José Marcelo Balseca Manzano, Mg.
Miembro Calificador
Director de Proyecto

Hugo Rogelio Altamirano Villarroel, Dr.
Secretario General

Galo Mauricio López Sevilla, Ing. Mg.
Miembro Calificador

Fecha de aprobación: marzo 2019

Ficha Técnica

Programa: Magister en Gerencia Informática

Tema: Implementación de algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba

Tipo de Trabajo: Proyecto de Investigación y Desarrollo

Clasificación técnica del Trabajo: Desarrollo

Autor: Víctor Manuel Oquendo Coronado

Director: José Marcelo Balseca Manzano, Mg.

Líneas de Investigación, Innovación y Desarrollo

Principal: Inteligencia Artificial, Robótica, Domótica y/o Sistemas Expertos

Resumen Ejecutivo

El problema radica en la desproporción a las oficinas municipales existentes de la ciudad de Riobamba, por parte de la ciudadanía, generando molestias a los usuarios, y afectando a las recaudaciones municipales en los meses de congestión. Para el estudio de planificación de nuevos puntos de atención a los usuarios en la ciudad, serán asistidos por la aplicación software aplicando algoritmos genéticos para determinar las posibles ubicaciones de los nuevos puntos de atención a los usuarios en el mapa urbano de Riobamba, además se visualiza la posible distribución de congestión en tiempo real a los nuevos puntos de atención. El algoritmo genético utilizado aplica las etapas de un algoritmo genético simple las cuales son: evaluación de la población, selección, cruces o recombinación y mutación, para encontrar la solución óptima. Para la resolución del problema, se creó vectores asociativos por parroquias de la ciudad de Riobamba, cada vector contiene si existe o no un punto de atención usando 0 o 1, y la posición referenciada a una matriz con fila y columna del mapa de Riobamba el cual fue instanciado mediante Google maps.

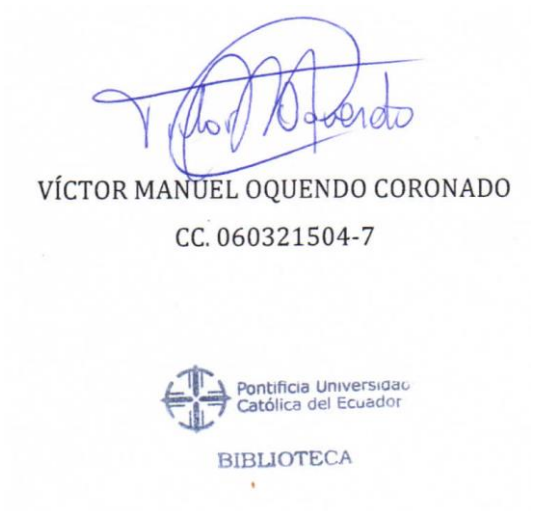
DECLARACIÓN Y AUTORIZACIÓN

Yo: OQUENDO CORONADO VÍCTOR MANUEL, con CC. 060321504-7, autora del trabajo de graduación intitulado: "**:** IMPLEMENTACIÓN DE ALGORITMOS GENÉTICOS PARA EL ESTUDIO DE AFLUENCIA Y DESCONGESTIONAMIENTO DE USUARIOS EN EMPRESAS PÚBLICAS EN LA CIUDAD DE RIOBAMBA", previa a la obtención del título profesional de MAGISTER EN GERENCIA INFORMÁTICA, en la Oficina de Postgrados.

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE Ambato, el referido trabajo de graduación, respetando las políticas de propiedad intelectual de Universidad

Ambato, marzo 2019



Dedicado a mis padres

Víctor Oquendo Ruiz

Lourdes Coronado Villacrés

Reconocimientos

Agradezco el apoyo incondicional brindado por mis padres y por todas las personas que, de una u otra manera, apoyaron este proyecto.

Resumen

El objetivo principal de este proyecto de investigación y desarrollo es implementar algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba. El problema radica en la desproporción de las oficinas municipales existentes de la ciudad de Riobamba, por parte de la ciudadanía, generando molestias a los usuarios, y afectando a las recaudaciones municipales en los meses de congestión. Para el estudio de planificación de nuevos puntos de atención a los usuarios en la ciudad, estos usuarios serán asistidos por la aplicación software aplicando algoritmos genéticos para determinar las posibles ubicaciones de los nuevos puntos de atención a los usuarios en el mapa urbano de Riobamba, además se visualiza la posible distribución de congestión en tiempo real de los nuevos puntos de atención. El algoritmo genético utilizado aplica las etapas de un algoritmo genético simple las cuales son: evaluación de la población, selección, cruces o recombinación y mutación, para encontrar la solución óptima. Para la resolución del problema, se creó vectores asociativos por parroquias de la ciudad de Riobamba, cada vector contiene si existe o no un punto de atención usando 0 o 1, y la posición referenciada a una matriz con fila y columna del mapa de Riobamba el cual fue instanciado mediante Google maps.

Palabras claves: ubicación, ubicaciones, algoritmo, genética, algoritmo genético, vector, vector asociativo.

Abstract

The main aim of this research and development project is to implement genetic algorithms for the study of affluence and the decongestion of users in public companies in the city of Riobamba. The problem lies in the disproportion of existing municipal offices in the city of Riobamba, by citizens, which causes inconvenience for the users and affects the city's tax collections during the months with a lot of congestion. For the study of a plan for new points of service for city users, the users will be assisted by the software application by applying genetic algorithms to determine the possible locations of new points of service for users on the urban map of Riobamba, as well as the possible distribution of congestion of the new points of service in real time. The genetic algorithm used applies the stages of a simple genetic algorithm - population evaluation, selection, crosses or recombination and mutation - to find the optimal solution. In order to solve the problem, associative vectors were created by regions of the city of Riobamba. Each vector indicates whether or not there is a point of service using 0 or 1, and the position referenced to a matrix with rows and columns of the map of Riobamba, which was instantiated using Google maps.

Key words: location, locations, algorithm, genetics, genetic algorithm, vector, associative vector.

Tabla de Contenidos

Lista de Tablas

Lista de Figuras

CAPÍTULO 1

Introducción

Los avances tecnológicos en temas de algoritmos de búsqueda, apoyados en la genética y su proceso de transmisión y supervivencia de los genes más aptos de generación en generación, denominados Algoritmos Genéticos (AGs), el cual es uno de los máximos exponentes de la computación evolutiva, permite realizar simulaciones de comportamientos de situaciones mediante funciones objetivo, obteniendo resultados para toma de decisiones posteriores.

La empresa OmbuGames Cia. Ltda., propone desarrollar una aplicación software y brindar el servicio de simulación, en el mapa de Riobamba, de la afluencia de usuarios a las oficinas municipales, para realizar sus trámites. Visualizando en tiempo real la congestión de usuarios que esto genera. Y aplicando algoritmos genéticos para determinar las posibles sugerencias de ubicación de nuevos puntos de atención, y observar la afluencia de los usuarios a dichos puntos.

El presente Informe Final del Trabajo de Titulación para Postgrados de la PUCESA (Pontificia Universidad Católica del Ecuador Sede Ambato), expone el desarrollo de la aplicación software aplicando algoritmos genéticos y usando un motor de gráficos Unity 3D, para que la empresa OmbuGames Cia. Ltda., brinde el servicio a personas interesadas, de la simulación de descongestionamiento de usuarios en las oficinas municipales.

1.1. Presentación del Trabajo

Se implementará algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios de empresas públicas en la ciudad de Riobamba, dicho servicio proveerá la empresa OmbuGames Cia. Ltda. en una aplicación software exponiendo los comportamientos actuales y futuros de la afluencia de la ciudadanía a las oficinas municipales. Mediante la aplicación software, los responsables de planificación y crecimiento, obtienen datos simulados de la congestión ciudadana, tomando la mejor decisión de ubicación de las nuevas oficinas públicas.

La aplicación software, se la utilizará tanto en un computador de escritorio, como en un computador portátil y en dispositivos móviles, brindando movilidad al personal de planificación de realizar sus simulaciones,

análisis y presentaciones no únicamente en oficina. Al ser una aplicación interactiva y amigable con el usuario, con tan solo un clic en el mapa de la ciudad de Riobamba se simulará la creación de una nueva oficina municipal, y se observará visualmente los íconos que simulan a los ciudadanos, acudiendo a esta nueva dependencia, descongestionando las ya existentes.

1.2. Descripción del Documento

En el presente apartado, se expone una visión general del Informe Final del Trabajo de Titulación para Postgrados, con una breve explicación del contenido de cada capítulo.

Capítulo 1.- introduce la aplicación de un algoritmo genético al proyecto de desarrollo.

Capítulo 2.- expone la propuesta de trabajo, con su información técnica básica, el problema con su descripción, juntamente con las preguntas básicas. También se formula la meta a alcanzar, y su respectivo objetivo general y específicos. Además, se presenta la delimitación funcional para determinar el alcance del proyecto de investigación y desarrollo.

Capítulo 3.- hace mención del marco teórico, abordando las definiciones y conceptos de relevancia sobre algoritmos genéticos, metodologías ágiles, unity 3D y el manejo del api de Google maps, así como se referencia a las investigaciones que ya se han realizado acerca de la aplicación de algoritmos genéticos alrededor del mundo.

Capítulo 4.- presenta la metodología de trabajo, siendo usada Scrum una metodología ágil para el desarrollo de la aplicación software, con la descripción de sus fases.

Capítulo 5.- muestra los resultados del desarrollo de la aplicación software, en forma de simulado en tiempo real mediante unity 3D, de la aplicación de los algoritmos genéticos.

Capítulo 6.- presenta las conclusiones, recomendaciones derivadas al desarrollo de la aplicación software.

CAPÍTULO 2

Planteamiento de la Propuesta de Trabajo

2.1. Información técnica básica

Tema: Implementación de algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba

Tipo de Trabajo: Proyecto de Investigación y Desarrollo

Clasificación técnica del Trabajo: Desarrollo

Líneas de Investigación, Innovación y Desarrollo

Principal: Inteligencia Artificial, Robótica, Domótica y/o Sistemas Expertos

2.2. Descripción del problema

El problema se enfoca en la conglomeración desproporcionada de usuarios debido a la concurrencia de la ciudadanía a las oficinas municipales existentes en la ciudad de Riobamba, para realizar sus trámites pertinentes. Actualmente esto genera congestión, generando molestias al usuario por el tiempo de espera, y muchas veces el ciudadano desiste de su trámite para realizarlo otro día. En consecuencia, el número de recaudaciones se ve afectado, para la empresa municipal de la ciudad de Riobamba.

Es necesario realizar los estudios pertinentes de ubicación y disponibilidad de terrenos o arriendos disponibles en el mapa de la ciudad de Riobamba, para la planificación de nuevas oficinas para agilitar los tramites y recaudaciones ciudadanas, mediante la implementación de algoritmos genéticos en la aplicación software usando unity3d que es un motor de gráficos en tres y dos dimensiones, el cual permite simular y visualizar de forma interactiva los resultados de ubicar oficinas en puntos estratégicos en el mapa urbano de la ciudad de Riobamba, observando los nuevos comportamientos de afluencia de los usuarios a las oficinas municipales de la ciudad de Riobamba. Brindando así una perspectiva en tiempo real, al departamento de

planificación, para la toma de decisiones de las posibles mejores ubicaciones para las oficinas públicas proyectadas, en la búsqueda constante de la calidad en el servicio a la ciudadanía.

2.3. Preguntas básicas

¿Cómo aparece el problema que se pretende solucionar?

La ilustre Municipalidad de Riobamba al no contar con puntos de atención, a la ciudadanía suficientes, paras los trámites pertinentes, surge la necesidad de contar con una herramienta tecnológica ya sea este, producto o servicio software, sirviendo como apoyo para toma de decisiones en la más acertada distribución y ubicación de la o las oficinas necesarias para satisfacer la demanda de atención a la ciudadanía, la empresa OMBUGAMES CIA LTDA tiene la capacidad de proveer este servicio requerido.

¿Por qué se origina?

Por el crecimiento natural de la población de la ciudad de Riobamba, y la afluencia desproporcionada de ciudadanos a un punto central donde se encuentran ubicadas las oficinas de atención a los usuarios de la ciudad, y realizar sus trámites personales.

¿Qué lo origina?

La demora en la atención por los numerosos trámites diarios que sobrelleva la oficina municipal existente, genera aglomeración de ciudadanos muchas veces molestos por el tiempo de espera. Además, el no poseer una herramienta tecnológica que simule en tiempo real, interactivamente y observar el comportamiento de los habitantes, en la decisión a que oficina acudir para realizar los trámites pertinentes. Y con la información presentada en la simulación visual, ayudar a la toma de decisiones para la ubicación planificada de nuevos puntos de atención a los usuarios.

¿Cuándo se origina? No aplica

¿Dónde se origina? No aplica

¿Dónde se detecta? No aplica

2.4. Formulación de meta

Desarrollar una aplicación software, aplicando algoritmos de búsqueda genéticos a motores gráficos en tercera y segunda dimensión, usando parámetros de porcentajes de habitantes por parroquias urbanas y comportamientos de selección de objetivos en este caso las oficinas de atención al usuario. (Caso de estudio: servicio proporcionado por Ombugames Cia. Ltda., usando el mapa de la ciudad de Riobamba)

2.5. Objetivos

2.5.1. Objetivo general

Implementar algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba.

2.5.2. Objetivos específicos

1. Fundamentar aspectos teóricos acerca de la funcionalidad de algoritmos genéticos.
2. Diseñar la función objetivo, que representará el comportamiento de los individuos al acudir a las oficinas municipales públicas.
3. Implementar una aplicación software para simular los comportamientos en tiempo real, tanto para dispositivos móviles, laptops y computadores de escritorio.
4. Generar reportes como complemento de la aplicación, con datos estadísticos y visuales de la afluencia y descongestionamiento de usuarios, para la toma de decisiones de ubicación de nuevas oficinas públicas municipales de la ciudad de Riobamba.

2.6. Delimitación Funcional

2.6.1. ¿Qué será capaz de hacer el producto final del trabajo de titulación?

- Simular los comportamientos de afluencia de los usuarios, mediante la incorporación de algoritmos genéticos, a las oficinas municipales públicas en el mapa de la ciudad de Riobamba, en una aplicación software usando un motor de gráficos.
- Proporcionará información estadística y visual de los resultados de afluencia y congestión en las oficinas ubicadas en el mapa de la ciudad de Riobamba.
- Generar reportes para una adecuada toma de decisiones para la planificación de nuevas oficinas públicas en la ciudad de Riobamba.

2.6.2. ¿Qué no será capaz de hacer el producto final del trabajo de titulación?

- No se alojará en clouds públicas ni privadas para descargas a computadores.
- No se alojará-en Apple Store, tampoco en Google Store-para descargas en dispositivos móviles.
- El proceso de ubicación de nuevas oficinas en el mapa de la ciudad de Riobamba no será automático, se lo realizará manualmente en interacción con el usuario de la aplicación software.

CAPÍTULO 3

Marco Teórico

3.1. Definiciones y conceptos

3.1.1. Algoritmos Genéticos

En los últimos años ha incrementado el empeño en aprender los procesos de búsqueda guiada, redes neuronales, el adiestramiento de algoritmos genéticos entre otros. Los algoritmos empiezan con varias soluciones candidatas, estos analizan a las candidatas, la peor es excluida y las mejores se disponen y se reproducen, se combinan dos candidatas emparejadas de alguna forma para originar una nueva, si la descendencia heredó los aspectos positivos de sus progenitores, por lo tanto se convierte en un nuevo candidato de nueva generación para reproducirse, caso contrario si heredó los aspectos negativos de sus progenitores, esta será descartada (Norvig, 2015).

Holland (1975) mediante la estructura y funcionamiento de su algoritmo genético, describe que existe la posibilidad de transición de una población de individuos expresados en cromosomas de bits, a una población distinta y nueva. En el proceso de transición de poblaciones, se imita a la selección natural de la vida, realizando procesos de cruces, selecciones de cromosomas, y mutaciones del valor de los mismos, como nos enseña el estudio de la genética. El algoritmo genético de Holland se lo considera un algoritmo base, donde cada bit puede tener un valor de 0 o 1, representa a cada gen de un cromosoma. Los cromosomas son seleccionados por sus características de reproducción a nuevas generaciones, y mediante su compatibilidad se define su descendencia, mientras mayor sea la compatibilidad mayor será la descendencia. El proceso de cruce separa los cromosomas de dos individuos distintos de la población, para recombinarlos y generar un nuevo individuo tal cual la biología realiza su combinación de cromosomas. El proceso de mutación, de forma aleatoria selecciona un gen bit y cambia su valor establecido. El proceso de inversión en cambio altera, a un cromosoma un conjunto de información, su orden (Rojas Hernández, 2014).

Los Algoritmos Genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas

soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de estas.

Algoritmo genético es usado para encontrar una solución óptima a problemas de búsqueda, utilizando métodos adaptativos. Según la teoría de la evolución de Darwin, la descendencia de una población evoluciona en cada generación, por medio de mecanismos de selección natural, tal cual como lo realizan los organismos con vida, permitiendo la supervivencia de los más adaptables. Al imitar este proceso de evolución, un algoritmo genético está capacitado de generar soluciones a problemas planteados. Una buena codificación permite que las soluciones del algoritmo genético converjan en valores óptimos al problema, cada posible solución evolucionará en cada generación (Mejía, 2012).

El algoritmo genético como parte de la inteligencia artificial y exponente de la computación evolutiva, es un método de búsqueda y optimización de problemas, utilizando una analogía directa con la selección y comportamiento natural. Partiendo de una población inicial, en cada población el individuo con mayor nivel de adaptación al problema, más alta será la probabilidad de poder reproducirse con otro individuo seleccionado de la misma forma, y permitir que su información genética trascienda en las nuevas generaciones.

La siguiente tabla muestra la relación entre elementos genéticos y sus equivalentes en los algoritmos genéticos.

Tabla 1. Equivalencia de elementos entre Genética y Algoritmo Genético.

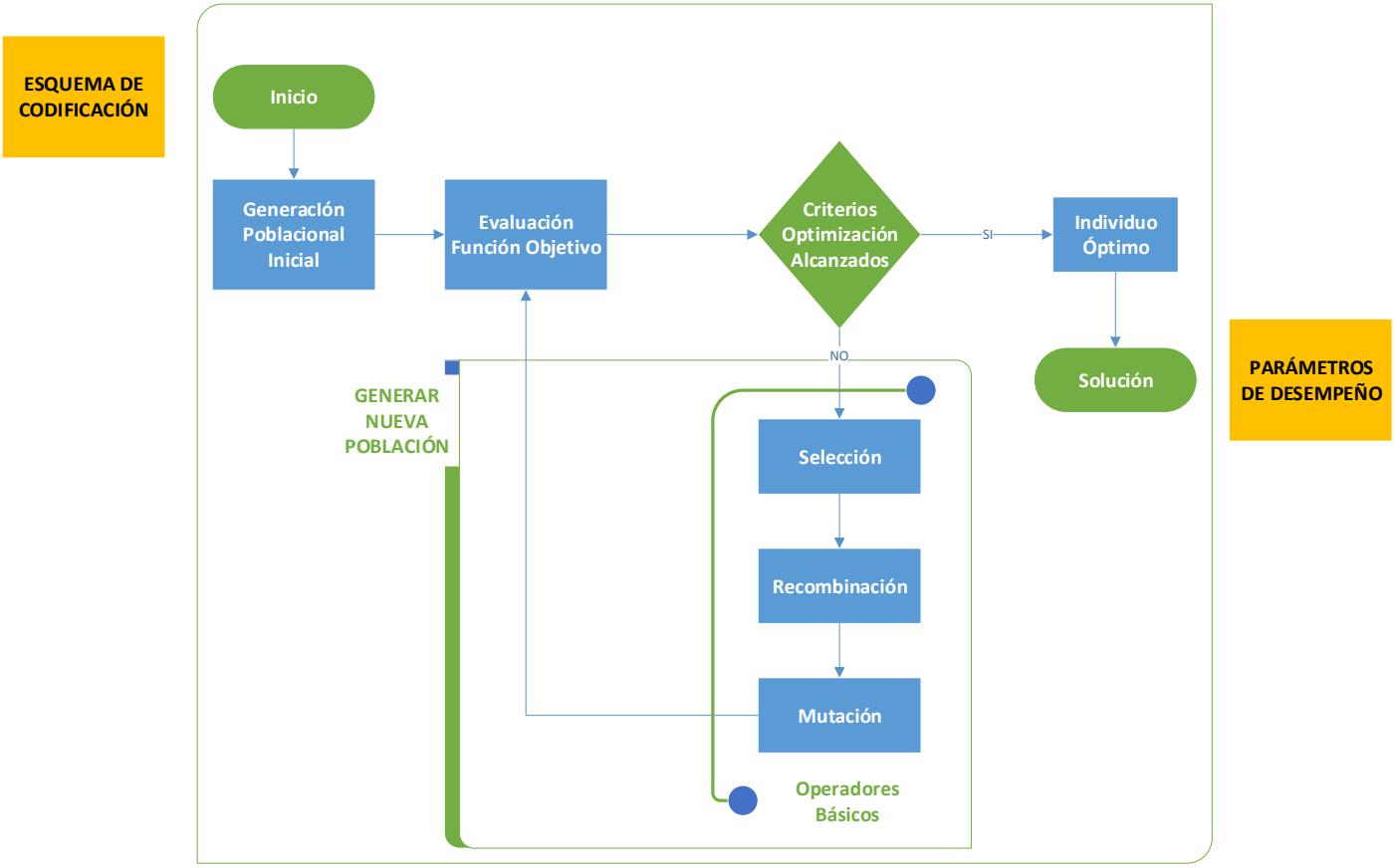
GENETICA	ALGORITMO GENETICO
Genotipo	Cadena codificada
Fenotipo	Punto no codificado
Cromosoma	Cadena
Gen	Posición de la cadena
Alelo	Valor en una cierta posición
Aptitud	Valor de la función objetivo

Fuente: elaboración propia

3.1.1.1. Funcionamiento del algoritmo genético

El algoritmo genético parte de una población inicial, con individuos, a los cuales se los codifica en 0 y 1, bits. Cada individuo es evaluado por la función objetivo, los individuos que cumplan con dicha función objetivo serán los más aptos, y son seleccionadas para intercambiar cromosomas con otros individuos aptos, generando nuevos individuos, con nueva información genética. Estos nuevos individuos se someten a mutaciones aleatorias de un gen en específico de cromosomas, y nuevamente serán evaluados por la función objetivo, repitiendo nuevamente el proceso con la nueva población generada. Estas iteraciones se realizan hasta encontrar uno o varios individuos que cumplan con los criterios de optimización esperados, obteniendo la solución óptima.

Figura 1. Esquema del Funcionamiento - Algoritmo Genético



Fuente: elaboración propia

Una plantilla del funcionamiento de un Algoritmo Genético se muestra en la figura 2., cada parte individual son expuestas más adelante.

Figura 2. Algoritmo Genético Básico

```
Escoger la población inicial de cromosomas;
while condición terminal no satisfecha do
    repeat
        if condición de cruce satisfecha then
        {
            Seleccionar cromosomas de los padres;
            Escoger parametros de cruce;
            Realizar cruce;
        }
        If condición de mutación satisfecha then
        {
            Escoger puntos de mutación;
            Realizar mutación;
        }
        Evaluación de aptitud de la descendencia;
    until suficiente descendencia creada;
    seleccionar nueva población;
endwhile;
```

Fuente: elaboración propia

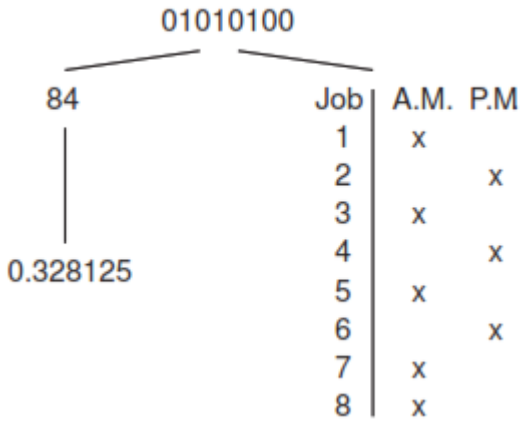
Esta es una formulación de codificación general, que admite formas de selección, cruce y mutación. Todas estas son especificadas por el usuario, mediante las cuales se realizan los cruces y mutaciones, permitiendo el origen de una nueva población, y repitiendo el proceso, hasta que se cumpla la condición final de todo el proceso.

3.1.1.2. Codificación Genética

Representación genética, también conocida como codificación genética, describe los elementos del genotipo y como estos elementos se mapean. La codificación depende mucho del problema.

Codificación Binaria, representación mediante ceros (0) o unos (1).

Figura 3. Representación binaria



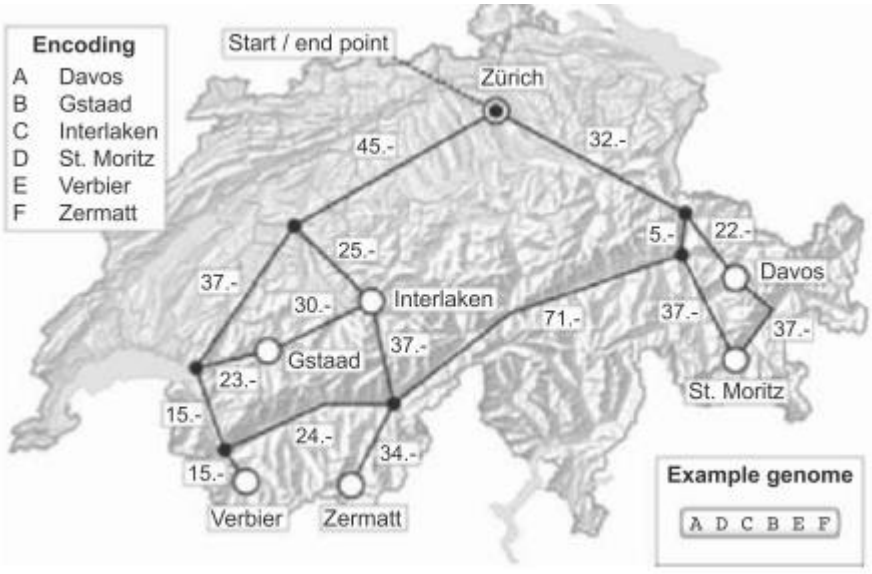
Fuente: (Floreano & Mattiussi, 2008)

Figura 3. Representación binaria de un genoma de 8 bits de longitud de, a la izquierda: número entero y número real en el intervalo del [0,1]. A la derecha: horario de trabajo para turnos de la mañana y tarde.

Muchos cromosomas son posibles con la codificación binaria incluso número reducido de alelos. Esta codificación a menudo no es adecuada para problemas y, a veces, se deben hacer correcciones después del cruce y / o mutación.

Codificación de Permutación, representación mediante secuencias de nodos.

Figura 4. Problema de ruta de viaje



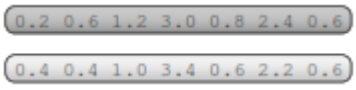
Fuente: (Floreano & Mattiussi, 2008)

Figura4. Planear una ruta a seguir por varios puntos, con el mínimo costo de transportación (los números indican el costo de la ruta). Al tope los nombres de los puntos y las conexiones entre ellos. Abajo a la derecha, se muestra la representación secuencial de los nodos a seguir.

La codificación de permutación solo es útil para problemas de ordenamiento secuencial, incluso en esta representación en algunos tipos de cruce se debe realizar correcciones de mutación para dejar el cromosoma consistente.

Codificación de valor real, el genotipo consiste de conjunto de *n* números que pertenecen al dominio de números reales, representados como valores de punto flotante también llamados números decimales. Esta representación es adecuada para soluciones que requieren alta precisión en la optimización de parámetros.

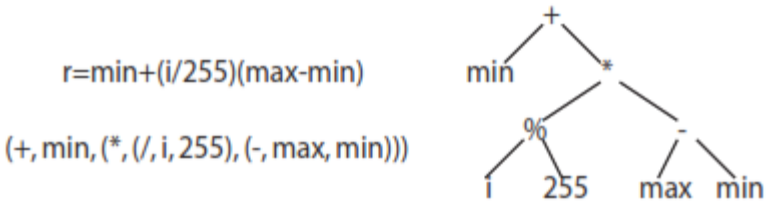
Figura 5. Representación mediante números decimales



Fuente: (Floreano & Mattiussi, 2008)

Codificación basada en árboles, estas representaciones son adecuadas para representar, puntos y condiciones de ramificación de estructuras jerárquicas.

Figura 6. Representación mediante árbol.



Fuente: (Floreano & Mattiussi, 2008)

Figura 6. Representación en árbol de la expresión para mapear números enteros en números reales. A la izquierda: formato de la expresión y la lista anidada. A la derecha, representación del árbol.

Las representaciones genéticas pueden afectar dramáticamente la capacidad de evolución del sistema, es decir, la probabilidad de generar mejoras mediante la aplicación de operadores genéticos (Wagner & Altenberg, 1996). Por ejemplo, las representaciones directas, utilizan genotipos cuyas longitudes son proporcionales al número de parámetros libres del problema. A menos que la representación genética esté bien adaptada para resolver el problema, los genotipos más largos tienden a corresponder a espacios de búsqueda más grandes y

probablemente disminuyen la probabilidad de producir una mejora a través de mutaciones aleatorias de pocos genes.

Otro problema es que la dimensión del espacio de búsqueda y la cantidad de soluciones posibles están predefinidas y son constantes, lo que limita el potencial de generar soluciones más complejas a lo largo de generaciones. El problema de la evolución y de los mapeos genotipo-fenotipo más adecuados surge cuando uno está interesado en evolucionar fenotipos hechos de varios componentes diferentes, como dispositivos electrónicos complejos o robots autónomos, o cuando uno está interesado en una evolución abierta. Investigadores han intentado idear representaciones genéticas más eficientes y procesos de mapeo inspirados en los principios de la expresión del gen. Sin embargo, no siempre está claro en qué medida un realismo biológico superior en la codificación genética es útil para la evolución artificial. Los ejemplos más prometedores capturan un subconjunto de características biológicas que aportan una ventaja específica al problema en cuestión. Por lo tanto, a menudo se abordan diferentes dominios de problemas con diferentes asignaciones de genotipo a fenotipo (Floreano & Mattiussi, 2008).

3.1.1.3. Función de Evaluación

La función de evaluación o también llamada función de aptitud, evalúa que tan cerca esta una solución de la solución óptima basado en un problema, determinando que tan adecuada es dicha solución.

Cada solución, en algoritmos genéticos, tiene representaciones específicas de acuerdo al problema a resolver como se detalla en el apartado 3.1.1.2; cada solución deberá ser evaluada y encontrar un conjunto de soluciones cada vez mejor para solucionar un problema. Por lo tanto, toda solución de este conjunto deberá recibir una valoración, la que indicará cuan cerca estará de cumplir con la solución óptima y sus especificaciones generales. Esta valoración será generada después de aplicar la función de evaluación. La función de evaluación deberá cumplir con los siguientes requisitos:

- La función de evaluación deberá estar claramente definida.
- La función de evaluación deberá implementarse de manera eficiente. Si la función de evaluación se convierte un problema del algoritmo, por lo tanto, reducirá la eficacia del algoritmo genético.
- La función de evaluación, para saber qué tan adecuada es una solución dada, deberá medirse cuantitativamente para resolver el problema.
- La función evaluación deberá generar resultados intuitivos. Los mejores - peores individuos deberán tener altas – bajas valoraciones respectivamente.

La función de evaluación define el criterio para ordenar las hipótesis que potencialmente pueden pasar a formar parte de la siguiente generación (Crevillén & Díaz, 2012).

Función de Evaluación o también llamada función de fitness o función objetivo constituye, a lo que una generación de individuos debe adecuarse, mediante condiciones deseadas, lo que resulta en la asignación de valores reales, los cuales serán puestos a prueba para su posterior selección (ASAP, 2015).

Otro autor trata a la función objetivo como una optimización mediante un algoritmo genético a una función, y la función evaluación menciona que es una operación que proyecta el valor de adaptación de cada individuo para su posterior supervivencia (Alba Torres, 1999).

La función de aptitud asocia una puntuación numérica a cada fenotipo en la población. Cuando el fenotipo es el resultado de un proceso de crecimiento o puede ser modificado por otros procesos no genéticos, como las fluctuaciones de por vida o el aprendizaje, la función de aptitud evalúa indirectamente también la calidad de los procesos de desarrollo o aprendizaje. Hay dos aspectos importantes involucrados en el diseño de una función de aptitud: (a) la elección y combinación de componentes de aptitud, y (b) la forma en que se evalúa la función (Floreano & Mattiussi, 2008).

3.1.1.4. Operadores básicos

Instanciando a la evolución de las especies, aplicando el método basado en algoritmo genético, se necesita establecer sobre la población una imposición selectiva, y como resultado solo los individuos mejor adaptados a resolver el problema, se reproduzcan y transmitan su información genética a la siguiente generación que conformaran la nueva población. Para esto se especifica tres operadores básicos de un algoritmo genético:

- Selección
- Cruce, llamado también recombinación
- Mutación

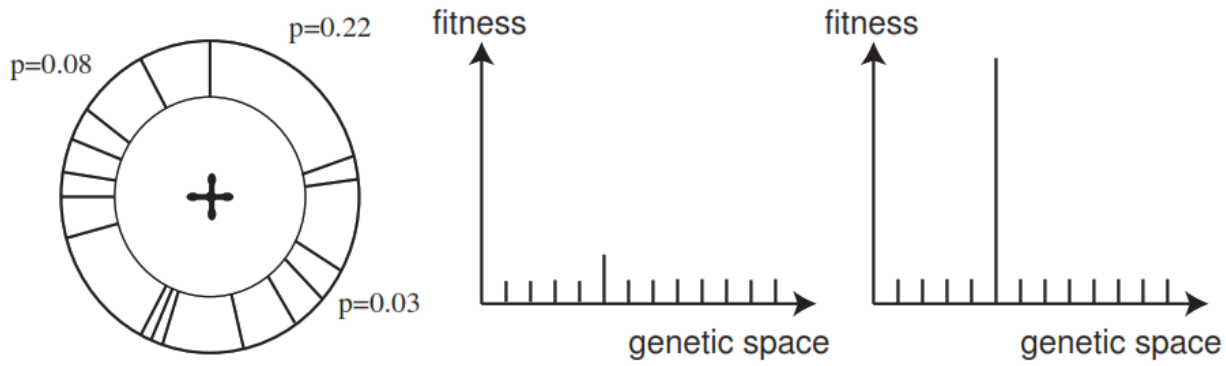
- **Selección**

La selección es fundamental puesto que permite la búsqueda y los individuos que conforman una generación de población convergen, encontrando una posible solución óptima, esto sucede tanto en algoritmos paralelos o secuenciales, sean estos evolutivos (Alba Torres, 1999).

Según la teoría de Darwin, sobreviven y transmiten su información para futuras generaciones, solo los individuos con mayor adaptabilidad, estos son seleccionados y tendrán mayores posibilidades de reproducirse, creando poblaciones con mayores cualidades y facultades de solución a un problema (Arranz de la Peña & Parra, 2015).

Después de aplicar la función objetivo a cada individuo de la población se seleccionarán los individuos más aptos, para compartir su información genética a las siguientes generaciones de individuos, por uno o varios criterios, tales como selección randómica, elitista, jerárquica, o mediante competencias internas, entre otros. El rol de selección es asignar el mayor número de descendientes como mejores individuos a la siguiente generación. La presión de la selección asegura la fracción de individuos con descendencia para la próxima generación de población, mientras más alto sea el grado de la presión de selección, un número más reducido de individuos será seleccionado para la reproducción, esta estrategia puede llevar muy rápido a la sección de una solución de aptitud optima, pero conlleva un riesgo de perder la diversidad en las poblaciones y llegar a una convergencia prematura, por tal motivo se debe llegar a un balance entre la presión de la selección y otras factores que puedan generar diversidad en las poblaciones generadas.

Figura 7. Tipos de selección



Fuente: (Floreano & Mattiussi, 2008)

Izquierda figura 3. Rueda de ruleta de selección proporcional, cada individuo está representado por una casilla de la ruleta y su tamaño es proporcionado por la probabilidad de reproducción del individuo.

Centro figura3. El eje horizontal, cada línea vertical representa a los individuos del espacio genético, todos los individuos tienen una aptitud similar determinado por el tamaño de la línea vertical, en este caso existe una probabilidad baja de que el mejor individuo genere más descendencia que sus competidores.

Derecha figura 3. En el espacio genético representado en el eje horizontal, sobresale la aptitud de un individuo sobre los demás, en consecuencia, las generaciones posteriores, casi todos sus individuos serán copias de sí mismo.

- **Cruce o recombinación**

Existen numerosas variantes de recombinación, todas ellas clasificables en tres categorías de alto nivel:

Operador de cruce puro: aplicables en cualquier algoritmo genético.

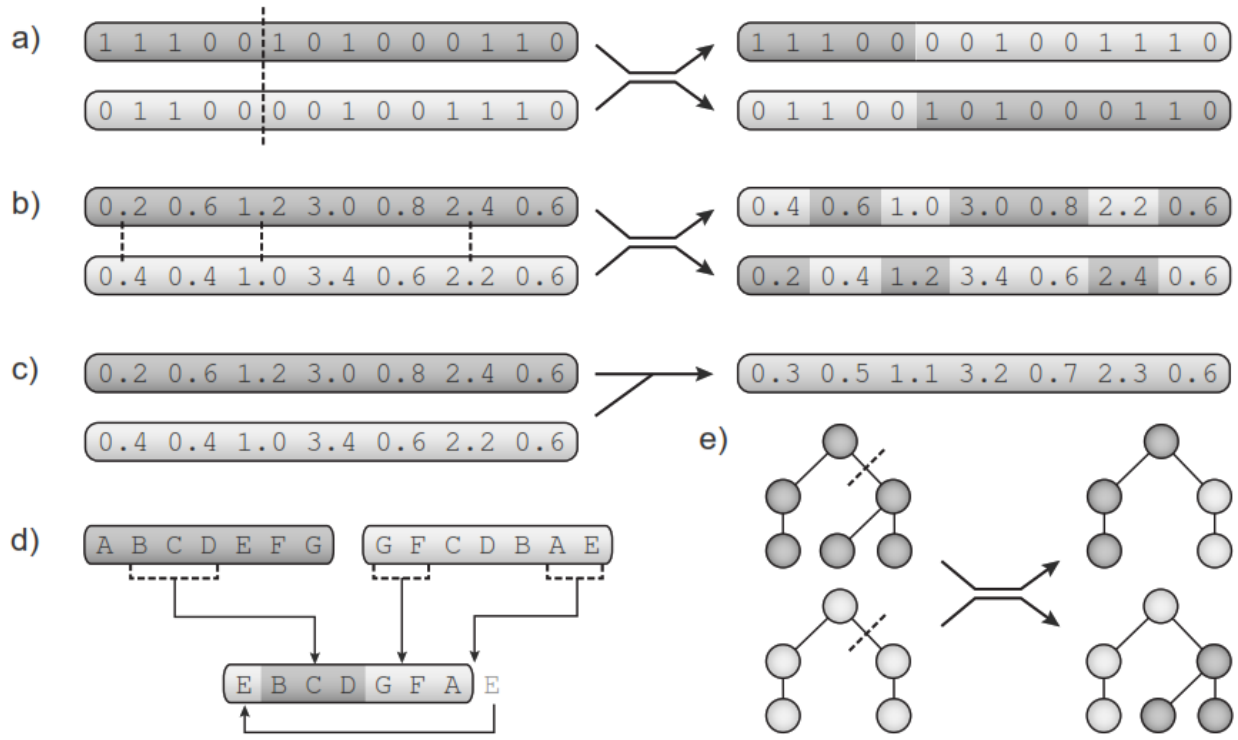
Operador de cruce híbrido: usan procesos sin ninguna relación a la genética, y lo combinan con un operador puro,

Operador de cruce basado a la problemática: son operadores adaptados específicamente a un problema, basados en el análisis y estudio de dicho problema (Alba Torres, 1999).

La recombinación asevera que los individuos generados hereden de los progenitores características mediante la construcción de recombinaciones por pares, de los genomas de individuos seleccionados. La idea detrás de la recombinación genética es que algunos de los herederos resultantes, pueden beneficiarse de la sinergia resultante de la combinación de subsoluciones encontradas por los padres.

El cruce se asegura de que la descendencia hereda las características de los padres mediante la creación de recombinaciones por pares de los genomas de los individuos seleccionados. Este operador también se conoce como recombinación. Los herederos recién creados se emparejan al azar y el operador de cruce intercambia partes de sus genotipos con una probabilidad. Los operadores de cruce vienen en diferentes formas, que se adaptan a las representaciones genéticas. La idea detrás de la recombinación genética es que algunos de los descendientes resultantes pueden beneficiarse del efecto sinérgico que resulta de la combinación de las subsoluciones encontradas por los dos padres.

Figura 8. Operadores de cruce



Fuente: (Floreano & Mattiussi, 2008)

Figura 4, a) Cruce en un punto, consiste en seleccionar un punto de cruce de forma randómica, en cada uno de las cadenas, e intercambiando el material genético para crear los nuevos individuos.

Figura 4, b) Cruce uniforme, fundamenta en intercambiar el contenido genético en N puntos seleccionados de forma randómica.

Figura 4, c) Cruce aritmético, crea un genotipo únicamente, tomando el promedio de N posiciones de forma randómica de dos cadenas.

Figura 4, d) Cruce por secuencia, crea un genotipo con la secuencia escogida de una de las cadenas, y completándola con secuencias de la otra cadena, cabe recalcar que, en este tipo de cruce, no deben estar repetidos elementos de cada cadena.

Figura 4, e) Cruce para representaciones de árbol, selección randómica de un nodo, para intercambiar los sub arboles correspondientes.

- **<u>Mutación</u>**

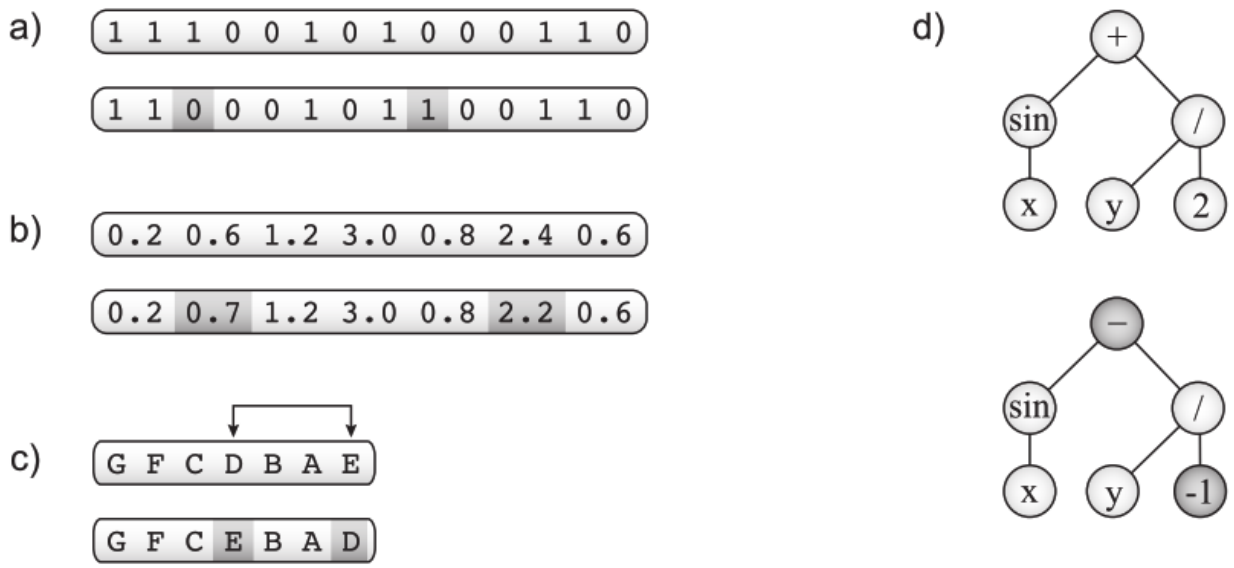
La mutación opera al nivel del individuo. Las mutaciones son pequeñas modificaciones aleatorias del genotipo que permiten a la evolución explorar variaciones de soluciones existentes. El operador de la mutación debe diseñarse de manera que cada punto en el espacio de la representación genética pueda ser potencialmente

alcanzado. Las mutaciones son útiles para escapar de los mínimos locales y para lograr un mayor progreso en poblaciones altamente convergentes donde la recombinación genética tiene poco efecto. Sin embargo, el número y tamaño de las mutaciones debería ser relativamente bajo para evitar la pérdida de soluciones descubiertas previamente (Floreano & Mattiussi, 2008).

La mutación se aplica a un bajo porcentaje de la población y su efecto no es muy notable en la mayor parte de los casos. Los propósitos del método de mutación son

- Salvar la diversidad genética evitando la convergencia prematura de la población.
- Indagar áreas probablemente no estudiadas y evitar de una convergencia prematura al algoritmo genético.

Figura 9. Ejemplos de mutaciones



Fuente: (Floreano & Mattiussi, 2008)

Figura 5, a) En representaciones binarias, la mutación consiste en alternar el valor del bit seleccionado.

Figura 5, b) En representaciones de valores reales, la posición seleccionada es modificada por un valor aleatorio extraído de una distribución Gaussiana N (0, σ), donde 0 es la media y σ es la varianza.

Figura 5, c) En representaciones que describen secuencias, la mutación consiste en intercambiar los contenidos de dos posiciones elegidas al azar en el genotipo del individuo.

Figura 5, d) En representaciones de árboles, la mutación consiste en cambiar el contenido de un nodo seleccionado con otro elemento del mismo conjunto.

3.1.1.5. Parámetros que controlan el desempeño del algoritmo genético

Como se mostró en la Figura 1., existen parámetros de control en el desempeño del algoritmo genético:

a. Tamaño de la población
b. Probabilidad de cruce o recombinación
c. Probabilidad de la mutación
d. Número de generaciones

a. Tamaño de la población

En este parámetro se debe encontrar un criterio balanceado con respecto al número de individuos que conformará cada población generada, puesto que, si el número es muy limitado, la solución del algoritmo genético puede ser deficiente, puesto que tiene pocas posibilidades de realizar reproducciones entre individuos, resultando en búsquedas escasas. Al otro extremo si los individuos de la población son exuberantes, el desempeño del algoritmo será excesivamente lento y en muchos de los casos no podrá visualizar una solución óptima.

El tamaño de la población está estrechamente relacionado con las características y capacidades del computador, depende de la memoria RAM y el procesador del computador en el que se está ejecutando el algoritmo genético. Y esta es otra razón para elegir un correcto número de individuos.

b. Probabilidad de cruce o recombinación

La recombinación indica la frecuencia con la que se producen cruces y en que posiciones del cromosoma de los progenitores, en otras palabras, dos progenitores pueden reproducirse entre ellos con un alto índice de probabilidad, puede ocurrir que los hijos tomarán la información idéntica de los padres siendo sus copias puntuales, pero de no ser así los hijos podrán adquirir parte de información de ambos padres.

c. Probabilidad de la mutación

La mutación se debe utilizar en un bajo porcentaje, se recomienda del 0.1 al 5 % en codificación binaria, debido al riesgo de que se le aplique sobre el único individuo más apto de una buena solución y la mute. Pero

esto no sucede en general, puesto que los individuos con las mejores soluciones sean varios, y es poco probable que se aplique la mutación a todos ellos. Un porcentaje elevado de mutación puede provocar que la búsqueda de la solución en el algoritmo genético sea aleatoria, mientras que un porcentaje demasiado bajo puede provocar una búsqueda con convergencia prematura, o que ciertas zonas del espacio de búsqueda no sean exploradas.

d. Número de generaciones

El número de generaciones es el número de iteraciones que genera el algoritmo genético, se lo puede designar que el número de iteraciones controle el propio algoritmo genético al llegar a una convergencia única, o se lo puede parametrizar manualmente para que arroje los individuos alcanzados en una iteración dada.

3.2. Estado del Arte

En la actualidad las instituciones públicas municipales buscan prestar un servicio de calidad en los trámites ciudadanos de Riobamba, distribuyendo adecuadamente las oficinas para la atención ciudadana, y disminuyendo el tiempo de espera y maximizando el número de recaudaciones.

Para lograr una adecuada planificación de ubicación urbana de nuevas oficinas municipales, para la atención ciudadana, se necesita realizar estudios y búsqueda de datos con las posibles locaciones en el mapa de la ciudad para la misma. Al no contar con una aplicación software que agilite el proceso, se lo realiza manual. He aquí la importancia de la herramienta software y la aplicación de algoritmos genéticos en la misma.

Las siguientes citas de artículos científicos, papers, revelan la aplicación de Algoritmos genéticos en diversas áreas, para solucionar diferentes problemas que se presentan.

H-DARP (Hybrid Genetic Algorithm for the Heterogeneous Dial-A-Ride Problem) consiste en determinar la planeación de la ruta de las demandas de los usuarios, mientras se minimiza el costo total de la ruta. Para solucionar el problema se propone a Algoritmo genético híbrido, incorporando técnicas efectivas de búsquedas locales en un Algoritmo genético en orden para mejorar la convergencia y reducir el tiempo de procesamiento computacional. (Masmoudi, Brakers, & Masmoudi, 2017).

Evolutionary Algorithm – Based Multi – objective Control Scheme for Food Drying Process. El secado de los alimentos es uno de los métodos más importantes para prevenir el crecimiento microbiano durante la conservación. Se desarrolla un esquema de control de modelo interno para el secado de piña usando algoritmos evolutivos, usando algoritmo genético (GA) y optimizando el enjambre de partículas, para lograr la calidad deseada. Con el fin de reducir el esfuerzo de control y por lo tanto el coste, sin comprometer la calidad deseada,

se formula también un esquema de control multi objetivo utilizando el método de suma ponderada (Manonmani , 2017).

En aplicaciones de visión por computadora es esencial detectar la especialidad de bordes de objetos en aplicaciones de imágenes médicas. En este papel propuesto el umbral se utiliza basado en el algoritmo genético para detectar el borde del objeto para las imágenes médicas. En el algoritmo propuesto, existen principios para generar soluciones útiles para la optimización y la búsqueda de objetos (Kawther, 2016).

El algoritmo genético (GA) y el enjambre de partículas completamente informado se hibridan para resolver el problema de programación de proyectos con limitación de recursos multi-modo con la minimización del proyecto makespan como objetivo sujeto a restricciones de recursos y precedencia. En el algoritmo genético híbrido propuesto-algoritmo de enjambre de partículas completamente informado. Una clave aleatoria y los esquemas de representación de listas de modos relacionados se usan como esquemas de codificación, y el esquema de generación de programación en serie multi-modo se considera como el procedimiento de decodificación (Sebt, Afshar, & Alipouri, 2016).

Se formula un nuevo problema de colocación de sensores para cubrir dos objetivos: (1) asegurar la calidad del agua entregada a los consumidores; Y (2) la detección de cualquier evento de contaminación lo más pronto posible para minimizar sus consecuencias, mediante la maximización de: (1) la cobertura de la demanda; Y (2) probabilidad de detección con restricciones de tiempo para redes deficiente en presión. La red puede resultar deficiente en la presión debido al uso continuo de la red de distribución de agua más allá de su vida de diseño. Los dos objetivos se combinan usando pesos. El algoritmo genético se utiliza para obtener ubicaciones óptimas del sensor (Rathi & Gupta, 2017).

Las fuentes de generación distribuida (DG) son cada vez más prominentes en los sistemas de distribución debido a las demandas incrementales de energía eléctrica. Las ubicaciones y capacidades de las fuentes DG han impactado profundamente en las pérdidas del sistema en una red de distribución. Se presenta una nueva combinación de algoritmo genético (GA) / optimización de enjambres de partículas (PSO) para una óptima ubicación y dimensionamiento de DG en sistemas de distribución. El objetivo es minimizar las pérdidas de potencia de la red, mejorar la regulación del voltaje y mejorar la estabilidad del voltaje en el marco del funcionamiento del sistema y las restricciones de seguridad en los sistemas de distribución radial (Moradi & Abedini, 2012).

Hoy en día es muy importante mantener una seguridad de alto nivel para garantizar una comunicación inequívoca y confiable de información entre diversas organizaciones. Pero la comunicación fehaciente de datos a través de Internet y cualquier otra red está siempre bajo amenaza de intrusiones y malversaciones. Así que los sistemas de detección de intrusos se convierten en un componente necesario en términos de seguridad

informática y de red. Hay varios enfoques que se utiliza en las detecciones de intrusión, pero por desgracia, ninguno de los sistemas hasta el momento no es completamente perfecto. Por lo tanto, la búsqueda de mejora continua. En esta progresión, aquí presentamos una Intrusion Detection System (IDS), mediante la aplicación del algoritmo genético (GA) para detectar eficientemente diversos tipos de redes intrusiones. Los parámetros y los procesos de evolución de GA se discuten en detalle y se implementan. Este enfoque utiliza la teoría de la evolución para la evolución de la información con el fin de filtrar los datos de tráfico y, la complejidad (Sazzadu, Mukit, & Bikas, 2012).

Las pruebas se utilizan en una variedad de contextos en la actividad del aprendizaje cotidiano y en todas partes. Son un método específico en el proceso de evaluación (evaluación), que es una parte importante de la actividad educativa. El establecimiento de una secuencia optimizada de ensayos (SOT) procedente de un grupo de ensayos que tienen el mismo sujeto, con ciertas restricciones correspondientes a un cierto deseo del evaluador, puede ser una tarea lenta, porque la restricción puede ser diversa y el número de las pruebas puede ser alto. En este caso, este trabajo presenta un método para generar secuencias optimizadas de pruebas dentro de una batería de pruebas utilizando un algoritmo genético. Asociamos un número de palabras clave representativas con una prueba. El usuario expresa la restricción estableciendo un número de palabras clave que se aproximan mejor al sujeto que se desea probar. El algoritmo genético ayuda a encontrar las soluciones optimizadas y utiliza menos recursos de hardware (Popescu, Bold, & Nijloveanu, 2016).

Algoritmo genético el cual permita determinar los tamaños de lote de producción y su programación en un sistema de manufactura de una máquina para órdenes multiproducto, cuya función objetivo minimiza la suma de los costos de inventario por terminaciones tardías y de alistamiento. El problema contempla un conjunto de órdenes a ser procesadas con sus respectivas fechas de entrega. Cada orden debe ser entregada en su totalidad. Dentro de la programación de los trabajos se consideran tiempos de alistamiento dependientes de la secuencia. En la metaheurística implementada se utiliza de manera embebida un método heurístico para el cálculo de la función de adaptación. El método heurístico presentado es una variación del Optimal Timming Algorithm el cual involucra los tiempos de alistamiento dependientes de la secuencia. Se desarrolla un diseño de experimentos para probar el desempeño del algoritmo utilizando instancias generadas de forma aleatoria y comparando sus soluciones contra las encontradas por un método exacto. Los resultados muestran que el algoritmo logra un buen desempeño tanto en tiempo de ejecución como en calidad de la solución especialmente en instancias grandes (Peña Arenas & López Castro, 2016).

Implementación de un algoritmo genético orientada al cálculo de patrones de reconfiguración de arreglos fotovoltaicos. La solución propuesta se compara con el enfoque clásico de fuerza bruta, el cual es muy restrictivo debido a sus tiempos de procesamiento excesivamente largos. Se describe el proceso de puesta

punto del algoritmo y se presentan varios casos de estudio, incluyendo perfiles de sombreado parcial para el arreglo fotovoltaico. Los resultados muestran un desempeño muy superior del algoritmo genético en contraste con el enfoque de fuerza bruta (Camarillo Peñaranda, Ramírez Quiroz, González Montoya, Bolaños Martínez, & Ramos-Paja, 2015).

CAPÍTULO 4

Metodología

4.1. Método aplicado

4.1.1. Metodología Scrum

En el transcurso del desarrollo de software existen inconvenientes en los puntos de gestión de cada actividad, por lo cual se buscó un marco de trabajo que pueda manejar estos inconvenientes, aplicando técnicas para que el proyecto sea entregado en los tiempos estimados. Scrum al ser un marco de trabajo ágil, permite subdividir las actividades en pequeñas tareas conocidas como sprints los cuales derivan en entregables funcionales al final de cada uno de estos ciclos.

El marco de trabajo Scrum permite trabajar de forma directa con el usuario final, verificando y aprobando cada tarea realizada por el equipo de desarrollo. El momento de existir alguna falla Scrum permite refactorizar para llegar al correcto funcionamiento del entregable. Es muy importante en Scrum cada estimación y rol, en consecuencia, actividad que se asigna a cada persona del equipo de trabajo, debe ser muy concisa y clara.

4.1.2. Marco Trabajo Scrum Roles

Existen diversos roles en Scrum, para el desarrollo de la aplicación software se mencionarán los que se usaron, cada uno de estos juegan un papel importante, ya sea para validar actividades, o para mediar ideas entre los mandos superiores, con el fin de cumplir con los tiempos estimados de cada una de las actividades.

- **Product Manager**.

Es la persona encargada de exponer los requerimientos, conocido como usuario experto, el cliente es quien delega a esta persona, para presentar las historias de usuario originando y validando requerimientos para conseguir el producto requerido. Es la persona que tiene la potestad de pedir cambios de ser necesarios en el proceso, teniendo siempre la última palabra.

- **Scrum Manager.**

Individuo encargado de trabajar con la persona que tiene el rol de Product Manager, se lo puede decir que es el orquestador del proyecto, el cual permite la interacción de la información recopilada en las historias de

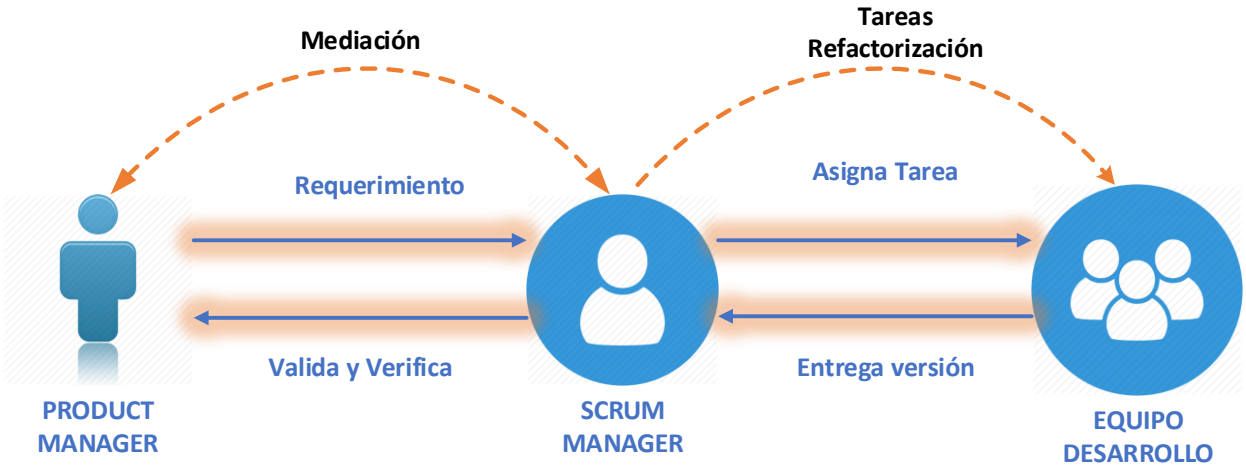
usuario hacia el equipo de desarrollo, cada requerimiento será validado en primera instancia por el Scrum Manager, encargado de mediar retrasos o cambios con el usuario final.

- **Equipo Desarrollo.**

Está conformado por la o las personas de desarrollo, con el propósito de concretar los requerimientos plasmándolos en pequeños softwares funcionales para el análisis por parte del Scrum Manager, dichos entregables estarán sujetos a pruebas unitarias para garantizar su correcto funcionamiento, todo esto elaborado bajo el contexto del Product Manager.

En la figura 6, se ilustra en conjunto el funcionamiento de los roles usados de la metodología Scrum, aplicado para el desarrollo de la aplicación software del presente trabajo, tendiendo las respectivas retroalimentaciones de los procesos establecidos.

Figura 10. Interacciones entre los Roles Scrum



Fuente: elaboración propia

4.1.3. Recopilación de Requerimientos mediante diagnóstico

En la ciudad de Riobamba los puntos de atención para los trámites de impuestos municipales, como son: pago de predio, mejoras de la ciudad, bomberos, creación de patentes, renovación de patentes, permisos de construcción entre otros. Se encuentra en el centro de la ciudad, exactamente en las calles 5 de junio entre Primera Constituyente y José de Veloz. Existiendo 4 ventanillas para la atención al público.

Figura 11. Edificio de la Ilustre Municipalidad de Riobamba

Fuente: elaboración propia

El tiempo de atención y número de usuarios atendidos se los detalla en la tabla 4, datos obtenidos a la observación y conteo el viernes 6, lunes 9 y martes 10 de enero del 2017, viernes 10, lunes 13 y martes 14 de febrero del 2017, desde el instante que el usuario llega a la cola de espera.

Tabla 2. Datos obtenidos mediante observación y conteo

PARÁMETRO	UNIDAD
Media Aritmética del Tiempo de espera por usuario	20 minutos
Número de puntos para atención	8 puntos
Horas laborables por ventanilla	8 horas

Fuente: elaboración propia

De los datos obtenidos se concluye lo siguiente:

- El número total de posibles usuarios atendidos al día es igual a:

Datos

$8\ horas = 480\ minutos$

$$\frac{480\ minutos\ por\ ventanilla}{20\ minutos\ por\ usuario} = \boldsymbol{24\ usuarios\ por\ punto\ de\ atención\ al\ día}$$

En total son **24 usuarios por punto de atención al día**, es el mejor escenario, sin que ocurra ningún evento externo a la cobranza, como ha ocurrido por fallas del sistema de cobranza, licencias médicas de los empleados públicos, entre otros.

- Al mes de 22 días laborables, tomando en cuenta los 8 puntos de atención de cobranza, se concluye:

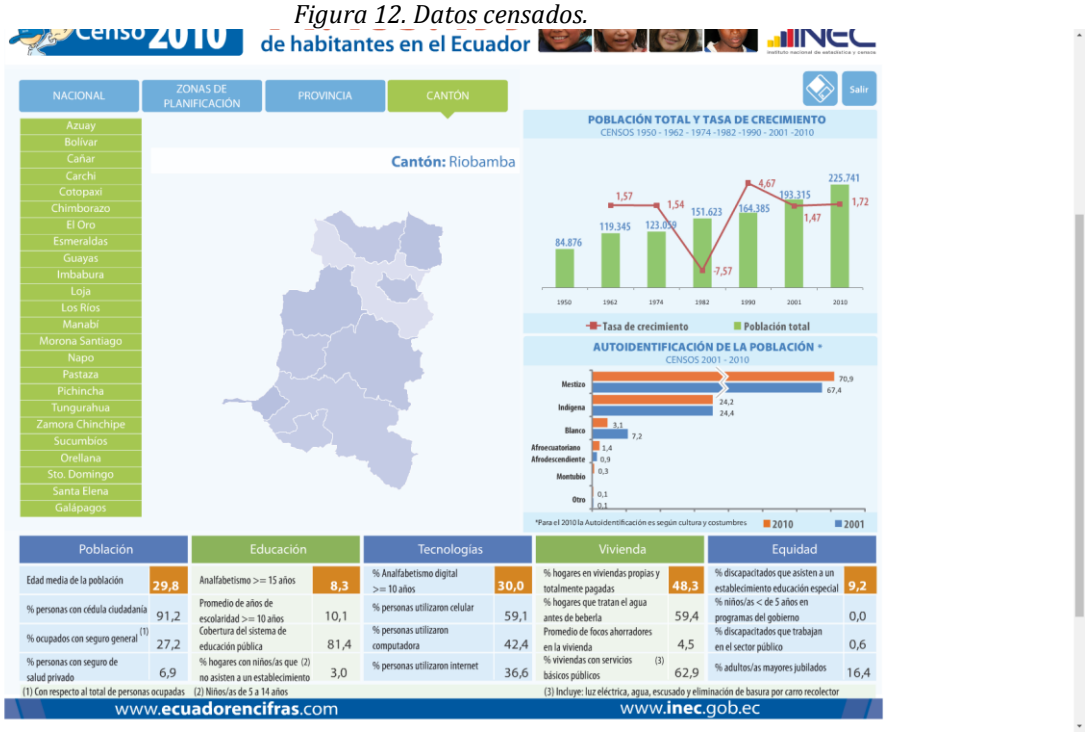
$(22\ días) * (24\ usuarios\ por\ punto\ al\ día) = 580\ usuarios\ por\ punto\ \ al\ mes$

$(8\ puntos) * (580\ usuarios\ por\ ventanilla\ al\ mes) = \boldsymbol{4640\ usuarios\ al\ mes}$

Da un total de **4640 usuarios** al mes aproximadamente, en el mejor de los casos, sin ningún agente externo que agregue tiempo de espera adicional al tiempo ya establecido.

En la ciudad de Riobamba existe un total de 225.741 habitantes (INEC, 2018).

Figura 12. Datos censados.



Fuente: Instituto nacional de estadísticas y censos. http://www.ecuadorencifras.gob.ec/resultados/

Los resultados del censo en el año 2010 de población y vivienda en el Ecuador, generó la siguiente tabla de viviendas en la provincia de Chimborazo en el cantón Riobamba. Tabla 4.

Tabla 3. Resultados Censo 2010 de población y vivienda. Cantón Riobamba.

Cantón	Hombres	%	Mujeres	%	Total	Viviendas	Analfabetismo	Edad promedio
Riobamba	106 840	48.7%	118 901	49.7%	225 741	79 842	8.3%	30

Fuente: Instituto nacional de estadísticas y censos. http://www.ecuadorencifras.gob.ec/resultados/

Encontrados los datos del Censo del año 2010, e intersecándolos con los datos calculados que dio como resultado 4640 usuarios atendidos al mes, por los puntos de atención existentes. Se realiza el cálculo de atención anual.

$(4640\ usuarios\ al\ mes) * (12\ meses) = \mathbf{55680\ usuarios\ al\ año}$

Mientras que al año 2010 existe un total de 79842 viviendas, (sin tomar en cuenta el posible crecimiento al año 2018), que sus propietarios tienen obligaciones tributarias y acudir a los puntos de atención.

$(79842\ viviendas\ con\ propietario) - (55680\ usuarios) = \mathbf{24162\ \ usuarios\ deficit}$

24162 usuarios aproximadamente quedarían sin ser atendidos según los datos proporcionados por el Instituto nacional de estadísticas y censos INEC, he ahí la necesidad de implementar nuevos puntos de atención a usuarios, para las recaudaciones tributarias municipales, sin sobresaturar los ya existentes, y ubicándolos en lugares disponibles en el mapa de Riobamba.

4.1.4. Recopilación de Requerimientos mediante deducción

Basado en todas las teorías y conocimiento adquirido en los años de estudio de la carrera juntamente con los proyectos de desarrollo de software realizados, en la presente aplicación de software, particularizando las necesidades para ofrecer el servicio de simulación de afluencia de usuarios a los puntos de atención en el mapa de Riobamba, por parte de la empresa OmbuGames Cia. Ltda., adaptando la realidad ligada a la observación ejecutada, hace que el método deductivo sea más directo; Unity 3d admite flexibilidad para el desarrollo del software, al ser un motor de gráficos en 2d y 3d, se convirtió en la herramienta base para plasmar en el proyecto la metodología, coordinando con el usuario final directamente, convirtiendo su proceso en personalizado facilitando verificaciones y validaciones de cada entregable funcional del proyecto, mermando el riesgo a fallos y demoras en los tiempos establecidos de entrega.

Es importante comprender, los datos conocidos para sacar premisas de los requerimientos en su proceso de análisis como, en esta particular se manifestó que en el mapa de Riobamba existen 8 puntos de atención a los usuarios, existiendo un déficit de atención a aproximadamente a 24162 usuarios, diagnóstico realizado en la sección 4.1.3. Conllevando a la congestión de usuarios que desean realizar sus pagos tributarios en los puntos de atención, con tal premisa se ve la necesidad de tener una herramienta informática para simular en tiempo real la congestión y descongestión de usuarios a los existentes y nuevos puntos de atención a la ciudadanía. Con esta conclusión se podría generar más premisas que sería la necesidad de almacenar en archivos json, las ubicaciones de los posibles nuevos puntos de atención a los usuarios.

La empresa OmbuGames Cia. Ltda., brindará el servicio mediante la instalación de la aplicación software en ordenadores con capacidad de procesar carga de datos alta, y encontrar las soluciones del algoritmo genético.

4.1.5. Análisis de Requerimientos

4.1.5.1. Requerimientos tipo funcionales

En la tabla 6, se detalla la lista de requerimientos funcionales, los mismos que se relacionan directamente con el funcionamiento fundamental de la aplicación, dichos requerimientos guían el proceso de desarrollo en interacción con el sistema.

Tabla 4. Requerimientos Funcionales

Código	Requerimiento	Tipo
Req01	Petición Mapa Riobamba	Ingreso
Req02	Pintar Parroquias Urbanas en el Mapa de Riobamba	Ingreso
Req03	Ingreso de puntos de atención existentes en el mapa de Riobamba	Ingreso
Req04	Visualización de las posibles nuevas ubicaciones de puntos de atención	Consulta
Req05	Modificar estado de los puntos de atención generados por el algoritmo genético	Modificación
Req06	Visualización de la afluencia de usuarios a los puntos de atención	Consulta
Req07	Generar archivo Json, con las ubicaciones de los puntos de atención	Consulta

Fuente: elaboración propia

4.1.5.2. Requerimientos tipo no funcionales

En la tabla 7, se lista los requerimientos no funcionales, relacionados con las funcionalidades del sistema para el momento de su implantación.

Tabla 5. Requerimientos no funcionales

Requerimiento
El sistema se encuentre disponible después de su instalación local en los dispositivos que cumplan con los requerimientos mínimos, para su funcionamiento.
El diseño de interfaces de usuario sea agradable.
El rendimiento sea bueno, en el procesamiento de los cálculos del algoritmo genético.
Mantenimiento sea fácil de realizarlo.

Fuente: elaboración propia

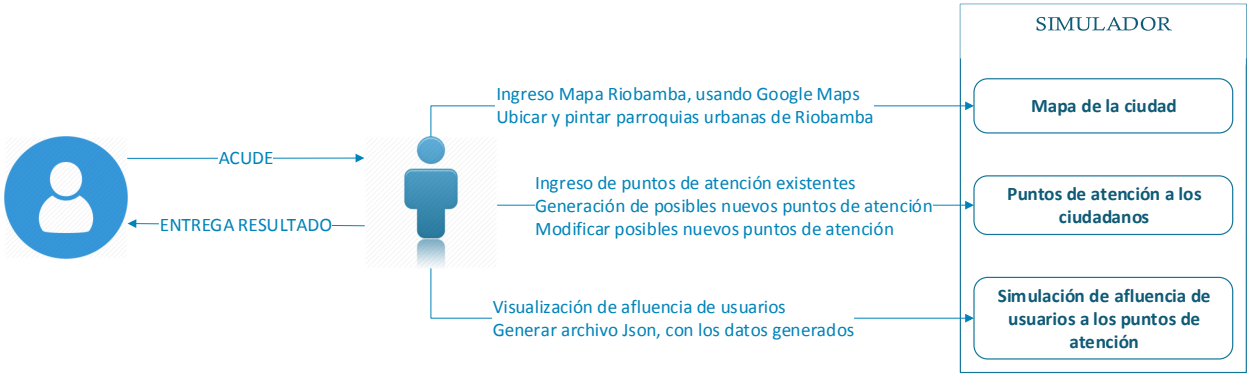
4.1.6. Diseño. Planeamiento Sprint

En el proceso del desarrollo de software, se consideró a la solución con todos sus componentes que lo integran, basados en los requerimientos funcionales y no funciones, detallados en la sección 4.1.5.1 y 4.1.5.2, generando los siguientes diseños.

4.1.6.1. Diagrama de procesos

Por medio de la indagación y observación en las reuniones de trabajo, se lo pudo representar en un diagrama para representar los flujos de tareas e interacciones, adaptados a la realidad. Se lo detalla en la figura 20.

Figura 13. Diagrama de procesos

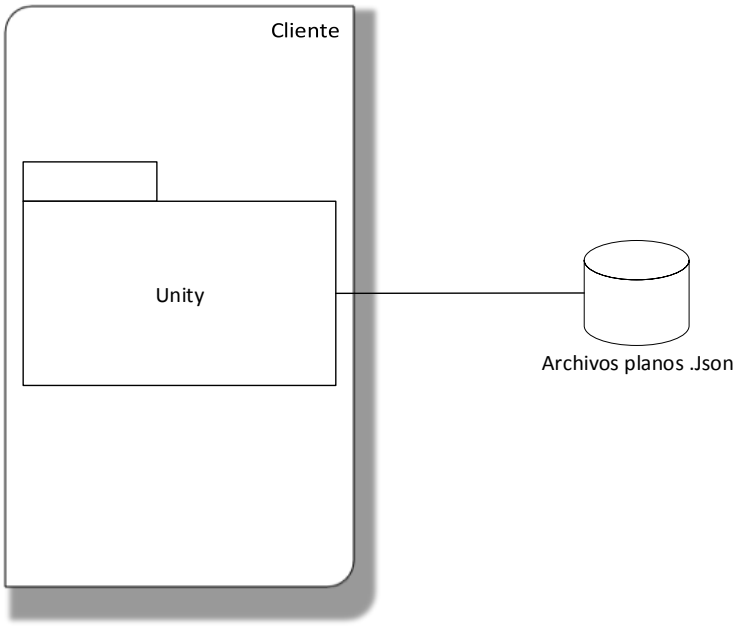


Fuente: elaboración propia

4.1.6.2. Diseño arquitectura del sistema

La aplicación software será instalada localmente en los dispositivos deseados, se ha diseñado una arquitectura "Cliente – Servidor", teniendo como cliente a Unity 3d, los datos de la simulación serán almacenados en un archivo plano tipo Json, guardados localmente en el directorio específico que utiliza la aplicación desarrollada en unity 3d. La figura 10, muestra el diagrama de diseño de la arquitectura.

Figura 14. Arquitectura del sistema

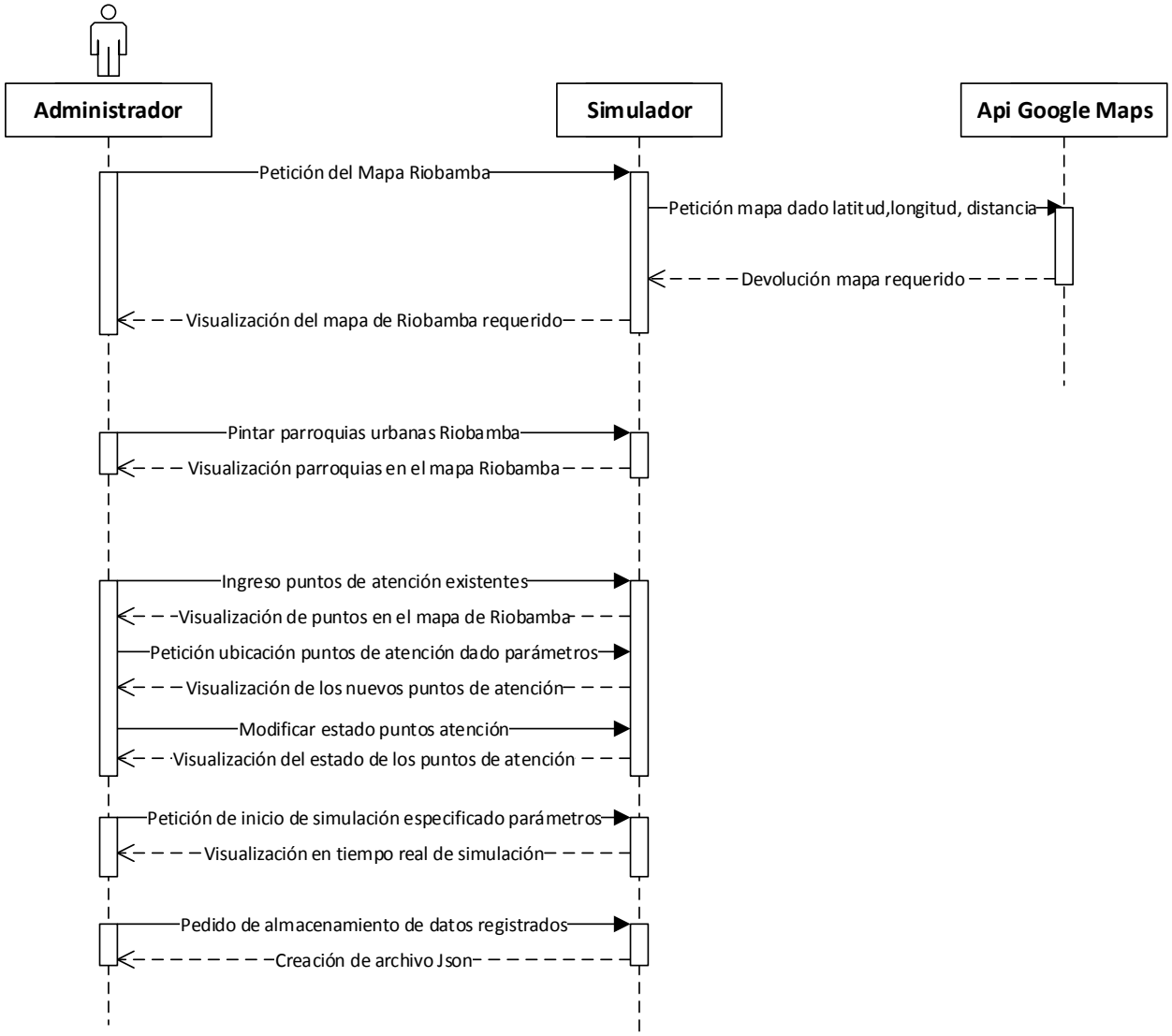


Fuente: elaboración propia

4.1.6.3. Diagrama de secuencia

Comprende el proceso de interacción entre el usuario final y los distintos módulos de la aplicación software, dando la facilidad que, al momento de la programación, se tome en consideración las entradas y salidas de las interacciones propuestas. Se detalla en la figura 11.

Figura 15. Diagrama de secuencia



Fuente: elaboración propia

4.1.7. Desarrollo. Algoritmo genético

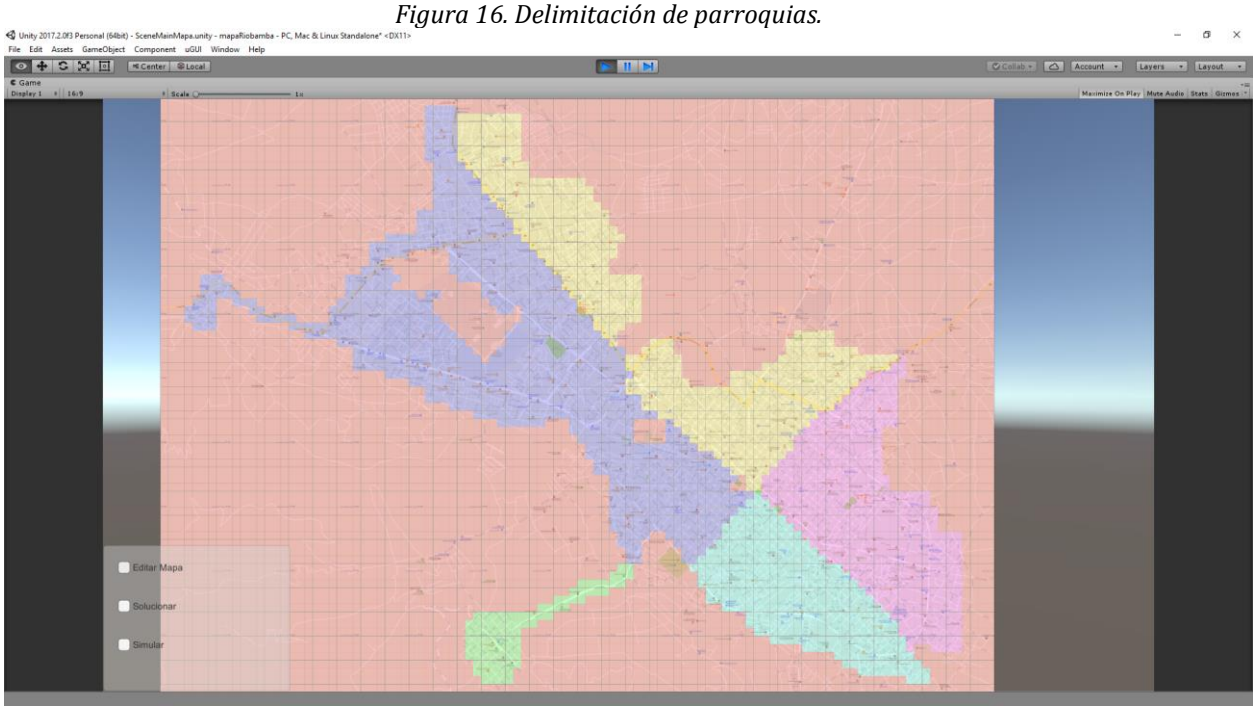
Codificación Genética

El cálculo del total de la longitud del genoma que se usa depende primeramente de la identificación de cada parroquia urbana de Riobamba.

Lizarzaburu: color púrpura id:1. **Velasco**: color amarillo, id:2. **Veloz**: color celeste, id:3. **Maldonado**: color morado claro, id:4. **Yaruquies**: color verde agua, id: 5.

Dentro de cada parroquia, mediante una cuadriculación del Mapa de Riobamba con su identificación de posición en una matriz general, se determina el estado de las casillas habilitadas y deshabilitadas mediante colores que pertenecen a cada parroquia. Como se muestra en la figura 12.

Figura 16. Delimitación de parroquias.



Fuente: elaboración propia

De lo que genera vectores con celdas donde se establecen la siguiente codificación:

0: celda vacía

1: celda con punto de atención

De esta forma se lo pasa como parámetro a la función del algoritmo genético la longitud de cada vector de cada parroquia existente, teniendo en cuenta que el número de puntos de atención es una cifra parametrizable, que se lo ingresa por interfaz de la aplicación. La figura 13 muestra una representación gráfica de un vector asociativo de una parroquia, cada posición del vector incluye si existe o no un punto de atención y la posición en la matriz del mapa.

Figura 17. Vector de la parroquia Yaruquies

YARUQUIES

Punto Atención	0	1	0	0	1	1
Posición	(10, 11)	(10, 12)	(10, 13)	(14, 9)	(14, 10)	(14, 11)

Punto Atención
0: celda vacía
1: celda con punto de atención

Posición
(x, y)
x: fila
y: columna

Fuente: elaboración propia

Función Evaluación

Para la función objetivo que se usa en el algoritmo genético, se determina un criterio de distancia de separación entre puntos de atención que se parametriza en interfaz de la aplicación, la fórmula se determina mediante las posiciones de dos ubicaciones en dos dimensiones.

$P1 = (x1, y1)$

$P2 = (x2, y2)$

$d^2 = (x2 - x1)^2 + (y2 - y1)^2$

Si la distancia resultante del cálculo es menor al criterio de distancia de separación, se lo considera como una colisión, en cada vector de cada parroquia se realiza un barrido de todo 1 (número uno) encontrado y se lo compara con todos los 1 (número uno) restantes, para contabilizar el número total de colisiones detectadas. La solución óptima es si el total de colisiones es igual a cero.

Operadores Básicos

Selección

El algoritmo genético usado en la aplicación software realiza una selección por probabilidad randómica, las colisiones que se calcula en la función objetivo, se la denomina puntuación la cual se la convierte en porcentaje mediante una interpolación linear entre 0.05 y 0.98.

Figura 18. Condiciones de selección

```
public static float LinearInterpolation(float x, float x1,float y1,float x2,float y2)
{
	return y1+((x-x1)*(y2-y1)/(x2-x1));
}


float percent = LinearInterpolation(cScore,minScore,0.98f,maxScore,0.05f);
float sqrPercent = Mathf.FloorToInt((percent*percent)*100);

int rnd = Random.Range(0,100);
if(rnd<sqrPercent)
{
	index = i;
	break;
}
```

Fuente: elaboración propia

Teniendo en cuenta lo siguiente:

- Menor número de colisiones el porcentaje de puntuación es mayor.
- Mayor número de colisiones el porcentaje de puntuación es menor.

De esta forma se escogen pares de individuos de la población que cumplan con las condiciones. Para su posterior reproducción. Este tipo de selección se lo denomina selección por rueda de ruleta, todos los individuos tienen la misma probabilidad de ser seleccionados. Como se muestra en la tabla 7.

Tabla 6. Selección por rueda de ruleta

POBLACION	Porcentaje %	Random %	Elección
individuo 1	80	62	✓
individuo 2	15	10	✓
individuo 3	20	50	✗
individuo 4	40	25	✓
individuo 5	50	49	✓
individuo 100	90	95	✗

Fuente: elaboración propia

Cruce o recombinación

En la aplicación software, el algoritmo genético realiza el cruce o recombinación entre pares de individuos seleccionados mediante probabilidades de rueda de ruleta.

Los individuos padres se los divide en dos partes iguales, formando 4 mitades. Padre 1 mitad de arriba y mitad de abajo. Padre 2 mitad de arriba y mitad de abajo. De estas 4 mitades se saca la información útil, todos los edificios con su ubicación ya asignados. De estas 4 mitades generadas se realiza los cruces dando como resultado 4 nuevos hijos, de la siguiente manera.

Hijo 1: Padre1 mitad de arriba + Padre2 mitad de abajo.

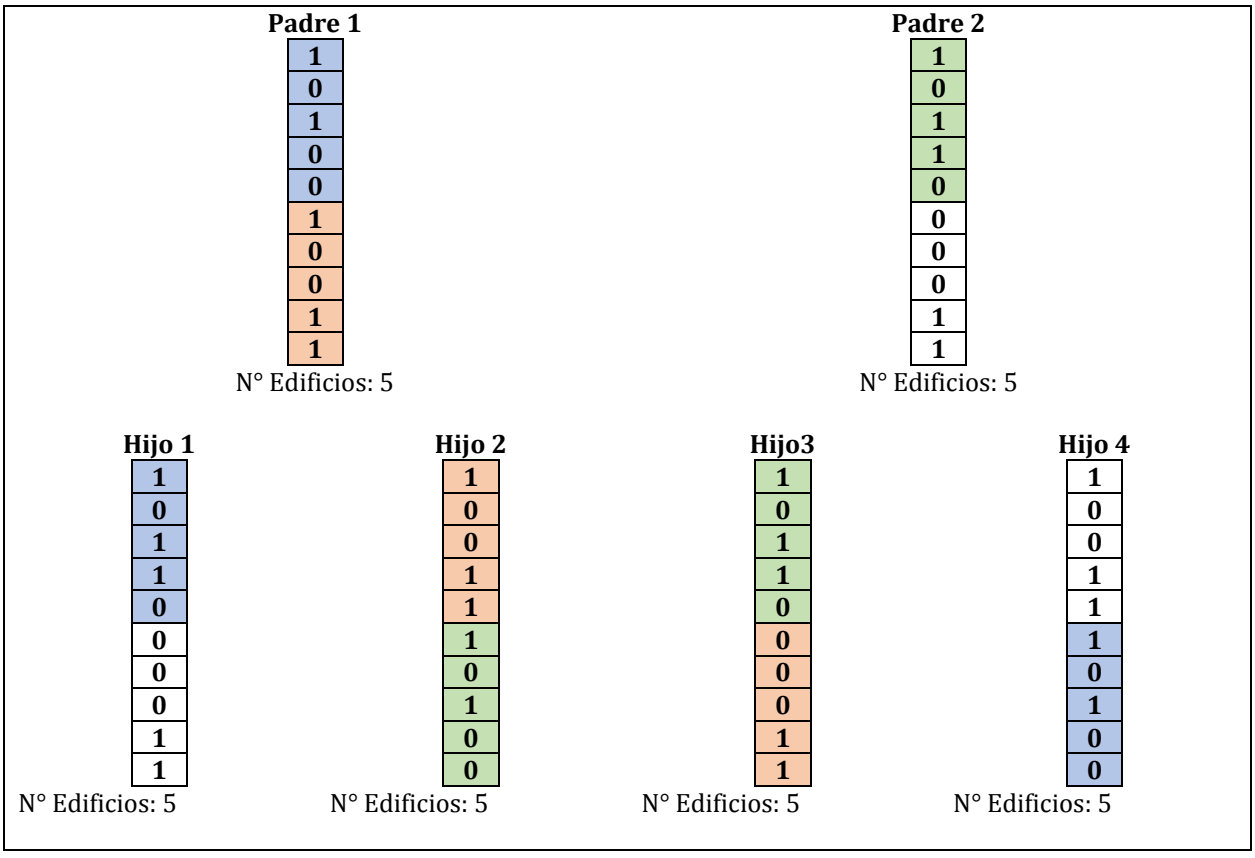
Hijo 2: Padre1 mitad de abajo + Padre2 mitad de arriba.

Hijo 3: Padre2 mitad de arriba + Padre1 mitad de abajo.

Hijo 4: Padre2 mitad de abajo + Padre1 mitad de arriba.

Todos los cruces que se realizan, siempre tienen en cuenta el total de edificios, no se puede dejar más o menos edificios de los preestablecidos, en cada hijo, la mitad de abajo es la parte donde se controla y se realizan las compensaciones si el total de edificios no coinciden. En la figura 15, se muestra gráficamente los cruces y la generación de los 4 hijos por cada par de padres seleccionados.

Figura 19. Cruces y generación de hijos.



Fuente: elaboración propia

Mutación

En la aplicación software, el algoritmo genético realiza la etapa de mutación a cada uno de los hijos, la mutación es adaptativa, se aplica a este tipo de problema. En cada hijo se comprueba todos sus elementos la existencia o no de un edificio en el vector asociativo; al encontrar un edificio, se realiza un barrido del vector para encontrar una colisión con otro edificio que no cumpla con la condición de distancia. Al encontrar la colisión se realiza un porcentaje randómico entre 0 y 100 de la siguiente forma:

```
if(Random.Range(0,100) < dna.mutationPercent)
```

Si arroja verdadero el resultado, los vectores posición tanto del edificio uno y del edificio dos, se los prepara, y mediante una probabilidad del 50% se puede realizar una las siguientes operaciones con el cálculo del vector posición relativa:

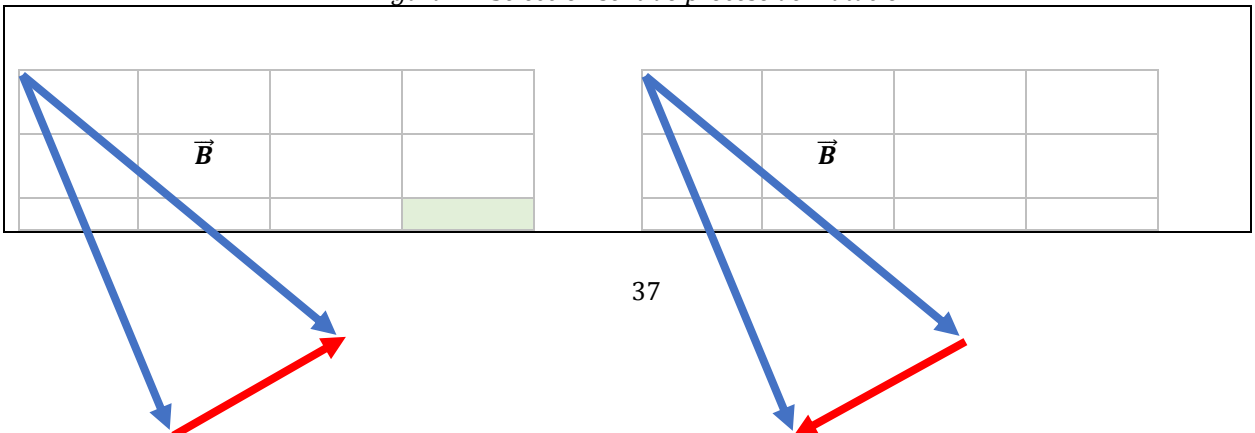
Vector edificio uno : $\vec{A}$

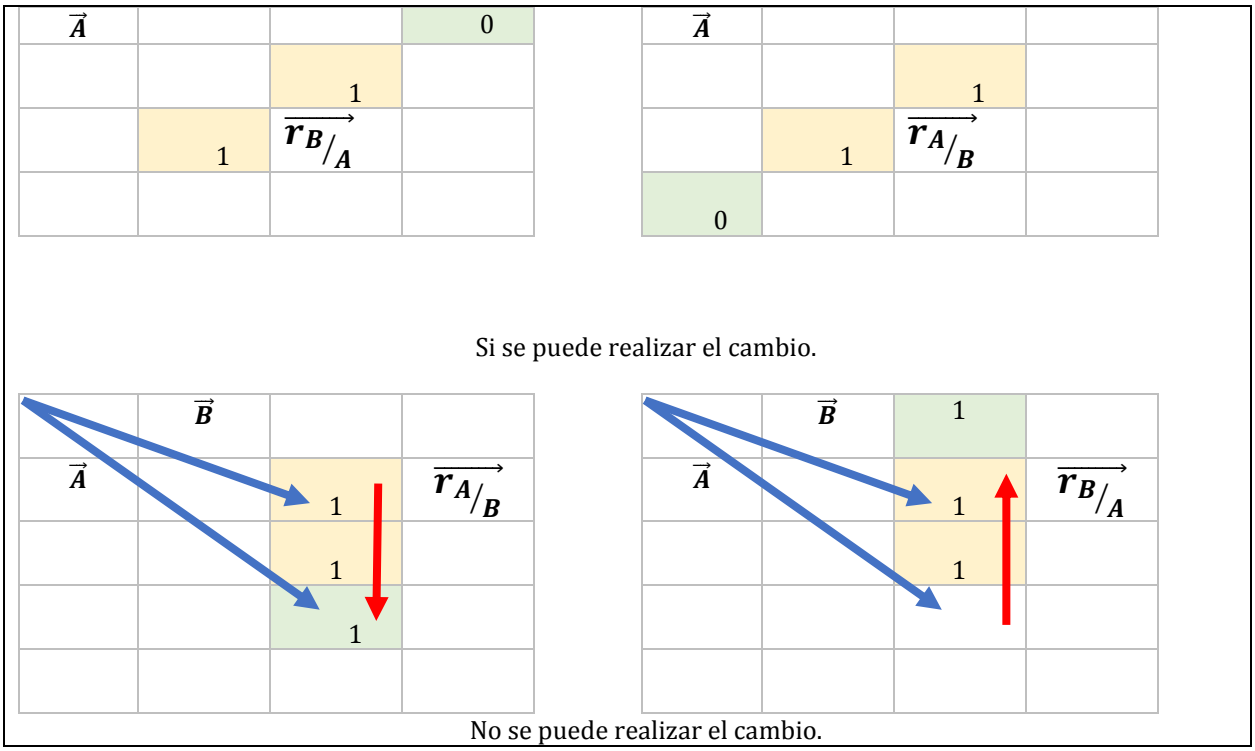
Vector edificio uno : $\vec{B}$

- Opción 1: $\overrightarrow{r_{A/B}} = \vec{A} - \vec{B}$
- Opción 2: $\overrightarrow{r_{B/A}} = \vec{B} - \vec{A}$

Cualquiera de los dos vectores de posición relativa arroja el sentido de hacia dónde debe moverse el edificio en colisión. Si la nueva posición ya contiene un edificio, no se realiza el movimiento para completar la mutación. Y se continúa con el resto de elementos en el vector del individuo a mutar. En la figura 16 se demuestra mediante una matriz el procedimiento de mutación.

Figura 20. Selección sentido proceso de mutación.





Fuente: elaboración propia

Cabe resaltar que el proceso de mutación aplicado, no es como los convencionalmente aplicados en un algoritmo genético simple donde solamente se cambia el valor de un cromosoma de 0 a 1 o viceversa. La mutación aplicada maneja primordialmente ubicaciones en una matriz general generada para el mapa de Riobamba, en el cual se maneja vectores de posición y utilizando conceptos de física vectorial se calcula el vector de posición relativa para determinar el sentido de la nueva ubicación del edificio en conflicto. Este proceso al hacer un barrido de todo el vector de todos los individuos, detectando las colisiones entre edificios, y encontrando la nueva posible ubicación, no solamente realiza la mutación, además aporta a la formación de individuos con menos colisiones, los cuales aportan significativamente a encontrar la solución del mas apto.

Parámetros que controlan el desempeño del algoritmo genético

Tamaño de la población

La aplicación software maneja poblaciones de 100 individuos, ya que se elige un número múltiplo de 4; debido a que se generan 4 hijos por cada dos progenitores seleccionados. Este tamaño de población es el adecuado para no saturar los recursos hardware del computador, e igual logra encontrar la solución óptima con el individuo más apto para la solución del algoritmo genético.

Probabilidad de cruce o recombinación

Los cruces se lo realizan entre dos individuos seleccionados de la población mediante probabilidades de rueda de ruleta, como se lo especifica en la sección 3.1.1.4. De la población de 100 individuos en cada iteración, se selecciona solamente 50 individuos los cuales forman pares, por orden de llegada sin ningún criterio en específico de organización de pares. En total se forman 25 pares que se convierten en los progenitores, de cada par se genera 4 hijos, los cuales conforman la nueva generación de 100 individuos nuevamente.

Probabilidad de mutación

El proceso de mutación es adaptado para este tipo de problema, donde se asiste a la ubicación de los edificios como nuevos puntos de atención en la matriz que se generó en el mapa de la ciudad. Al término del proceso de mutación los individuos son cada vez más aptos para ser evaluados por la función objetivo, la población se mejora con cada iteración. La mutación para la resolución de este problema se convierte en un proceso complementario fundamental para la consecución de la solución del algoritmo genético.

Número de Generaciones

El algoritmo genético al encontrar una solución óptima termina su proceso, pero de no ser así, realiza 40000 (cuarenta mil) iteraciones, mostrando en pantalla los individuos de dicha iteración, sin esta restricción del número de iteraciones máxima el algoritmo genético podría involucrarse en un bucle infinito, causando sobrecargas en los recursos hardware.

Google Maps

En el desarrollo de la aplicación software se necesita el mapa de la ciudad de Riobamba, y para esta necesidad se requirió Google Maps, en primera instancia se necesitó conocer los elementos de su api, integrados con el leguaje de desarrollo C#.

Los primeros pasos fue personalizar cada mapa como piezas individuales que conforman el mapa de Riobamba total (Svennerberg, 2010), determinando las líneas de código necesarias para la creación, declaración de los parámetros de tamaño y zoom adecuados, como se lo muestra en la Figura 17., Figura 18., Figura 19.

Figura 21. Variables iniciales de distancia, zoom, y coordenadas referenciales del mapa

```
public GameObject mapPrefab; //objeto mapa
  private float distanceX = 640f;
  private float distanceY = 640f;
  private float zoom = 17f;
  private float latitudCentro = -1.67098f;   //centro riombamba
private float longitudCentro = -78.64712f; //centro riombamba
```

Fuente: elaboración propia

Figura 22. Método para creación del mapa, coordenadas de latitud, longitud y zoom

```
private void CreateMap(float lat,float lng, Vector3 pos) {
        GameObject go;
        string url =
"https://maps.googleapis.com/maps/api/staticmap?maptype=road&center="+lat+","+lng+"&zoom="+zoom+"&siz
e="+distanceX+"x"+distanceY+"&key=" + googleMapsApiKey;
        go = Instantiate(mapPrefab.gameObject, pos, Quaternion.identity);
        StartCoroutine(SetMap(url,go));
    }
```

Fuente: elaboración propia

Figura 23. Creación del mapa central.

```
    CreateMap(latitud, longitud, new Vector3(0, 0, 0));//creacion CENTRO
```

Fuente: elaboración propia

Figura 24. Resultado de la creación del mapa central de Riobamba

Fuente: elaboración propia

Consiguientemente de la creación de pieza central del mapa de la ciudad de Riobamba, se determinaron los parámetros calculados de distancia y un factor de compensación entre las piezas, para que la unión de todas las piezas necesarias se agregue una a continuación de la otra en perfecta armonía formando el mapa de Riobamba. El código necesario que se creó para este caso en específico usando los límites de la ciudad de Riobamba requeridos, se muestra en la figura 21, y su resultado en la figura 22.

Figura 25. Límites del mapa, distancia de cada pieza de mapa y factor compensación

```
private float limiteDerecho = -78.631f;
private float limiteIzquierdo = -78.705f;
private float limiteSuperior = -1.630f;
private float limiteInferior = -1.690f;

var result = GoogleMapsAPI.GetBounds(Coordinates, (int)zoom, (int)distanceX,
(int)distanceY);
float distancia = (float)(result.NorthEast.X - longitud);
float factor = 5.0f;
```

Fuente: elaboración propia

Figura 26. Resultado de la unión de las piezas del mapa

Fuente: elaboración propia

Y finalmente se desarrolló, el código restante para crear todas las piezas de mapa de extremo a extremo y poder tener en perfectas condiciones el mapa de Riobamba figura 23 y figura 24, que nos servirá de base para la aplicación del algoritmo genético, figura 25.

Figura 27. Código necesario para unir piezas de mapa hacia el norte y sur

```
private void drawUpMap(int x, float distancia, float longitud, float latitud, float limite,
float factor)
{
    int i = 10;

    while(latitud < limite)
    {
        latitud = latitud + factor * distancia;
        CreateMap(latitud, longitud, new Vector3(x, i, 0));
        i+=10;
    }
}

private void drawDownMap(int x, float distancia, float longitud, float latitud, float limite,
float factor)
{
    int i = -10;
    while (latitud > limite)
    {
        latitud = latitud - factor * distancia;
        CreateMap(latitud, longitud, new Vector3(x, i, 0));
        i-=10;
    }
}
```

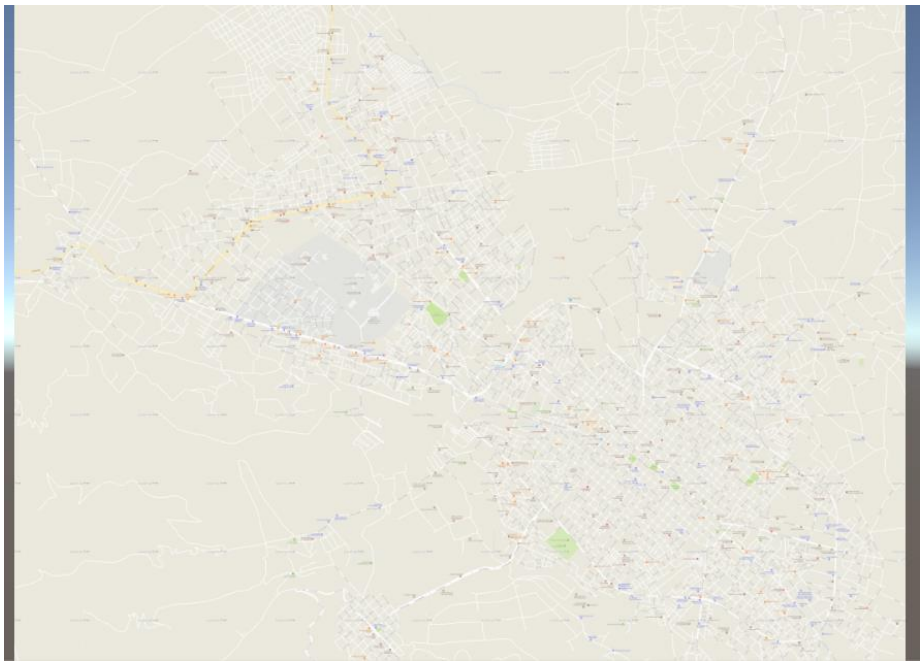
Fuente: elaboración propia

Figura 28. Código para crear mapas hacia el este y oeste

```
CreateMap(latitud, longitud, new Vector3(0, 0, 0));//creacion CENTRO
drawUpMap(0, distancia, longitud, latitud, limiteSuperior, factor);
latitud = latitudCentro;
longitud = longitudCentro;
drawDownMap(0, distancia, longitud, latitud, limiteInferior, factor);
//centro - derecha ***********************************************************
latitud = latitudCentro;
longitud = longitudCentro;
i = 10;
while(longitud < limiteDerecho){
    longitud = (longitud + factor * distancia);
    CreateMap(latitud, longitud , new Vector3(i,0,0));
    drawUpMap(i, distancia, longitud, latitud, limiteSuperior,factor);
    drawDownMap(i, distancia, longitud, latitud, limiteInferior, factor);
    i+=10;
}
//centro - izquierda **********************************************************
latitud = latitudCentro;
longitud = longitudCentro;
i = -10;
while (longitud > limiteIzquierdo){
    longitud = (longitud - factor * distancia);
    CreateMap(latitud, longitud, new Vector3(i, 0, 0));
    drawUpMap(i, distancia, longitud, latitud, limiteSuperior, factor);
    drawDownMap(i, distancia, longitud, latitud, limiteInferior, factor);
    i-=10;
}
```

Fuente: elaboración propia

Figura 29. Resultado final mapa de Riobamba completamente unido



Fuente: elaboración propia

4.1.8. Desarrollo. Ejecución Sprint

Al realizar los diseños necesarios, se procedió al desarrollo de la aplicación, empleando el marco de trabajo antes mencionada SCRUM, creando dos sprints con tareas subsanando las necesidades descritas en las historias de usuario, especificado en el ANEXO A, estableciendo tiempos adecuados mediante aplicación de posible esfuerzo en cada tarea, basados en experiencias previas de desarrollos.

SPRINT 1

Tiempo estimado: 2 semanas laborables = 10 días

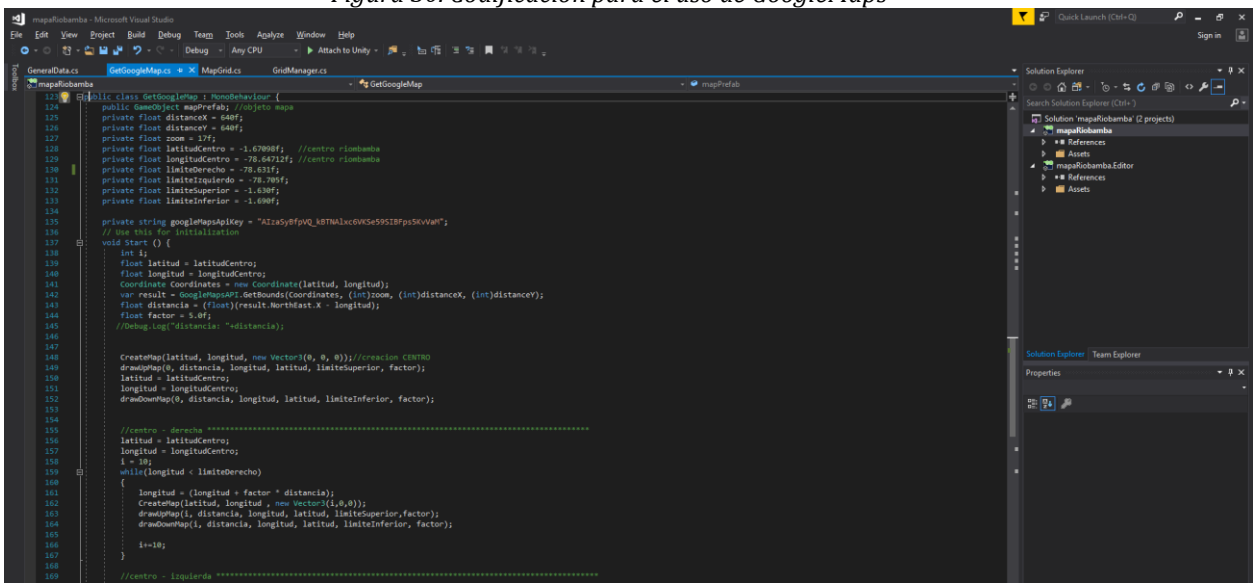
Tareas:

- **Tarea 1.** Visualizar el mapa urbano de Riobamba cuadriculado, usando el api de google Maps en unity3D.
- **Tarea 2.** Mediante clics en la cuadricula generada, delimitar pintando de distinto color las parroquias urbanas de la ciudad de Riobamba.
- **Tarea 3.** Mediante clics ubicar los puntos de atención existentes en la cuadrícula del mapa de Riobamba generado.
- **Tarea 4.** Parametrizar datos de entrada necesarios para la aplicación del algoritmo genético, por medio de interfaz gráfica.
- **Tarea 5.** Codificación del algoritmo genético, para poder visualizar en la cuadrícula del mapa generado, la solución de las posibles ubicaciones de los nuevos puntos de atención.

Plan codificación: Por cada tarea planificada, se presenta extractos de código.

Sprint 1 – Tarea 1

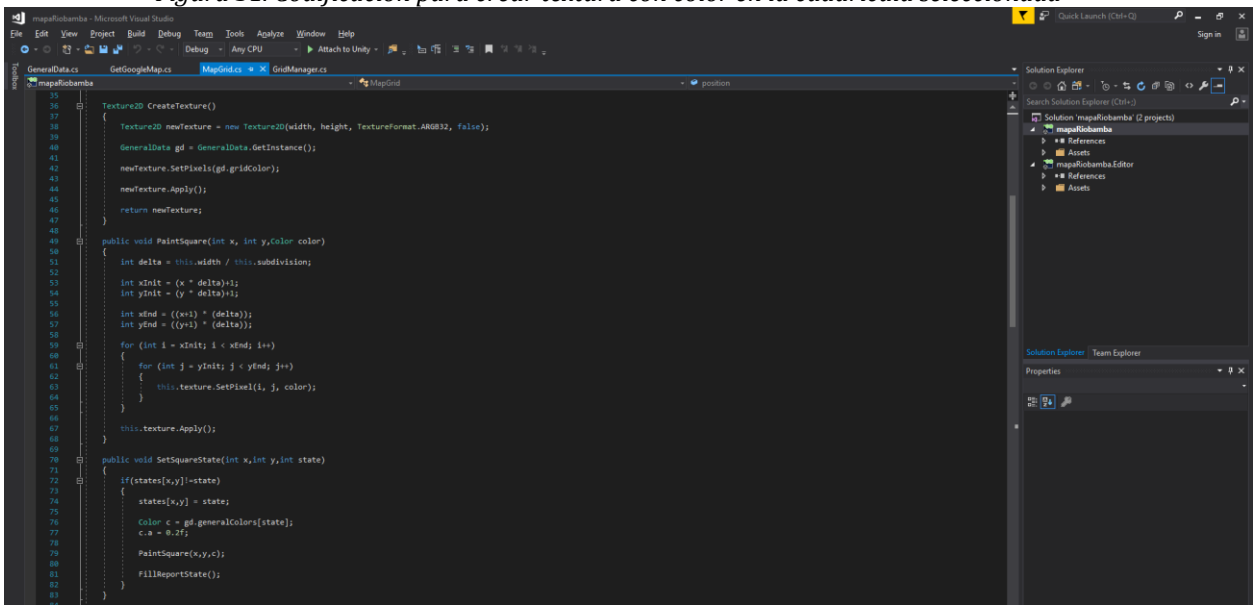
Figura 30. Codificación para el uso de GoogleMaps



Fuente: elaboración propia

Sprint 1 – Tarea 2

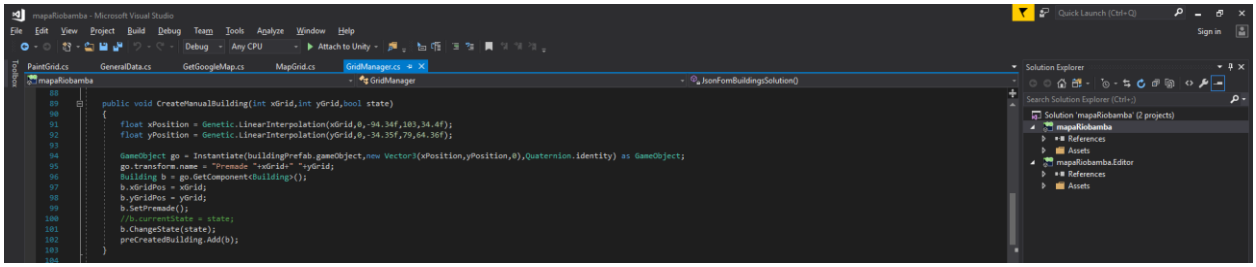
Figura 31. Codificación para crear textura con color en la cuadricula seleccionada



Fuente: elaboración propia

Sprint 1 – Tarea 3

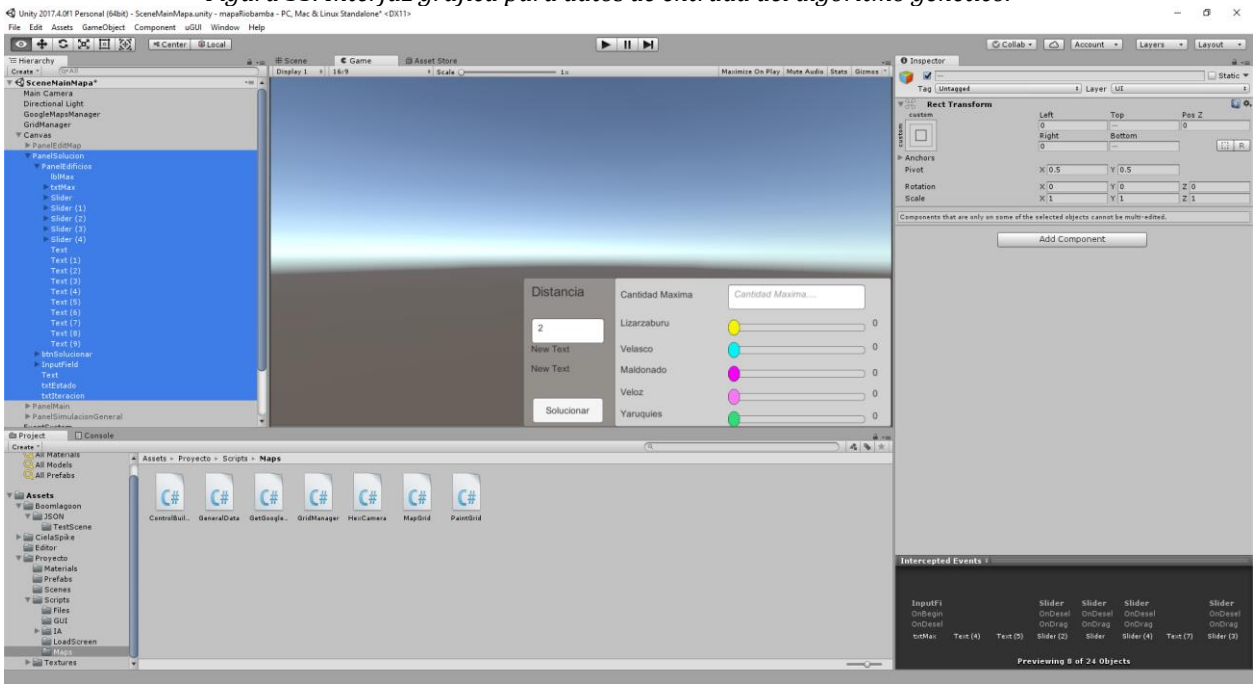
Figura 32. Codificación para la ubicación manual de puntos de atención.



Fuente: elaboración propia

Sprint 1 – Tarea 4

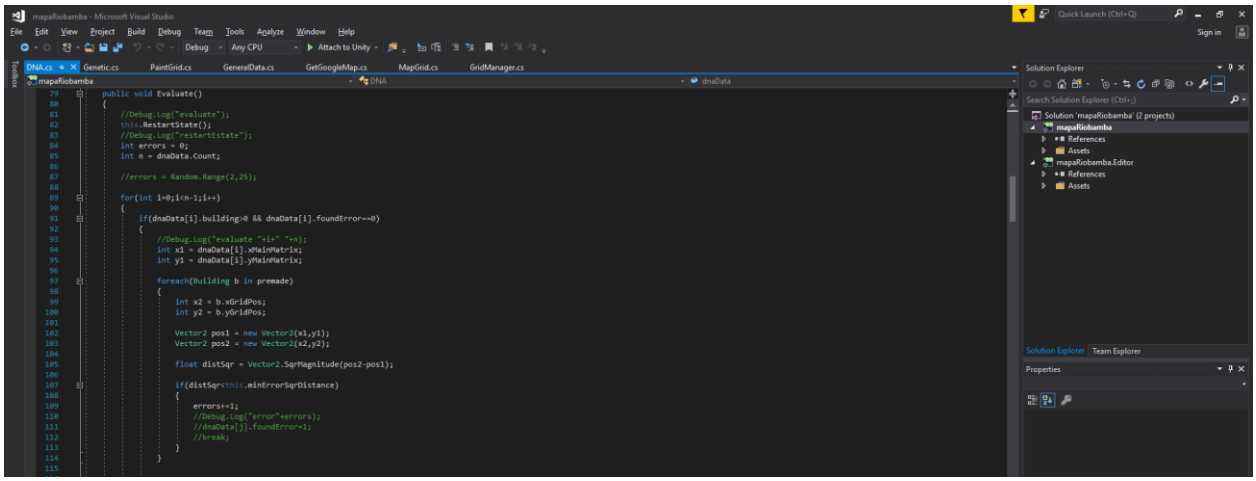
Figura 33. Interfaz gráfica para datos de entrada del algoritmo genético.



Fuente: elaboración propia

Sprint 1 – Tarea 5

Figura 34. Codificación de los métodos del algoritmo genético.



Fuente: elaboración propia

Prueba tipo funcional:

En el transcurso del desarrollo, las actividades basadas en cada historia de usuario, se verificó y validó para conseguir el requerimiento, para esto se manejó pruebas funcionales, las mismas que ayudaron para detectar errores y controlarlos. En la tabla 7, se presenta un formato de una prueba funcional.

Tabla 7. Prueba funcional Sprint 1

PRUEBA FUNCIONAL	
Número Prueba: 1	**Historia de Usuario 1**
Nombre de la Prueba: Petición Mapa Riobamba	
Descripción: El usuario, al seleccionar la opción del menú “Editar mapa”, se le brinda la opción de cargar el mapa urbano de la ciudad de Riobamba, dando clic en el botón “Cargar”, mostrando en pantalla el mapa deseado, usando el api de google Maps.	
Condiciones de ejecución: Ninguna	
Entrada: - El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción “Editar mapa”. - El sistema verificará la conexión a google Maps, ubicando latitud y longitud deseada y mostrará en pantalla el mapa de la ciudad de Riobamba. - El sistema creará la cuadricula respectiva sobre la extensión del mapa.	

Resultado esperado: Visualizar el mapa concatenado de la ciudad de Riobamba delimitado sus extremos de latitud y longitud
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

En cada prueba funcional, se detalla la descripción de la prueba, las entradas, el resultado esperado y la evaluación de la prueba, de forma que se puede tomar acciones correctivas de los errores detectados en el proceso del desarrollo del software. En el APÉNDICE B se encuentra todas las pruebas funcionales realizadas. Para el sprint 1, se realizaron 4 pruebas funcionales # 1-2-3-4 pertenecientes a la historia de usuario #1-2-3-4.

Prueba tipo unitaria de código:

A continuación, se presenta las dificultades que se presentaron el momento de probar los resultados de la codificación en el Sprint 1.

Tabla 8. Prueba unitaria de codificación 1

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 1**
Dificultad presentada: El momento de ejecutar el programa, no se generaba el mapa de Riobamba, solamente quedaba en pantalla la cuadrícula.	
Solución: Para utilizar el api de google maps se necesita un "key", esta llave no debe usarse en varios proyectos, puesto que las peticiones a google maps con la misma llave se saturan las peticiones a los servidores de google. Al cambiar la llave, volvió a generarse el mapa de la ciudad de Riobamba.	

Fuente: elaboración propia

Tabla 9. Prueba unitaria de codificación 2

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 2**
Dificultad presentada: El mapa de Riobamba, está formado por sub partes de forma cuadrada, concatenadas entre sí. El momento de ejecución del programa los cuadrados quedaban con espacios vacíos, o a su vez se sobreponían las imágenes del mapa.	
Solución: El zoom de la visualización del mapa, se lo determino en 17f, y la distancia de cada cuadrado se lo definió en 640f tanto en el eje horizontal y vertical.	

Fuente: elaboración propia

Tabla 10. Prueba unitaria de codificación 3

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 3**

Dificultad presentada: Sobre cada cuadrado de 640 x 640 con un segmento de mapa de Riobamba, se creó una matriz. Al momento de generar visualmente la solución del algoritmo genético en tiempo real, tomaba demasiado tiempo, muchas veces un par de días, usando muchos recursos hardware del computador usado.
Solución: El grado de granularidad de la matriz sobre cada cuadrado de mapa de Riobamba, se lo redujo a 8 filas por 8 columnas, de esta forma se optimizó el tiempo a horas de procesamiento de la solución del algoritmo genético. Salvaguardando el hardware del equipo de cómputo.

Fuente: elaboración propia

SPRINT 2

Tiempo estimado: 2 semanas laborables + dos días = 12 días

Tareas:

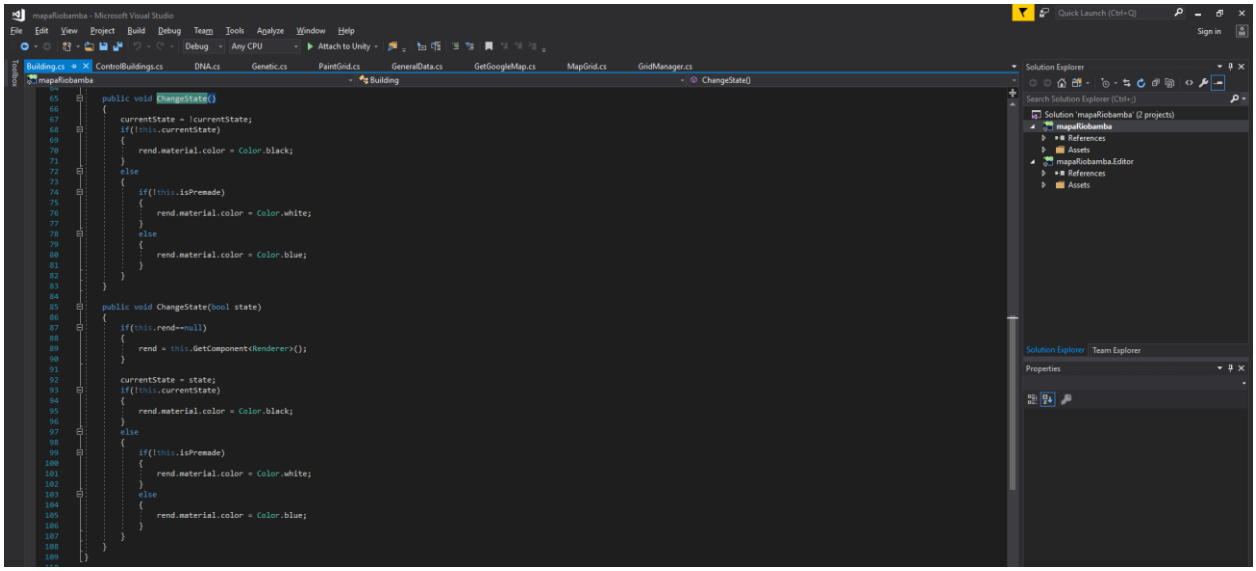
- **Tarea 1.** Mediante un clic sobre un punto de atención generado, permitir cambiar el estado del mismo, de habilitado a deshabilitado y viceversa.
- **Tarea 2.** Parametrizar datos necesarios para la simulación de afluencia, mediante interfaz gráfica.
- **Tarea 3.** Codificar y mostrar en tiempo real la simulación de afluencia a los puntos de atención generados.
- **Tarea 4.** Generar un archivo Json con los datos de ubicación y afluencia de los puntos de atención.
- **Tarea 5.** Convertir los datos del archivo Json, a una tabla en una hoja de cálculo Excel.

Plan codificación:

Por cada tarea planificada, se presenta extractos de código.

Sprint 2 – Tarea 1

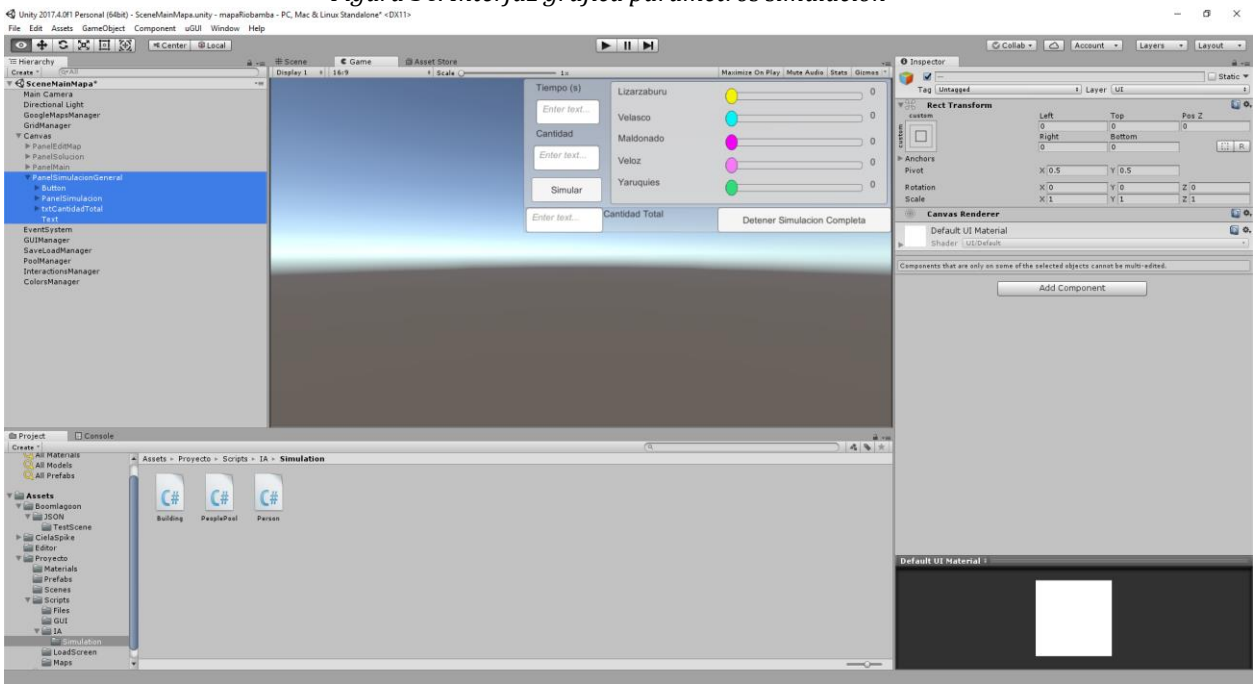
Figura 35. Codificación para cambio de estado de un edificio



Fuente: elaboración propia

Sprint 2 – Tarea 2

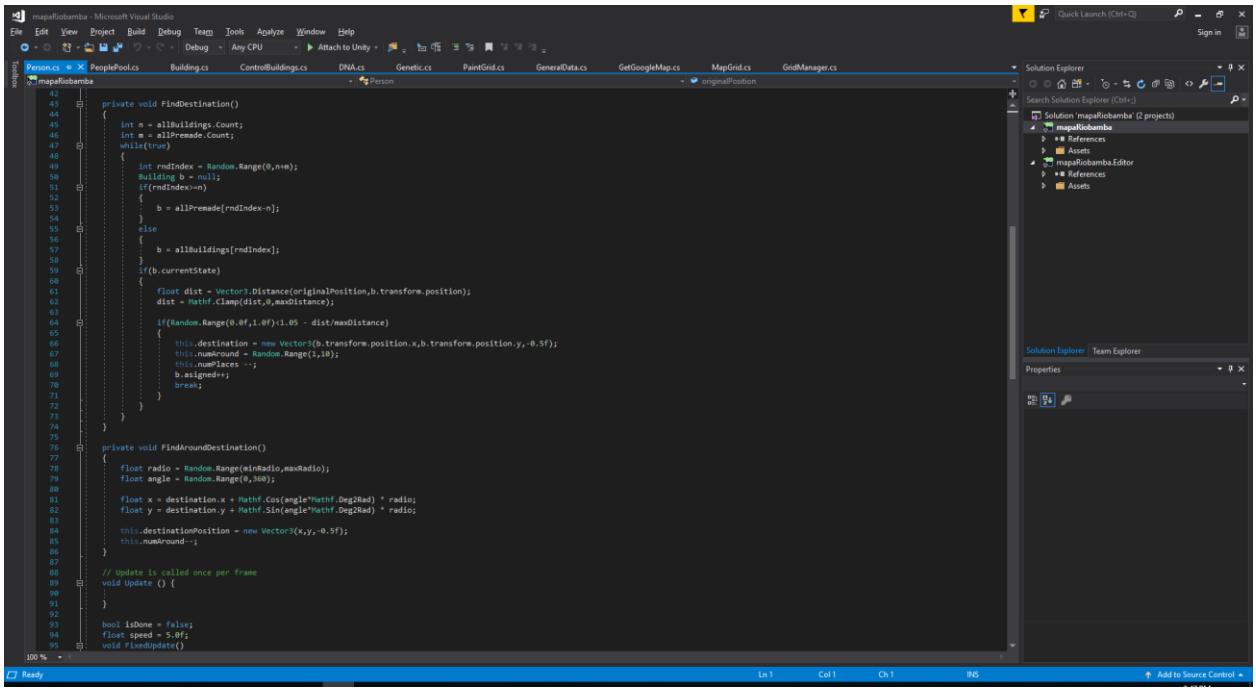
Figura 36. Interfaz gráfica parámetros simulación



Fuente: elaboración propia

Sprint 2 – Tarea 3

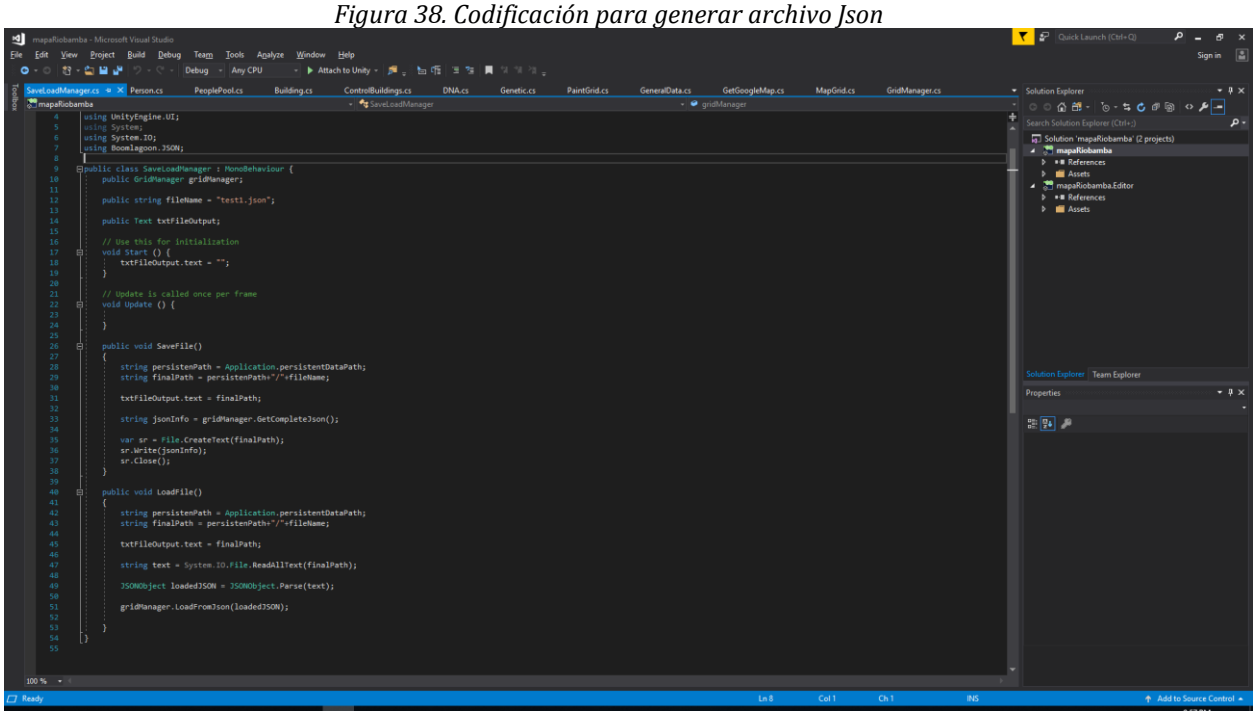
Figura 37. Codificación para el inicio de la simulación



Fuente: elaboración propia

Sprint 2 – Tarea 4

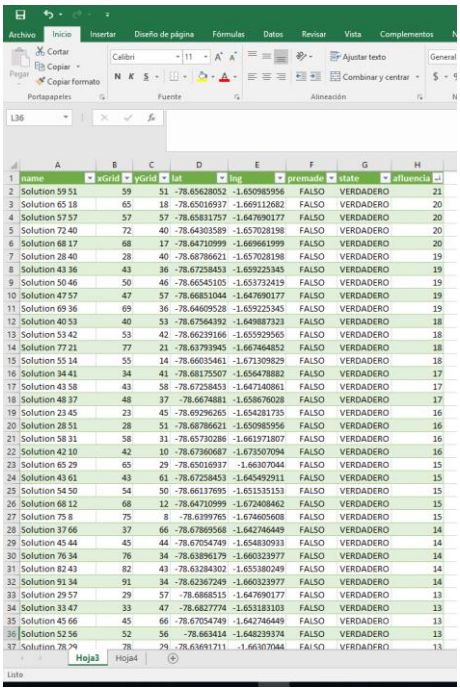
Figura 38. Codificación para generar archivo Json



Fuente: elaboración propia

Sprint 2 – Tarea 5

Figura 39. Hoja de cálculo Excel



	name	xGrid	yGrid	lat	lng	premade	state	afluencia
2	Solution 59 51	59	51	-78.65628052	-1.650985956	FALSO	VERDADERO	21
3	Solution 65 18	65	18	-78.65016937	-1.669112682	FALSO	VERDADERO	20
4	Solution 57 57	57	57	-78.65831757	-1.647690177	FALSO	VERDADERO	20
5	Solution 72 40	72	40	-78.64303589	-1.657028198	FALSO	VERDADERO	20
6	Solution 68 17	68	17	-78.64710999	-1.669661999	FALSO	VERDADERO	20
7	Solution 28 40	28	40	-78.68786621	-1.657028198	FALSO	VERDADERO	19
8	Solution 43 36	43	36	-78.67258453	-1.659225345	FALSO	VERDADERO	19
9	Solution 50 46	50	46	-78.66545105	-1.653732419	FALSO	VERDADERO	19
10	Solution 47 57	47	57	-78.66851044	-1.647690177	FALSO	VERDADERO	19
11	Solution 69 36	69	36	-78.64609528	-1.659225345	FALSO	VERDADERO	19
12	Solution 40 53	40	53	-78.67564392	-1.649887323	FALSO	VERDADERO	18
13	Solution 53 42	53	42	-78.66239166	-1.655929565	FALSO	VERDADERO	18
14	Solution 77 21	77	21	-78.63793945	-1.667464852	FALSO	VERDADERO	18
15	Solution 55 14	55	14	-78.66035461	-1.671309829	FALSO	VERDADERO	18
16	Solution 34 41	34	41	-78.68175507	-1.656478882	FALSO	VERDADERO	17
17	Solution 43 58	43	58	-78.67258453	-1.647140861	FALSO	VERDADERO	17
18	Solution 48 37	48	37	-78.6674881	-1.658676028	FALSO	VERDADERO	17
19	Solution 23 45	23	45	-78.69296265	-1.654281735	FALSO	VERDADERO	16
20	Solution 28 51	28	51	-78.68786621	-1.650985956	FALSO	VERDADERO	16
21	Solution 58 31	58	31	-78.65730286	-1.661971807	FALSO	VERDADERO	16
22	Solution 42 10	42	10	-78.67360687	-1.673507094	FALSO	VERDADERO	16
23	Solution 65 29	65	29	-78.65016937	-1.66307044	FALSO	VERDADERO	15
24	Solution 43 61	43	61	-78.67258453	-1.645492911	FALSO	VERDADERO	15
25	Solution 54 50	54	50	-78.66137695	-1.651535153	FALSO	VERDADERO	15
26	Solution 68 12	68	12	-78.64710999	-1.672408462	FALSO	VERDADERO	15
27	Solution 75 8	75	8	-78.6399765	-1.674605608	FALSO	VERDADERO	15
28	Solution 37 66	37	66	-78.67869568	-1.642746449	FALSO	VERDADERO	14
29	Solution 45 44	45	44	-78.67054749	-1.654830933	FALSO	VERDADERO	14
30	Solution 76 34	76	34	-78.63896179	-1.660323977	FALSO	VERDADERO	14
31	Solution 82 43	82	43	-78.63284302	-1.655380249	FALSO	VERDADERO	14
32	Solution 91 34	91	34	-78.62367249	-1.660323977	FALSO	VERDADERO	14
33	Solution 29 57	29	57	-78.6868515	-1.647690177	FALSO	VERDADERO	13
34	Solution 33 47	33	47	-78.6827774	-1.653183103	FALSO	VERDADERO	13
35	Solution 45 66	45	66	-78.67054749	-1.642746449	FALSO	VERDADERO	13
36	Solution 52 56	52	56	-78.663414	-1.648239374	FALSO	VERDADERO	13
37	Solution 78 29	78	29	-78.63691711	-1.66307044	FALSO	VERDADERO	13

Fuente: elaboración propia

Prueba tipo funcional:

En cada prueba funcional, se detalla la descripción de la prueba, las entradas, el resultado esperado y la evaluación de la prueba, de forma que se puede tomar acciones correctivas de los errores detectados en el proceso del desarrollo del software. En el APÉNDICE B se encuentra todas las pruebas funcionales realizadas. Para el sprint 2, se realizaron 4 pruebas funcionales # 5-6-7-8 pertenecientes a la historia de usuario #5-6-7-8. Una muestra de la prueba funcional se lo visualizará en la tabla 11.

Tabla 11. Prueba funcional Sprint 2

PRUEBA FUNCIONAL	
Número Prueba: 5	**Historia de Usuario 5**
Nombre de la Prueba: Modificar estado de los puntos de atención generados por el algoritmo genético	
Descripción: El usuario, al escoger la opción del menú “Simular”, el sistema le dará la opción de dar clic en los puntos de atención creados por el sistema, para deshabilitarlos y cambiándolos a color negro. El usuario podrá habilitarlos al dar clic nuevamente.	
Condiciones de ejecución: Ninguna	
Entrada:	

- El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción “Simular”. - El usuario dará clic en el punto de atención que se desee deshabilitar, o habilitar de nuevamente. - El sistema cambiará a color negro los puntos de atención deshabilitados, y color blanco los puntos de atención habilitados.
Resultado esperado: Cambiar de estado de habilitado y deshabilitado, de cada uno de los nuevos puntos de atención sugeridos.
Evaluación de la prueba: Prueba satisfactoria.

Fuente: elaboración propia

Prueba tipo unitaria de código:

Tabla 12. Prueba unitaria de codificación 4

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 4**
Dificultad presentada: El momento de iniciar la simulación de afluencia, la memoria ram se saturaba demasiado, debido a que los objetos se creaban según los parámetros inicializados.	
Solución: Para optimizar los recursos de memoria, se creó un “pool” de objetos, los mismos que son reutilizados después de cumplir su función, con esta medida se redujo la creación de nuevos objetos, y se redujo significativamente el uso de memoria ram.	

Fuente: elaboración propia

Tabla 13. Prueba unitaria de codificación 5

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 4**
Dificultad presentada: Generar archivo con el formato adecuado para el reporte con los resultados de la simulación.	
Solución: El formato Json, brinda las facilidades para almacenar en un archivo de texto plano, creando las listas necesarias para guardar las parroquias delimitadas, la solución con la ubicación de los edificios mediante el algoritmo genético, y los resultados de la simulación de afluencia a los puntos ubicados en el mapa.	

Fuente: elaboración propia

Tabla 14. Prueba unitaria de codificación 6

PRUEBA UNITARIA DE CODIFICACIÓN	**Número de Prueba: 4**
Dificultad presentada: Convertir el archivo Json con los datos resultantes a una hoja de cálculo.	

Solución: Micorsoft Excel, proporciona las facilidades para leer archivo en formato Json, en donde se muestra que lista se quiere transformar en una tabla con los datos necesarios para realizar reportes gráficos.

Fuente: elaboración propia

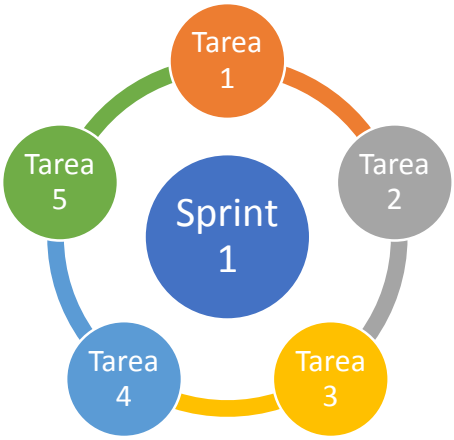
Prueba de Integración

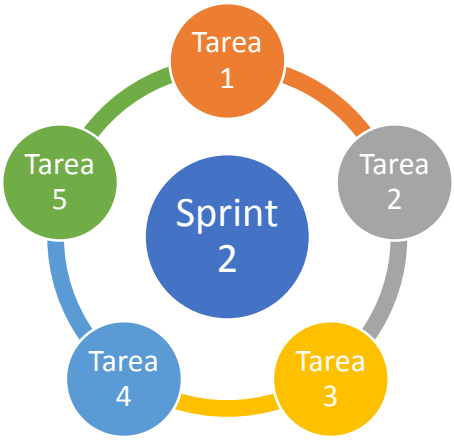
Sprint 1

Tarea 1. Visualizar el mapa urbano de Riobamba cuadriculado, usando el api de google Maps en unity3D.
Tarea 2. Mediante clics en la cuadricula generada, delimitar pintando de distinto color las parroquias urbanas de la ciudad de Riobamba.
Tarea 3. Mediante clics ubicar los puntos de atención existentes en la cuadrícula del mapa de Riobamba generado.
Tarea 4. Parametrizar datos de entrada necesarios para la aplicación del algoritmo genético, por medio de interfaz gráfica.
Tarea 5. Codificación del algoritmo genético, para poder visualizar en la cuadrícula del mapa generado, la solución de las posibles ubicaciones de los nuevos puntos de atención.

Sprint 2

Tarea 1. Mediante un clic sobre un punto de atención generado, permitir cambiar el estado del mismo, de habilitado a deshabilitado y viceversa.
Tarea 2. Parametrizar datos necesarios para la simulación de afluencia, mediante interfaz gráfica.
Tarea 3. Codificar y mostrar en tiempo real la simulación de afluencia a los puntos de atención generados.
Tarea 4. Generar un archivo Json con los datos de ubicación y afluencia de los puntos de atención.
Tarea 5. Convertir los datos del archivo Json, a una tabla en una hoja de cálculo Excel.

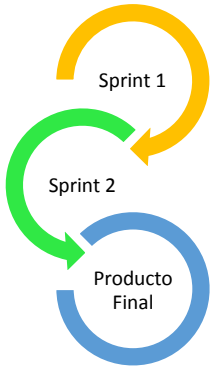




Al término del Sprint 1, la tarea 5 necesito que las tareas 1, 2 ,3, 4 estén finalizadas, permitiendo que el algoritmo genético muestre en pantalla la solución optima.

El inicio del Sprint 2, con la tarea 1 se basa en la solución optima del algoritmo genético. Y consiguientemente el resto de tareas del sprint 2 incrementan el producto final.

En conclusión la unificación del Sprint 1 y Sprint 2, formarón incrementalmente el producto final, resultando en una integracion 100 % eficaz.



4.1.8.1. Estimación de tiempo.

En primera instancia, se definieron las personas con los roles de Scrum necesarios para el desarrollo del Sprint de la aplicación, las funciones de los roles empleados se encuentran detallados en la sección 4.1.2 del presente documento.

A continuación, se muestra la asignación de roles de las personas involucradas en el proyecto, detallado en la tabla 15.

Tabla 15. Roles Scrum

Nombre	Rol
Ing. Jairo González	Product Manager
Ing. Pablo Romero	Scrum Manager
Ing. Víctor Oquendo	Desarrollador

Fuente: elaboración propia

Posterior a la asignación de roles, se procedió a la creación de las historias de usuario, las cuales proveen las actividades para conseguir la generación del proyecto. Se realizaron siete historias de usuario, cada historia tiene relación directa a cada uno de los requerimientos funcionales, en función del tiempo de cada uno de ellos. Cada historia de usuario tiene el formato como se muestra en la tabla 16., mientras que el resto de las historias de usuario se lo detalla en el APÉNDICE A.

Tabla 16. Historia de Usuario 1

	HISTORIA DE USUARIO	**SPRINT 1**
NUMERO: 1	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Petición Mapa Riobamba		
PRIORIDAD: ALTA	**RIESGO:** ALTO	**HORAS ESTIMADAS:** 24
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 24
DESCRIPCION: Se necesita la o las imágenes concatenadas del mapa urbano de la ciudad de Riobamba, detallando el nombre de las calles y puntos referenciales.		
VALIDACION: Se visualiza el mapa de Riobamba en el entorno de Unity, usando el api de Google Maps, con el zoom adecuado para detallar los nombres de calles y referencias.		

Fuente: elaboración propia

Para el análisis del proyecto, referente al tiempo de trabajo estimando y real, según la planificación de cada una de las historias de usuario relacionada a una tarea scrum, se generó una tabla que proporcionó cálculos, tabla 17., y consiguientemente un diagrama denominado "Burndown Chart" detallado en la figura 35.

Tabla 17. Historia Usuario y su tiempo estimado y real

Historia Usuario	Tiempo estimado	DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7	DIA 8	DIA 9	DIA 10
1	24	**8**	**8**	**8**							
2	12				**8**	**2**					
3	12					**6**	**4**				
4	44						**4**	**8**	**8**	**8**	**8**
5	12										
6	44										
7	24										
TOTAL	**172**	8	8	8	8	8	8	8	8	8	8

DIA 11	DIA 12	DIA 13	DIA 14	DIA 15	DIA 16	DIA 17	DIA 18	DIA 19	DIA 20	DIA 21	DIA 22	Tiempo Real Total
												24
												10
												10
8	**4**											48
	4	**8**										12
			8	**8**	**8**	**8**	**8**					40
								8	**8**	**4**		20
8	8	8	8	8	8	8	8	8	8	4	0	**164**

Fuente: elaboración propia

Los cálculos de horas restantes reales y las horas restantes estimadas, que dio un total de 172 horas, para el resultado día por día de las horas restantes reales se restó total de horas del día menos las horas laboradas en ese día. Ejemplo día 1: 172 – 8 = 164 horas; día 2: 164 – 8 = 156 horas.

El resultado de las horas restantes estimadas se realizó mediante el siguiente cálculo: total de horas restantes estimadas del día – (horas totales estimadas dividido para total de días). Ejemplo día 1: 172horas – (172horas / 21 días) = 163.8 horas; día 2: 163.8 horas – (172horas / 21 días) = 155.6 horas. Todos estos cálculos de los 21 días totales se lo representan en la tabla 18.

Tabla 18. Resultados de las horas restantes reales y estimadas

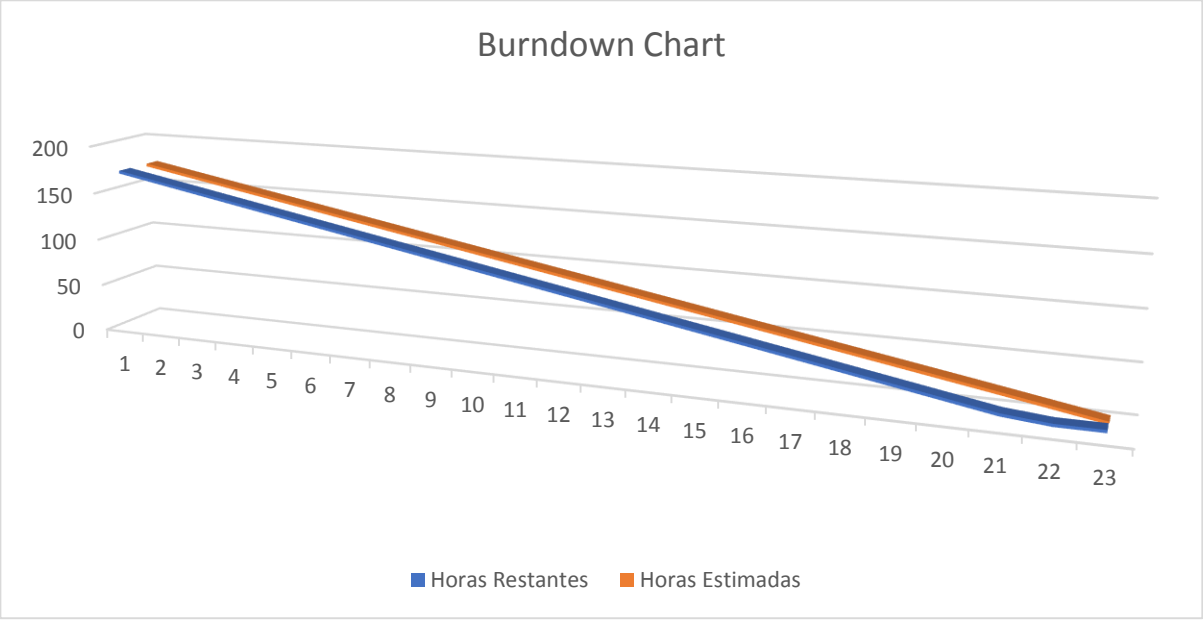
	h	DIA 1	DIA 2	DIA 3	DIA 4	DIA 5	DIA 6	DIA 7	DIA 8	DIA 9	DIA 10
Horas Restantes	172	164	156	148	140	132	124	116	108	100	92
Horas Estimadas	172	164.18	156.36	148.55	140.73	132.91	125.09	117.27	109.45	101.64	93.82

DIA 11	DIA 12	DIA 13	DIA 14	DIA 15	DIA 16	DIA 17	DIA 18	DIA 19	DIA 20	DIA 21	DIA 22	
84	76	68	60	52	44	36	28	20	12	8	8	real
86.00	78.18	70.36	62.55	54.73	46.91	39.09	31.27	23.45	15.64	7.82	0.00	ideal

Fuente: elaboración propia

El Burndown Chart presentado en la figura 26, detalla la relación entre las horas estimadas color naranja y las horas restantes reales color azul , los días se encuentran en el eje de las abscisas (x), y el tiempo medido en horas se encuentra en el eje de las ordenadas (y), si la línea azul está por debajo de la línea naranja esto indica que el proyecto esta adelantado al tiempo estimado, mientras que la línea azul está por encima de la línea naranja indica que el proyecto está retrasado al tiempo estimado. Si las dos líneas se encuentran a la par indica que el proyecto está dentro de la planificación realizada.

Figura 40. Burndown Chart



Fuente: elaboración propia

4.1.9. Implantación. Inspección e Iteración

La aplicación software se instaló en un computador de escritorio y un computador portátil de la empresa OmbuGames Cia. Ltda., se generó un archivo .exe, que se lo podrá guardar en CD o memorias flash. El software es instalado por técnicos propios de la empresa. Los requerimientos mínimos principales para el funcionamiento requerido son: procesador core I5, memoria RAM: 8Gb, conexión a internet. Una vez instalado se invitó a personas para que usen el sistema y realicen las pruebas de funcionamiento y funcionalidad que se puntualizan a continuación.

4.1.9.1. Pruebas tipo usabilidad

La valoración de usabilidad de la aplicación software, se realizó mediante un test donde se valoró identidad, contenido, navegación, utilidad y retroalimentación.

Tabla 19. Cuestionario, prueba de Identidad

Identidad	Si	No
¿La identidad corporativa se representa en la pantalla de inicio expuesta en colores, de la empresa?		x
¿Existe algún componente gráfico o de texto, que le ayudó a comprender a qué empresa pertenece?	x	
¿El sistema cumple con todos los requerimientos funcionales deseados?	x	
¿La pantalla de inicio y sus elementos, existe algo que usted crea que no está fuera de contexto?	x	
TOTAL:	3	1

Fuente: elaboración propia

Análisis: La persona capacitada para el uso del sistema, y después realizada la encuesta de Identidad, de 4 preguntas, 3 son positivas y 1 es negativa, obteniendo un 75% de efectividad en la identidad del sistema con la empresa ofertante de este servicio. Evidenciando que se debe trabajar más en los colores de la empresa y plasmarlos en los sistemas desarrollados.

Tabla 20. Cuestionario, prueba de Contenido

Contenido	Si	No
¿Los contenidos están conocidos en la aplicación?	x	
¿El contenido más relevante ofrecido, usted pudo distinguirlo a primera vista?	x	
¿Se identifica de forma clara los paneles para el ingreso de datos parametrizables?	x	
¿Para ingresar los datos al sistema, tiene una descripción clara de lo que se requiere?	x	
¿No se repite, en el sistema, información?	x	
TOTAL:	5	0

Fuente: elaboración propia

Análisis: Aplicado el cuestionario de contenido, de 5 preguntas todas fueron positivas, equivalente al 100%, se evidencia que los requerimientos funcionales fueron cumplidos, y el contenido es fácil de entender y el ingreso de datos es guiado correctamente. El sistema es personalizado para este caso en particular.

Tabla 21. Cuestionario, prueba de Utilidad

Utilidad	Si	No
¿Al observar la pantalla de inicio se puede advertir qué ofrece el sistema?	x	0
¿Cree que el servicio que se brinda en el sistema es de utilidad para este caso en particular?	x	0
TOTAL:	2	0

Fuente: elaboración propia

Análisis: Aplicado el cuestionario de Utilidad, basado en la problemática a solucionar, se obtuvo de 2 preguntas el 100%, por lo que se evidencia que el sistema ayuda a la toma de decisiones de los nuevos puntos de atención al cliente en el mapa de Riobamba.

Tabla 22. Cuestionario, prueba de Navegación

Navegación	Si	No
¿La navegación en las opciones del sistema, es fácil de realizar?	x	

¿La navegación es opacada por confusión en los elementos a ingresar por su volumen?		x
¿Los datos ya ingresados, son distinguidos de los datos no ingresados?	x	
TOTAL:	2	1

Fuente: elaboración propia

Análisis: Aplicado el cuestionario de navegación, la segunda pregunta es negativa, obteniendo un 66.66% de efectividad de 3 preguntas aplicadas. Se tomó especial atención a la pregunta negativa, evidenciando que por la cantidad de datos para iniciar la simulación correctamente, desde la carga del mapa de Riobamba, el usuario debe tener un nivel medio de práctica para adecuarse al ingreso de los datos necesarios.

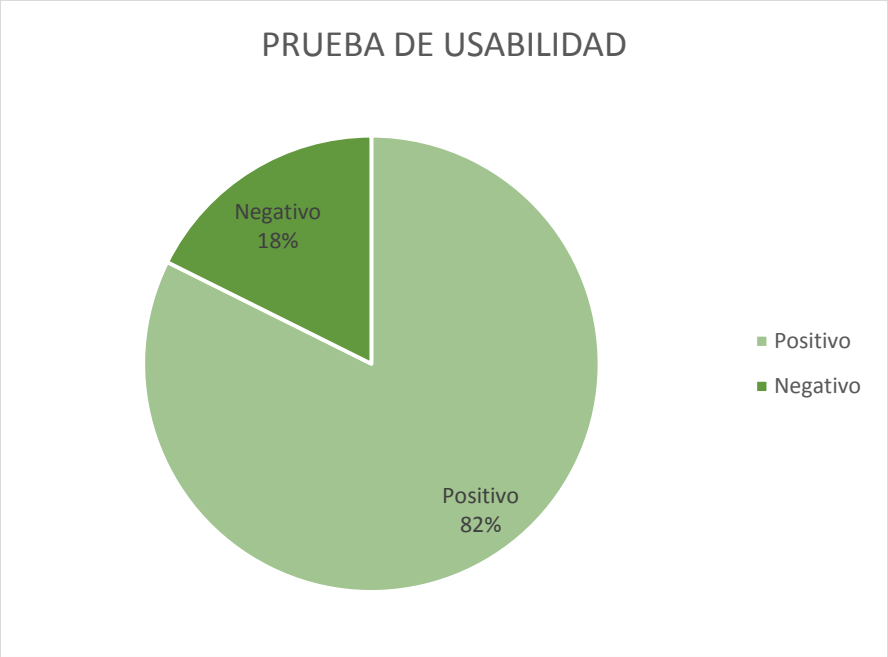
Tabla 23. Cuestionario, Retroalimentación

Retroalimentación	Si	No
¿Son las expectativas cubiertas por la aplicación?	x	
¿Visualizando los datos de la simulación, todo marcho bien?		x
¿Al momento de generar los archivos planos Json, con la información requerida, todo marcho bien?	x	
TOTAL:	2	1

Fuente: elaboración propia

Análisis: La retroalimentación posterior al uso por varias ocasiones del sistema, arrojo un 66.66% de efectividad, la pregunta que obtuvo una respuesta negativa, se lo analizó, y el algoritmo genético al no encontrar una solución después de 40 mil iteraciones, el tiempo de espera es significativo, y se mostrará los resultados de la última iteración, debido a que los datos de distancia y número de puntos a ubicar no concuerdan con los espacios delimitados de las parroquias limitadas en el mapa de Riobamba.

Figura 41. Resultados de Usabilidad



Fuente: elaboración propia

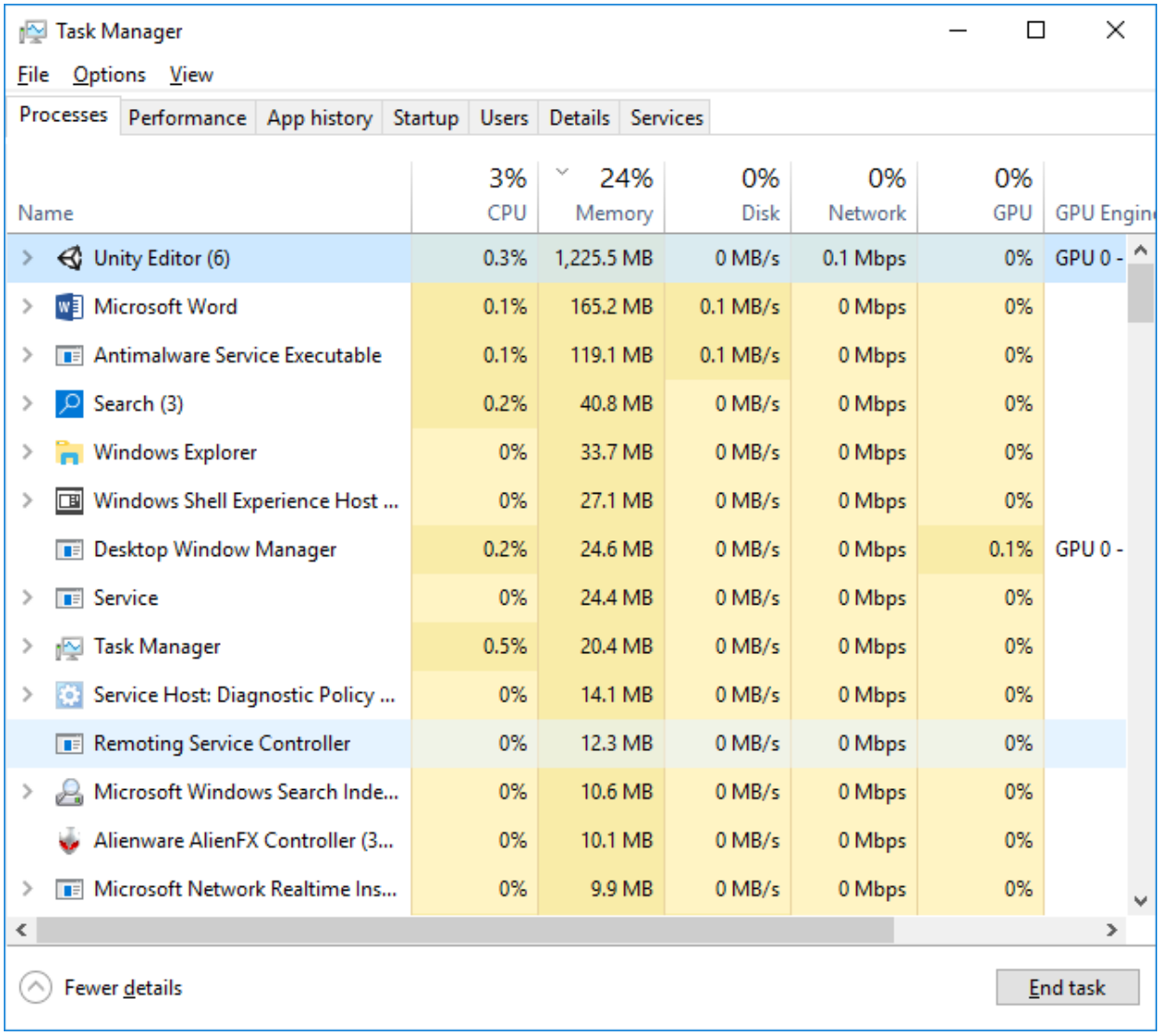
Posterior a la tabulación de los resultados totales de todos los cuestionarios aplicado a 30 personas afines y no afines a sistemas informáticos, entre los que se destaca grupo de ingenieros electrónicos, dedicados a telecomunicaciones, ingenieros y licenciados electricistas, y por último profesional en arquitectura, cada grupo de profesionales dio sus puntos de vista de cómo el sistema puede aplicarse en cada una de sus áreas de profesión,

Se obtuvo un 82% favorable de 17 preguntas realizadas. Se concluye que el sistema cumple con parámetros básicos de uso, y que requiere práctica en el uso de este, para solventar deficiencias en el ingreso de datos necesarios, por parte del usuario.

4.1.9.2. Pruebas tipo rendimiento

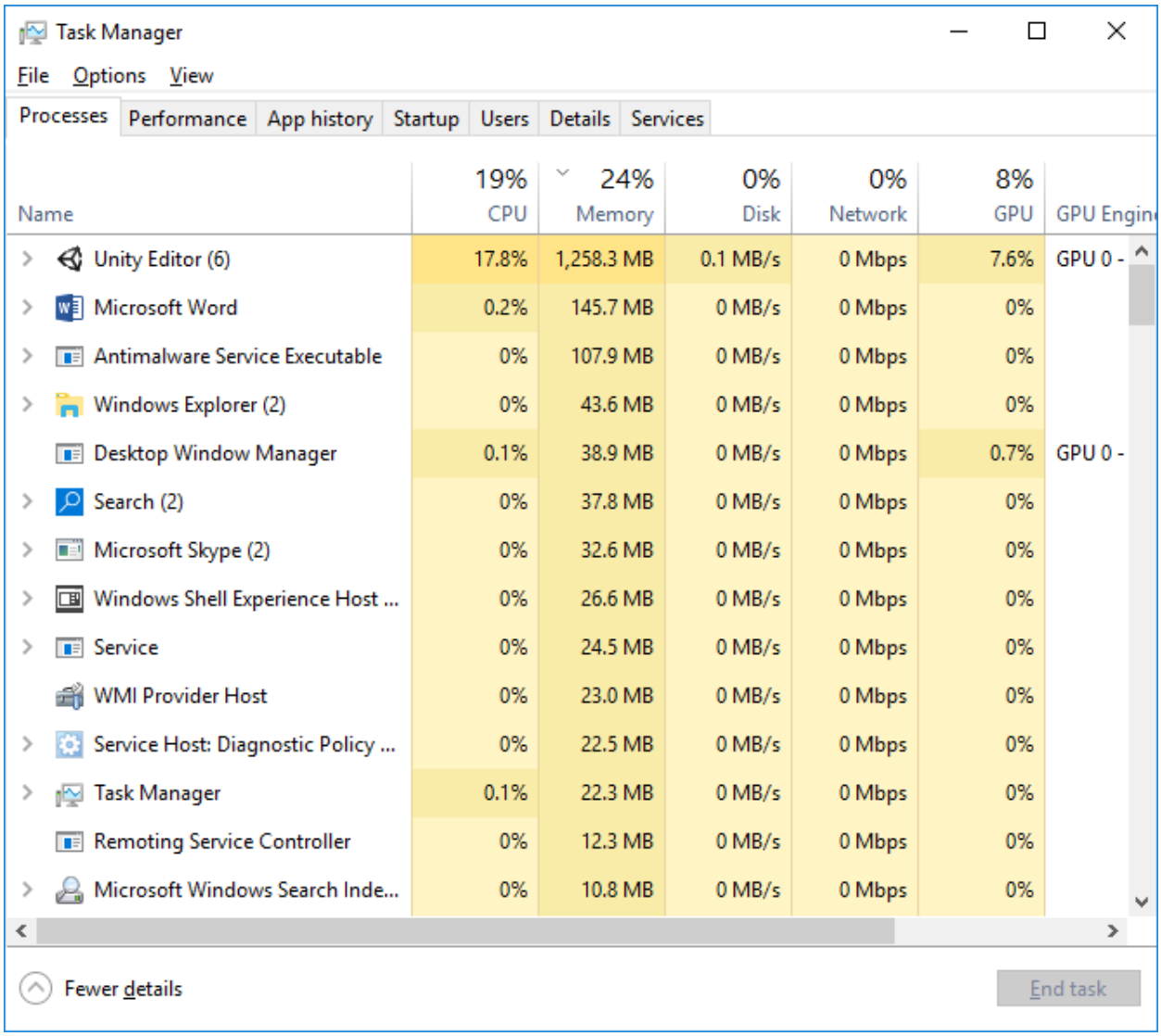
Para probar el rendimiento y uso de recursos, del computador especificado en la sección 4.3., se visualizó el rendimiento en el administrado de tareas, durante dos procesos de simulación, obteniendo los siguientes resultados detallados en las figuras a continuación.

Figura 42. Uso de recursos al iniciar la aplicación



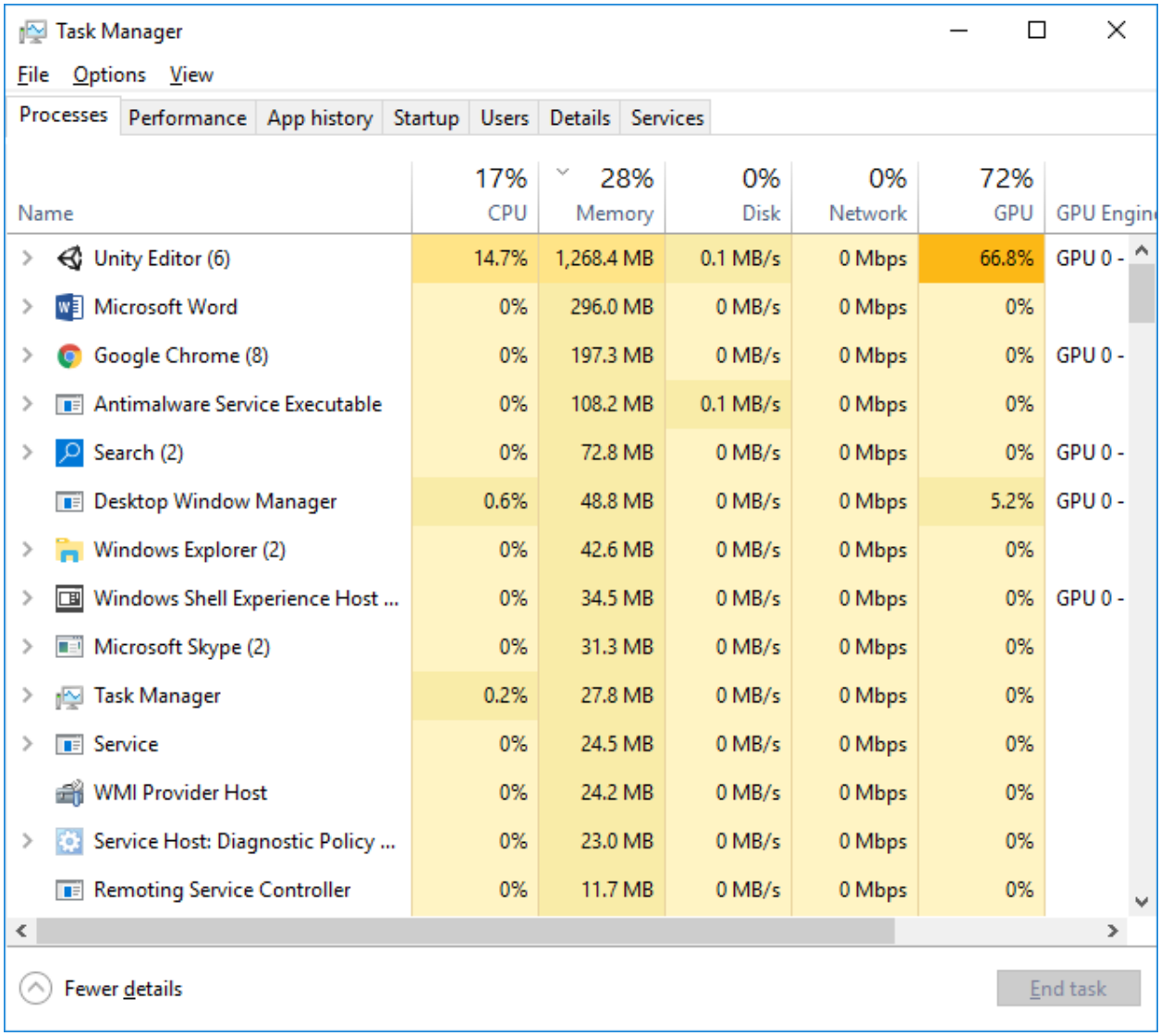
Fuente: elaboración propia

Figura 43. Uso de recursos aplicando el algoritmo genético



Fuente: elaboración propia

Figura 44. Uso de recursos visualizando la simulación de afluencia en tiempo real.



Fuente: elaboración propia

4.2. Materiales y herramientas

Para el desarrollo de la aplicación software, se detalla en la Tabla 24., todos los elementos empleados tanto de hardware como de software.

Tabla 24 Detalles de Hardware y Software utilizados

HARDWARE	
Computador	Procesador: Intel Core i7-6700HQ CPU @ 2.60 GHz Memoria Ram: 16.0 GB Tarjeta de video: NVIDIA® GeForce® GTX 970 3Gb Disco Duro: 256GB PCIe SSD (Boot) + 1TB 7200RPM SATA 6Gb/s (Storage)
SOFTWARE	
Sistema Operativo	Windows 10 Home
Motor de gráficos	Unity 2017.3
Entorno de Desarrollo Integrado (IDE)	Microsoft Visual Studio 2012
Lenguaje de programación	C#
Mapas	API de Google Maps

Fuente: elaboración propia

CAPÍTULO 5

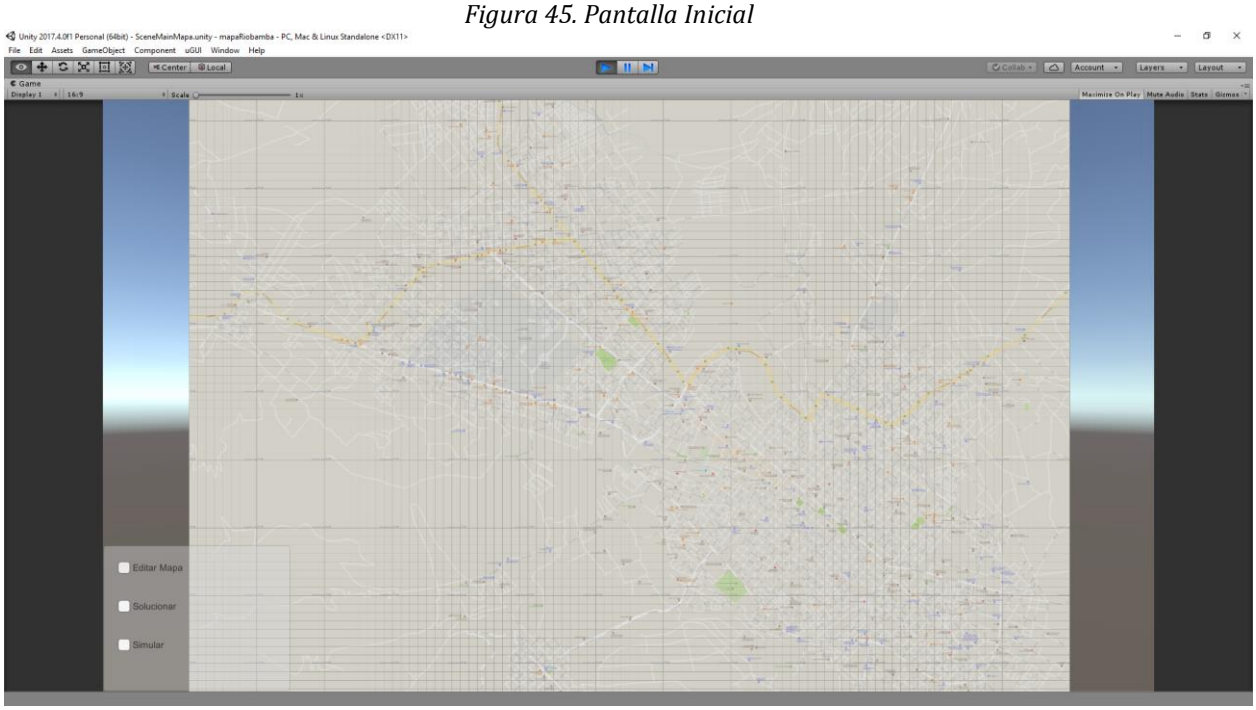
Resultados

5.1. Producto final del proyecto de titulación

Al término de los sprints del entorno de trabajo Scrum, siendo validados por el ProductManager y ScrumManager, se presentan las pantallas finales del funcionamiento del simulador visualizando la aplicación del algoritmo genético, y posterior afluencia de los usuarios a los nuevos puntos de atención generados.

Pantalla Inicial

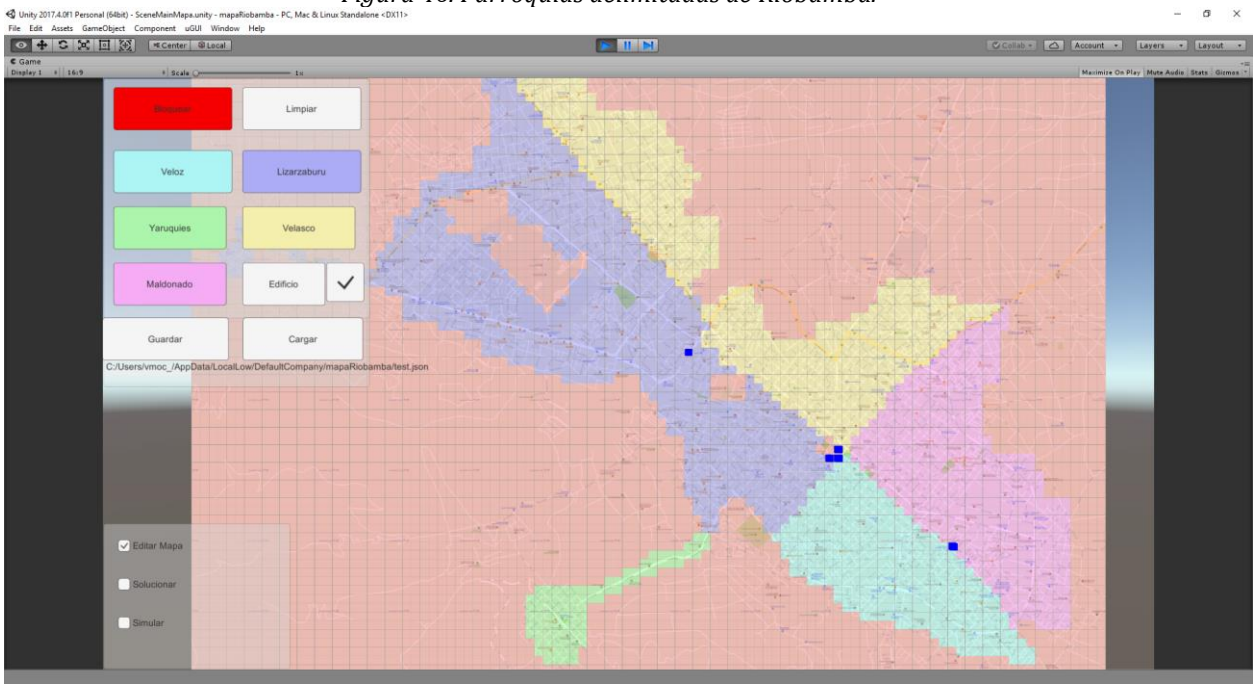
Figura 45. Pantalla Inicial



Fuente: elaboración propia

Pantalla Mapa Riobamba delimitada por sus parroquias

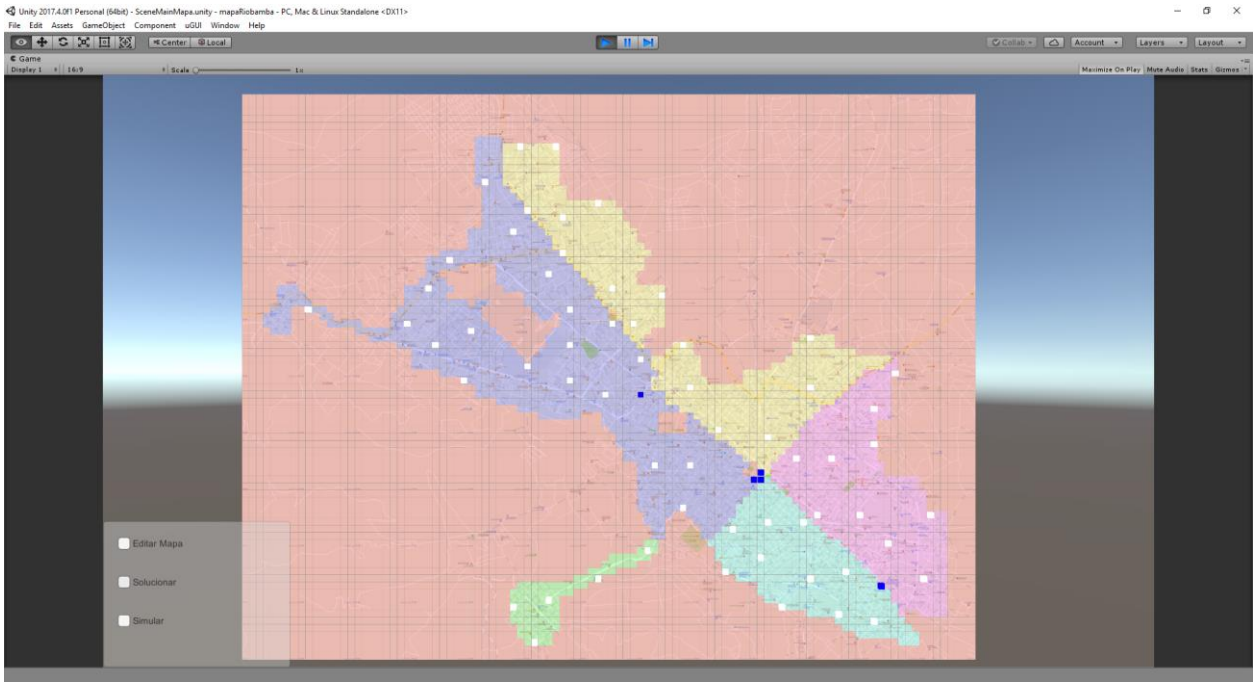
Figura 46. Parroquias delimitadas de Riobamba.



Fuente: elaboración propia

Aplicación del Algoritmo Genético

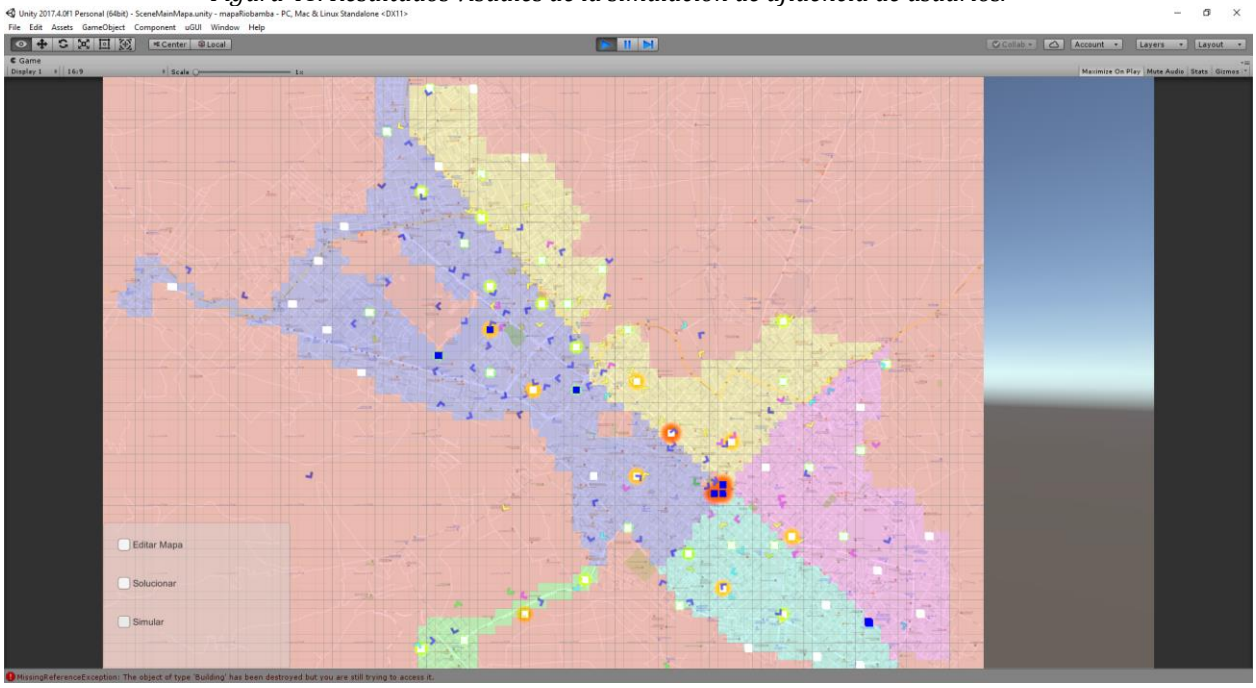
Figura 47. Resultado del algoritmo genético de forma visual



Fuente: elaboración propia

Simulación de afluencia

Figura 48. Resultados visuales de la simulación de afluencia de usuarios.

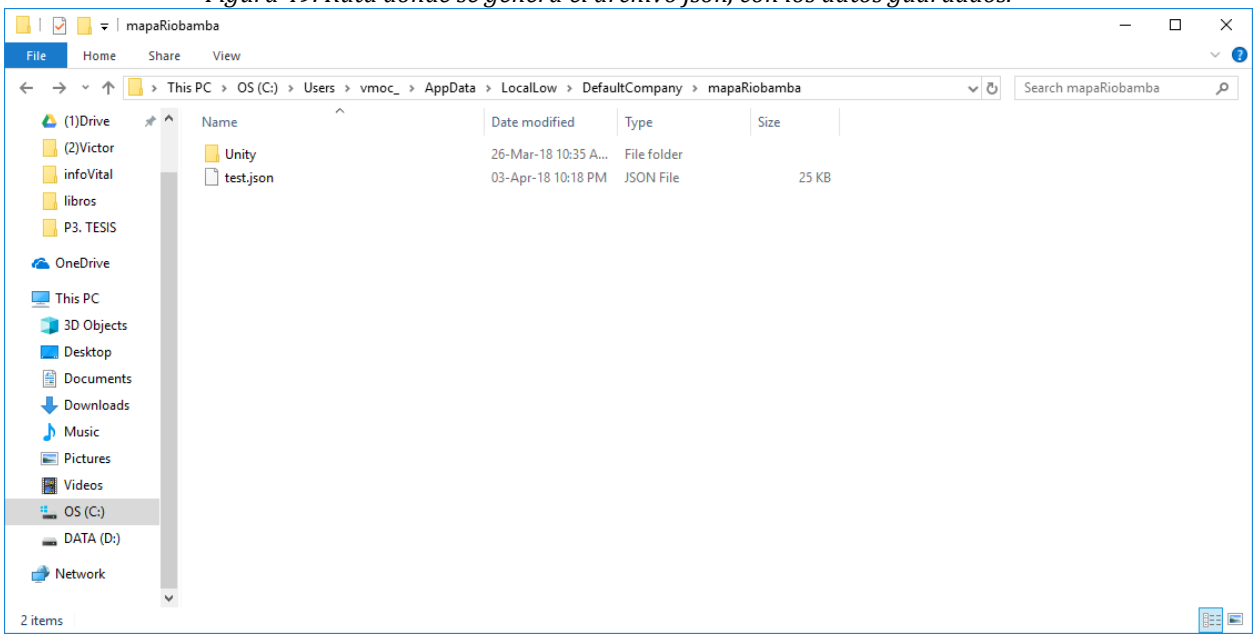


Fuente: elaboración propia

Generación de los archivos planos Json

C:\Users\nombreUsuario\AppData\LocalLow\DefaultCompany\mapaRiobamba

Figura 49. Ruta donde se genera el archivo json, con los datos guardados.

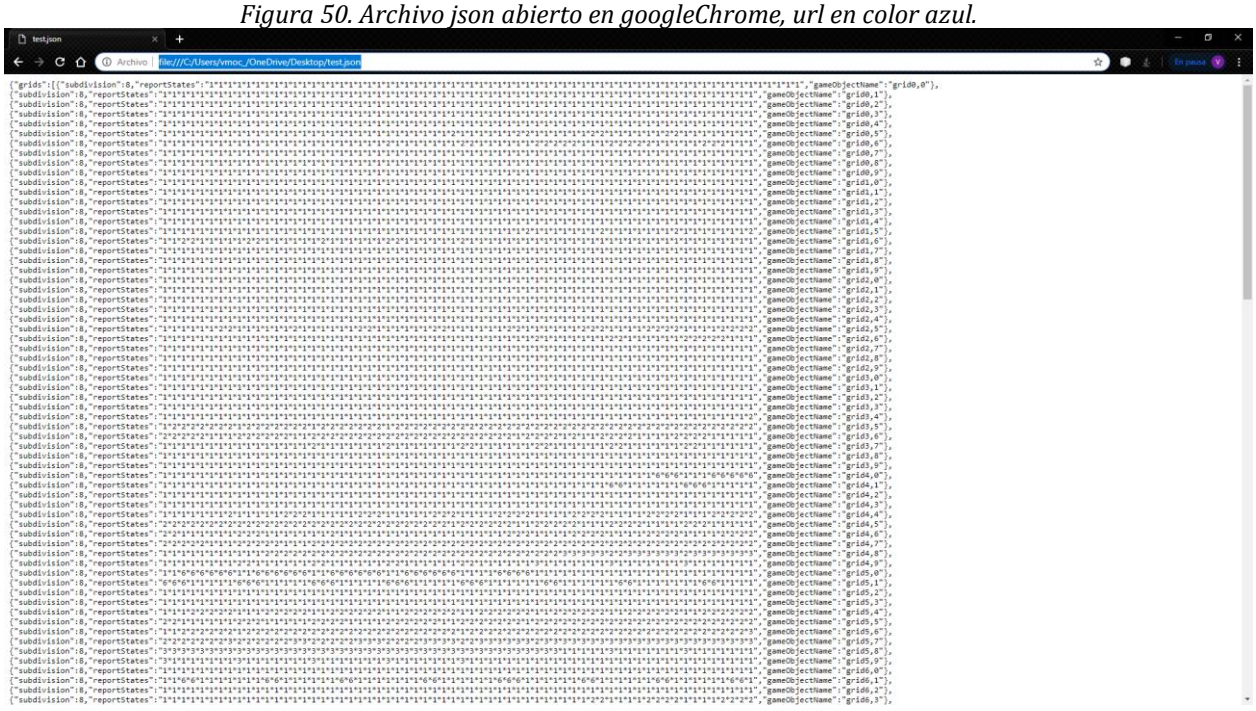


Fuente: elaboración propia

Proceso para convertir el archivo con extensión json a una hoja de cálculo en Excel.

El archivo json, se lo abrió en cualquier navegador de internet, y copiamos la dirección URL que se muestra.

Figura 50. Archivo json abierto en googleChrome, url en color azul.

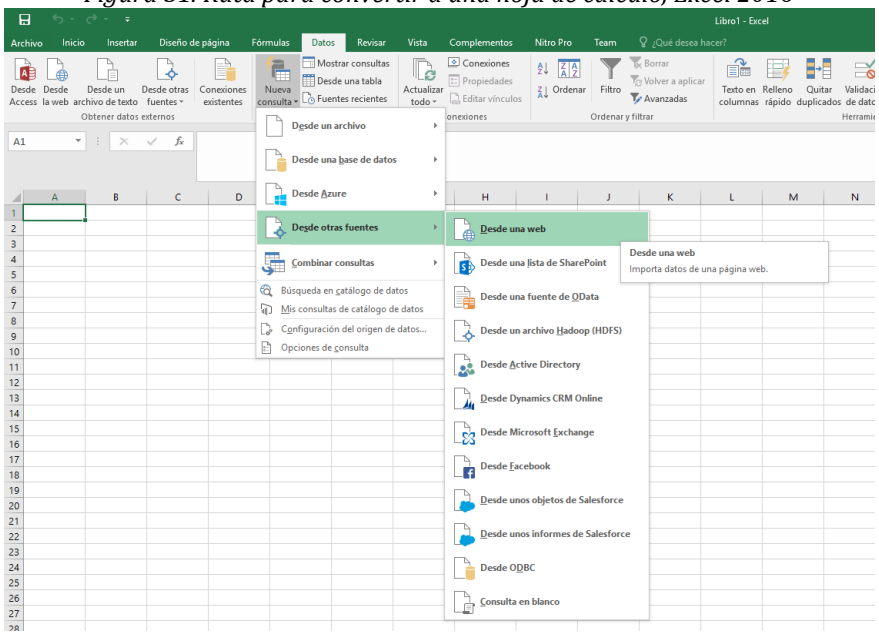


Fuente: elaboración propia

En la barra de menú de la hoja de cálculo Excel, escogemos “Datos”, y en la barra de opciones escogemos:

Nueva consulta – Desde otras fuentes – Desde una web, como se muestra en la siguiente figura.

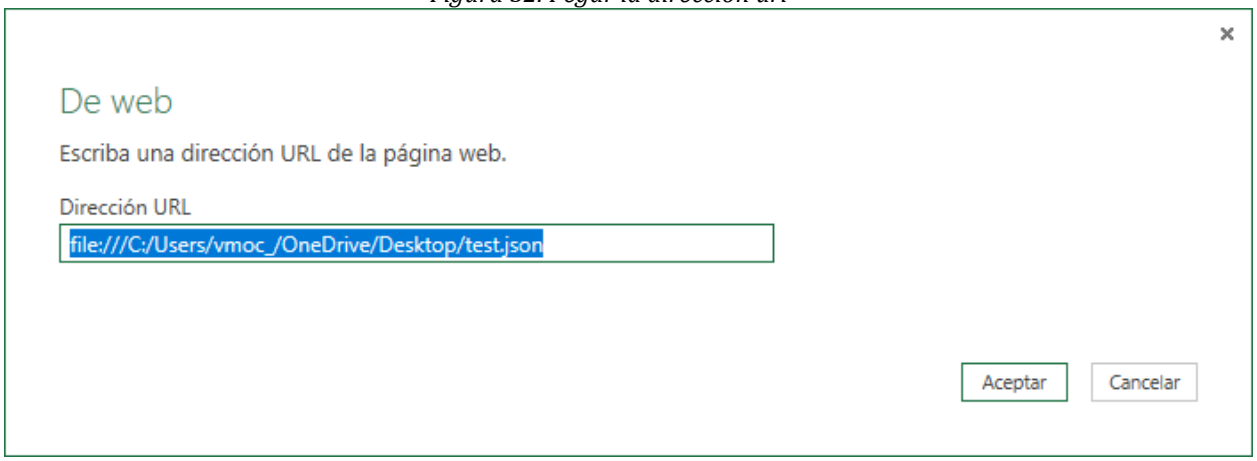
Figura 51. Ruta para convertir a una hoja de cálculo, Excel 2016



Fuente: elaboración propia

Consiguiente mente aparecerá una ventana donde pegaremos la dirección url copiada anteriormente, y damos clic en Aceptar.

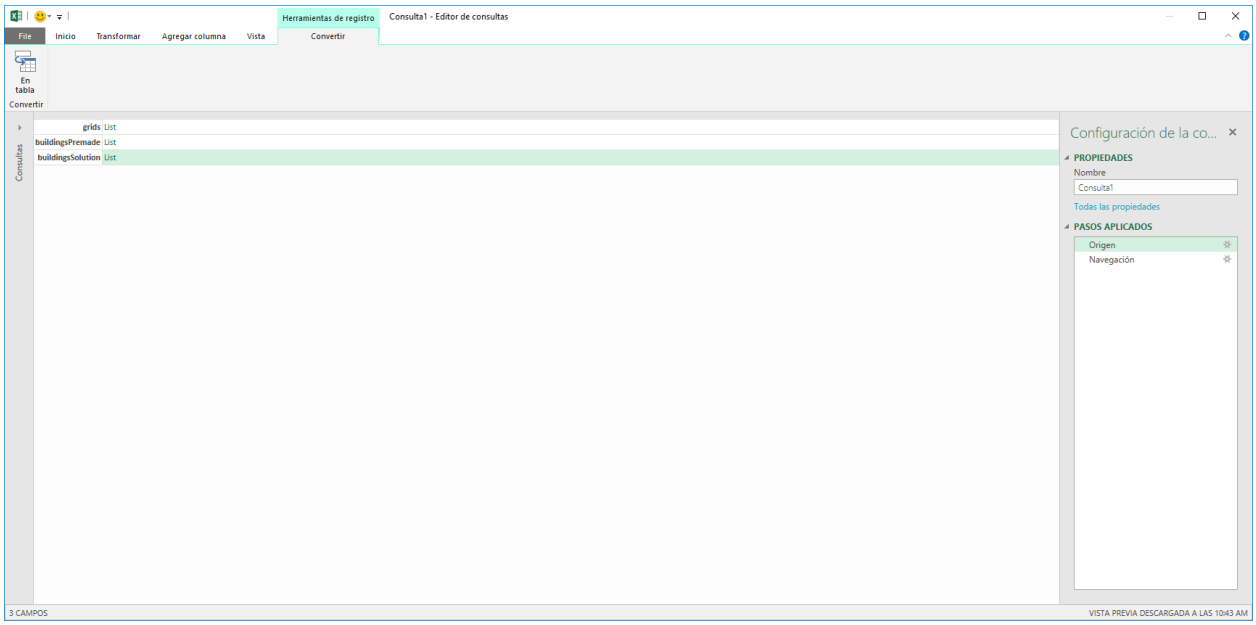
Figura 52. Pegar la dirección url



Fuente: elaboración propia

En la ventana consiguiente se mostrará las tres listas que contiene el archivo json, de las cuales se escogerá "buildingsSolution".

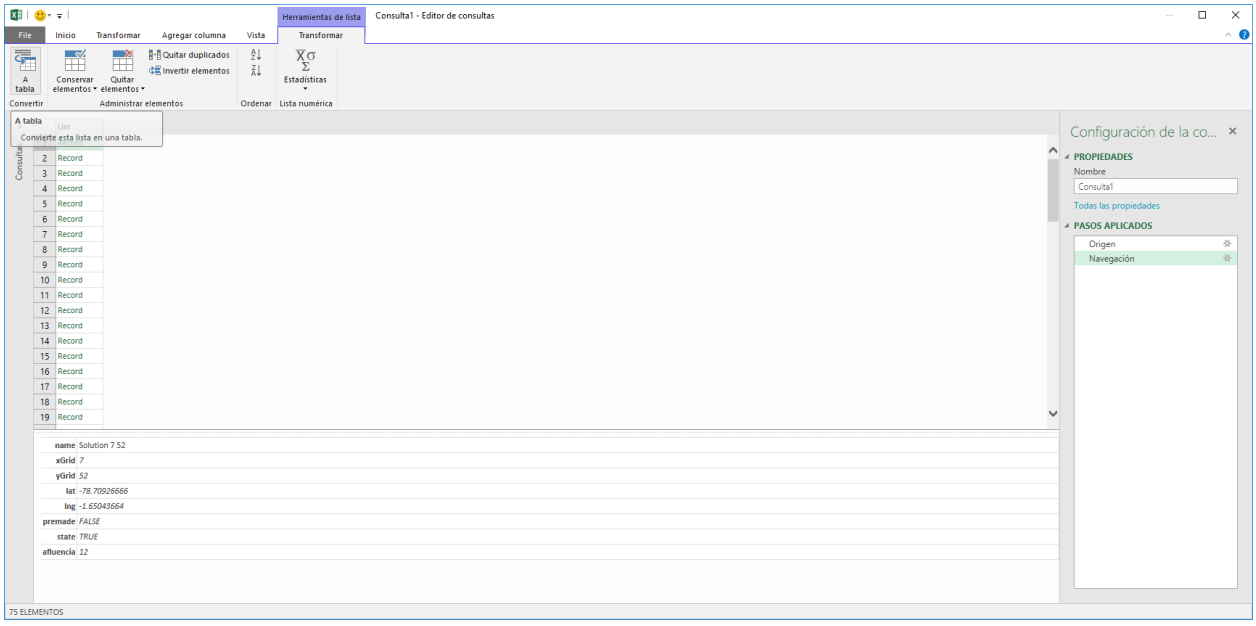
Figura 53. Dar click en la lista llamada "buildingsSolution"



Fuente: elaboración propia

Se muestra todos los elementos de la lista escogida, al dar clic sobre uno de ellos se visualiza en la parte inferior una tabla con los datos del elemento seleccionado. En la barra de opciones escogemos “Convertir A tabla”, ubicado en la parte izquierda de la barra.

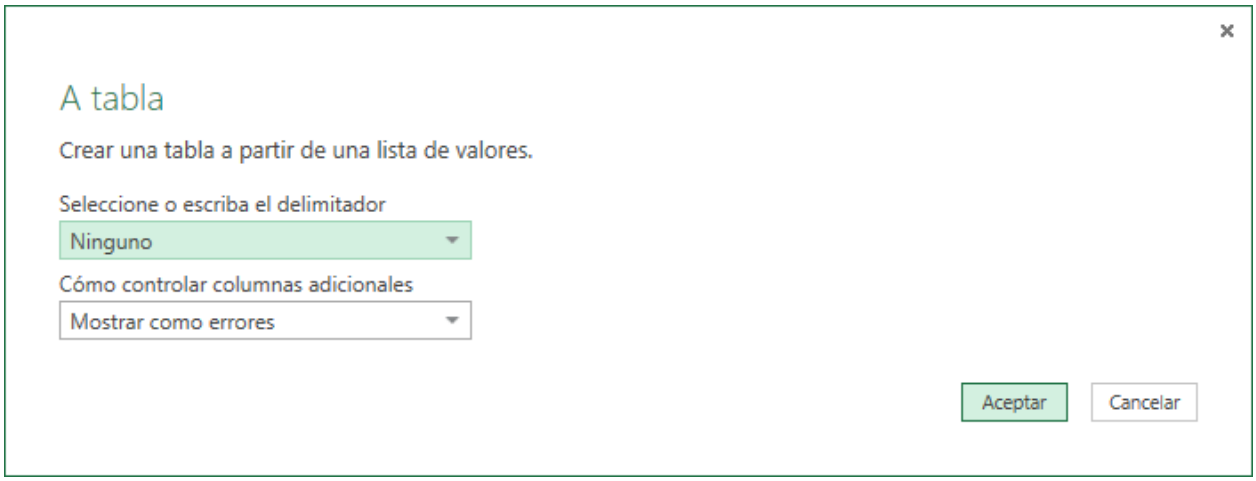
Figura 54. Elementos de la lista a convertir a tabla de datos de Excel.



Fuente: elaboración propia

En la siguiente ventana, se escogerá el delimitador, en este caso la opción “ninguno”, y damos clic en aceptar.

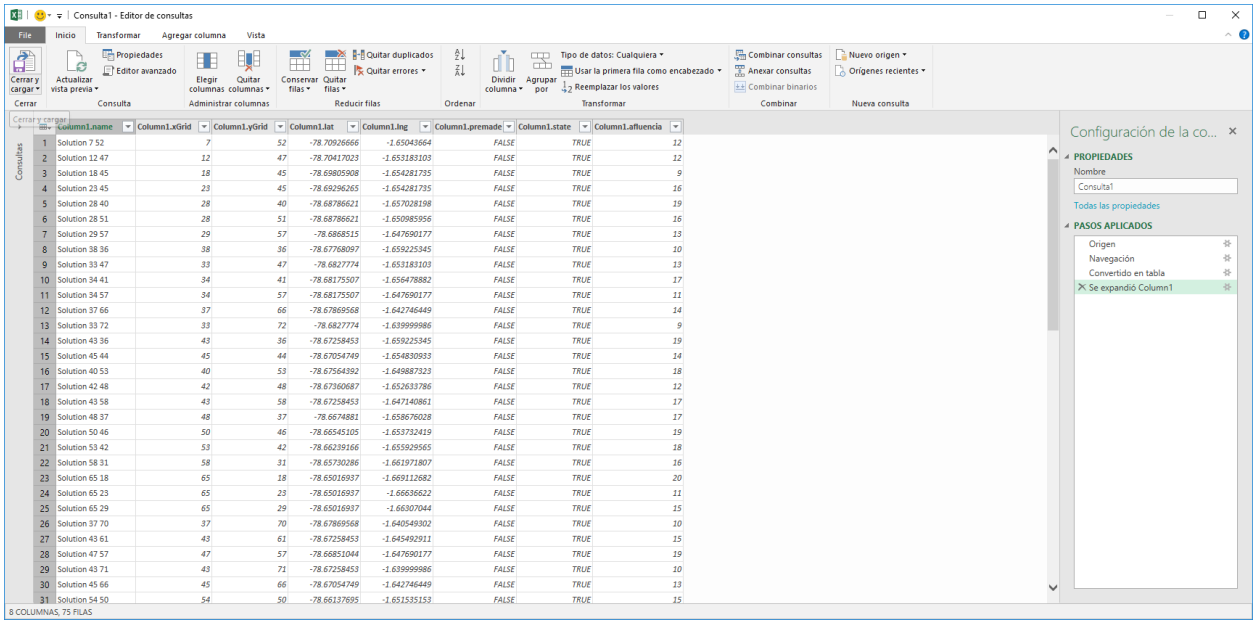
Figura 55. Seleccionar el tipo de delimitador de los datos.



Fuente: elaboración propia

Se mostrará la consulta provisional con los datos generados, damos clic en la barra de opciones a "Cerrar y cargar". Y de esta forma se creará la hoja de cálculo para su posterior gestión.

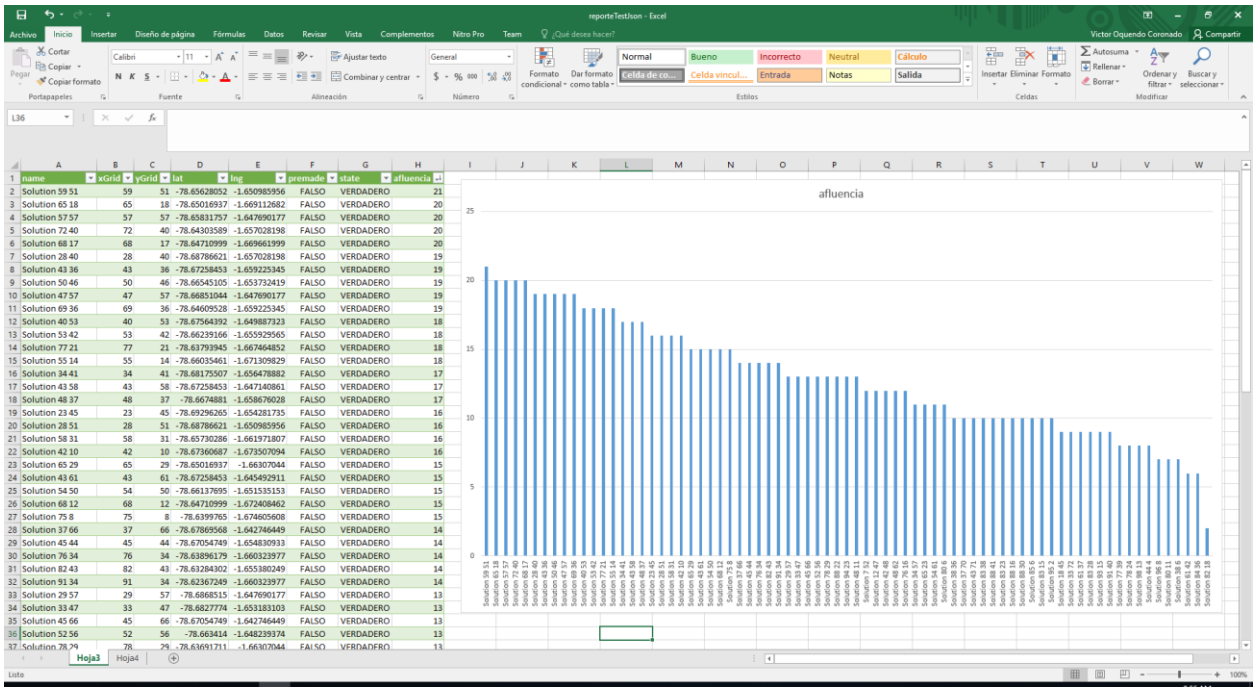
Figura 56. Generación de la consulta con los datos generados.



Fuente: elaboración propia

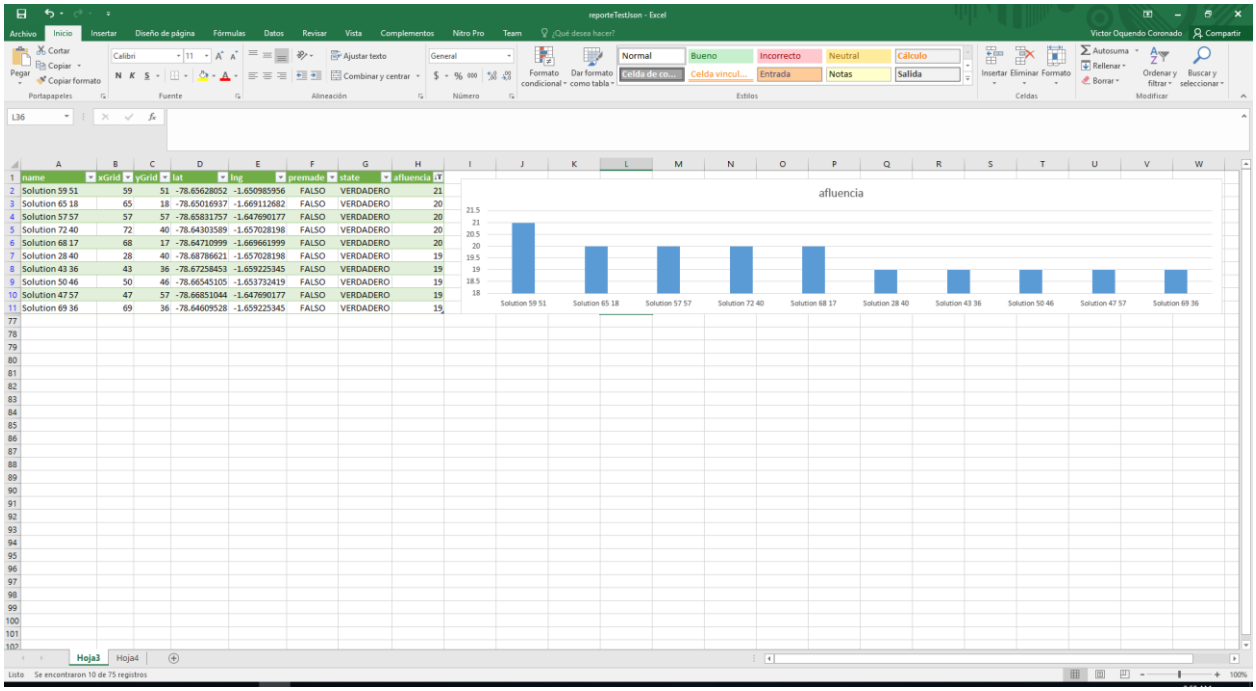
Gestión de resultados en tablas y gráficos estadísticos.

Figura 57. Resultados en archivo Excel.



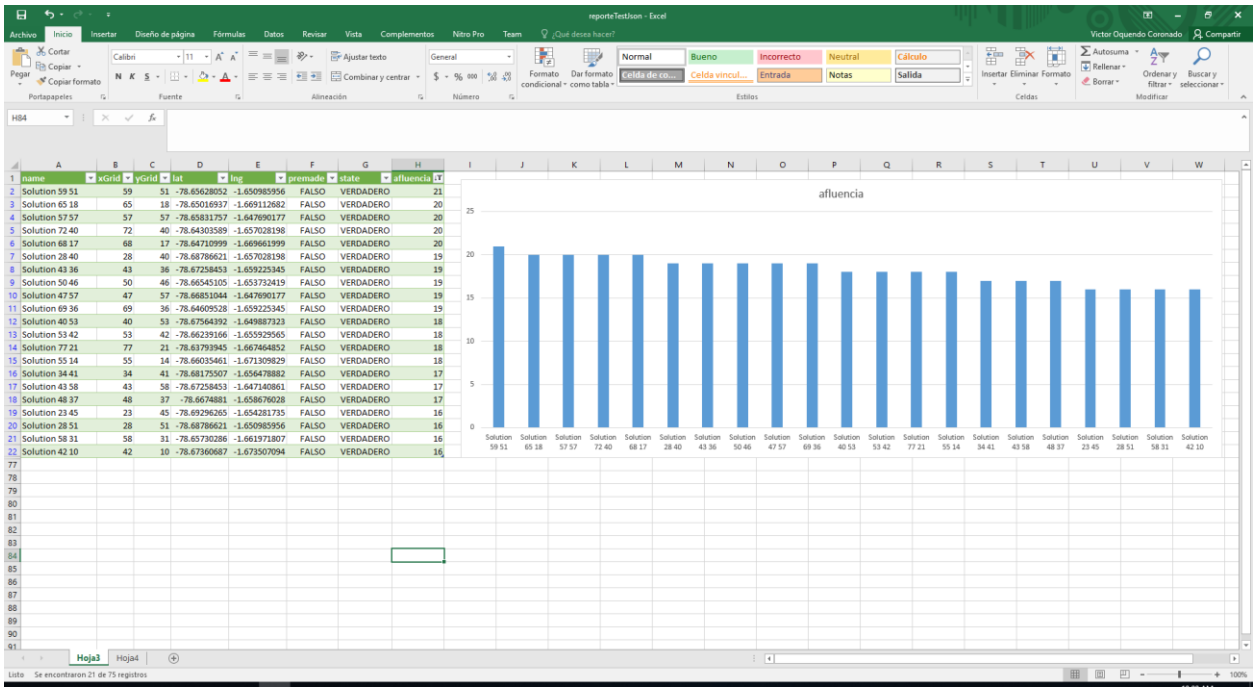
Fuente: elaboración propia

Figura 58. Top 10 de mayor afluencia.



Fuente: elaboración propia

Figura 59. Top 20 afluencia



name	xGrid	yGrid	lat	lng	premade	state	afluencia
Solution 59 51	59	51	-78.65628052	-1.650985956	FALSO	VERDADERO	21
Solution 65 18	65	18	-78.65016937	-1.669112682	FALSO	VERDADERO	20
Solution 57 57	57	57	-78.65831757	-1.647690177	FALSO	VERDADERO	20
Solution 72 40	72	40	-78.64303589	-1.657028198	FALSO	VERDADERO	20
Solution 68 17	68	17	-78.64710999	-1.669661999	FALSO	VERDADERO	20
Solution 28 40	28	40	-78.68786621	-1.657028198	FALSO	VERDADERO	19
Solution 43 36	43	36	-78.67258453	-1.659225345	FALSO	VERDADERO	19
Solution 50 46	50	46	-78.66545105	-1.653732419	FALSO	VERDADERO	19
Solution 47 57	47	57	-78.66851044	-1.647690177	FALSO	VERDADERO	19
Solution 69 36	69	36	-78.64609528	-1.659225345	FALSO	VERDADERO	19
Solution 40 53	40	53	-78.67564392	-1.649887323	FALSO	VERDADERO	18
Solution 53 42	53	42	-78.66239166	-1.655925565	FALSO	VERDADERO	18
Solution 77 21	77	21	-78.63793945	-1.667464852	FALSO	VERDADERO	18
Solution 55 14	55	14	-78.66035461	-1.671309829	FALSO	VERDADERO	18
Solution 34 41	34	41	-78.68175507	-1.656478882	FALSO	VERDADERO	17
Solution 43 58	43	58	-78.67258453	-1.647140861	FALSO	VERDADERO	17
Solution 48 37	48	37	-78.6674881	-1.658676028	FALSO	VERDADERO	17
Solution 23 45	23	45	-78.69296265	-1.654281735	FALSO	VERDADERO	16
Solution 28 51	28	51	-78.68786621	-1.650985956	FALSO	VERDADERO	16
Solution 58 31	58	31	-78.65730286	-1.661971807	FALSO	VERDADERO	16
Solution 42 10	42	10	-78.67360687	-1.673507094	FALSO	VERDADERO	16

Fuente: elaboración propia

5.2. Evaluación preliminar

El impacto de la aplicación software, aplicada al ProductManager fue de total satisfacción por parte de este, al observar que las ubicaciones fueron creadas por medio de distancias entre puntos parametrizables, visualizando en tiempo real en la simulación, con la facilidad de habilitar o deshabilitar los estados de los puntos de atención, que tiene impacto con la afluencia de usuarios representados por iconos a dichos puntos de atención. El detalle de los cuestionarios realizados al ProductManager se detalla en la sección 4.2.7.2, donde se encontrará el análisis de usabilidad del sistema.

Además, el mismo sistema fue presentado a distintos profesionales en distintas áreas, eléctrico, arquitecto, electrónico.

Versión profesional eléctrico. La ubicación mediante distancias sería muy útil para optimizar la ubicación de transformadores en el mapa de Riobamba, mientras que la simulación sería diferente mediante las cargas de corriente que debe soportar cada transformador, dependiendo del número de viviendas a abastecer. [1]

[1] Versión rendida por el Ing. Marco Bustos, tras observar la simulación de la aplicación.

Versión profesional, electrónico en telecomunicaciones. Al observar el resultado de la simulación, pude constatar que podría aplicar en la distribución de los equipos de última milla para la conexión de internet mediante fibra óptica, no solamente en el mapa de Riobamba, sino en cualquier mapa precargado en el sistema mismo. [2]

Versión profesional, arquitecto. La aplicación sería muy útil para la planificación de espacios verdes en el mapa de Riobamba para fomentar el cuidado y la importancia de estos espacios. [3]

5.3. Análisis de resultados

Basado en el objetivo general descrito en la sección 2.5.1., se cumplió al 100% la implementación del algoritmo genético y para el análisis se generó un simulador donde se puede visualizar en tiempo real la creación de nuevos puntos de atención a usuarios y los comportamientos de afluencias de estos a los puntos creados.

Los objetivos específicos están cumplidos, y se los plasma en la aplicación software final, aplicando la teoría de algoritmo genético adaptativo a este caso en específico, mientras que la simulación de la afluencia de las personas se lo realizó de forma de elección randómica probabilística a los puntos habilitados de atención generados en el mapa de Riobamba.

La aplicación de algoritmo genético alrededor del mundo es muy variada, desde la deducción de las mejores rutas en un mapa hasta sistemas de detección de intrusos en tráfico de red de internet, todos demuestran que los parámetros y procesos de evolución del algoritmo genético resuelven problemas. La selección de individuos mediante función objetivo, la reproducción de los más aptos, y el proceso de mutación todos estos detallados en la sección 3.1.1.3 y 3.1.1.4 permitió alcanzar la solución de ubicación de nuevos puntos de atención en el mapa de Riobamba. Además, por medio de los criterios de varios profesionales citados en la sección 5.2., se determinó que la aplicación desarrollada puede servir para la solución de otros problemas, mediante la adaptación de nuevos mapas.

El algoritmo genético que se usó en la aplicación software, conlleva todas la partes fundamentales del mismo, y adaptándolas para la solución del problema, algo muy interesante es que la mutación aplicada no solamente ayuda para generar variedad en las poblaciones generadas, sino que a la vez permite seguir refinando la solución, al momento de encontrar dos puntos de atención representados por el número 1 que no cumplen con la condición de distancia, la mutación realiza el cálculo del vector posición relativa para

[2] Versión rendida por el Ing. Cristhian Vallejo, profesional dedicado a las telecomunicaciones.
[3] Versión rendida por el Arq. Andrés Guerra.

determinar la nueva ubicación del punto de atención sin entrar en conflicto con la condición de distancia entre puntos de atención, con este tipo de mutación se demuestra que los procesos de evolución pueden ser adaptativos sin perder el objetivo de resolución del problema.

Con respecto al uso de recursos hardware, a la aplicación software se lo optimizó para que no necesite demasiados recursos hardware, los puntos optimizados son: el tamaño de cada celda del mapa de Riobamba que se lo solicita mediante el api de Google Maps, mediante el uso de Unity el procesamiento de los gráficos no se lo realiza únicamente en el procesador del computador, sino que se lo envía a la tarjeta gráfica para que la simulación en tiempo real sea completamente fluida, y la operación genética de mutación se lo adapto para este caso en específico como esta detallado en el párrafo anterior, consiguiendo la reducción de tiempo en el procesamiento del algoritmo genético.

CAPÍTULO 6

Conclusiones y Recomendaciones

6.1. Conclusiones

- La aplicación del algoritmo genético determinó las posibles ubicaciones de nuevos puntos de atención en el mapa de Riobamba, además permitió la visualización de calles por medio del api de Google maps, en tiempo real.
- El presente trabajo determinó la extracción de los principios de la inteligencia biológica de los datos y teorías de los biólogos, aplicados mediante esfuerzos de ingeniería para plasmarlo en un sistema funcional adaptable a las condiciones que requirió la solución, La forma de aplicación de la teoría de mutación que se realizó, fue un punto estratégico para encontrar la solución en el algoritmo genético, la mutación permitió acelerar el tiempo y reducir el uso de recursos propios del computador.
- La función objetivo al tratarse de un problema de ubicaciones en el mapa urbano de la ciudad de Riobamba, se la diseño en base a distancias mínimas parametrizables, entre los puntos de los posibles nuevos sitios generados, en una matriz de dos dimensiones fila columna.
- Para implementar la aplicación software, se basó en una simulación visual en tiempo real, apoyándose en Unity3d, software que facilita realizar dichas tareas, complementando con el api de google map para entrelazar la simulación en el mapa de Riobamba.
- La aplicación software generó un reporte en formato json, con las ubicaciones fila columna de la matriz creada en la capa del mapa de Riobamba, de los puntos de atención generados por el algoritmo genético, con porcentajes de afluencia de usuarios en la simulación. Este archivo json, facilitará a posterior a los usuarios especializados para realizar reportes posteriores basados en necesidades específicas.

6.2. Recomendaciones

- Se recomienda para la aplicación del proceso genético, adaptar a las necesidades de resolución de las ubicaciones de este problema, sin menospreciar ninguno de las etapas de la genética para encontrar al individuo más apto.

- Se solicita para el buen funcionamiento de la aplicación software, contar con dispositivos ya sean estos computadores de escritorio, portátiles, tabletas o celulares, que cuenten con un mínimo de 4 Gb de memoria RAM y si es posibles chipset de aceleradores gráficos.
- Se requiere aplicar conocimientos matemáticos, geométricos y vectoriales para la optimización de los cálculos que se implementan en las etapas genéticas, para encontrar la solución óptima, para no sobrecargar los recursos hardware, y optimizar los tiempos de solución.
- Se recomienda tener conocimientos medios en el entorno visual de un simulador 3d, además de conocimientos de sintaxis del lenguaje de programación C#.
- Para el manejo del archivo json, se debe conocer dicho esquema, para su posterior procesamiento y poder lograr reportes en los formatos que se desee.

APÉNDICES

APÉNDICE A. Documentación Proyecto - Historias de Usuario Scrum

	HISTORIA DE USUARIO	**SPRINT 1**
NUMERO: 1	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Petición Mapa Riobamba		
PRIORIDAD: ALTA	**RIESGO:** ALTO	**HORAS ESTIMADAS:** 24
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 24
DESCRIPCION: Se necesita la o las imágenes concatenadas del mapa urbano de la ciudad de Riobamba, detallando el nombre de las calles y puntos referenciales.		
VALIDACION: Se visualiza el mapa de Riobamba en el entorno de Unity, usando el api de Google Maps, con el zoom adecuado para detallar los nombres de calles y referencias.		

	HISTORIA DE USUARIO	**SPRINT 1**
NUMERO: 2	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Pintar Parroquias Urbanas en el Mapa de Riobamba		
PRIORIDAD: ALTA	**RIESGO:** Medio	**HORAS ESTIMADAS:** 12
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 10
DESCRIPCION: Se necesita delimitar las parroquias urbanas en el mapa de la ciudad de Riobamba		
VALIDACION: Se visualiza las 5 parroquias urbanas de la ciudad de Riobamba, delimitada por distintos colores.		

HISTORIA DE USUARIO		SPRINT 1
NUMERO: 3	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Ingreso de puntos de atención existentes en el mapa de Riobamba		
PRIORIDAD: MEDIA	**RIESGO:** MEDIO	**HORAS ESTIMADAS:** 12
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 10
DESCRIPCION: Se necesita ubicar manualmente los puntos de atención existentes en el mapa de Riobamba.		
VALIDACION: Visualización en tiempo real mediante objetos representativos, de los puntos de atención existentes después de dar clic en la ubicación deseada en el mapa de Riobamba.		

HISTORIA DE USUARIO		SPRINT 1
NUMERO: 4	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Visualización de las posibles nuevas ubicaciones de puntos de atención		
PRIORIDAD: ALTA	**RIESGO:** MEDIO	**HORAS ESTIMADAS:** 44
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 48
DESCRIPCION: Se necesita que el sistema de forma automática sugiera ubicaciones de las posibles nuevas ubicaciones para puntos de atención, en el mapa de Riobamba.		
VALIDACION: Visualización en tiempo real, de las nuevas posibles ubicaciones en el mapa de Riobamba, después de aplicar algoritmo genético mediante distancias entre puntos de atención.		

HISTORIA DE USUARIO		SPRINT 2
NUMERO: 5	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Modificar estado de los puntos de atención generados por el algoritmo genético		
PRIORIDAD: MEDIA	**RIESGO:** BAJO	**HORAS ESTIMADAS:** 12
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 12

DESCRIPCION: Se necesita que, los posibles nuevos puntos de atención, se modifique su estado de habilitado o deshabilitado.
VALIDACION: Visualización del estado mediante colores rojo: deshabilitado, verde: habilitado, de los puntos de atención generados, mediante un clic en la aplicación.

	HISTORIA DE USUARIO	**SPRINT 2**
NUMERO: 6	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Visualización de la afluencia de usuarios a los puntos de atención		
PRIORIDAD: MEDIA	**RIESGO:** MEDIO	**HORAS ESTIMADAS:** 44
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 40
DESCRIPCION: Se necesita observar, los posibles comportamientos de afluencia a los puntos de atención habilitados, en el mapa de Riobamba.		
VALIDACION: Visualización en tiempo real, la elección y trayectoria del usuario a los puntos de atención habilitados, aplicados en el mapa de Riobamba.		

	HISTORIA DE USUARIO	**SPRINT 2**
NUMERO: 7	**USUARIO:** Analista Planificación	
NOMBRE HISTORIA: Generar archivo Json, con las ubicaciones de los puntos de atención		
PRIORIDAD: MEDIA	**RIESGO:** ALTO	**HORAS ESTIMADAS:** 24
ITERACION: 1	**RESPONSABLE**: VICTOR OQUENDO	**HORAS REALES:** 20
DESCRIPCION: Se necesita almacenar en algún formato las ubicaciones y datos generados por la aplicación, para cargarlos en otras aplicaciones instaladas en otros dispositivos.		
VALIDACION: Generación de un archivo Json, con los datos necesarios para precargar los datos de una simulación ya realizada, en otros computadores con la aplicación instalada.		

APÉNDICE B. Documentación Proyecto - Pruebas Funcionales

PRUEBA FUNCIONAL	
Número Prueba: 1	**Historia de Usuario 1**
Nombre de la Prueba: Petición Mapa Riobamba	
Descripción: El usuario, al seleccionar la opción del menú "Editar mapa", se le brinda la opción de cargar el mapa urbano de la ciudad de Riobamba, dando clic en el botón "Cargar", mostrando en pantalla el mapa deseado, usando el api de google Maps.	
Condiciones de ejecución: Ninguna	
Entrada: - El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción "Editar mapa". - El sistema verificará la conexión a google Maps, ubicando latitud y longitud deseada y mostrará en pantalla el mapa de la ciudad de Riobamba. - El sistema creará la cuadricula respectiva sobre la extensión del mapa.	
Resultado esperado: Visualizar el mapa concatenado de la ciudad de Riobamba delimitado sus extremos de latitud y longitud	
Evaluación de la prueba: Prueba satisfactoria.	

PRUEBA FUNCIONAL	
Número Prueba: 2	**Historia de Usuario 2**
Nombre de la Prueba: Pintar Parroquias Urbanas en el Mapa de Riobamba	
Descripción: El usuario, en la opción del menú "Editar Mapa", se le brinda las opciones para delimitar las parroquias urbanas de la ciudad de Riobamba, diferenciándolas por colores predeterminados, el usuario dará clic en las casillas de la cuadricula que corresponde a cada parroquia.	

Condiciones de ejecución: Ninguna
Entrada: - El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción “Editar mapa”. - Escoge el botón de la parroquia a delimitar - El usuario da clic en la cuadricula del mapa de Riobamba que considere que pertenece a la parroquia. - El usuario tiene la opción de bloquear zonas de color rojo, que no son hábiles. - El usuario da clic en el botón guardar, para almacenar los cambios realizados - El usuario tiene la opción de continuar el trabajo realizado posteriormente, dando clic en el botón cargar. - El sistema almacenará el avance en un archivo Json, para su posterior carga.
Resultado esperado: Visualizar interfaz para escoger la parroquia. pintar con colores predeterminados, usando el botón bloquear para pintar de color rojo todas las zonas a bloquear. Se guarda y carga el avance en un archivo Json.
Evaluación de la prueba: Prueba satisfactoria.

PRUEBA FUNCIONAL	
Número Prueba: 3	**Historia de Usuario 3**
Nombre de la Prueba: Ingreso de puntos de atención existentes en el mapa de Riobamba	
Descripción: El usuario, al seleccionar la opción del menú “Editar mapa”, encontrará el botón “Edificio”, con esta opción el usuario podrá escoger la posición de la cuadrícula para ubicar un edificio fijo manualmente. En caso de ubicarlo mal, el usuario podrá quitar dicho edificio.	
Condiciones de ejecución: Ninguna	
Entrada: - El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción “Editar mapa”.	

- Escoge el botón “Edificio”, con el check box activo.
- El usuario determina la posición y da clic para ubicar el edificio, en la cuadrícula.
- En caso de ubicar mal el edificio, escoger el botón “Edificio” con el check box desactivado, y dar clic en el edificio mal ubicado.
- El usuario da clic en el botón guardar, para almacenar los cambios realizados
- El usuario tiene la opción de continuar el trabajo realizado posteriormente, dando clic en el botón cargar.
- El sistema almacenará el avance en un archivo Json, para su posterior carga.

Resultado esperado: En el mapa de Riobamba se da la opción de ubicar los puntos de atención mediante un clic, y a la vez se da la opción de borrar mediante un clic si el punto fue ubicado erróneamente.

Evaluación de la prueba: Prueba satisfactoria.

PRUEBA FUNCIONAL	
Número Prueba: 5	**Historia de Usuario 5**

Nombre de la Prueba: Visualización de las posibles nuevas ubicaciones de puntos de atención

Descripción: El usuario, al seleccionar la opción del menú “Solucionar”, se le brinda la opción de ingresar los parámetros de distancia entre puntos de atención, el total de puntos de atención y la distribución por parroquias del número de puntos de atención.

Condiciones de ejecución: Ninguna

Entrada:

- El usuario ingresa al sistema
- Aparece el menú principal y escoge la opción “Solucionar”.
- El usuario ingresa la cantidad máxima de puntos de atención deseados.
- El usuario ingresa la distancia entre puntos de atención
- El usuario ingresa la distribución por parroquia del número de puntos de atención en cada uno de estos.
- El sistema procesa estos datos y empezará las iteraciones del algoritmo genético, para encontrar la solución óptima.
- El sistema mostrará en la cuadricula del mapa de la ciudad de Riobamba, las ubicaciones de los puntos de atención generadas por el algoritmo genético.

Resultado esperado: Visualizar en el mapa de Riobamba, los resultados de las ubicaciones que arroja el algoritmo genético, mediante iconos representativos.
Evaluación de la prueba: Prueba satisfactoria.

PRUEBA FUNCIONAL	
Número Prueba: 5	**Historia de Usuario 5**
Nombre de la Prueba: Modificar estado de los puntos de atención generados por el algoritmo genético	
Descripción: El usuario, al escoger la opción del menú "Simular", el sistema le dará la opción de dar clic en los puntos de atención creados por el sistema, para deshabilitarlos y cambiándolos a color negro. El usuario podrá habilitarlos al dar clic nuevamente.	
Condiciones de ejecución: Ninguna	
Entrada: - El usuario ingresa al sistema. - Aparece el menú principal, y escoge la opción "Simular". - El usuario dará clic en el punto de atención que se desee deshabilitar, o habilitar de nuevamente. - El sistema cambiará a color negro los puntos de atención deshabilitados, y color blanco los puntos de atención habilitados.	
Resultado esperado: Cambiar de estado de habilitado y deshabilitado, de cada uno de los nuevos puntos de atención sugeridos.	
Evaluación de la prueba: Prueba satisfactoria.	

PRUEBA FUNCIONAL	
Número Prueba: 6	**Historia de Usuario 6**
Nombre de la Prueba: Visualización de la afluencia de usuarios a los puntos de atención	

Descripción: El usuario, al escoger la opción del menú "Simular", el sistema le dará la opción de ingresar los parámetros de cantidad total de población a simular en tiempo real, el tiempo en segundos entre intervalos, y la cantidad de punteros por intervalo.

Condiciones de ejecución: Ninguna

Entrada:

- El usuario ingresa al sistema.
- Aparece el menú principal, y escoge la opción "Simular".
- El usuario ingresará el tiempo en segundos, que simulará la salida de los punteros hacia los puntos de atención.
- El usuario ingresará la cantidad por intervalos que aparecerán en la simulación.
- El usuario ingresará la cantidad total que simulará a la población que será atendida.
- El sistema procesará la información, y empezará la simulación mostrando en tiempo real la afluencia de los punteros a los puntos de atención.

Resultado esperado: Visualizar en tiempo real la afluencia y la toma de decisiones de los usuarios a los distintos puntos de atención habilitados en el Mapa de Riobamba.

Evaluación de la prueba: Prueba satisfactoria.

PRUEBA FUNCIONAL	
Número Prueba: 7	**Historia de Usuario 7**

Nombre de la Prueba: Generar archivo Json, con las ubicaciones de los puntos de atención, y resultados de la simulación.

Descripción: El usuario, al escoger la opción del menú "Editar mapa", el sistema le dará la opción de guardar los datos de la simulación al dar clic en el botón guardar.

Condiciones de ejecución: Ninguna

Entrada:

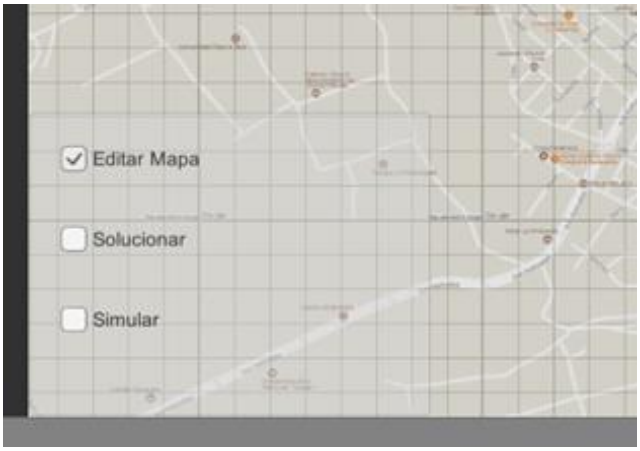
- El usuario ingresa al sistema.
- Aparece el menú principal, y escoge la opción "Editar mapa".
- El usuario dará clic en el botón "Guardar"
- El sistema almacenará los datos resultantes de la simulación en un archivo de tipo Json.

Resultado esperado: En un archivo de tipo Json se guarde las ubicaciones con sus respectivos datos de los puntos de atención generados, y los datos de la simulación.
Evaluación de la prueba: Prueba satisfactoria.

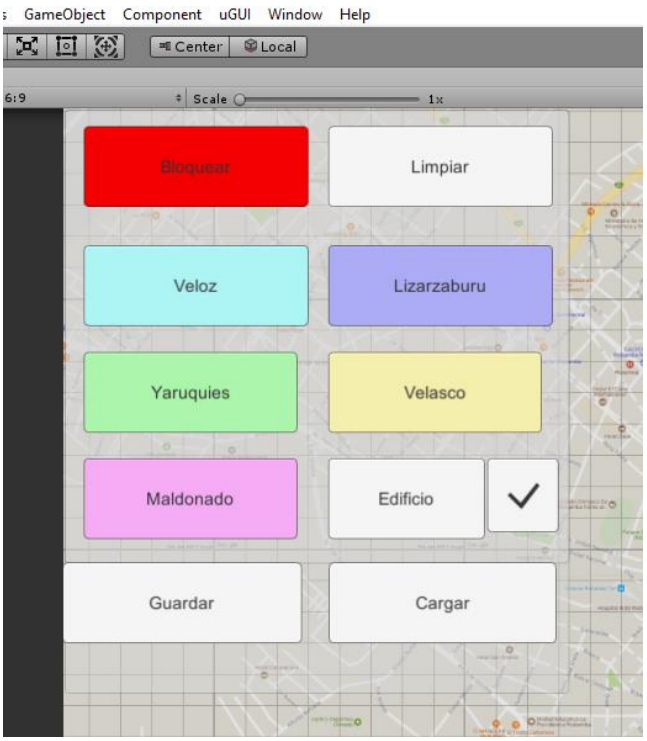
APÉNDICE C. Manual de Usuario

Prerrequisitos: La aplicación software para iniciar necesita conexión a internet, para poder cargar el mapa de Riobamba, usando Google maps.

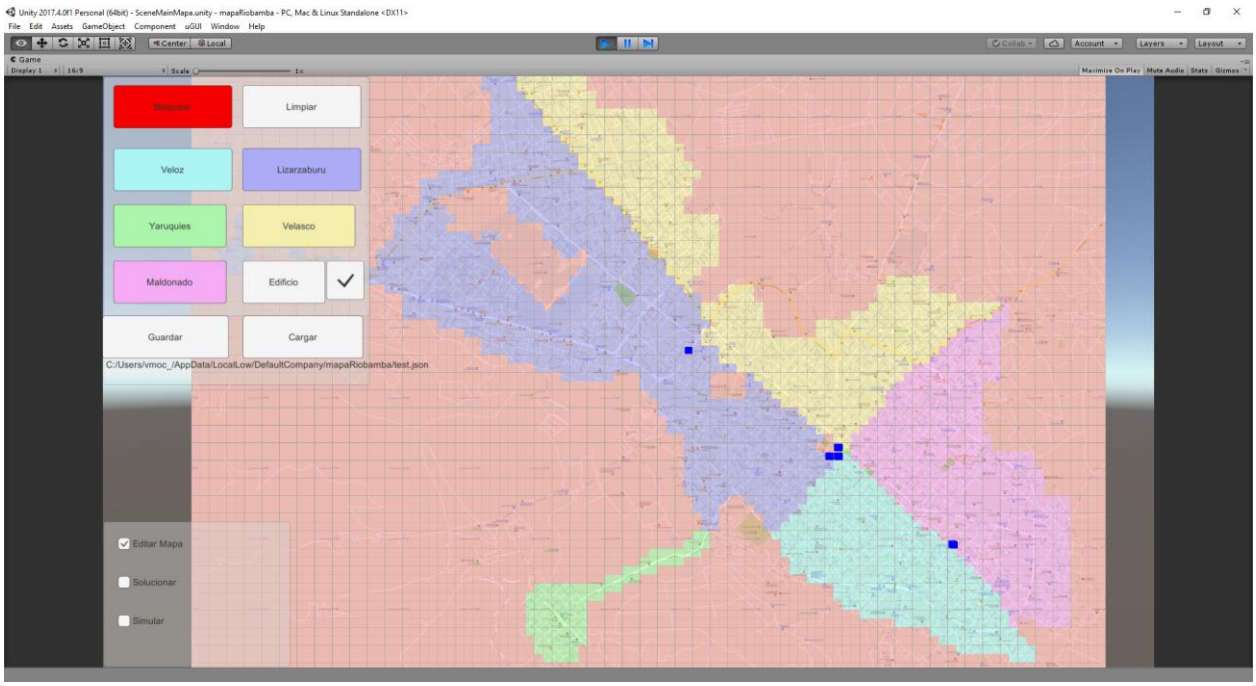
Paso 1. En el menú inferior izquierdo escoger la opción “Editar Mapa”.



Paso 2. Escoger la parroquia deseada y pintar en el mapa dando clic izquierdo o arrastrando el mismo en las zonas del mapa de Riobamba. A la vez escoger la opción “Bloquear” para bloquear sectores. Si se comete algún error escoger la opción “Limpiar”. La opción “Edificio” permite ubicar los puntos de atención ya existentes en el mapa. Todas estas opciones se encuentran el menú superior izquierdo.

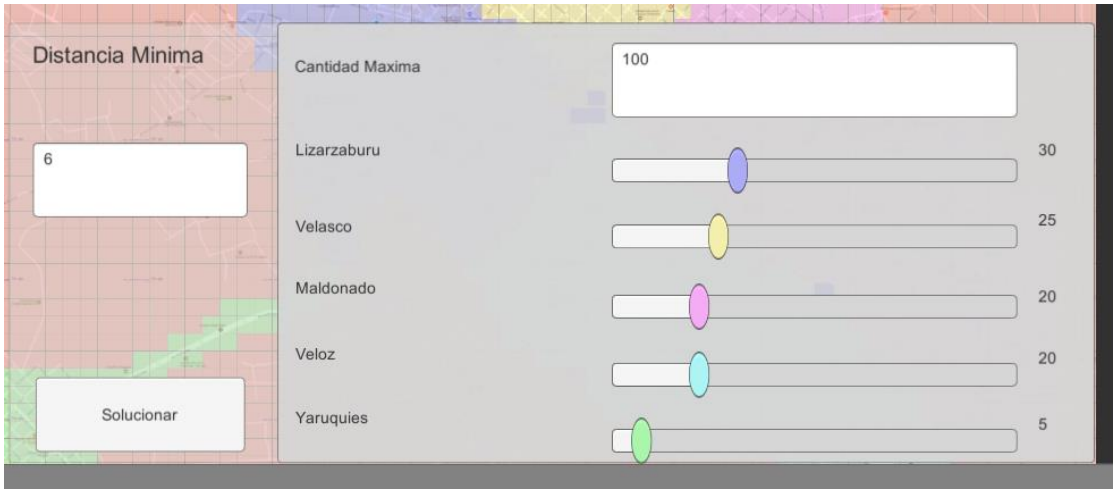


Paso 3. Al terminar de pintar y bloquear todas las parroquias y secciones del mapa de Riobamba. Dar clic en la opción “Guardar”. La opción “Cargar” permite precargar un mapa donde ya se realizó los pasos 1 y 2, este mapa precargado se puede modificar y guardar los cambios. Nota: Evitar dar clic la opción “Guardar” si el mapa está vacío, debido a que guardará y sobrescribirá la información de un mapa pre cargado.

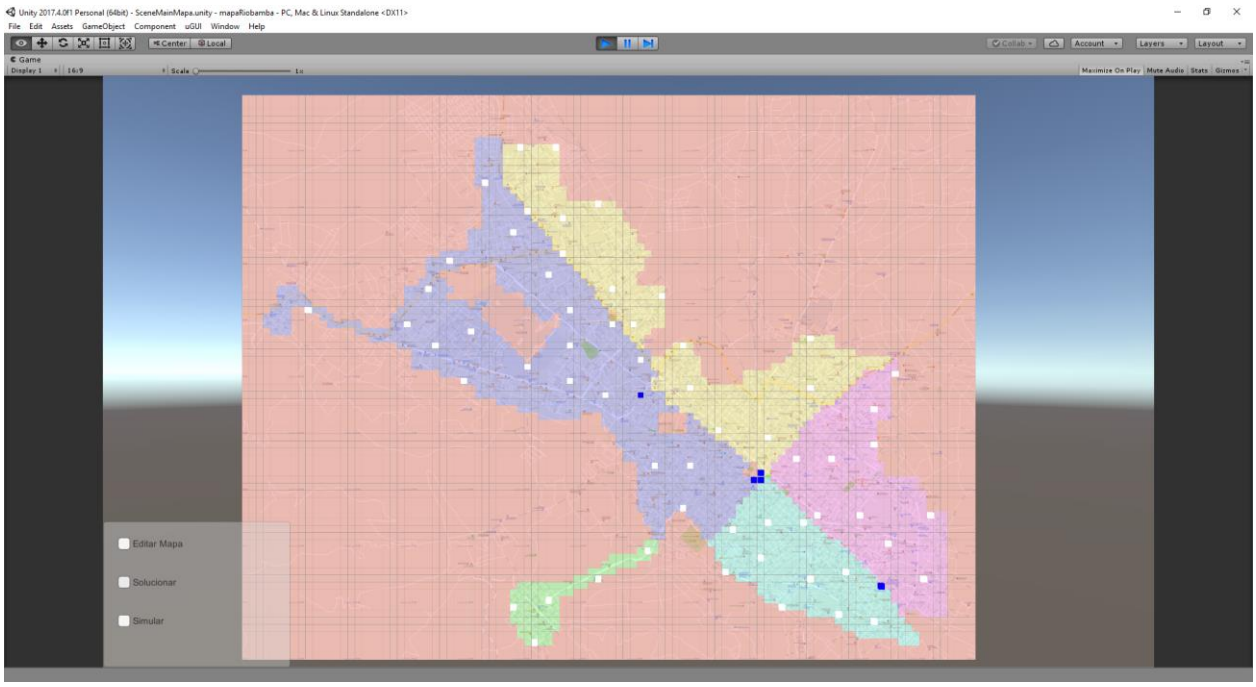


Paso 4. En el menú inferior izquierdo escoger la opción “Solucionar”.

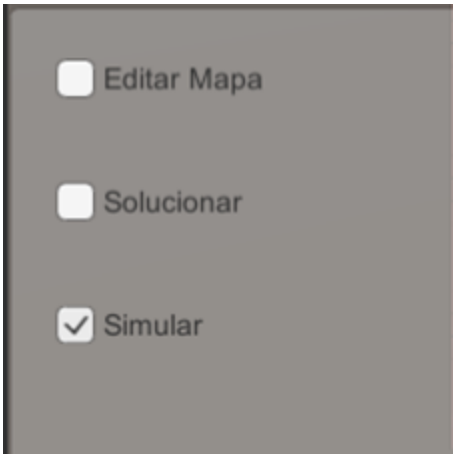
Paso 5. Ingresar en el campo “Distancia mínima” la distancia de separación entre edificios en cada parroquia. Ingresar el máximo de edificios y la distribución en cada parroquia. Todo esto se encuentra en la parte inferior derecha de la pantalla.



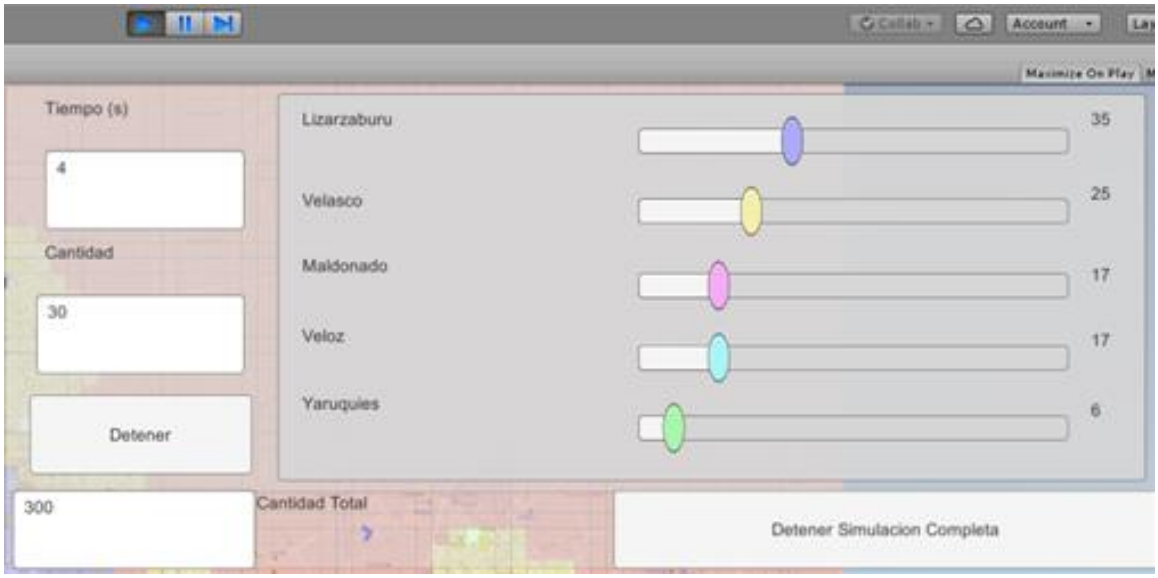
Paso 6. Dar clic en el botón “Solucionar” para iniciar la aplicación del algoritmo genético, se puede observar en tiempo real la ubicación de los edificios en tiempo real en cada parroquia. Se recomienda esperar a que todos los edificios sean ubicados para proseguir con la simulación.



Paso 7. En el menú inferior izquierdo escoger la opción “Simular”.

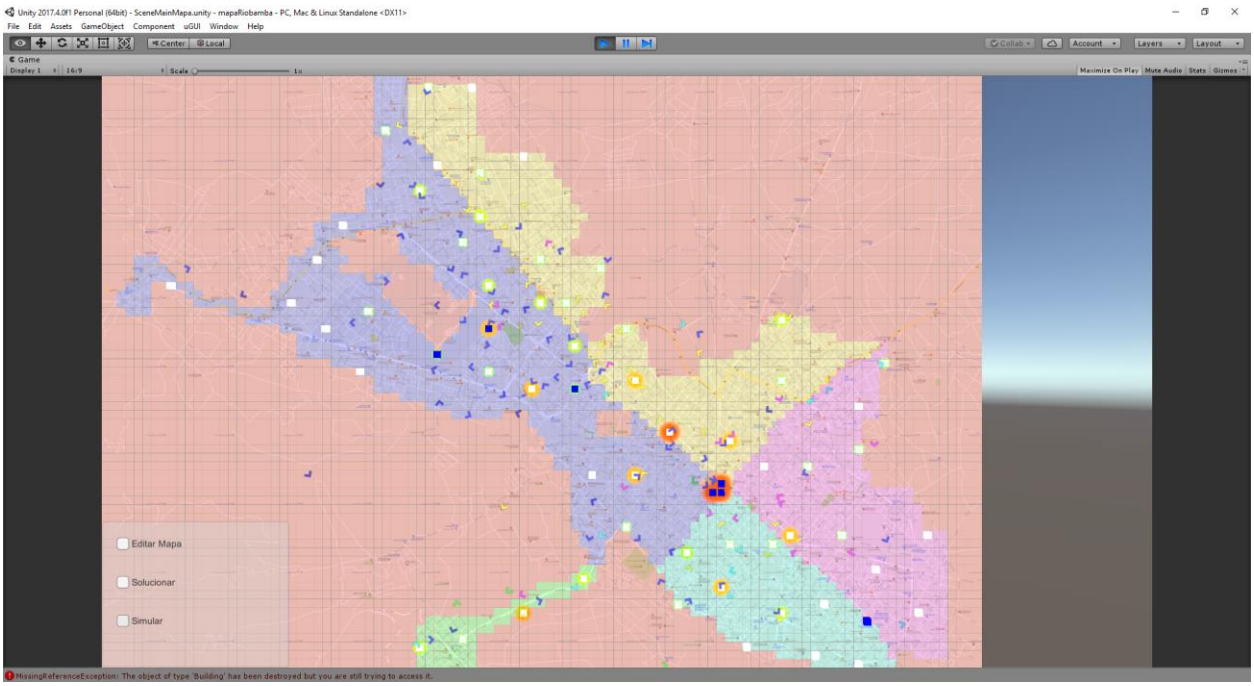


Paso 8. Ingresar los campos requeridos para la simulación. Campo “Tiempo” en segundos que un usuario representado por una flecha permanece en los edificios. Campo “Cantidad” el número de usuarios que aparece en el tiempo ingresado. Campo “Cantidad Total” el total de usuarios a simular. Y escoger el porcentaje de población de cada parroquia.



Paso 9. Dar clic en el botón “Simular” para iniciar la simulación. Se puede observar la generación de los usuarios a los edificios generados por el algoritmo genético. En la simulación los usuarios se dirigen a los edificios por preferencia de distancia más corta o preferencia randómica por congestión.

Se puede bloquear edificios por medio de un clic izquierdo y tomarán color negro, para que los usuarios no acudan a tal edificio, lo que permite visualizar los comportamientos de afluencia en cada edificio habilitado. Para desbloquear los edificios dar clic izquierdo nuevamente, y regresaran al color blanco.



Nota: Para ocultar los paneles de parametrización de las opciones del menú de la parte inferior izquierda, tan solo desactivar la opción activa, por medio del clic izquierdo. Lo que permite visualizar el mapa sin obstrucciones visuales causadas por los paneles de ingreso de datos.

Referencias

Alba Torres, E. (1999). *Análisis y Diseño de Algoritmos Genéticos Paralelos Distribuidos.* Obtenido de Tesis Doctoral: http://neo.lcc.uma.es/tesis/PhD-Alba99.pdf

Arranz de la Peña, J., & Parra, A. (2015). *Algoritmos Genéticos.* Obtenido de http://www.it.uc3m.es/~jvillena/irc/practicas/06-07/05.pdf

ASAP. (2015). *Introducción a los Algoritmos Evolutivos.* Obtenido de Automated Scheduling Optimization Planning: http://www.exa.unicen.edu.ar/escuelapav/cursos/bio/l2.pdf

Camarillo Peñaranda, J. R., Ramírez Quiroz, F. A., González Montoya, D., Bolaños Martínez, F., & Ramos-Paja, C. A. (2015). Reconfiguration of photovoltaic arrays based on genetic algorithm. *Revista Facultad de Ingeniería Universidad de Antioquia*, 75.

Crevillén , G., & Díaz, D. (2012). *Función de evaluación y selección.* Obtenido de http://www.depi.itch.edu.mx/apacheco/expo/html/ai14/ga.html#page13

Floreano, D., & Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies.* Massachusetts: Massachusetts Institute of Technology.

Kawther, T. (2016). Edge Detection For Medical Images Using Threshold Based. *Al-Mustnariyh University - Magistra No. 96 Th. XXVIII*, 114-119.

Manonmani , A. (2017). Evolutionary Algorithm-Based Multi-objective. *International Journal of Food Engineering*.

Masmoudi, M., Brakers, K., & Masmoudi, M. (2017). A hybrid genetic algorithm for the heterogeneous dial-a-ride problem. *Computers & operations research, 81*, 1-13.

Mejía, F. (2012). *Algotimos genéticos.* Obtenido de http://nando1-utb.blogspot.com/p/algoritmos-geneticos.html

Moradi, M., & Abedini, M. (2012). A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems. *Electrical Power and Energy Systems 34*, 66-74.

Norvig, P. (2015). *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp.* Obtenido de Google Books: https://books.google.com.ec/books?id=eH6jBQAAQBAJ&dq=Inteligencia+Artificial+Peter+Norvig+algoritmos+geneticos&q=genetic+Algorithms#v=snippet&q=genetic%20Algorithms&f=false

Peña Arenas, I. G., & López Castro, L. F. (2016). Algoritmo genético para solucionar el problema de dimensionamiento y programación de lotes con costos de alistamiento dependientes de la secuencia. *INGENIERIA Y DESARROLLO*, vol 34. no.1.

Popescu, D., Bold, N., & Nijloveanu, D. (2016). A Method Based on Genetic Algorithms for Generating Assessment Tests Used for Learning. *Polibits*, PB-54-7.

Rathi, S., & Gupta, R. (2017). Optimal sensor locations for contamination detection in pressure-deficient water distribution networks using genetic algorithm. *Urban Water Journal*, 160-172.

Rojas Hernández, J. (2014). *Algoritmos Genéticos.* Obtenido de https://www.academia.edu/9605579/ALGORITMOS_GEN%C3%89TICOS

Sazzadu, M., Mukit, M. A., & Bikas, M. A. (2012). AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM. *International Journal of Network Security & Its Applications (IJNSA)*, Vol.4, No.2.

Sebt, M., Afshar, M., & Alipouri, Y. (2016). Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem. *Engineering Optimization*, 1-18.

Svennerberg, G. (2010). *Beginning Google Map API 3.* New York: Springer Science+Business Media, LLC.

Resumen Final

Implementación de algoritmos genéticos para el estudio de afluencia y descongestionamiento de usuarios en empresas públicas en la ciudad de Riobamba

Víctor Manuel Oquendo Coronado

67 páginas

Proyecto dirigido por: José Marcelo Balseca Manzano, Mg.

El problema radica en la desproporción a las oficinas municipales existentes de la ciudad de Riobamba, por parte de la ciudadanía, generando molestias a los usuarios, y afectando a las recaudaciones municipales en los meses de congestión. Para el estudio de planificación de nuevos puntos de atención a los usuarios en la ciudad, serán asistidos por la aplicación software aplicando algoritmos genéticos para determinar las posibles ubicaciones de los nuevos puntos de atención a los usuarios en el mapa urbano de Riobamba, además se visualiza la posible distribución de congestión en tiempo real a los nuevos puntos de atención. El algoritmo genético utilizado aplica las etapas de un algoritmo genético simple las cuales son: evaluación de la población, selección, cruces o recombinación y mutación, para encontrar la solución óptima. Para la resolución del problema, se creó vectores asociativos por parroquias de la ciudad de Riobamba, cada vector contiene si existe o no un punto de atención usando 0 o 1, y la posición referenciada a una matriz con fila y columna del mapa de Riobamba el cual fue instanciado mediante Google maps.