

**DISEÑO Y DESARROLLO DE UN ROBOT ASISTENTE
DE RUTA PARA LA SEDE PRINCIPAL DE LA
UNIVERSIDAD SANTO TOMÁS**

**CARLOS ALBERTO BOHÓRQUEZ CERÓN
MARÍA ELIZABETH PÁEZ RAMÍREZ**

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C.**

2018

**DISEÑO Y DESARROLLO DE UN ROBOT ASISTENTE
DE RUTA PARA LA SEDE PRINCIPAL DE LA
UNIVERSIDAD SANTO TOMÁS**

**CARLOS ALBERTO BOHÓRQUEZ CERÓN
MARÍA ELIZABETH PÁEZ RAMÍREZ**

Trabajo de grado

**Director:
ING. CARLOS ANDRÉS QUINTERO PEÑA**

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
BOGOTÁ D.C.**

2018

CONTENIDO

	pág.
1. PROBLEMA	5
2. ANTECEDENTES	6
3. JUSTIFICACIÓN	8
4. OBJETIVOS	9
4.1. OBJETIVO GENERAL	9
4.2. OBJETIVOS ESPECÍFICOS	9
5. MARCO TEÓRICO	10
5.1. ROBÓTICA	10
5.1.1. DEFINICIÓN DE ROBOT	10
5.1.2. MODELO DE INTERACCIÓN HUMANO-MÁQUINA	10
5.2. TEORÍA DE GRAFOS	11
5.2.1. ALGORITMO DE DIJKSTRA	12
5.3. SISTEMAS EMBEBIDOS	13
5.3.1. COMPILACIÓN CRUZADA	14

6. DISEÑO DE LA SOLUCIÓN	15
6.1. MODELAMIENTO DEL AMBIENTE	15
6.1.1. PLANOS FÍSICOS	15
6.1.2. IDENTIFICACIÓN DE SITIOS DE INTERÉS	16
6.1.3. DEFINICIÓN DE ARCOS	17
6.2. CARACTERIZACIÓN DE ALGORITMOS	17
6.2.1. ALGORITMO A*	18
6.2.2. ALGORITMO ANT COLONY OPTIMIZATION	18
6.2.3. LIBRERÍA IGRAPH	19
6.2.4. RESULTADOS	21
6.3. DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE ASISTENTE DE RUTA	22
6.3.1. PLANEAMIENTO DE REQUERIMIENTOS FUNCIONALES	22
6.3.2. SELECCIÓN DE ELEMENTOS DE HARDWARE	23
6.3.3. SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN	24
6.3.4. SELECCIÓN DEL ENTORNO DE DESARROLLO	25
6.3.5. DISEÑO UML	25
6.4. VALIDACIÓN DEL SISTEMA	27
6.4.1. DISEÑO DE LA ENCUESTA	27
6.4.2. SELECCIÓN DE LA MUESTRA	28
7. RESULTADOS	29
7.1. ESTADÍSTICAS PARA LOS TIEMPOS DE CÓMPUTO	29

7.2. SOFTWARE ASISTENTE DE RUTA	31
7.3. RESULTADOS DE LA ENCUESTA	36
8. IMPACTO SOCIAL	43
9. CONCLUSIONES	44
10. TRABAJO FUTURO	45
11. BIBLIOGRAFÍA	47
ANEXOS	49
ANEXO A. Tabla de sitios de interés	49
ANEXO B. Encuesta	56
ANEXO C. Planos	60

1. PROBLEMA

Actualmente, en la sede principal de la Universidad Santo Tomás, existe dificultad con respecto al recorrido que se debe realizar para llegar a los diferentes puntos de interés dentro de esta, como salones, auditorios u oficinas. Una de las principales causas es la ausencia de un sistema de guía apropiado, lo cual dificulta el recorrido a una persona que no está muy familiarizada con la Universidad. La ausencia de apoyo especializado a los visitantes provoca inconvenientes como por ejemplo retrasos para realizar los diferentes trámites.

Como solución a este problema se desea diseñar y posteriormente construir una plataforma robótica que proporcione, a través de su interacción con el humano, un sistema de guía e información para un usuario que desea trasladarse desde un punto de interés a otro dentro de la sede principal, entregándole la ruta óptima de desplazamiento hasta su destino.

Con una aplicación de este tipo la solución se puede expandir a que contenga información no solo de la sede principal, sino que el código fuente pueda ser proporcionado a otros grupos de trabajo que implementen el sistema en sedes como el campus San Alberto Magno y la sede en la calle 72 Dr. Angélico.

Un problema subyacente es la falta de información sobre tramites, procesos académicos y administrativos, aunque este es un problema fuera del alcance del proyecto, al tratarse de una solución mayormente de software, no está cerrada a futuras mejoras. Específicamente hablando de información sobre trámites y procesos, el proyecto original está abierto a revisiones y propuestas por parte de cualquier grupo de trabajo, que pueda añadir un valor agregado al resolver un problema de naturaleza social.

2. ANTECEDENTES

Las aplicaciones de planeación de ruta están en constante expansión en variados entornos. Recientemente se han propuesto nuevas soluciones que optimizan los algoritmos tradicionales de cálculo de ruta óptima.

En la Universidad de Toronto se diseñó un sistema de cálculo de ruta el cual presenta una mejora en costos para sistemas que no requieren componentes de infraestructura para ser implementados [1]. En ambientes más complejos existen avances actualmente en torno a la simulación de trayectorias en escenarios virtuales, los cuales son más flexibles a la hora de evaluar el desempeño de los sistemas de navegación [2].

Una de las principales áreas que demanda algoritmos de planeación de ruta es la robótica móvil, en la que se debe tener en cuenta no solo el camino óptimo sino también factores como la información del entorno, medida por distintos sensores [3] [4]. Los algoritmos tradicionales de planeación de ruta óptima han sido mejorados para adaptarse a distintas necesidades como obstáculos encontrados en el camino, de forma que la trayectoria se actualice en tiempo real para suplir dicha necesidad [5].

En el campo militar se ven algoritmos aún más robustos que tienen en cuenta condiciones adicionales como la longitud mínima de la ruta y el máximo ángulo de giro para entregar resultados de mayor precisión [6].

A nivel nacional se han visto avances en aplicaciones directas de los algoritmos de planeación de ruta como la asistencia en situaciones de emergencia, en las que se necesita de un trazado de escape frente a una catástrofe. La Facultad de Ingeniería Electrónica de la Universidad Santo Tomás de Tunja diseñó una aplicación basada en el procesamiento de imágenes para identificación de obstáculos con un análisis posterior de la viabilidad de tránsito por la ruta determinada [7].

Con respecto a la topología de los lugares en que se implementan sistemas de navegación, existen enfoques con sistemas difusos en los que se definen reglas lógicas, basadas en datos que se conocen sobre el entorno, para tener una base de conocimiento previa al cálculo de una ruta [8].

Muchos nuevos algoritmos de optimización de trayectorias han sido evaluados para valorar su rendimiento frente a los algoritmos tradicionales. En algunos casos presentan

ventajas como la reducción de consumo de memoria en sistemas embebidos [9]. Otros resultados positivos provienen de algoritmos de post-procesamiento como los algoritmos de filtrado de ida y suavizado de vuelta, que analizan datos adicionales en tiempo real para mejorar la planeación de trayectorias en curso [10]. También hay avances actualmente que expanden la utilidad de los sistemas de navegación dándoles valor agregado dentro del sector industrial y comercial. Es este el caso del uso de teoría de grafos en la planeación de rutas para sitios turísticos, con mejoras tanto para el visitante que tendrá un camino óptimo con sus puntos de interés, como para el operador que puede prever el flujo de visitantes a puntos específicos con capacidad limitada [11].

En el estudio de MM, por sus siglas en ingles MAP MATCHING, como método de obtención de trayectorias, existen en la actualidad mejoras sobre los modelos tradicionales, que llevan las curvas obtenidas a formas más naturales para un peatón, llegando incluso a tener en cuenta variables como la velocidad del peatón para el desarrollo de algoritmos de MM dentro de centros comerciales [12].

3. JUSTIFICACIÓN

La sede principal de la Universidad está físicamente compuesta de varios edificios interconectados en puntos específicos por lo que no existen rutas directas entre todos los puntos. Esto dificulta la movilidad dentro de la universidad, específicamente porque las conexiones entre edificios no son claras a una persona que se moviliza dentro de la Universidad.

Este panorama hace que desplazarse de un punto a otro, especialmente si estos están en edificios separados, resulte en un recorrido poco natural, por lo que las personas que no conocen el camino tienden a extender su recorrido más tiempo del necesario, debido a que toman rutas que parecen más cortas pero que no llevan al lugar que necesitan.

La implementación de un asistente de ruta para la totalidad de los visitantes es una forma de fomentar la buena atención al usuario de la Universidad Santo Tomás. Además, posiciona a la Facultad de Ingeniería Electrónica como generador de soluciones tecnológicas de impacto al interior de la USTA y para otros sectores empresariales.

Este proyecto abre las puertas a implementaciones futuras, como la incorporación de descripciones detalladas de cada dependencia de la Universidad y sus funciones, así como una expansión de la aplicación que no la limite a estar presente en puntos estáticos dentro de la Universidad, sino que sea accesible a los visitantes de forma portable a través de su dispositivo móvil.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Diseñar y desarrollar un robot asistente que provea al usuario la ruta óptima de desplazamiento entre dos puntos de interés dentro de la sede principal de la Universidad Santo Tomás.

4.2. OBJETIVOS ESPECÍFICOS

- Implementar y comparar el rendimiento de distintos algoritmos de planeación de rutas.
- Construir un mapa de ubicaciones de los puntos de interés involucrados en el proyecto.
- Proponer e implementar un sistema de interacción humano-máquina que permita al usuario solicitar la ruta óptima para movilizarse entre dos puntos de interés dentro de la sede principal de la Universidad.
- Implementar uno de los algoritmos de ruta óptima e integrarlo en el sistema de interacción humano-máquina para encontrar la ruta más corta entre dos puntos de interés dentro de la sede principal de la Universidad Santo Tomás.
- Validar el desempeño del sistema seleccionado de acuerdo a su funcionalidad y su correcta interacción con el usuario.

5. MARCO TEÓRICO

5.1. ROBÓTICA

El término robótica, atribuido a Isaac Asimov, se deriva de la palabra robot, por lo que se define como la rama de la ciencia que se ocupa del estudio, desarrollo y aplicaciones de los robots. Esta disciplina tiene como finalidad la realización de tareas por parte de máquinas automáticas programables, que pueden ir desde el ensamble de automóviles, hasta la asistencia personalizada en la consecución de tareas cotidianas [13].

5.1.1. DEFINICIÓN DE ROBOT

En robótica existen diversas definiciones para “Robot” sin embargo, considerando el enfoque del proyecto, la definición de robot más adecuada para este caso es que un robot es un dispositivo que puede estar conectado a un computador, y obtiene los datos de entrada mediante sensores. El computador al recibir la información de entrada envía una orden al robot para que este ejecute una acción determinada [13].

El robot necesita tanto de un componente de software como de hardware. Ambos soportes son imprescindibles para la operación del sistema en su totalidad. Siendo el hardware el conjunto de circuitos electrónicos, cables, dispositivos electromecánicos y demás elementos físicos. El software hace referencia a el conjunto de programas mediante los cuales el robot cumple con las tareas programadas [14].

5.1.2. MODELO DE INTERACCIÓN HUMANO-MÁQUINA

La forma de conectar el usuario con la máquina es la base del funcionamiento de la solución que se propone, para que esta conexión sea exitosa es necesario entonces aplicar un modelo de interacción humano-máquina adecuado. El modelo de interacción de D.A. Norman es el de más influencia en este campo, en parte por su cercanía con nuestro entendimiento intuitivo de la interacción entre un humano y un computador.

El modelo de Norman conocido como el ciclo de ejecución-evaluación esencialmente dice que: El usuario formula un plan de acción, que luego es ejecutado en la interfaz de

computador. Cuando el plan, o parte del plan, ha sido ejecutado, el usuario observa la interfaz para evaluar el resultado del plan ejecutado, y determinar futuras acciones.

El modelo de Norman se divide en dos fases principales, ejecución y evaluación, que a su vez están divididas en un total de 7 etapas. Las etapas del modelo de interacción de Norman son:

- Establecer la meta..
- Formar los objetivos.
- Especificar la secuencia de acciones.
- Ejecutar la acción.
- Percibir el estado del sistema.
- Interpretar el estado del sistema.
- Evaluar el estado del sistema respecto a las metas y objetivos [15].

5.2. TEORÍA DE GRAFOS

Dependiendo de la tarea que se busque realizar, habrá determinadas rutinas en el software que al ser ejecutadas de forma sincronizada permiten la consecución de la tarea necesaria. En el caso del problema de encontrar la ruta más corta entre dos puntos, dichas rutinas constan esencialmente de algoritmos basados en lo que se conoce como teoría de grafos, definida a continuación:

Los grafos son estructuras de datos no lineales, que pueden ser usadas para modelar diversas aplicaciones. Son una parte importante de las matemáticas, más específicamente de la teoría combinatoria, lo que le da un carácter bastante amplio y complejo.

Un grafo G consta de un conjunto de vértices V y un conjunto de arcos A , los cuales unen un vértice con otro.

$$G = (V, A) = (N, A)$$

Los arcos que unen los nodos también se llaman aristas y se representan por medio de un par de elementos, (v_i, v_j) , donde los elementos son los nodos que une el arco. Los nodos de un grafo se pueden usar para representar los objetos y los arcos para representar las relaciones entre esos objetos [16].

5.2.1. ALGORITMO DE DIJKSTRA

La implementación de la teoría de grafos es un campo muy explorado, por lo que en la actualidad existen varias aproximaciones a la resolución de este problema, tales como el algoritmo de Dijkstra, el cual está definido como:

Sea G un grafo ponderado. Para encontrar la ruta P entre dos vértices v_0 y v_k con la mínima distancia. La distancia de una ruta $P = (v_0, v_1, \dots, v_k)$ es la suma de las distancias de sus aristas constituyentes:

$$w(P) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

A continuación, se muestra la implementación en pseudocódigo del algoritmo de Dijkstra.

Procedure Dijkstra(**graph** G , **vertex** $root$)

$d(root) = 0$

$Parent[root] = -1$

for each vertex $v \neq root$ **do**

$d(v) = \infty$

$Parent[v] = -1$

end-of-for

$T = V(G)$

$U = root$

while ($T \neq \emptyset$) **do**

$T = T \setminus \{ u \}$ // u es vértice procesado.

$N(u) = getNeighborhood(u)$

for each vertex $v \in N(u)$ **do**

```

if( $v \in T$  and  $d(v) > d(u) + w(u,v)$ ) then

     $d(v) = d(u) + w(u,v)$ 
    Parent[v] = u

end-of-for

    U = getMinEstimate(T)

end-of-while
return

end-of-procedure [16].

```

5.3. SISTEMAS EMBEBIDOS

Un sistema computacional en tiempo real diseñado para realizar ciertas funciones específicas, se conoce como un sistema embebido. Se refiere a estos dispositivos como embebidos ya que la mayoría de sus componentes como procesador, interfaces de comunicación, memoria volátil y no volátil están contenidos en una placa central que a su vez suele ser puesta en un encapsulado con un aspecto que difiere de la visión tradicional de un computador. Las características básicas de un sistema embebido son:

- **Confiabilidad:** Es la probabilidad de que el sistema trabaje correctamente dado que está funcionando.
- **Mantenibilidad:** Es la probabilidad de que el sistema vuelva a trabajar correctamente tras tener un fallo.
- **Disponibilidad:** Es la probabilidad de que el sistema este funcionando en un momento dado.
- **Seguridad informática:** Disposición de comunicación confidencial y autenticada.

Adicionalmente, los sistemas embebidos funcionan con una fuente de alimentación limitada por lo que deben regular de manera eficiente el consumo en todo momento. Esta propiedad permite alimentar un sistema embebido ya sea directamente desde la red eléctrica mediante un convertidor de corriente AC-DC, o haciendo uso de una batería con voltaje adecuado [17].

5.3.1. COMPILACIÓN CRUZADA

Un compilador se compone de tres lenguajes, el lenguaje fuente F es el encargado de compilar, el lenguaje objeto O para el cual se genera el código y el lenguaje de implantación I en el que está escrito el compilador. No es necesario que los tres lenguajes del compilador sean semejantes, y hay casos en los que incluso una máquina ejecuta un compilador para producir código para otra máquina, este tipo de compilador es el que se conoce como compilador cruzado.

Un compilador cruzado es ideal en situaciones en las que, por comodidad o limitaciones en la plataforma de software, no es lo más adecuado compilar el código fuente de manera local. En estos casos, el compilador cruzado ofrece la posibilidad de crear una aplicación desde una máquina remota con mejores capacidades, y posteriormente desplegar los archivos necesarios para la ejecución del programa en la máquina de destino [18].

6. DISEÑO DE LA SOLUCIÓN

El diseño del asistente de ruta se dividió en etapas de selección de las herramientas a usar, y posteriormente de implementación de las mismas. En esta sección se describe el proceso mediante el cual se dio construcción a una aplicación que resuelve de forma efectiva el problema de encontrar la ruta mas corta hacia un destino dentro de la sede principal de la Universidad.

6.1. MODELAMIENTO DEL AMBIENTE

Resolver el problema de encontrar la ruta óptima entre un origen y destino cualquiera dentro de la Universidad requiere encontrar un equivalente matemático a la distribución espacial de los edificios que componen la sede principal. El modelo que mejor se acomodó a las necesidades del proyecto fue el de teoría de grafos, el cual permite modelar la totalidad del espacio de la Universidad como un grafo, que a su vez se compone de varios sitios de interés, llamados vértices, y varias uniones entre estos sitios llamadas arcos. De forma que el modelo matemático de la Universidad son todas las ubicaciones a las que un usuario puede llegar, y las distancias que hay entre una ubicación cualquiera y otra, siempre y cuando dicha distancia pueda ser recorrida por el usuario.

Tener el modelo matemático equivalente de la Universidad no es suficiente para saber las distancias mínimas de desplazamiento entre dos puntos cualquiera, el cual es finalmente el problema principal del proyecto, por lo que fue necesario implementar algoritmos de solución al problema de la ruta más corta y compararlos, para poder hacer una elección informada de cuál es el algoritmo que mejor se ajusta a las necesidades del proyecto.

6.1.1. PLANOS FÍSICOS

Para comenzar en el proceso de modelar la Universidad como un grafo, se requiere conocer todos los destinos posibles que podría tomar una persona como visitante, por tanto, el primer paso a seguir fue la solicitud de los planos arquitectónicos necesarios en el departamento de planta física, dichos planos contienen los cuatro edificios de la Sede Central de la Universidad Santo Tomas, junto con el nombre de cada ubicación y distancias reales, para situar la ruta más corta al destino deseado. En la figura 1

se aprecia el primer piso del edificio Fray Alberto Ariza, todos los demás planos se encuentran agregados como anexos.

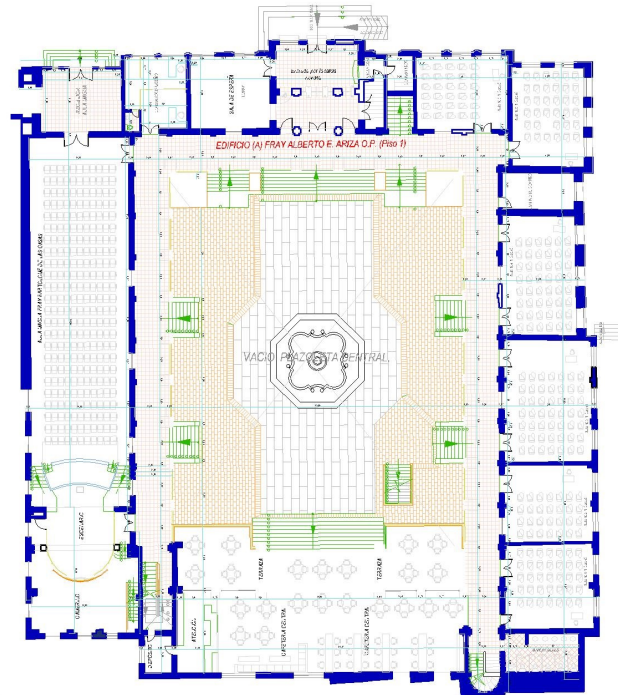


Figura 1. Plano del primer piso del edificio Fray Alberto Ariza.

6.1.2. IDENTIFICACIÓN DE SITIOS DE INTERÉS

Al encontrar cada ubicación en los planos, se llevó a cabo la creación una lista y dentro de esta, se instauró un único código de identificación, el cual pertenece a cada una de las ubicaciones escogidas, dicho nombre único para cada ubicación debía contener en primer lugar el edificio y el piso que lo contenían. El edificio se representa como una única letra, en donde las opciones debido que existen cuatro edificios son, A para el edificio Fray Alberto Ariza, B para el edificio Luis J Torres, C para el edificio Gregorio XIII y F para el edificio Calatayud. Seguido a esa primera letra de identificación del edificio, continua un número que corresponde al piso en el cual se ubica cada destino y para los dos últimos identificadores se define en que categoría entran y un número que hace la función de contador para los destinos que comparten categoría, de este modo no se repetirá el código de identificación en ningún caso.

Para la identificación de las categorías, se hizo uso de diez letras las cuales son, H

conexión entre pisos o edificios, E para las escaleras, L para los laboratorios, A para aulas o salones, W para baños, M para escaleras pequeñas, C para decanaturas, D para los departamentos, S para la salida principal y finalmente X para las ubicaciones que no entran en ninguna de estas categorías. Por ejemplo, en el caso específico del aula 106 del edificio Fray Alberto Ariza ubicada en primer piso, el código de identificación es A1A06.

En el anexo A se adjunta la tabla que relaciona cada punto de interés dentro de la universidad con un código único.

6.1.3. DEFINICIÓN DE ARCOS

Al plantear el grafo y encontrar todos los sitios de interés posibles, se detalló que existían algunas de estas ubicaciones que compartían la misma entrada, y por tanto el mismo vértice. Debido a esto fue necesaria una asignación de código de identificación igual para todos estos puntos ubicados geográficamente en el mismo lugar, como es el caso de las secretarías de división de ingenierías, ubicadas en el cuarto piso del edificio Gregorio XIII.

Para el modelo de la Universidad se decidió que la ruta a recorrer sería a través de los pasillos, debido a que escoger un vértice por cada lugar, conllevaría muchas más conexiones entre vértices haciendo el modelado excesivamente complejo e innecesario, y debido a que todas las ubicaciones son fácilmente visibles desde el pasillo por el que la persona tiene acceso. Se resolvió que la construcción del modelo de la Universidad sería a través de un único camino central. En sí la dinámica consiste en trazar un camino a través de los pasillos de cada piso en el cual una persona puede caminar de manera continua, incluso si existen puntos de interés a la derecha o a la izquierda del camino.

6.2. CARACTERIZACIÓN DE ALGORITMOS

Al investigar sobre el problema de la ruta más corta entre dos vértices de un grafo, se encontró un total de tres algoritmos capaces de resolverlo, el algoritmo de Dijkstra, el algoritmo A* y el algoritmo Ant Colony Optimization. En esta sección se presentan las ventajas y desventajas de uno frente a otro con la finalidad de escoger el que mejor se ajuste a las condiciones del sistema deseado.

Con la finalidad de reducir los tiempos de ejecución del proyecto y estar lo mas cercano posible del cronograma, se decidió no implementar ninguno de los algoritmos propuestos directamente, y en cambio, hacer uso de librerías que resolvieran el problema en cuestión y que fueran de acceso libre para poder ser utilizadas dentro del proyecto.

6.2.1. ALGORITMO A*

El algoritmo A*, conocido también como A estrella, es ampliamente usado en problemas de cálculo de ruta mas corta y es especialmente útil en situaciones en que las rutas se componen de muchos arcos. Dicho algoritmo calcula la ruta mas corta entre dos vértices de un grafo mediante la suma de dos funciones, la primera es una función de costo, la misma usada por el algoritmo de Dijkstra, la segunda es una función heurística, que lo convierte en un algoritmo informado de búsqueda. Los algoritmos informados de búsqueda hacen uso de información adicional para cada el cálculo de cada iteración, en donde se compara la información del estado inicial hasta el nodo actual. Como este algoritmo tiene en cuenta dos factores, los resultados serán acertados dependiendo del criterio de selección de la función heurística, que básicamente es una función que prioriza la velocidad o la certeza del proceso.

Al considerar el algoritmo A* frente al algoritmo de Dijkstra, sobresale la conclusión de que el algoritmo de Dijkstra es una versión simplificada del algoritmo A* en la que la función heurística siempre es nula, por lo que el cálculo depende únicamente de la función de costo real. Considerando que el objetivo es obtener la ruta correcta que represente el camino mas corto en el menor tiempo posible, la heurística no es necesaria y el algoritmo de Dijkstra es la mejor opción.

6.2.2. ALGORITMO ANT COLONY OPTIMIZATION

El algoritmo Ant Colony Optimization fue probado haciendo usó de la herramienta MATLAB, allí se pudo apreciar que este es un algoritmo mas robusto que los demás, que puede ser adaptado a varios problemas matemáticos, por lo que naturalmente consume mayores recursos y no es el adecuado si se busca una aplicación ligera. Este algoritmo permite entre otras cosas, calcular la distancia más corta entre dos vértices que pueden

moverse a lo largo del tiempo, cambiando su ubicación en el espacio definido por el grafo y en consecuencia modificando los valores de los arcos.

Para el caso del grafo que representa la Universidad, los arcos son estáticos y no cambiarían en el tiempo bajo ninguna circunstancia, por lo que las ventajas del algoritmo Ant Colony no son importantes para el desarrollo del proyecto. Por otro lado, al comparar el algoritmo de Dijkstra frente al algoritmo Ant Colony, se observó que el primero es más rápido, demanda menos recursos computacionales y es muy acertado, características primordiales a la hora de elegir un algoritmo a implementar en un sistema embebido de recursos limitados.

6.2.3. LIBRERÍA IGRAPH

Existen en la red variadas librerías que resuelven los problemas relacionados con la teoría de grafos, sin embargo, para este caso en particular era necesario que la librería fuese rápida y eficiente en el cálculo de la ruta, esto quiere decir que se buscó una librería que no necesitara de muchos recursos de cómputo, y que no estuviera limitada a un único lenguaje de programación. La librería que mejor se ajustó a estos requerimientos es la librería denominada igraph, que más que una librería es un conjunto de herramientas para el análisis de grafos, que además es de código abierto y gratuita. Una de las principales ventajas de igraph frente a otras herramientas similares es la portabilidad, igraph está disponible para los lenguajes de programación R, Python, C y C++, por lo que la plataforma para programar nunca fue una limitación con esta librería. Por último, igraph ofrece una gran ventaja en la capacidad de seleccionar el algoritmo deseado para realizar los cálculos de la ruta más corta, dentro de los algoritmos posibles se encuentra el algoritmo de Dijkstra.

Durante la ejecución de este proyecto, se hizo uso de igraph en sus formatos de R y C++, inicialmente con R se modeló la Universidad como un grafo no dirigido en el cual se ingresaron los vértices que fueron considerados pertinentes y los arcos que los unen, esta información fue suficiente para empezar a elegir las funciones de igraph que serían utilizadas en la aplicación final. Igraph es una vasta librería que contiene funciones para virtualmente cualquier problema relacionado con grafos, sin embargo, en este proyecto las necesidades no son tan grandes, por lo que igraph dentro de este contexto es una librería que permite:

- **Crear un grafo:** A pesar de que igraph ofrece varias formas y figuras para definir un grafo, se eligió crear un grafo vacío para que todos los nodos tuvieran la misma importancia.
- **Agregar vértices a un grafo existente:** Es posible agregar todos los vértices a la vez a un grafo vacío, o agregarlos a medida que sea necesario a un grafo que ya tiene algunos vértices. Estos vértices pueden tener atributos adicionales como nombre y forma. Se agregó un total de 225 vértices con atributo de nombre a un grafo vacío.
- **Agregar arcos o uniones entre vértices:** Haciendo uso de la información cartográfica contenida en los planos suministrados por planta física, se agregó un total de 258 arcos con un número real que representa una relación de 1:1 con la distancia física entre dos vértices medida en metros.
- **Encontrar la matriz de adyacencia para un grafo:** Contiene la información de la distancia mínima acumulada entre un vértice cualquiera y otro. Es una matriz cuadrada de dimensiones n^2 , donde n es el número de vértices, y cada posición i, j corresponde a la distancia más corta posible desde el vértice i al vértice j . En caso de que una distancia no sea posible, porque no hay forma de llegar de un vértice a otro, dicha posición de la matriz es ocupada por el infinito.
- **Calcular el camino mas corto entre dos vértices cualquiera:** Permite calcular la ruta óptima. Esta función retorna, según los parámetros enviados, un camino de vértices por donde pasa la ruta, o un conjunto de arcos que deben ser recorridos por la ruta. Igraph permite elegir un algoritmo para este cálculo, se prefirió el algoritmo de Dijkstra por su manejo previo en el proyecto.
- **Guardar un grafo como un archivo de texto:** Igraph maneja varios formatos, de acuerdo a la información del grafo que se quiere guardar o a la aplicación de destino del archivo. Se prefirió el formato Pajek, que guarda tanto vértices como arcos identificados con números enteros consecutivos desde 1 hasta la cantidad total, adicionalmente permite guardar las distancias de cada arco como números reales.

6.2.4. RESULTADOS

El criterio de selección del algoritmo de Dijkstra obedeció inicialmente a la comodidad de uso del mismo, sin embargo, al llevar a la práctica los algoritmos de cálculo de ruta más corta propuestos, se encontraron varias limitantes para el uso de estos, el grafo en cuestión no es de un orden tan grande, tampoco varía en el tiempo, y el interés principal es calcular la ruta más corta de un punto a otro en el menor tiempo posible, buscando que este resultado siempre sea acertado. De manera que el algoritmo que mejor se ajusta a las necesidades del proyecto es el algoritmo de Dijkstra.

Se implementó un algoritmo que calcula una matriz con el tiempo de cómputo de la distancia más corta entre todas las parejas de vértices posibles, para mejorar los resultados se obtuvo el promedio de las matrices resultantes de ejecutar el algoritmo durante cincuenta iteraciones. Adicionalmente para estas cincuenta matrices se calculó la desviación estándar para tener un parámetro de medición de que tan distantes son los valores. Sobre los datos de las cincuenta iteraciones, también se calculó el promedio para cada columna, que representa el valor medio del tiempo de cómputo del camino más corto desde un vértice hasta todos los demás vértices, de igual forma se obtuvo el máximo valor por columna, correspondiente al mayor tiempo de cómputo de un vértice a otro.

Con la información de los tiempos de cómputo promedio resultante de cincuenta iteraciones se obtuvo un promedio total que representa el tiempo medio que se demora la aplicación en obtener la ruta más corta entre un par de vértices. Del mismo modo, se obtuvo el valor máximo y mínimo de las cincuenta iteraciones, correspondientes al mayor y menor tiempo que le toma a la aplicación obtener la ruta más corta entre dos vértices.

Es pertinente calcular los tiempos de cómputo de la ruta más corta manteniendo como punto de partida el vértice correspondiente a la entrada principal de la Universidad, que será donde se ubicará el robot, esto para tener un registro de los tiempos de las trayectorias reales a utilizar en el proyecto. Como en los casos anteriores, es importante encontrar el tiempo promedio, máximo y mínimo del cálculo de una ruta.

6.3. DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE ASISTENTE DE RUTA

La etapa de desarrollo de software se centró en la experiencia de usuario, el público objetivo de este proyecto son todas las personas que entran a la Universidad, de manera que se buscó facilitar el uso de la aplicación al máximo, con ayudas visuales y controles intuitivos que permitan al usuario entender como debe elegir su destino, calcular su ruta y movilizarse a través de los edificios.

En esta sección se muestran los criterios de selección de todas las herramientas e implementos utilizados, teniendo en cuenta que el factor de mayor peso para el diseño de la solución es la facilidad de uso por parte del usuario final.

6.3.1. PLANEAMIENTO DE REQUERIMIENTOS FUNCIONALES

Partiendo del objetivo de ofrecer una experiencia de uso intuitiva y accesible, se planteó una serie de requerimientos funcionales que la aplicación debía cumplir, con el fin de abarcar todos los aspectos que se consideraron importantes para que el usuario tuviera una experiencia óptima.

Con estos requerimientos funcionales en cuenta se consideró que la aplicación debía:

- Mostrar como pantalla de inicio La cara del robot.
- Reconocer la intención del usuario de interactuar con el robot.
- Agrupar distintas ubicaciones, las opciones son: Salones, Auditorios, Facultades, Representaciones, Laboratorios, Oficinas, Sala de profesores, Sitios de interés.
- Poder mostrar que está procesando.
- Permitir que el usuario navegue de forma fácil.
- Mostrar una animación por pisos y lista de pasos para el recorrido.
- Permitir al usuario la visualización del número total de mapas y en cual se encuentra.

- Mostrar un mensaje, animación u opción de volver.
- El sistema no requiere batería.
- El usuario no puede cerrar la aplicación.

6.3.2. SELECCIÓN DE ELEMENTOS DE HARDWARE

Para la implementación en hardware del proyecto se buscaron dispositivos que cumplieran con las funciones necesarias al menor costo posible, también se buscó adquirir dispositivos con prestaciones adicionales con el fin de expandir o adaptar el proyecto de acuerdo con el curso del mismo. Los dispositivos seleccionados para desarrollar el proyecto fueron:

- **Sistema embebido:** Raspberry Pi es un dispositivo con un procesador potente en relación con su compacto tamaño, la arquitectura ARM de este procesador no representaba un impedimento ya que tanto C++ como Qt son compatibles con esta arquitectura, este sistema embebido cuenta también con memoria RAM más que suficiente para la ejecución de la aplicación. Al elegir este dispositivo se cuenta también con prestaciones adicionales como puertos de propósito general, salida de audio análoga, puertos USB, conexión bluetooth y la capacidad de conectarse a internet de forma cableada vía ethernet o inalámbricamente mediante wifi. La tabla 1 muestra las características más importantes del sistema embebido Raspberry Pi 3.

Procesador	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
Memoria RAM	1GB
Conectividad	LAN, WiFi, Bluetooth
Puertos	4 USB 2.0, 40-pin GPIO, HDMI, MicroSD, MicroUSB

Tabla 1. Características técnicas Raspberry Pi 3.

- **Interfaz con el usuario:** Al contar con una aplicación con la necesidad de presentar y requerir un gran volumen de información al usuario constantemente, se debe hacer uso de un dispositivo capaz de presentar estos datos de la manera mas entendible posible, para que el usuario entienda que debe acciones debe realizar

para lograr su objetivo. Una pantalla con entrada táctil es un dispositivo que además de presentar la información de manera visualmente agradable, permite también la lectura de instrucciones por parte del usuario que solo deberá tocar la pantalla en el área correspondiente a la opción que desee, logrando así que la experiencia de uso sea más entendible para el usuario común.

Al definir que una pantalla táctil es la solución mas integral para comunicarse con el usuario, ya que en un solo dispositivo incluye una interfaz de entrada y salida, necesarias para la ejecución de la aplicación, se procede a elegir una pantalla acorde al proyecto. Teniendo en cuenta el alcance de este proyecto, el tamaño de la pantalla debe ser el adecuado para que el usuario vea cómodamente todo el contenido necesario, y además lo suficientemente espaciosa para que el usuario pueda tocar la opción de su preferencia con certeza.

A continuación, se muestran en la tabla 2 las características técnicas de la pantalla seleccionada, que sin exceder el presupuesto previsto logra cumplir con lo propuesto.

Tamaño	10.1 pulgadas
Resolución	1280x800
Tecnología de control táctil	Capacitivo de 10 puntos de tacto
Salida de video	HDMI
Salida de control táctil	USB

Tabla 2. Características técnicas de la pantalla táctil seleccionada.

6.3.3. SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN

La librería igraph está implementada en tres lenguajes de programación diferentes, R, C++ y Python, y cada una de estas versiones cuenta con las mismas capacidades y funciones.

La prioridad en la selección del lenguaje de programación fue la velocidad de cómputo y la efectividad en el consumo de recursos para realizar las tareas propuestas. R es un lenguaje fácil de usar y tiene ciertas ventajas como la posibilidad de consultar el valor de una variable en tiempo real sin tener que ejecutar scripts adicionales, Python es un lenguaje de fácil lectura que permite integrar varios sistemas de manera efectiva, por

último, C++ es un lenguaje muy robusto que permite realizar cualquiera de las tareas necesarias para este proyecto.

Se eligió el lenguaje de programación C++ ya que es el que cumple de mejor manera los criterios expuestos, para el caso específico de igrph, la misma tarea realizada en C++, R y Python siempre es más rápida en el lenguaje C++, y, el lenguaje C++ permite moderar el consumo de memoria al hacer uso de apuntadores a variables, que no reservan espacio adicional innecesario en el sistema donde se ejecute la aplicación.

6.3.4. SELECCIÓN DEL ENTORNO DE DESARROLLO

Para la aplicación de cálculo de ruta se planteó una interfaz gráfica de usuario que permitiera visualizar de manera clara todos los contenidos necesarios para elegir un destino, y posteriormente recibir las indicaciones para llegar a este. De tal forma que el entorno de desarrollo integrado (IDE) que se seleccionó necesitaba tener la capacidad de mostrar una interfaz gráfica fluida y agradable a la vista. Se eligió el entorno de desarrollo Qt Creator, basado en el framework multiplataforma Qt, el cual puede ser programado en lenguaje C++ bajo el paradigma de la programación orientada objetos. Al ser una herramienta multiplataforma, Qt permite el desarrollo para varios dispositivos, incluyendo computadores con sistema operativo Linux, y el sistema embebido Raspberry Pi. Se eligió Qt sobre otras opciones principalmente por su capacidad multiplataforma, también por la opción de obtener una licencia no comercial de forma libre y gratuita, y adicionalmente por que maneja gráficos de gran calidad. A lo largo del desarrollo del proyecto, surgieron varias necesidades que fueron resueltas de manera sencilla en Qt, ya que es un entorno muy completo con variadas librerías para el uso de archivos de texto e imágenes entre otros.

6.3.5. DISEÑO UML

Para el diseño del software se aplicó la técnica de programación orientada a objetos, se consideraron un total de 8 clases, incluyendo las clases que corresponden a los formularios de la interfaz gráfica. La clase principal es la clase nodo, que representa una ubicación dentro de la Universidad, a esta clase le corresponde un nombre, un número identificador, el piso en que se encuentra la ubicación, y unas coordenadas en el plano

correspondiente a este piso. La clase universidad contiene toda la información sobre ubicaciones dentro del grafo, incluyendo vectores de nodos y arcos, los cuales son leídos de archivos para que la aplicación sea fácil de adaptar a los cambios.

La librería igraph se importó directamente en una clase llama grafo, que hace uso de los métodos disponibles y los adapta a las necesidades del proyecto, como los tipos de datos disponibles en Qt. La clase lista es la clase que enlaza la parte grafica con el algoritmo de ruta, por lo que esta tiene conexión con las clases menú, edificio, piso y ruta, que son las clases que representan las ventanas de la interfaz gráfica del programa. Pero la clase lista además hace uso de las clases grafo, nodo y universidad que le permiten tener acceso a la información de las localizaciones.

Cada clase correspondiente a una ventana tiene métodos para llamar la ventana siguiente o anterior, y métodos adicionales que le permiten mandarle información a una clase hijo o padre. De esta forma se logran comunicar las clases para que las acciones en una ventana previa tengan repercusión en la siguiente.

En la figura 2 se muestra el diagrama de clases completo que representa la totalidad de los métodos y atributos utilizados por la aplicación.

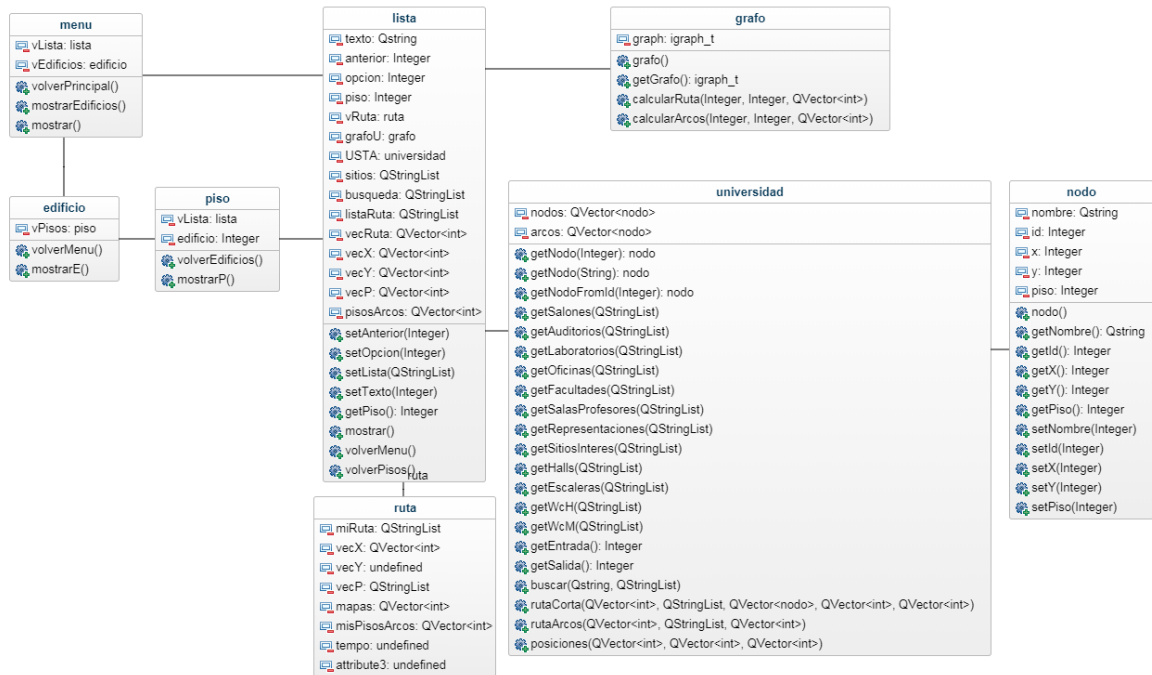


Figura 2. Menú de visualización con la ruta a seguir.

6.4. VALIDACIÓN DEL SISTEMA

Para validar la velocidad y certeza del algoritmo escogido, se implementó un algoritmo que calcula una matriz con el tiempo de cómputo de la distancia más corta entre todas las parejas de vértices posibles, para mejorar los resultados se obtuvo el promedio de las matrices resultantes de ejecutar el algoritmo durante cincuenta iteraciones. Adicionalmente para estas cincuenta matrices se calculó la desviación estándar para tener un parámetro de medición de que tan distantes son los valores. Sobre los datos de las cincuenta iteraciones, también se calculó el promedio para cada columna, que representa el valor medio del tiempo de cómputo del camino más corto desde un vértice hasta todos los demás vértices, de igual forma se obtuvo el máximo valor por columna, correspondiente al mayor tiempo de cómputo de un vértice a otro. Con la información de los tiempos de cómputo promedio resultante de cincuenta iteraciones se obtuvo un promedio total que representa el tiempo medio que se demora la aplicación en obtener la ruta más corta entre un par de vértices. Del mismo modo, se obtuvo el valor máximo y mínimo de las cincuenta iteraciones, correspondientes al mayor y menor tiempo que le toma a la aplicación obtener la ruta más corta entre dos vértices. Es pertinente calcular los tiempos de cómputo de la ruta más corta manteniendo como punto de partida el vértice correspondiente a la entrada principal de la Universidad, que será donde se ubicará el robot, esto para tener un registro de los tiempos de las trayectorias reales a utilizar en el proyecto. Como en los casos anteriores, es importante encontrar el tiempo promedio, máximo y mínimo del cálculo de una ruta.

6.4.1. DISEÑO DE LA ENCUESTA

Tras la finalización del robot, prosigue la realización de una investigación sobre la aceptación que este tendrá dentro de la Universidad, el medio escogido para esta investigación fue la encuesta, como búsqueda sistemática de información, esta información recolectada hace referencia a lo que las personas opinan sobre el robot, en particular la encuesta permite conocer la aceptación y valoración del robot en el entorno para el que fue elaborado, para este caso es la sede principal de la Universidad Santo Tomas. Las preguntas se enfocan hacia un marco de satisfacción de modo que las diferencias localizadas son atribuidas a las personas entrevistadas, dejando la mayoría de estas en forma cerrada para que la encuesta sea más contundente.

Con la encuesta se busca validar la implementación de este proyecto, responder la inquietud de si la Universidad realmente necesitaba un asistente de ruta. Para esto se tuvieron en cuenta dos temáticas centrales, el diseño de la aplicación y la efectividad del algoritmo de cálculo de ruta óptima. Dentro de la temática de diseño, el aspecto principal a evaluar fue la facilidad de interacción con la aplicación, por otro lado en el tema de algoritmo de ruta se busco validar que el algoritmo implementado fuera rápido y diera resultados acertados.

En el anexo B se muestra la encuesta a realizar en la muestra.

6.4.2. SELECCIÓN DE LA MUESTRA

En la selección de la muestra es necesario escoger a personas que ya tengan un contacto previo con la Universidad debido a que el objetivo en concreto del proyecto es ayudar a las personas a encontrar el destino al cual desean llegar por el camino más corto, por tanto, la comparación entre diferentes caminos es una tarea que pueden realizar solo personas que ya conozcan la Universidad y por tanto puedan identificar varios caminos, para llegar al mismo destino.

Teniendo en cuenta que la población de la universidad cumple en su gran mayoría con el rol de estudiante, es adecuado definir este grupo de personas como el objetivo de este análisis. Los estudiantes en general tienen destinos variados dentro de los edificios que componen la sede principal de la Universidad, y sus destinos son asignados en muchos casos sin que ellos conozcan la ubicación a la que se dirigen.

7. RESULTADOS

En esta sección se muestran los resultados obtenidos con el proyecto, incluyendo la aplicación final con varias imágenes que muestran el diseño que se utilizó. También se muestran los datos recopilados pertinentes para evaluar el cumplimiento de los criterios principales de la aplicación, la velocidad de cálculo, la precisión del resultado y la presentación.

7.1. ESTADÍSTICAS PARA LOS TIEMPOS DE CÓMPUTO

Se obtuvieron dos tablas de 225 filas por 225 columnas, en donde las filas representan el vértice de destino, las columnas el vértice de origen, y la celda interceptada por una fila y columna representa el promedio o desviación estándar del tiempo que le toma al algoritmo calcular la ruta más corta entre origen y destino. La primera tabla corresponde a la información del promedio del tiempo de cálculo de la ruta más corta entre un vértice y otro, en la figura se observa una parte de la tabla resultante, en la que se ilustra que los tiempos obtenidos son del orden de los milisegundos y no interfieren con la experiencia del usuario.

Tiempo(s)	A0H01	A0X01	A0E01	A0X03	A0E02	A0M01	A0X02
A0H01	0,00087	0,00089	0,00089	0,00099	0,00099	0,00087	0,00094
A0X01	0,00081	0,00084	0,00160	0,00093	0,00637	0,00088	0,00088
A0E01	0,00091	0,00086	0,00297	0,00084	0,00096	0,00091	0,01139
A0X03	0,00093	0,00090	0,00091	0,01765	0,00092	0,00085	0,00083
A0E02	0,00082	0,00088	0,00085	0,00090	0,00088	0,00096	0,00098
A0M01	0,00087	0,00091	0,00094	0,00094	0,00093	0,00088	0,00092
A0X02	0,00087	0,00085	0,00087	0,00091	0,00082	0,00094	0,00088

Tabla 3. Promedio del tiempo de cómputo entre un vértice y otro.

En el anexo "Tabla de promedios.xlsx" se encuentra la tabla con la información completa sobre el promedio en los tiempos de cómputo.

Respecto a la certeza del algoritmo, se observa que la matriz resultante es aproximadamente simétrica, de forma que no importa en que tiempo se ejecute la tarea del cálculo

de ruta, siempre el resultado será similar. Otra consideración tenida en cuenta es qué para el cálculo de los promedios se dejó en ejecución ininterrumpida un equipo que únicamente cumplía con esta tarea, por lo que también es cierto que el punto en que convergen los tiempos de computo es aquel en que el procesador se centra en esta única tarea.

La información sobre la desviación estándar de la misma variable de tiempo de cómputo de la ruta mas corta entre dos vértices, esta contenida en una segunda tabla representada en la figura

Tiempo(s)	A0H01	A0X01	A0E01	A0X03	A0E02	A0M01	A0X02
A0H01	0,00036	0,00039	0,00031	0,00021	0,00036	0,00043	0,00035
A0X01	0,00039	0,00041	0,00484	0,00025	0,03887	0,00031	0,00037
A0E01	0,00028	0,00034	0,01419	0,00037	0,00020	0,00028	0,07418
A0X03	0,00025	0,00030	0,00028	0,11798	0,00037	0,00041	0,00043
A0E02	0,00047	0,00044	0,00041	0,00035	0,00043	0,00028	0,00032
A0M01	0,00039	0,00026	0,00031	0,00031	0,00025	0,00033	0,00040
A0X02	0,00038	0,00034	0,00039	0,00035	0,00038	0,00022	0,00031

Tabla 4. Desviación estándar del tiempo de cómputo entre un vértice y otro.

En el anexo "Tabla de desviacion.xlsx" se encuentra la tabla con la información completa sobre la desviación estándar en los tiempos de cómputo.

Como métricas adicionales se tuvieron en cuenta el valor máximo y el valor promedio de tiempo cuando el vértice de origen es la entrada de la Universidad, lugar donde se plantea ubicar el robot inicialmente, los valores obtenidos fueron:

$$max(t) = 0,037402902s$$

$$avg(t) = 0,002099974s$$

Nuevamente se trata de tiempos que no causaran afectación durante la etapa de ejecución de la aplicación.

7.2. SOFTWARE ASISTENTE DE RUTA

Para la pantalla de bienvenida se requirió de una cara amigable para el usuario, ya que es la primera imagen que este se lleva de la aplicación, se pensó en una cara que ilustrará el factor de que se está interactuando con un robot, por lo que además de resemlar una figura robótica, también es una imagen animada que inspira agrado. La aplicación muestra la animación indefinidamente hasta que el usuario manifieste su intención de interactuar con esta, la animación terminará una vez el usuario toque la pantalla para decirle al programa que desea iniciar. En la figura 3 se muestra la pantalla de bienvenida.

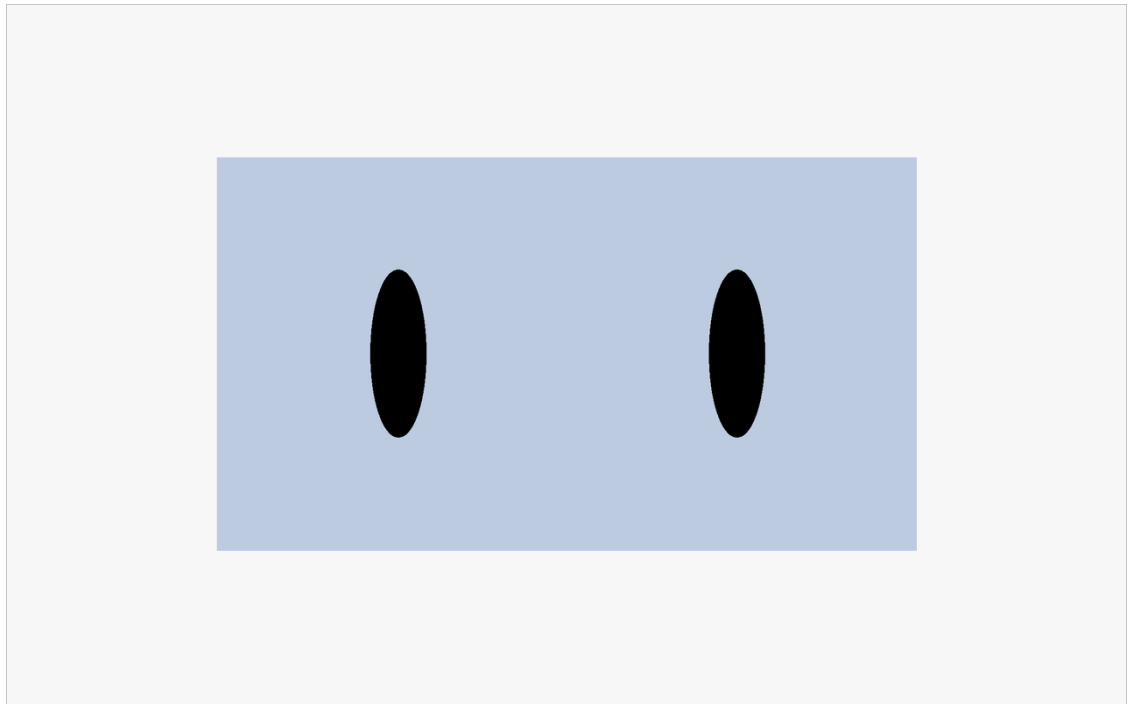


Figura 3. Animación de bienvenida al asistente de ruta.

Una vez el usuario toque la pantalla para iniciar, se despliega el menú principal, el cual consta de dos secciones: en la parte superior de la ventana se encuentra un cuadro de búsqueda en el cual el usuario puede ingresar el nombre de su destino o parte de este nombre, en la parte inferior de este cuadro de búsqueda, el usuario tiene la opción de elegir entre varios submenús, los cuales representan las categorías principales de ubicaciones dentro de la Universidad. De acuerdo a la elección que haga el usuario, se desplegara un menú nuevo en el que pueda continuar con la tarea de selección de su

destino. Adicionalmente se cuenta con un botón para volver a la pantalla de inicio y unos marcos representativos de la imagen de la Universidad. Dentro de estos marcos se ubican, el logo de la Universidad Santo Tomás y un mensaje de bienvenida para el usuario. En la figura 4 se muestra el resultado final del menú principal.



Figura 4. Menú principal de la aplicación.

En el caso de que el usuario elija una de las categorías presentadas en el menú principal, la aplicación pasara a un submenú en el que se listan todos los destinos que coinciden con esa categoría. En la figura 5 se muestra el ejemplo de la pantalla a la que se avanza tras elegir la opción Representaciones desde el menú principal, se observa en la figura un listado con todas las representaciones académicas. En este punto el usuario tiene la opción de seleccionar un destino dentro de la lista y posteriormente usar el botón calcular ruta para continuar con las indicaciones, o, pulsar el botón volver para regresar al menú principal. La aplicación no permite que el usuario elija la opción calcular ruta si no hay ningún destino de la lista seleccionado.

En el caso de que el usuario elija la opción salones del menú principal, dado que en su mayoría la Universidad esta compuesta de estos, esta opción se divide en un botón por cada piso en el que haya salones. De forma que el submenú que se genera al ir a la opción salones, representado en la figura 6, tiene 3 botones, uno para el edificio Fray



Figura 5. Opción de destino Representaciones.

Alberto Ariza, otro para el edificio J. Torres y otro para el edificio Gregorio XIII. Como en todas las ventanas hasta ahora, se dispone también de una opción para volver al menú anterior.

De acuerdo a la elección realizada en el submenú de edificios, se mostrará seguidamente una ventana en la que el usuario puede elegir el piso en el que se encuentra el aula a la que quiere dirigirse. De esta forma no se muestran todos los salones de un edificio, lo que generaría confusión y dificultades de uso. En la figura 7 se muestra el menú de pisos, que es dinámico y únicamente muestra los pisos de un edificio en los que existen salones, en caso de no tener salones el piso se omitirá por completo.

Cuando la aplicación reciba la información por parte del usuario de un destino específico, ya sea mediante la herramienta de búsqueda o navegando por las categorías disponibles, se ofrecerá la opción mediante un botón de calcular la ruta como se ve en la figura 3. Inmediatamente el usuario registre la pulsación en esta opción de calcular ruta, se mostrará en pantalla una vista cartográfica de el piso en que se encuentra ubicado, también se le mostrara una animación de como debe moverse a través de este piso.

En la figura 8 se muestra la pantalla de la ruta calculada, que además de contener

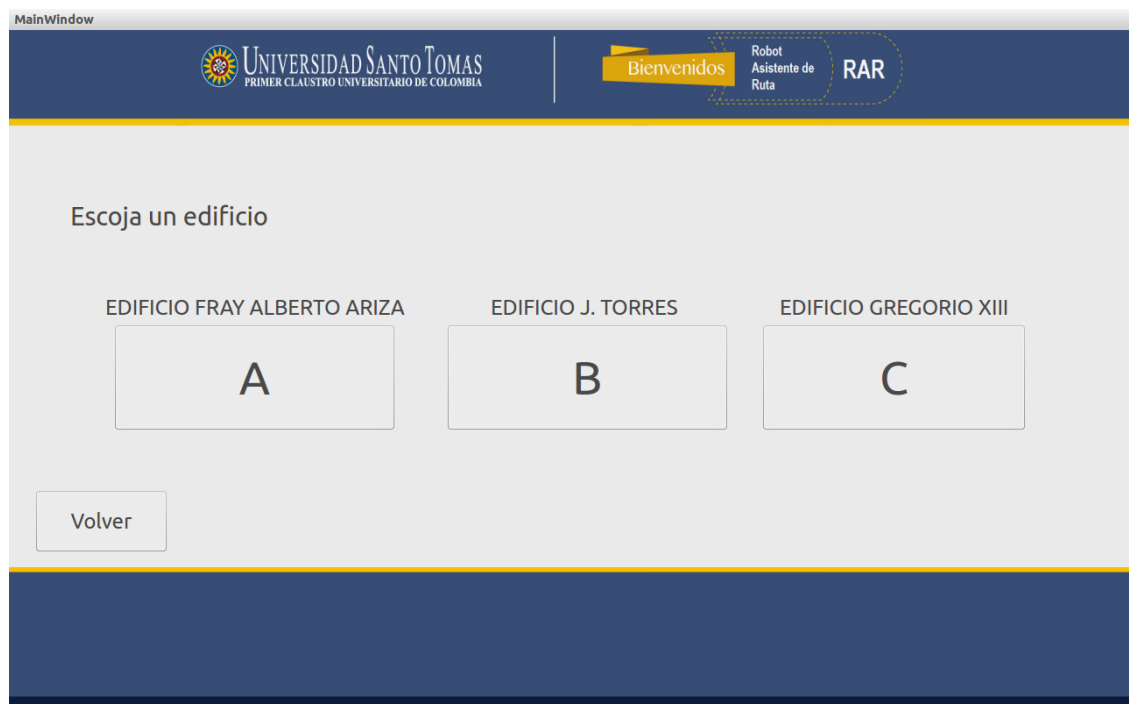


Figura 6. Submenú de edificios para la opción salones.

un plano y los elementos animados, muestra en la parte inferior la cantidad total de pisos que debe recorrer el usuario para llegar a su destino, junto con un indicador de en que piso va la animación. El usuario también puede elegir entre varias opciones dentro de esta ventana, incluyendo moverse hacia adelante y hacia atrás sobre el numero de pisos contenidos en la ruta, reiniciar la animación en cualquier momento, y la opción de volver al menú anterior. En la parte izquierda de la pantalla, el usuario puede leer una lista de instrucciones a seguir para llegar a su destino, esta lista se refiere únicamente al piso que se esta visualizando.

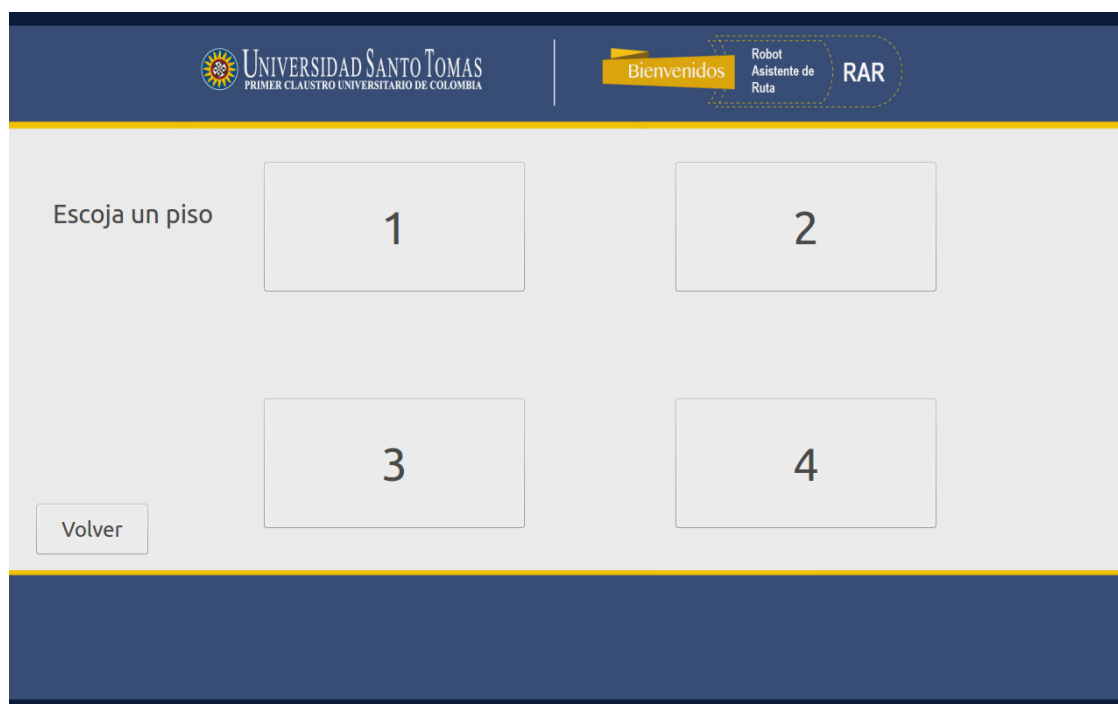


Figura 7. Submenu de pisos para la opción edificios.



Figura 8. Menú de visualización con la ruta a seguir.

7.3. RESULTADOS DE LA ENCUESTA

Partiendo de la población de la Universidad, la muestra se realizó buscando estudiantes que estuvieran familiarizados con las instalaciones de la sede principal. Los estudiantes encuestados tuvieron la posibilidad de calcular múltiples rutas a destinos conocidos y desconocidos, e interactuar con la totalidad de las funciones de la aplicación durante el tiempo que quisieran.

La tabla 5 muestra la relación del nombre de cada persona encuestada y su edad.

Nombre	Edad
Carol Tatiana Botia Barragán	20
Erika Valentina Moya Porras	20
Juan Pablo Romero	21
María Alejandra Barrios Villalobos	19
David Paez Ramirez	19
Daniela Bautista Diaz	21
Sergio Andrés Lagos Fuquen	19
Mateo Parra Posse	20
Andres Geovanny Lopez Faustino	23
Miguel Alberto Esteban López	18

Tabla 5. Descripción de personas encuestadas.

El grupo encuestado resaltó la justificación expuesta para el proyecto, ya que como estudiantes todos tuvieron en común la dificultad para movilizarse dentro de la Universidad sin una guía cuando iniciaron sus estudios. Adicionalmente, a pesar de que algunos de los estudiantes encuestados ya llevan varios años en la Universidad, estos comentaron que aún tienen problemas para encontrar ciertos destinos.

A continuación, se describen los resultados para cada una de las secciones en que está dividida la encuesta. Cada pregunta va acompañada de un gráfico que resalta la tendencia en las respuestas.

Valore los siguientes aspectos de software de acuerdo con su experiencia con la aplicación

- Útil: La información dada por la aplicación le permitió ubicar el destino solicitado.

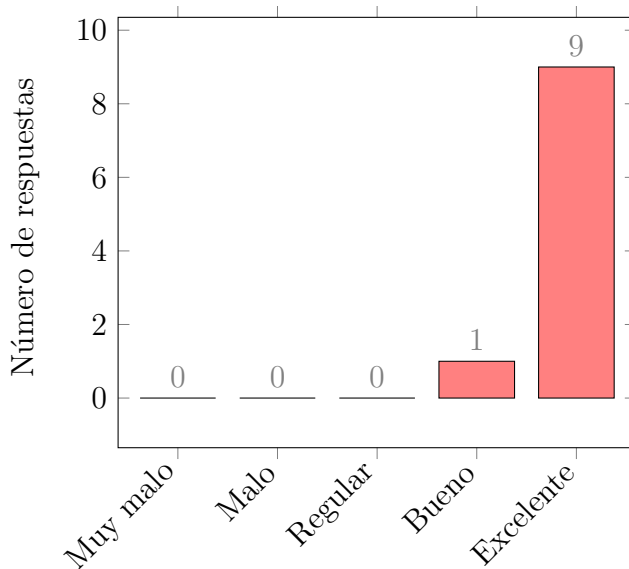


Figura 9. Resultados de la encuesta para el criterio de utilidad.

- Intuitivo: Es fácil navegar a través de las distintas opciones de la aplicación.

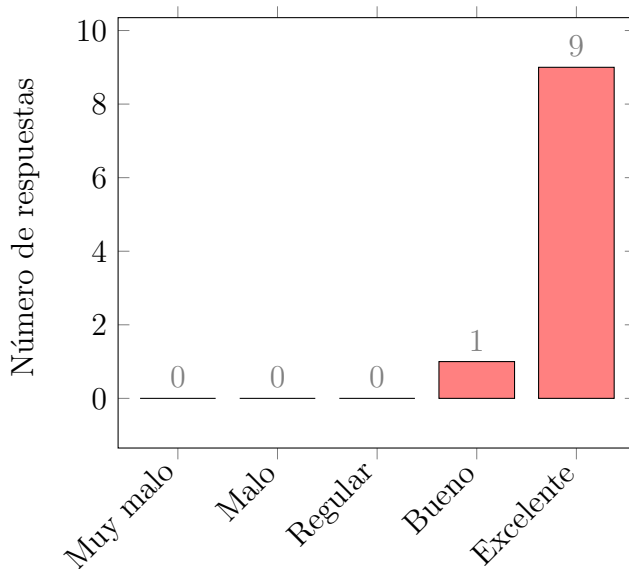


Figura 10. Resultados de la encuesta para el criterio de intuición de uso.

- Agradable: El entorno gráfico de la aplicación es visualmente atractivo.

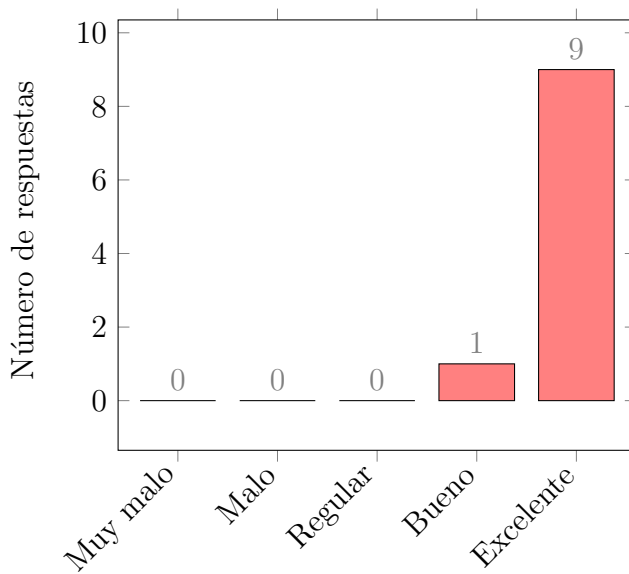


Figura 11. Resultados de la encuesta para el criterio de agrado.

- Interactivo: La forma de interacción con la aplicación (pantalla táctil) le resulta apropiada.

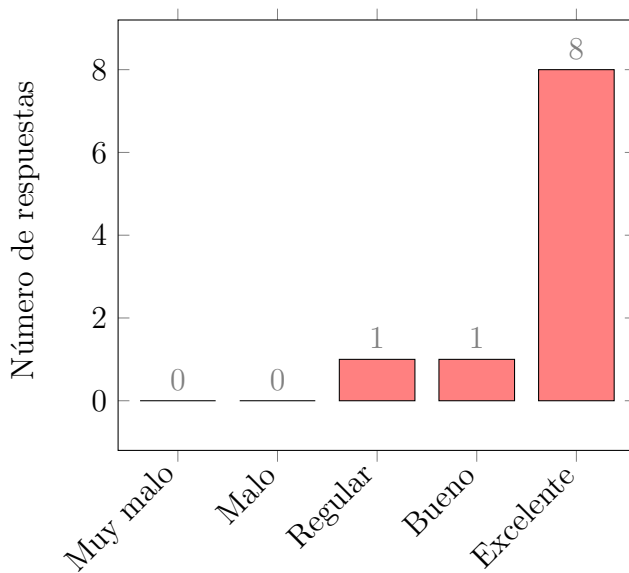


Figura 12. Resultados de la encuesta para el criterio de interacción.

Hay una alta aceptación por parte de los usuarios del software implementado, la solución propuesta buscaba naturalidad en su uso, por lo que las instrucciones de uso se limitaron a responder algunas inquietudes muy puntuales sobre la implementación del proyecto, mas no sobre como navegar en la aplicación. El ítem de agrado tuvo inconvenientes relacionados con la pantalla utilizada para la prueba, ya que no tenía la precisión suficiente y dificultaba en ciertas instancias la navegabilidad de la aplicación, sin embargo, conociendo esto la aplicación se diseño para tener controles y botones grandes, que permitieron una experiencia optima en el 80 % de la muestra.

Valore los siguientes aspectos del cálculo de la ruta de acuerdo con su experiencia con la aplicación

- ¿Qué tan adecuado le parece el tiempo del cálculo de la ruta a su destino?

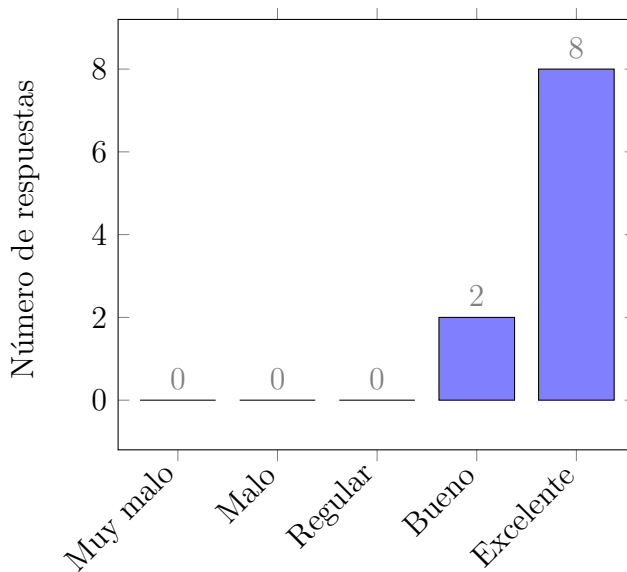


Figura 13. Resultados de la encuesta para el criterio de tiempo de cálculo.

- De acuerdo con sus conocimientos de la universidad, ¿qué tan apropiada le parece la ruta resultante?

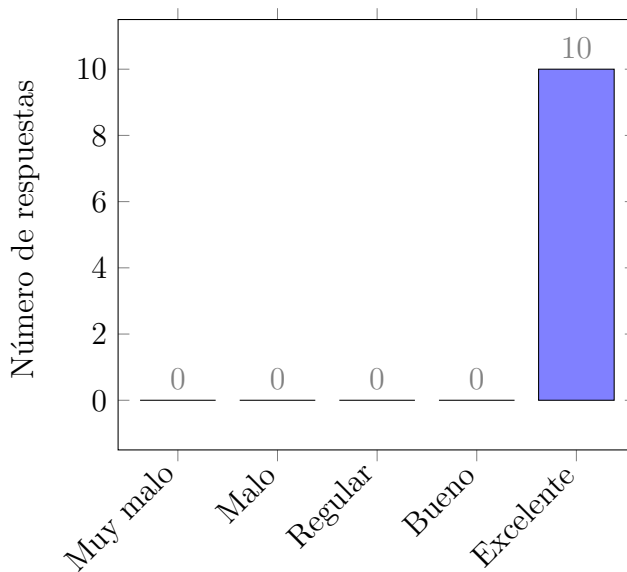


Figura 14. Resultados de la encuesta para el criterio de ruta óptima.

Los resultados de la sección que evalúa la certeza del algoritmo de cálculo de ruta evidencian el trabajo de modelado e implementación detrás de la interfaz gráfica. Al basarse en un algoritmo tan confiable como el de Dijkstra y usar una librería tan robusta como `igraph`, las rutas calculadas siempre son óptimas. Las personas encuestadas mostraron agrado frente a consideraciones adicionales como no tener en cuenta la plaza principal en las rutas planeadas, ya que es un espacio que se encuentra constantemente intransitable a causa de eventos programados en la Universidad.

Entre 0% y 100%, ¿Qué tan necesario le parece para la Universidad la aplicación asistente de ruta?

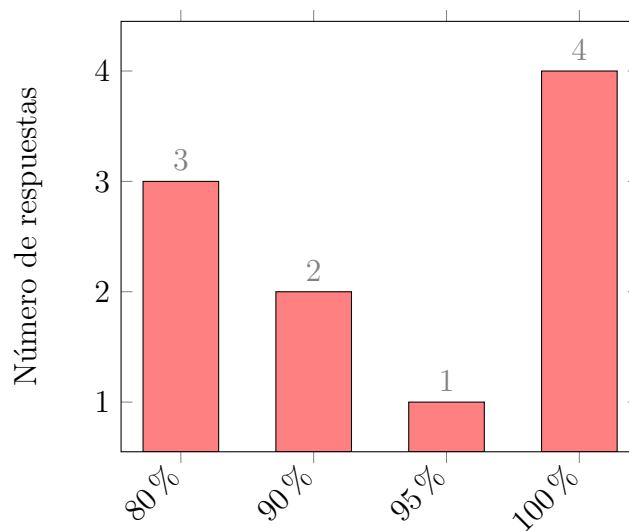


Figura 15. Resultados de la encuesta a la necesidad de la aplicación.

¿Qué le mejoraría a la aplicación?

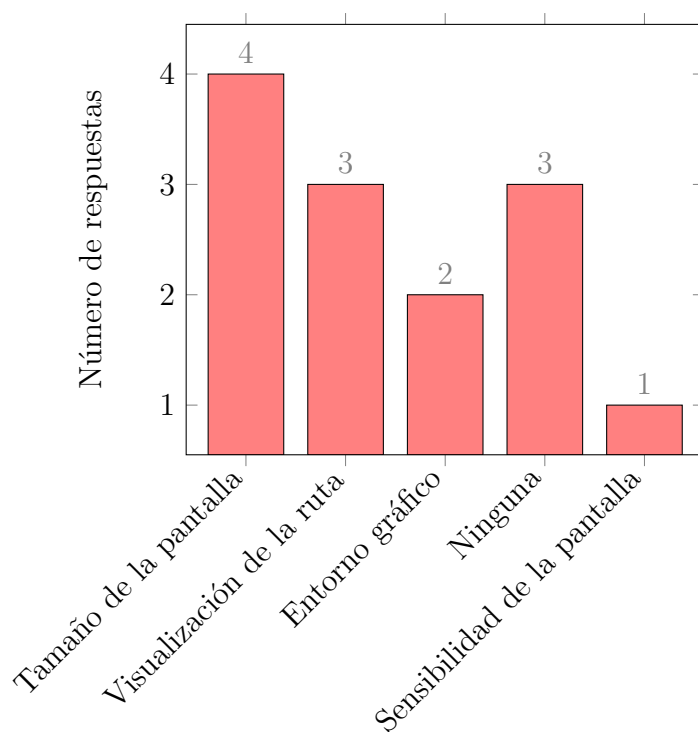


Figura 16. Resultados de la encuesta sobre mejoras a implementar.

¿Fue satisfactoria su experiencia con la aplicación?

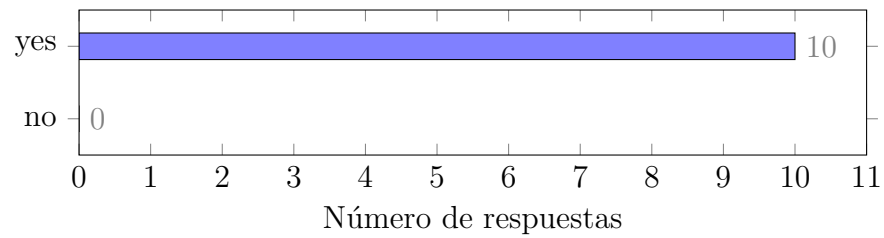


Figura 17. Resultados de la encuesta sobre satisfacción.

¿Recomendaría la aplicación?

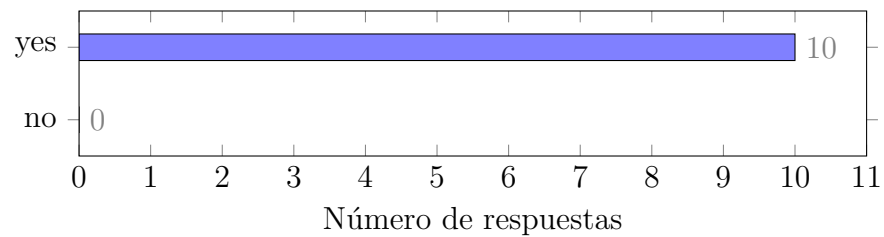


Figura 18. Resultados de la encuesta sobre divulgación.

Finalmente, la respuesta general frente al uso y necesidad de la aplicación es absolutamente positiva, algunos cambios funcionales son propuestos, pero los resultados evidencian que dichos cambios no alteran la percepción positiva del proyecto y su correcto funcionamiento.

8. IMPACTO SOCIAL

El robot asistente de ruta ha tenido siempre un enfoque social, desde sus orígenes se postuló la idea como una solución a los problemas que tanto visitantes como usuarios frecuentes de la Universidad tuvieran para ubicarse dentro de la sede principal. Es claro que la ubicación de algunas dependencias dentro del conjunto de edificios que componen la Universidad no es una tarea sencilla, por esto la solución que ofrece este proyecto es de gran provecho para todas las personas de la comunidad tomasina.

Como valor agregado a los requerimientos que ya se tenían, se agregó la posibilidad de modificar el grafo que describe la Universidad, de forma que se tuvieran dos opciones de grafo disponibles, una en la que las escaleras no cuentan como distancia acumulada ya que son la única manera de pasar de un piso a otro, y otra en la que las escaleras tengan un peso grande en relación a las distancias que existen entre un punto y otro, logrando de esta manera que las escaleras tengan una penalización de acuerdo a el tipo de usuario que ingrese al sistema.

Al comparar las rutas con ambas versiones de grafos, se observó que no hay una diferencia significativa, principalmente porque la sede principal de la Universidad no dispone de ascensores ni elementos que faciliten el acceso a personas en situación de discapacidad motriz, sin embargo, en la actualidad la Universidad Santo Tomás esta en proceso de adecuación de los edificios que componen la sede principal, buscando que estos tengan opciones de movilidad para las personas que mas lo necesitan, por lo que a futuro la disposición de un modo de accesibilidad será de gran importancia para los usuarios en condiciones especiales.

9. CONCLUSIONES

Se eligió el algoritmo de Dijkstra de entre las opciones propuestas gracias a su velocidad y efectividad en el cálculo de ruta óptima, este algoritmo presenta claras ventajas en el problema del proyecto ya que el grafo en cuestión es un grafo invariante en el tiempo y de tamaño moderado.

La construcción de un grafo que contuviera las ubicaciones de la Universidad fue una tarea dispendiosa, en la que se decidió recorrer la Universidad sobre los pasillos, sin ir directamente a la posición exacta de un destino en algunos casos, ya que esto implicaba un mayor despliegue técnico que no ofrecía ventajas considerables. Esto desembocó en que varias localizaciones compartieran la misma posición dentro del mapeado de la Universidad, sin embargo, la poca distancia física entre estas localizaciones hace que esto no tenga un efecto negativo sobre la experiencia de navegación.

Separar los diferentes puntos de interés en categorías hizo que la interacción humano-maquina fuera más sencilla, logrando de esta manera que las personas no se sintieran intimidadas por la aplicación y prefieran hacer uso de este frente a otras alternativas como explorar los edificios o pedir ayuda a otras personas.

El uso de una interfaz basada en pantalla táctil resolvió varios problemas a la vez, incluyendo la necesidad de ofrecer al usuario una experiencia visualmente agradable, así como permitir la entrada y salida de información a la aplicación de forma natural y simple. Al usar el mismo dispositivo para cumplir múltiples funciones, se redujeron los costos de implementación sin perder ningún beneficio tangible.

10. TRABAJO FUTURO

El estado actual del proyecto es una solución que ya se encuentra en marcha, y que resuelve el problema central del proyecto de encontrar un destino dentro de la Universidad. A lo largo de la ejecución del proyecto fueron surgiendo ideas que se postularon como posibles implementaciones en la aplicación desarrollada, posibles ya que el tiempo era limitado y se salían de los objetivos propuestos. Sin embargo, como trabajo futuro se pueden incluir varias mejoras a la aplicación, para que esta sea mas robusta, dentro de estas mejoras las mas destacables son las siguientes:

- **Implementación de la aplicación en dispositivos móviles:** El dispositivo Raspberry Pi cumple a la perfección con las tareas propuestas, por lo que fue natural pensar que la mayoría de teléfonos celulares de los que disponen los usuarios igualan o superan en potencia a este dispositivo también tendrían un desempeño ideal. El código fuente de la aplicación es multiplataforma y con herramientas como la compilación cruzada puede llegar a ser ejecutado en dispositivos móviles.
- **Permitir al usuario elegir el punto de partida para el cálculo de la ruta:** La razón de fijar un punto de partida es la ubicación en la que el robot funciona, es decir la entrada de la Universidad, en el caso de llevar la aplicación a otras plataformas y ejecutarla de manera móvil, es necesario que el origen sea variable de acuerdo a la ubicación de inicio para calcular la ruta. Las funciones implementadas para el cálculo de la ruta pueden modificarse de manera sencilla permitiendo elegir un origen variable.
- **Incluir edificios adicionales:** Mientras el proyecto se desarrollaba, había adecuaciones en el bloque G de la Universidad, por lo que no se conocía como sería la distribución espacial de este, y a pesar de estar contenido en la sede principal, no se pudo incluir en la aplicación. La aplicación esta diseñada para que se pueda ingresar un grafo nuevo y leerlo desde un directorio especificado, de tal manera que los cambios a la distribución espacial de la Universidad puedan ser añadidos de forma sencilla. Se requiere de ligeros ajustes adicionales, pero en general agregar vértices nuevos a el grafo es una tarea fácil de implementar.
- **Enviar información desde el robot al usuario mediante Internet:** Originalmente se buscaba hacer entrega al usuario de un mapa con la ruta para que

la tuviera durante todo su recorrido, sin embargo, las ideas planteadas no fueron concretadas por falta de recursos como una impresora. Otra propuesta fue subir los datos de los mapas a la nube para que el usuario pudiera consultarlos online, sin embargo, de nuevo la falta de un recurso clave como un servidor no permitió ejecutar esta propuesta. Con el apoyo necesario, el robot puede subir los datos presentados en la interfaz gráfica a un servidor dedicado que permita a los usuarios consultar la información desde cualquier dispositivo con conexión a internet.

11. BIBLIOGRAFÍA

- [1] DENIZ, Oscar y PATON, Julio. A vision-based localization algorithm for an indoor navigation app. Eighth International Conference on Next Generation Mobile Apps, Services and Technologies. 2014, páginas 7-12.
- [2] HONG, Zong, JUNDONG, Zhang, RUIZHENG, Jiang y JINJHONG, Feng. Automatic navigation algorithm in virtual complex indoor scenes. International Conference on Virtual Reality and Visualization. 2011, páginas 279-282.
- [3] TAHRI, Mohammed, TATSUNO, Kyoichi, KURABE, Koichi, ANDO, Hiroto, OBITSU, Hideharu, ITAKURA, Ryoma, AOTO, Takatoshi y YOSHINO, Katsumi. Improvement of a Tele-Presence Robot. International Symposium on Micro-NanoMechatronics and Human Science (MHS). 2016, páginas 1-7.
- [4] ALVES, Paulo, COSTELHA, Hugo y NEVES, Carlos. Localization and Navigation of a Mobile Robot in an Office-like Environment. 3th International Conference on Autonomous Robot Systems. 2013, páginas 1-6.
- [5] Wang, Ying. Indoor navigation for passengers in underground transis stations using smartphones. Partial fulfillment of the requierements for the degree of Doctor of Philosophy in Transportation at the Massachusetts institute of technology. 2016. 199 p.
- [6] SZCZERBA, Robert, GALKOWSKI, Peggy, GLICKSTEIN, Ira y TERNULLO, Noah. Robust algorithm for real-time route planning. IEEE Transactions on Aerospace and Electronic Systems. pages 869-878.
- [7] PARDO, Camilo, SOSA, Luis, ROJAS, Lina, LÓPEZ, Jonathan y MONSALVE, Julián. Prototype of a system to determine evacuation routes: A case in the USTA Tunja. 20th Symposium on Signal Processing, Images and Computer Vision (STSIVA). 2015, páginas 1-5.
- [8] KOSNAR, Karel, VONASEK, Voljtech y PFEUCIL, Libor. Knowledge-base Topological Exploration for Mobile Robots. European Conference on Mobile Robots (ECMR). 2015, páginas 1-6.
- [9] OSATI, Nafiseh, LÓPEZ-COLINO, Fernando, CASTRO, Angel y GARRIDO, Javier. NafisNav: an Indoor Navigation Algorithm for embedded Systems and based on

Grid Maps. IEEE International Conference on Industrial Technology (ICIT). 2015, páginas 345-350.

[10] ZOUBERT-OUSSENI, Kersane, VILLIEN, Christophe y LE GLAND, Francois. Comparison of Post-processing Algorithms for Indoor Navigation Trajectories. International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2016, páginas 1-6.

[11] CHUEN, Jue y TANG, Daijian. Tour routes optimization based on Graph theory for improving services in scenic spots. IEEE Conference Publications. 2016, páginas 1-3.

[12] WILK, Pawel and KARCIARZ, Jaroslaw. Optimization of map matching algorithms for indoor navigation in shopping malls. International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2014, páginas 661-669.

[13] CHONG, Marisol. Robótica e inteligencia artificial. El cid editor 2009.

[14] PRIETO, Alberto, LLORIS, Antonio y TORRES, Juan. Introducción a la informática. Tercera. ed. España: McGraw-Hill, 2002.

[15] DIX, Alan, FINLAY, Janet, ABOWD, Gregory and BEALE, Russell. Human-computer interaction. 3th ed. Pearson Education Limited. 2004, páginas 216-217.

[16] PEREZ, Ricardo. Una Introducción a las matemáticas discretas y la teoría de grafos. el cid editor. 2013, páginas 258-263.

[17] GANDARIAS, Peio. Seguridad en sistemas embebidos. 2006.

[18] AHO, Alfred V. Compiladores principios, técnicas y herramientas. Pearson. 2008, páginas 987-989.

ANEXOS

ANEXO A Tabla de sitios de interés

Ubicación	ID
Hall de conexión a edificio Calatayud	A0H01
Salon de teatro	A0X01
Escalera 0A (1) a 1A	A0E01
Escalera 0A (2)	A0E02
Mini escalera 0A	A0M01
Sala de profesores	A0X02
Hall de conexion a edificio Gregorio	A0H02
Sala de estudio	A0X03
Entrada carrera novena	A1H01
Carnetizacion	A1X01
Recepcion	A1X02
Mini escaleras 1A (1)	A1M01
Escalera 1A (1)	A1E01
Aula 106A	A1A06
Aula 105A	A1A05
Centro de copiado - Papeleria	A1X03
Aula 104A	A1A04
Mini escaleras 1A (2)	A1M02
Aula 103A	A1A03
Mini escaleras 1A (3)	A1M03
Escalera 1A (2)	A1E02
Aula 102A	A1A02
Aula 101A	A1A01
WC mujeres	A1W01
Escalera 1A (3)	A1E03
Cafeteria Central	A1X04
Mini escaleras 1A (4)	A1M04
Escalera 1A (4)	A1E04
Mini escaleras 1A (5)	A1M05
Mini escaleras 1A (6)	A1M06
Aula magna Fray Bartolome de las Casas	A1X05
Consultorio medico	A1X06
Escalera 2A (1)	A2E01
Aula 209A	A2A09
Aula 208A	A2A08
Cocineta	A2X01
Sala de profesores	A2X02
Cordinadores de Area de Contaduria publica	A2X03
Aula 207A	A2A07
Aula 206A	A2A06
Aula 205A	A2A05
Coordinadores de area de humanidades	A2X04
WC Hombres 2A	A2W01
Escalera 2A (2)	A2E02
Mini escalera 2A (1)	A2M01
Auditorio fundadores	A2X05

Pastoral Univeristario	A2X06
Mini escalera 2A (2)	A2M02
Escalera 2A (3)	A2E03
Aula 204A	A2A04
Representacion estudiantil de ingenieria de telecomunicaciones	A2X08
Aula 203A	A2A03
Aula 202A	A2A02
Aula 201A	A2A01
Hall de conexión con el edificio B	A2H01
Capilla	A2X09
Centro de atencion Universitaria Bogota	A2X10
Escaleras 3A (1)	A3E01
Decanatura de la dividion se ciencias economicas	A3C01
Decanatura de la facultad de Ngocios Internacionales	A3C01
Decanatura de la facultad de estadística	A3C01
Secretarias de las decanaturas de negocios ciencias economicas y estadística	A3C01
Secretarias de las decanaturas economia contaduria publica administracion de empresas y mercadeo	A3C01
Decanatura de la facultad de economia	A3C01
secretaria de division de ciencias economicas y administrativas	A3C01
Decanatura de la facultad de contaduria publica	A3C01
Decanatura de la facultad de administracion de empresas, mercadeo	A3C01
Centro de investigacion division de ciencias economicas y administrativas	A3X01
Aula 312A	A3A12
Sala de profesores de ciencias basicas	A3X02
WC mujeres 3A	A3W01
Escaleras 3A (2)	A3E02
Mini escalera 3A (1)	A3M01
Aula 311A	A3A11
Aula 310A	A3A10
Sala de profesores negocios internacionales	A3X03
Sala de tutorias de economia	A3X04
Mini escalera 3A (2)	A3M02
Escaleras 3A (3)	A3E03
Centro Investigativo de Robotica	A3X05
Representacion de economia	A3X06
Aula 307A	A3A07
Aula 306A	A3A06
Aula 305A	A3A05
Aula 304A	A3A04
Aula 303A	A3A03
Aula 302A	A3A02
Aula 301A	A3A01
Escalera 4A	A4E01
Aula 401A	A4A01
Auditorio Torreón	A4A01

Escaleras 0.5B	B0E01
Departamento editorial	B0D01
Mini escalera 0.5B	B0M01
Escalera 1B (1)	B1E01
Cafeteria edificio B piso 1	B1X01
Aula 101B	B1A01
Aula 102B	B1A02
Aula 103B	B1A03
Auditorio Joseph Luis Sanz Tena	B1X02
WC mujeres	B1W01
Escalera 1B (2)	B1E02
Escalera 2B (1)	B2E01
Mini escalera 2B	B2M01
Cafeteria edificio B piso 2	B2X01
Aula 201B	B2A01
Representacion ciencias economicas	B2X02
Aula 202B	B2A02
Aula 203B	B2A03
Aula 204B	B2A04
Aula 205B	B2A05
Aula 206B	B2A06
Aula 207B	B2A06
Aula 208B	B2A08
WC hombres	B2W01
Escalera 3B	B3E01
Aula 301B	B3A01
Representacion diseño grafico	B3X01
Aula 302B	B3A02
Aula 303B	B3A03
Aula 304B	B3A04
Aula 305B	B3A05
Aula 306B	B3A06
Aula 307B	B3A07
Aula 308B	B3A08
WC mujeres	B3W01
Escalera 4B (1)	B4E01
Guardamaleteros	B4X01
Biblioteca	B4X02
Facultad de comunicación social	B4X03
Sala de profesores	B4X03
Decano de ciencias sociales	B4X03
Facultad de diseño grafico	B4X03
Servicios y atencion tecnica al estudiante	B4X03
Aula 401B	B4A01
Aula 402B	B4A02
Estudio grabacion	B4X04
Estudio television	B4X05

Sala de edicion	B4X06
Estudio de radio 1	B4A01
Cabina de control 1	B4X07
Cabina de control 2	B4X07
Estudio de radio 2	B4X08
Departamento de comunicaciones	B4D01
Direccion de comunicaciones	B4X09
Hall 1A	C1H01
Mini escaleras 1C (1)	C1M01
WC mujeres 1C	C1W01
Cajeros	C1X01
Departamento de ciencias basicas humanidades e idiomas	C1D01
Bienestar universitario	C1X02
Tienda Tomasina	C1X03
Gimnasio Univeristario	C1X04
Escaleras 1C	C1E01
Mini escaleras 1C (2)	C1M02
Salida Principal 1A	C1S01
Escaleras 2C (1)	C2E01
WC Hombres 2C	C2W01
Hall 2A	C2H01
Mini escaleras 2C	C2M01
Oficina de Registro y control	C2X01
Laboratorio ETM1	C2L01
Laboratorio ETM2	C2L02
Laboratorio ETM3	C2L03
Sala de profesores de Electronica	C2L04
Laboratorio ETM5	C2L05
Laboratorio ETM6	C2L06
Investigacion Electronica	C2L07
Laboratorio ETM8	C2L08
Laboratorio ETM9	C2L09
Laboratorio ETM10	C2L10
Atencion Prestamos	C2X02
Mantenimiento equipos	C2X03
Laboratorio ETM11	C2L11
Investigacion Telecomunicaciones	C2L04
Laboratorio ETM 13	C2L03
Representacion estudiantil Ingenieria Electronica	C2X04
Representacion estudiantil ingenieria Industrial	C2X05
Escaleras 3C (1)	C3E01
Sala de proyecciones	C3X01
Hall 3C	C3H01
Sala de tutorias	C3X02
Aula 313C	C3A13
Aula 312C	C3A01

Aula 311C	C3A02
Aula 310C	C3A03
Aula 309C	C3A04
Aula 308C	C3A05
Aula 307C	C3A07
Aula 306C	C3A06
Sala de tutorias division de Ingenierias	C3X03
Escaleras 3C (2)	C3E02
WC hombres	C3W01
Aula 305C	C3A05
Aula 304C	C3A04
Aula 303C	C3A03
Aula 302C	C3A02
Aula 301C	C3A01
WC mixto	C3W02
IEEE	C3X04
Escalera 4C	C4E01
Oficina de educacion Virtual	C4X01
Rack de sistemas	C4X02
Oficina de sistemas	C4X02
Sala de computo 403	C4A03
Sala de computo 404	C4A04
Sala de computo 405	C4A03
Sala de computo 407	C4A07
Decanatura Diseño industrial	C4A04
Decano Academico ingenieria electronica	C4C01
Decano Academico ingenieria Ambiental	C4C01
Decano de division de Ingenieria	C4C01
Decano Academico de ingenieria civil	C4C01
Secretaria de division de ingenierias	C4C01
Decano Academico de ingenieria mecanica	C4C01
Sala de computo 408	C4A08
Sala de computo 406	C4A06
Sala de computo 402	C4A02
Sala de computo 401	C4A01
WC hombres 4C	C4W01
Escalera 5C (1)	C5E01
Escalera 5C (2)	C5E02
WC Mixto	C5W01
Recepcion	C5X01
Sindicatura	C5X01
Oficina de proyectos especiales	C5X01
secretaria general	C5X01
Auditoria interna	C5X01
Oficina Juridica	C5X01
Salon Luis J torres OP	C5X01
Vicerrectoria academica	C5X01

Vicerrectoria administrativa y financiera	C5X01
Talento humano	C5X01
Nomina y seguridad social	C5X01
Contabilidad	C5X01
Escalera 6C	C6E01
Secretaría de idiomas	C6X01
Departamento de idiomas	C6D01
Sala de profesores	C6X02
Aula 603C	C6A03
WC Hombres y mujeres 6C	C6W01
Laboratorio de idiomas	C6L01
Rectoria	C6X03
Aula 601C	C6A01
Hall de conexión entre pisos (TRIANGULO)	F1X01
Mini escalera 1F (1)	F1M01
Escalera 1F (1)	F1E01
WC Hombres 1F	F1W01
Escalera 1F (2)	F1E02
Mini escalera 1F (2)	F1M02
Taller de maquinaria y herramienta	F1L01
Centro de formacion en nuevas tecnologias de produccion	F1X02
Laboratorio de hidraulica	F1L02
Escalera 2F (1)	F2E01
Escalera 2F (2)	F2E02
Representacion estudiantil ingenieria civil	F2X01
Representacion estudiantil ingenieria mecanica	F2X02
Laboratorio de suelos y hormigon	F2L01
Laboratorio de metrologia	F2L02
Escalera 3F	F3E01
Laboratorio de materiales	F3L01
Laboratorio de electricidad	F3L02
Laboratorio de aguas quimica y biologia	F3L03
Escalera 4F	F4E01
Laboratorio de fotogrametria	F4L01
Laboratorio de tratamientos termicos y metalografia	F4L02
Laboratorio de motores de soldadura	F4L03

ANEXO B Encuesta

Interacción con aplicación asistente de ruta

Noviembre 2017

Estimado Usuario,

La actividad en la que usted va a participar consiste en la interacción con una aplicación en la que se calculará la ruta más corta desde la entrada de la Universidad Santo Tomás hasta cualquier lugar dentro de ésta, indicando por medio de una interfaz gráfica, y de una lista de instrucciones cómo llegar a dicho punto.

Para esta experiencia usted deberá seguir las instrucciones dadas inicialmente por el instructor y éste lo acompañará durante la interacción, luego de ello le pediremos que responda unas preguntas sobre su experiencia.

Si está de acuerdo con estas condiciones, le pedimos el favor de diligenciar la siguiente información

Nombres y apellidos:

Edad:

Fecha:

¿Qué rol desempeña dentro de la Universidad?

☐ Estudiante

☐ Administrativo

☐ Docente

Encuesta sobre la experiencia de interacción con la aplicación asistente de ruta

Valore los siguientes aspectos de software de acuerdo con su experiencia con la aplicación, donde 1 es muy malo y 5 es excelente.

Software	Muy malo	Malo	Regular	Bueno	Excelente
	1	2	3	4	5
Útil: La información dada por la aplicación le permitió ubicar el destino solicitado.					
Intuitivo: Es fácil navegar a través de las distintas opciones de la aplicación.					
Agradable: El entorno gráfico de la aplicación es visualmente atractivo.					
Interactivo: La forma de interacción con la aplicación (pantalla táctil) le resulta apropiada.					

Valore los siguientes aspectos del cálculo de la ruta de acuerdo con su experiencia con la aplicación donde 1 es muy malo y 5 es excelente.

Calculo de la ruta	Muy malo	Malo	Regular	Bueno	Excelente
	1	2	3	4	5
¿Qué tan adecuado le parece el tiempo del cálculo de la ruta a su destino?					
De acuerdo con sus conocimientos de la universidad, ¿qué tan apropiada le parece la ruta resultante?					

Entre 0% y 100%, ¿Qué tan necesario le parece para la Universidad la aplicación asistente de ruta?

Escoja una o varias de las respuestas de acuerdo con la siguiente pregunta

¿Qué le mejoraría a la aplicación?

☐ Tamaño de la pantalla

☐ Visualización de la ruta

☐ Tipo de interacción

☐ Opciones de búsqueda

☐ Lista de instrucciones

☐ Entorno gráfico

☐ Ninguna

☐ Otra

Si su respuesta fue **otra**, indique ¿Cuál?

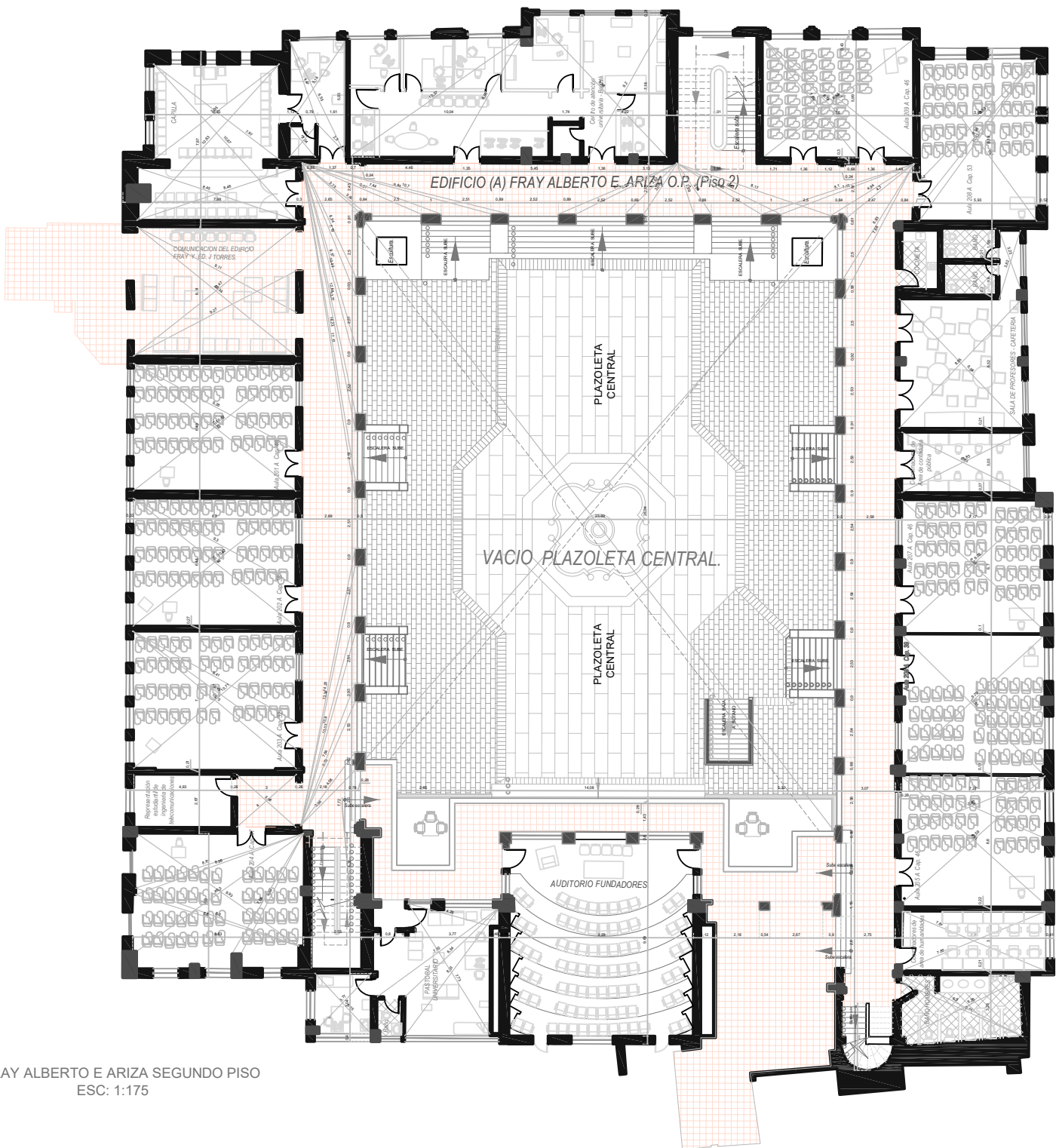
¿Fue satisfactoria su experiencia con la aplicación? ¿Por qué?

¿Recomendaría la aplicación? ¿Por qué?

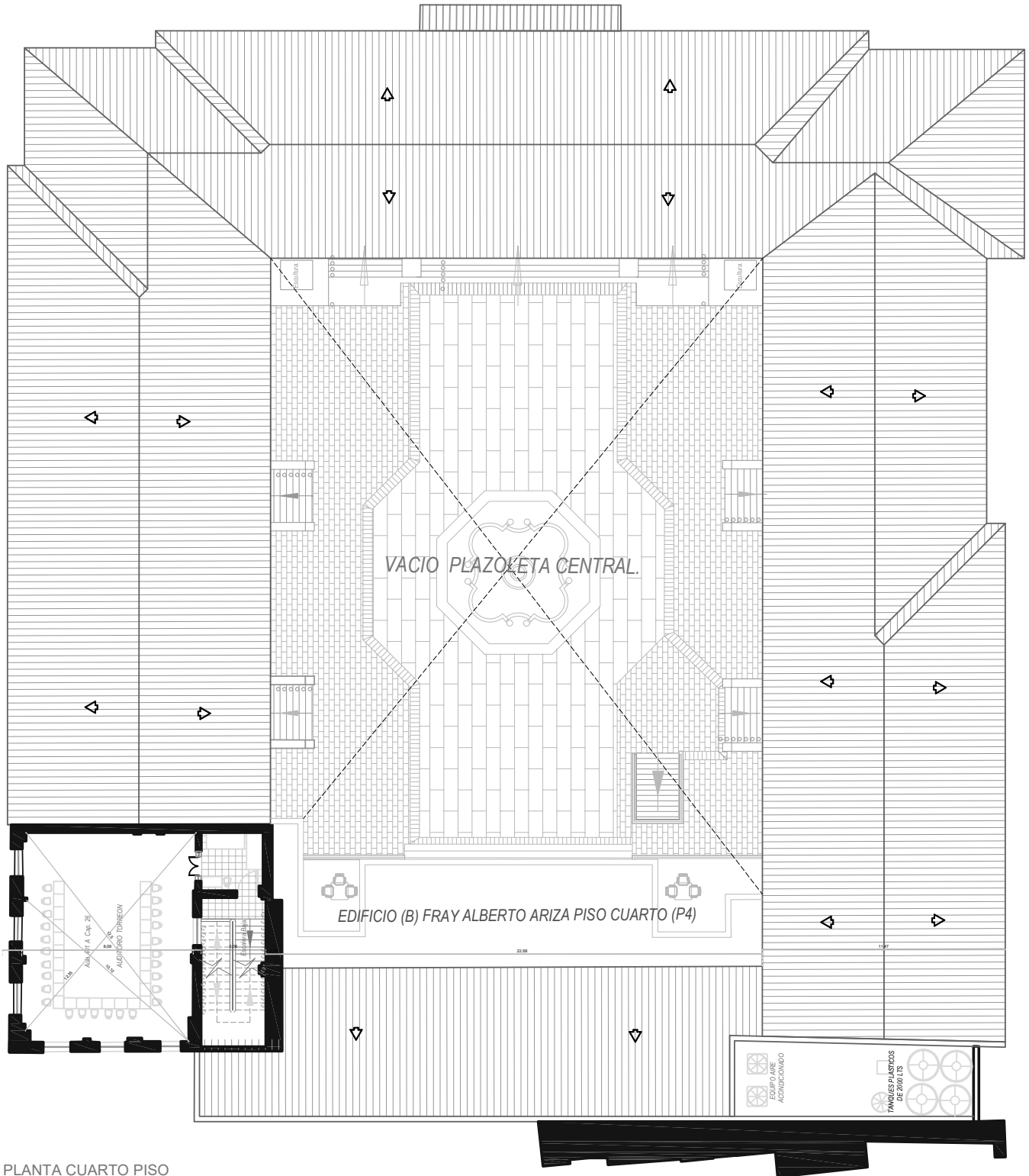
ANEXO C Planos



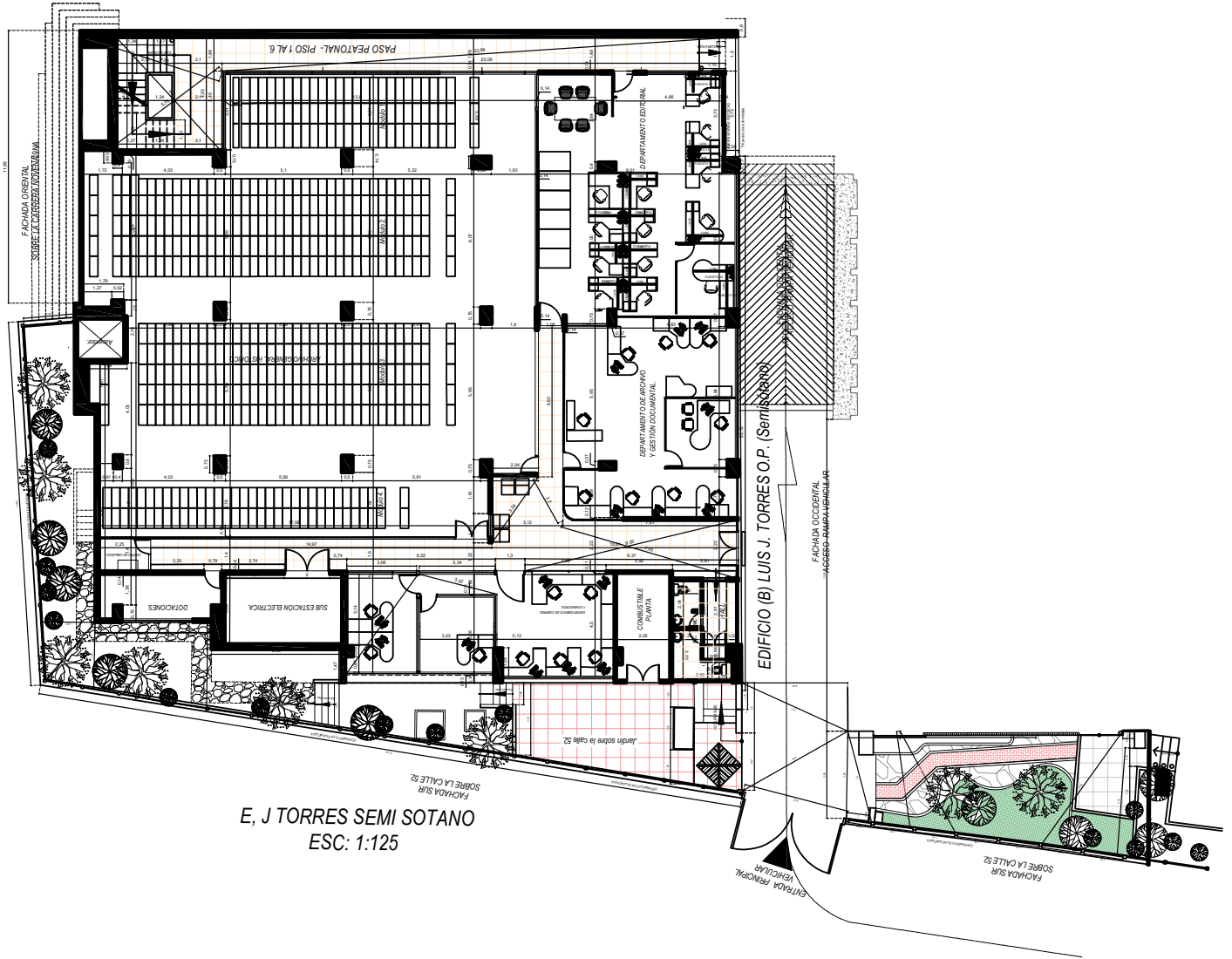




FRAY ALBERTO E ARIZA SEGUNDO PISO
ESC: 1:175

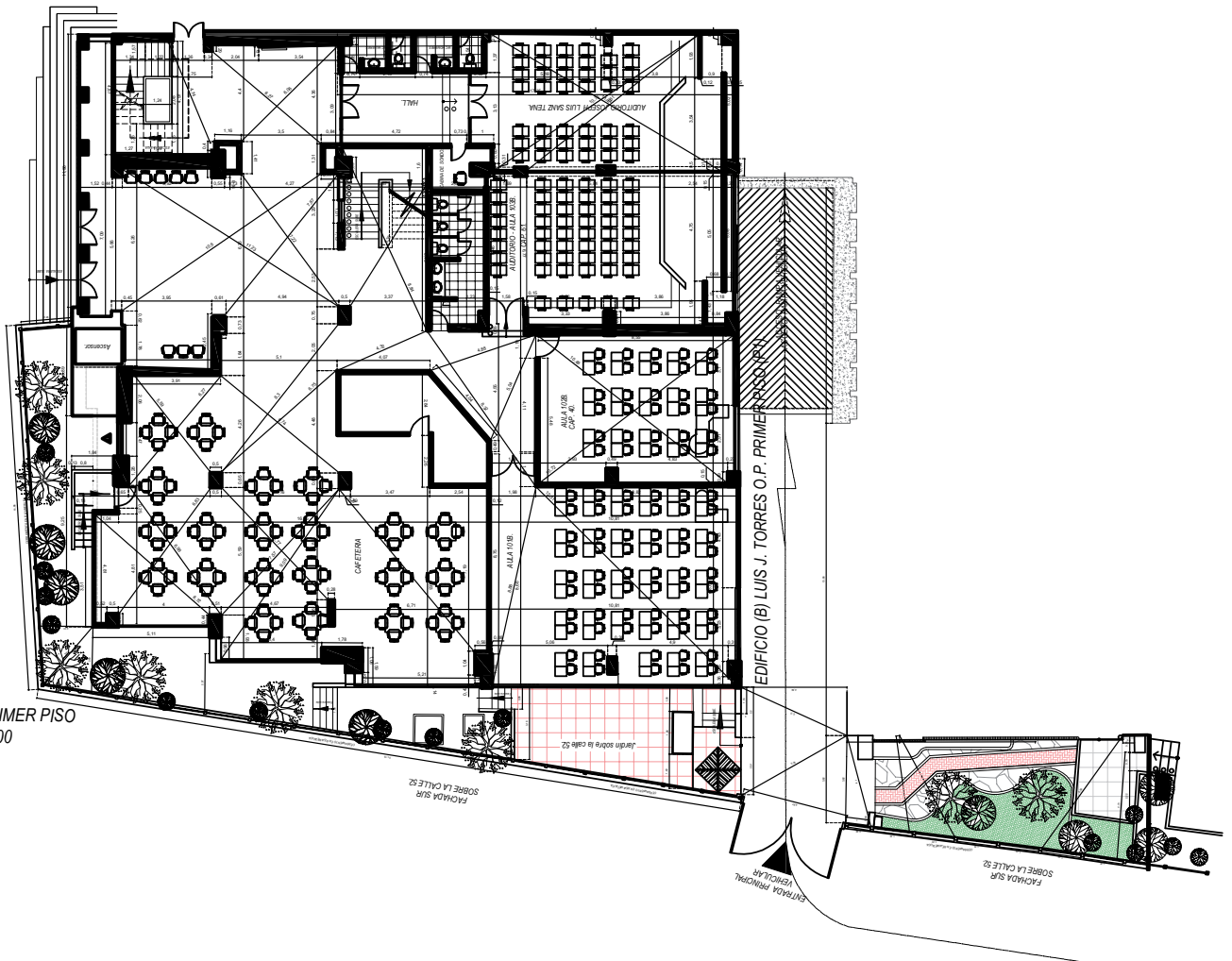


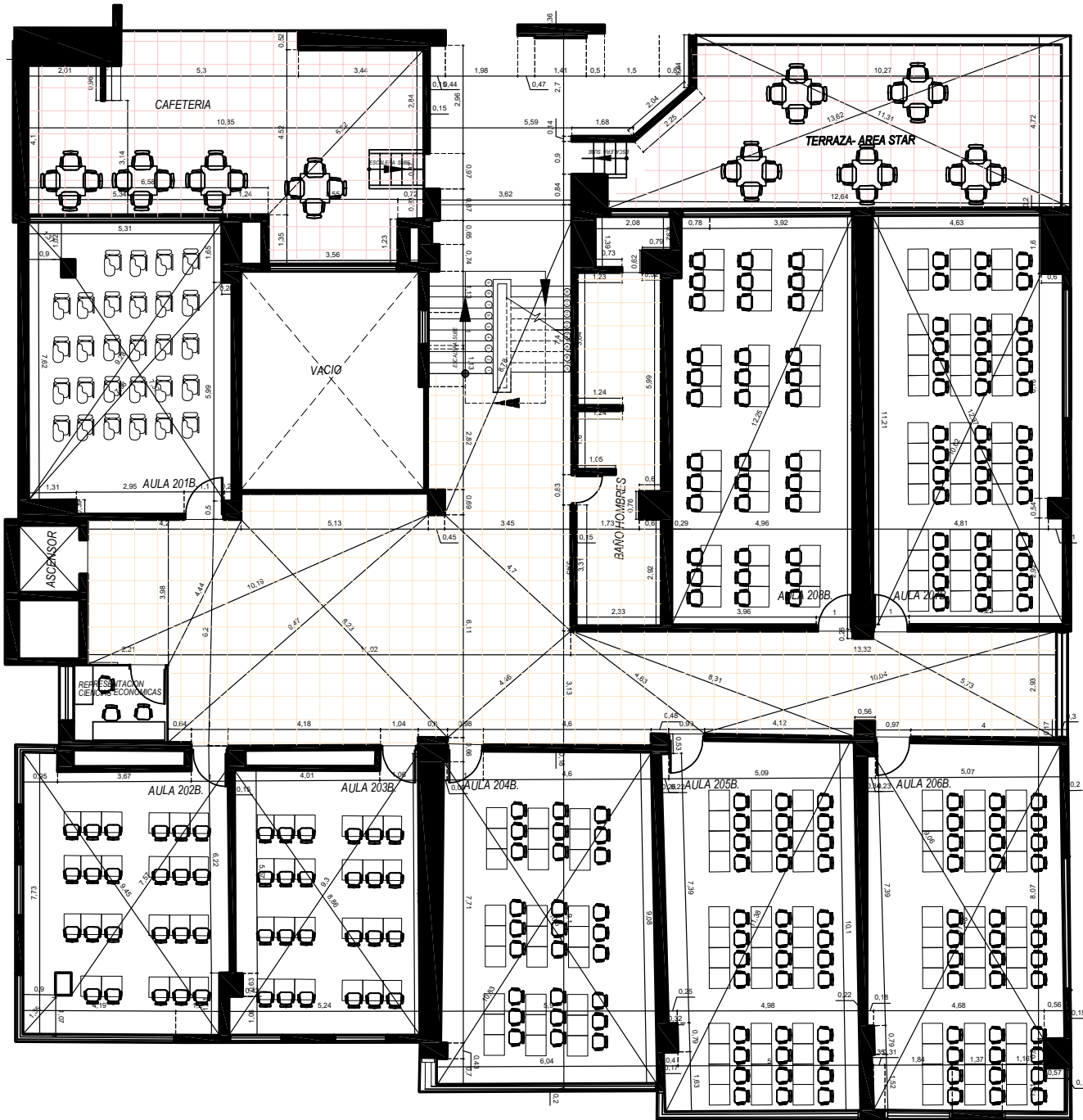
FRAY ALBERTO ARIZA PLANTA CUARTO PISO
 ESC: 1:175



FACHADA ORIENTAL
SOBRE LA CARRERA NOVENA Y UNA

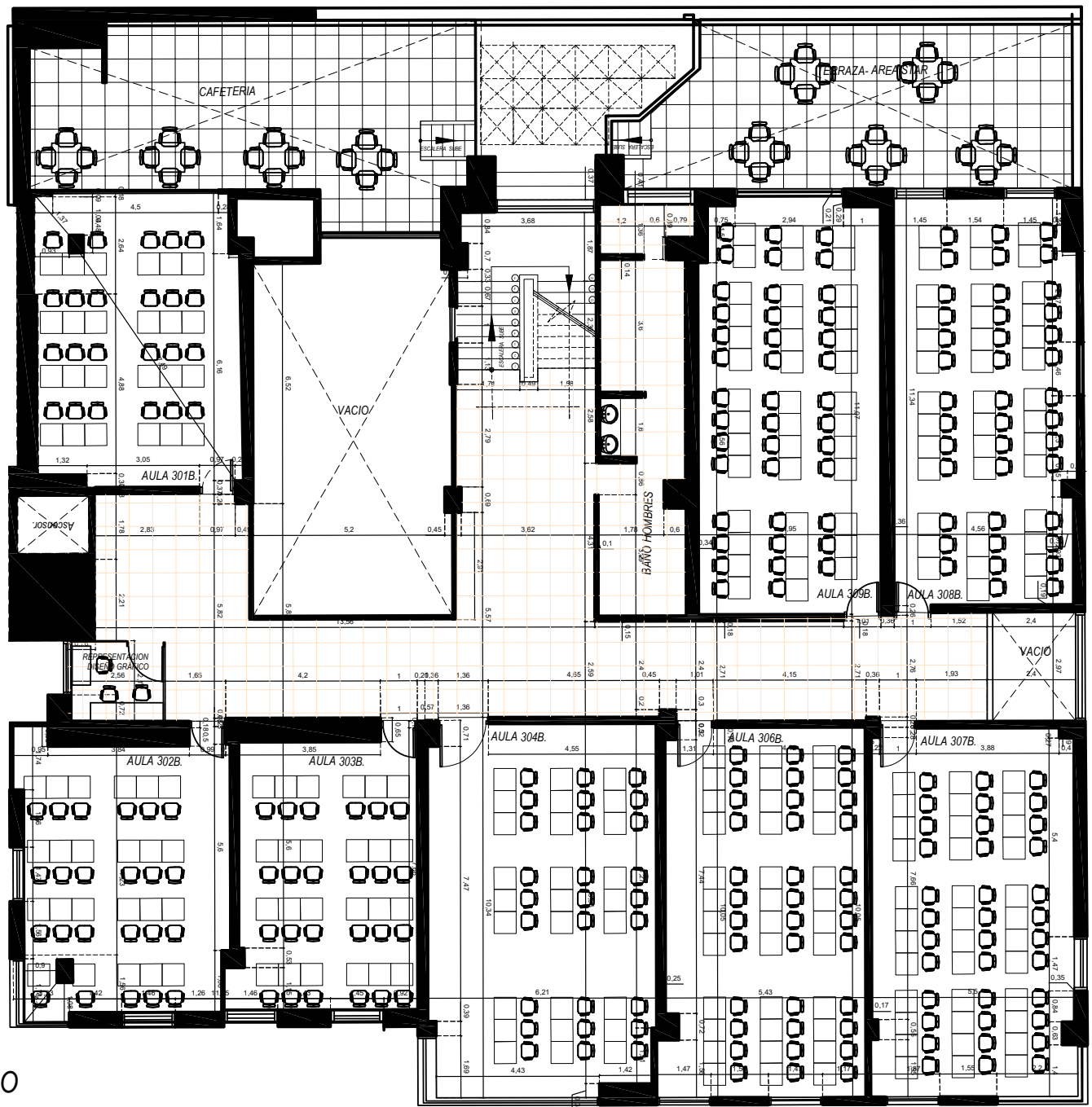
E, J TORRES PRIMER PISO
ESC: 1:100





EDIFICIO (B) LUIS J. TORRES O.P. PISO SEGUNDO (Piso 2)

VDO PISO



ER PISO

CUARTO PISO

