



**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y  
Computación**

**“ESTUDIO Y SIMULACIÓN DE ALGORITMOS PARA LA  
EVACUACIÓN DE PERSONAS EN SITUACIONES DE  
EMERGENCIA SOBRE UNA ESTRUCTURA SIMILAR AL  
RECTORADO DE LA ESPOL.”**

**Informe de  
Proyecto de Graduación**

Previo a la obtención del Título de:  
**INGENIERO EN TELEMÁTICA**

Presentado por  
**MANUEL EDUARDO SANTOS NAVARRETE**

**GUAYAQUIL – ECUADOR**

**2015**

## **AGRADECIMIENTO**

Agradezco a todas las personas que me han apoyado y me han ayudado a alcanzar cada una de las metas que me he propuesto, sus palabras de apoyo en momentos difíciles es lo que me ayudó a no darme por vencido. Una mención especial para mi mamá y mi abuela porque simplemente les debo todo lo que soy y todo lo que llegue a ser.

## **DEDICATORIA**

A mi hermana y mi abuela que guían mi camino.

## **TRIBUNAL DE SUSTENTACIÓN**

---

MSc. Sara Ríos Orellana  
Subdecana de la FIEC

---

Prof. Ignacio Marin Garcia  
Director de Proyecto de Graduación

---

Prof. Patricia Chávez Burbano  
Miembro del Tribunal

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”.

---

Manuel E. Santos Navarrete

## RESUMEN

Este trabajo de investigación forma parte del proyecto KISHWAR y corresponde al estudio y simulación de una red que representará al edificio del Rectorado de la ESPOL sobre la cual se implementó un algoritmo que será encargado de recalcular rutas hacia las salidas más seguras durante una situación de emergencia y con esto, poder determinar cuál tiene un mejor rendimiento. Los algoritmos que fueron puestos a prueba en este estudio fueron: Dijkstra y Bellman-Ford.

Con las simulaciones que realizamos, se pudieron obtener diversos datos que fueron determinantes a la hora de elegir un algoritmo por encima del otro. El algoritmo Dijkstra según nuestro análisis es ligeramente superior a Bellman-Ford en rendimiento. Consideramos que nuestro estudio fue un éxito ya que cumplimos con los objetivos planteados, y pudimos concluir cuál algoritmo tiene un mejor desempeño para el cálculo de rutas de evacuación y esperamos que este análisis sea tomado en cuenta durante la implementación el proyecto KISHWAR de la ESPOL.

# ÍNDICE GENERAL

AGRADECIMIENTO .....	ii
DEDICATORIA .....	iii
TRIBUNAL DE SUSTENTACIÓN .....	iv
DECLARACIÓN EXPRESA .....	v
RESUMEN .....	vi
ÍNDICE GENERAL .....	vii
ABREVIATURAS Y SIMBOLOGÍA .....	x
ÍNDICE DE FIGURAS .....	xi
ÍNDICE DE TABLAS .....	xii
ÍNDICE DE ECUACIONES .....	xiii
GLOSARIO .....	xiv
INTRODUCCIÓN .....	xv
CAPÍTULO 1 .....	1
1 DESCRIPCIÓN DEL PROBLEMA .....	1
1.1 ANTECEDENTES .....	1
1.2 DESCRIPCIÓN DEL PROBLEMA .....	4
1.3 JUSTIFICACIÓN .....	5
1.4 PROPUESTA Y ALCANCE .....	6

1.5	OBJETIVOS	9
1.5.1	OBJETIVO GENERAL	9
1.5.2	OBJETIVOS ESPECÍFICOS	9
1.6	METODOLOGÍA	10
CAPÍTULO 2		12
2	MARCO TEÓRICO	12
2.1	SISTEMAS DE EVACUACIÓN	12
2.2	SISTEMAS DE EVACUACIÓN DINÁMICOS	18
2.3	ALGORITMOS DE ENRUTAMIENTO	22
CAPÍTULO 3		28
3	ANÁLISIS, REQUERIMIENTOS Y ARQUITECTURA	28
3.1	ANÁLISIS	29
3.2	REQUERIMIENTO	31
3.3	ARQUITECTURA	32
CAPÍTULO 4		34
4	DISEÑO E IMPLEMENTACIÓN	34
4.1	SOFTWARE DE SIMULACIÓN	34
4.2	SITIO DE PRUEBAS	39
4.3	SIMULACIÓN	42
CAPÍTULO 5		49
5	PRUEBAS Y ANÁLISIS DE RESULTADOS	49
5.1	DISEÑO DE PRUEBAS	49



5.2	PRUEBAS .....	53
5.3	ANÁLISIS DE RESULTADOS .....	58
	CONCLUSIONES Y RECOMENDACIONES .....	62
	ANEXO A – TABLA DE RESULTADOS DE LAS SIMULACIONES CON EL ALGORITMO DIJKSTRA	
	ANEXO B – TABLA DE RESULTADOS DE LAS SIMULACIONES CON EL ALGORITMO BELLMAN-FORD	
	BIBLIOGRAFÍA .....	67

## ABREVIATURAS Y SIMBOLOGÍA

ADSS	Active Dynamic Signage System	Sistema Activo de Señalización Dinámica
BGP	Border Gateway Protocol	
BW	Bandwidth	Ancho de Banda
CBR	Constant Bit Rate	Tasa Constante de Bits
ESPOL	---	Escuela Superior Politécnica del Litoral
GPS	Global Positioning System	Sistema de Posicionamiento Global
GUI	Graphical User Interface	Interfaz Gráfica de Usuario
LED	Light-Emitting Diode	Diodo Emisor de Luz
NAM	Network AniMator	---
NS2	Network Simulator V2	---
OSI	Open System Interconnection	Interconexión Abierta de Sistemas
OSPF	Open Shortest Path First	El Camino Más Corto Primero
PktSize	Packet Size	Tamaño del Paquete
P2P	Point-to-Point	Punto a Punto
RIP	Routing Information Protocol	Protocolo de Información de Enrutamiento
SE	---	Sistemas de Evacuación
SED	---	Sistemas de Evacuación Dinámicos
SEE	---	Sistemas de Evacuación Estáticos
SPF	Short Path First	Primero el Camino Más Corto
UDP	User Datagram Protocol	Protocolo de Datagramas de Usuario
VLSI	Very Large Scale Integration	Integración a Escala Muy Grande

## ÍNDICE DE FIGURAS

- Figura 1.1 Metodología utilizada en la investigación
- Figura 2.1 Tipo de señalética en un SEE.
- Figura 2.2 Tipo de señalética a base de LEDs en un SED.
- Figura 2.3 Señalética de Getaway indicando la dirección a seguir.
- Figura 2.4 Señalética de Getaway indicando dirección a no seguir.
- Figura 4.1 Ejemplo de archivo de trazas para NAM.
- Figura 4.2 Captura de gráficas usando Xgraph.
- Figura 4.3 Vista general del Rectorado de la ESPOL.

## ÍNDICE DE TABLAS

Tabla 1 Comparativa de SEE vs SED.

Tabla 2 Diferencias entre eventos: Incendios y Fugas de Gas.

## ÍNDICE DE ECUACIONES

Ecuación 4.1 Tiempo de transmisión de un paquete desde un nodo.

Ecuación 4.2 Tiempo en el que arribará un paquete a un nodo dado.

## GLOSARIO

**Aleatorio** .- Dependiente de algún suceso fortuito.

**Algoritmo** .- Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**Ancho de Banda** .- Medida de datos y recursos de comunicación disponible o consumida expresados en bit/s o múltiplos de él.

**Enlace** .- Unión, conexión entre nodos.

**Nodo** .- Punto de intersección o unión de varios elementos que confluyen en el mismo lugar.

**UDP** .- Protocolo del nivel de transporte basado en el intercambio de datagramas que no necesita establecer una conexión.

# INTRODUCCIÓN

Hoy en día los algoritmos son piezas fundamentales en la implementación de diversos sistemas tales como cálculo de rutas en los protocolos de encaminamiento, video juegos, o elección del mejor camino en Google Maps, pero en este estudio se van a utilizar, o mejor dicho, se van a poner a prueba ciertos algoritmos bajo condiciones un tanto inusuales para poder determinar cuál de ellos tiene el mejor rendimiento.

Este estudio tiene como punto de partida ser una base al momento de elegir un algoritmo de encaminamiento para el proyecto KISHWAR de la ESPOL. Este proyecto consiste en la implementación de un sistema de evacuación dinámica debido a que las actuales señaléticas estáticas dentro de los edificios que indican hacia donde se encuentra la salida más cercana no llegan a satisfacer las expectativas en el aspecto de cantidad de vidas salvadas. Por el contrario, este proyecto contempla la implementación de una red de sensores y de señaléticas a base de LEDs para redirigir a las personas a las salidas de emergencia de una forma más eficiente. Como es de imaginar, pese a que solamente está contemplado que se implemente en

el edificio del rectorado de la ESPOL, este sistema está basado en muchas tecnologías que deberán trabajar conjuntamente para poder desempeñarse de la forma esperada. Debido a esto, se subdividió el proyecto, en módulos más pequeños, uno de ellos es el encargado de la toma de decisiones en el mismo.

Esta toma de decisiones está basada en las diversas variables que la red de sensores va a entregar, pero que en definitiva se van a traducir a dos estados: disponible o no disponible. Teniendo en cuenta estos dos estados, se deberá proceder a calcular las rutas más óptimas hacia las salidas del edificio, contemplando que debido a algún tipo de evento podrán haber secciones del edificio que no estén disponibles para ser transitadas. En este punto es que entran en escena los algoritmos de enrutamiento, que serán los encargados de dicha toma de decisiones, pero siendo esta una topología muy particular, ya que representarán los elementos que se encuentran en un edificio tales como: pasillos, puertas, escaleras, etc. es necesario evaluar cuál algoritmo tiene un mejor desempeño. Estos algoritmos serán puestos a prueba en fallas en la red que emulen o tengan un comportamiento muy similar a alguna situación de emergencia. Gracias a estas pruebas concluirá acerca del algoritmo de mejor desempeño y los resultados y conclusiones



serán entregadas a los representantes del proyecto KISHWAR para que sean tomados en cuenta.

Teniendo en cuenta nuestros objetivos, limitantes, entre otras cuestiones, se optó por realizar todas las pruebas correspondientes en un entorno simulado, utilizando un simulador de redes muy general, ya que no requerimos comportamientos específicos de un enrutador, conmutador u otro equipo de red. Este simulador puede trabajar a nivel de nodos y a partir de ellos se pueden utilizar los módulos que sean necesarios para nuestro estudio. Pero antes de realizar las pruebas, había que definir los roles que los elementos en la red iban a representar las características físicas del edificio, por ejemplo, que iban a representar los paquetes, los enlaces, los nodos e, incluso, los eventos en la red. De esta forma, al tener bien definido todo el esquema de la red se procederían a realizar las pruebas que contengan eventos específicos que representen comportamientos similares a situaciones de emergencia en las que sea necesario evacuar el edificio en el menor tiempo posible. De esta forma garantizaremos la validez de los resultados obtenidos a partir de numerosas simulaciones ya que, como es de prever, no íbamos a recrear situaciones de emergencia en el edificio por todos los riesgos que supondrían dichas situaciones.

Este trabajo corresponde, además de darle una posible solución a un módulo del proyecto KISHWAR, a una nueva forma de utilizar algoritmos para resolver problemas de entornos reales, haciendo un acercamiento al mismo de forma virtual a base de simulaciones con los algoritmos en cuestión.

# **CAPÍTULO 1**

## **1 DESCRIPCIÓN DEL PROBLEMA**

Antes de empezar con los detalles de nuestro estudio es necesario conocer de forma global acerca de cuál fue la problemática que nos llevo a considerar realizar nuestro análisis, además se de plantear cuál fue la metodología que se llevó a cabo, además de cuáles son los resultados que se esperaban conseguir con todas las limitantes y restricciones del caso.

### **1.1 ANTECEDENTES**

El estudio que se realizó forma parte del proyecto de investigación KISHWAR de la ESPOL, siendo este un proyecto que tiene como finalidad implementar un sistema de evacuación dinámica de forma que se pueda optimizar la evacuación de

personas que se encuentran dentro del edificio del rectorado de esta universidad durante situaciones de emergencia.

El edificio del rectorado, así como la gran mayoría de edificios no solamente en Ecuador sino en el mundo, utilizan señaléticas estáticas que indican, desde ese punto en particular cuál es la salida más cercana. Si bien esta solución es muy popular debido a cuestiones monetarias, es un sistema que está catalogado como sumamente ineficiente al momento de salvar una mayor cantidad de vidas por diversas razones que serán mencionadas a lo largo de este estudio. Por este motivo, la alternativa que se contrapone a estas implementaciones estáticas es una que tenga características dinámicas, que se recupere ante catástrofes debido a alguna situación de emergencia y continuamente este monitoreando una red de sensores para en base a los mismos tomar decisiones de hacia donde redirigir a las personas durante la evacuación de la estructura. Desde este punto nace el iniciativa del proyecto KISHWAR de la ESPOL que a breves rasgos es implementar un sistema de evacuación dinámica.

El proyecto KISHWAR, en su totalidad, sería un sistema bastante complejo, por lo que fue necesario implementar el paradigma “dividir y conquistar” para subdividir el problema en módulos para poder llegar los resultados esperados en el proyecto. Uno de esos módulos es la parte que corresponde a, vamos a llamarlo, inteligencia del sistema, debido a que para que exista un sistema de evacuación dinámica, es necesario contar con un algoritmo que se encargue de tomar las decisiones acerca de hacia donde deberían señalar los indicadores con el fin de que las personas puedan ser evacuadas de forma más eficiente. Pero antes de llegar a implementarlo, era necesario realizar un estudio previo acerca de cuál algoritmo se desempeña con un mejor rendimiento debido a que no es lo mismo utilizar un algoritmo para enrutar paquetes en una red de computadoras, ni para utilizarlo en algún juego de video, ni para determinar la mejor ruta utilizando un GPS, menos aún para la situación planteada que es el redireccionamiento de personas dentro de un edificio durante alguna situación de emergencia. Este estudio es el que presentará, como recomendación, cual es el algoritmo que tiene un mejor desempeño en este caso tan particular de el cálculo de rutas.

Con estos antecedentes ya que puede definir de una forma más formal y más explícita la descripción de la problemática que deseamos resolver.

## **1.2 DESCRIPCIÓN DEL PROBLEMA**

Como ya se mencionó previamente, los sistemas de evacuación estáticos no ofrecen mucha eficacia al momento de evacuar la mayor cantidad de personas que se encuentran dentro de un edificio, el cual es el problema principal y gracias a este surgió la idea del proyecto KISHWAR

Pero dentro de este proyecto, y lo que atañe más a nuestro análisis, el problema se centra poder determinar cuál algoritmo puede redirigir a las personas hacia las salidas de una forma más óptima. Por tanto, es necesario encontrar el algoritmo que más se adapte al tipo de topología que representará al edificio del rectorado y pueda calcular las mejores rutas para que de esta forma los indicadores puedan señalar hacia donde se encuentra la salida más cercana con el fin de poder salvar más vidas humanas y en el menor tiempo posible. Debe ser un algoritmo que se recupere de forma muy rápida ante posibles

fallos en la red, ya que los mismos se traducen a situaciones de emergencia tales como: incendios, terremotos/temblores y/o fugas de gas. Para lograr esa premisa, es necesario contar con un algoritmo que calcule las rutas hacia la salida más segura.

De esta forma, hemos definido de una forma más explícita al problema planteado que es una parte esencial al momento de empezar cualquier estudio ya que gracias al mismo, se puede dilucidar la justificación y los objetivos de nuestro análisis.

### **1.3 JUSTIFICACIÓN**

En este punto la justificación de nuestro estudio se puede visualizar casi de inmediato ya que habiendo explicado los antecedentes y habiendo descrito la problemática, la justificación viene automáticamente.

Nuestra justificación para haber emprendido este estudio es dar una solución a una de las problemáticas del proyecto KISHWAR, que es el módulo del algoritmo a utilizar. Si bien este trabajo no comprende de la implementación de un algoritmo directamente en el proyecto, si proporcionará una base sólida para ofrecer como opción de implementación a un

algoritmo específico. Aunque las personas que implementarán el sistema en el proyecto KISHWAR tendrán la potestad de utilizar el algoritmo que prefieran, con nuestro estudio se pretende orientar de una forma fiable y segura dicha toma de decisiones.

Habiendo definido cuál es la justificación para este estudio el siguiente paso es definir cuál es la propuesta y el alcance de nuestro análisis ya que el mismo tiene sus limitantes como todo proyecto.

#### **1.4 PROPUESTA Y ALCANCE**

En esta sección se expondrá cual es nuestra propuesta, tanto en la forma como se pensó abordar nuestro análisis así como también cuáles son los resultados que se esperan obtener a fin de poder concluir de forma precisa y, finalmente, cuáles son las limitantes y restricciones que tendrá nuestro estudio.

El primer punto a mencionar como propuesta, es la forma en que decidimos encarar el problema dado que existen dos formas: la teórica y la experimental. El acercamiento que se adoptó para obtener los resultados esperados no fue un método



teórico (en el cual la complejidad matemática sería muy alta para determinar que algoritmo tendría un mejor desempeño, sin mencionar que sería muy complejo la representación de un modelo matemático para una situación de emergencia), por el contrario se realizó un análisis experimental de cómo se comportarían los algoritmos durante dichos eventos y como se recuperan de las fallas de la red para, de esta forma, llegar a concluir cuál sería el algoritmo idóneo para la implementación en el proyecto KISHWAR. A lo largo de este trabajo nos explicaremos ventajas de haber escogido este tipo de acercamiento para la resolución del problema.

Siguiendo con lo planteado, es necesario conocer cuales eran los resultados esperados a nuestro estudio con el fin de determinar si fue un éxito o si haría falta contemplar más variables para concluir de una forma más consistente. Los resultados que se esperaban conseguir giran en torno a evaluar algunos algoritmos en el simulador de redes, y mediante los datos recopilados poder determinar cuantas personas son evacuadas para cada algoritmo, cuantas quedarían atrapadas y finalmente cual algoritmo lo realiza en el menor tiempo posible. De esta forma se cuantificará la eficiencia de los algoritmos en

una red que vaya presentando fallas de forma sistemática que será una fiel representación del rectorado y las posibles situaciones de emergencia que puedan suscitarse en el mismo. Además dentro de los resultados esperados están contempladas las limitaciones lógicas que llega a tener este estudio, como lo es el hecho de que es un análisis puramente experimental y bajo ningún concepto se procedió a realizar pruebas de incendios, fugas de gas o cualquier otro evento de esta naturaleza ya que esto supondría un riesgo innecesario para las personas que se encuentren dentro del edificio. Estos resultados, con sus limitantes, serán determinantes para que un algoritmo específico sea recomendado para que sea implementado en el proyecto KISHWAR de sistemas de evacuación dinámicos.

De esta forma, se ha realizado una breve descripción acerca de en que consiste este estudio, como se lo va a realizar y que resultados se esperaban obtener que vayan acorde a las limitaciones inherentes de este análisis.

## **1.5 OBJETIVOS**

Los objetivos que nos hemos planteado para poder catalogar a nuestro estudio como un éxito son los siguientes:

### **1.5.1 OBJETIVO GENERAL**

El objetivo general es estudiar, simular y evaluar diferentes algoritmos de enrutamiento para evacuar en casos de emergencia una estructura similar al rectorado de la ESPOL.

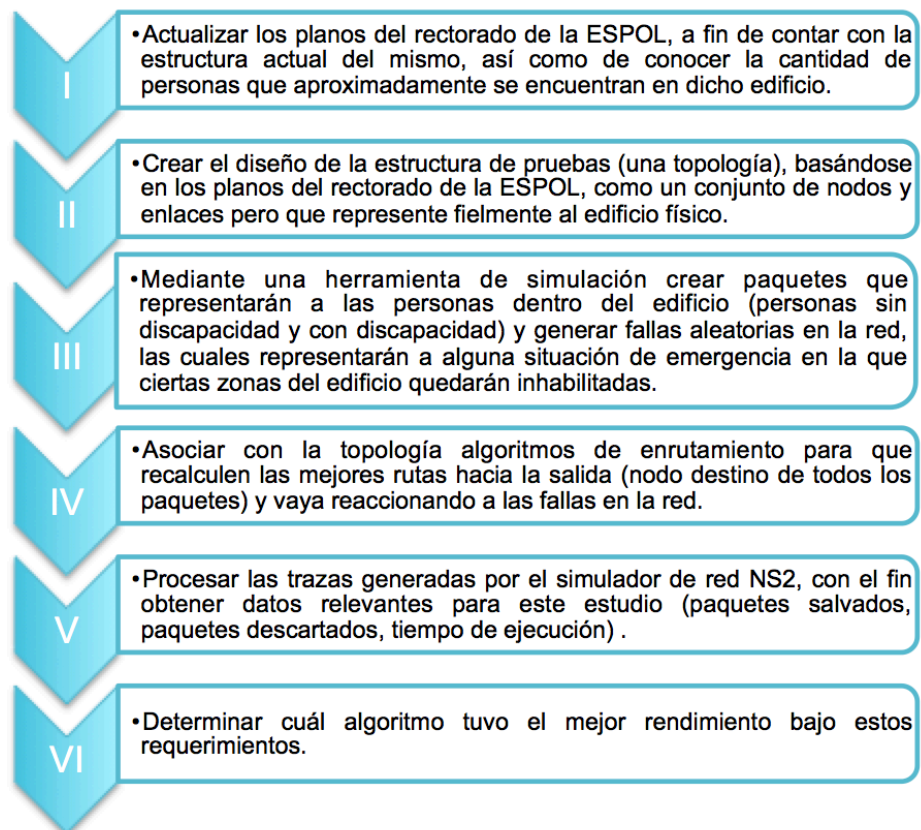
### **1.5.2 OBJETIVOS ESPECÍFICOS**

Los objetivos específicos que han sido planteados en la realización de este trabajo son:

- Evaluar diferentes algoritmos de enrutamiento en función de su desempeño para generar rutas de evacuación en la estructura seleccionada.
- Crear el diseño de la estructura de pruebas basado en el rectorado de la ESPOL.
- Implementar la solución a través de simulaciones en el entorno NS2.
- Valorar los resultados y sugerir su uso en el proyecto KISHWAR.

## 1.6 METODOLOGÍA

La metodología que se utilizó para alcanzar nuestros objetivos es la que se muestra en la Figura 1.1:



**Figura 1.1 Metodología utilizada en la investigación**

La metodología planteada corresponde, como se mencionó previamente, a un modelo experimental de análisis por lo que fue necesario con un simulador de redes que soporte la implementación de los algoritmos a ser evaluados. Se realizó nuestro estudio gracias a un simulador de redes, debido a

nuestra más grande limitante como lo es el hecho de que bajo ningún concepto es posible, como es de suponer, recrear o realizar situaciones de emergencias en el entorno real como lo es el rectorado de la ESPO. Esta metodología esta orientada a poder realizar varias interpretaciones con los datos que son arrojados por el simulador por lo que las limitantes de nuestro estudio no representaron un factor determinante que nos impidieran concluir de forma adecuada.

La metodología descrita como pudimos ver es relativamente simple, pero muy concisa y orientada a obtener los resultados esperados de una forma muy confiable ya que cada paso está enlazado con el paso anterior y con el siguiente.

En este capítulo se explicó de forma breve la problemática que se desea solucionar, se dieron pautas acerca de cómo se implementaron las simulaciones, cuales son los objetivos de este trabajo de tesis de grado tanto generales como específicos pero para comprender más a fondo todo lo relacionado a este análisis, en el siguiente capítulo se conocerán los tipos de sistemas de evacuación y todo lo implica la implementación de cada sistema.

## **CAPÍTULO 2**

### **2 MARCO TEÓRICO**

En este capítulo se va a explicar, entre otra cosas, acerca de los tipos de sistemas de evacuación que existen para desalojar a personas que se encuentran dentro de un área específica en caso de que exista alguna emergencia de cualquier índole. Existen dos categorías de sistemas de evacuación: los estáticos y los dinámicos, cada uno con diferentes características; en los siguientes subcapítulos se profundizará acerca de los mismos, pero ubicando a los sistemas de evacuación dinámica un escalón por encima de los estáticos.

#### **2.1 SISTEMAS DE EVACUACIÓN**

Los sistemas de evacuación son fundamentales en el diseño de estructuras físicas que vayan a albergar personas, ya que permiten el desalojo del edificio de una forma más sistemática,

eficiente y segura, permitiendo salvar a más personas en menor tiempo. Existen dos grupos generales de sistemas de evacuación: los Sistemas de Evacuación Estáticos (SEE) y los Sistemas de Evacuación Dinámicos (SED).

Los SEE son los sistemas de evacuación más antiguos y más utilizados a nivel mundial. Estos sistemas constan de un conjunto de señaléticas que no cambian la información mostrada en ellas con el paso del tiempo, ni cuando la situación en el edificio cambia debido a alguna situación extrema. Como se muestra en la Figura 2.1, estos sistemas comprenden las señaléticas que generalmente indican donde se encuentra la salida más próxima tomando su ubicación como punto de partida. La colocación de este tipo de señaléticas es muy sencilla y no supone la implementación de sistemas complejos. Los SEE son ampliamente usados por diversos factores tales como: limitaciones de recursos en décadas pasadas, simplicidad, cuestiones económicas, etc. Pero no siempre proveen de información fiable, ya que alguna salida puede no estar disponible y ante ese tipo de inconvenientes, no pueden redireccionar a las personas a otra salida.



Figura 2.1 Tipo de señalética en un SEE[11]

Como contrapunto se encuentran los SED, que son sistemas que recién se están comenzando a implementar, existe un sitio que han empezado a utilizar este tipo de sistemas como alternativa para evacuar a las personas de una forma más eficaz. Constan de un conjunto de tecnologías e implementaciones tanto a nivel de hardware como de software ya que su característica de determinar la mejor ruta de escape e indicarlo mediante señaléticas dinámicas no podría llevarse a cabo sin dichas implementaciones. La colocación de estos sistemas tiene un grado de complejidad mayor en comparación con los SEE, ya que no solo consiste en la ubicación de las señaléticas, sino de los respectivos sensores,



servidores, entre otras cosas, que se discutirán más en detalle en el siguiente subcapítulo. Como se puede observar en la Figura 2.2, la mayoría de implementaciones de este tipo, suelen utilizar señaléticas a base de LEDs para indicar la dirección hacia donde se deben dirigir las personas, otro tipo de señaléticas podría incurrir a elevar aún más los costos de un posible implementación de sistemas SED. Con el avance de los sistemas informáticos y del hecho que exista una mayor interoperabilidad entre sistemas hoy en día, ha permitido que se lleguen a implementar en ciertos sitios concurridos sistemas SED para evitar que se forme un caos si ocurre alguna situación de emergencia.

Para poder contrastar ambos sistemas de evacuación se diseñó la Tabla 1 la cual muestra características principales, tanto positivas como negativas de ambos sistemas con el fin de que pueda ser tomada en cuenta al momento de escoger que tipo de sistema de desalojo se va a implementar en una estructura.



Figura 2.2 Tipo de señalética a base de LEDs en un SED [12]

Tabla 1 Comparativa de SEE vs. SED

SEE	SED
Señaléticas estáticas.	Señaléticas dinámicas.
Implementación sencilla.	Implementación compleja.
Más Económico.	Más costoso.
Indican a puerta de salida cercana.	Indican puerta de salida segura y cercana.
No ofrece información tan fiable/segura	Ofrece información muy fiable/segura.
No se actualizan si existen cambios en la estructura.	Actualizan información si existe cambios en la estructura

Como podemos observar en la Tabla 1 ambos sistemas ofrecen ventajas y desventajas, pero todo dependerá del tamaño de la estructura, la afluencia de personas en el edificio y el factor económico. Se podría decir que si la estructura en cuestión es no es muy grande o no tiene una gran cantidad de personas en el edificio, se podría optar por una implementación sencilla y económica como lo son los SEE; por el contrario, si la estructura es extensa, es muy concurrida, lo ideal sería diseñar un SED ya que bajo ese entorno, estos ofrecen una mayor eficacia al momento de salvar vidas.

Hemos expuesto cuales son los tipos de sistemas de evacuación que existen actualmente para el desalojo de personas dentro de una estructura durante una situación de emergencia; se pudo deducir que, bajo ciertas condiciones, se puede optar por implementar un SEE o un SED. Pero nuestro análisis se centra en los SED, por lo que en la siguiente subsección de este capítulo se analizará más a fondo estos sistemas.

## **2.2 SISTEMAS DE EVACUACIÓN DINÁMICOS**

Así como se mencionó brevemente en el subcapítulo previo, los SED no son ampliamente utilizados en la actualidad, aún en el mercado existe una notable preferencia hacia los SEE. Una de las razones por las cuales existe esta brecha entre ambos SE, es definitivamente el alto costo y la complejidad de la implementación, esto es, ya que ambas características están estrechamente relacionadas, debido a que si aumenta la complejidad de la implementación, lógicamente, el costo final aumentará. La complejidad es alta, ya que estos sistemas deberían contar con una gran cantidad de sensores, sean de incendios, materiales inflamables, de obstrucción de alguna de las vías, etc. dependiendo del alcance del sistema, y esto es más notorio cuando la estructura en cuestión es más extensa. Además de los sensores, se requieren servidores que concentren la información y que dichos datos sean procesados y, utilizando algún algoritmo, determinar las rutas de escapes más óptimas y seguras hacia el exterior del edificio; finalmente se debe de mostrar la información pertinente en cada una de las señaléticas dinámicas que incluya el sistema. Estos sistemas de evacuación pueden llegar a ser muy costosos y complejos, según sea el alcance deseado, pero la cantidad de

vidas que puede salvar, hace que sea una alternativa a tomar en cuenta al momento de diseñar sistemas de escape.

Los SED, pueden ser implementados de tres formas diferentes: centralizado, distribuido o mixto; la diferencia entre estos tipos de sistemas es la forma en la que fluyen los datos a través de la red. El SED centralizado se basa en tener la red de sensores y que los mismos envíen sus datos hacia un servidor local y este a su vez, concentre todos los datos para poder realizar todo el procesamiento requerido en la toma de decisiones; las acciones tomadas son en torno a toda la red y a poder evacuar a las personas de la edificación. Los SED distribuidos son basados en que entre los sensores exista una interacción directa, sin la presencia de un servidor o concentrador de datos; las acciones tomadas en estos sistemas suele ser más a nivel local, por ejemplo, una habitación que contenga un sensor de humo y este posea una conexión P2P con un LED afuera de la habitación que cambie de estado según la información recibida del sensor. Finalmente, se encuentran los SED mixtos, que básicamente utiliza las implementaciones distribuida y centralizada, es decir, consta de una red de sensores que envían información a un servidor que procesará los datos, pero

también existirán enlaces P2P según sea requerido. Los sistemas distribuidos son ideales para que usar datos de forma local, sin necesidad de ser procesados en un servidor, pero si se requiere un procesamiento adicional, lo más conveniente sería usar un sistema centralizado que calcule rutas, que envíe información a los LEDs de las señaléticas, etc. o se puede optar por un SED mixto, si se desea tener mayor robustez en la implementación.

Previamente se mencionó que existía la implementación de un sistema de evacuación dinámico, esta es la diseñada en el proyecto Getaway y que realizó unas pruebas en una estación de ferrocarriles de Sant Cugat del Vallès en la provincia de Barcelona, España. Este sistema contaba con señaléticas luminiscentes, denominadas ADSS, que se encargan de guiar a los usuarios del ferrocarril hacia las salidas más cercanas, de forma dinámica. Ejemplos de estas señalizaciones se encuentran en la Figura 2.3 y la Figura 2.4, en ellas se pueden observar que indica la dirección que deben seguir las personas y que dirección no deben seguir, respectivamente. El proyecto Getaway realizó un simulacro en el andén de la estación en una hora que la afluencia de personas era la máxima (cerca de 250

personas), en esta prueba existieron usuarios que no siguieron los indicadores lumínicos, dado que preferían salir por la que ellos conocían, pero que los indicadores señalaban que no era la correcta, pese a ello muchas personas si siguieron las señaléticas. De esta forma podemos ver, que los primeros simulacros del proyecto Getaway parecen haber sido un éxito y que definitivamente son más eficientes que los sistemas de evacuación convencionales.



**Figura 2.3 Señalética de Getaway indicando la dirección a seguir [9]**



**Figura 2.4 Señalética de Getaway indicando dirección a no seguir[9]**

Hemos descrito a los sistemas de evacuación dinámica, cuales son sus características internas, los tipos de SED que existen y en que consisten cada uno de ellos y finalmente, se mencionó acerca de sitios donde se han implementado este tipo de sistemas. El resto de nuestro análisis se enfocará en los SED centralizados y/o mixtos, ya que se requiere un alto grado de procesamiento para que un algoritmo de enrutamiento tome decisiones adecuadas sobre el desalojo del edificio.

### **2.3 ALGORITMOS DE ENRUTAMIENTO**

Debido a que nuestro estudio se va a enfocar en los algoritmos de enrutamiento es necesario conocer más a fondo en que consisten, cuáles son sus propiedades, como se clasifican los mismos y, por último, se darán detalles acerca de dos



algoritmos de enrutamiento muy utilizados en la actualidad, como lo son: Bellman-Ford [7] y Dijkstra [8].

Antes que nada, es necesario conocer acerca de los algoritmos en general, para luego irnos encasillando en lo que propiamente son los algoritmos de enrutamiento. Se puede definir como un algoritmo a un conjunto de instrucciones que generalmente son secuenciales y finitas que permite llevar a cabo una actividad específica de una forma más estructurada. De esta forma, existen diversos algoritmos para un cantidad infinita de funciones, y una de ellas en particular son para tomar decisiones de encaminamiento de objetos como lo son los algoritmos de enrutamiento, que tienen como objetivo poder determinar la ruta más óptima, según un determinado criterio, a cualquier punto que pueda ser marcado como destino. Entre las propiedades que se espera que cumplan los algoritmos de enrutamiento son las siguientes:

- Tener un alto grado de robustez y ser adaptables ante eventos/fallas en la red.
- Tener tiempos de convergencia muy pequeños.
- No deben general lazos.

Después de haber definido a los algoritmos de enrutamiento y haber explicado cuáles son las propiedades que más los caracterizan, es necesario conocer acerca de su clasificación.

Al mencionar la clasificación de los algoritmos de enrutamiento, nos vamos a referir en realidad al lugar en donde realiza la toma de decisiones ya que en dicho punto de la red es donde se implementará el algoritmo que arroje los resultados de las rutas más óptimas a todos los destinos de la red. Si bien existen los algoritmos de enrutamiento estáticos o no adaptivos, estos corresponden a calcular las rutas al inicio (y no se vuelven a calcular), antes que exista tráfico en la red o haya ocurrido algún tipo de evento por el cual haya que recalcular las rutas nuevamente, este tipo de casos no estarán considerados en nuestro estudio debido a que lo que se requiere es que el algoritmo se vuelva a ejecutar en el caso de un evento. Es por esto que se tomarán en cuenta el tipo de encaminamiento dinámico o adaptivo y bajo esta consigna, el lugar de decisión de las rutas puede ser: centralizado, distribuido, o puede encontrarse en origen; sin embargo, la tercera opción no es utilizada comúnmente ya que el algoritmo para el cálculo de rutas se implementa en los nodos de origen y este nodo

comunica sus decisiones a todos los nodos de la red. Por otra parte, los algoritmos de enrutamiento que se encuentran centralizados corresponden a que un solo nodo en la red funciona como una especie de “administrador de rutas” puesto que es el encargado de implementar el algoritmo de enrutamiento y ejecutarlo cada vez que sea requerido, para luego notificar las mejores rutas a los demás nodos en la red; si bien son más sencillos de implementar tienen una gran desventaja que es el hecho de tener un solo punto de falla, por lo que se decidió no implementar este tipo de encaminamiento en las simulaciones que se realizaron en nuestro análisis. Finalmente, nos queda el tipo de enrutamiento dinámico distribuido que consiste en que en cada nodo de la red exista una instancia del algoritmo para que cada uno realice el cálculo de las rutas y cuando arribe un paquete, en base a sus propias decisiones de encaminamiento, envíe el paquete al siguiente salto. Esta es la clasificación de los algoritmos de enrutamiento basándonos en el lugar donde se realizan las tomas de decisiones; ahora procederemos a profundizar en dos tipos de algoritmos de enrutamiento dinámico distribuido que son muy utilizados: Bellman-Ford y Dijkstra.

Dijkstra y Bellman-Ford son algoritmos ampliamente utilizados en las redes de computadoras debido a que son implementados en los protocolos de enrutamiento de paquetes OSPF y RIP respectivamente. Bellman-Ford es un algoritmo de vector distancia, llamado así ya que indica por donde tiene que ser enrutado el paquete para llegar a su destino y cual es el costo para llegar al mismo pero no conoce toda la topología de la red; este algoritmo es eficiente hablando de términos de convergencia inicial, pero se lo puede catalogar como un tanto lento cuando se tiene que recuperar de fallas que ocurran en la red. El algoritmo de Dijkstra es un algoritmo de estado de enlace que conoce toda la topología y sirve para encontrar la ruta con menor costo hacia todos los destinos de la red teniendo como parámetro los costos de los enlaces; este algoritmo se lo considera una optimización del algoritmo Bellman-Ford que trabaja también mediante iteraciones. Ambos algoritmos considerados de SPF tienen diversas aplicaciones, no solamente el de formar parte de protocolos de enrutamiento, sino también como por ejemplo: Rutas en mapas (por ejemplo Google Maps), sistemas de navegación de robots, planificación de tráfico urbano, subrutinas en algoritmos avanzados, protocolos de enrutamiento (OSPF, RIP, BGP), canalización

óptima en chips VLSI, aproximaciones de funciones lineales por tramos, entre otros. Como se puede ver, estos algoritmos de enrutamiento similares entre si, tienen una extensa cantidad de funciones que pueden desempeñar, y gracias a nuestro análisis, podría añadirse la implementación de sistemas de evacuación dinámica a esta lista.

En esta subsección se ha profundizado en el tema de los algoritmos y más específicamente, en los algoritmos de enrutamiento, conociendo cuales son sus características principales, conociendo cuáles son sus categorías y explicando a breves rasgos a dos algoritmos muy utilizados en la actualidad en distintos ámbitos.

Este capítulo estuvo orientado a conocer más en detalle los sistemas de evacuación y más específicamente a los sistemas de evacuación dinámica, además de los algoritmos de enrutamiento de paquetes, sus propiedades, tipos e incluso en donde se los implementa.

## **CAPÍTULO 3**

### **3 ANÁLISIS, REQUERIMIENTOS Y ARQUITECTURA**

Previo a la implementación del proyecto fue necesario analizar los requerimientos que dicho sistema, el cual debía cumplir a la hora de evacuar de una forma adecuada y pertinente. En base a los requerimientos se pudo definir adecuadamente la arquitectura del mismo. Una vez que dichos requerimientos fueron correctamente ponderados y la arquitectura del sistema pudo ser determinada y en base a ambos parámetros, requerimientos y arquitectura, se realizó el diseño tal y como se muestra en el Capítulo 4.

### **3.1 ANÁLISIS**

No solamente se debió de analizar los requerimientos reglamentarios de evacuaciones en casos de emergencia, sino que en nuestro caso se debió de analizar la estructura en la cual el proyecto KISHWAR se implementara: El rectorado de la ESPOL. Para dicho análisis se tomo en cuenta la ubicación del edificio, el tipo de estructura y las rutas verticales de movimiento.

El rectorado es un edificio de los años 90 situado en la parte superior de una loma y rodeado por tanto edificios donde se realizan actividades docentes, como por áreas naturales como son parte del bosque y el lago de la ESPOL. La disponibilidad de espacio al cual evacuarse fue tomada en cuenta y se considero que existía mas que suficiente espacio para cualquiera de los escenarios planteados.

La estructura realizada en hormigón reforzado y cristal, cuenta con cuatro niveles de construcción: sótano, planta baja, primera planta y segunda planta. Debido a la construcción y a los materiales utilizados se planteo que los escenarios mas probables de evacuación fueran el de terremoto, con múltiples

daños estructurales los cuales una vez iniciado el evento no crecerían; incendios y fugas de gases nocivos donde se esperaba que el área dañada fuera incrementada a lo largo del tiempo del evento tomando en cuenta la falta de dispositivos automáticos de extinción de incendios o de evacuación de gases.

El movimiento de las personas a través del edificio se realiza por tres escaleras principales de las cuales únicamente una tiene acceso al sótano. Así mismo existe un ascensor que permite el acceso a las cuatro plantas pero que además de no deberse usar en casos de emergencia, se encuentra habitualmente deshabilitado.

Las decisiones sobre el tipo de escenarios se basaron en el análisis anterior y junto a los requerimientos determinaron los valores a utilizar en las simulaciones.



### 3.2 REQUERIMIENTOS

Debido a que se plantea el sistema KISHWAR como una solución de evacuación dinámica, y a través de este proyecto se planteaba validar y seleccionar el algoritmo de evacuación óptimo para dichos casos, las decisiones fueron basadas en los siguientes requerimientos mínimos:

- Las evacuaciones deberán de realizarse en un tiempo inferior o igual a quince (15) minutos basados en estándares internacionales de evacuaciones de emergencia.
- El objetivo de las evacuaciones deberá de ser evacuar al mayor número de personas posibles dentro del tiempo previamente indicado y se considerará que aquellas que no sean evacuadas en dicho tiempo habrían resultado muertas en el evento.
- El sistema de enrutamiento utilizado debe de ser capaz de recalcular las rutas de evacuación de forma dinámica al entenderse que diferentes “nodos” pueden desaparecer durante una emergencia al ser afectados por el evento. Por ejemplo en el caso de un incendio, a lo largo del tiempo del evento los nodos situados en la periferia del mismo serían incluidos en el incendio al extenderse el mismo.

En base a los requerimientos de tiempo de evacuación, maximizar de posibilidades de evacuación y estado dinámico de los eventos, se resolvió la arquitectura a utilizar así como los parámetros máximos y mínimos de las simulaciones.

### **3.3 ARQUITECTURA**

La arquitectura del sistema está basada en los requerimientos, la infraestructura tecnológica a utilizar en la implementación piloto y la infraestructura tecnológica utilizada en las simulaciones de eventos con las que se validaría el uso de un algoritmo de encaminamiento u otro.

Los requerimientos de tiempos de evacuación, capacidad reactiva y evacuación optima de personal determinaron la utilización de paquetes de datos estandarizados y de dos tamaños que representarían mediante su llegada o no la viabilidad de las simulaciones. La selección de los tamaños fue en base a la movilidad de los evacuados simulados siendo de mayor tamaño los paquetes que representaban a las personas con discapacidades funcionales motoras por lo que el tiempo de procesamiento de los mismos se vería demorado y por lo tanto reflejaría dichas limitaciones motoras.

La infraestructura tecnología a utilizar, directamente relacionada con el entorno físico preexistente, determinó que en las simulaciones se utilizarán colas con prioridad que reflejarán adecuadamente la realidad del lugar de implementación.

Debido a la imposibilidad de utilizar el lugar de implementación físico, el proyecto fue implementado a través de un simulador, NS2, utilizando una arquitectura distribuida de nodos y colas y considerando unos recursos comunicacionales limitados.

Basándonos en el análisis, los requerimientos de evacuación y la arquitectura del sistema utilizada, se procedió a realizar un diseño en el cual se segmentaron los planos proveídos por la Unidad de Planificación e Infraestructura Física de la ESPOL entre otros.

## **CAPÍTULO 4**

### **4 DISEÑO E IMPLEMENTACIÓN**

Después de conocer las bases de nuestro estudio, es necesario enfocarnos a los temas como el software que se utilizó para las simulaciones y las pruebas a los algoritmos, además de la estructura sobre la cual se realizó el diseño de la topología de forma que esta represente lo encontrado en los planos. Finalmente se detallará la equivalencia de cada uno de los elementos de la red, en relación a la estructura física.

#### **4.1 SOFTWARE DE SIMULACIÓN**

Para poder realizar las simulaciones y las pruebas requeridas a los algoritmos, es necesario utilizar un software especializado en redes, pero que no sea en redes de computadoras, es decir,

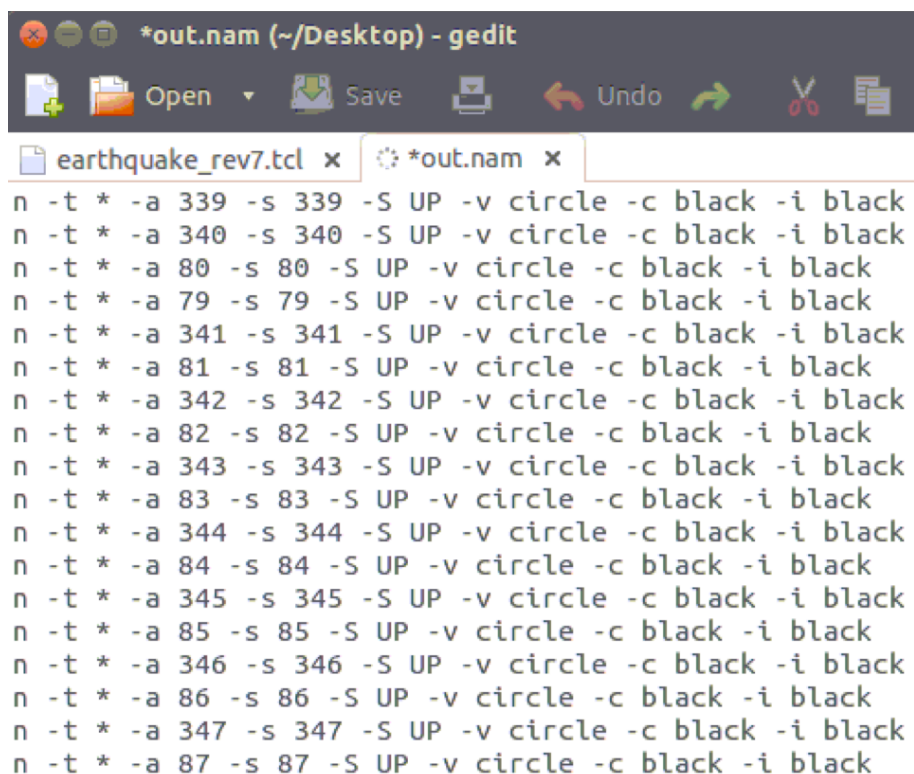
existen diversos simuladores en los cuales uno puede configurar enrutadores, conmutadores, bridges, etc. pero para nuestro estudio, necesitamos un simulador que sea más general y trabaje a nivel de nodos solamente y, de ser necesario, poderle agregar módulos según se requiera.

En simulador que se escogió para realizar este estudio, se llama Network Simulator versión 2 (NS2), es un simulador de redes a nivel general de código abierto y libre distribución. Está orientado a la investigación de varios protocolos en todas las capas del modelo OSI, tanto para redes cableadas, inalámbricas y móviles. El NS2 es un simulador basado en eventos, que comprende de dos lenguajes de programación: C++ (lenguaje de programación Orientado a Objetos) que es el núcleo del mismo ya que define su comportamiento interno y OTcl que es el encargado de establecer y configurar los parámetros de la topología actual y los eventos que ocurrirán en ella. Para establecer una topología en este simulador es necesario programar los objetos desde la capa de Aplicación hasta la capa de Enlace de Datos si es necesario, para lo cual se debe tener buenos conocimientos de programación, ya que NS2 no es un simulador que cuente con una GUI. Al no ser un

simulador de redes no tan amigable con el usuario, es necesario utilizar herramientas adicionales si se desea observar la topología de forma gráfica o si se desea realizar tablas en base a la información obtenida del simulador, unas de éstas herramientas son: NAM y Xgraph.

La herramienta NAM es utilizada como parte de NS2 ya que sirve para poder visualizar la topología que se ha creado y como fluyen los paquetes de datos por la misma. Ésta herramienta soporta el diseño de la topología (ubicación de nodos, longitud de los enlaces, etc.), la animación a nivel de paquete, y varias herramientas de inspección de datos. Para recrear todas las animaciones pertinentes, NAM se basa en un archivo de trazas, que básicamente es un archivo de texto en el que cada línea corresponde a una traza o registro, que contiene información para establecer la animación. En la Figura 4.1 se muestra un ejemplo de archivo de trazas que son útiles para NAM, estas trazas son más complicadas de interpretar para los usuarios, por lo que si se desean interpretar datos, es necesario recurrir a otro tipo de trazas. NAM es muy útil al momento de visualizar la topología y el flujo de paquetes, sin embargo existe

otra herramienta llamada Xgraph que sirve para graficar otros tipos de parámetros.



```

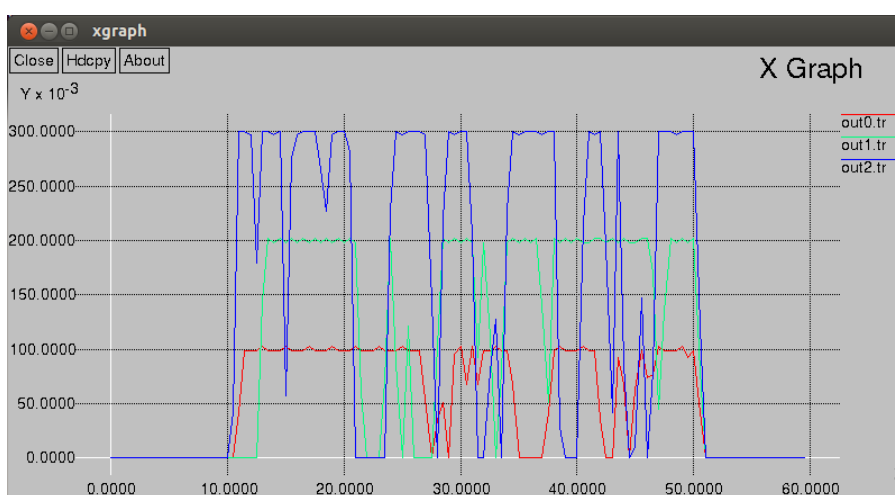
n -t * -a 339 -s 339 -S UP -v circle -c black -i black
n -t * -a 340 -s 340 -S UP -v circle -c black -i black
n -t * -a 80 -s 80 -S UP -v circle -c black -i black
n -t * -a 79 -s 79 -S UP -v circle -c black -i black
n -t * -a 341 -s 341 -S UP -v circle -c black -i black
n -t * -a 81 -s 81 -S UP -v circle -c black -i black
n -t * -a 342 -s 342 -S UP -v circle -c black -i black
n -t * -a 82 -s 82 -S UP -v circle -c black -i black
n -t * -a 343 -s 343 -S UP -v circle -c black -i black
n -t * -a 83 -s 83 -S UP -v circle -c black -i black
n -t * -a 344 -s 344 -S UP -v circle -c black -i black
n -t * -a 84 -s 84 -S UP -v circle -c black -i black
n -t * -a 345 -s 345 -S UP -v circle -c black -i black
n -t * -a 85 -s 85 -S UP -v circle -c black -i black
n -t * -a 346 -s 346 -S UP -v circle -c black -i black
n -t * -a 86 -s 86 -S UP -v circle -c black -i black
n -t * -a 347 -s 347 -S UP -v circle -c black -i black
n -t * -a 87 -s 87 -S UP -v circle -c black -i black

```

**Figura 4.1 Ejemplo de archivo de trazas para NAM**

El programa Xgraph es útil para realizar un análisis de parámetros como rendimiento, retardo, etc. utilizando gráficas. Xgraph dibuja las gráficas según los datos que, de forma similar a NAM y a NS2, se recuperan a través de un archivo de trazas. Puede mostrar hasta 64 conjunto de datos independientes y cada uno de ellos puede ser representado con su respectivo color y estilo de líneas. En la Figura 4.2 se puede observar un

ejemplo de gráficas usando Xgraph que corresponde a una prueba del retardo en una topología de pruebas. Este programa es muy útil para interpretar, de forma más profunda, el rendimiento de una red, sea por su respectivo retardo, ancho de banda utilizado, etc.



**Figura 4.2** Captura de gráficas usando Xgraph [10]

En esta subsección se mencionó el software de simulación de redes que se utilizó para este estudio, cuales son sus características y cuales son sus herramientas y programas relacionados para realizar investigaciones de varios protocolos, algoritmos, etc. En el siguiente subcapítulo se conocerá, en detalle, el sitio de pruebas sobre el cual nos basamos para hacer el diseño de la topología.



## **4.2 SITIO DE PRUEBAS**

Dado que no se iba a tratar de una topología común, se tuvieron que realizar visitas periódicas a la estructura para ir dilucidando como se iban a representar cada uno de los elementos dentro del edificio, para que exista concordancia entre la topología y el sitio físico.

El edificio en cuestión es el rectorado de la Escuela Superior Politécnica del Litoral (ESPOL), que se encuentra en el campus Gustavo Galindo ubicado en la vía Perimetral Km. 30.5, consta de cuatro pisos: el sótano, la planta baja, el primer piso y el segundo piso como se puede observar en la Figura 4.3. La afluencia de personas a este edificio es considerable, puesto que dentro del mismo existen muchos departamentos orientados a temas académicos así como también a temas administrativos, además de eventos y reuniones celebradas aquí, lo que incrementa la cantidad de personas que podrían estar dentro en algún momento dado. Aquí desarrollan sus funciones el rector de la institución, el Ing. Sergio Flores, máxima autoridad de la ESPOL, la vicerrectora académica, la Ing. Cecilia Paredes, sus colaboradores, entre otras autoridades, por lo que es muy importante tener un sistema de

evacuación eficiente si llegase a ocurrir alguna situación de emergencia. Además, funcionan los departamentos de: El Proyecto Ancón, Relaciones Externas, Asesoría Jurídica, Auditoría, Consejo Politécnico, Personal, Secretaría General, Comunicación Social, Suministros, Correo, STA, Unidad de Planificación de Infraestructura Física, Planificación Estratégica, Vicerrectorado Académico, Decanato de Investigación y Posgrados, Gerencia Jurídica, Rectorado, Dirección de Calidad, entre otros. Esta es la estructura que se escogió para realizar pruebas a los algoritmos para determinar cual tiene un mejor desempeño para determinar rutas de escape.



Fuente: <http://www.panoramio.com/photo/3740311>

**Figura 4.3 Vista general del Rectorado de la ESPOL [13]**

El edificio del rectorado contiene nueve salidas de emergencia entre los pisos del sótano y la planta baja, el cual es un buen número de salidas para evacuar a las personas si consideramos que es un edificio de solamente cuatro pisos. Son tres las salidas de emergencia están ubicadas a nivel del sótano: una a nivel de las bodegas, otra como salida del auditorio del rectorado y otra correspondiente a las puertas del comedor. También cuenta con seis salidas en la planta baja: tres puertas que tienen como destino la explanada principal del rectorado, una que se encuentra a la altura del departamento de Comunicación Social, y dos que sirven para salir a la explanada que es compartida con el edificio de Tesorería, sin embargo, solo una de ellas está disponible para todas las personas dentro del rectorado, ya que la otra sólo está disponible para trabajadores de la Secretaría Técnica Académica. Además existe un aspecto importante a ser mencionado con respecto a las salidas de emergencia: de las nueve salidas, supuestamente disponibles de acuerdo a los planos proporcionados, solo cuatro están habilitadas para todas las personas que se encuentren en el edificio ya que, por ejemplo, dos de las tres salidas principales en la primera planta, la del auditorio y el comedor generalmente se encuentran

cerradas con seguro y con ello, menos personas pueden llegar a ser evacuadas. De esta forma, aunque la estructura en sí tiene múltiples salidas de evacuación considerando que es un edificio relativamente pequeño, no todas se encuentran habilitadas, con lo que el número de vidas que puedan llegar a ser salvadas se reduce drásticamente.

Es muy importante conocer los detalles de la estructura que fue tomada como base para realizar la topología para nuestro estudio, ya que no solamente comprende de la cantidad de nodos y enlaces, sino también de la cantidad de personas que puedan llegar a estar en el edificio y cuanto tiempo les tomaría trasladarse de un sitio a otro.

### **4.3 SIMULACIÓN**

Antes de realizar las pruebas correspondientes al estudio planteado, fue necesario definir que rol iba a desempeñar cada objeto en la topología y como estos se correlacionaban con el edificio en sí y todos sus elementos físicos; en este subcapítulo se explicará dichas relaciones y traducciones del dominio real/físico al dominio lógico/red y viceversa.

Partiremos de definir lo que representa un nodo en nuestra topología, cuales son sus características y como este y sus atributos desempeñan un rol similar a alguna variable de la estructura física. El nodo en la implementación de NS2 tiene una forma muy particular de trabajar, ya que este objeto no contiene una cola de procesamiento, por el contrario, las colas se encuentran ubicadas en los enlaces, de lo que se encarga el nodo es de procesar un paquete a la vez y calcular las rutas a todos los nodos de la red. La función de procesar un paquete corresponde a abrir el paquete, ver cuál es el identificador del nodo destino (no se trabajó con direcciones IPv4 ya que no aplica para nuestro análisis) para colocar dicho paquete en la cola de un enlace. La otra función, calcular rutas, es en donde se ejecuta el algoritmo previamente establecido o programado con el fin que obtenga las rutas más óptimas a todos los nodos en la topología; se configuró para que se calcule de forma dinámica, es decir, que cada vez que exista algún nodo o enlace que se deshabilite, se recalculen las rutas, de lo contrario sólo se calcularían al inicio de la simulación y no se recuperaría la red a las fallas. Los nodos forman parte fundamental en nuestro estudio, ya que son los que se encargan del cálculo de rutas, procesan paquetes y envían los

paquetes a las colas de los enlaces, las cuales representan la capacidad de una sección determinada en el edificio.

Los enlaces, otro objeto fundamental en nuestra red, son utilizados para conectar nodos, ya que solamente se han considerado el uso de enlaces punto a punto, y entre sus funciones se encuentran la de agendar eventos, basándose atributos que posee estos objetos. Los enlaces utilizados en la topología fueron dúplex, que equivalen a dos enlaces simplex, ya que se debe representar que un paquete pueda enviarse de un nodo a otro y viceversa. Además, como se mencionó previamente, los enlaces son los encargados de manejar las colas para los paquetes, pero pese a que los enlaces son dúplex, en el simulador NS2 es necesario configurar la cola en los dos sentidos del enlace ya que, en algún otro contexto, puede ocurrir que un enlace en un sentido tenga cola con una capacidad determinada, pero en el otro sentido tenga una capacidad diferente; en nuestro caso, la cola en ambos sentidos tiene que ser la misma, ya que representa a la capacidad de una sección del edificio. Una función, por demás importante, de los enlaces es la de agendar dos eventos: la de la partida de un paquete del nodo transmisor y la del arribo del paquete al nodo

del siguiente salto; el primer evento se refiere al tiempo de transmisión de un paquete desde el punto de vista del nodo que lo envía, y durante ese tiempo (calculado utilizando el tamaño del paquete y el ancho de banda del enlace como se muestra en la ecuación 4.1) el nodo transmisor no puede enviar otro paquete por ningún enlace; el segundo evento que se debe agendar, corresponde al tiempo en el que el paquete arriba al nodo del siguiente salto y es calculado utilizando el tiempo de transmisión del paquete y a este se le suma el retardo del enlace (ver ecuación 4.2). Los enlaces son muy importantes, ya que NS2 al ser un simulador basado en eventos, necesita los enlaces que agendan los eventos y, en nuestro caso, emulan la conectividad entre nodos y el tiempo que se demoran los paquetes para moverse hasta hallar la salida.

$$t_{T_x} = \frac{PktSize}{BW} \quad (4.1)$$

$$t_{R_x} = t_{T_x} + \text{retardo} \quad (4.2)$$

Ya se ha mencionado brevemente a los paquetes, pero dentro de nuestro análisis son sumamente importantes debido a que estos representan a las personas dentro del edificio por lo que si dentro de nuestras simulaciones un paquete queda atrapado, o se descarta o no logra alcanzar una salida en un determinado tiempo, representará un vida perdida en la realidad. De forma lógica (en el simulador), los paquetes no necesitamos que sea un tipo de tráfico específico, como web, ftp, etc. por lo que decidimos utilizar un tipo de tráfico denominado CBR, ya que corresponde a un tipo de paquete estándar, en el que se puede configurar el tamaño del paquete y la cantidad máxima que transmitirá, por lo que es ideal para el entorno que queríamos recrear. Como se mencionó anteriormente el tamaño de los paquetes está estrechamente relacionado con el tiempo de transmisión de un paquete y por ende, con el tiempo de recepción del mismo, por lo que se decidió crear dos tipos de paquetes, cada uno con su propio tamaño, los cuales representan a personas sin discapacidad (un tamaño menor) y a personas con alguna capacidad disminuida (un tamaño mayor), ya que de esta forma existen paquetes que atraviesan más rápido los enlaces que otros, tal como sucede en la realidad. Para poder transmitir los paquetes, es necesario



contar con un protocolo de capa de Transporte (según el modelo OSI) y puesto que no requerimos acuses de recibos y una extrema fiabilidad de este tipo de protocolos, se optó por utilizar UDP ya que es un protocolo que no es orientado a la conexión y es más ligero en términos del tamaño de su encabezado. Los paquetes, de la forma que se ha implementado, representan fielmente a las personas dentro del rectorado, ya que incluso se ha considerado la existencia de personas con discapacidad que les impida seguir el mismo paso que una persona sin discapacidad.

En esta sección se mostró en detalle el funcionamiento y características de cada uno de los objetos que están inmersos en nuestra simulación, tales como los nodos, los enlaces, las colas, los paquetes y el protocolo de capa de Transporte a fin de demostrar representan fielmente el entorno real que pretendemos emular.

En el capítulo se analizó la herramienta que se utilizó para nuestro estudio, además de los detalles acerca de la estructura que se tomó en consideración al momento de realizar el diseño de la red y

finalmente cuales son los objetos que se utilizaron dentro de la simulación y como representaban al entorno real de un edificio.

## **CAPÍTULO 5**

### **5 PRUEBAS Y ANÁLISIS DE RESULTADOS**

Ahora se procederá a explicar el esquema del diseño de pruebas que se le realizaron a los algoritmos, además de explicar que tipos de datos se espera obtener de dichas pruebas, sea en cantidad de paquetes salvados, paquetes perdidos y en que tiempo se lo realizó y, finalmente, mostrar cuales fueron los resultados con los algoritmos que se probaron en nuestra topología.

#### **5.1 DISEÑO DE PRUEBAS**

Antes de mostrar cuales fueron los resultados de las pruebas realizadas a los algoritmos, es necesario conocer exactamente cuáles son los parámetros de dichas pruebas, como se las van a configurar y en que se van a basar, para poder determinar de

esta forma cuál va a ser el criterio de evaluación de rendimiento de dichos algoritmos.

El primer tipo de pruebas que se realizaron fueron las que correspondían a seleccionar aleatoriamente los nodos que quedarían inhabilitados en toda la estructura, en este caso en toda la topología. Con esto nos referimos a que de forma totalmente aleatoria se eligieron cuáles eran los nodos que iban a fallar, ya que contamos con 936 nodos que corresponden a la topología, más diez nodos entre los cuales se encuentran los nueve relacionados a todas las puertas de salidas que posee el edificio y un nodo final que es al cual todos los paquetes deberían llegar para ser considerados que se encuentran a salvo. Por tanto al generar, aleatoriamente, un número de índice que esté dentro del rango desde cero hasta 935, y darle la orden al simulador que deshabilite los nodos con dichos índices, estamos generando las fallas deseadas en la topología y podemos evaluar como responden los algoritmos ante las mismas. Además, tratando de realizar una analogía con la estructura real resulta que un evento similar puede ocurrir en una situación de emergencia como un temblor o un terremoto en la que varias zonas de la estructura queden deshabilitadas e

intransitables. Como podemos ver, este tipo de prueba es ideal para evaluar el desempeño de los algoritmos y también proporciona la emulación de una situación real que podría ocurrir en el edificio del rectorado.

Además del tipo de prueba mencionada previamente, se realizaron otros tipos de pruebas, las mismas que ya no eran de carácter aleatoria, pero que representan a otros tipos de catástrofes, como por ejemplo el caso de un incendio o alguna posible fuga de gas. Debido a esto, se procedió a implementar dichos eventos; en el caso de los incendios se procedió a escoger un nodo que represente a una sección del edificio en la que sea muy posible que en algún momento este sea el foco de un incendio y cada cierto tiempo que se vayan deshabilitando los nodos aledaños al nodo foco y pasado el mismo intervalo de tiempo, ocurra lo mismo con los nodos vecinos a estos, continuando con la misma idea por seis o siete intervalos de tiempo. En el caso de las fugas de gas la idea es muy similar, salvo pequeñas variaciones tales como diferentes intervalos de tiempo, así como también se designaron diferentes nodos para que sean foco para estos eventos, pero en lo demás es la misma idea. Para tener más claras las diferencias entre un

incendio y una fuga de gas se encuentra la Tabla 2 con más detalles al respecto.

**Tabla 2 Diferencias entre eventos: Incendios y Fugas de Gas**

<b>Evento</b>	<b>Nodo Foco</b>	<b>Tiempo en que nodos vecinos fallan</b>
<b>Incendio</b>	Suministros (PB)	45 [seg]
	Cafetería Vrect. (1 <sup>er</sup> Piso)	
<b>Fuga de Gas</b>	Cocina (Sótano)	30 [seg]
	Cocina (2 <sup>do</sup> Piso)	

Como podemos ver cambian los intervalos de tiempo así como los nodos que actúan como foco del evento, si bien los nodos foco podrían ser los mismo en ambos casos, se consideró más probables que se puedan dar en esos nodos los respectivos eventos ya que, por ejemplo, los incendios en el departamento de Suministros, es debido a que en esta oficina se encuentra un microondas y en el Vicerrectorado en el primer piso existe una cafetería, por lo que en ambos casos es posible un incendio, pero en el caso de las fugas de gas, por obvias razones, en ambas cocinas está latente la posibilidad de un evento como el mencionado. Ahora, con respecto a los diferentes intervalos de tiempo en el que los nodos aledaños pasan a estar deshabilitados es debido a que la propagación de una fuga de gas es ligeramente más rápida que la propagación de un

incendio. Estas pruebas son de gran ayuda para evaluar el desempeño de los algoritmos bajo una serie de eventos que podrían ocurrir en la estructura en algún momento dado.

## **5.2 PRUEBAS**

Después de haber definido en que consisten las pruebas a realizar, debemos de establecer que resultados o datos se pueden obtener de las mismas, con el fin de que nos ayuden a concluir cual algoritmo se desempeñó mejor y de esta forma, poder llegar a decantarnos por un algoritmo en específico que pueda ser recomendado para la implementación del proyecto KISHWAR de la ESPOL.

Primero, uno de los datos más importantes que se pueden determinar debido a las pruebas realizadas, es en torno a los paquetes/personas que son salvados bajo las instrucciones un algoritmo dado. Entre los datos que son arrojados gracias a los paquetes salvados, es la cantidad de paquetes totales salvados en general, así como también la cantidad de paquetes salvados pequeños y más grandes que, como se explicó previamente, representan a las personas sin discapacidad y con capacidades disminuidas, respectivamente. Además se puede conocer

desde donde iniciaron su recorrido los paquetes que se salvaron, y también cual puerta de salida cruzaron para poder desalojar el edificio; estos datos son muy importantes para nuestro estudio debido a que es necesario tener una idea aproximada de por donde tuvieron que transitar los paquetes para poder llegar al nodo que representa haber salido del edificio a salvo. Otros parámetros que se pudieron obtener fueron los promedios de paquetes salvados a lo largo de las diversas pruebas que se realizaron; esto es debido a que al ser la naturaleza de las pruebas de carácter aleatorio, con una sola prueba no es posible determinar si ese comportamiento corresponde a la media de los valores que debería de arrojar según el algoritmo utilizado, o es un evento aislado, por lo que es necesario calcular la media y concluir en base a estos valores. Como vemos, se pueden obtener muchos datos de los paquetes salvados en las pruebas realizadas, pero también existe la otra cara de la moneda, que en este caso son los paquetes que no lograron llegar al nodo destino fueron descartados en el camino.

Los datos obtenidos en torno a los paquetes atrapados en la topología o que fueron descartados son muy similares a los



datos que se pueden obtener de los paquetes salvados, pero con la gran diferencia que estos datos corresponden a puntos en contra para el algoritmo que se esté empleando para esas pruebas. De igual forma que los paquetes salvados, se puede llegar a conocer la cantidad de paquetes totales atrapados en general, así como llegar a tener un análisis más granular conociendo cuántos de ellos eran paquetes que representaban a personas con discapacidad y sin discapacidad, si bien se podría intuir que el número de paquetes perdidos que representen a personas con discapacidad sería mayor, es necesario conocer en que porcentaje lo es. Además, gracias a las trazas generadas por NS2, se puede determinar desde donde partieron los paquetes que finalmente no pudieron llegar al nodo destino, pero más aún, podemos conocer en que nodo (zona del edificio) quedaron atrapados y hacia dónde se dirigían para después poder cuantificar en que zona del rectorado se quedaron más personas atrapadas. Finalmente, de similar forma al análisis de los paquetes salvados, es necesario conocer todos los valores promedio de los datos que se puedan llegar a obtener para garantizarnos cual es el rendimiento promedio del algoritmo para los paquetes descartados y de esa forma asegurarnos que no estamos tratando solamente con

eventos aislados. Estas estadísticas son igual o más importantes que las de los paquetes salvados, puesto que estos datos son puntos negativos en el rendimiento de los algoritmos y son los que más se deben tener en consideración al momento de tomar la decisión de cuál implementar, ya que estaríamos hablando de vida que se perderían durante la evacuación del edificio.

Es necesario considerar el tiempo de acción de los algoritmos ya que el tiempo en el que un edificio debe ser evacuado durante una situación de emergencia es finito y no puede prolongarse indefinidamente puesto que conforme pasen los minutos la eficacia de la evacuación va mermando. Es por esto que se ha establecido un tiempo quince minutos como tiempo máximo en que la estructura deberá ser evacuada en su totalidad y pasado este tiempo los paquetes que aún se encuentren en la estructura, podrán ser considerados como perdidos, pese a que en situaciones de emergencia pueden ocurrir casos aislados en el que se exceda este tiempo de evacuación sin problema alguno. Pero ante esta limitante, se la puede considerar como otro punto con el que los algoritmos serán evaluados, debido a que no es lo mismo que un algoritmo

tenga un gran desempeño y minimice los paquetes perdidos pero lo haga en un tiempo mucho mayor en comparación con otro que tal vez evacue menos personas pero en un tiempo considerablemente menor. El tiempo de evacuación es un parámetro a tener muy en cuenta dado que es fundamental en situaciones de emergencia poder desalojar a las personas lo más rápido posible y para esto, el algoritmo de enrutamiento para encontrar rutas de evacuación debe reaccionar de forma veloz y recuperarse de inmediato ante una red que va cambiando conforme pasa el tiempo.

Esta subsección presentó cuáles son los datos que se pueden obtener a partir de las trazas que genera automáticamente el simulador de redes NS2, y partiendo de los mismos, luego de ser procesados, se puede evaluar el desempeño de los algoritmos de una forma muy precisa con el fin de concluir cuál tendría un mejor desempeño para ser utilizado en el proyecto KISHWAR de la ESPOL.

### 5.3 ANÁLISIS DE RESULTADOS

Después de conocer en detalle cuáles son las pruebas que se iban a realizar a la topología que representa al rectorado de la ESPOL y mostrar cuáles son los datos que pueden ser obtenidos luego de haber ejecutado dichas pruebas, se va a proceder a informar acerca de los resultados que se obtuvieron luego de realizar las pruebas antes mencionadas con los siguientes algoritmos: Bellman-Ford y Dijkstra.

Partiremos nuestro análisis con los resultados obtenidos de las simulaciones utilizando el algoritmo de Dijkstra. En el Anexo A se muestra un resumen de los resultados arrojados por el simulador teniendo como base el algoritmo antes mencionado. Como se puede observar, los resultados son muy alentadores con respecto al porcentaje de personas que logran ser evacuadas utilizando este algoritmo, ya que se puede hablar de un porcentaje promedio de 75.6% con una confiabilidad del 95% de personas que logran ser evacuadas. En contraste, podemos apreciar que la cantidad promedio de personas que no lograron salir del edificio es inferior al 25%. Ahora procederemos a analizar los resultados con Bellman-Ford.

De forma similar, realizamos otras simulaciones considerando los mismos eventos y parámetros, pero esta vez, para cuantificar el rendimiento del algoritmo Bellman-Ford. En el Anexo B se muestra un resumen de los resultados arrojados por el simulador teniendo como base el algoritmo antes mencionado. Como se puede observar, los resultados son similares a los obtenidos con Dijkstra con respecto al porcentaje de personas que logran ser evacuadas utilizando este algoritmo, ya que se puede hablar de un porcentaje promedio del 75.9% con una confiabilidad del 95% de personas que logran ser evacuadas, por consiguiente es muy pequeño el porcentaje de personas que no lograron salir del edificio. Su rendimiento en cuanto a números al parecer es ligeramente superior a los de Dijkstra, sin embargo hay que tener en cuenta que durante las simulaciones, le tomo mucho mas tiempo completarlas, lo cual a la hora de implementarlo, podría traducirse en una sobrecarga de recursos del equipo en donde se vaya a implementar. Este análisis fue con respecto a los algoritmos como tal, pero otra parte muy importante es, sin duda alguna, cuánto tiempo duró la evacuación de las personas dentro del edificio.

El tiempo que se tomó cada algoritmo para poder evacuar el edificio de la mejor forma posible es muy importante al momento de decidirnos a la hora de implementar un algoritmo en particular. Pero al final de las simulaciones realizadas, resulta interesante darnos cuenta que con ningún algoritmo se excedió del tiempo que inicialmente habíamos establecido como máximo para evacuar la estructura, que era de quince minutos a veinte minutos. Con ambos algoritmos no se llegó incluso al minuto trece, en el caso de Dijkstra, le tomó en promedio 12.22 minutos evacuar el edificio; por otra parte, con Bellman-Ford la evacuación tomó en promedio 12.26 minutos. Como se puede observar, es muy pequeña la diferencia entre ambos tiempos en promedio, sin embargo, ya que nuestro análisis pretende ser lo más preciso posible, en este rubro hay que ponderar el desempeño de Dijkstra, por más que no sea muy amplia esta diferencia.

Las simulaciones realizadas y analizadas en este subcapítulo fueron de vital importancia al momento de tomar la decisión de que algoritmo se debería recomendar para la implementación del proyecto KISHWAR, si bien es cierto que ningún algoritmo sacó amplias diferencias de rendimiento por sobre el otro

algoritmo, dichas diferencias son lo suficientemente significativas para tomar concluir esta investigación.

En este capítulo se trataron temas más orientados a las pruebas como tal y al análisis de los resultados de las simulaciones, se explicó en que consistían las pruebas, que situaciones de emergencia se iban a representar, cuáles son los datos que se iban a obtener de las simulaciones y finalmente la interpretación de los datos.

## CONCLUSIONES Y RECOMENDACIONES

Como conclusiones a este trabajo de investigación se tiene lo siguiente:

Si bien es cierto que ambos algoritmos tuvieron desempeños muy similares en casi todos los rubros que se analizaron; sin embargo, si hubo pequeñas diferencias que son importantes de mencionar ya que estamos hablando de un sistema que esta pensado para salvar la mayor cantidad de vidas durante una situación de emergencia. Con esto en mente, podemos concluir que Dijkstra es el algoritmo que se recomienda para que sea utilizado en el proyecto KISHWAR debido a las siguientes razones:

1. Estadísticamente el algoritmo Dijkstra tiene un mejor porcentaje promedio de personas evacuadas en todos los escenarios, a excepción del que representaba a un terremoto, en comparación con Bellman-Ford tal como se muestra en los anexos A y B.



2. Aunque ambos algoritmos hacen posible la evacuación del edificio en un tiempo inferior a los 15 minutos que se estipuló como tiempo límite, el algoritmo Dijkstra lo hizo en un tiempo promedio menor en comparación al tiempo promedio requerido por Bellman-Ford para desalojar la estructura, sin embargo, son estadísticamente similares.
3. Dado que la desviación estándar de los datos es inferior al 5% de la media en todos los casos, se puede inferir que la confiabilidad de las pruebas con Bellman-Ford y Dijkstra es estadísticamente igual.
4. Finalmente, pese a que depende de la robustez del equipo en donde se implementará, igualmente hay que considerar que Bellman-Ford requirió mucho más tiempo para completar las simulaciones, por lo que podemos deducir que tiene una mayor carga de procesamiento, en comparación a Dijkstra.

Entre las recomendaciones que podemos hacer al momento que se implemente el proyecto KISHWAR están:

1. Se recomienda tratar de utilizar SED mixtos, ya que es recomendable que ciertos indicadores cambien de estado automáticamente, sin necesidad de que el controlador principal envíe la orden explícita de cambio de estado.
2. Se recomienda tratar de colocar una escalera externa como salida de emergencia para poder tener otra salida disponible y también otro punto por el cuál puedan ingresar al edificio el personal de emergencias (bomberos, paramédico, etc.).

## Anexo A – Tabla de resultados de las simulaciones con el algoritmo Dijkstra

TEST	EVENTO	ALGORITMO	PUERTAS	T [MIN]	% SALVADOS	% PERDIDOS
TEST 01	Terremoto	Dijkstra	Completas	12,14	71%	29%
TEST 02	Terremoto	Dijkstra	Completas	12,04	76%	24%
TEST 03	Terremoto	Dijkstra	Completas	12,36	74%	26%
TEST 04	Terremoto	Dijkstra	Completas	12,36	75%	25%
TEST 05	Terremoto	Dijkstra	Completas	12,15	77%	23%
TEST 11	Incendio VRec	Dijkstra	Completas	12,16	77%	23%
TEST 12	Incendio VRec	Dijkstra	Completas	12,36	76%	24%
TEST 13	Incendio VRec	Dijkstra	Completas	12,36	77%	23%
TEST 14	Incendio VRec	Dijkstra	Completas	12,36	76%	24%
TEST 15	Incendio VRec	Dijkstra	Completas	12,14	76%	24%
TEST 21	Incendio Suministros	Dijkstra	Completas	12,36	75%	25%
TEST 22	Incendio Suministros	Dijkstra	Completas	12,36	76%	24%
TEST 23	Incendio Suministros	Dijkstra	Completas	12,36	76%	24%
TEST 24	Incendio Suministros	Dijkstra	Completas	12,16	76%	24%
TEST 25	Incendio Suministros	Dijkstra	Completas	12,02	76%	24%
TEST 31	Fuga Gas Cocina	Dijkstra	Completas	12,16	75%	25%
TEST 32	Fuga Gas Cocina	Dijkstra	Completas	12,14	75%	25%
TEST 33	Fuga Gas Cocina	Dijkstra	Completas	12,36	75%	25%
TEST 34	Fuga Gas Cocina	Dijkstra	Completas	12,14	75%	25%
TEST 35	Fuga Gas Cocina	Dijkstra	Completas	12,10	75%	25%
TEST 41	Fuga Gas Rectorado	Dijkstra	Completas	12,16	76%	24%
TEST 42	Fuga Gas Rectorado	Dijkstra	Completas	12,16	76%	24%
TEST 43	Fuga Gas Rectorado	Dijkstra	Completas	12,36	76%	24%
TEST 44	Fuga Gas Rectorado	Dijkstra	Completas	12,16	76%	24%
TEST 45	Fuga Gas Rectorado	Dijkstra	Completas	12,14	76%	24%

## Anexo B – Tabla de resultados de las simulaciones con el algoritmo Bellman-Ford

TEST	EVENTO	ALGORITMO	PUERTAS	T [MIN]	% SALVADOS	% PERDIDOS
TEST 51	Terremoto	Bellman-Ford	Completas	12,36	79%	21%
TEST 52	Terremoto	Bellman-Ford	Completas	12,36	79%	21%
TEST 53	Terremoto	Bellman-Ford	Completas	12,13	80%	20%
TEST 55	Incendio VR	Bellman-Ford	Completas	12,36	75%	25%
TEST 56	Incendio VR	Bellman-Ford	Completas	12,16	75%	25%
TEST 57	Incendio VR	Bellman-Ford	Completas	12,36	75%	25%
TEST 59	Incendio Suministros	Bellman-Ford	Completas	12,15	75%	25%
TEST 60	Incendio Suministros	Bellman-Ford	Completas	12,16	75%	25%
TEST 63	Fuga Gas Cocina	Bellman-Ford	Completas	12,15	74%	26%
TEST 64	Fuga Gas Cocina	Bellman-Ford	Completas	12,16	74%	26%
TEST 67	Fuga Gas Rectorado	Bellman-Ford	Completas	12,36	75%	25%
TEST 68	Fuga Gas Rectorado	Bellman-Ford	Completas	12,36	75%	25%

## BIBLIOGRAFÍA

- [1] Pauls, J. L., "The movement of people in buildings and design solutions for means of egress", Fire Technology, Vol.20, No.1, 1984
- [2] Casadesús, S.; Garriga, F., "Optimal time of building evacuation considering evacuation routes", EJOR, Vol.192, Issue 2, 2009
- [3] Mejía, R; Bernal, J.; López, D., Análisis de tipos de Gestores de Colas en NS-2.", Revista TIA, Octubre 15 de 2013
- [4] Gento, M.; López B.; Posada M., "Simulación del desalojo de edificios. Situación en España: E.T.S. Ingenieros Industriales (Universidad de Valladolid)", V Congreso de Ingeniería de Organización, 2003.
- [5] Lovas, G., "On performance measures for evacuation systems", European Journal of Operational Research, Vol.85, Issue 2,1995

- [6] Lu, Q.; Huang, Y.; Shekha S., "Evacuation Planning: A Capacity Constrained Routing Approach",  
[http://www.cc.ntut.edu.tw/~cmliu/SD/SD\\_F06g/papers/SN2\\_lu\\_george\\_shekhar.pdf](http://www.cc.ntut.edu.tw/~cmliu/SD/SD_F06g/papers/SN2_lu_george_shekhar.pdf), 2005
- [7] Bellman, Richard, "On a routing problem", Quarterly of Applied Mathematics, 1958
- [8] Dijkstra, E. W., "A note on two problems in connexion with graphs", Numerische Mathematik 1, 1959
- [9] Rodríguez, G., "Proyecto Getaway: Nuevos sistemas de evacuación en emergencias", <http://prevenblog.com/proyecto-getaway-nuevos-sistemas-de-evacuacion-en-emergencias/>, Marzo 24 de 2014
- [10] Avendaño, A., "Instalando y ejecutando NS2"., <http://aveoctavo.blogspot.com/2013/03/laboratorio-de-tele-ns2.html>, Marzo 4 de 2013
- [11] Anónimo, "Señalamiento Ruta Evacuación Derecha", [http://www.codigofuego.com/detalles\\_proteccion-civil-SENALAMIENTO-RUTA-DE-EVACUACION-DERECHA-,117,10,4.htm](http://www.codigofuego.com/detalles_proteccion-civil-SENALAMIENTO-RUTA-DE-EVACUACION-DERECHA-,117,10,4.htm), 2015

- [12] Anónimo, “Single Face Green LED Edge-Lit Exit Sign”.  
[http://www.aliexpress.com/store/product/Single-Face-Green-LED-Edge-Lit-Exit-Sign/929161\\_1495439538.html](http://www.aliexpress.com/store/product/Single-Face-Green-LED-Edge-Lit-Exit-Sign/929161_1495439538.html), fecha de consulta Abril 2015
- [13] Anónimo, “Rectorado ESPOL”,  
<http://www.panoramio.com/photo/3740311>, Agosto 5 de 2007