

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

**MAESTRÍA EN SISTEMAS DE INFORMACIÓN, MENCIÓN
INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS**

**UTILIZACIÓN DE ALGORITMOS GENÉTICOS PARA LA IDENTIFICACIÓN DE
LA MEJOR UBICACIÓN DE CENTROS DE CABLEADO ESTRUCTURADO EN
PLANOS ARQUITECTÓNICOS**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN
SISTEMAS DE INFORMACIÓN MENCIÓN EN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA
DE DATOS MASIVOS**

DIEGO ALBERTO IZA VITERI

diego.izaviteri@epn.edu.ec

Director: Fredy Marcelo Gavilanes Sagnay, Msc.

fredy.gavilanes@epn.edu.ec

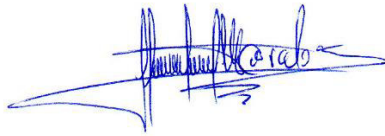
Codirector: Edison Fernando Loza Aguirre, PhD.

edison.loza@epn.edu.ec

2021

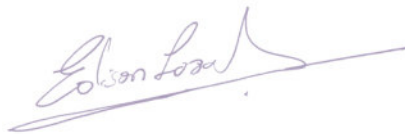
APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación “Utilización de algoritmos genéticos para la identificación de la mejor ubicación de centros de cableado estructurado en planos arquitectónicos.” desarrollado por Diego Alberto Iza Viteri estudiante de la Maestría de Sistemas de Información mención Inteligencia de Negocios y Analítica de Datos Masivos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.



Fredy Gavilanes Sagnay, Msc.

DIRECTOR



Edison Loza Aguirre, PhD

CODIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Diego Alberto Iza Viteri declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Diego Alberto Iza Viteri

ÍNDICE DE CONTENIDO

| | |
|--|-----|
| ÍNDICE DE CONTENIDO | iv |
| LISTA DE FIGURAS | vii |
| LISTA DE TABLAS | ix |
| RESUMEN..... | x |
| ABSTRACT | xi |
| 1 INTRODUCCION..... | 12 |
| 1.1 Planteamiento del problema | 12 |
| 1.2 Objetivos | 13 |
| 1.2.1 Objetivo General..... | 13 |
| 1.2.2 Objetivo Específico..... | 13 |
| 1.3 Alcance..... | 13 |
| 1.4 Marco Teórico | 14 |
| 1.4.1 Cableado Estructurado | 14 |
| 1.4.2 Diseño asistido por computador..... | 17 |
| 1.4.3 Diseño arquitectónico asistido por computadora | 18 |
| 1.4.4 Inteligencia Artificial | 18 |
| 1.4.5 Búsqueda A* | 23 |
| 1.5.6 Algoritmos genéticos | 26 |
| 1.4.7 Aprendizaje por refuerzo | 29 |
| 1.4.8 Relación entre Algoritmos genéticos y el aprendizaje por refuerzo..... | 30 |
| 2 METODOLOGIA | 32 |
| 2.1 Ciencia del diseño..... | 32 |
| 2.1.1 Entorno | 32 |
| 2.1.2 Base de conocimientos | 33 |
| 2.1.3 Investigación | 33 |
| 2.2 Uso del framework de Ciencias del diseño..... | 33 |
| 2.3 Pruebas | 35 |
| 2.3.1 Pruebas funcionales..... | 35 |
| 2.3.2 Pruebas de desempeño | 35 |
| 3 RESULTADOS..... | 36 |
| 3.1 Diseño de la aplicación | 36 |
| 3.1.1 Componentes de la aplicación | 36 |
| 3.1.2 Diagramas de casos de uso | 37 |

| | | |
|-------|---|----|
| 3.1.3 | Diagrama de secuencia y colaboración..... | 38 |
| 3.1.4 | Diagrama de clases | 39 |
| 3.1.5 | Interfaz de usuario..... | 42 |
| 3.2 | Algoritmos utilizados por la aplicación | 44 |
| 3.2.1 | Pseudocódigo Algoritmo de búsqueda A* | 44 |
| 3.2.2 | Pseudocódigo Algoritmo genético..... | 44 |
| 3.2.3 | Pseudocódigo Algoritmo de carga al entorno virtual de aprendizaje..... | 44 |
| 3.3 | Funcionamiento de la aplicación..... | 44 |
| 3.4 | Visualización del resultado dentro de un EVA | 45 |
| 3.5 | Casos de utilización..... | 46 |
| 3.5.1 | Caso I – Cálculo de rutas sin optimización de resultados | 46 |
| 3.5.2 | Caso II – Cálculo de rutas con optimización de resultados | 49 |
| 3.5.3 | Caso III – Cálculo de rutas con optimización de resultados e inclusión de variables de tipo costo. | 52 |
| 4 | CONCLUSIONES Y RECOMENDACIONES | 56 |
| 5 | REFERENCIAS BIBLIOGRÁFICAS | 58 |
| 6 | ANEXOS..... | 60 |
| | Anexo I..... | 60 |
| | Anexo II..... | 62 |
| | Anexo III..... | 64 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1. Representación de los subsistemas de la norma EIA/TIA-568 | 16 |
| Figura 2. Representación gráfica de búsqueda por anchura..... | 20 |
| Figura 3. Representación gráfica de búsqueda en profundidad..... | 20 |
| Figura 4. Representación gráfica de búsqueda de costo uniforme | 21 |
| Figura 5. Representación gráfica de búsqueda bidireccional..... | 21 |
| Figura 6. Representación gráfica de búsqueda voraz primero el mejor. | 22 |
| Figura 7. Representación gráfica de búsqueda A*, ruta a seguir para llegar desde nodo salida a nodo objetivo | 22 |
| Figura 8. Diagrama de flujo de un algoritmo genético | 23 |
| Figura 9. Ejemplo de búsqueda A* | 24 |
| Figura 10. Paso 1, búsqueda A* | 25 |
| Figura 11. Paso 2, búsqueda A* | 25 |
| Figura 12. Paso 3, búsqueda A* | 26 |
| Figura 13. Nodos de la ruta solución, búsqueda A* | 26 |
| Figura 14: Codificación de elementos de un algoritmo genético | 27 |
| Figura 15: Selección de individuos de una población | 28 |
| Figura 16: Mecanismo de cruce..... | 28 |
| Figura 17: Mecanismo de mutación | 29 |
| Figura 18: Problema básico de aprendizaje por refuerzo. | 30 |
| Figura 19: Marco de trabajo de la ciencia del diseño. | 32 |
| Figura 20. Calcular mejor punto de entrada por cuarto | 37 |
| Figura 21. Calcular mejor ubicación cuarto de comunicaciones. | 38 |
| Figura 22. Calcular mejor punto de entrada por cuarto | 38 |
| Figura 23. Calcular mejor ubicación de cuarto de comunicaciones | 39 |
| Figura 24: Diagrama UML del aplicativo AppBestCommsRoom..... | 39 |
| Figura 25. Pantalla principal del aplicativo..... | 42 |
| Figura 26. Elementos de la sección 1. Paleta de marcadores | 42 |
| Figura 27. Elementos de la sección 2. Botones de acción..... | 43 |
| Figura 28. Elementos de la sección 3. Representación del piso | 43 |
| Figura 29. Elementos de la sección 4. Mensajes del aplicativo. | 43 |
| Figura 30. Estructura de archivo JSON resultado | 45 |
| Figura 31. Representación del proyecto final dentro un EVA | 46 |
| Figura 32. Representación gráfica del piso para el caso 1 | 47 |
| Figura 33. Resultado de ejecutar Paso1, proceso de cómputo de mejor punto de entrada mediante algoritmo A estrella..... | 47 |

| | |
|---|----|
| Figura 34. Resultado de ejecutar Paso2, proceso de cómputo de mejor ubicación del gabinete de telecomunicaciones mediante algoritmo genético..... | 48 |
| Figura 35. Total por proyecto | 48 |
| Figura 36. Representación gráfica de caso I y caso II..... | 50 |
| Figura 37. Resultado de “Calcular Puntos de Entrada” | 50 |
| Figura 38. a) Resultado de “Calcular Gabinete de Comms”. b) Total por proyecto | 51 |
| Figura 39. a) Resultado gráfico luego de optimizar cálculos. b) Total por proyecto | 51 |
| Figura 40. Representación gráfica del piso para Caso 3..... | 53 |
| Figura 41. Resultado de ejecución del aplicativo con 2.00 USD y 1.15 USD de costo por metro de bandeja y tubería respectivamente | 53 |
| Figura 42. Resultado de ejecución del aplicativo con 4.00 USD y 1.15 USD de costo por metro de bandeja y tubería respectivamente | 54 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1. Framework utilizado, y relación entre objetivo-etapa-actividad. | 34 |
| Tabla 2. Resumen de los eventos estudiados en el caso I..... | 49 |
| Tabla 3. Resumen de los eventos estudiados en el caso II..... | 52 |
| Tabla 4. Resumen de los eventos estudiados en el caso III..... | 55 |

RESUMEN

Elegir la ubicación del centro de cableado estructurado de un piso, es una tarea que normalmente se desarrolla de forma empírica, aplicando de primera mano la experiencia y trabajos previos que hayan permitido formar una mejor idea a quien debe desarrollar esta tarea. El presente proyecto de desarrollo permite evaluar la mejor ubicación física de centros de cableado en función de estándares, y complementando los cálculos con la utilización del algoritmo de búsqueda A* y algoritmos genéticos. Adicionalmente este proceso considera el uso de variables monetarias que permiten maximizar los recursos con los que se lleva a cabo un proyecto de cableado estructurado. El resultado de este proceso es observado en un entorno virtual de aprendizaje, lo que permite visualizar de una manera interactiva el proyecto final. El análisis de los casos expuestos permitirá comprender el resultado de las diferentes ejecuciones cuando se interactúa con el software.

Palabras clave: Algoritmos genéticos, Búsqueda A*, Cableado Estructurado, Inteligencia Artificial, A Estrella

ABSTRACT

Choosing the location of the structured cabling center of a floor is a task that is normally developed empirically, applying first-hand experience and previous work that have allowed a better idea of who should develop this task. This development project allows evaluating the best physical location of cabling centers based on standards, and complementing the calculations with the use of the A * search algorithm and genetic algorithms. Additionally, this process considers the use of monetary variables that allow maximizing the resources with which a structured cabling project is carried out. The result of this process is o

bserved in a virtual learning environment, which allows the final project to be viewed in an interactive way. The analysis of the exposed cases will allow to understand the result of the different executions when interacting with the software.

KEYWORDS: Genetic Algorithms, A * Search, Structured Cabling, Artificial Intelligence, A star

1 INTRODUCCION

1.1 Planteamiento del problema

Debido a la coyuntura actual que vivimos en nuestro planeta, varias instituciones educativas se han visto en la necesidad de adaptarse y dar un giro sobre los métodos habituales y aceptados que les permiten desarrollar sus actividades pedagógicas de manera normal. Ingresar a salas virtuales para recibir el contenido de una materia o asignatura, actualmente, es un recurso ampliamente utilizado a todo nivel educativo. Estas salas permiten que tanto alumnos como profesores interactúen simulando estar presentes en un salón de clases [1].

Sin embargo, para mantener al alumnado ávido de conocimiento es necesario contar siempre con componentes que generen valor agregado a estas interacciones. Es aquí donde juegan un papel preponderante los Entornos Virtuales de Aprendizaje (EVA), o Virtual Learning Environment (VLE) por sus siglas en inglés [2]. Los EVA son un instrumento eficaz dentro de los procesos educativos basados en la comprensión de los comportamientos expresados por profesor-alumno y alumno-alumno [3]. Permiten elevar y mejorar la comunicación e interacción entre los actores del aula de clases, a la vez que promueven la adquisición de conocimiento y capacidades espaciales basadas en experimentos, y entregan también un instrumento potente para el control de actividades y procesos educativos [3].

En las asignaturas en las cuales el proceso de aprendizaje se basa en actividades prácticas, talleres y capacidades de diseño, los EVA facilitan la integración de componentes desarrollados por terceros [1], [2]. Para nuestro caso, dentro de materias relacionadas con redes de información, y específicamente con instalaciones de Cableado Estructurado podríamos simular, por ejemplo, dentro de un EVA, la instalación de los diferentes componentes de estos sistemas de comunicaciones, entre los cuales contamos con ubicar al cuarto o centro de cableado de un piso en un determinado sitio. Esta última tarea podría ser resuelta por el estudiante de muchas formas, es decir existen de primera mano un amplio número de soluciones que podrían ser implementadas de forma correcta, errónea u óptima.

Para definir al tipo de problema que deseamos solucionar, decimos que: un problema *Well-defined* básicamente tiene una o varias soluciones, las mismas que son conocidas y correctas, y el camino a la solución se sustenta en seguir un procedimiento o protocolo [4]. Para hablar de un problema *well-defined* se deben cumplir con las siguientes condiciones: 1) se dispone de un

estado inicial; 2) existe un número limitado de reglas de transformación relativamente fáciles de formalizar; 3) se especifican las funciones de evaluación y 4) el objetivo no es ambiguo [5]. En cambio los problemas *Ill-defined* son aquellos cuyas soluciones pueden ser múltiples y diversas, y están definidos en función de los supuestos, evidencias u opiniones que se levanten [4]. También se consideran *Ill-defined* a aquellos que violen una o más de las condiciones de los problemas *well-defined* [5]. En nuestro caso, podríamos situar el centro de cableado en tantas ubicaciones como se nos ocurriría dentro de un plano, al igual que el tendido de cable podría seguir una multitud de caminos, estos escenarios terminarían rompiendo al menos la condición 2), por lo cual este caso cae en el espacio de problemas *Ill-defined*.

Por las razones descritas, en el presente proyecto de desarrollo se propone encontrar la solución óptima para la ubicación de un centro de cableado, mediante la utilización de algoritmos genéticos. El resultado obtenido podrá ser consumido por un VLE que tenga el fin de guiar a los estudiantes en el proceso de enseñanza-aprendizaje de la asignatura de cableado estructurado, y que ya ha sido desarrollado en trabajos previos [6].

1.2 Objetivos

El presente trabajo persigue los siguientes objetivos:

1.2.1 Objetivo General

Identificar la mejor ubicación de los centros de cableado estructurado en planos arquitectónicos mediante la utilización de algoritmos genéticos.

1.2.2 Objetivo Específico

- Modelar la problemática de ubicación de un centro de cableado estructurado para obtener la función de fitness correspondiente.
- Implementar el algoritmo genético y ajustar los parámetros del algoritmo para obtener la mejor solución posible.
- Evaluar los resultados y su aplicabilidad en un VLE.

1.3 Alcance

El presente trabajo busca encontrar la mejor ubicación de un centro de cableado en pisos o plantas de edificios nuevos o en construcción, en los cuales se instale cielo falso que facilite la

colocación de bandejas y tuberías para cableado estructurado, mediante la carga de la imagen que representa al piso, la distribución de paredes, y la ubicación de los puntos de red en formato 2D. El programa permitirá exportar los resultados a un entorno 3D.

La implementación de la solución no considera opciones de perforación de paredes de concreto, atravesar columnas, rotura de normas TIA/EIA, ni lectura de imágenes representativas de un piso en formato 3D.

1.4 Marco Teórico

1.4.1 Cableado Estructurado

- **Definición**

El cableado estructurado es todo el sistema de elementos de una red de medios mixtos, que aplicados con ingeniería se unen para dar soporte a toda la información, señales de voz y datos que se transmite por dichos elementos, maximizando sus principales características y minimizando los defectos [7].

- **Ventajas**

- **Consistencia:** los elementos de red utilizados dan soporte a los mismos sistemas de voz, datos y vídeo.
- **Soporte Multi-vendedor:** es posible utilizar e integrar dispositivos de diferentes proveedores.
- **Simplificar aplicación de cambios/adiciones:** un sistema de cableado estructurado permite cambiar o actualizar su configuración.
- **Simplificar la resolución de problemas:** encontrar problemas relacionados con la red es más fácil, debido a que distintos segmentos de red se encuentran aislados y reconocidos dentro del sistema.
- **Soporte para aplicaciones futuras:** un sistema de cableado estructurado admite nuevas aplicaciones desarrolladas que pueden ejecutarse sobre los elementos que ya se encuentran instalados [7].

- **Elementos**

- **Cableado Vertical:** También conocido como *backbone*, tiene como tarea principal conectar los armarios o cuartos de telecomunicaciones que se encuentran en un edificio.
- **Cableado Horizontal:** Es el cableado que se encarga de conectar el cuarto de telecomunicaciones o cuarto de cableado con los puntos de red de un mismo piso.
- **Cuarto de Equipos:** En este espacio, se colocan los equipos grandes que tienen relación con los sistemas de telecomunicaciones del piso [8], [9].
- **Cuarto de Telecomunicaciones:** Aquí se colocan los dispositivos que permiten centralizar las conexiones, por ejemplo *racks* o closets. A estos elementos llegarán los cables que conectan cada punto de red con los diferentes *hubs* de conexiones [8], [9].
- **Área de trabajo:** Comprende los puntos de red dentro de un área determinada, por ejemplo un cuarto de gerencia, una sala de reuniones, un área de desarrollo y *testing* de software, etc.[8], [9].

- **Normas y estándares de cableado estructurado**

El objetivo principal de las normas y estándares, es regular y proporcionar las mejores prácticas empleadas para el diseño e instalación de sistemas de cableado estructurados. Su adecuada implementación posibilita que servicios de video, voz y datos existentes puedan ser soportados con un buen nivel de rendimiento, y también da la posibilidad de soportar servicios que se desarrollen a futuro y que tengan como base los estándares actuales [9].

Norma EIA/TIA-568

Esta norma define los requisitos mínimos que deben ser implementados en sistemas de cableado estructurados para edificios comerciales, y se divide en:

- EIA/TIA-568.0-D, establece los requerimientos generales.
- EIA/TIA-568.C-2, componentes de cableado utilizando par trenzado.
- EIA/TIA-568.3-D, componentes de cableado utilizando fibra óptica.
- EIA/TIA-568.4-D, componentes de cableado utilizando cable coaxial.

Los subsistemas de esta norma incluyen:

1. Acometida principal, normalmente se refieren a la instalación entregada por el proveedor público o privado del servicio de telecomunicaciones.
2. Cuarto de comunicaciones, en esta área se ubican los equipos de telecomunicaciones, habitualmente están ligados o cerca de las acometidas principales.
3. Eje de cableado central, es la bandeja principal por donde recorre todo el sistema de cables que alimentan a los puntos de red, concentradores de red, etc.
4. Gabinete de telecomunicaciones, alberga y ordena a los cables de distribución horizontal.
5. Cableado horizontal, es el medio físico utilizado para conectar a los elementos de la red.
6. Área de trabajo, son cada uno de los cuartos que se encuentran definidos o separados en función de las características de trabajo que comparten determinados puntos de red, por ejemplo podemos tener un área o cuarto de trabajo destinado a una gerencia, un call center de atención al cliente, un call center de agendamiento de citas, etc.
7. Cableado de *backbone*, es el cableado utilizado para conectar elementos de forma vertical, por ejemplo acometida, cuarto de comunicaciones y gabinete de telecomunicaciones.

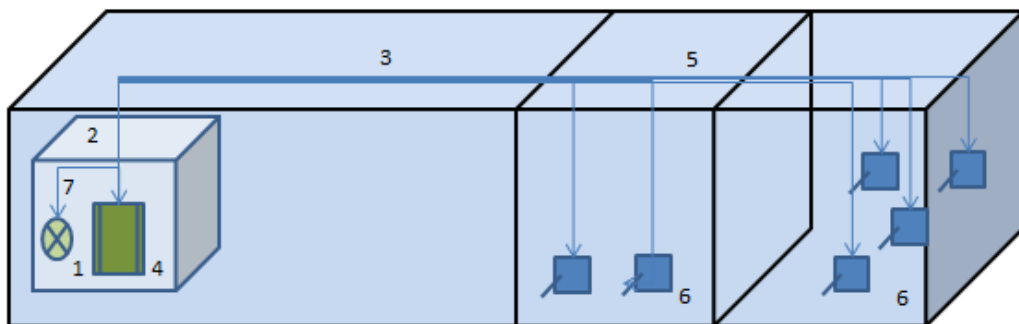


Figura 1. Representación de los subsistemas de la norma EIA/TIA-568

- **Enrutamiento**

Para el enrutamiento, tenemos la norma ANSI/TIA/EIA-569-A, la cual define las reglas para los siguientes subsistemas establecidos en la norma ANSI/TIA/EIA-568-B.

Cuarto de entrada de servicios

- Generalmente ubicado en el sótano o primer piso.
- Debe ser un área seca, que evite inundaciones.
- Debe estar lo más cerca de la ruta por donde ingresan los cables de acometida al piso o edificio.
- Debe tener únicamente equipos relacionados con el objetivo del cuarto.

Cuarto de telecomunicaciones

- Debe existir uno por cada piso.
- Debe tener medidas para control de temperatura.
- Si es posible, estos cuartos pueden estar alineados verticalmente a lo largo del edificio.
- Dos paredes deben ser de 20mm y deben tener 2.4 metros de alto.
- Debe tener precauciones contra sismos.
- Puede el cuarto de telecomunicaciones y de entrada de servicios ocupar el mismo espacio junto con sus elementos.

Rutas de cableado horizontal

- La ruta a recorrer debe encontrarse entre el techo del piso y el cielo falso.
- El cableado no debe estar apoyado sobre el cielo falso.
- Si el cable no se encuentra tendido sobre alguna estructura de apoyo (bandeja), este debe ser sujetado al techo con ganchos "J", los mismos que se deben colocar cada 1.52 metros.
- De usar bandejas o conductos, el material utilizado puede ser metal o plástico.

1.4.2 Diseño asistido por computador

El diseño asistido por computador, CAD por sus siglas en inglés. Es el conjunto de aplicaciones de software y hardware que automatizan tareas relacionadas con la creación, modificación y análisis de imágenes y diseños [10], [11].

Requisitos de implementación

Con el objetivo de asegurar un desarrollo eficiente de un aplicativo utilizado para actividades propias de CAD, se debe considerar los siguientes puntos[10]:

- Definir los procesos de búsqueda, procesamiento y producción de información. Es decir, en cada fase se debe identificar a todos los elementos que están relacionados e influyen sobre nuestro aplicativo CAD.
- Perfeccionar los métodos de diseño, basados en métodos matemáticos y computarizados.
- Elaborar interfaces intuitivas y debidamente documentadas que faciliten las tareas del usuario final.
- Unificar y estandarizar los métodos de diseños.
- Interactuar con otros sistemas CAD formalmente conocidos y aceptados.

Un aplicativo CAD se compone de rutinas y procedimientos de cálculos específicos y especializados, en otras palabras tienen un propósito bien definido, todos estos elementos deben ser simples, flexibles y confiables. Adicionalmente deben producir suficientes datos de calidad para la actividad que se ejecuta luego de cada interacción [10], [11].

1.4.3 Diseño arquitectónico asistido por computadora

El diseño arquitectónico asistido por computador, CAAD por sus siglas en inglés. Es el conjunto de técnicas que mediante el uso de herramientas de computación gráfica, dan soporte a tareas propias de proyectos arquitectónicos [12]. Un aplicativo CAAD normalmente posee un amplio conjunto de objetos y herramientas relacionados con la construcción.

Representación de planos

CAAD permite representar de dos formas un plano arquitectónico:

- Dos dimensiones 2D (Estructura superficial), y
- Tres dimensiones 3D (Estructura profunda).

Las dos formas mencionadas, realizan el mapeo uno a uno de los elementos que conforman el plano; por ejemplo. El diagrama de una puerta, escalera, etc. [12].

1.4.4 Inteligencia Artificial

La IA es la ciencia que se encarga del estudio, diseño y construcción de agentes inteligentes. Estos agentes inteligentes deben lograr realizar tareas que normalmente un ser humano las hace de mejor manera [13]-[14]. Muchos científicos durante las últimas décadas han tratado de crear y dar forma a sistemas computacionales con capacidades de aprendizaje y

razonamiento, similares a las de un ser humano. La inteligencia artificial (IA) busca que dichos sistemas imiten el comportamiento de nuestro cerebro [13]. La IA es una ciencia de reciente aparición, su desarrollo se da a partir de la Segunda Guerra Mundial. Su aplicación incluye a varias áreas como por ejemplo: robótica, sistemas expertos, diagnóstico de enfermedades [13].

Agentes

Un agente es cualquier entidad que tiene la capacidad de percibir su entorno, reaccionar internamente ante estos estímulos, e interviene sobre el medio a través de actuadores [13]. Los agentes contienen lógica funcional que interviene sobre sus procesos decisivos y al basarse en objetivos específicos, buscan alcanzar una meta. Su eficacia puede ser mejorada mediante mecanismos de aprendizaje.

Un agente decide y establece una secuencia de acciones que pueden llevarle a un estado deseable u objetivo. Este proceso es conocido como búsqueda y formalmente se lo denomina algoritmo de búsqueda, su parámetro de entrada es el problema a solucionar, y la secuencia de acciones se constituye en la solución [13].

Algoritmos de búsqueda

Los algoritmos de búsqueda están diseñados para encontrar un elemento dadas ciertas características dentro de una estructura de datos determinada. Se dividen en dos grandes grupos de acuerdo con la información con la que cuentan para cumplir su cometido, a saber: búsqueda informada y no informada.

Búsqueda no informada

En general, este tipo de algoritmos son los que presentan un nivel alto de ineficiencia con respecto al tiempo de ejecución y consumo de recursos computacionales [13]. Sin embargo, mencionamos las estrategias más utilizadas.

- **Búsqueda en anchura:** se realiza un recorrido por todos los nodos del primer nivel de izquierda a derecha, luego se avanza a los siguientes niveles con la misma estrategia hasta encontrar el nodo objetivo, normalmente se implementa una cola FIFO para dar con la solución [13], como se muestra en la Figura 2.

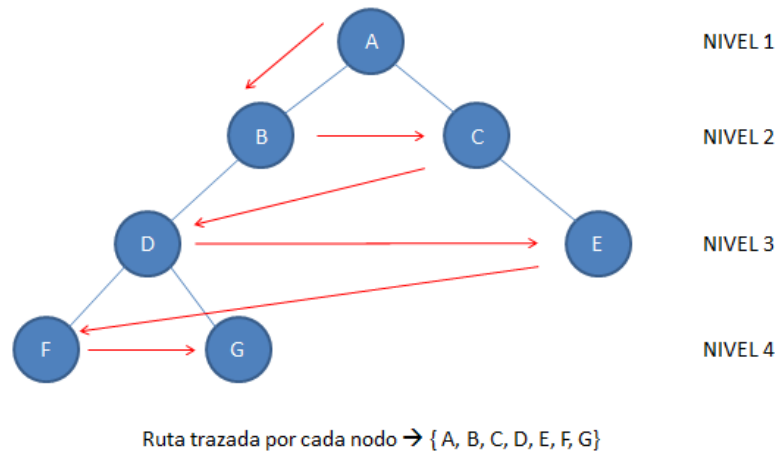


Figura 2. Representación gráfica de búsqueda por anchura.

- **Búsqueda en profundidad:** este método realiza un barrido de forma vertical por sobre los nodos que puedan representar el objetivo o la solución. Cuando no puede profundizar, el algoritmo retornará al anterior nodo que no tenga sucesores inexplorados [13], como se observa en la Figura 3.

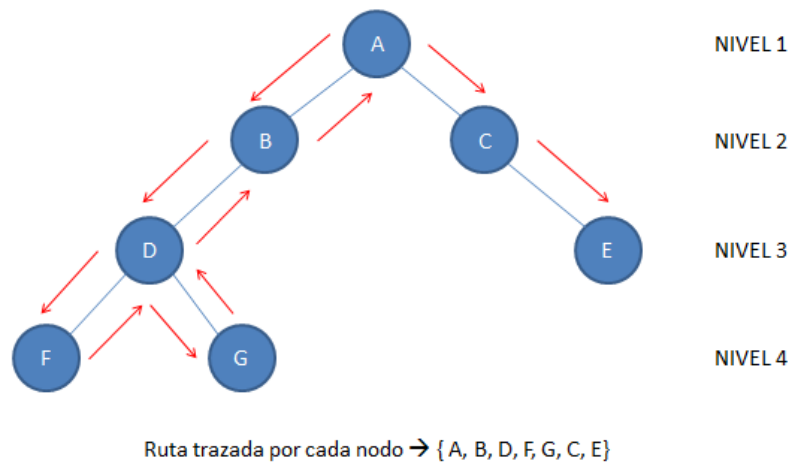


Figura 3. Representación gráfica de búsqueda en profundidad.

- **Búsqueda de costo uniforme:** la búsqueda inicia por el nodo raíz y avanza al siguiente nodo que presenta menor costo. Por lo tanto, es importante conocer el valor de cada iteración, pero si estos valores son los mismos para todos los nodos, estaremos hablando de una búsqueda en anchura, como se visualiza en la Figura 4.

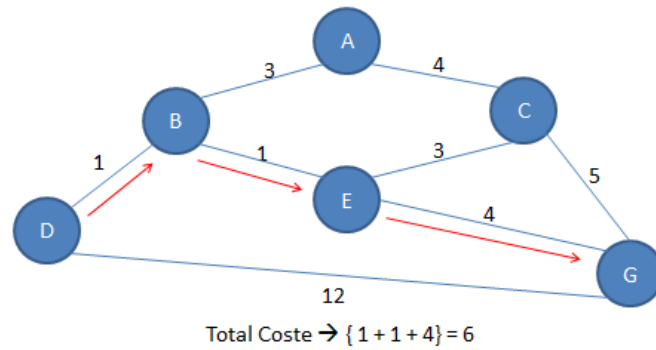


Figura 4. Representación gráfica de búsqueda de costo uniforme

- **Búsqueda bidireccional:** de forma paralela inicia la búsqueda desde el nodo inicial hacia el nodo objetivo y viceversa, como se observa en la Figura 5.

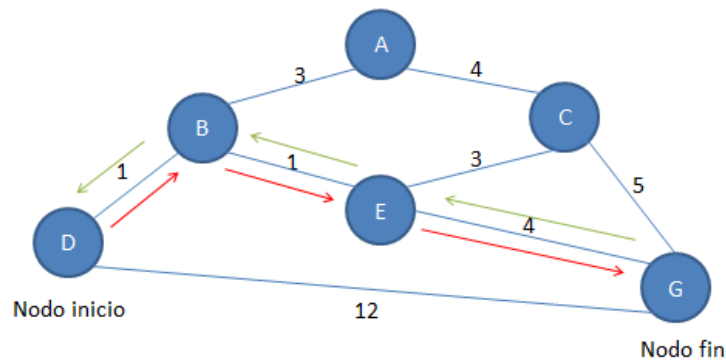


Figura 5. Representación gráfica de búsqueda bidireccional

Búsqueda informada y exploración

De modo general, la búsqueda informada utiliza el conocimiento específico del problema para encontrar soluciones de una manera más eficiente en términos de recursos utilizados.

- **Funciones Heurísticas:** Una función heurística se encarga de calcular el coste estimado más barato desde el nodo n hasta el nodo objetivo. También permite que nuestro algoritmo de búsqueda haga uso de información adicional relacionada con el problema [13].
- **Búsqueda voraz primero el mejor:** este algoritmo realiza una expansión del nodo más cercano al nodo objetivo, la evaluación a cada nodo se realiza utilizando la función heurística $f(n) = h(n)$, como se visualiza en la Figura 6.

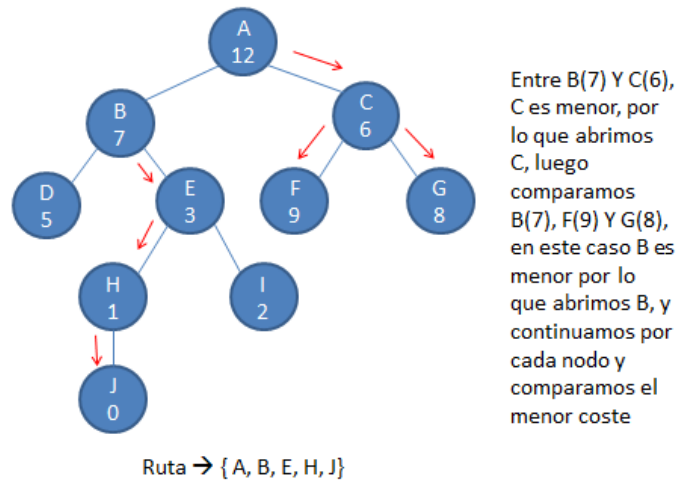


Figura 6. Representación gráfica de búsqueda voraz primero el mejor.

- Búsqueda A*:** minimizar el costo estimado total de la solución: conocida como búsqueda A-estrella o A-asterisco, este algoritmo utiliza dos componentes dentro de la ecuación. El primero $g(n)$, que nos entrega el coste del camino para ir desde el nodo n al nodo objetivo, y $h(n)$, el coste estimado del camino más barato desde el nodo n hacia el nodo objetivo. Con base en lo anteriormente dicho, entonces nuestro $f(n)$ será el coste más económico de la solución buscada [13] como se observa en la Figura 7. Este tema es ampliado en el punto 1.5.5.

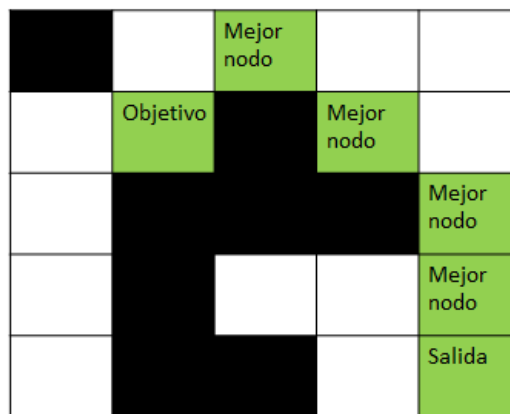


Figura 7. Representación gráfica de búsqueda A*, ruta a seguir para llegar desde *nodo salida* a *nodo objetivo*

Algoritmos de búsqueda local

Los algoritmos de búsqueda local son utilizados cuando el camino al objetivo es irrelevante. Por ejemplo, el problema de las 8-reinas, en el cuál no es importante el orden en el que se disponen al final las fichas, sino que se cumpla con la definición del mismo [13]. Este tipo de algoritmos gozan de dos ventajas significativas: usan muy poca memoria, y por lo general encuentran soluciones aceptables en espacios de estados grandes o continuos.

- **Búsqueda de ascensión de colinas:** el comportamiento de este algoritmo se basa en ejecutar un bucle que constantemente se mueve en dirección creciente, finaliza al alcanzar el pico en donde ningún vecino tiene un valor más alto [13].
- **Búsqueda de temple simulado:** este algoritmo tiene como objetivo evitar mínimos locales, ya que por ejemplo con la búsqueda de ascensión de colinas siempre tomamos el siguiente nodo y podríamos estancarnos en una solución que no es la óptima, en este caso el temple simulado escogerá nodos al azar para continuar.
- **Búsqueda por haz local:** mediante la ejecución en paralelo del algoritmo, se guardan k estados que son utilizados por los demás hilos para encontrar la mejor solución.
- **Algoritmos Genéticos:** es una variante de la búsqueda de haz estocástica, debido a que los estados generados se forman mediante la combinación, cruce, mutación de dos estados padres como se observa en la Figura 8. Este tema es ampliado en el punto 1.5.6.

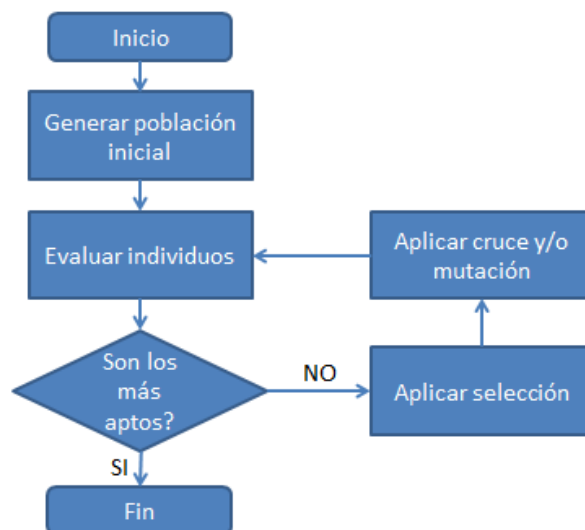


Figura 8. Diagrama de flujo de un algoritmo genético

1.4.5 Búsqueda A*

Pertenece a la familia de algoritmos de búsqueda informada. El algoritmo A* aparece en 1964, cuando su inventor Nils Nilsson buscaba mejorar el rendimiento del algoritmo de Dijkstra para el proyecto Shakey [15], proponiendo que el algoritmo mediante el uso de heurísticas encuentre el camino óptimo entre dos nodos mejorando el rendimiento con respecto al tiempo de recorrido y menor distancia visitada [16].

La ejecución de este algoritmo se basa en seguir una ruta de menor costo calculado. Adicionalmente mantiene una lista de rutas alternativas, esto debido a que si por ejemplo durante la corrida se encuentra un coste mayor, se retomará la ruta de menor costo; también puede encontrar que la ruta no tiene salida, y realizará el proceso anteriormente mencionado. Este proceso se lleva a cabo hasta que se alcanza el nodo objetivo [13], [16].

En cada nodo de la ruta definimos los siguientes componentes:

- $g(n)$ = la función de costo para llegar del nodo inicial al nodo actual.
- $h(n)$ = la función heurística admisible, que se calcula desde el nodo actual hasta el nodo final

Por lo tanto, tenemos

$F(n) = g(n) + h(n)$, que será el costo estimado del nodo por el cual el algoritmo está atravesando.

El siguiente ejemplo nos ayudará a entender la estrategia de este algoritmo. La finalidad es llegar desde la celda denominada nodo inicio hasta el nodo objetivo. Las celdas en color blanco permiten avanzar, y las celdas en color negro son obstáculos como se visualiza en la Figura 9.

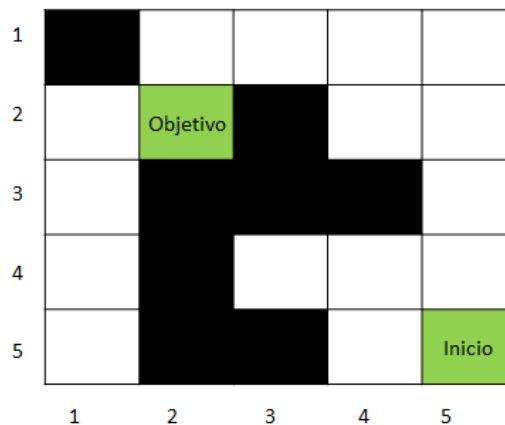


Figura 9. Ejemplo de búsqueda A*

La función heurística que utilizaremos será:

$$h(n) = \sqrt{(x(n) - x(nf))^2 + (y(n) - y(nf))^2}, \text{ también conocida como distancia Euclidiana.}$$

Desde la celda inicio, podemos avanzar hacia las celdas 1, 2 y 3 como se indica en la Figura 10; y aplicar la fórmula de cálculo de la siguiente manera:

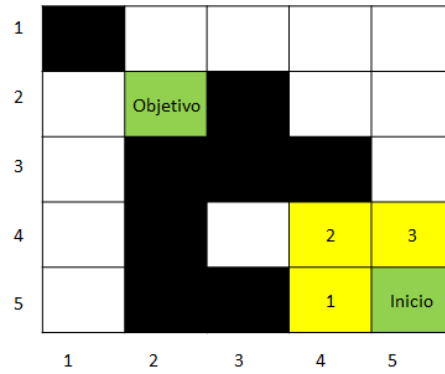


Figura 10. Paso 1, búsqueda A*

$$f(1) = g(1) + \sqrt{(4 - 2)^2 + (5 - 2)^2} = 4.6$$

$$f(2) = g(2) + \sqrt{(4 - 2)^2 + (4 - 2)^2} = 3.82$$

$$f(3) = g(3) + \sqrt{(4 - 2)^2 + (5 - 2)^2} = 4.6$$

Seleccionaremos la que presente menor f , en este caso 3.82 que se encuentra en la celda 2. Desde esta celda podemos hacia las celdas 4 y 5. Como se muestra en la Figura 11.

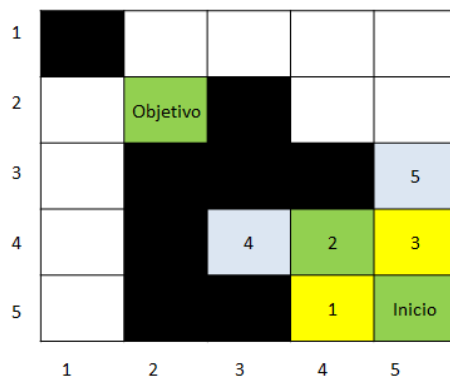


Figura 11. Paso 2, búsqueda A*

Nuevamente aplicaremos la fórmula de cálculo de la siguiente manera

$$f(4) = g(4) + \sqrt{(4 - 2)^2 + (3 - 2)^2} = 3.23$$

$$f(5) = g(5) + \sqrt{(3 - 2)^2 + (5 - 2)^2} = 4.16$$

El menor f es de la celda 4, sin embargo está celda no tiene nodos adyacentes por los cuales avanzar, por lo que nuestro camino será tomar la celda 3 y luego seguir por la celda 5. Las siguientes iteraciones del algoritmo generan el camino a continuación, tal como se muestra en la Figura 12.

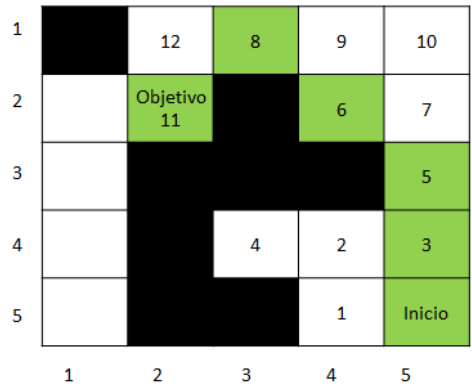


Figura 12. Paso 3, búsqueda A*

Mientras que los nodos generados, y los nodos de la ruta solución en color verde se pueden observar en la Figura 13.

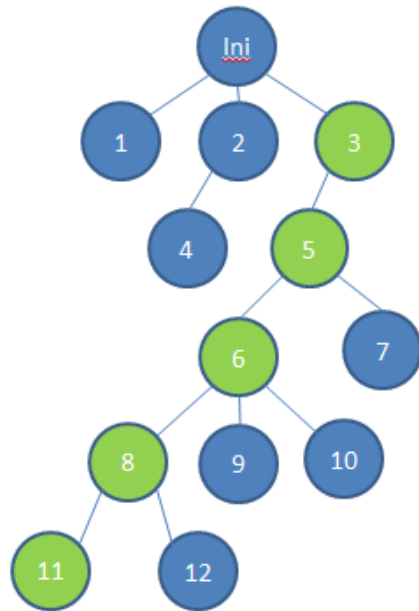


Figura 13. Nodos de la ruta solución, búsqueda A*

1.5.6 Algoritmos genéticos

Los algoritmos genéticos (AG) son una familia de algoritmos de búsqueda inspirados en los principios de la evolución de la naturaleza. Su formulación se basa en el principio fundamental de la supervivencia del más fuerte enunciada por Charles Darwin en 1859; el cual establece que los individuos de una población se someten a una reproducción continua y la competencia tiende a preservar a los mejores [6]. El trabajo, sobre lo que hoy se llama computación evolutiva, comenzó en los años sesenta del siglo XX en Estados Unidos y Alemania [7].

Al imitar el proceso de selección y reproducción natural, los AG pueden producir soluciones de alta calidad para diversos problemas relacionados con la búsqueda, la optimización, y el aprendizaje. Al mismo tiempo, su analogía con la evolución natural permite superar algunos de los obstáculos que encuentran las búsquedas tradicionales y algoritmos de optimización, especialmente para problemas con una gran cantidad de parámetros y representaciones matemáticas complejas [15].

Codificación

- **Gen:** Parámetro básico de una posible solución.
- **Cromosoma:** Conjunto de genes, también conocido como cadena de valores.
- **Individuo:** Posible solución de un problema.
- **Población:** Conjunto de individuos. Tal como se muestra en Figura 14.

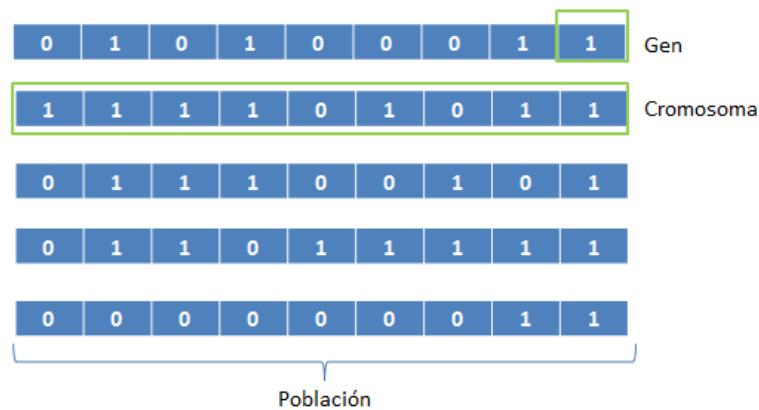


Figura 14: Codificación de elementos de un algoritmo genético

Funcionamiento

- **Función de evaluación:** también conocida como función de adaptación, es utilizada para determinar la idoneidad de un cromosoma o individuo asignándole un valor real. De esta forma se puede dependiendo del objetivo del problema tomar los valores máximos o mínimos y propagarlos a través de las generaciones siguientes.
- **Inicialización:** generamos una población inicial de forma totalmente aleatoria, que se compone de un grupo n de cromosomas que representaran a las posibles soluciones al problema. Es importante que esta población tenga características totalmente heterogéneas para así evitar que nuestro algoritmo caiga en un agujero del que no pueda salir.
- **Evaluación:** como habíamos mencionado, se debe aplicar a cada cromosoma la función de evaluación.

- **Condición de finalización:** habitualmente se aplican dos condiciones para dar término a la corrida de un AG. Por un lado, se puede ejecutar un número máximo de iteraciones, y detener cuando en la población ya no existan cambios. Sin embargo, también podemos parar si el objetivo de nuestro AG ha sido cumplido, es decir hemos hallado una solución que satisface a nuestro problema.

Operaciones

Durante la ejecución del AG, podemos aplicar operaciones que permitan que los próximos individuos a ser evaluados sufran cambios que imitando los procesos evolutivos, nos ayuden a encontrar la solución[17], [18].

- **Selección:** consiste en seleccionar a los mejores individuos que después de aplicar la función de adaptación, demostraron ser más idóneos, tal como se muestra en la Figura 15.

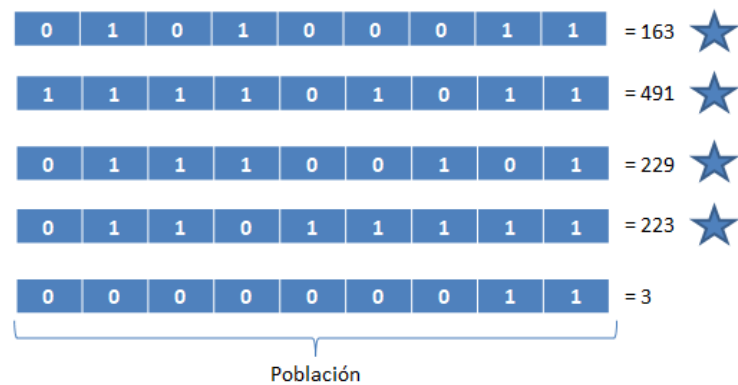


Figura 15: Selección de individuos de una población

- **Combinación o Cruce:** se basa en utilizar dos cromosomas y combinar sus genes, tal como se observa en la Figura 16.

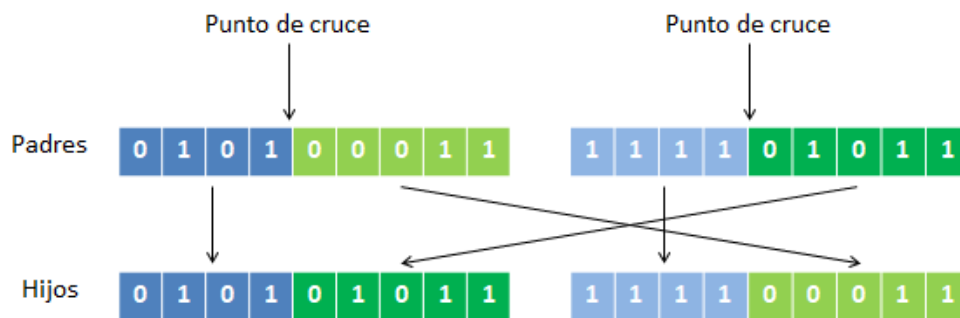


Figura 16: Mecanismo de cruce

- **Mutación:** Altera de forma aleatoria los genes de los individuos. Es de vital uso para impedir que se presente una convergencia prematura en nuestra población, tal como se indica en la Figura 17.

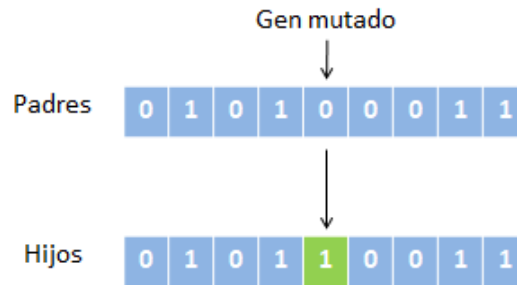


Figura 17: Mecanismo de mutación

1.4.7 Aprendizaje por refuerzo

Esta rama del aprendizaje automático, explora las acciones que un agente de software puede tomar sobre un entorno dado, y al ser ejecutadas, le permite descubrir cuáles de ellas son las que originan un sentido de recompensa o estímulo [13], [19]. En casos más complejos, el agente de software es capaz de determinar la posibilidad de maximizar las recompensas posteriores en un ambiente interactivo e iterativo.

Por ejemplo, tenemos un robot que decide si ingresar a un cuarto a recolectar la basura del tacho, o regresar a sus estación de carga, la decisión es tomada con base en el nivel de batería actual (estado) y que tan rápido y fácil le ha sido encontrar en otras situaciones similares la estación de carga.

Los principales elementos que intervienen en este tipo de aprendizaje son[19]:

- **Política:** define como debe comportarse el agente en un determinado momento de la ejecución del algoritmo, es decir la regla o conjunto de reglas a las cuales se atiene. Pueden ser una función simple, una tabla de búsqueda o un cálculo más extenso.
- **Recompensa:** es el objetivo del problema de aprendizaje por refuerzo, el entorno genera esta señal y el agente debe interpretarla y maximizarla con cada iteración.
- **Función de valor:** establece la cantidad total de recompensas que el agente puede acumular.
- **Modelo del entorno:** imita el comportamiento del entorno y permite inferir la manera en que actuará el entorno ante cada acción.

Para facilitar estos conceptos, pensemos en el popular juego Pacman, mediante la ejecución de un bucle: estado \rightarrow acción \rightarrow recompensa, tiene como objetivo acumular la mayor cantidad de puntos. Así tenemos:

- El Agente, en este caso el elemento Pacman.
- Estado, la ubicación dentro del tablero.
- Modelo o ambiente, el laberinto por el cual recorre el elemento Pacman.
- Recompensa, ganará puntos cuando pueda comer las galletas presentes dentro del laberinto, y será negativa cuando un fantasma lo atrape.
- Acción, el movimiento que realizará el elemento Pacman hacia una nueva posición, tal como se observa en la Figura 18

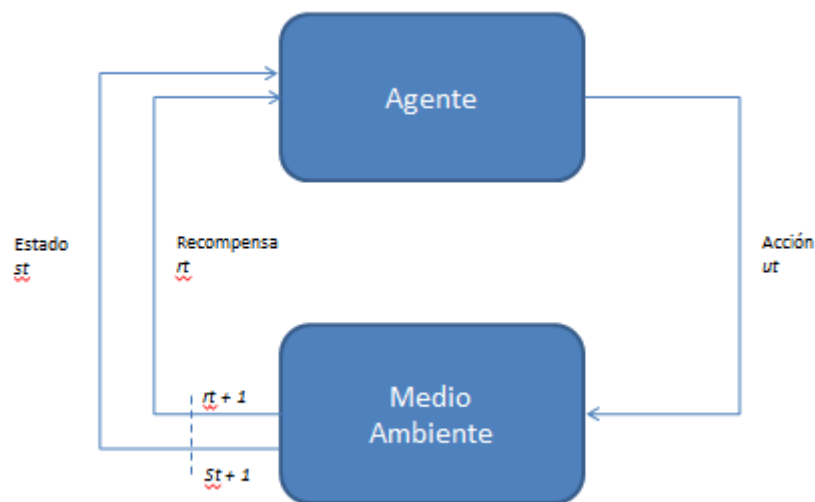


Figura 18: Problema básico de aprendizaje por refuerzo.

1.4.8 Relación entre Algoritmos genéticos y el aprendizaje por refuerzo

El desarrollo de agentes de software que realicen tareas complejas dentro de entornos o ambientes complejos y variables es un objetivo clave para la inteligencia artificial. El marco de trabajo que se ha venido utilizando, involucra a algoritmos de aprendizaje por refuerzo que se basan en los procesos de decisión de Markov y la implementación de funciones de valor [20].

Sin embargo, existen otro tipo de alternativas para resolver problemas de aprendizaje por refuerzo, por ejemplo tenemos el uso de optimización de tipo caja negra (black box). Un conjunto específico de este tipo de algoritmos, son las estrategias de evolución y específicamente en este caso tenemos a los algoritmos genéticos[20].

Recientes estudios sobre el tema, demuestran que el uso en conjunto de estas estrategias, logró que las tareas del modelo humanoide de MuJoCo sean resueltas en menos de 10 minutos. MuJoCo es el sitio web que permite realizar pruebas de algoritmos que aprenden a mover un robot [20].

2 METODOLOGIA

2.1 Ciencia del diseño

Para la realización del presente trabajo, se aplicará la Ciencia del Diseño como metodología de investigación dado que a través de conocimientos específicos queremos resolver un problema concreto de manera articulada desarrollando un artefacto. Esta metodología nos orienta en la consecución de varios objetivos: crear, aplicar, evaluar y mejorar los artefactos que se desarrollan dentro de las tecnologías de la información. Dichos elementos cuando ya se encuentran implementados favorecen a las organizaciones en tareas como: apoyo de operaciones, gestión, análisis y toma de decisiones. En la Figura 19, observamos un marco de trabajo propuesto para la ciencia del diseño [21].

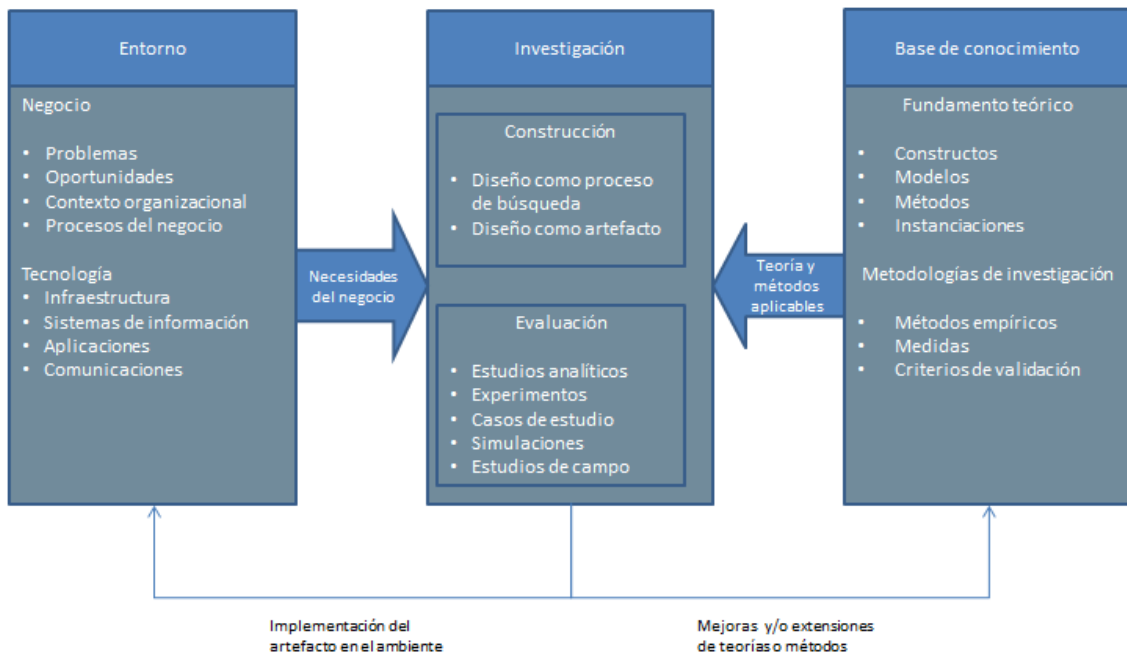


Figura 19: Marco de trabajo de la ciencia del diseño.

2.1.1 Entorno

Este pilar se encarga de definir el espacio del problema sobre el cual se encuentra el objeto de investigación [21]. Se compone por dos grandes familias: negocios y tecnología. Aquí hallamos los problemas, las oportunidades, entendemos la situación de la organización y las circunstancias sobre las que desarrolla sus procesos comerciales. Mantiene un inventario de activos de tecnología que pueden ser reemplazados o integrados con un nuevo artefacto.

Cuando implementamos el artefacto sobre este entorno encontramos la razón principal para usar ciencia del diseño sobre nuestra investigación [21]–[23].

2.1.2 Base de conocimientos

Facilita los materiales mediante los cuales los artefactos serán construidos y evaluados. Está compuesto por Fundamento teórico y metodologías de investigación. Los trabajos realizados en esta fase se constituyen en la base para el desarrollo de nuevos artefactos, luego estos mismos resultados obtenidos son evaluados a medida que van mejorando y contribuyen para incrementar la base de conocimientos de proyectos e investigaciones a futuro [21]–[23].

2.1.3 Investigación

La investigación en ciencia del diseño aborda problemas importantes de forma única o innovadora, y aporta de forma comprobada a desarrollar la base de conocimientos. Los artefactos que vamos desarrollando son experimentos que proporcionan respuestas positivas o negativas a las técnicas que se han implementado dentro del proyecto, enmarcados dentro de un ambiente de búsqueda de prueba y error. Finalmente, el resultado perseguido de todo este proceso consiste en obtener teorías y métodos nuevos o mejorados que podrían formalizarse en mejores prácticas [21]–[23].

2.2 Uso del framework de Ciencias del diseño

El diseño científico nos proporciona un plan articulado que establece prototipos para conseguir resultados. El aplicar este modelo nos permite solucionar problemas concretos que se presentan en un entorno en el cual el ser humano desarrolla sus actividades [24]. Por otro lado, conocemos que las Tecnologías de la Información, es una disciplina que se relaciona y aplica teorías de otras áreas del conocimiento como son economía, ciencias de la computación y ciencias sociales entre otras con el objetivo de solucionar problemas vinculados con las organizaciones, por lo que muchos investigadores han hecho uso de varios modelos conceptuales que les permita enlazar dentro de una plantilla las fases y tareas propias de la construcción del artefacto [22]. Por consiguiente, basados en la teoría que dicta esta metodología, necesitamos asegurar que nuestro producto final será el “artefacto” identificado como el algoritmo genético capaz de encontrar soluciones óptimas al dilema descrito en el planteamiento del problema. Lo anteriormente comentado, se resume en Tabla 1, aquí

podemos observar la relación existente entre los objetivos específicos del proyecto de investigación, la etapa asociada a la metodología, y la actividad en concreto a realizar.

| Objetivo | Etapa | Actividad |
|---|---|---|
| Modelar la problemática para obtener la función de fitness | Identificación del problema y motivación Objetivos de la solución Diseño y desarrollo | Definir el alcance de la solución propuesta. Obtener documentación relacionada con nuestra solución en anteriores estudios. Definir los estándares de cableado estructurado que se usarán como parámetros. Representar de forma matemática el plano arquitectónico de una red de cableado y sus componentes. |
| Implementar algoritmos | Diseño y desarrollo | Implementación del algoritmo A estrella. Definir la población inicial para AG. Ejecutar la función de aptitud AG. Seleccionar las soluciones más aptas. Implementar mecanismo de cruzamiento. Implementar mecanismo de mutación. Definir condiciones para terminar la ejecución del AG. |
| Realizar pruebas y ajustar los parámetros del algoritmo | Demostración | Realizar pruebas con casos de estudio, simulaciones, datos de práctica. Afinar los parámetros de ejecución del algoritmo genético para elevar su rendimiento. Elaborar métricas y gráficas con las variables objetivo del algoritmo y comparar cada ejecución con distintos parámetros. |

Tabla 1. Framework utilizado, y relación entre objetivo-etapa-actividad.

2.3 Pruebas

Para el presente caso de estudio, se han utilizado dos modelos de pruebas ampliamente usados en el desarrollo de software;

- Pruebas funcionales, y
- Pruebas de desempeño.

2.3.1 Pruebas funcionales

Este tipo de pruebas asegura que las funcionalidades deseadas o requisitos funcionales se encuentren implementadas. Además validan que los requisitos de navegación, ingreso de datos, procesamiento y la generación de resultados se encuentren correctamente desarrolladas. Estas pruebas se elaboran conforme a los casos de uso del sistema [25].

Al verificar el sistema mediante la técnica de caja negra, es decir dejando a un lado la implementación interna del software. El tester interactúa con la aplicación mediante la interfaz proporcionada y únicamente analiza los inputs y outputs del aplicativo. Cada uno de los casos de uso es ejecutado utilizando datos válidos e inválidos [25]. Con esto se valida:

- Resultados esperados, con entrada de datos correctos.
- Despliegue de mensajes o advertencias relacionadas con entrada de datos incorrectos.
- Aplicación correcta de reglas de negocio.

2.3.2 Pruebas de desempeño

El objetivo principal de este tipo de pruebas es asegurar que el tiempo de respuesta al ejecutar procesos sea óptimo cuando los mismos son sensibles al factor tiempo. Estas pruebas se ejecutan en reiteradas ocasiones con diferentes cargas de datos. Los resultados entregados por la ejecución de cada prueba son aprovechados para encontrar cuellos de botella en los procesos computacionales o dispositivos electrónicos, y por consiguiente permiten afinar el código fuente del aplicativo o en su defecto delimitar el hardware sobre el que se debe correr el aplicativo [25].

3 RESULTADOS

En este Capítulo se presentan los resultados obtenidos al emplear los conceptos analizados en el Capítulo 2. También se describen los componentes principales del artefacto desarrollado, y la forma de utilizarlo. Adicionalmente se ejecutan pruebas de la aplicación contra casos didácticos con el objetivo que el estudiante forme una idea clara de uso y beneficio del mismo.

3.1 Diseño de la aplicación

Para desarrollar el aplicativo, se han utilizado técnicas de ingeniería de software ampliamente conocidas, de igual forma han sido aplicados estándares de programación y mejores prácticas.

3.1.1 Componentes de la aplicación

El artefacto desarrollado tiene los siguientes componentes:

- SlnBestCommsRoom: carpeta contenedora de la solución implementada sobre Microsoft visual Studio 2019.
- AppBestCommsRoom: Aplicación Windows forms que abarca a los formularios, librerías, algoritmos y clases utilizadas en el artefacto resultado.

Dentro de la librería AppBestCommsRoom, tenemos los siguientes elementos:

- Referencias
 - Librería log4Net, utilizada para registrar log de eventos del aplicativo.
 - Librería Newtonsoft.Json, usada para leer archivos en formato json.
- GraphicsLib
 - WindowGraphic, usada para graficar líneas dentro de la cuadrícula.
- Model
 - Clases, contiene todas las clases utilizadas por el artefacto.
 - PisoModel, modelo utilizado para mapear las propiedades de un objeto json, en este caso específicamente relacionado con los datos de un piso.

- Src
 - AlgoritmoGenetico, código del algoritmo genético utilizado por el aplicativo.
 - PathFinder, código del algoritmo de búsqueda A* utilizado por el aplicativo.
 - Programa, entrada principal del aplicativo.

3.1.2 Diagramas de casos de uso

- Caso de uso: Calcular mejores puntos de entrada por cuarto. Observar Figura 20.

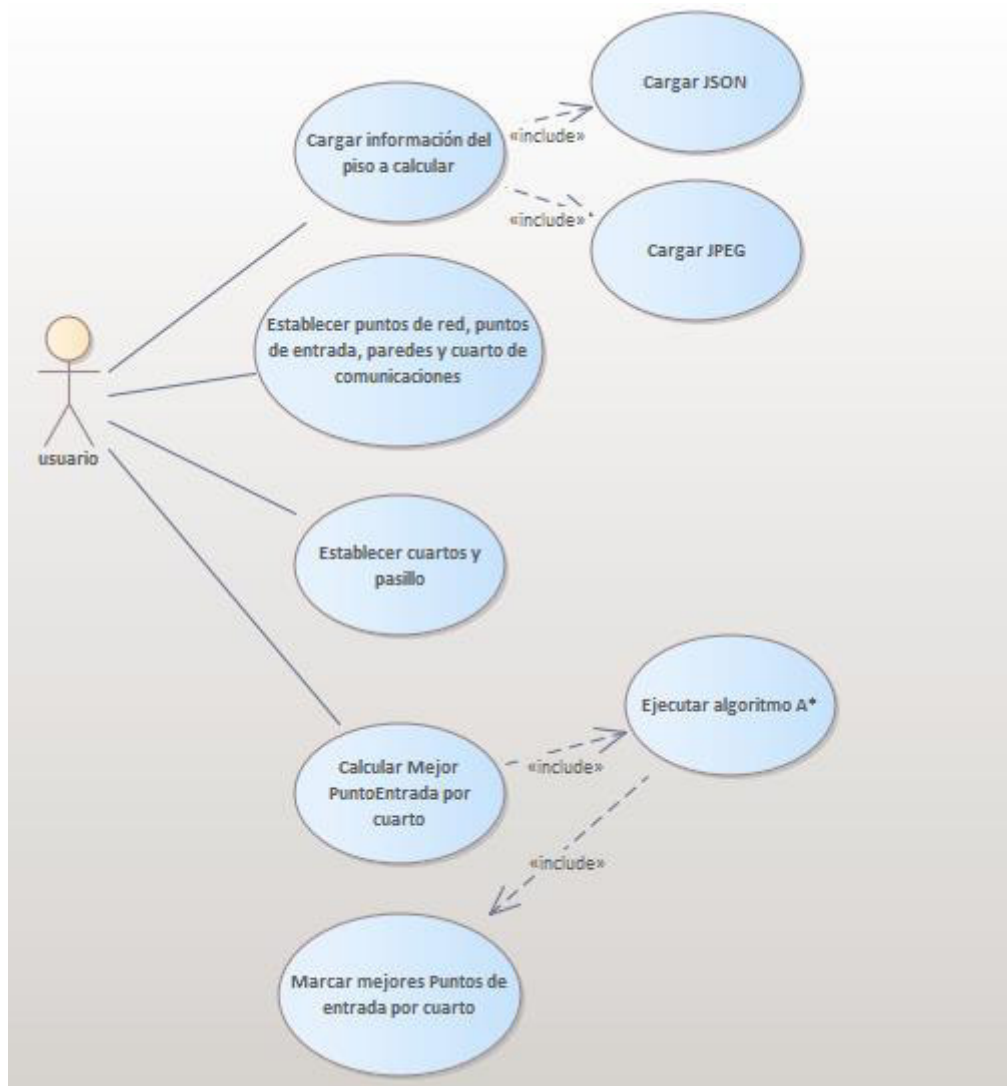


Figura 20. Calcular mejor punto de entrada por cuarto

- Caso de uso: Calcular mejor ubicación de cuarto de comunicaciones. Observar Figura 21.

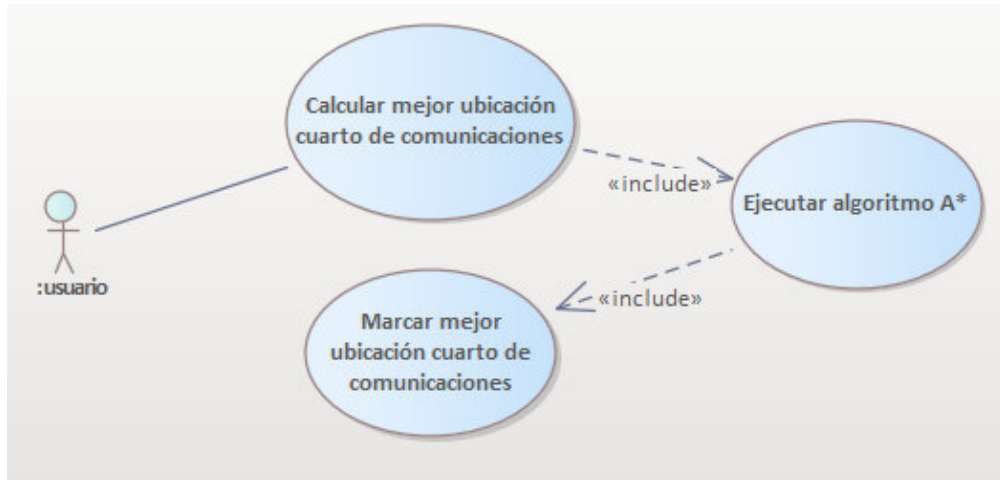


Figura 21. Calcular mejor ubicación cuarto de comunicaciones.

3.1.3 Diagrama de secuencia y colaboración

- Caso de uso: Calcular mejores puntos de entrada por cuarto. Observar Figura 22.

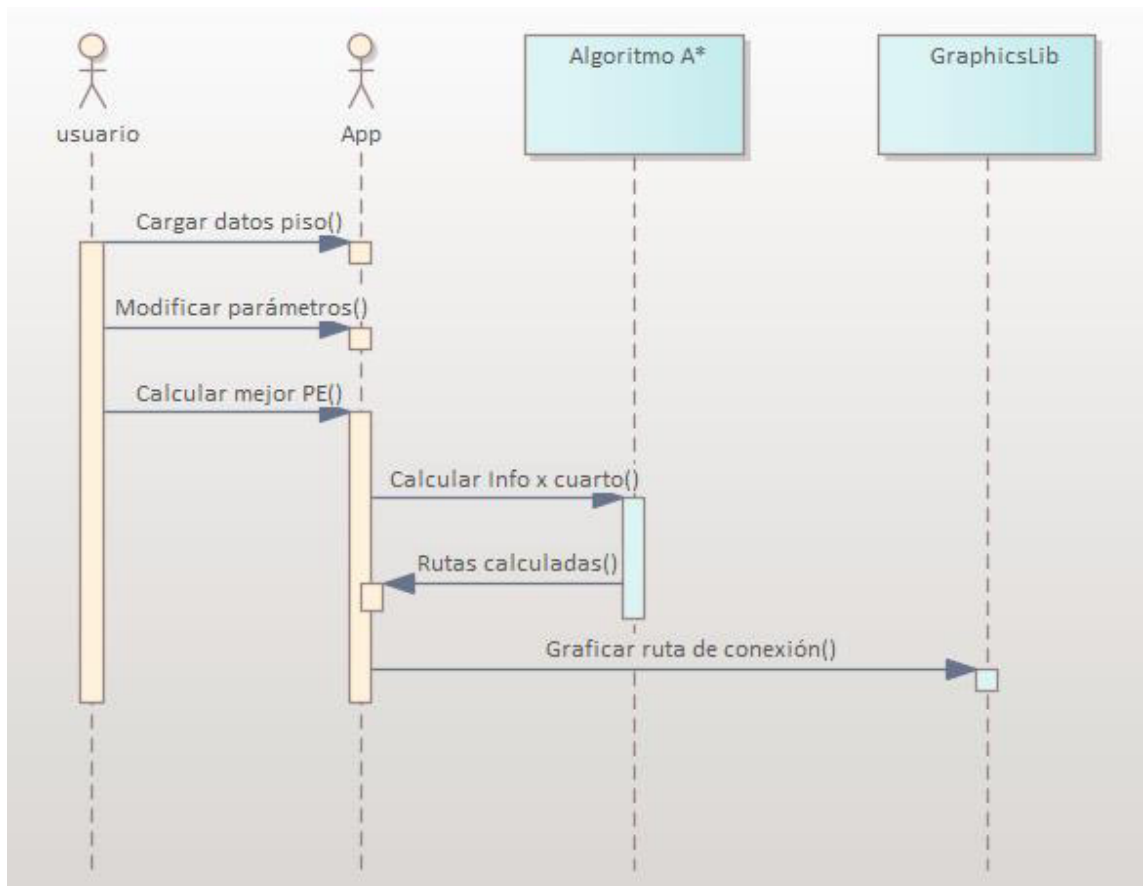


Figura 22. Calcular mejor punto de entrada por cuarto

- Caso de uso: Calcular mejor ubicación de cuarto de comunicaciones. Observar Figura 23.

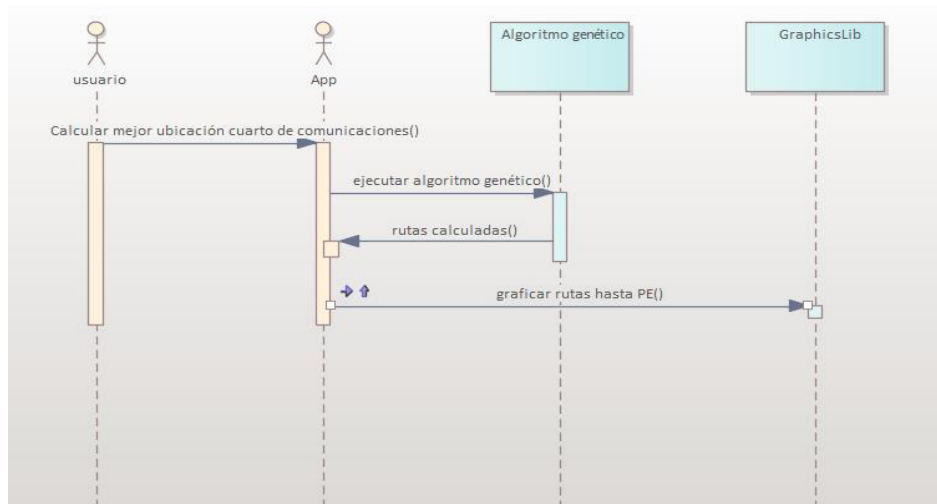


Figura 23. Calcular mejor ubicación de cuarto de comunicaciones

3.1.4 Diagrama de clases

El diagrama de clases utilizado en el aplicativo es el observado en la Figura 24.

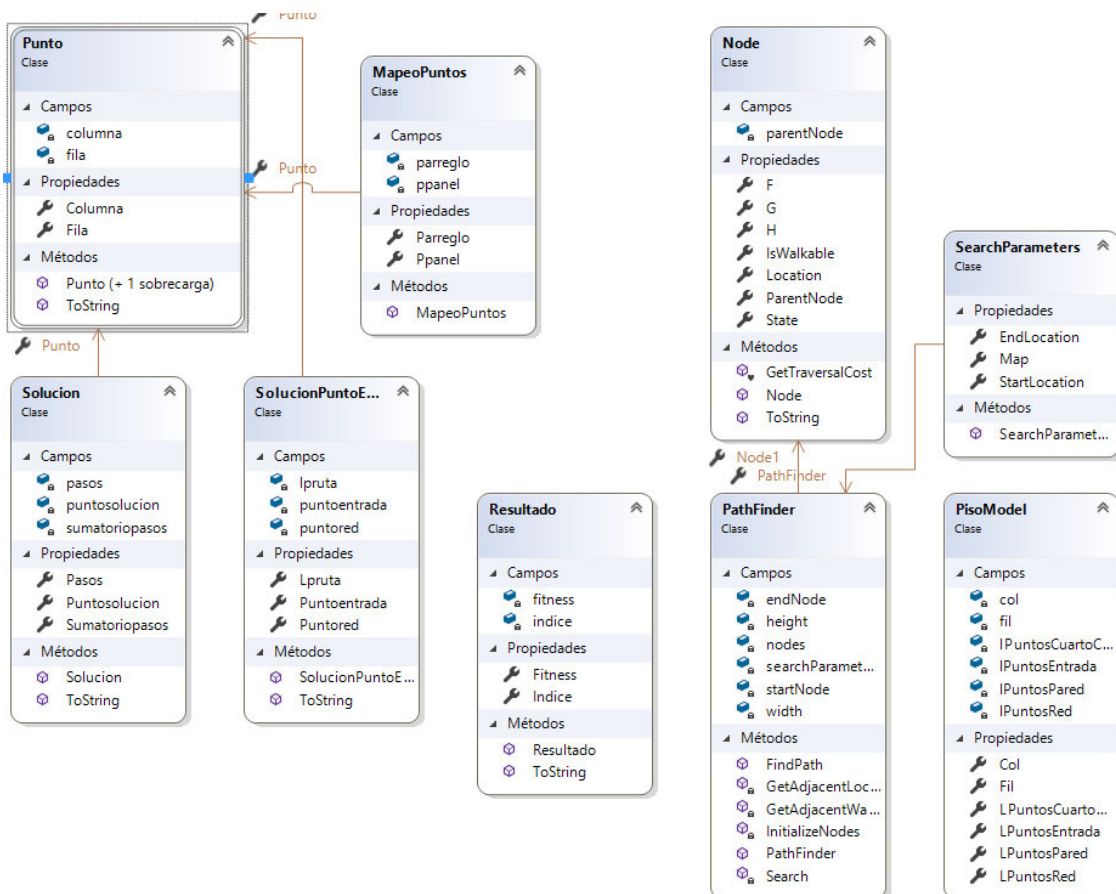


Figura 24: Diagrama UML del aplicativo AppBestCommsRoom.

El detalle de cada una de las clases utilizadas es:

- **Punto:**

Representa un punto de un plano cartesiano en forma de Fila, Columna.

Propiedades:

- Columna; Identifica a la columna del punto instanciado.
- Fila; Identifica a la fila del punto instanciado.

Métodos:

- ToString(); Permite sobre escribir el método genérico para visualizar el par ordenado instanciado.

- **MapeoPuntos:**

Representa la relación entre un punto de la matriz general del piso y su respectivo punto dentro de la matriz reducida correspondiente a un cuarto o sección.

Propiedades:

- Parreglo; punto dentro de la matriz reducida.
- Ppanel; punto de la matriz o panel del piso.

Métodos:

- MapeoPuntos(); Constructor.

- **Solucion:**

Clase utilizada para evaluar los resultados del algoritmo genético.

Propiedades:

- Pasos; detalle de celdas recorridas.
- Puntosolución; coordenadas del punto evaluado.
- Sumatoriopasos: sumatorio de pasos.

Métodos:

- Solucion(); Constructor.

- **SolucionPuntoEntrada:**

Clase utilizada para evaluar los resultados del algoritmo A*

Propiedades:

- Lpruta: Lista de puntos con ruta recorrida.
- Puntoentrada: Coordenadas del punto de entrada.
- Puntored: Coordenadas del punto de red.

Métodos:

- SolucionPuntoEntrada(); Constructor.

- **Resultado:**

Clase utilizada para resumir y ordenar los resultados de la ejecución del algoritmo genético.

Propiedades:

- Fitness: Valor calculado.
- Indice: Índice del elemento calculado.

- **Node:**

Representa a un nodo en una cuadrícula

Propiedades:

- ParentNode: Nodo padre.
- Location: Ubicación del nodo.
- IsWalkable

3.1.5 Interfaz de usuario

La interfaz gráfica de usuario diseñada para el aplicativo busca que la interacción con el usuario sea rápida e intuitiva. Mediante el uso de teclado y mouse se realizan las principales entradas y en pantalla se observan los resultados después de presionar cada uno de los botones que llaman a la ejecución de los algoritmos empleados, tal como se observa en la Figura 25.

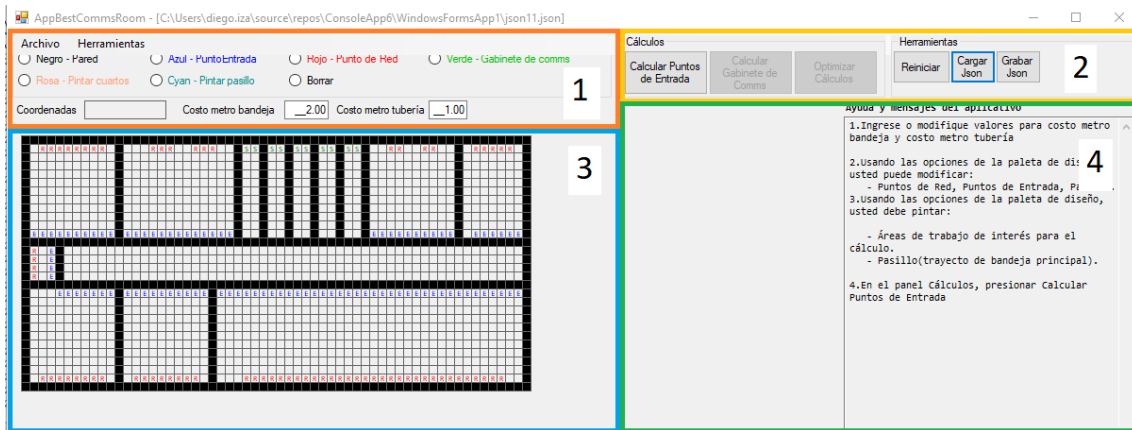


Figura 25. Pantalla principal del aplicativo

- Sección 1: Paleta de marcadores, utilizada para agregar o modificar los elementos del piso que se quiere calcular, tal como se indica en la Figura 26.

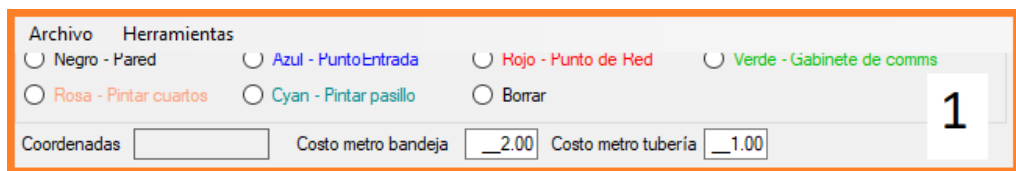


Figura 26. Elementos de la sección 1. Paleta de marcadores

Adicional, en esta área se encuentra la etiqueta que permite visualizar la coordenada (fila, columna) de la celda seleccionada por el puntero del mouse dentro del panel, y los cuadros de texto que permiten ingresar el costo por metro de bandeja y tubería.

- Sección 2: Botones de acción, área usada para interactuar con la ejecución de los procesos principales del aplicativo, adicional poseen acciones destinadas a cargar y almacenar objetos JSON que representan a un piso, tal como se observa en la Figura 27.

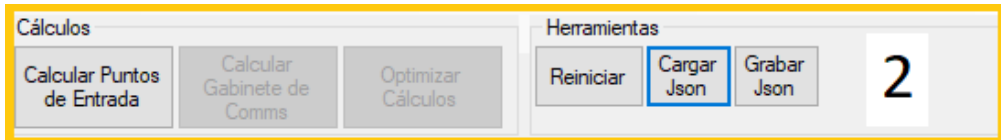


Figura 27. Elementos de la sección 2. Botones de acción

- Sección 3: Figura que representa al piso con el que se va a trabajar, tal como se indica en la Figura 28.



Figura 28. Elementos de la sección 3. Representación del piso

- Sección 4: Mensajes y ayuda del aplicativo, tal como se observa en la Figura 29.

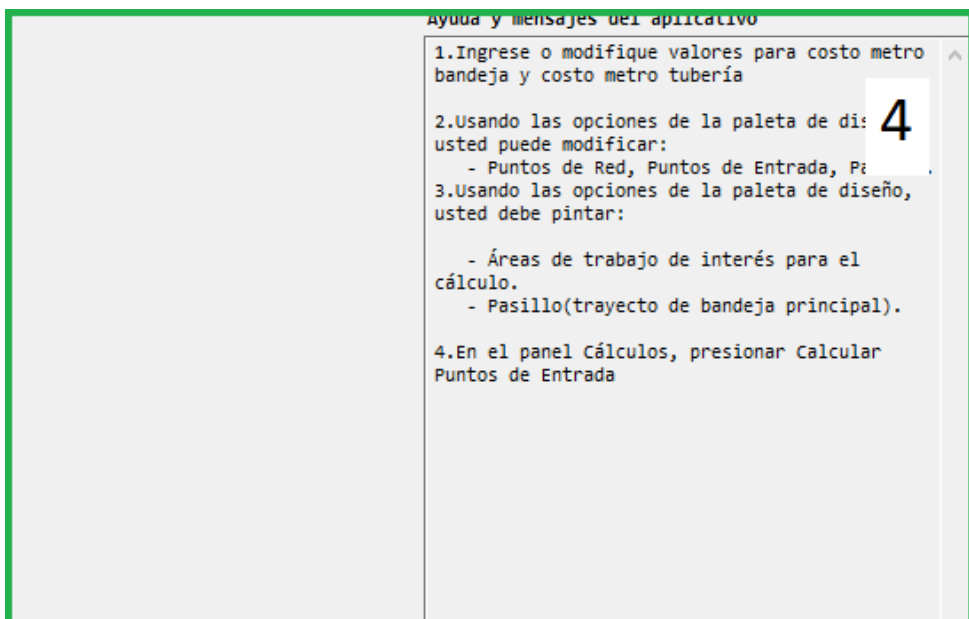


Figura 29. Elementos de la sección 4. Mensajes del aplicativo.

3.2 Algoritmos utilizados por la aplicación

3.2.1 Pseudocódigo Algoritmo de búsqueda A*

El pseudocódigo del algoritmo de búsqueda A* se encuentra en el Anexo I.

3.2.2 Pseudocódigo Algoritmo genético

El pseudocódigo del algoritmo genético se encuentra en el Anexo II.

3.2.3 Pseudocódigo Algoritmo de carga al entorno virtual de aprendizaje

El pseudocódigo del algoritmo se encuentra en el Anexo III.

3.3 Funcionamiento de la aplicación

Para utilizar el aplicativo de manera correcta, seguiremos los siguientes pasos:

1. Desde el panel Herramientas (Figura 27), presionaremos el botón Cargar Json, y navegaremos hasta la ruta del archivo deseado. A continuación se dibujará la imagen del piso en el aplicativo (Figura 28).
2. Mediante el panel Paleta de marcadores (Figura 26), editaremos los puntos de red, puntos de entrada, gabinete de telecomunicaciones, paredes, etc. en el caso de ser necesario.
3. Mediante el panel Paleta de marcadores (Figura 26), definiremos los cuartos o áreas de trabajo del diagrama y el pasillo central. Estas secciones deben ser marcadas de forma obligatoria para habilitar la ejecución de los cálculos.
4. Desde el panel Botones de acción (Figura 27), presionaremos el botón “Calcular puntos de entrada”, luego de realizados los cálculos, el aplicativo actualizará las celdas del piso.
5. Desde el panel Botones de acción (Figura 27), presionaremos el botón “Calcular gabinete de comms”, luego de realizados los cálculos, el aplicativo actualizará las celdas del piso, y presentará el resumen de ejecución en el panel resultados y mensajes (Figura 29).
6. De acuerdo con los resultados y el modo de ejecución, el aplicativo determina la posibilidad de mejorar los cálculos presentados. Si el botón “Optimizar cálculos” se

encuentra habilitado, el usuario puede presionar para realizar esta tarea. El resumen será presentado en el panel resultados y mensajes (Figura 29).

3.4 Visualización del resultado dentro de un EVA

El resultado de los cálculos generados por el artefacto, pueden ser llevados a un entorno virtual de aprendizaje 3D, en el cual la visualización de la distribución real de bandejas, tuberías, gabinete de telecomunicaciones y puntos de red, permitan que el alumno capte fácilmente la disposición física de dichos elementos.

Para conseguir este objetivo, el artefacto tiene la capacidad de exportar dichos resultados en un archivo en formato JSON. La Figura 30 indica la estructura del archivo exportado.

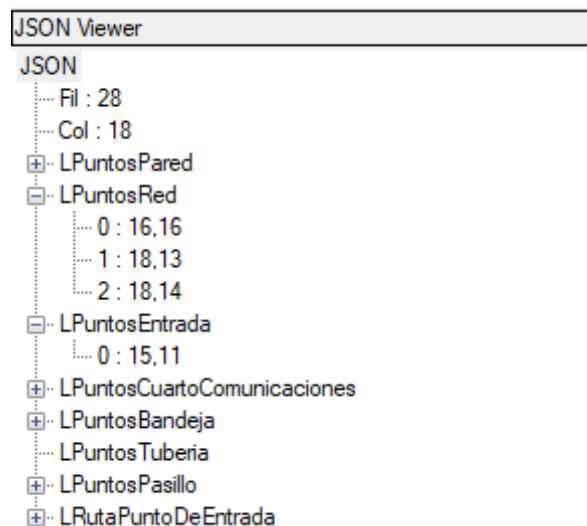


Figura 30. Estructura de archivo JSON resultado

Cada elemento del archivo contiene uno o varios elementos con las coordenadas fila-columna de los puntos que representan a paredes, puntos de red, puntos de entrada, cuarto de comunicaciones, secciones de bandeja, secciones de tubería y área de pasillo. En la Figura 31, se observa la representación de la carga del archivo resultado.

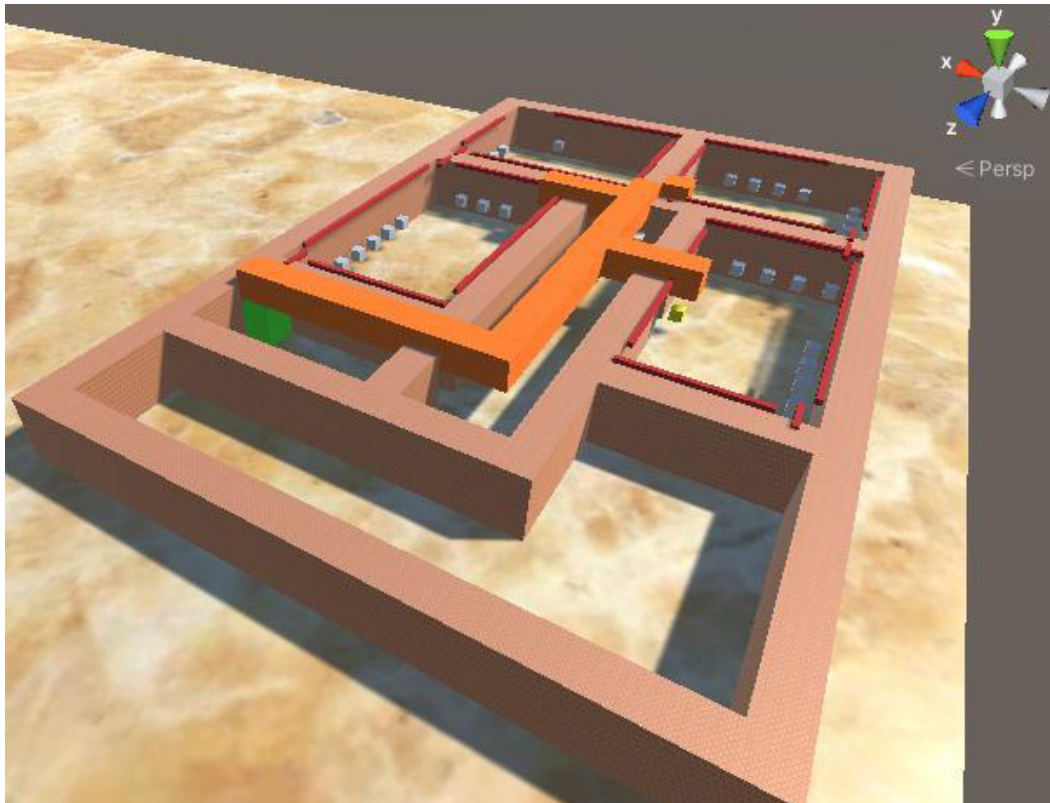


Figura 31. Representación del proyecto final dentro un EVA

3.5 Casos de utilización

En este apartado, expondremos casos representativos, que nos permitan entender de mejor manera el funcionamiento del aplicativo, conocer sus fortalezas y debilidades, y formar una idea clara de su empleo en ambientes reales.

3.5.1 Caso I – Cálculo de rutas sin optimización de resultados

En la Figura 32 a) visualizamos la distribución original del piso. Aquí se puede diferenciar a las áreas de trabajo, y dos posibles ubicaciones para el cuarto de telecomunicaciones. Es importante mencionar que para el diagrama original es necesario que solo estos elementos se encuentren presentes. Seguidamente en la Figura 32 b) tenemos la representación en la aplicación, aquí ya se han colocado dentro de cada área de trabajo los puntos de red mediante la letra “R” en color rojo, se han definido los posibles puntos de entrada a cada área de trabajo usando la letra “E” en color azul, y se han marcado las ubicaciones viables del gabinete dentro del cuarto de comunicaciones empleando la letra “S” en color verde.

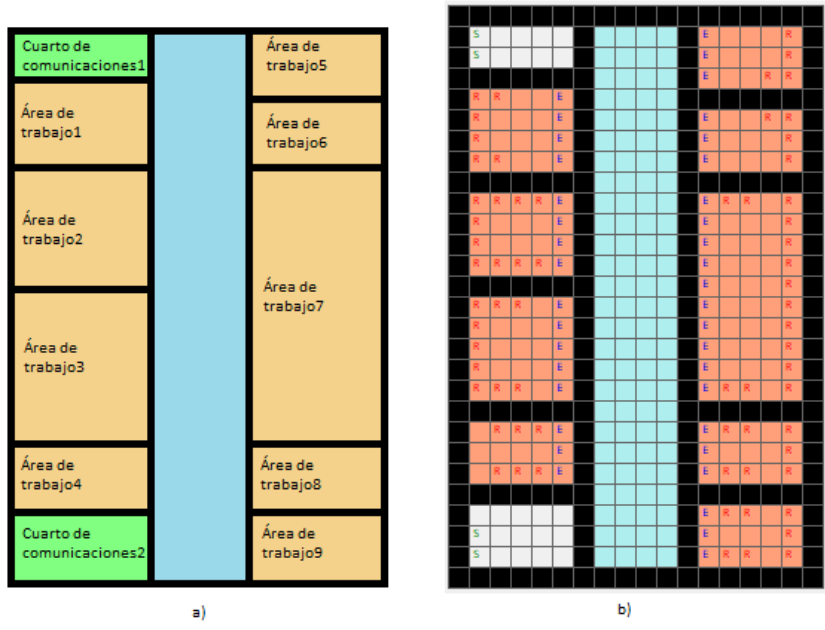


Figura 32. Representación gráfica del piso para el caso 1

Utilizando las herramientas disponibles en el aplicativo, se han definido los cuartos o áreas de trabajo que son de interés para el cómputo. A continuación ejecutamos el primer paso “Calcular Puntos de Entrada”. Mediante el uso del algoritmo A estrella, el aplicativo nos entregará resaltado en color amarillo el punto de entrada que minimiza los costos de cableado en cada área de trabajo. La Figura 33 nos muestra el resultado de la ejecución.

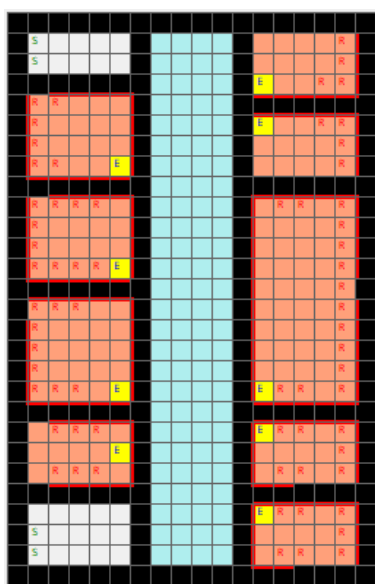


Figura 33. Resultado de ejecutar Paso1, proceso de cómputo de mejor punto de entrada mediante algoritmo A estrella.

Con los valores calculados que ahora posee el aplicativo, podemos avanzar y ejecutar el siguiente paso, el mismo que se encargará de procesar el algoritmo genético. En el diagrama observamos que tenemos dos posibles cuartos de comunicaciones y dentro de ellos dos ubicaciones factibles para el gabinete, en este caso el AG se encargará de seleccionar la solución que represente menor costo.

En la Figura 34, se trazan las rutas de la bandeja partiendo desde la mejor ubicación calculada para el gabinete de telecomunicaciones, hacia los mejores puntos de entrada de cada área de trabajo. El recorrido de la bandeja se identifica con sombreado naranja en cada uno de las celdas del diagrama.

Finalmente, en la sección “Ayuda y mensajes del aplicativo” se observa el resultado final de ejecución, la Figura 35 indica el total del proyecto de cableado.

La Tabla 2, resume las entradas con las que el aplicativo ejecutó los cálculos, y las respectivas salidas a pantalla.

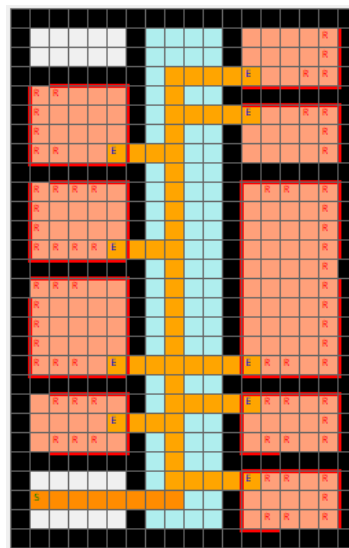


Figura 34. Resultado de ejecutar Paso2, proceso de cómputo de mejor ubicación del gabinete de telecomunicaciones mediante algoritmo genético.

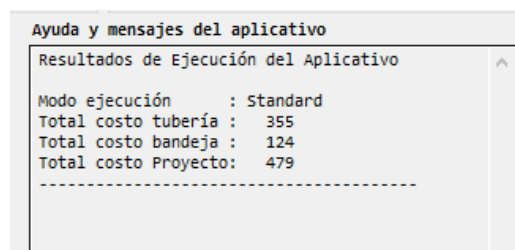


Figura 35. Total por proyecto

| Entradas | | Salidas | | |
|---------------------|---------------------|--------------------------|----------------------|--|
| Costo metro bandeja | Costo metro tubería | Ejecución Standard | Ejecución Optimizada | Comentario |
| USD 2.00 | USD 1.00 | Total CT 355 | N/A | El aplicativo determinó que la ejecución en modo standard es suficiente para establecer las rutas de bandeja y tubería, por lo que no habilita el botón para optimizar los cálculos iniciales. |
| | | Total CB 124 | | |
| | | Total Proyecto 479.00 | | |

Tabla 2. Resumen de los eventos estudiados en el caso I

3.3.2 Caso II – Cálculo de rutas con optimización de resultados

El caso a continuación, parte de la distribución de espacios del caso uno, pero se incluye una variante que permite al aplicativo optimizar los cálculos. En la Figura 36 se han resaltado los cambios introducidos y que básicamente son; eliminar puntos de red colaterales a los posibles puntos de entrada de las áreas de trabajo. De esta forma se consigue que los puntos de entrada y puntos de red queden frente a frente. Analizaremos la ejecución del aplicativo cuando encuentra estas situaciones.

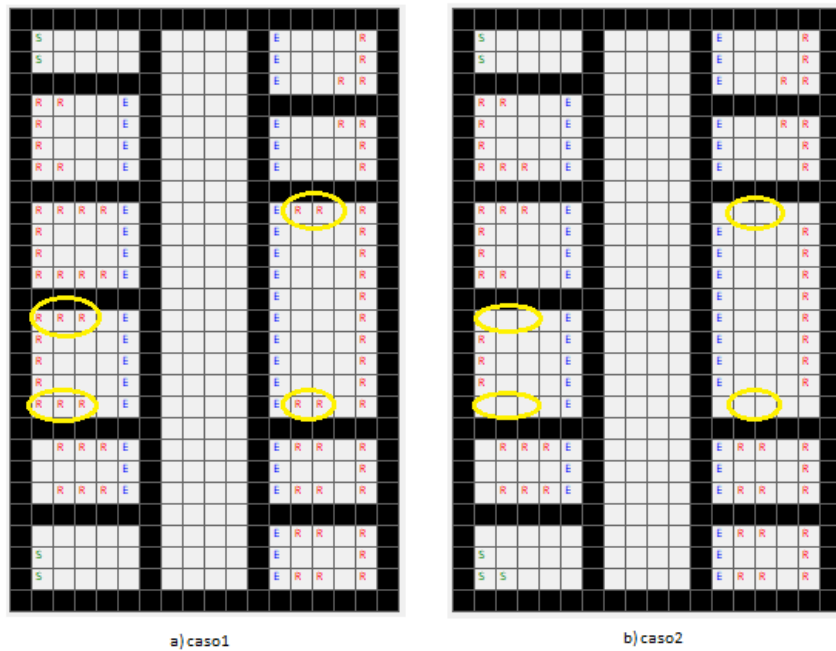


Figura 36. Representación gráfica de caso I y caso II

Luego de definir las diferentes secciones en la representación gráfica del piso, se realiza el cálculo de puntos de entrada. En la Figura 37. Se puede observar el resultado de este proceso.

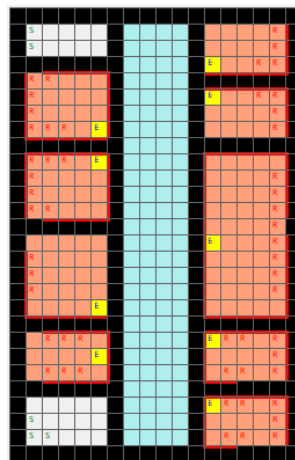


Figura 37. Resultado de “Calcular Puntos de Entrada”

Con los datos que ahora tiene el aplicativo, se ejecuta el proceso “Calcular Gabinete de Comms”. La Figura 38 indica el resultado gráfico de las rutas de bandejas y tuberías, y valor total del proyecto. Sin embargo se observa un mensaje especial del aplicativo, que hace referencia a optimizar los cálculos. Mediante el uso de esta opción, es posible minimizar el costo del proyecto dependiendo de los valores promedio por metro de tubería y bandeja que se ingresan antes de ejecutar los algoritmos de cálculo. La Figura 39. Presenta el resultado

gráfico final de cómo se debe realizar la colocación de bandeja y tubería, y el total del proyecto.

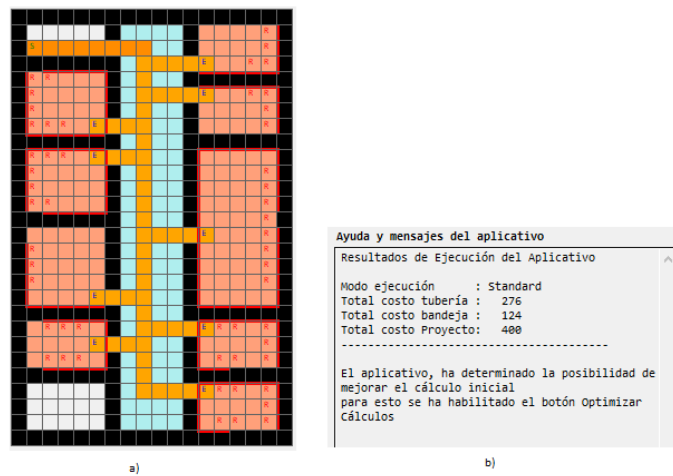


Figura 38. a) Resultado de “Calcular Gabinete de Comms”. b) Total por proyecto

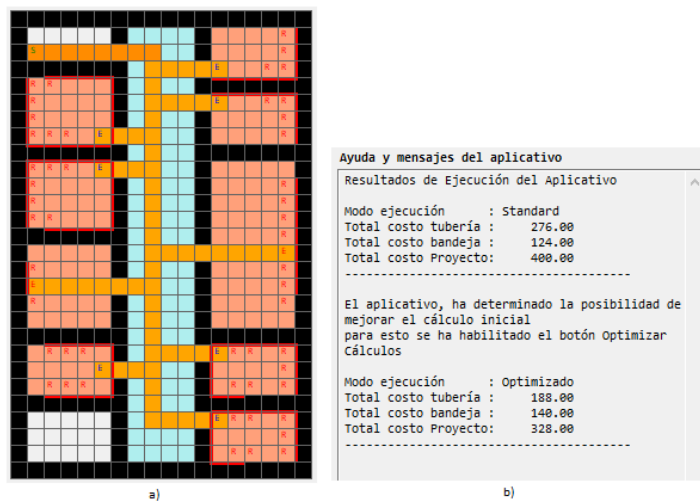


Figura 39. a) Resultado gráfico luego de optimizar cálculos. b) Total por proyecto

La Tabla 3, resume las entradas con las que el aplicativo ejecutó los cálculos, y las respectivas salidas a pantalla.

| Entradas | | Salidas | | |
|---------------------|---------------------|--------------------|----------------------|------------|
| Costo metro bandeja | Costo metro tubería | Ejecución Standard | Ejecución Optimizada | Comentario |
| USD 2.00 | USD 1.00 | Total CT | Total CT | |

| | | | | |
|--|--|----------------|----------------|--|
| | | 276.00 | 188.00 | bandeja ingresando al área de trabajo 3 y 7, en contraste con la ruta hacia las otras áreas. Este comportamiento del aplicativo asegura que el cálculo de los algoritmos en cada etapa, siempre estará guiado por minimizar los costos de enrutamiento. Para este caso, se determinó que ingresar con bandeja a estas secciones sería más económico que llegar a los puntos de entrada calculados mediante el modo standard. |
| | | Total CB | Total CB | |
| | | 124.00 | 140.00 | |
| | | Total Proyecto | Total Proyecto | |
| | | 400.00 | 328.00 | |

Tabla 3. Resumen de los eventos estudiados en el caso II

3.3.3 Caso III – Cálculo de rutas con optimización de resultados e inclusión de variables de tipo costo.

El caso que estudiaremos, se enfoca en el análisis de costos del proyecto, esto en función de los montos que recibe el aplicativo en los campos *Costo metro bandeja* y *Costo metro tubería*. Analizaremos el comportamiento de la aplicación cuando procesa estas circunstancias.

En la Figura 40, se observa la distribución de las diferentes áreas. Sin embargo, utilizaremos para efectos didácticos y de cálculo solo dos áreas de trabajo que se encuentran debidamente marcadas en color rosa, adicionalmente se encuentra marcado el pasillo central, y las dos posibles ubicaciones del gabinete de comunicaciones.

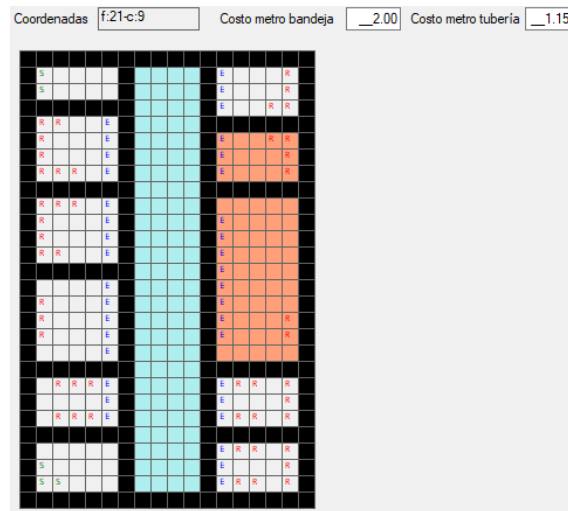


Figura 40. Representación gráfica del piso para Caso 3

En la Figura 41. Se presentan los resultados de ejecución. Aquí distinguimos que existe una mejora con respecto al costo total del proyecto, en la primera corrida el valor total es 97.85 USD, mientras que al optimizar el valor se reduce a 91.85 USD.

Ahora, se modificarán los valores por metro de instalación de bandeja y tubería. En la Figura 42. Vemos que el resultado del enrutamiento es idéntico al gráfico de la Figura 41; no obstante, el aplicativo nos indica mediante una nota luego del cálculo, que, si bien existe una ruta optimizada, por otro lado, los costos totales no son menores al cálculo inicial.

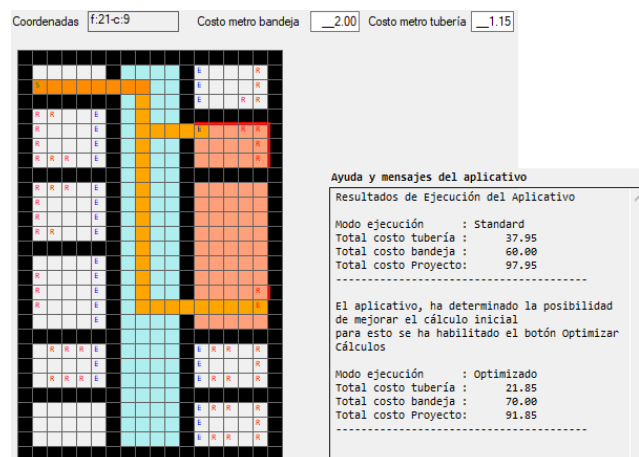


Figura 41. Resultado de ejecución del aplicativo con 2.00 USD y 1.15 USD de costo por metro de bandeja y tubería respectivamente

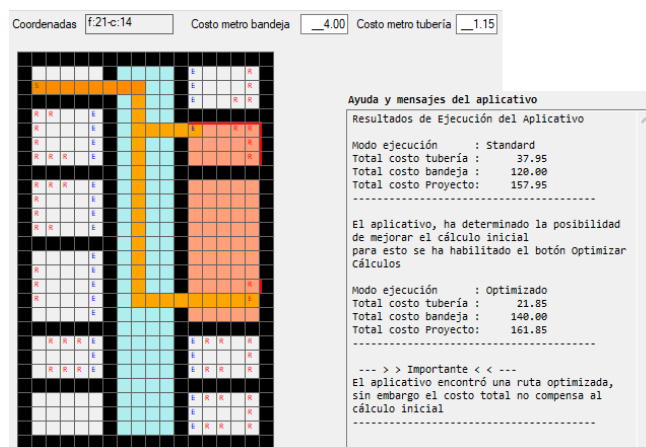


Figura 42. Resultado de ejecución del aplicativo con 4.00 USD y 1.15 USD de costo por metro de bandeja y tubería respectivamente

La Tabla 4, resume las diferentes entradas con las que el aplicativo ejecutó los cálculos y las respectivas salidas para cada una.

| Entradas | | Salidas | | |
|---------------------|---------------------|---|---|--|
| Costo metro bandeja | Costo metro tubería | Ejecución Standard | Ejecución Optimizada | Comentario |
| USD 2.00 | USD 1.15 | Total CT 37.95 Total CB 60.00 Total Proyecto 97.95 | Total CT 21.85 Total CB 70.00 Total Proyecto 91.85 | El aplicativo determina la existencia de un mejor cálculo en función de las rutas trazadas para la tubería y/o bandeja, el valor final optimizado se compara con el standard y presenta el resultado. |
| USD 4.00 | USD 1.15 | Total CT 37.95 Total CB 120.00 Total Proyecto | Total CT 21.85 Total CB 140.00 Total Proyecto | A pesar de encontrar un cálculo optimizado, este resulta mayor al standard, por consiguiente el aplicativo advierte al usuario esta descompensación. Sin embargo, el operador podría hacer uso de esta información para ejecutar un proceso de mejora de precios con proveedores |

| | | | | |
|--|--|--------|--------|--------------------|
| | | 157.95 | 161.85 | y minimizar costos |
|--|--|--------|--------|--------------------|

Tabla 4. Resumen de los eventos estudiados en el caso III

4 CONCLUSIONES Y RECOMENDACIONES

La implementación del artefacto propuesto en este presente trabajo, consiguió que el objetivo principal sea contrastado con la experiencia que el alumno desarrollaría en un laboratorio físico o mediante la construcción de modelos a escala.

El desarrollo permitió que varias de las normas, estándares y buenas prácticas de cableado estructurado sean consideradas y materializadas dentro de un proyecto. Al aplicarlas estamos claros que muchas de sus ventajas serán maximizadas, así también podremos minimizar las desventajas que en un momento dado podrían presentarse.

Representar gráficamente en dos dimensiones el problema, facilitó la interacción con los algoritmos que hacen parte del artefacto. La estrategia usada permitió dividir un gran problema en pequeños problemas dentro de cada área de trabajo. Mediante el uso de técnicas de búsqueda informada y exploración, y específicamente a través del algoritmo A Estrella, se pudo encontrar una solución efectiva, que en términos de recursos computacionales y de tiempo, entrega valores completamente aceptables.

Luego, la información calculada en la primera ejecución, se constituyó en los datos de entrada del algoritmo genético que se encargó de la optimización de los resultados y establecer la mejor ubicación del cuarto de cableado y por consiguiente la mejor posición para el gabinete de comunicaciones. Aquí, el resultado del análisis de las pruebas de los casos I, II y III, facilitó el afinamiento de la función de *fitness* que, como resultado mejoró los tiempos de ejecución.

Modificar los valores de entrada en cada caso analizado, permite interactuar sencillamente con la aplicación, pero lo más importante, es que nos entrega respuestas rápidas a interrogantes como: ¿Qué pasaría si elimino un punto de red? ¿Qué pasaría si el costo de tubería sube o baja? ¿Qué pasaría si agrego una nueva posición para el gabinete de comunicaciones? Esto sin duda habilita al alumno a resolver dudas que se le presentarían en un proyecto real.

Observar la implementación final en un entorno virtual de aprendizaje en 3D, eleva la percepción que el estudiante tiene al interactuar con los elementos del proyecto resultado. Las soluciones calculadas por los estudiantes en base a sus criterios, pueden ser evaluadas y comparadas contra la entregada por el artefacto.

El hecho de considerar y utilizar dentro de los cálculos a variables de entrada asociadas con los costos, en este caso: el metro lineal de bandeja y tubería respectivamente, genera en el alumno una cultura de minimización de costos y maximización de recursos que normalmente en el salón de clases no es contemplada.

Finalmente, queda completamente claro que la virtualización de experiencias sensoriales dentro de entornos virtuales de aprendizaje, facilita la enseñanza de las asignaturas que demandan de actividades prácticas como talleres, laboratorios.

Como trabajo a futuro, se recomienda introducir otro tipo de variables dentro de los cálculos, con la intención de generar un mayor nivel de interacción con la aplicación. Y de esta forma lograr que los alumnos puedan involucrar dentro de sus actividades de aprendizaje más elementos del mundo real que se encuentran relacionados con proyectos de cableado estructurado.

Se recomienda valorar la factibilidad de la implementación del artefacto en un solo ambiente de desarrollo y ejecución, y de esta forma evitar la generación de resultados y posterior carga.

Se sugiere introducir la posibilidad de visualizar la lista de materiales dentro de los resultados, de manera que se incluyan a todos los elementos propios de una instalación de un proyecto de cableado estructurado, junto con sus capacidades, marcas, restricciones, entre otros.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Berry, "Teaching to Connect: Community-Building Strategies for the Virtual Classroom," *Online Learn.*, vol. 23, no. 1, pp. 164–183, 2019.
- [2] P. Dillenbourg, D. Schneider, P. Synteta, and others, "Virtual learning environments," in *Proceedings of the 3rd Hellenic conference information & communication technologies in education*, 2002, pp. 3–18.
- [3] F. Gavilanes-Sagnay, E. Loza-Aguirre, D. Riofrío-Luzcando, and M. Segura-Morales, "A systematic literature review of indicators for the understanding of interactions in Virtual Learning Environments," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 596–600.
- [4] C. A. Sensibaugh, N. J. Madrid, H.-J. Choi, W. L. Anderson, and M. P. Osgood, "Undergraduate Performance in Solving Ill-Defined Biochemistry Problems," *CBE—Life Sci. Educ.*, vol. 16, no. 4, p. ar63, 2017.
- [5] N.-T. Le and W. Menzel, "Using Constraint-Based Modelling to Describe the Solution Space of Ill-defined Problems in Logic Programming," 2007, vol. 4823, pp. 367–379, doi: 10.1007/978-3-540-78139-4_33.
- [6] H. X. J. Viera and C. D. E. Carrillo, "Implementación de un entorno virtual de aprendizaje en 3D Mono-usuario para el entrenamiento en sistemas de cableado estructurado," ESCUELA POLITÉCNICA NACIONAL, 2020.
- [7] S. Dukda, "Introduction to structured cabling," *Div. Inf. Technol. Minist. Commun.*, 2000.
- [8] ANSI/TIA-568-C.2, "Balanced Twisted-Pair Telecommunication Cabling and Components Standard." .
- [9] ANSI/TIA-568.1-D, "Commercial Building Telecommunications Cabling Standard." .
- [10] V. M. Kureičhik, *Genetic algorithms for applied CAD problems*. Berlin: Springer-Verlag, 2009.
- [11] P. Brown, "CAD: Do computers aid the design process after all?," *Intersect Stanford J. Sci. Technol. Soc.*, vol. 2, no. 1, pp. 52–66, 2009.
- [12] P. Szalapaj, *CAD principles for architectural design : analytical approaches to computational representation of architectural form*. Oxford Boston: Architectural Press, 2001.
- [13] S. Russell, *Inteligencia artificial : un enfoque moderno*. Madrid: Pearson Prentice Hall, 2004.
- [14] D. Poole, A. Mackworth, and R. Goebel, *Computational Intelligence: A Logical Approach*. 1998.

- [15] N. Nilsson, *Artificial Intelligence : a new synthesis*. San Francisco, Calif: Morgan Kaufmann Publishers, 1998.
- [16] M. Nosrati, R. Karimi, and H. A. Hasanvand, "Investigation of the*(star) search algorithms: Characteristics, methods and approaches," *World Appl. Program.*, vol. 2, no. 4, pp. 251–256, 2012.
- [17] M. Affenzeller, *Genetic algorithms and genetic programming : modern concepts and practical applications*. Boca Raton: CRC Press, 2009.
- [18] S. Clinton, *Genetic algorithms with Python*. Place of publication not identified: publisher not identified, 2017.
- [19] R. Sutton, *Reinforcement learning : an introduction*. Cambridge, Massachusetts London, England: The MIT Press, 2018.
- [20] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv Prepr. arXiv1703.03864*, 2017.
- [21] A. Hevner *et al.*, "Design Science in Information Systems Research," *Manag. Inf. Syst. Q.*, vol. 28, pp. 75-, 2004.
- [22] K. Peffers *et al.*, "The design science research process: A model for producing and presenting information systems research," *Proc. First Int. Conf. Des. Sci. Res. Inf. Syst. Technol. DESRIST*, 2006.
- [23] S. March and G. Smith, "Design and Natural Science Research on Information Technology," *Decis. Support Syst.*, vol. 15, pp. 251–266, 1995, doi: 10.1016/0167-9236(94)00041-2.
- [24] W. Ibez, *Las ciencias de diseño : racionalidad limitada, predicción y prescripción*. La Coruña: Netbiblo, 2007.
- [25] I. Sommerville, *Ingeniería del software*. Pearson educación, 2005.

6 ANEXOS

Anexo I

```
PROCEDURE CalcularPuntosDeEntrada
```

```
BEGIN
```

```
    IF listadecuartos = 0 OR listadepasillos = 0
```

```
    BEGIN
```

```
        WRITELN "No se han seleccionado cuartos y/o pasillos para calcular distancias"
```

```
    END PROCEDURE.
```

```
    END
```

```
    CALL CargarPuntosDesdePanellImagen
```

```
    FOR EACH item in listadecuartos
```

```
    BEGIN
```

```
        CALL CalcularMejorPuntoDeEntradaXCuarto
```

```
        CALL GraficarRutas
```

```
    END
```

```
END
```

```
PROCEDURE CargarPuntosDesdePanellImagen
```

```
BEGIN
```

```
    DECLARE ARRAY listapuntospared OF PUNTO
```

```
    DECLARE ARRAY listapuntosred OF PUNTO
```

```
    DECLARE ARRAY listapuntosentrada OF PUNTO
```

```
    DECLARE ARRAY listapuntosrack OF PUNTO
```

```
    FOR EACH i IN FILAS
```

```
        FOR EACH j IN COLUMNAS
```

```
            IF panel[i,j].BackColor = "Black"
```

```
                listapuntospared.ADD(Punto(i,j))
```

```
            CASE panel[i,j].text
```

```
                "R": listapuntosred.ADD(Punto(i,j))
```

```
                "S": listapuntosrack.ADD(Punto(i,j))
```

```
                "E": listapuntosentrada.ADD(Punto(i,j))
```

```
                "ER": listapuntosentrada.ADD(Punto(i,j)),
```

```
listapuntosred.ADD(Punto(i,j))
```

```
END
```

```
PROCEDURE CalcularMejorPuntoDeEntradaXCuarto
```

```
BEGIN
```

```
    DECLARE Punto startLocation = Cuarto.PuntoEntrada
```

```
    DECLARE Punto endLocation = Cuarto.PuntoRed
```

```
    DECLARE ARRAY IPuntos = AlgoritmoAEstrella.FindPath();
```

```
EACH
```

```
PROCEDURE ALGORITMOESTRELLA.FINDPATH
```

```
BEGIN
```

```
    DECLARE NODO_START
```

```
    DECLARE NODO_END
```

```
    DECLARE NODO_ACTUAL
```

```
    DECLARE ARRAY listaAbierta, listacerrada
```

```
    listaAbierta.ADD(NODO_START)
```

```
    WHILE listaAbierta IS NOT EMPTY
```

```

BEGIN
  IF (NODO_ACTUAL = NODO_END)
    DEVOLVER nodos_sucesores
  FOR EACH nodos_sucesores OF NODE_CURRENT
    BEGIN
      SUCCESSOR_CURRENT_COST = G(NODE_CURRENT) + W(NODE_CURRENT,
NODE_SUCESSOR)
      IF NODE_SUCESSOR IS IN THE listaAbierta
        IF G(NODE_SUCESSOR) <= SUCCESSOR_CURRENT_COST CONTINUE
      ELSE IF NODE_SUCESSOR IS IN THE listacerrada
        IF G(NODE_SUCESSOR) <= SUCCESSOR_CURRENT_COST CONTINUE
        MOVE NODE_SUCESSOR FROM THE listacerrada TO THE
listaAbierta
      ELSE
        ADD NODE_SUCESSOR TO THE listaAbierta
        SET H(NODE_SUCESSOR) TO BE THE HEURISTIC DISTANCE
TO NODE_GOAL
      SET G(NODE_SUCESSOR) = SUCCESSOR_CURRENT_COST
      SET THE PARENT OF NODE_SUCESSOR TO NODE_CURRENT
    END
    ADD NODE_CURRENT TO THE CLOSED LIST
  END
END

```

Anexo II

```
PROCEDURE CalcularCuartoDeComunicaciones
BEGIN
    DECLARE AlgoritmoGenetico ag
    DECLARE Punto mejorPuntoCC
    CALL mejorPuntoCC = ag.Main
    IF mejorPuntoCC = null
        WRITE "No fue posible calcular cuarto de comunicaciones"
    ELSE
        BEGIN
            CALL DibujarRutaCuartoComunicacionesAPuntosEntrada
        END
    END
END
```

```
PROCEDURE ALGORITMOGENETICO
BEGIN
    DECLARE FLOAT PORCENTAJE_ELITE_SELECCIONADA
    DECLARE INTEGER maximoValorManhatan
    DECLARE INTEGER populationSize
    DECLARE BOOLEAN CALCULAR
    DECLARE ARRAY soluciones OF INTEGER,INTEGER
    CALL GENERARPOBLACIONINICIAL

    WHILE CALCULAR
        BEGIN
            CALCULAR = EVALUAR()
        END
    END
END
```

```
PROCEDURE GENERARPOBLACIONINICIAL
BEGIN
    FOR i = 0 TO populationSize
        FOR j = 0 to listapuntosred
            soluciones[i,j] = RAND( maximoValorManhatan )
        END
    END
END
```

```
PROCEDURE EVALUAR
BEGIN
    DECLARE INTEGER fitness = 0
    DECLARE INTEGER distancia = 0
    FOR I = 0 TO soluciones.UpperBound
        BEGIN
            fitness = 0
            FOR J = 0 TO listapuntosred
                BEGIN
                    distancia = CALCULODISTANCIAMANHATAN
                    IF soluciones[i,j]=distancia
                        fitness = fitness + 1
                    ELSE
                        fitness = fitness + 0
                END
            END
        END
    END
END
```

```

        lResultados.Add(new Resultado(i, fitness));
    END

    //CONDICION DE SALIDA
    IF listaresultados[0].Fitness = lpuntosred
        RETURN FALSE

    //OPERACION: SELECCION DE RESULTADOS, ELITE SELECCIONADA
    DECLARE ARRAY auxSoluciones OF INTEGER, INTEGER
    FOR I = 0 TO PORCENTAJE_ELITE_SELECCIONADA
        auxSoluciones = soluciones[i]

    //OPERACION: CRUZAMIENTO ALEATORIO
    FOR I = 0 TO numeroSolucionesSeleccionadas
    BEGIN
        FOR J = 0 TO listapuntosred
            IF J < crossoverpoint
                soluciones[i,j] = auxSoluciones[padre1, j]
            ELSE
                soluciones[i,j] = auxSoluciones[padre2, j]
        END
    END

    //OPERACION: MUTACION ALEATORIA
    DECLARE INTEGER pos1
    DECLARE INTEGER pos2
    FOR I = 0 TO numeroSolucionesSeleccionadas
    BEGIN
        pos1 = randomNumber
        pos2 = randomNumber
        soluciones[i,pos1] = randomNumber
        soluciones[i,pos2] = randomNumber
    END

    RETURN TRUE
END

```

Anexo III

PROCEDURE STARTUP

BEGIN

```
DECLARE posInicialX FLOAT = -35
DECLARE posInicialZ FLOAT = -45
DECLARE ajuste FLOAT = 1.5f
DECLARE rutaArchivo Varchar(100)
DECLARE pisoModel = JSONUTILITY.FromJson(rutaArchivo)
DECLARE vPosicionInicial Vector3
DECLARE vParedScale Vector3
DECLARE puntoRedScale Vector3
DECLARE vPuntoCuartoComunicacionesScale Vector3
DECLARE vPuntoBandejaScale Vector3
DECLARE vPuntoTuberiaScale Vector3
DECLARE vPuntoTuberiaRotationX Quaternion
DECLARE vPuntoTuberiaRotationY Quaternion
DECLARE mPared Material
DECLARE mPuntoRed Material
DECLARE mPuntoEntrada Material
DECLARE mPuntoBandeja Material
DECLARE mPuntoTuberia Material
DECLARE mPuntoCuartoComms Material
DECLARE IPared List of GameObject
FOR i = 0 TO pisoModel.LPuntosPared
    DECLARE ARRAY coordenadasFilCol = LPuntosPared[i].Split(',');
    DECLARE cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
    cube.GetComponent<Renderer>().material = mPared;
    cube.transform.localScale = vParedScale;
    coordenadaX = posInicialX + (float.Parse(coordenadasFilCol[1]) * ajuste);
    coordenadaZ = posInicialZ + (float.Parse(coordenadasFilCol[0]) * ajuste);
    cube.transform.position = new Vector3(coordenadaX, vPosicionInicial.y, coordenadaZ);
    IPared.Add(cube);
FOR i = 0 TO pisoModel.LPuntosRed
    DECLARE ARRAY coordenadasFilCol = LPuntosRed[i].Split(',');
    DECLARE cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
    cube.GetComponent<Renderer>().material = mPuntoRed;
    cube.transform.localScale = vParedScale;
    coordenadaX = posInicialX + (float.Parse(coordenadasFilCol[1]) * ajuste);
    coordenadaZ = posInicialZ + (float.Parse(coordenadasFilCol[0]) * ajuste);
    cube.transform.position = new Vector3(coordenadaX, vPosicionInicial.y, coordenadaZ);
FOR i = 0 TO pisoModel.LPuntosEntrada
    DECLARE ARRAY coordenadasFilCol = LPuntosEntrada[i].Split(',');
    DECLARE cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
    cube.GetComponent<Renderer>().material = mPuntoEntrada;
    cube.transform.localScale = vParedScale;
    coordenadaX = posInicialX + (float.Parse(coordenadasFilCol[1]) * ajuste);
    coordenadaZ = posInicialZ + (float.Parse(coordenadasFilCol[0]) * ajuste);
    cube.transform.position = new Vector3(coordenadaX, vPosicionInicial.y, coordenadaZ);
FOR i = 0 TO pisoModel.LRutaPuntoDeEntrada
    DECLARE ARRAY coordenadasFilCol = LRutaPuntoDeEntrada[i].Split(',');
    DECLARE cube = GameObject.CreatePrimitive(PrimitiveType.Cube);
    cube.GetComponent<Renderer>().material = mPuntoEntrada;
    cube.transform.localScale = vParedScale;
    coordenadaX = posInicialX + (float.Parse(coordenadasFilCol[1]) * ajuste);
    coordenadaZ = posInicialZ + (float.Parse(coordenadasFilCol[0]) * ajuste);
```

```

        cube.transform.position = new Vector3(coordenadaX, vPosicionInicial.y, coordenadaZ);
        cube.transform.position = ObtenerVector3Tuberia(ref IPared, cube);
END
PROCEDURE ObtenerVector3Tuberia
BEGIN
    DECLARE PosicionPared pp
    DECLARE pDerecha GameObject
    DECLARE plzquierda GameObject
    DECLARE pArriba GameObject
    DECLARE pAbajo GameObject
    DECLARE vector3 Vector3
    IF IPared.Exists(x => x.transform.position.x == pDerecha.transform.position.x
        AND x.transform.position.z ==
pDerecha.transform.position.z))
        BEGIN
            pp = PosicionPared.Derecha;
        END
    IF (IPared.Exists(x => x.transform.position.x == plzquierda.transform.position.x
        AND x.transform.position.z ==
plzquierda.transform.position.z))
        BEGIN
            pp = PosicionPared.Izquierda;
        END
    IF (IPared.Exists(x => x.transform.position.x == pArriba.transform.position.x
        AND x.transform.position.z == pArriba.transform.position.z))
        BEGIN
            pp = PosicionPared.Arriba;
        END
    IF (IPared.Exists(x => x.transform.position.x == pAbajo.transform.position.x
        AND x.transform.position.z == pAbajo.transform.position.z))
        BEGIN
            pp = PosicionPared.Abajo;
        END

    CASE SWITCH pp
    BEGIN
        PosicionPared.Derecha:
            vector3.Set(pTuberia.transform.position.x + (0.5f),
pTuberia.transform.position.y, pTuberia.transform.position.z)
            break
        PosicionPared.Izquierda:
            vector3.Set(pTuberia.transform.position.x + (-0.5f),
pTuberia.transform.position.y, pTuberia.transform.position.z)
            break
        PosicionPared.Arriba:
            vector3.Set(pTuberia.transform.position.x, pTuberia.transform.position.y,
pTuberia.transform.position.z + (0.5f))
            break
        PosicionPared.Abajo:
            vector3.Set(pTuberia.transform.position.x, pTuberia.transform.position.y,
pTuberia.transform.position.z + (-0.5f))
            break
        default:
            break
    END
    RETURN vector3
END

```