



## PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para  
obtener el Título de INGENIERO DE SISTEMAS

# DISEÑO E IMPLEMENTACIÓN DE UNA INTERFAZ GRÁFICA DE USUARIO PARA EL SIMULADOR DE PROTOCOLO DE ENRUTAMIENTO ARAP.

Por

Br. Sergio Quintana

Tutor: Prof. Rodolfo Sumoza

Octubre 2017

©2017 Universidad de Los Andes Mérida, Venezuela

C.C. Reconocimiento

# Diseño e implementación de una interfaz gráfica de usuario para el simulador de protocolo de enrutamiento ARAP.

Br. Sergio Quintana

Proyecto de Grado — Sistemas Computacionales, 66 páginas

**Resumen:** En este trabajo se diseñó y desarrolló la interfaz gráfica de un simulador para protocolo de enrutamiento que provee anonimato basado en algoritmos de optimización de colonias de hormigas artificiales para facilitar las pruebas a realizarse en el mismo. El simulador ARAP fue desarrollado como un módulo sobre el simulador de redes de propósito general basado en eventos discretos NS3. Esta Interfaz fue desarrollada para facilitar el ingreso de parámetros y la ejecución de experimentos así como la interpretación de resultados en los mismos.

**Palabras clave:** interfaz gráfica, simulación, redes, arap

# Índice general

<b>Agradecimientos</b>	<b>ix</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Planteamiento del problema . . . . .	2
1.3. Justificación . . . . .	3
1.4. Objetivos . . . . .	3
1.4.1. Objetivo general . . . . .	3
1.4.2. Objetivos específicos . . . . .	3
1.5. Metodología . . . . .	4
1.6. Alcance . . . . .	5
<b>2. Marco Teórico</b>	<b>6</b>
2.1. Interfaz Gráfica de Usuario (GUI) . . . . .	6
2.2. Interfaz de Línea de Comandos (CLI) . . . . .	6
2.2.1. GUI Wrapper . . . . .	7
2.3. WIMP . . . . .	7
2.3.1. Ventana . . . . .	8
2.3.2. Menú . . . . .	8

2.3.3.	Widget . . . . .	8
2.3.4.	Pestaña . . . . .	9
2.3.5.	Puntero . . . . .	9
2.4.	Simulador . . . . .	9
2.5.	Protocolo de enrutamiento . . . . .	10
2.6.	Anonimato . . . . .	10
2.7.	ns-3 . . . . .	11
2.8.	ARAP . . . . .	11
2.8.1.	Protocolo simulado por ARAP . . . . .	11
2.9.	Qt Creator y C++ . . . . .	12
2.10.	C++ . . . . .	13
<b>3.</b>	<b>Interfaz de usuario para el simulador ARAP</b>	<b>14</b>
3.1.	Estructura de la Interfaz de Usuario del Simulador ARAP . . . . .	15
3.1.1.	Estructura de ARAPGUI . . . . .	15
3.1.2.	Estructura de ARAP Results . . . . .	16
3.1.3.	Estructura de ARAP Compare . . . . .	17
3.1.4.	Estructura de ARAP Results . . . . .	17
3.2.	ARAPGUI . . . . .	17
3.2.1.	ARAPResults . . . . .	26
3.2.2.	Animación . . . . .	28
3.3.	ARAP Compare . . . . .	31
3.4.	Arquitectura . . . . .	36
<b>4.</b>	<b>Pruebas manuales</b>	<b>39</b>
4.1.	Casos de prueba . . . . .	40

4.1.1.	Pruebas sobre la Navegación en ARAPGUI . . . . .	40
4.1.2.	Pruebas sobre la Restricción de Valores en ARAPGUI . . . . .	41
4.1.3.	Pruebas sobre el cambio de Distribuciones en parámetros de ARAPGUI . . . . .	44
4.1.4.	Pruebas sobre la Modificación de Rangos en parámetros de ARAPGUI . . . . .	45
4.1.5.	Pruebas sobre la lectura de parámetros . . . . .	48
4.1.6.	Pruebas sobre las Características de ARAP Results . . . . .	49
4.1.7.	Pruebas sobre las Características de ARAPCompare . . . . .	50
<b>5.</b>	<b>Conclusiones y Recomendaciones</b>	<b>53</b>
5.1.	Conclusiones . . . . .	53
5.2.	Recomendaciones . . . . .	54
<b>A.</b>	<b>Capturas de pantalla de la interfaz de usuario ARAPGUI</b>	<b>56</b>
	<b>Referencias Bibliográficas</b>	<b>65</b>

# Capítulo 1

## Introducción

La intención de la creación de una interfaz gráfica de usuario es relacionar la funcionalidad de un programa al uso que le dan los usuarios a los que va dirigido, poniendo en segundo plano a la arquitectura interna del mismo. Una buena interfaz gráfica de usuario puede traer muchos beneficios como: aumento de la productividad, menor requerimiento de conocimiento técnico o entrenamiento para usar el sistema y una menor carga cognitiva al usuario mostrando sólo la información más importante. [9]

En el caso de la interfaz de usuario desarrollada en este proyecto de grado para el simulador ARAP, se persiguió principalmente agilizar el ingreso de parámetros para la ejecución de experimentos, resumir y mostrar lo esencial de los resultados y comparar diferentes resultados de simulaciones con parámetros distintos para estudiar su comportamiento.

Este trabajo está estructurado de la siguiente manera: en el capítulo uno (1) se hace una introducción a este trabajo, donde se explican los objetivos y la justificación de este proyecto, en el capítulo dos (2) se construye un marco teórico con los conceptos y tecnologías usadas en este proyecto de grado, en el capítulo tres (3) se explica la estructura de la interfaz gráfica de usuario desarrollada, en el capítulo cuatro (4) se muestran los resultados de las pruebas a las que fué sometida la interfaz de usuario y en el capítulo cinco (5) se discuten las conclusiones sobre el proyecto y se ofrecen

recomendaciones para trabajos futuros.

## 1.1. Antecedentes

Las interfaces gráficas de usuario cambiaron la forma en que las personas se relacionaban con las computadoras. El uso de instrucciones específicas para cada función de un programa fué reemplazado por el uso de interacciones genéricas entre el puntero, los demás elementos del entorno gráfico y en menor frecuencia el teclado. Una vez dominadas estas interacciones, pueden ser usadas en cualquier otro programa con una interfaz de usuario similar. Además de la usabilidad, las GUI pueden aumentar la productividad en el trabajo, con interfaces simples que ofrezcan un fácil acceso a las acciones repetidas con mayor frecuencia.

Existen herramientas con interfaces gráficas que modelan y simulan el comportamiento de redes complejas con fines educativos como Cisco Packet Tracer[1] o GNS3[7]. También existen algunas interfaces gráficas compatibles con el simulador de redes NS3 sobre el cual se desarrolló el simulador ARAP, como por ejemplo PyViz[2], NetAnim[5] y Topology Generator[8], sin embargo éstas herramientas son de propósito general y no son apropiadas para las pruebas que se desea realizar sobre el protocolo de enrutamiento.

## 1.2. Planteamiento del problema

En [10] se desarrolló un simulador para probar un protocolo de enrutamiento que provee anonimato basado en algoritmos de optimización de colonias de hormigas artificiales propuesto en [11], dicho simulador fue realizado utilizando el simulador de redes de propósito general basado en eventos discretos NS3. El simulador no contaba con una interfaz gráfica de usuario, era necesario modificar los parámetros de la simulación en un archivo de texto, que servía como entrada de parámetros para el simulador. También los resultados de las simulaciones eran almacenados en múltiples archivos de texto. Por estas razones se decidió desarrollar una interfaz gráfica de

usuario que facilite el ingreso de parámetros de la simulación y la interpretación de sus resultados. Ésto con el fin de simplificar el diseño de los experimentos que pondrán a prueba al protocolo. La interfaz gráfica de usuario facilita la entrada de parámetros y presenta la información pertinente en la salida de datos.

### 1.3. Justificación

La finalidad del simulador ARAP es estudiar el desempeño de un protocolo de enrutamiento para proveer anonimato que no ha sido probado y es pertinente que el simulador posea una interfaz gráfica de usuario que facilite el análisis de los resultados de las simulaciones y la implementación de las pruebas.

### 1.4. Objetivos

#### 1.4.1. Objetivo general

Crear una interfaz gráfica de usuario para el Simulador ARAP de protocolo de enrutamiento que provee anonimato basado en algoritmos de optimización de colonias de hormigas artificiales.

#### 1.4.2. Objetivos específicos

- Identificar las características del simulador y los elementos necesarios en la interfaz gráfica.
- Diseñar de la interfaz: elegir las herramientas que estarán disponibles y su distribución, escoger los gráficos para mostrar los resultados de las simulaciones.
- Diseñar la arquitectura del sistema.
- Implementar la interfaz gráfica para ARAP.
- Diseñar un plan de pruebas y llevarlas a cabo.

## 1.5. Metodología

Este proyecto se estructura en dos (2) partes: La creación de una interfaz de usuario para el ingreso de parámetros al simulador ARAP y la creación de una interfaz de usuario para presentar los resultados de las simulaciones y comparar los resultados de diferentes experimentos.

Para ambas partes de este proyecto se empleó una metodología de desarrollo incremental e iterativo hasta la culminación de la interfaz. En cada iteración se escogieron los puntos a desarrollar, se programaron y se probaron para hacer correcciones. Luego se diseñaron y ejecutaron las pruebas para la interfaz culminada y se hicieron modificaciones y correcciones generales.

En la primera parte de la interfaz se realizaron las siguientes actividades:

- Estudiar las características del simulador ARAP.
- Identificar los elementos con los que interactúa el usuario en la entrada de parámetros.
- Elegir la tecnología a ser usada para desarrollar la interfaz.
- Configurar el ambiente de trabajo.
- Modelar la interfaz de ingreso de parámetros.
- Codificar los distintos componentes de la interfaz.
- Verificar el correcto funcionamiento de la interfaz y depurar errores.
- Diseñar y ejecutar pruebas del funcionamiento.

En la segunda parte de la interfaz se realizaron las siguientes actividades:

- Estudiar los resultados del simulador.
- Identificar los elementos de mayor importancia en los resultados.

- Modelar la interfaz de análisis de resultados.
- Codificar los distintos componentes de la interfaz.
- Verificar el correcto funcionamiento de la interfaz y depurar errores.
- Diseñar y ejecutar pruebas del funcionamiento.

Las pruebas finales, a modo de caja negra, consistieron en:

- Enlistar los casos de prueba para buscar debilidades en el sistema.
- Caracterización de los diferentes casos de prueba.
- Ejecución de las pruebas manualmente, ya que se trata de una interfaz de usuario.
- Anotar los resultados obtenidos.

## 1.6. Alcance

Este proyecto representa un componente del simulador ARAP; la interfaz gráfica con la cual el usuario realizará experimentos sobre el comportamiento del protocolo de enrutamiento para proveer anonimato basado en algoritmos de optimización de colonias de hormigas artificiales. Este trabajo busca agilizar las pruebas sobre el protocolo antes mencionado y facilitar la interpretación de resultados por medio de gráficas. No se incluye el diseño de pruebas formales para estudiar el comportamiento del protocolo.

# Capítulo 2

## Marco Teórico

Este capítulo está dedicado a explicar los conceptos fundamentales para el desarrollo de este proyecto. Desde algunos términos básicos como interfaz gráfica de usuario, hasta la descripción del simulador ARAP y el protocolo que simula. También se describen las tecnologías a utilizar para el desarrollo del proyecto.

### 2.1. Interfaz Gráfica de Usuario (GUI)

Una interfaz gráfica de usuario (GUI, del inglés Graphical User Interface) es una interfaz humano-computadora que utiliza ventanas, íconos y menús y que puede ser manipulada de manera gráfica mediante algún dispositivo apuntador, como un ratón o una pantalla táctil. Las GUI se introdujeron como reacción a la empinada curva de aprendizaje de las interfaces de línea de comandos (CLI), que utilizan sólo texto y son accedidas únicamente mediante un teclado.[14]

### 2.2. Interfaz de Línea de Comandos (CLI)

La interfaz de línea de comandos (del inglés command-line interface, CLI) es un método que permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple.[13]

Dado que los comandos disponibles en las interfaces de línea de comandos pueden ser muchos, se pueden realizar operaciones complejas utilizando secuencias cortas de palabras y símbolos. Esto permite una mayor eficiencia y productividad una vez que se aprenden muchos comandos, sin embargo, alcanzar este nivel lleva tiempo porque las palabras de comando pueden no ser intuitivas y fáciles de recordar. Adicionalmente, el uso de la línea de comandos puede volverse lento y propenso a errores cuando los usuarios deben introducir comandos largos que comprenden muchos parámetros o varios nombres de archivo diferentes a la vez. En contraste, las interfaces gráficas de ventanas, iconos, menús y puntero (WIMP) presentan a los usuarios con widgets que reemplazan y pueden activar algunos de los comandos disponibles del sistema de manera intuitiva.

### 2.2.1. GUI Wrapper

Los envoltorios (Wrappers) de interfaz gráfica de usuario (GUI) reemplazan el uso de las versiones de interfaz de línea de comandos (CLI) de aplicaciones de software y sus interfaces de usuario basadas en texto y la escritura de comandos. Mientras que la CLI permite a los usuarios ejecutar un programa de forma no interactiva, los envoltorios de GUI que los envuelven evitan la empinada curva de aprendizaje de la línea de comandos. Al usar un envoltorio de GUI, los usuarios pueden interactuar, ejecutar, detener y cambiar parámetros de trabajo intuitivamente, a través de iconos gráficos e indicadores visuales de un entorno gráfico. [14]

## 2.3. WIMP

En la interacción humano-ordenador, WIMP es un acrónimo que significa “ventanas, iconos, menús, puntero”, que denota un estilo de interacción que utiliza estos elementos de la interfaz de usuario.[17]

### 2.3.1. Ventana

En computación, una ventana es un elemento de control gráfico que muestra información independientemente del resto de la pantalla. Consiste en un área visual que contiene parte de la interfaz gráfica de usuario del programa al que pertenece. Muestra la salida y puede permitir la entrada de datos a uno o más procesos.[15]

Un programa puede tener múltiples ventanas, habitualmente se tiene una ventana principal y ventanas de diálogo generalmente más pequeñas y básicas que son abiertas por un programa para mostrar información al usuario u obtener información del usuario. Éstas últimas ventanas pueden ser no modales, lo que significa que permiten alternar el foco a cualquier otra ventana presente dentro del entorno gráfico, o modales, que no permiten regresar el foco a la ventana que le dió origen hasta que el usuario tome una acción determinada.

### 2.3.2. Menú

Los menús permiten al usuario ejecutar comandos seleccionando de una lista de opciones. Las opciones se seleccionan con un ratón, teclado u otro dispositivo señalador dentro de una GUI. Los menús son convenientes porque muestran qué comandos están disponibles dentro del software. Esto limita la cantidad de documentación que el usuario necesita leer para entender el software.[15]

### 2.3.3. Widget

En una interfaz gráfica de usuario, un widget es cualquier elemento de interacción directa para leer o editar información sobre una aplicación, como un botón o una barra de desplazamiento.[15]

### 2.3.4. Pestaña

Una pestaña o tab es un elemento de control gráfico que permite que varios documentos o paneles sean contenidos dentro de una sola ventana, utilizando las pestañas como un widget de navegación para cambiar entre conjuntos de documentos. Cuando se activa el panel de vista, o ventana, muestra widgets asociados con esa pestaña; grupos de pestañas permiten al usuario cambiar rápidamente entre diferentes widgets.[15]

### 2.3.5. Puntero

En computación, un puntero es un símbolo en el monitor de la computadora u otro dispositivo de visualización que refleja los movimientos de un dispositivo señalador, comúnmente un ratón, un touchpad o un estilete (lapicero). Señala el punto donde las acciones del usuario tienen lugar. Es distinto al cursor, que responde a la entrada del teclado.[15]

## 2.4. Simulador

La simulación es la imitación de un proceso o sistema del mundo real a través del tiempo. Así, un simulador es una herramienta de análisis que nos permite estudiar un sistema real sin necesidad de interactuar directamente con el mismo. Esto es particularmente útil cuando no contamos con el sistema a estudiar, o cuando los experimentos planteados conllevan altos costos o riesgos.[3] A la hora de simular sistemas muy complejos se vuelve una necesidad realizar simulaciones por computadoras. Existen tres categorías importantes de simulación por computadora:

- Tipo Monte Carlo: Utiliza muestras aleatorias y probabilidad para simular sistemas complejos que pueden ser deterministas.
- Simulaciones Continuas: Sistemas modelados por ecuaciones diferenciales o algebraicas que dependen del paso del tiempo de forma continua.

- Por Eventos discretos: A diferencia de las simulaciones continuas en éstas se asume ocurren eventos que cambian el sistema en instantes determinados de tiempo y que no ocurre nada de importancia durante los bloques de tiempo cada evento discreto.

## 2.5. Protocolo de enrutamiento

En las telecomunicaciones un protocolo se define como un conjunto de normas que permite la comunicación entre computadoras, estableciendo la forma de identificación de los mismos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse.[16]

Enrutamiento es el proceso de selección de una ruta para el tráfico en una red, o a través de múltiples redes. Un protocolo de enrutamiento es entonces un protocolo que regula específicamente los caminos que recorren los mensajes intercambiados en una red.

## 2.6. Anonimato

El anonimato se refiere a la capacidad de un usuario de realizar una acción en Internet sin que su identidad sea asociada a dicha acción.

En Internet la mayoría de las comunicaciones ocurren de manera esencialmente anónima, la gente publica información y se comunica a través de pseudónimos no asociados a su identidad real. No obstante, cada usuario de internet accesa información a través de una dirección IP particular. Ésta dirección IP puede aportar información del usuario, y mediante técnicas de análisis de tráfico se pueden crear perfiles de comportamiento de los usuarios sin su consentimiento.

Existen servicios como TOR o I2P que proveen anonimato en internet ante ataques de análisis de tráfico. Éstos aplican varias capas de cifrado a los paquetes y los envían por una ruta predeterminada a través de una red anónima. Cada nodo en

la ruta conoce el nodo anterior y el siguiente, pero ninguno puede conocer el origen y el destino del mensaje, por lo que un análisis de tráfico no podría saber qué usuarios de la red anónima se están comunicando. [12]

## 2.7. ns-3

El software ns-3 es un simulador de redes basado en eventos discretos, usado principalmente para fines educacionales y de investigación. ns-3 es un software libre, licenciado bajo la licencia GNU GPLv2. Además de simular ciertos elementos y protocolos básicos de redes, permite al usuario agregar nuevos elementos por medio de módulos personalizados. [6]

## 2.8. ARAP

ARAP (Ant Routing Anonymity Protocol) es un simulador para protocolo de enrutamiento que provee Anonimato basado en algoritmos de optimización de colonias de hormigas artificiales, el cual se desarrolló en [10] como un módulo de ns-3 bajo la licencia GNU GPLv2.

### 2.8.1. Protocolo simulado por ARAP

Como se mencionó antes el simulador realizado en [10] está diseñado en torno al protocolo de enrutamiento a nivel de capa aplicación propuesto en [11]. El protocolo está basado en la técnica para proveer anonimato llamada Onion Routing (enrutamiento cebolla), la cual consiste en cifrar los mensajes en capas, los cuales se envían a través de varios nodos, que van eliminando una capa cada uno, hasta que el mensaje llega al nodo destino y al éste remover la última capa puede ver el mensaje original que se envió desde el origen.

Existen ya protocolos y programas que aprovechan el Onion Routing, como por ejemplo la red anónima TOR, en ellos los mensajes son enviados a nodos elegidos de

manera aleatoria para llegar a su destino. Estos saltos aleatorios tienen un impacto negativo en la latencia de las comunicaciones en una red anónima.

En [11] se propone un protocolo que aprovecha el algoritmo de optimización por colonia de hormigas (ACO) para recopilar y mantener información de la red que permita seleccionar caminos óptimos que reducen la latencia mientras se mantiene el anonimato. La información de la red se guarda en tablas de probabilidad de enrutamiento en cada nodo. Además del anonimato que provee el Onion Routing, los mensajes son enviados por hormigas de carga, indistinguibles de las hormigas exploradoras que recopilan la información de la red, lo que hace las comunicaciones más anónimas.

## 2.9. Qt Creator y C++

Qt Creator es un entorno de desarrollo integrado (IDE) multiplataforma para crear aplicaciones de C++ y QML para múltiples plataformas de escritorio, embebidas y móviles. Qt Creator fué desarrollado para el framework multiplataforma Qt, generalmente utilizado para desarrollar programas que utilicen interfaz de usuario, pero que también es capaz de crear programas para la consola de comandos. Posee un editor de código y herramientas para diseñar, codificar, probar, implementar y mantener software. Facilita la creación de GUIs por medio de su interfaz para diseñar interfaces de usuario de arrastrar y soltar (*drag and drop*) que se puede ver en la figura 2.1, entre otras herramientas especializadas. [4]

Se decidió utilizar Qt Creator por sus herramientas para la creación de GUIs y su extensa documentación y popularidad, además de tener experiencias previas con el entorno de desarrollo.

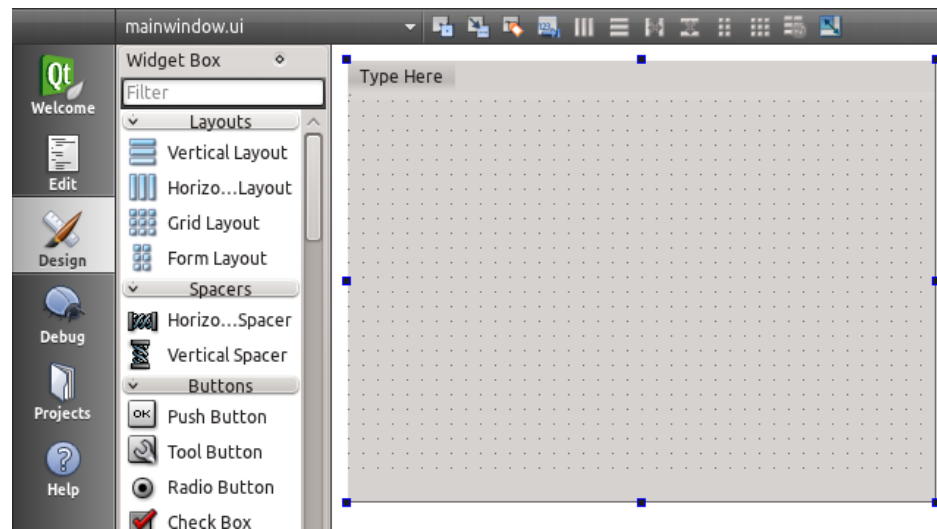


Figura 2.1: Interfaz de creación de interfaces de QtCreator

## 2.10. C++

C++ es un lenguaje de programación orientado a objetos de uso general. Se le considera un lenguaje de nivel intermedio, ya que posee características de lenguajes de alto y bajo nivel. Dichas características lo convierten en un lenguaje flexible y eficiente en el uso de recursos.

# Capítulo 3

## Interfaz de usuario para el simulador ARAP

En este trabajo se desarrolló una interfaz de usuario que envuelve la funcionalidad del simulador de protocolo de enrutamiento ARAP, haciéndola más accesible y eficiente. Además agrega nuevas funcionalidades, que facilitan el análisis de resultados de las simulaciones realizadas en el simulador ARAP e ilustran el funcionamiento del protocolo de enrutamiento. Este software está desarrollado bajo la licencia GNU GPLv2.

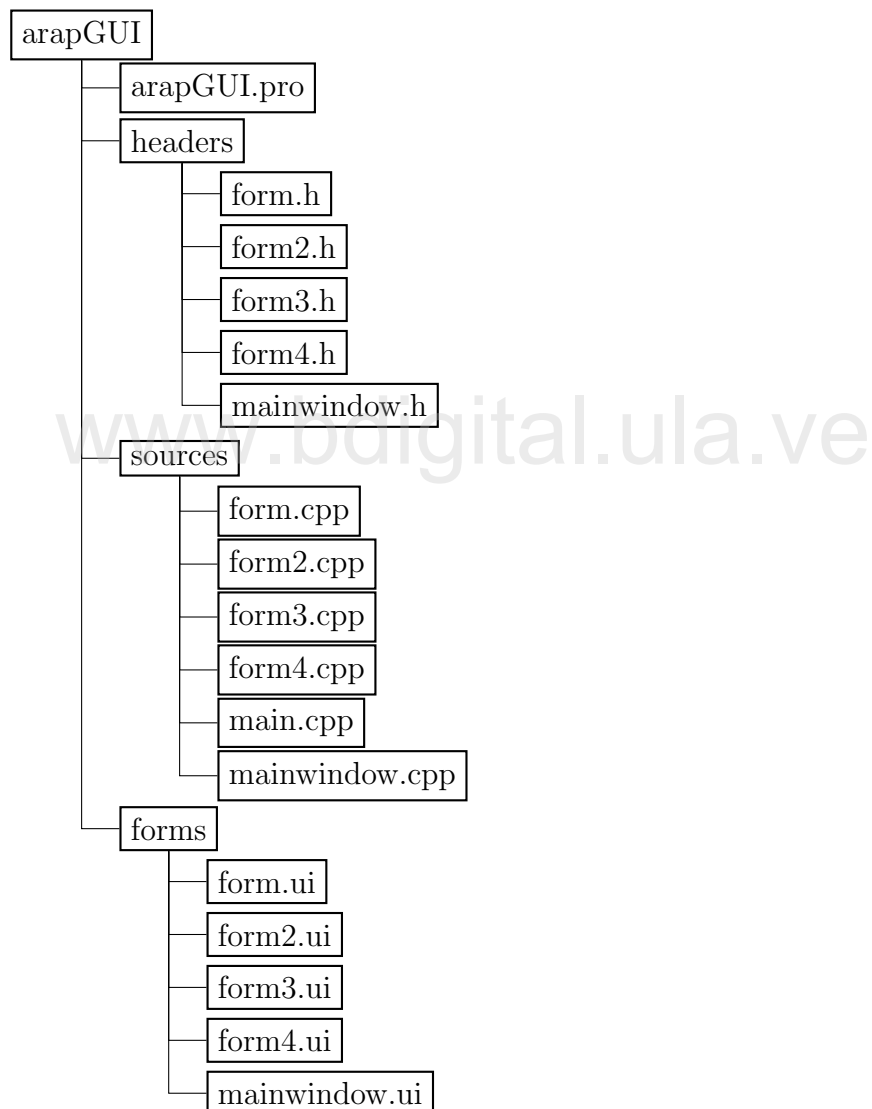
La interfaz desarrollada se puede dividir en 2 componentes que pueden ser utilizados separadamente o en conjunto:

- Una interfaz de usuario para el simulador como tal, llamada ARAPGUI que abarca el ingreso y manejo de parámetros, la ejecución del simulador con dichos parámetros, la organización de los resultados en directorios, el despliegue de los resultados y un ejemplo animado del funcionamiento del protocolo de enrutamiento en la simulación.
- Una interfaz llamada ARAP Compare para comparar y estudiar diferentes experimentos, ordenarlos por parámetros y acceder a los documentos organizados si se desean examinar detalladamente.

## 3.1. Estructura de la Interfaz de Usuario del Simulador ARAP

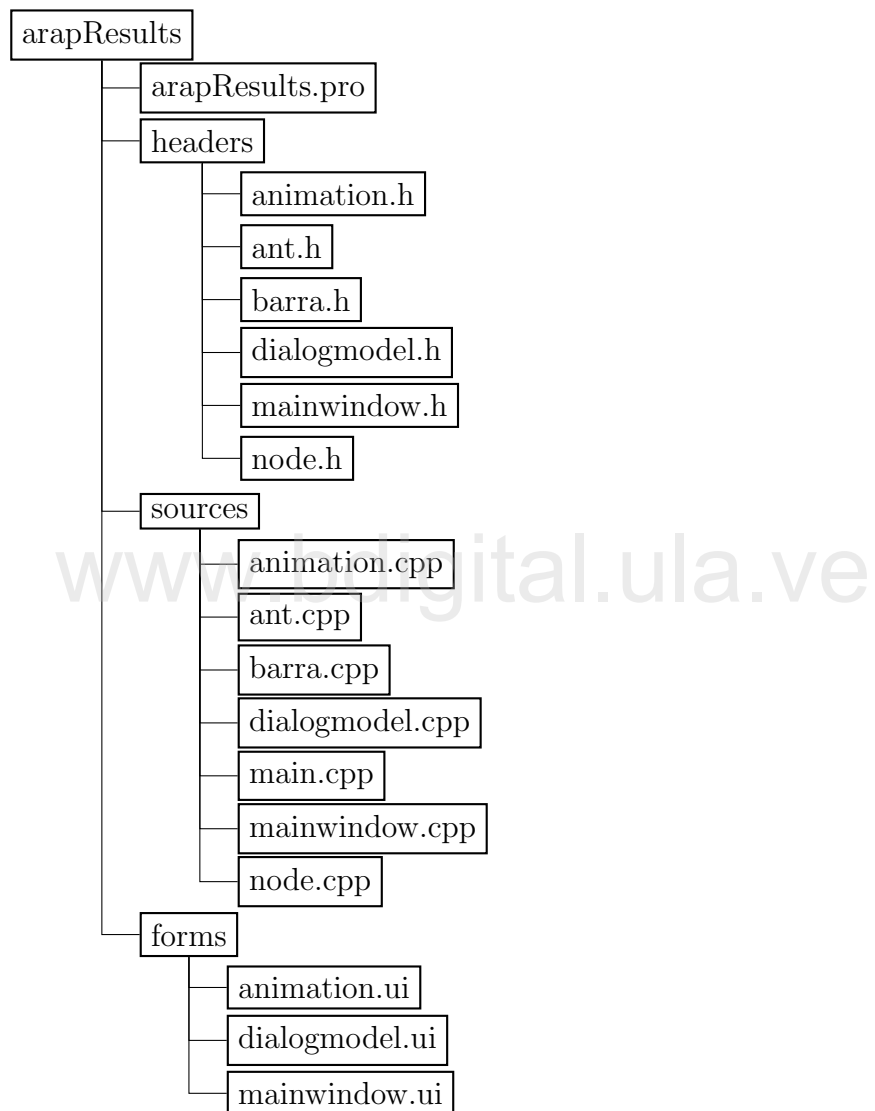
### 3.1.1. Estructura de ARAPGUI

A continuación se muestra la estructura del proyecto ARAPGUI, la clase mainwindow corresponde a la ventana principal de la interfaz de usuario.



### 3.1.2. Estructura de ARAP Results

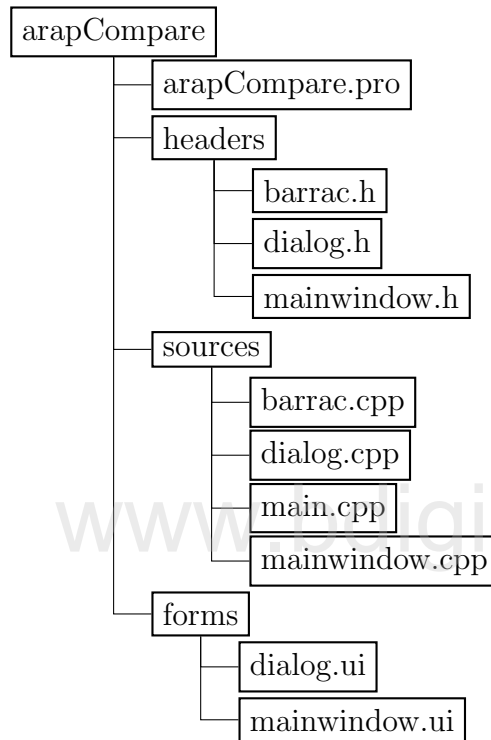
A continuación se muestra la estructura del proyecto ARAP Results, la clase *mainwindow* corresponde a la ventana principal de la interfaz de usuario. La clase *animation* muestra la animación haciendo uso de las clases *ant* y *node*.



### 3.1.3. Estructura de ARAP Compare

### 3.1.4. Estructura de ARAP Results

A continuación se muestra la estructura del proyecto ARAP Compare, la clase *mainwindow* corresponde a la ventana principal de la interfaz de usuario.



## 3.2. ARAPGUI

Antes de comenzar el diseño de la interfaz de usuario para ingresar parámetros al simulador se examinaron cuidadosamente las características de los parámetros y los valores posibles para los mismos. De esta forma se determinaron formas intuitivas y eficientes de ingresar parámetros haciendo uso de los widgets disponibles en Qt Creator o creando nuevos widgets cuando fuera necesario.

El simulador ARAP fue diseñado de manera tal que el usuario no necesita modificar el código fuente para realizar simulaciones, a menos que desee agregar nuevas

funcionalidades o comportamientos. Los parámetros son ingresados a través de un archivo de texto como el de la figura 3.1 con los siguiente formatos:

M1. <nombre de parámetro> <valor>

M2. <nombre de parámetro> <rango mínimo> <rango máximo> <valor>

El primer caso se usa para variables que tienen un solo valor en la simulación, como el tiempo de simulación o el número de nodos, el segundo caso se utiliza cuando la variable requiere un valor por cada nodo. En estos casos se pueden escribir múltiples líneas asignando valores diferentes a rangos de nodos.

Otros casos especiales son los parámetros que corresponden a variables aleatorias, se reemplaza <valor> con el nombre de la distribución de probabilidad a utilizar, seguido de los valores que recibe ésta, que varían entre distribuciones y pueden ser una cantidad diferente de parámetros, por ejemplo una distribución constante recibe un solo valor, mientras que una distribución normal necesita tres: media, varianza y límite. Así pues se consideraron dos escenarios más:

M3. <nombre de parámetro> <nombre distribución> <parámetros distribución>

M4. <nombre de parámetro> <rango mínimo> <rango máximo> <nombre distribución> <parámetros distribución>

Y por último existe un caso especial donde un parámetro que puede requerir más valores dependiendo de la configuración que se use, por lo que se consideró un caso aparte:

M5. <nombre de parámetro> <nombre PathManager> <parámetros>

---

```

nodos 4
semilla 1
usar-trazas 0
saltos 3
puerto 200
tamano-hormiga 512
intervalo-exploradoras 450
data-rate 0 0 1
app-tiempo-iniciar 0 0 1
tiempo-detener-simulador 1000
delay-enlaces-intervalo 1
delay-enlaces-dist UniformRandomVariable 0 50
delay-computo-dist 0 0 UniformRandomVariable 0 50
delay-computo-incremento 0 0 0
habilitar-exploradoras 0
tamano-cola 0
tamano-segmento 512
path-manager SmartPathManagerDefault
imprimir-tablas-intervalo 1000
hormigacarga-tiempo-dist 0 0 UniformRandomVariable 0 50
hormigacarga-cantidad-dist 0 0 UniformRandomVariable 0 50
hormigacarga-destino-dist 0 0 UniformRandomVariable 0 50

```

Figura 3.1: Archivo de entrada de parámetros del simulador ARAP

Se clasificaron por caso y tipo de dato los 23 parámetros usados en la simulación, para buscar la mejor manera de representar e ingresar cada uno al simulador. A continuación se presentan los 23 parámetros seguidos del tipo de dato (entero, booleano, etc) y el caso (M1, M2, M3, M4 o M5), la descripción de las mismas se encuentra en [cadavid].

1. Cantidad Nodos (*entero*  $\geq 4$ ) (M1)
2. Tamaño Hormigas (*entero*  $\geq 128$ ) (M1)
3. Habilitar Exploradoras (booleano) (M1)
4. Habilitar Trazas (booleano) (M1)
5. Intervalo de envío de exploradoras (*real*  $> 0$ ) (M1)
6. Saltos de las hormigas (*entero*  $\in [2, CantidadNodos - 1]$ ) (M1)
7. Distribución de retardo en los enlaces (Distribución) (M3)

8. Intervalo de cambio en los enlaces (*real* > 0) (M1)
9. Tamaño máximo de segmento (*entero* >= 0) (M1)
10. Puerto (*entero* ∈ [1, 65535]) (M1)
11. Intervalo de Impresión de tablas de probabilidad (*real* > 0) (M1)
12. Tamaño de cola (*entero* >= 0) (M1)
13. Número de ejecución (*entero* >= 0) (M1)
14. Semilla (*entero* > 0) (M1)
15. Tiempo de Simulación (*real* > 0) (M1)
16. Tiempo de Inicio de Aplicaciones (*real* >= 1) (M2)
17. Factor de incremento de tiempo de cómputo (*real* ∈ [0, 1]) (M2)
18. Data-Rate (*entero* > 0) (M2)
19. Distribución de tiempo de cómputo (Distribución) (M4)
20. Cantidad de hormigas de carga por envío (Distribución) (M4)
21. Distribución para destinos de hormigas de carga (Distribución) (M4)
22. Distribución para tiempos para envíos de hormigas de carga (Distribución) (M4)
23. Path Manager (PathManager y parámetros) (M5)

Para crear la interfaz de usuario de la entrada se estableció que es necesario que cada parámetro tenga una etiqueta que identifique el parámetro a ingresar, y un widget de entrada del tipo de dato requerido para ese parámetro. En los casos M2 y M4 además es necesario poder agregar o quitar rangos para asignar valores al mismo parámetro en diferentes nodos. En los casos M3 y M4 además se necesita crear un widget personalizado para recibir distribuciones y los parámetros que éstas necesitan. El caso

M5 debe ser capaz de recibir el nombre de la especialización de ArapPathManager a utilizar, así como los parámetros que ésta necesita.

El espacio en ventana que ocupan 23 pares de etiquetas y widgets, con un tamaño aceptable para monitores modernos, es demasiado grande para una sola ventana, por lo que se pensaron dos (2) posibles soluciones: una interfaz de usuario tipo wizard que guíe al usuario por pasos para realizar el ingreso de parámetros en diferentes ventanas de diálogos, o una interfaz de pestañas en la que el usuario pueda ver grupos de parámetros en diferentes áreas dentro de una misma ventana navegando a través de las pestañas.

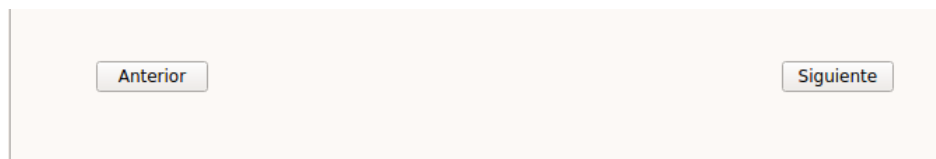


Figura 3.2: Botones de navegación: Anterior y Siguiente

Se decidió que la GUI tipo wizard era muy restrictiva en su navegación así que se implementó un sistema de pestañas como el de la figura 3.3, agregando además botones de navegación entre pestañas que se muestran en la figura 3.2, parecidos a los botones de progreso de una GUI tipo wizard para que los usuarios puedan realizar el ingreso de parámetros en orden si desean.

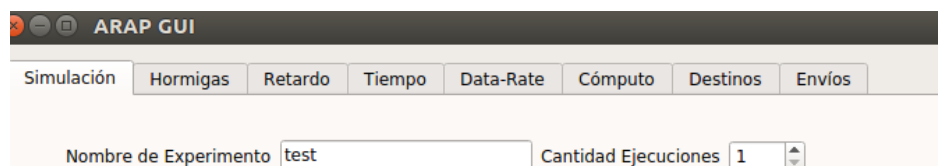


Figura 3.3: Pestañas de navegación

En el caso M1, para aquellos parámetros con el formato <nombre de parámetro> <valor>, se identificaron 3 tipos de datos, a los cuales se les asignó un widget apropiado para su ingreso.

Para los valores enteros existe un widget llamado Spin Box, mostrado en la parte inferior de la figura 3.4, que permite ingresar el valor con el teclado o usar el puntero

del mouse para aumentar o disminuir su valor en una cantidad determinada. Además se le puede asignar valores mínimo y máximo, así como el valor predeterminado que se le da al parámetro.

Asimismo para los parámetros que reciben valores reales se usó un widget que puede verse en la parte superior de la figura 3.4, llamado Double Spin Box, que funciona de manera similar a un Spin Box pero con valores del tipo double. En este widget además de ser posible modificar los valores máximo, mínimo y predeterminado, es posible seleccionar la cantidad de decimales a utilizar.

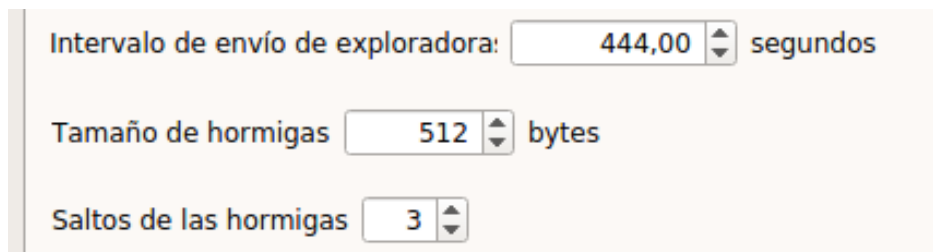


Figura 3.4: Widgets para recibir parámetros enteros y reales

Para los parámetros booleanos, es decir, los que son verdaderos o falsos, existe un widget sencillo llamado Check Box que, como puede verse en la Figura 3.5, consiste en una simple casilla que puede ser marcada o desmarcada con un simple click, representando así el estado de la variable.

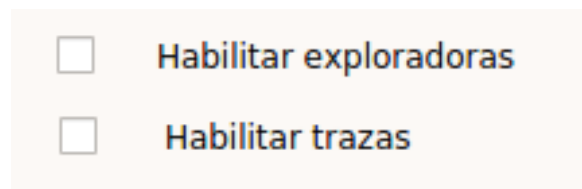


Figura 3.5: Widgets para recibir parámetros de tipo booleano

Como se mencionó antes, los widgets Spin Box y Double Spin Box pueden limitar los valores que se pueden ingresar, modificando los valores mínimo y máximo. Sin embargo esto no cubre casos como el del parámetro 6 (Saltos de las hormigas), en el que se necesita que el valor sea menor a otra variable (cantidad de nodos) menos 1.

Esta restricción se codificó como un evento que se ejecuta al terminar de editar el valor del parámetro 6. El valor ingresado se compara el valor del parámetro 1 y en caso de ser inválido se le muestra al usuario un diálogo de advertencia.

En cuanto a los casos M3 y M4, se dijo anteriormente que éstos reciben distribuciones de probabilidad, por lo que es necesario crear un widget personalizado para recibir la distribución con sus parámetros. Como cada distribución recibe parámetros diferentes, y en diferentes cantidades, se creó un widget que cambia de manera dinámica al seleccionar la distribución de probabilidad. El widget personalizado está conformado por otros widgets interconectados para ajustarse a las necesidades de los parámetros. Primero se utiliza un widget llamado *Combo Box* que permite seleccionar una entre las cinco distribuciones posibles, al ser seleccionada cualquier opción, se dispara una función que cambia los widgets que se muestran dentro de un *StackedWidget* que, de manera similar al sistema de pestañas, muestra un solo conjunto de widgets seleccionado. En la figura 3.6 se puede ver un widget listo para recibir una distribución Uniforme:

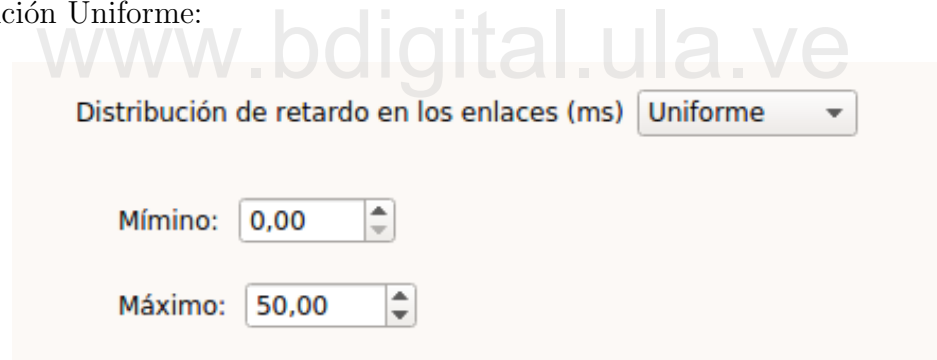
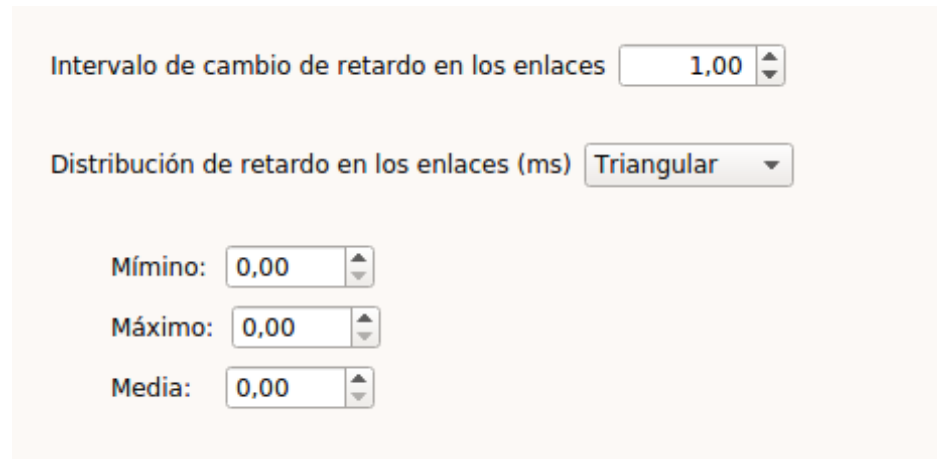
The image shows a screenshot of a web-based GUI. At the top, there is a label "Distribución de retardo en los enlaces (ms)" followed by a dropdown menu currently set to "Uniforme". Below this, there are two input fields: "Mínimo:" with a value of "0,00" and "Máximo:" with a value of "50,00". Each input field has a small up/down arrow icon to its right. A large, semi-transparent watermark "www.bdigital.ula.ve" is overlaid across the center of the image.

Figura 3.6: Widget para recibir parámetros de distribuciones de probabilidad

Mientras que en la figura 3.7 se observa que el mismo widget cambió al ser modificado el valor del *Combo Box* para recibir una distribución Triangular:



Intervalo de cambio de retardo en los enlaces

Distribución de retardo en los enlaces (ms)

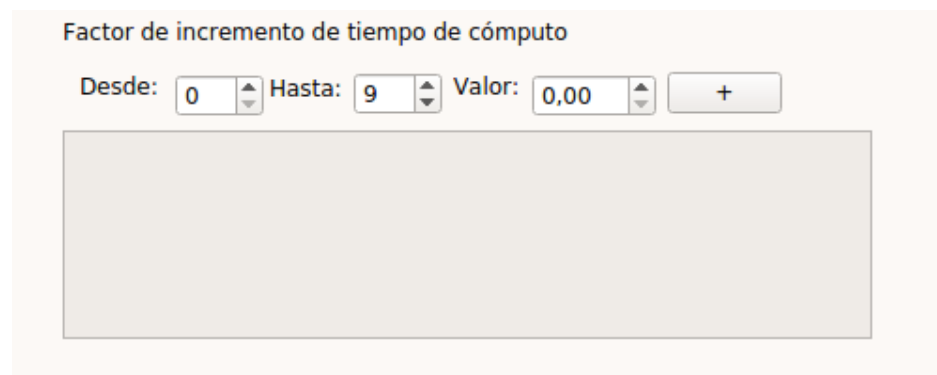
Mínimo:

Máximo:

Media:

Figura 3.7: Párametros widget de distribución de probabilidad al cambiar de distribución por la triangular.

Los casos M2 y M4 deben ofrecer al usuario la posibilidad de crear nuevos rangos de nodos para asignar valores de parámetro diferentes a cada nodo. Para lograr esto se tuvieron que crear widgets personalizados, y copiar dicho widget en una lista en la ventana principal al accionar un botón con el símbolo más (+). Cada widget personalizado, para lo que se utilizaron los nombres form, form2, form3 y form4, se desarrolló como una clase separada debido a que cada uno necesitaba recibir tipos de dato diferentes con restricciones diferentes. Cada widget cuenta también con un botón con el símbolo menos (-) que borra y retira dicho widget de la lista de rangos para esa variable. En la figura 3.8 se observa un widget antes de agregar nuevos rangos de valores:



Factor de incremento de tiempo de cómputo

Desde:  Hasta:  Valor:

Figura 3.8: Widget para recibir parámetros para rangos de nodos

Y en la figura 3.9 se puede ver el resultado de presionar dos (2) veces el botón (+) para agregar dos (2) rangos:

Factor de incremento de tiempo de cómputo

Desde: 0 Hasta: 5 Valor: 0,03 +

Desde: 6 Hasta: 6 Valor: 0,07 -

Desde: 7 Hasta: 10 Valor: 0,12 -

Figura 3.9: Widget para recibir parámetros para rangos de nodos con tres rangos

Finalmente el caso C5 necesita recibir el nombre de la especialización de ArapPathManager, seguida de los parámetros que pueda o no recibir. Dada la naturaleza inconsistente de este parámetro, se buscó una solución flexible y se utilizó el widget Line Edit que recibe una cadena de caracteres, donde el usuario puede ingresar el nombre y los parámetros necesarios del PathManager.

PathManager

SmartPathManagerDefault

Figura 3.10: Widget para recibir la especialización de PathManager y sus parámetros

Además de los veintitrés (23) parámetros se agregaron dos (2) campos más que se pueden ver en la figura 3.11. El primero es el nombre que se le dará al experimento, anteriormente los resultados de todas las pruebas se guardaban de manera desorganizada en una misma carpeta, ahora se guardan las ejecuciones que comparten parámetros en un directorio con el nombre elegido. El segundo campo es el número de ejecuciones que se desea realizar con estos parámetros.

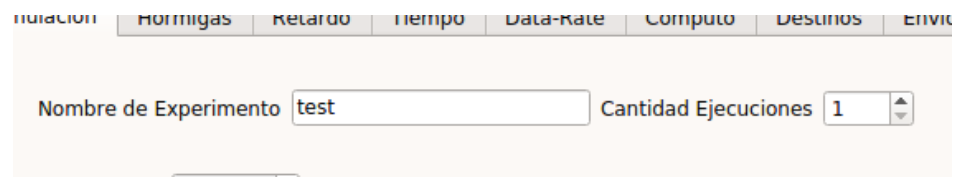
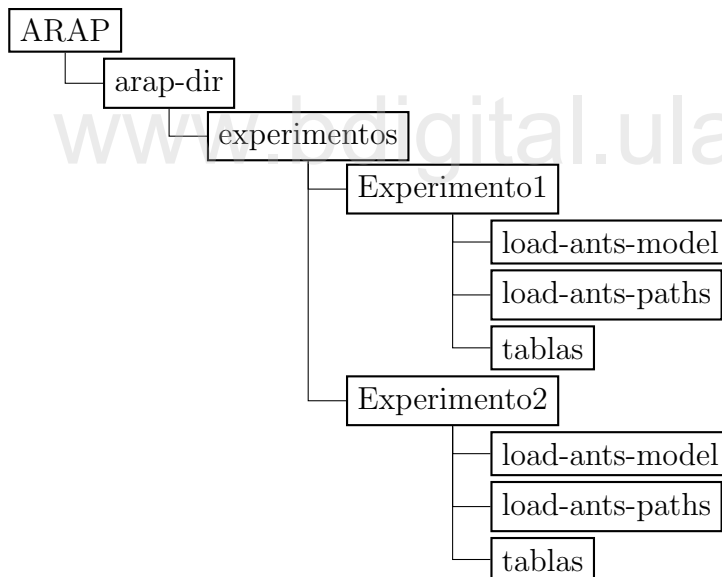


Figura 3.11: Nuevos parámetros, nombre de experimento y número de ejecuciones

Una vez ingresados todos los parámetros el usuario debe presionar el botón *Correr Simulación* en la última pestaña para iniciar el conjunto de simulaciones con los parámetros ingresados. ARAPGUI lee todos los parámetros de sus widgets y los escribe entonces en un archivo de texto con el nombre de experimento dado, que luego pasa como parámetro al simulador ARAP. Al finalizar la simulación ARAP se llama a un programa de consola que crea y organiza los archivos en el directorio del experimento con la siguiente estructura:



En el Apéndice A se encuentran las capturas completas de cada panel de la interfaz de usuario de ingreso de parámetros ARAPGUI.

### 3.2.1. ARAPResults

Habiendo culminado la simulación y la organización de los archivos, el simulador llama a otro módulo de la GUI, llamado ARAP Results. En ARAP Results se muestran

los resultados de las simulaciones que el usuario realiza. Un histograma compara los tiempos de respuesta promedios de las diferentes ejecuciones dentro del experimento. Si el usuario presiona una barra del histograma, podrá ver el modelo estadístico de la ejecución que representa en una tabla que se despliega en una ventana de diálogo. El usuario puede acceder desde esta ventana de resultados a la interfaz de comparación de resultados con el botón *Comparar Resultados*.

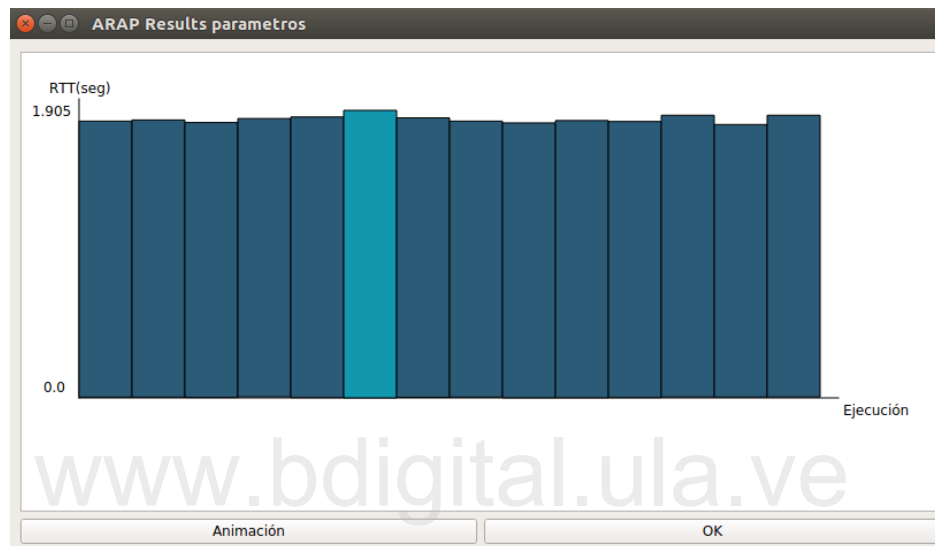
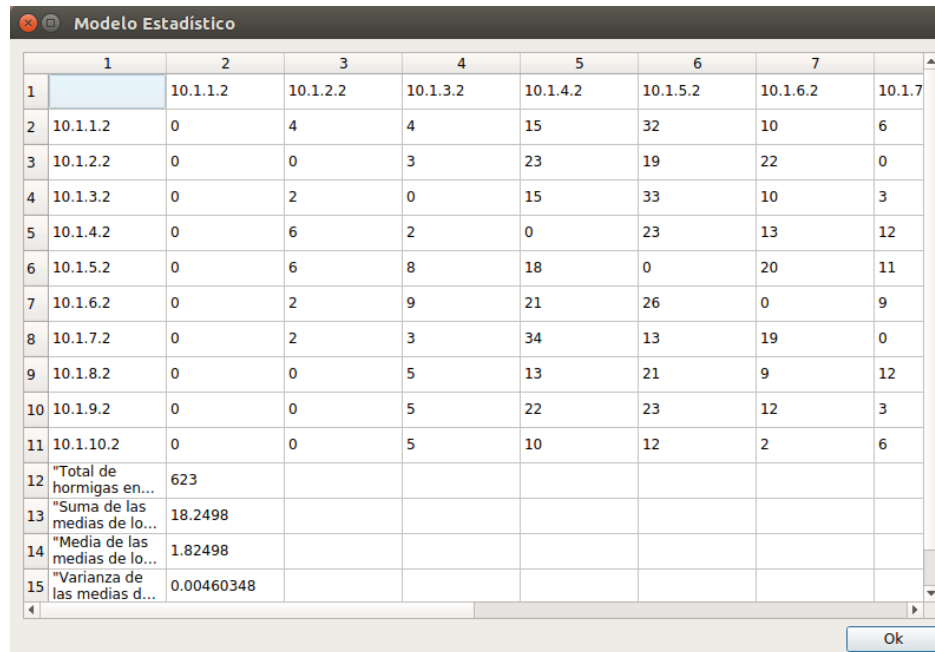


Figura 3.12: Interfaz gráfica de resultados ARAP Results



	1	2	3	4	5	6	7	
1		10.1.1.2	10.1.2.2	10.1.3.2	10.1.4.2	10.1.5.2	10.1.6.2	10.1.7
2	10.1.1.2	0	4	4	15	32	10	6
3	10.1.2.2	0	0	3	23	19	22	0
4	10.1.3.2	0	2	0	15	33	10	3
5	10.1.4.2	0	6	2	0	23	13	12
6	10.1.5.2	0	6	8	18	0	20	11
7	10.1.6.2	0	2	9	21	26	0	9
8	10.1.7.2	0	2	3	34	13	19	0
9	10.1.8.2	0	0	5	13	21	9	12
10	10.1.9.2	0	0	5	22	23	12	3
11	10.1.10.2	0	0	5	10	12	2	6
12	"Total de hormigas en..."	623						
13	"Suma de las medias de lo..."	18.2498						
14	"Media de las medias de lo..."	1.82498						
15	"Varianza de las medias d..."	0.00460348						

Figura 3.13: Modelo estadístico de ejecución seleccionada en ARAP Results

### 3.2.2. Animación

También en ARAP Results se encuentra un botón de *Animación*, el cual despliega en una ventana, una animación de la ruta que toma una hormiga de carga para ir desde un nodo a otro y su retorno, esta animación utiliza valores reales de una de las ejecuciones realizadas y su objetivo es ilustrar el funcionamiento del protocolo.

La animación lee los valores del archivo de caminos de una ejecución elegida al azar. Selecciona la última hormiga como ejemplo para la animación y crea la ruta que recorre. Además lee la última tabla de probabilidad impresa por la misma ejecución para pintar los caminos posibles de colores que representan la probabilidad de que la hormiga pase por cada nodo.

La animación se controla en una `QGraphicsScene`, que es un contenedor de objetos gráficos del tipo `QGraphicsItems` donde se manejan las posiciones y propiedades de los objetos a dibujar. La `QGraphicsScene` se despliega entonces en una `QGraphicsView`, una región dibujable de QT que permite el renderizado de figuras

bidimensionales y mapas de bits de forma dinámica.

Antes de comenzar la animación se dibujan los nodos, representados como cuadrados y se posicionan aleatoriamente en la `QGraphicsScene`, de manera tal que no se superpongan. Seguidamente se crea un objeto hormiga, representado por un círculo, y se posiciona en el nodo origen del mensaje.

Al iniciar la animación la hormiga se desplaza del nodo inicial al nodo seleccionado para su primer salto, luego al segundo y así sucesivamente hasta llegar al nodo destino. En este punto se dibuja la ruta tomada y la hormiga regresa al nodo origen a través de dicha ruta, representando el comportamiento real de una hormiga de carga.

En la figura 3.14 se muestra una captura de la animación tomada mientras la hormiga se dirige del nodo inicial a uno intermedio, en la figura 3.15 se muestra una captura de la hormiga regresando al nodo de origen por la misma ruta que usó de ida:

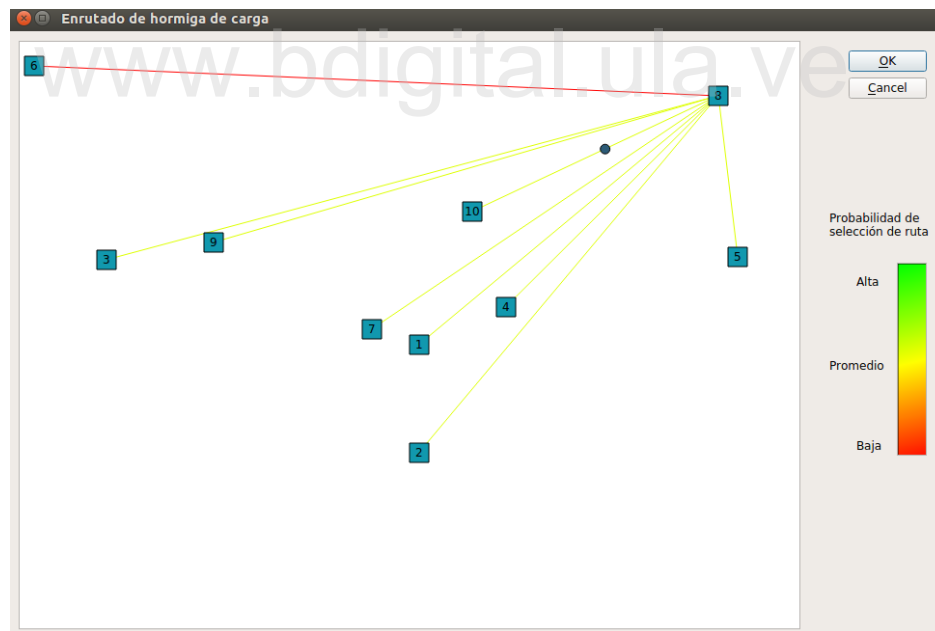


Figura 3.14: Comienzo de animación, la hormiga parte del nodo 8 al nodo 10

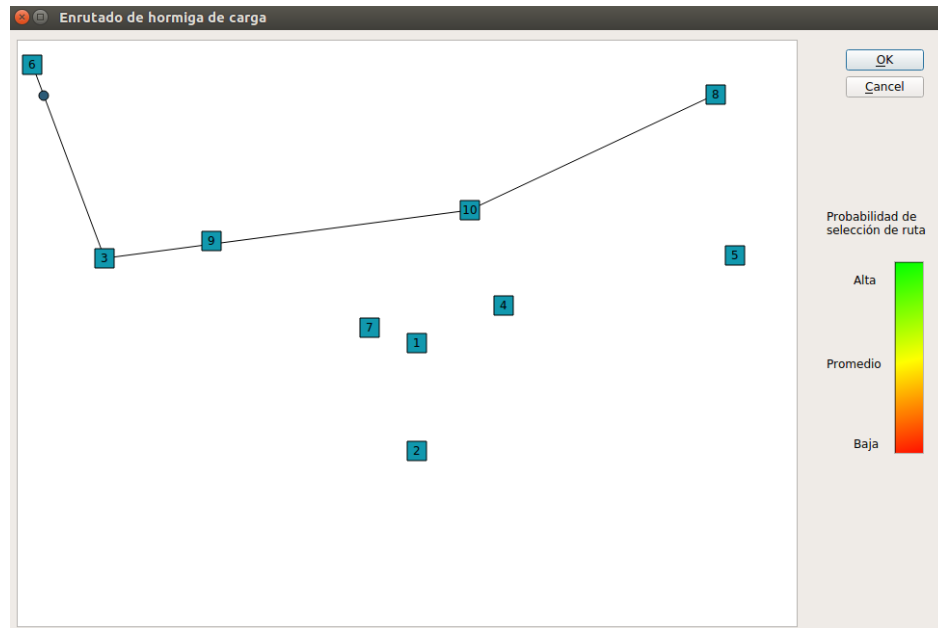


Figura 3.15: Animación, hormiga regresa por la ruta seleccionada

En cada nodo por el que pasa la hormiga, excepto el nodo final, se dibujan arcos desde el nodo actual en el que se encuentra la hormiga hasta cada nodo candidato a ser usado como nodo intermedio. Los colores de los arcos representan la probabilidad de que la hormiga elija ese nodo como próximo nodo intermedio. Utilizando el sistema de colores RGB se eligieron los siguientes colores: Verde (0,255,0) para representar una probabilidad por encima del promedio de que un nodo sea elegido como nodo intermedio en una ruta, Amarillo(255,255,0) para una probabilidad promedio y Rojo (255,0,0) para una probabilidad nula. El rango de colores entre cada uno de éstos representa el rango de probabilidades que puede tener un nodo. Se agregó una leyenda explicando el significado de los colores.

Se utilizó el siguiente algoritmo para determinar los colores de los arcos:

Dado que:

Verde (0,255,0) -> Amarillo (255,255,0) -> Rojo (255,0,0)

Representan:

1.0 (100 % prob.) ->  $1/(n - 1)$  (Prob. Promedio) -> 0.0 (Prob. Nula)

Entonces evaluamos la probabilidad:

- Si se encuentra entre 1 y  $1/(n-1)$ , se mantienen estáticos los últimos dos valores  $(X, 255, 0)$  entonces utilizamos la siguiente ecuación para determinar la  $X$  y el color:  $X = 255 * (1 - Prob)$
- Si se encuentra entre  $1/(n-1)$  y 0, se mantienen estáticos el primero y el último valor  $(255, X, 0)$  entonces se utiliza la siguiente ecuación para determinar la  $X$  y el color:  $X = 255 * (Prob)$

### 3.3. ARAP Compare

Al igual que en la primera mitad de este trabajo se estudiaron los parámetros de entrada a la simulación y se creó una interfaz utilizando esa información como base, en la segunda parte se estudiaron los resultados, sus características y presentaciones. Se estudió también cuáles resultados mostrar para ayudar al usuario a estudiar el rendimiento del protocolo de enrutamiento y se diseñó una interfaz para analizar estos resultados y compararlos con los resultados de múltiples experimentos.

El Simulador ARAP almacena los resultados de cada ejecución en tres (3) tipos de archivos:

- Un modelo estadístico de las hormigas de carga para cada nodo.
- Tablas de probabilidad de cada nodo en tiempos determinados de la simulación.
- Caminos generados para las hormigas de carga.

Se determinó que el factor más importante es el tiempo de ida y vuelta (RTT, por el inglés round-trip time) dado que el objetivo de la creación del protocolo a estudiar es reducir la latencia en una red anónima. Por lo tanto se le dio prioridad a la comparación del RTT entre diferentes experimentos por medio de histogramas.

El programa ARAPCompare puede utilizarse de manera independiente a la interfaz de usuario ARAPGUI, pueden compararse todos los experimentos que se han

realizado anteriormente con ARAPGUI y observar las relaciones que pueden haber entre sus tiempos de respuesta promedio y diferentes parámetros. Se diferencia de la interfaz de despliegue de resultados ARAPResults en que ésta muestra los tiempos de respuesta promedios de diferentes experimentos con parámetros distintos, mientras que el ARAPResults muestra los tiempos promedios de las ejecuciones dentro de un experimento, usando los mismos parámetros. Además, el ARAPCompare puede ordenar los resultados por parámetros.

Inicialmente la interfaz de usuario muestra un histograma con todos los experimentos que están guardados en el directorio Experiments, dentro de arap-dir, el directorio donde se encuentra el simulador. También desde el inicio del programa se muestra una lista de los nombres de experimentos que se muestran en el histograma. El orden de la lista es el mismo orden de las barras en el histograma de izquierda a derecha.

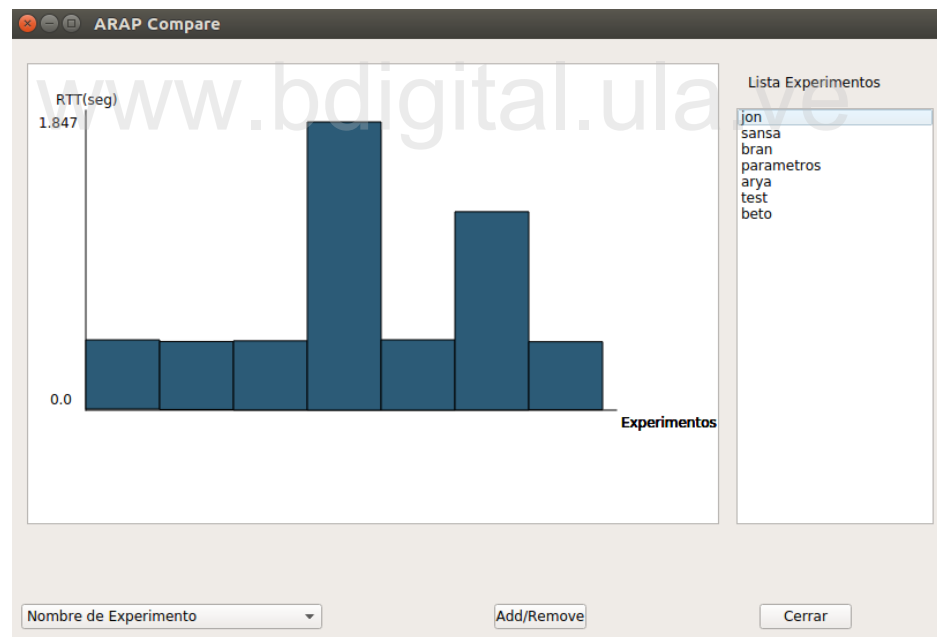


Figura 3.16: Interfaz de comparación de experimentos ARAP Compare

En la figura 3.16 se observa la interfaz gráfica de ARAP Compare con siete (7) experimentos. El usuario cuenta con una ComboBox para cambiar el orden en el que se muestran las barras en el histograma, el ComboBox tiene una lista de algunos

parámetros que, según se determinó estudiando cada uno, pudieran tener incidencia en el tiempo promedio de respuesta de los experimentos. En la figura 3.17 se puede ver el resultado del cambio de orden del histograma.

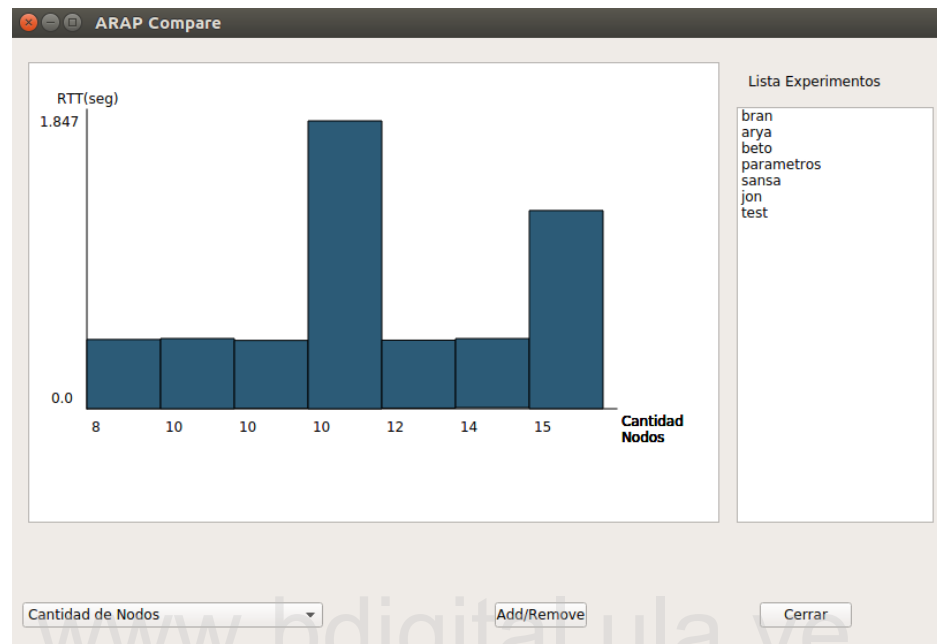


Figura 3.17: ARAP Compare, ordenado por cantidad de nodos

Si el usuario desea agregar o remover experimentos del histograma y la lista, puede presionar el botón *Add/Remove* el cual crea una ventana de diálogo en la que puede seleccionar cuáles experimentos desea comparar usando botones para agregar o remover experimentos. Se muestran dos listas, una para los experimentos que se van a mostrar en el histograma y otra para los que no. Los botones >> y << agregan y remueven respectivamente de la lista de experimentos a comparar. Además hay dos botones para agregar o remover rápidamente todos los elementos de la lista de experimentos a comparar. La ventana de diálogo para agregar o remover experimentos del histograma se pueden ver en las figuras 3.18 y 3.20, los resultados de modificaciones hechas en estas figuras pueden verse en las figuras 3.19 y 3.21

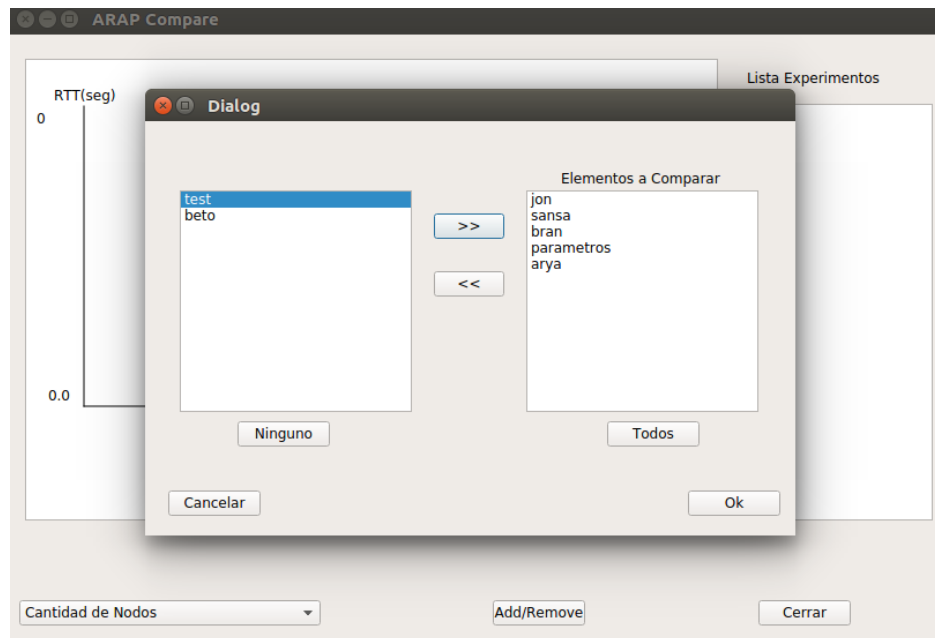


Figura 3.18: Ventana de diálogo de selección de experimentos de ARAP Compare

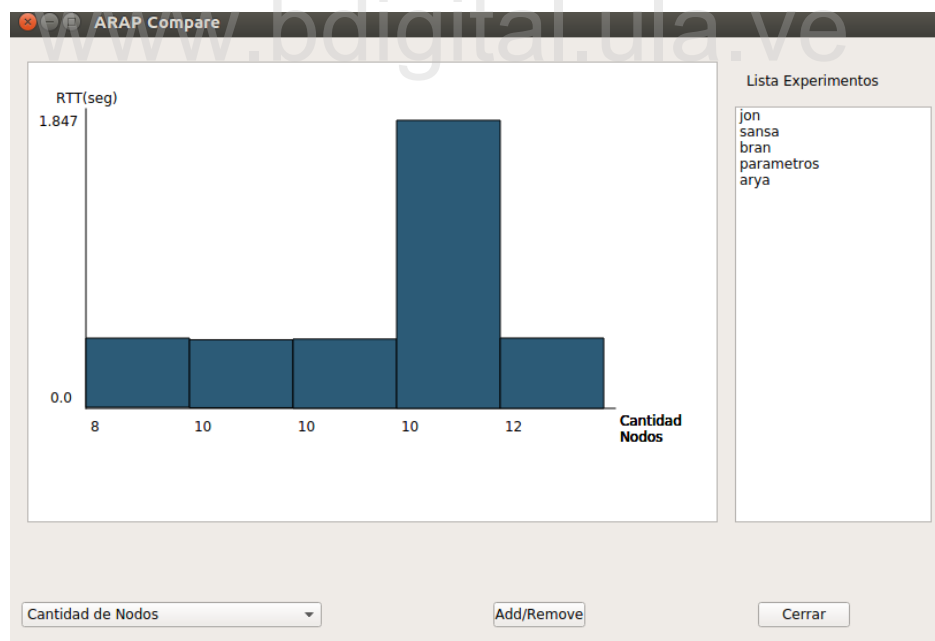


Figura 3.19: ARAP Compare, dos experimentos removidos

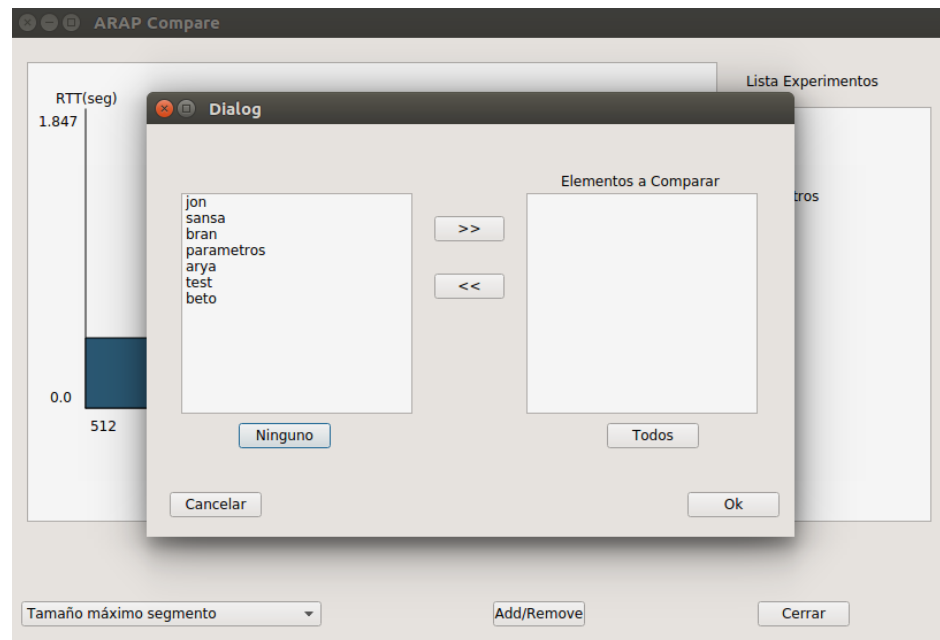


Figura 3.20: Botón ninguno ventana de selección de experimentos de ARAP Compare

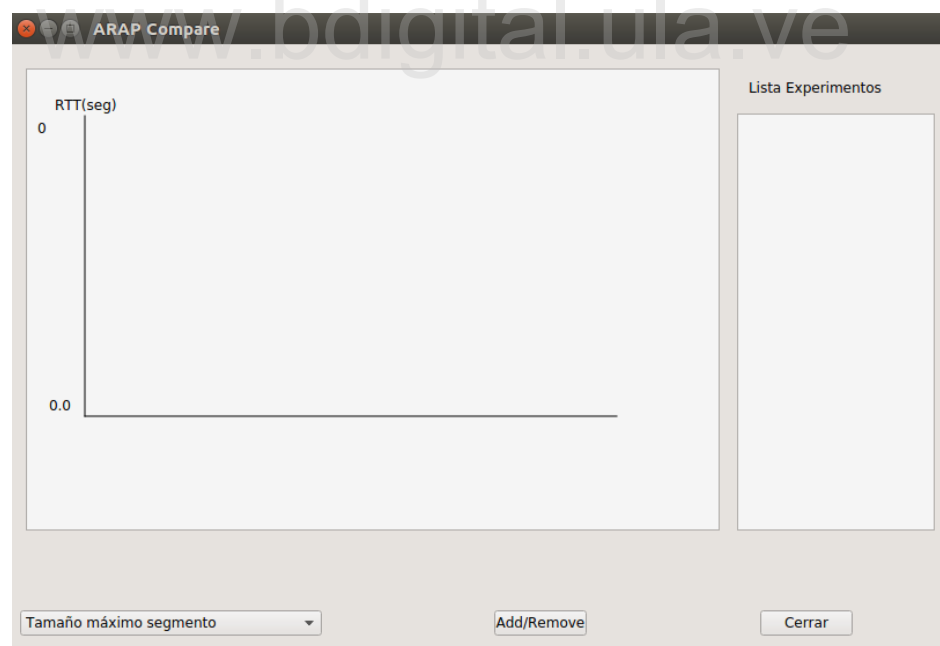


Figura 3.21: ARAP Compare con todos los experimentos removidos

### 3.4. Arquitectura

A lo largo de este documento se han mencionado los diferentes componentes y cómo se relacionan, en las figuras 3.22, 3.23 y 3.24 se muestran los diagramas de clases de los componentes del sistema y en la figura 3.25 se muestra la relación de éstos componentes en un diagrama de flujo del uso de la aplicación.

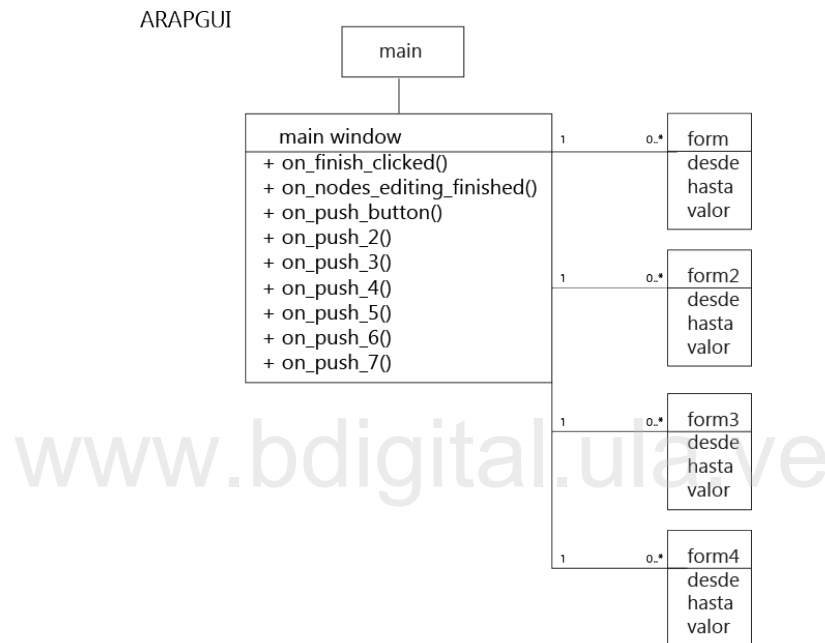


Figura 3.22: Diagrama de clases ARAPGUI

ARAP Results

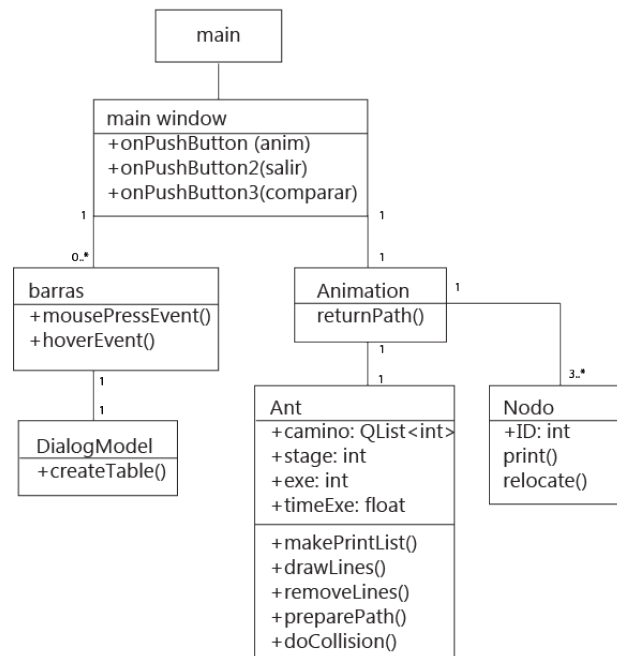


Figura 3.23: Diagrama de clases ARAP Results

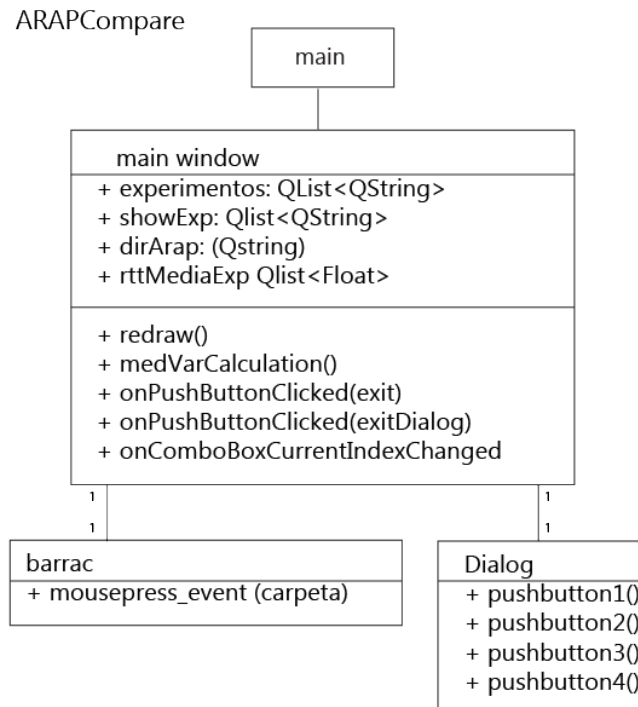


Figura 3.24: Diagrama de clases ARAP Compare

Estructura de relaciones entre programas

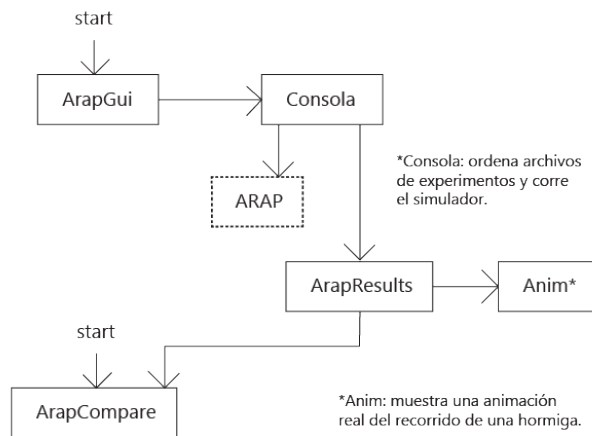


Figura 3.25: Diagrama de relaciones entre clases

# Capítulo 4

## Pruebas manuales

A lo largo del desarrollo de la interfaz gráfica de usuario se ejecutaron pruebas de caja blanca para la funcionalidad de cada componente agregado, es decir, se probó el código con diferentes entradas válidas e inválidas para observar el comportamiento del componente y reparar cualquier error que pudiera surgir, de esta manera se aseguró que cada componente del sistema funciona correctamente. Debido al gran número de pruebas de caja blanca realizadas durante el desarrollo de este proyecto se diseñó una serie de casos de prueba de caja negra para este documento.

## 4.1. Casos de prueba

### 4.1.1. Pruebas sobre la Navegación en ARAPGUI

**Cuadro 4.1** Cambio de panel por medio de pestañas

**Precondición:** panel Simulación activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
N-01	1	Clic sobre pestaña Hormigas		Cambio de panel a Hormigas	Cambio de panel a Hormigas
	2	Clic sobre pestaña Tiempo		Cambio de panel a Tiempo	Cambio de panel a Tiempo
	3	Clic sobre pestaña Data-Rate		Cambio de panel a Data-Rate	Cambio de panel a Data-Rate
	4	Clic sobre pestaña Envíos		Cambio de panel a Envíos	Cambio de panel a Envíos
	5	Clic sobre pestaña Simulación		Cambio de panel a Simulación	Cambio de panel a Simulación

**Cuadro 4.2** Cambio de panel por medio de botones de navegación**Precondición:** panel Simulación activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
N-02	1	Clic sobre botón Siguiete		Cambio de panel a Hormigas	Cambio de panel a Hormigas
	2	Clic sobre botón Siguiete		Cambio de panel a Retardo	Cambio de panel a Retardo
	3	Clic sobre botón Anterior		Cambio de pestaña a Hormigas	Cambio de pestaña a Hormigas
	4	Clic sobre botón Anterior		Cambio de pestaña a Simulación	Cambio de pestaña a Simulación

**4.1.2. Pruebas sobre la Restricción de Valores en ARAPGUI**

www.bdigital.ula.ve

**Cuadro 4.3** Restricción de valores inválidos**Precondición:** panel Simulación activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
V-01	1	Escribir valor inválido en Cantidad de nodos	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	2	Escribir valor inválido en Tamaño hormigas	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	3	Escribir valor inválido en Saltos	v=100 v:valor ingresado	Despliegue de Mensaje de error	Despliegue de Mensaje de error
	4	Escribir valor inválido en Intervalo de cambio en los enlaces	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	5	Escribir valor inválido en tamaño máximo de segmento	v=-1 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	6	Escribir valor inválido en Puerto	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	7	Escribir valor inválido en Intervalo de impresión de tablas de probabilidad	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	8	Escribir valor inválido en Tamaño de Cola	v=-1 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior

**Cuadro 4.4** Restricción de valores inválidos**Precondición:** panel Simulación activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
V-02	1	Escribir valor inválido en Semilla	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	2	Escribir valor inválido en Tiempo de Simulación	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	3	Escribir valor inválido en Tiempo de Inicio de Aplicaciones	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	4	Escribir valor inválido en Factor de Incremento Tiempo de Cómputo	v=2 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior
	5	Escribir valor inválido Data-Rate	v=0 v:valor ingresado	Regreso a valor anterior	Regreso a valor anterior

### 4.1.3. Pruebas sobre el cambio de Distribuciones en parámetros de ARAPGUI

**Cuadro 4.5** Cambios de distribución

**Precondición:** ARAPGUI abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
D-01	1	Cambiar Distribución de Retardo en los Enlaces a Normal		Se muestran los parámetros de una distribución normal	Se muestran los parámetros de una distribución normal
	2	Cambiar Distribución de tiempo de Cómputo a Triangular		Se muestran los parámetros de una distribución triangular	Se muestran los parámetros de una distribución triangular
	3	Cambiar Distribución Cantidad de Hormigas por Envío a Uniforme		Se muestran los parámetros de una distribución uniforme	Se muestran los parámetros de una distribución uniforme
	4	Cambiar Distribución para Destinos de Hormigas a Constante		Se muestran los parámetros de una distribución constante	Se muestran los parámetros de una distribución constante
	5	Cambiar Distribución para tiempos de envío de hormigas de carga a Exponencial.		Se muestran los parámetros de una distribución exponencial	Se muestran los parámetros de una distribución exponencial

#### 4.1.4. Pruebas sobre la Modificación de Rangos en parámetros de ARAPGUI

**Cuadro 4.6** Adición de Rangos

**Precondición:** Panel de Tiempo activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-01	1	Presionar 2 veces el botón más (+) de tiempo de inicio de aplicaciones		Se muestran dos nuevos rangos de nodos para ingresar parámetros	Se muestran dos nuevos rangos de nodos para ingresar parámetros
	2	Presionar el botón menos en un rango de nodos de tiempo de inicio de aplicaciones		Se elimina el rango de nodos	Se elimina el rango de nodos

**Cuadro 4.7** Adición de Rangos

**Precondición:** Panel de Tiempo activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-02	1	Presionar 2 veces el botón más (+) de factor de incremento de cómputo		Se muestran dos nuevos rangos de nodos para ingresar parámetros	Se muestran dos nuevos rangos de nodos para ingresar parámetros
	2	Presionar el botón menos (-) en un rango de factor de incremento de cómputo		Se elimina el rango de nodos	Se elimina el rango de nodos

**Cuadro 4.8** Adición de Rangos**Precondición:** Panel de Data-Rate activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-03	1	Presionar 2 veces el botón más (+) de Data-Rate		Se muestran dos nuevos rangos de nodos para ingresar parámetros	Se muestran dos nuevos rangos de nodos para ingresar parámetros
	2	Presionar el botón menos (-) en un rango de Data-Rate		Se elimina el rango de nodos	Se elimina el rango de nodos

**Cuadro 4.9** Adición de Rangos**Precondición:** Panel de Cómputo activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-04	1	Presionar 2 veces el botón más (+) de distribución de tiempo de cómputo		Se muestran dos nuevos rangos de nodos para ingresar distribuciones	Se muestran dos nuevos rangos de nodos para ingresar distribuciones
	2	Presionar el botón menos (-) en un rango de distribución de tiempo de cómputo		Se elimina el rango de nodos	Se elimina el rango de nodos

**Cuadro 4.10** Adición de Rangos**Precondición:** Panel de Hormigas activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-05	1	Presionar 2 veces el botón más (+) de cantidad de hormigas de carga por envío		Se muestran dos nuevos rangos de nodos para ingresar distribuciones	Se muestran dos nuevos rangos de nodos para ingresar distribuciones
	2	Presionar el botón menos (-) en un rango de cantidad de hormigas de carga por envío		Se elimina el rango de nodos	Se elimina el rango de nodos

**Cuadro 4.11** Adición de Rangos**Precondición:** Panel de Destinos activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-06	1	Presionar 2 veces el botón más (+) de distribución para destinos de hormigas de carga		Se muestran dos nuevos rangos de nodos para ingresar distribuciones	Se muestran dos nuevos rangos de nodos para ingresar distribuciones
	2	Presionar el botón menos (-) en un rango de distribución para destinos de hormigas de carga		Se elimina el rango de nodos	Se elimina el rango de nodos

www.bdigital.ula.ve

**Cuadro 4.13** Adición de Rangos**Precondición:** Panel de Envíos activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
R-07	1	Presionar 2 veces el botón más (+) de distribución para tiempos de envío de hormigas de carga		Se muestran dos nuevos rangos de nodos para ingresar distribuciones	Se muestran dos nuevos rangos de nodos para ingresar distribuciones
	2	Presionar el botón menos (-) en un rango de distribución para tiempos de envío de hormigas de carga		Se elimina el rango de nodos	Se elimina el rango de nodos

C.C. Reconocimiento

### 4.1.5. Pruebas sobre la lectura de parámetros

**Cuadro 4.14** Ejecución de Simulación

**Precondición:** Panel de Simulación activo.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
P-01	1	Escribir Prueba en el campo Nombre de Experimento	s=Prueba s: valor ingresado	Campo modificado	Campo modificado
	2	Cambiar semilla a 1234	i=1234 i:valor ingresado	Campo modificado	Campo modificado
	3	Aumentar la cantidad de nodos a 11	n=11 n:valor ingresado	Campo modificado	Campo modificado
	4	Presionar la pestaña Envíos		Cambio de panel a Envíos	Cambio de panel a Envíos
	5	Presionar botón Correr Simulación		Se corre la simulación en ARAP y se abre ARAP Results, la carpeta del experimento tiene el nombre Prueba, dentro se encuentra el archivo de parámetros con el mismo nombre, en éste se leen los valores modificados: nodos 11 semilla 1234	Se corre la simulación en ARAP y se abre ARAP Results, la carpeta del experimento tiene el nombre Prueba, dentro se encuentra el archivo de parámetros con el mismo nombre, en éste se leen los valores modificados: nodos 11 semilla 1234

### 4.1.6. Pruebas sobre las Características de ARAP Results

**Cuadro 4.15** Botón de Animación

**Precondición:** ARAP Results abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
B-01	1	Presionar el botón Correr Simulación		Se corre ARAP y se abre ARAP Results	Se corre ARAP y se abre ARAP Results
	2	En ARAP Results: presionar el botón Animación		Se abre una ventana que muestra diez (10) cuadros azules (nodos) y un círculo (Hormiga), la hormiga realiza 2 saltos a nodos intermedios y luego uno al nodo final, desde donde se dibuja la ruta de ida, la hormiga regresa entonces por la misma ruta al nodo inicial.	Se abre una ventana que muestra diez (10) cuadros azules (nodos) y un círculo (Hormiga), la hormiga realiza 2 saltos a nodos intermedios y luego llega al nodo final, desde donde se dibuja la ruta de ida, la hormiga regresa entonces por la misma ruta al nodo inicial.
	3	Presionar botón OK		Se cierra la ventana de animación	Se cierra la ventana de animación

**Cuadro 4.16** Click en Barras del Histograma**Precondición:** ARAP Results abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
B-02	1	Presionar el botón Correr Simulación		Se corre ARAP y se abre ARAP Results	Se corre ARAP y se abre ARAP Results
	2	En ARAP Results: presionar la barra más a la izquierda		Se abre una ventana que muestra la tabla del modelo estadístico de una corrida	Se abre una ventana que muestra la tabla del modelo estadístico de una corrida
	3	Presionar botón Ok		Se cierra la ventana del modelo estadístico	Se cierra la ventana del modelo estadístico

**Cuadro 4.17** Botón de Comparar Experimentos**Precondición:** ARAP Results abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
B-03	1	Presionar el botón Correr Simulación		Se corre ARAP y se abre ARAP Results	Se corre ARAP y se abre ARAP Results
	2	En ARAP Results: presionar el botón Compare		Se cierra la interfaz de resultados ARAP Results y se abre ARAP Compare	Se cierra la interfaz de resultados ARAP Results y se abre ARAP Compare

**4.1.7. Pruebas sobre las Características de ARAPCompare****Cuadro 4.18** Clic en Barra del Histograma**Precondición:** ARAP Compare abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
C-01	1	Presionar la barra más a la izquierda del histograma		Se abre el directorio donde se guardan los resultados del experimento	Se abre el directorio donde se guardan los resultados del experimento

**Cuadro 4.19** Cambio de Orden del Histograma**Precondición:** ARAP Compare abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
C-02	1	Presionar el Combo Box que se encuentra bajo el histograma		Se muestran todos los parámetros por los que es posible ordenar los experimentos	Se muestran todos los parámetros por los que es posible ordenar los experimentos
	2	Elegir la opción Cantidad de Nodos		Se ordenan los experimentos por cantidad de nodos y se muestra la cantidad de nodos de cada experimento bajo su barra.	Se ordenan los experimentos por cantidad de nodos y se muestra la cantidad de nodos de cada experimento bajo su barra
	3	Presionar el ComboBox y elegir la opción Tamaño Máximo de Segmento		Se ordenan los experimentos por tamaño máximo de segmento y se muestra la cantidad de nodos de cada experimento bajo su barra.	Se ordenan los experimentos por tamaño máximo de segmento y se muestra la cantidad de nodos de cada experimento bajo su barra.

**Cuadro 4.20** Edición de experimentos a mostrar en Histograma**Precondición:** ARAP Compare abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
C-03	1	Presionar el botón Add/Remove		Se muestran la ventana de diálogo de selección de experimentos	Se muestran la ventana de diálogo de selección de experimentos
	2	Seleccionar el nombre de experimento test en la lista de elementos a Comparar y presionar el botón <<		Se elimina el test de la lista de elementos a comparar y se muestra test en la lista de la izquierda.	Se elimina test de la lista de elementos a comparar y se muestra test en la lista de la izquierda.
	3	Presionar botón Ok		Se dibuja una barra menos en el histograma	Se dibuja una barra menos en el histograma
	4	Seleccionar el primer nombre de experimento en la lista a la izquierda y presionar el botón >>		Se elimina test de la lista de la izquierda y se muestra en la lista de elementos a comparar	Se elimina test de la lista de la izquierda y se muestra en la lista de elementos a comparar
	5	Presionar botón Ok		Se dibujan todas las barras en el histograma	Se dibujan todas las barras en el histograma

**Cuadro 4.21** Edición de experimentos a mostrar en Histograma**Precondición:** ARAP Compare abierto.

Caso #	Paso	Ejecutar	Data	Resultado esperado	Resultado actual
C-04	1	Presionar el botón Add/Remove		Se muestran la ventana de diálogo de selección de experimentos	Se muestran la ventana de diálogo de selección de experimentos
	2	Presionar el botón Ninguno		Se eliminan todos los nombres de la lista de elementos a comparar y se muestran todos los nombres en la lista de la izquierda.	Se eliminan todos los nombres de la lista de elementos a comparar y se muestran todos los nombres en la lista de la izquierda.
	3	Presionar botón Ok		Se dibuja el histograma vacío	Se dibuja el histograma vacío
	4	Presionar el botón Todos		Se eliminan todos los nombres de la lista de la izquierda y se muestran en la lista de elementos a comparar	Se eliminan todos los nombres de la lista de la izquierda y se muestran en la lista de elementos a comparar
	5	Presionar botón Ok		Se dibujan todas las barras en el histograma	Se dibujan todas las barras en el histograma

# Capítulo 5

## Conclusiones y Recomendaciones

### 5.1. Conclusiones

En este trabajo se desarrolló una interfaz gráfica que envuelve la funcionalidad del simulador para protocolo de enrutamiento que provee Anonimato basado en algoritmos de optimización de colonias de hormigas artificiales llamado ARAP en el lenguaje de programación C++, haciendo uso del Framework multiplataforma QT con el entorno de desarrollo integrado QT Creator. Se determinaron las necesidades de una interfaz gráfica de usuario para el ingreso, la lectura y el análisis de datos. Se mostraron las ventajas de usar el entorno de desarrollo Qt Creator, al contar con herramientas especializadas para el desarrollo de interfaces de usuario. Además el Framework Qt provee mucha ayuda en línea, lo que permite enfocarse en las características de la GUI a desarrollar.

Se estudiaron patrones de diseño de GUI, cómo y cuándo utilizarlas, así como herramientas usadas de manera general en el desarrollo de interfaces de usuario, lo que fué de gran ayuda al seleccionar cuáles utilizar y cómo satisfacer las necesidades del usuario en la interfaz. Se dividió la interfaz en partes especializadas para tareas separadas, esto facilitó el proceso de búsqueda y corrección de errores, así como el diseño mismo de las herramientas de software.

Además de crear una interfaz de usuario para el simulador se agregaron nuevas

funcionalidades al simulador, como una simulación animada del recorrido de una hormiga

Los logros de este trabajo se enumeran a continuación:

- Creación de una interfaz Gráfica de Usuario para el Simulador ARAP de protocolo de enrutamiento que provee anonimato basado en algoritmos de optimización de colonias de hormigas artificiales. La interfaz cubre tanto la entrada de datos como el despliegue de resultados para simplificar su análisis.
- Se identificaron las características principales del simulador y se seleccionaron los elementos necesarios en su interfaz gráfica.
- Diseño de la interfaz gráfica de usuario, se determinaron los elementos que estarán disponibles en la interfaz y su distribución, se escogieron los gráficos de mayor importancia a ser mostrados en los resultados.
- Implementación de la interfaz gráfica para el simulador ARAP.
- Se creó una herramienta que simula en una animación, el recorrido de ida y vuelta de una hormiga de carga llevando un mensaje a través de la red anónima usando datos reales de la simulación. La animación respeta el comportamiento del protocolo y los parámetros introducidos en el simulador.
- Se creó otra herramienta, llamada ARAP Compare, que compara diferentes experimentos realizados en el simulador con parámetros distintos.

## 5.2. Recomendaciones

Estos son algunos trabajos que pudieran derivarse de éste y aportar al estudio del protocolo simulado por ARAP o otros similares:

- Realizar pruebas formales sobre el protocolo, teniendo a disposición el simulador ARAP, la interfaz gráfica ARAPGUI y teoría de diseño de experimentos para

verificar o rechazar la hipótesis propuesta y/o formular nuevas propuestas basadas en los resultados de las mismas.

- Creación de un módulo que se encargue de aplicar estudios estadísticos automatizados basados en los resultados del simulador y en la teoría de diseño de experimentos y generar gráficas de las estadísticas obtenidas.
- Creación de un simulador gráfico de redes educativo, similar al módulo de animación de ARAPGUI para estudiar el comportamiento de protocolos de enrutamiento y el tráfico en las redes.

Este proyecto fue realizado bajo licencia de software libre GNU GPLv3, por lo que cualquiera que desee modificarlo o construir sobre él está completamente autorizado.

Se recomienda a quien desee expandir este sistema a familiarizarse con el entorno de desarrollo integrado Qt Creator, con la manera en que el sistema ARAPGUI recibe parámetros y las formas en que muestra resultados.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

C.C. Reconocimiento

## Apéndice A

# Capturas de pantalla de la interfaz de usuario ARAPGUI

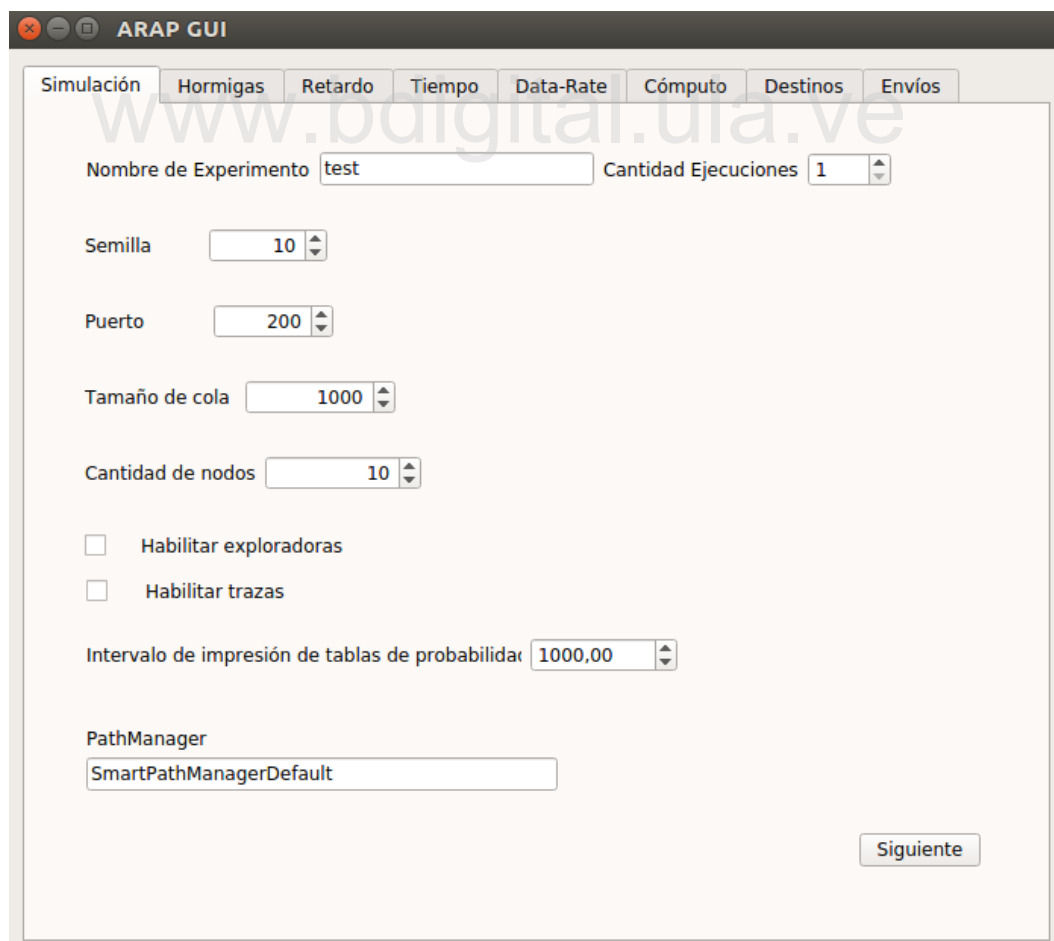


Figura A.1: Panel 1 de ARAPGUI

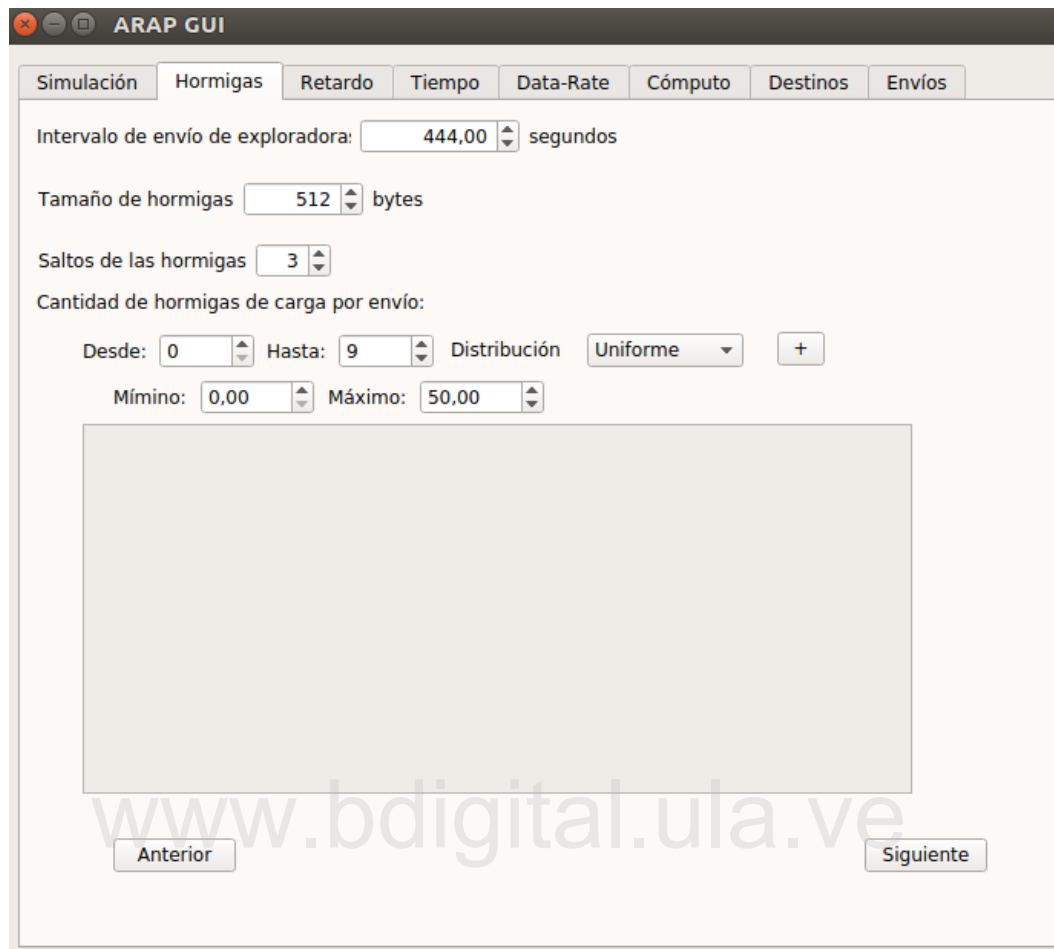


Figura A.2: Panel 2 de ARAPGUI



Figura A.3: Panel 3 de ARAPGUI



Figura A.4: Panel 4 de ARAPGUI

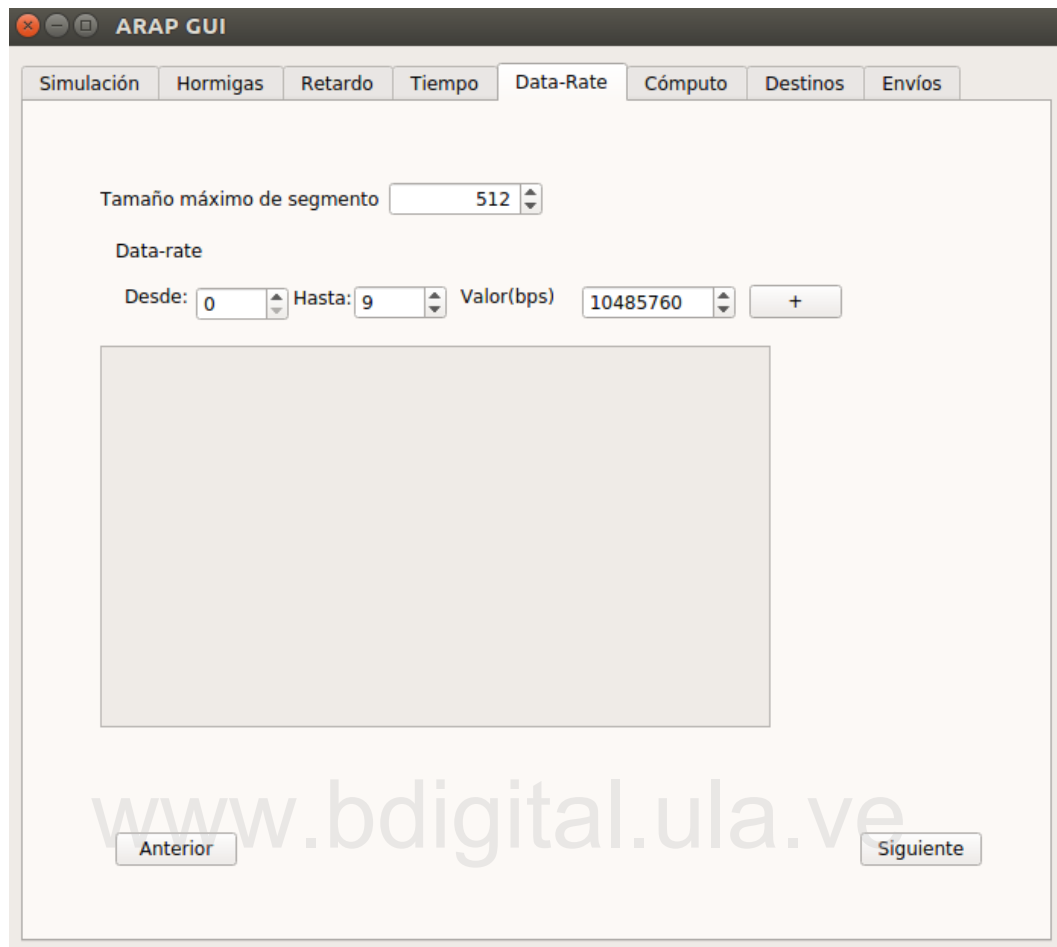


Figura A.5: Panel 5 de ARAPGUI

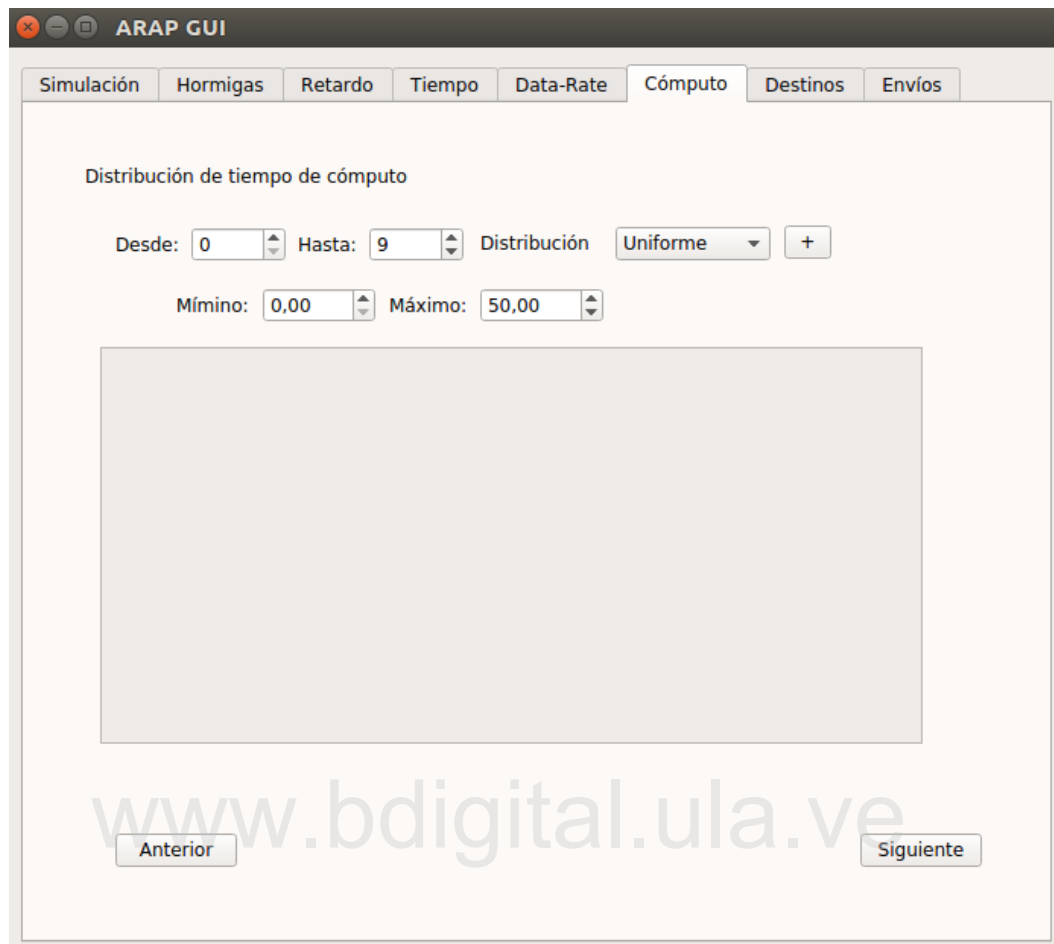


Figura A.6: Panel 6 de ARAPGUI

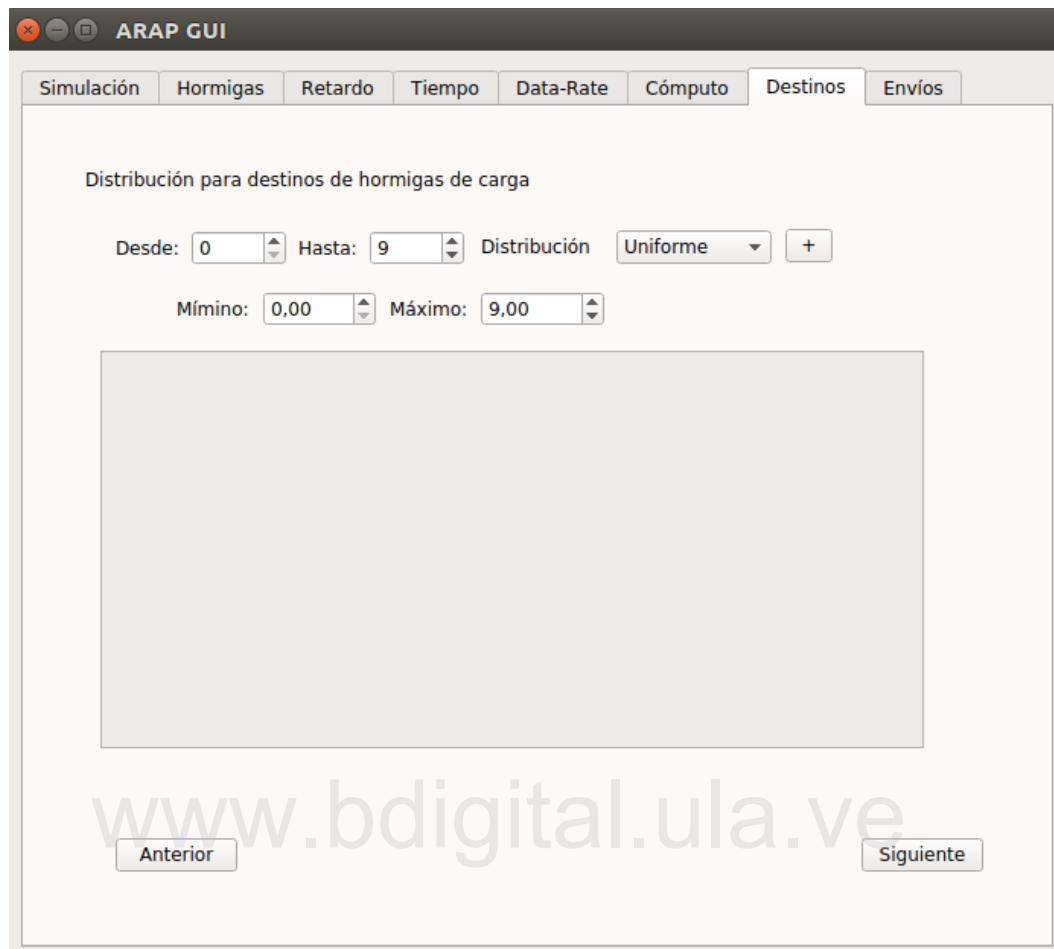


Figura A.7: Panel 7 de ARAPGUI

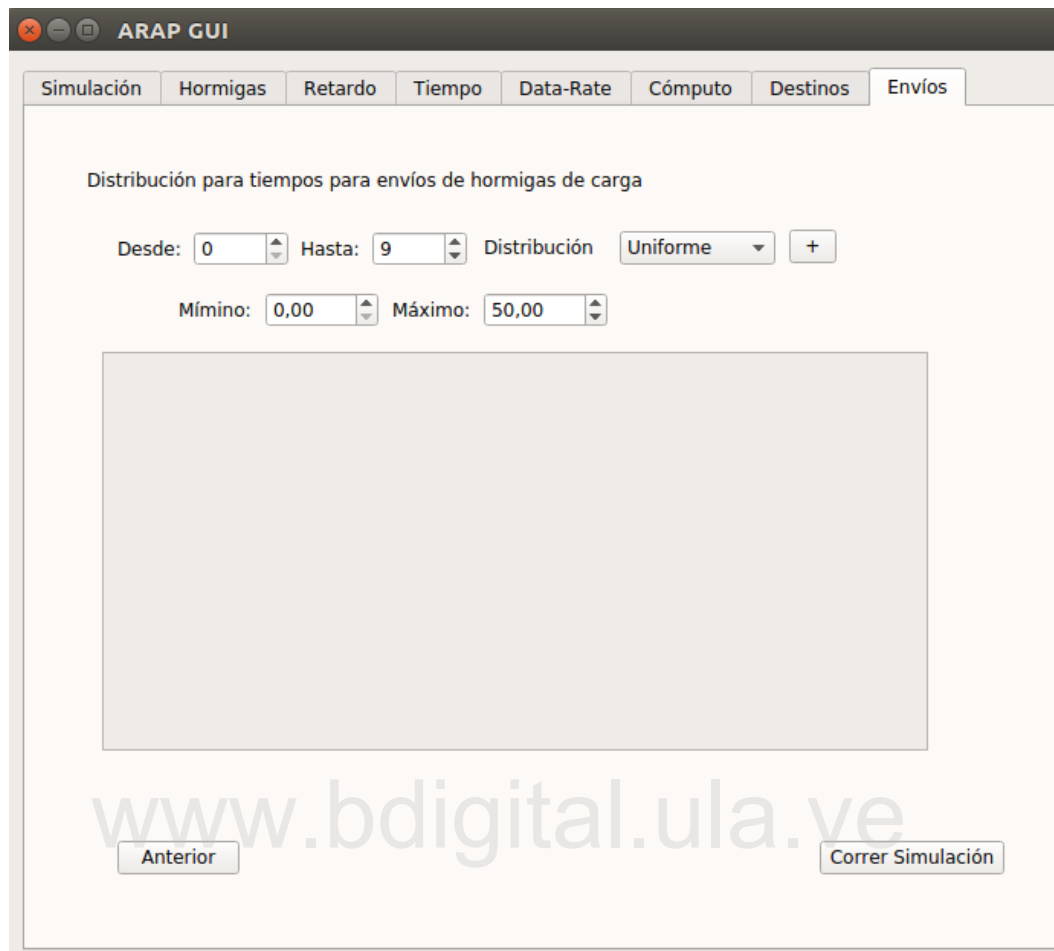


Figura A.8: Panel 8 de ARAPGUI

# Referencias Bibliográficas

- [1] Download packet tracer. En Internet, página web: <https://www.netacad.com/courses/packet-tracer-download/>. accesado junio, 2016.
- [2] Pyviz — nsnam. En Internet, página web: <https://www.nsnam.org/wiki/PyViz>, 2015. [Internet; accesado junio-2016].
- [3] Conceptos fundamentales, 2016. [Internet; accesado septiembre-2017].
- [4] Libraries & apis, tools and ide - qt, 2016. [Internet; accesado junio-2016].
- [5] Netanim — nsnam, 2016. [Internet; accesado junio-2016].
- [6] ns-3, 2016. [Internet; accesado junio-2016].
- [7] Software - gns3, 2016. [Internet; accesado junio-2016].
- [8] Topology generator — nsnam, 2016. [Internet; accesado junio-2016].
- [9] Chris Adams. Benefits of the graphical user interface. En Internet, página web: <https://www.thoughtco.com/benefits-of-graphical-user-interface-1206357>, septiembre 30, 2016.
- [10] David Cadavid. Simulador para protocolo de enrutamiento que provee anonimato basado en algoritmos de optimización de colonias de hormigas artificiales. Universidad de Los Andes, 2016. Trabajo de Grado.
- [11] R.L. Sumoza Matos, A.L. Sandoval Orozco, L.J. García Villalba, and T. Kim. Collective intelligence for anonymous systems. 2012.

- [12] Wikipedia. Anonymity, 2017. [Internet; accesado septiembre-2017].
- [13] Wikipedia. Command-line interface — wikipedia, the free encyclopedia, 2017. [Internet; accesado septiembre-2017].
- [14] Wikipedia. Graphical user interface — wikipedia, the free encyclopedia, 2017. [Internet; accesado 20-septiembre-2017].
- [15] Wikipedia. List of graphical user interface elements — wikipedia, the free encyclopedia, 2017. [Internet; accesado septiembre-2017].
- [16] Wikipedia. Protocolo de comunicaciones, 2017. [Internet; accesado septiembre-2017].
- [17] Wikipedia. Wimp (computing) — wikipedia, the free encyclopedia, 2017. [Internet; accesado septiembre-2017].

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)