ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"IMPLEMENTACIÓN DE UN SOFTWARE DE UBICACIÓN PARA LAS DIFERENTES EDIFICACIONES DEL CAMPUS GUSTAVO GALINDO DE LA ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL UTILIZABLE EN DIVERSOS DISPOSITIVOS CON ACCESO A INTERNET"

INFORME DE PROYECTO DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

PRESENTADA POR:

ANDRES ENRIQUE BARRETO ROSADO
SILVIA SOFIA CHIRIBOGA FERNANDEZ

GUAYAQUIL – ECUADOR
2012

AGRADECIMIENTO

Agradezco a Dios, a mi familia por creer y confiar siempre en mí, apoyándome incondicionalmente y dándome soporte en todas las decisiones que he tomado. Agradezco a las personas que me han acompañado en toda esta etapa y quienes siempre me dieron una mano, mis amigos. A nuestro director de tesis Ing. Xavier Ochoa por estar siempre dispuesto a brindarme su tiempo, y conocimientos no solo para el desarrollo de este proyecto sino también a lo largo de mi carrera universitaria.

Andrés Enrique Barreto Rosado.

A Dios por guiarme en todo este tiempo y por tantas bendiciones. A mis padres por todo el apoyo que me dieron para poder alcanzar la meta de obtener una carrera universitaria. A mis amigos que me acompañaron y me apoyaron en todo momento en estos años de mi carrera universitaria. A Xavier, nuestro director de tesis por apoyarnos en todas las dudas que tuvimos y guiarnos en este tiempo de desarrollo de la tesis.

Silvia Sofía Chiriboga Fernández.

DEDICATORIA

El esfuerzo y dedicación que he puesto en esta tesis y a lo largo de mi carrera va dedicada enteramente mi madre, Miriam Barreto quien ha sido mi guía e inspiración desde siempre.

Andrés Enrique Barreto Rosado.

A mi familia y amigos, por su apoyo incondicional en todo lo que he realizado, gracias a ellos estoy culminando esta etapa de mi vida.

Silvia Sofía Chiriboga Fernández

TRIBUNAL DE SUSTENTACIÓN

Ing. Sara Ríos

SUBDECANO DE LA FIEC

PRESIDENTE DEL TRIBUNAL

Dr. Xavier Ochoa Chehab

DIRECTOR DE PROYECTO

Dra. Katherine Chiluiza

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este proyecto de grado, nos corresponde exclusivamente; y, el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

_____

Andrés Enrique Barreto Rosado

_____

Silvia Sofía Chiriboga Fernández

RESUMEN

Para el presente proyecto de graduación se ha desarrollado un sistema de ubicación virtual para el campus Prosperina de la Escuela Superior Politécnica del Litoral, que les permitirá a estudiantes y visitantes guiarse dentro del mismo.

Este sistema ayudará a las personas que no se encuentran familiarizadas con el campus a llegar a tiempo a lugar donde desean evitando extraviarse.

En el primer capítulo se explica por qué es necesario este sistema y se da ejemplos de sistemas similares.

En el segundo capítulo se compara diferentes soluciones de hardware y software para tener una idea clara de lo que se necesita para que el sistema funcione de manera óptima.

En el tercer capítulo se explica con más detalle cómo está conformado el sistema y de qué manera funciona.

En el cuarto capítulo se detallan las pruebas que se realizaron luego de culminar con el desarrollo y se analizan los resultados comparándolos con los objetivos que se tenían antes de comenzar con el desarrollo.

Finalmente se presentan las conclusiones y recomendaciones, adjuntando como apéndice el manual de usuario y otros detalles importantes.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

ABREVIATURAS

AS3 Action Script 3

C Lenguaje de programación

C++ Lenguaje de programación

E/S Entrada/Salida

ESPOL Escuela Superior Politécnica del Litoral

GPS Global Positioning System

HTML HyperText Markup Language

IDE Integrated development environment

iOS iPhone Operative System

LCD Liquid crystal display

MXML Macromedia eXtensible Markup Language

PCA Principal component analysis

INTRODUCCIÓN

La Escuela Superior Politécnica del Litoral recibe cientos de nuevos estudiantes al inicio de cada término lectivo y un sinnúmero de visitas diarias de personas que no se encuentran familiarizadas con el campus Prosperina, el cual actualmente aloja a los estudiantes de casi todas las carreras.

Por causa de que el campus es muy extenso y consta de variedad de edificaciones las personas que no están familiarizadas con este suelen extraviarse y no llegar a tiempo ya sea a clases o a reuniones, y no existe alguna herramienta que ayude a solucionar este problema. La creación de un sistema de ubicación virtual permitirá a estudiantes y visitantes acceder de manera fácil y en cualquier momento a un plano completo del campus Prosperina de la ESPOL y buscar la ruta hacia el lugar que necesiten llegar.

CAPÍTULO 1

1. Análisis del problema

1.1. Definición del problema

Existen lugares que son tan amplios que muchas veces se necesita de una guía para que las personas puedan movilizarse o dirigirse a un determinado lugar sin perder tiempo.

En varios lugares esto se ha solucionado colocando un mapa o plano detallado del lugar, pero estos no muestran una guía entendible o adecuada para todos los usuarios. Por esto muchos establecimientos han optado por implantar un mapa interactivo, de fácil uso, entendible para el usuario y que muestra información más detallada y precisa. Un ejemplo de estos sistemas seria los GPS ya que con estos los usuarios saben dónde está su destino y tienen una opción más fácil y rápida de llegar a este [1].

En el caso de la ESPOL no cuenta con ningún tipo de guía que ayude a los visitantes, alumnos o personas en general a dirigirse a un determinado lugar, esto es muy importante puesto que el campus Prosperina de la ESPOL es un lugar muy amplio y cuenta con varias

construcciones factores que pueden hacer que una persona se pierda fácilmente o que no llegue a tiempo a su destino.

Nosotros como estudiantes al entrar por primera vez a la Universidad estuvimos en una situación como esta y siempre nos pareció importante que en el campus exista algún tipo de guía en diferentes puntos del mismo, como aún no existe, hemos tenido la idea de desarrollar un sistema de guía interactivo que pueda ayudar a las personas de manera sencilla y exacta.

1.2. Descripción de soluciones similares

Existen soluciones similares a nuestro sistema, entre las cuales podemos numerar las siguientes.

1.2.1. Sistema de navegación en interiores (indoor navigation system)

Este sistema carga el plano del lugar donde va a empezar la navegación, muestra la posición en que se encuentra el usuario. Mientras el usuario realiza sus movimientos, el punto de referencia se mueve por el plano de acuerdo a los movimientos de este [1].

Figura 1. Indoor navigation system (demostración)

1.2.2. Navegación peatonal multisensorial interiores / exteriores (multi-sensor pedestrian indoor / outdoor navigation)

Este video muestra un algoritmo de filtro de partículas que se une a varios sensores de navegación personal, que incluye un receptor GPS, una brújula electrónica y una plataforma inercia de pie de montaje de bajo costo. Al aire libre el receptor GPS proporciona la posición exacta, pero las señales del satélite se desactivan en el interior. En el ambiente interior la combinación de la detección del

paso de inercia y el mapa de la construcción permite navegar sin problemas y sin errores de acumulación [2].

Figura 2. Multi-sensor pedestrian indoor / outdoor navigation (demostración)

1.3. Justificación del problema

Este software se está desarrollando ya que en la universidad se vio la necesidad de implementar un sistema para que los visitantes y alumnos de esta sepan donde se encuentran los principales edificios y por donde tienen que ir si se dirigen a diferentes lugares, esto sucede ya que el campus de la ESPOL es un campus muy extenso y con muchos edificios.

Como desarrolladores del sistema estamos seguros de que una vez que este se ponga en funcionamiento y esté disponible para todos será de gran ayuda en especial para alumnos nuevos y personas que no se encuentran familiarizadas con el campus.

Entre las ventajas que se pueden obtener al desarrollar y poner en funcionamiento de este sistema tenemos:

- Los estudiantes y visitantes tendrán una manera sencilla para guiarse dentro del campus.
- Gracias a una interfaz de fácil uso el usuario podrá obtener la información que necesita de manera rápida y efectiva.
- Este sistema será muy útil en los inicios de cada término pues ingresan nuevos estudiantes que no están familiarizados con el campus, usando este sistema pueden dirigirse a su destino sin perder tiempo.

CAPÍTULO 2

2. Análisis de la solución

En este capítulo se encuentra el alcance del sistema y también se analizan las posibles soluciones de hardware y software usado para el desarrollo del sistema de ubicación, soluciones que serán detalladas a continuación.

2.1. Análisis comparativo de las soluciones de hardware

El sistema es capaz de funcionar en dispositivos con acceso a la red y navegadores que soporten Flash, pero los requerimientos son de que este funcione en una estación o kiosco fija, por esto en esta sección se realiza una comparación de componentes de hardware que al ser usados en conjunto serán los que conformen el kiosco.

El sistema de ubicación necesita el uso de componentes de hardware que permitan que la interacción entre el sistema y el usuario sea intuitiva y de fácil uso. Para lograr esto se ha decidido eliminar el uso de teclado y mouse y remplazarlos por pantallas táctiles. A continuación se detalla que opciones fueron consideradas para ser la parte principal del kiosco donde se ejecutara el sistema de ubicación.

2.1.1. Tablets

Una tablet es un tipo de computadora de tamaño reducido, cuentan con una pantalla LCD sobre la cual el usuario puede interactuar por medio del uso de sus dedos o un lápiz especial denominado stylus [3]. Actualmente estos dispositivos son muy populares ya que han sido desarrollados con muchas posibilidades en el campo multimedia.

En el mercado existen una variedad de marcas y sistemas operativos con las que vienen preinstalados, entre las principales: Windows, iOS de Apple y Android.

Para nuestro sistema se necesitaría una tablet que utilice un sistema operativo que cuente con soporte flash lo cual descarta cualquier solución proveniente de Apple.

Figura 3. Samsung Galaxy Tab 10.1 [4]

2.1.2. Pantalla táctil

La opción más viable es la de usar una pc que cuente con una pantalla táctil que remplace al teclado y mouse y con soporte para adaptador de red. Es la mejor opción porque se puede elegir entre una amplia gama de resolución, tamaño y tecnología.

Una pantalla táctil puede ser de tecnología resistiva o capacitiva, las cuales son usadas actualmente.

Una pantalla resistiva, funciona por medio de presión, la pantalla detecta la pulsación por medio de dos capas que entran en contacto al detectar presión lo cual produce un cambio en la corriente eléctrica. Son muy económicas y resistentes al polvo y agua, sin embargo el uso de múltiples capas hacen que el brillo de la pantalla

se reduzca aproximadamente en un 25%. Estas pantallas pueden ser usadas con los dedos o con un stylus.

La principal desventaja de esta pantalla es que la respuesta parece menos intuitiva y más lenta y no puede detectar varias pulsaciones o gestos y luego de un número determinado de toques esta comienza a perder durabilidad, lo cual lo hace descartarla como opción [5] [6].

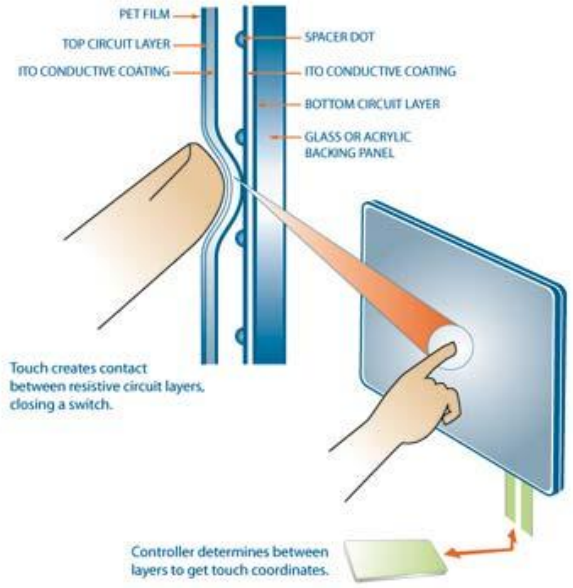


Figura 4. Funcionamiento de una pantalla resistiva [6]

Una pantalla capacitiva, por su tecnología deben ser manejadas mediante el dedo o un objeto que disponga de capacitancia, lo cual deja a los stylus normales inusables. La ventaja de esta tecnología es que pueden detectar diversas pulsaciones simultáneas o gestos

que no requieren presión, solo basta con deslizar el dedo lo que permite diversas formas de interactuar con ellas y una mejor experiencia para el usuario además de ser pantallas que no pierden durabilidad con el uso.

Esta tecnología también posee sus desventajas, una de ellas es que no pueden ser utilizadas si las manos están usando guantes o algún tipo de accesorio que cubra los dedos, también es una tecnología un poco más costosa con relación a las pantallas resistivas [5] [6].

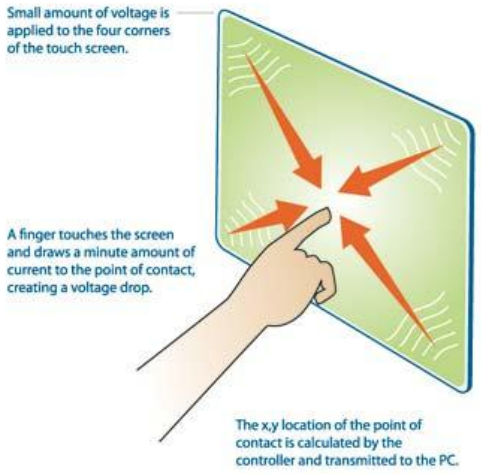


Figura 5. Funcionamiento de una pantalla capacitiva [6]

2.2. Análisis comparativo de las soluciones de software

Existen varias soluciones que cuentan de herramientas útiles para el desarrollo de un sistema con las características que se requerían. A continuación se detalla los frameworks y lenguajes de programación tomados en cuenta para la realización de este proyecto.

2.2.1. Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a inicios de los 90. Este lenguaje es algo familiar con C y C++ pero algo más simple, ya que elimina herramientas de bajo nivel causantes de muchos errores [7].

Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión [8].

Este lenguaje fue considerado para el uso del sistema porque su principal ventaja es la facilidad en desarrollar sobre el mismo y su portabilidad.

Su desventaja principal es que carece de herramientas para el desarrollo de aplicaciones con entorno grafico e intuitivo.

2.2.2. HTML 5

HTML5 es una colaboración entre el Consorcio de la World Wide Web (W3C) y el Web Hypertext Application Technology Working Group (WHATWG).

Entre la característica más destacadas de HTML5 están la posibilidad de crear dibujos en un canvas, reproducción de audio y video y soporte para almacenamiento local.

La desventaja de HTML5 es que aún no es un estándar web y no existe un navegador completamente compatible con este, aunque los principales navegadores como Internet Explorer, Google Chrome y Mozilla Firefox entre otros continúan añadiendo nuevas características de HTML5 en cada actualización [9].

2.2.3. Flex Framework

Flex es un conjunto de librerías para el desarrollo de interfaces cuyo funcionamiento depende de AS3 (Action Script 3) como leguaje de programación.

AS3 es un lenguaje de programación que es interpretado por Flash Player, este lenguaje cuenta con muchas características de un lenguaje de programación orientado a objetos.

La mayoría de aplicaciones Flex contienen MXML para crear la interface o UI y AS3 para crear la lógica. A final de cuentas en tiempo de compilación Flex toma el MXML y lo usa para generar código fuente en AS, y entonces lo compila como si hubiera sido escrito en AS3 [10], lo que significa que la aplicación desarrollada se ejecuta en Flash.

Para el desarrollo de aplicaciones realizadas con Flex se utiliza FlexBuilder, IDE que está basado en Eclipse.

Su principal ventaja es que permite desarrollar interfaces llamativas de manera rápida y eficiente pero AS3 es un lenguaje de programación que no cuenta con la sencillez de Java.

2.3. Librería de Software para el sistema de navegación

Las librerías utilizadas en el sistema de ubicación son un tipo de Application Software Interface (API). Los API son métodos específicos determinados por un sistema operativo o por una aplicación mediante los

cuales un programador puede realizar requerimientos al sistema operativo o a otra aplicación [11].

Además de las librerías por defecto de Flex 4.5 se utilizaron las siguientes librerías pre compiladas.

2.3.1. FaceRecognitionLib.swc

Librería utilizada para reconocimiento facial que funciona sobre Flex 3.5 y versiones superiores. Ésta librería usa el **método Eigenface** (PCA) para realizar el reconocimiento facial. Esta librería se puede usar para el reconocimiento de cualquier tipo de objetos no solo caras [12].

2.3.1.1. Método Eigenface

La idea de aplicar análisis de componentes principales (PCA) para representar imágenes de caras en una dimensión baja fue introducida por primera vez por Lawrence Sirovich y Robert M. Kirby en 1987. Comenzando con un conjunto original de imágenes de caras calculaban el mejor sistema de coordenadas para comprimir la imagen, donde cada coordenada es una imagen que ellos llamaban eigenpictures (imágenes propias).

Esta idea fue tomada poco después por Matthew Turk y Alex Pentland quienes desarrollaron el método de reconocimiento de Sirovich y Kirby y lo reformularon con la denominación Eigenfaces, (caras propias). Cuando se habla PCA y Eigenfaces se cree que son indistintos, lo cual no es del todo correcto: El método PCA es un método general de análisis de datos, por otra parte Eigenfaces es un método de reconocimiento que utiliza PCA con alguna variación [13].

2.3.1.2. Cálculo de eigenfaces

Sea una imagen de un rostro dada por una matriz NxN. Puede considerarse como un vector de dimensión N^2, de manera que una imagen típica (pequeña) de 256 por 256 será un vector de dimensión 65536 o de forma equivalente, un punto en un espacio de 65536 dimensiones. Un conjunto de imágenes trazara una nube de puntos en este gran espacio.

La idea de PCA es encontrar la base que mejor expresan la distribución de las imágenes de las caras dentro del espacio completo, al que Kirby y Sirovich llaman espacio de las imágenes. Estos vectores describen la base del subespacio de

las imágenes de rostros, el espacio de las caras. Cada vector de dimensión $d = N^2$ describe una imagen NxN, y es una combinación lineal de los vectores base del subespacio. Como estos vectores son los vectores propios de la matriz de covarianza correspondiente al espacio original de las imágenes, y como son parecidas a una cara, se les llama Eigenfaces.

Sea el conjunto de imágenes de caras $x_1, x_2, x_3 \dots x_m$ (considerando vectores columna de dimensión d podemos contruir la matrix X de dimensiones dxm). La media del conjunto (o la cara media) se define $u = \frac{1}{m}\sum_{i=1}^{m} x_i$. Cada cara difiere de la media por el vector $x_i - \mu$. Este conjunto de vectores grandes es sometido a PCA lo cual buscará un conjunto de m vectores ortonormales $u_k$que describe la distribución de los datos.

Los vectores u_ky los escalares $\lambda_k$son los vectores y valores propios respectivamente de la matriz de covarianza:

$$S = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T = AA^T$$

Donde A seria la matriz X normalizada (a cada x_i columna se le ha restado la media). La matriz S tiene una dimensión de $dxd(= N^2xN^2)$, lo que hace de extraer sus vectores y valores

propios una tarea computacionalmente prohibitiva para imágenes de un tamaño normal.

Si el número de puntos en el espacio de imágenes es menor que la dimensión del espacio $m < N^2$, habrá como mucho $m - 1$vectores propios significativos. Puede resolverse el problema tomando apropiadamente combinaciones lineales de las imágenes. Si en lugar de calcular AA^T consideramos v_i los vectores propios de A^TA (y A^TA (y α_i sus valores propios) resultará que:

$$A^TAv_i = \alpha_i v_i$$

Pre multiplicando a ambos lados por A, se tiene:

$$AA^TAv_i = \alpha_i Av_i$$

Donde Av_i son los vectores propios de $S = AA^T$.

Partiendo de este análisis, se construye la matriz A^TA de dimensión mxm y se encuentran los m vectores propios (Procedimiento tomado de [13]).

2.3.1.3. Algoritmo

Según [13] un algoritmo típico de reconocimiento de caras utilizando las Eigenfaces seguiría:

1) Tomar un conjunto inicial de imágenes (el conjunto de entrenamiento). Este conjunto debería incluir un número de imágenes para cada persona, con variaciones en la expresión y en la iluminación.
2) Calcular la matriz mxm normalizada y encontrar sus vectores y valores propios. Elegir los m' vectores propios con los valores propios asociados más altos.
3) Combinar el conjunto de entrenamiento con los vectores propios para calcular las m' Eigenfaces.
4) Para cada clase (para cada individuo) elegir al menos una (o promediando más de una) de sus imágenes y calcular el vector de clase (proyectándolo en el espacio de caras) Ω_k.
5) Para cada nueva imagen y calcular su vector $\Omega = u_k(y - \mu)$ y calcular la distancia a cada vector de clase.

2.3.2. greensock.swc

Librería que trabaja conjuntamente con org.rockholla-2.0.1.swc para agregar la funcionalidad de zoom y la posibilidad de mejorar el menú que muestra las ubicaciones al usuario.

2.3.3. org.rockholla-2.0.1.swc

Librería que trabaja conjuntamente con greensock.swc para agregar la funcionalidad de zoom al sistema.

2.4. Análisis de alternativas y selección de la solución

Aunque la finalidad del sistema es que este funcione sobre estaciones o kioscos fijas, como se explicó anteriormente, este también puede ser usado en otros dispositivos que tengan acceso a la red y un navegador web que sea compatible con flash como dispositivos Android y laptops.

En cuanto a las alternativas de hardware para la creación de un kiosco la ventaja de usar una tablet es que ya no se necesitaría el uso de un CPU pues ya todo está incorporado, pero sus desventajas son mayores pues son aparatos delicados que en su mayoría no son hechos para soportar un uso excesivo y estar a la intemperie, además la resolución nativa y

tamaño de pantalla de estos es muy reducida lo cual nos lleva a optar por un PC con pantalla táctil capacitiva.

Una de las desventajas de usar una pantalla táctil es que esta debe ser conectada a un PC donde se encontrara instalado el sistema. Para esto se necesitaría una PC resistente a estar a la intemperie y ya que el sistema no necesitara un nivel de procesamiento alto puesto a que el servidor se encarga de todo esto se puede optar por un PC que haga uso de un procesador básico para así evitar el uso de ventiladores que por el clima pueden retener polvo y luego descomponerse, lo que resultaría en un daño al PC. Debe ser considerada una cámara web de alta resolución, que permita tomar fotos sin ruido que pueda interferir en el procesamiento y comparación de imágenes que realiza el sistema. Este accesorio debe estar enfocado de tal manera que pueda captar el rostro del usuario.

2.5. Alcance del proyecto

Sin una apropiada definición del alcance en el proceso de definición del proyecto no se tendrá la oportunidad de gestionar efectivamente el alcance del mismo. Antes de iniciar el proyecto, es indispensable que el trabajo está entendido y que los responsables, tanto de la ejecución del

proyecto como quienes recibirán los resultados del mismo, posean una visión clara de los resultados esperados, cuando se terminará, como se dará por terminado el trabajo, y cuáles serán los beneficios.

Una vez finalizado el Sistema de Ubicación, este permitirá:

- Que el usuario sea reconocido por el sistema por medio de reconocimiento facial.
- Para representar al campus donde será usado el sistema, este presentará un plano del mismo.
- El usuario podrá consultar como llegar a una determinada ubicación dentro del campus, y el sistema será capaz de darle una guía de una ruta que permita llegar al usuario a su destino.
- Para la segunda vez que el usuario necesite usar el sistema, este será capaz de reconocerlo usando reconocimiento facial y obtener datos de su última consulta para poder continuar con esta.

CAPÍTULO 3

3. Diseño del sistema

En este capítulo se abarca el análisis y diseño de la solución propuesta, así como los procedimientos y métodos empleados para el funcionamiento del mismo, además se realiza un análisis de la tecnología y herramientas usadas a nivel de software para la implementación del sistema, describiendo cada uno de los módulos del Sistema de Ubicación.

3.1. Requerimientos de software

Los requerimientos de software que se cumplirán en el alcance del sistema son de tipo funcional y no funcional.

3.1.1. Requerimientos funcionales

1) El sistema debe permitir al usuario realizar una consulta.
2) El sistema tomará y almacenará una fotografía como único dato de identificación del usuario luego de realizar una consulta.

3) El sistema debe poder reconocer a un usuario que realizó una consulta anteriormente solo con la información tomada de su rostro.
4) El sistema debe permitir al usuario retomar una consulta desde otra ubicación.

3.1.2. Requerimientos no funcionales

Rendimiento del sistema:

- El sistema debe proporcionar, si existiera; una respuesta fiable y precisa al usuario luego de realizar una consulta.
- La respuesta a las acciones deben ser rápidas de tal forma que el usuario sienta un ambiente natural de uso.
- La interfaz debe ser de uso intuitivo y sencillo, que no requiera experiencia del usuario. Tendrá un mapa que acapare la mayor parte de la pantalla con colores representativos para distinguir entre vías y destinos.

Proceso de desarrollo:

- El sistema debe ser desarrollado con la última versión de framework de Flex, usando código en AS3 para la parte lógica.
- El sistema debe ser desarrollado en Adobe Flash Builder 4 como IDE.
- El sistema se ejecutará en un navegador web que soporte Flash.

3.2. Requerimientos de hardware

- El hardware del sistema estará constituido por una pantalla multitouch capacitiva como dispositivo de entrada y salida y de una cámara de alta resolución (requerido si se usa el sistema en kioscos).
- Adicionalmente se utilizará un servidor remoto.

3.3. Definición de escenarios

Requerimiento Funcional 1) El sistema debe permitir al usuario realizar una consulta.

- El sistema debe proporcionarle de forma clara la ubicación actual del usuario.

- El usuario podrá escoger donde desea dirigirse por medio de un listado de ubicaciones.
- El sistema marcará la ruta más corta desde donde se encuentra el usuario hasta su destino en el mapa.

Requerimiento Funcional 2) El sistema tomará y almacenará una fotografía como único dato de identificación del usuario luego de realizar una consulta.

- Cada vez que un usuario realice una consulta en el sistema tomará una fotografía de su rostro.
- La información tomada debe ser enviada a un servidor remoto conectado a la misma red de donde se encuentra conectado el sistema.

Requerimiento Funcional 3) El sistema debe permitir al usuario retomar una consulta desde otra ubicación.

- Si el usuario realizó una consulta, puede retomarla en cualquier otra ubicación donde se encuentre el kiosco.

- El sistema marcará la ruta más corta desde la nueva ubicación del usuario hasta su destino original.
- Si no se reconoce al usuario que intenta retomar una consulta el sistema le pedirá nuevamente que la realice.

Requerimiento Funcional 4) El sistema debe poder reconocer a un usuario que realizó una consulta anteriormente solo con la información tomada de su rostro.

- Cuando el usuario retoma una consulta el sistema deberá tomar una nueva foto del rostro, procesar el listado de fotos de usuarios con consultas pendientes almacenadas en un servidor remoto y compararlas con la fotografía actual.
- Si existe una coincidencia desde el servidor se enviará la información necesaria para que el sistema pueda proporcionar los datos de la consulta.
- Si no existe una coincidencia desde el servidor se enviará la información necesaria para que el sistema informe al usuario y le pida realizar una nueva consulta.

3.4. Arquitectura y diseño del sistema

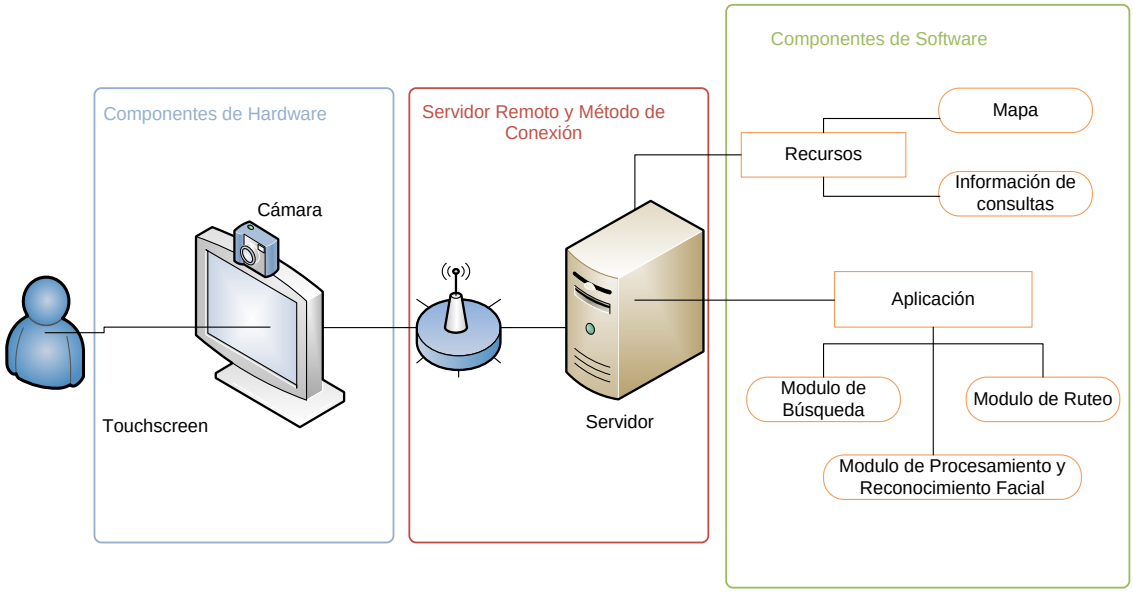


Figura 6. Diseño general del sistema

De manera general, el sistema consta de las siguientes partes:

- Componentes de hardware y dispositivos E/S.
- Servidor Remoto y,
- Aplicación (software).

En el diagrama anterior se explica claramente como es el funcionamiento del sistema en general, y como las partes ya mencionadas trabajan entre sí. A continuación se detalla de manera general de que están conformadas y en que colaboran para que el sistema funcione correctamente.

3.4.1. Componentes de hardware y dispositivos E/S

El sistema de componentes de hardware se encuentra conformado por:

- Una pantalla multitouch capacitiva.
- Un CPU que conste de una tarjeta de red y un sistema operativo compatible con flash.
- Una cámara fotográfica.

Mediante la pantalla multitouch que hace las veces de dispositivo de entrada y salida, pues por medio de este el usuario interactúa con el sistema y muestra los resultados de la consulta que este realice.

Es necesario que la pantalla esté conectada a un CPU que disponga de una tarjeta de red, pues la aplicación y recursos necesarios se encuentran en el servidor remoto y ahí es donde corre el software.

Por medio de la cámara fotográfica el sistema recibe información del rostro del usuario que ha realizado una consulta, estos datos son enviados y guardados directamente en el servidor para su procesamiento posterior si el usuario retoma su consulta desde otra ubicación.

3.4.2. Servidor remoto

En el servidor es donde se encontrara la aplicación en sí, también contiene los datos de consultas de usuario que en este caso son las fotografías de su rostro tomadas luego de una consulta, además realiza el procesamiento pesado como el de comparación de imágenes para que no sean necesarios un procesador y memoria RAM costosa en cada kiosco**.**

3.4.3. Aplicación

Como se especificó anteriormente, la aplicación de software se encontrara en el servidor. Le presentará al usuario en la pantalla un mapa del Campus Gustavo Galindo de la ESPOL y un listado de ubicaciones a la que el usuario se puede dirigir. Una vez realizada la consulta la aplicación mostrara la ruta más corta hacia el punto donde el usuario desea dirigirse**.**

3.5. Integración del hardware

3.5.1. Consideraciones especiales para la implementación del hardware del sistema

Para la realización del kiosco el cual es el que contiene todos los componentes de hardware necesarios se ha considerado los siguientes factores que podrían afectar el correcto funcionamiento de los componentes:

- Clima: Calor, lluvia, humedad, frio, etc…
- Uso constante.
- Correcta ventilación.

- Polvo.

Para resolver el problema del clima todos los componentes del sistema deben estar protegidos por un case hecho de un material resistente. En este caso se puede considerar usar acero inoxidable y en las zonas de la pantalla y el lente de la cámara una protección plástica transparente.

Para evitar la humedad y el polvo cada borde del case debe estar sellada herméticamente con un material de hule o caucho.

Como este será un sistema que se encontrara encendido las 24 horas del día debe tener su respectiva protección eléctrica para evitar alguna falla producida por fallas energéticas en el campus.

El sistema también será utilizado constantemente por eso se ha elegido utilizar una pantalla multitouch capacitiva, la cual como se explicó anteriormente es la que más soporta el usos constante sin afectar su funcionamiento por un largo tiempo.

Con respecto a la ventilación se ha optado por utilizar componentes que no necesiten mucho, ya que el sistema no necesita mucho poder de procesamiento se puede utilizar un procesador de bajo consumo. Se tomó en cuenta esto ya que los ventiladores atraen mucho polvo

lo cual produce que se atasquen. Es posible también utilizar un método de enfriamiento dentro del case.

Figura 7. Ejemplo de un kiosco

3.5.2. Ubicación de los componentes que conforman el hardware del sistema

Para la creación de cada kiosco se debe considerar factores importantes como la altura del mismo, y la ubicación y el ángulo de inclinación de la pantalla y cámara.

La **pantalla** debe estar soportada sobre un soporte de altura promedio de una persona adulta que son quienes usaran el sistema, el soporte debe tener aproximadamente una altura de 1.50 metros

para que de esa manera la pantalla se encuentre ubicada frente a la cara del usuario con una ligera inclinación de 85 grados para que el usuario pueda usar cómodamente el sistema.

La **cámara** debe estar ubicada encima de la pantalla, esto es aproximadamente a 1.60 metros del piso, de tal manera que esta apunte a la cara del usuario.

El resto de componentes deben estar ocultos y ubicados en una caja metálica que les de la protección adecuada.

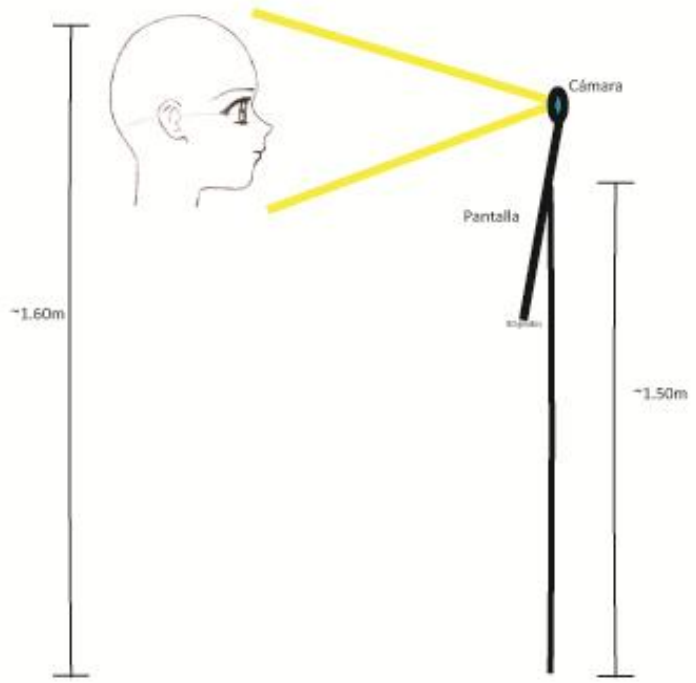


Figura 8. Ubicación de los componentes de hardware para un kiosco

3.6. Diseño de software

En el diseño de software se encuentra detallado los casos de uso, los módulos de este, los diagramas de interacción y los diagramas de clases.

El sistema de ubicación se encuentra conformado por los siguientes módulos principales:

- Módulo de búsqueda.
- Módulo de ruteo.
- Módulo de procesamiento y reconocimiento facial.

A continuación se explica detalladamente cada uno de estos módulos.

3.6.1. Módulo de búsqueda

El módulo de búsqueda es aquel que utilizará un conjunto de ubicaciones almacenadas en el sistema, las cuales se le presentarán al usuario a manera de listado.

El método de búsqueda recibe una elección del usuario, este toma id de la ubicación a la que se desea dirigir y lo envía al módulo de ruteo como un valor entero, que hará de punto de destino para que este

realice los cálculos necesarios para obtener la ruta más corta, el resultado que entrega el módulo y lo muestra marcado en la pantalla.

Figura 9. Esquema E/S del módulo de búsqueda

3.6.2. Módulo de ruteo

El módulo de ruteo es aquel que se encarga de realizar los cálculos necesarios para obtener la ruta más corta entre dos puntos, este módulo fue desarrollado por medio del algoritmo Dijsktra, recibe el punto inicial, el punto final y una matriz de adyacencias basada en un grafo bidireccional sobre la cual por medio del algoritmo se trata de encontrar la ruta más óptima.

Al encontrarse la ruta se devuelve un arreglo de puntos que son las ubicaciones por donde hay que pasar para llegar al destino deseado, marcando la ruta en el mapa.

Figura 10. Esquema E/S del módulo de ruteo

3.6.2.1. Algoritmo de Dijsktra

Dado un grafo a cuyos arcos se han asociado una serie de pesos (en este caso colocamos pesos con valor 1), se define el camino de coste mínimo de un vértice u a otro v, como el camino donde la suma de los pesos de los arcos que lo forman es la más baja entre las de todos los caminos posibles de u a v.

El algoritmo de Dijkstra es un algoritmo eficiente (de complejidad $O(n^2)$ donde n es el número de vértices) que sirve para encontrar el camino de coste mínimo desde un nodo origen a todos los demás nodos del grafo. Fue diseñado por el holandés Edsger Wybe Dijkstra en 1959.

El fundamento sobre el que se asienta este algoritmo es el principio de optimalidad: si el camino más corto entre los

vértices u y v pasa por el vértice w, entonces la parte del camino que va de w a v debe ser el camino más corto entre todos los caminos que van de w a v.

De esta manera, se van construyendo sucesivamente los caminos de coste mínimo desde un vértice inicial hasta cada uno de los vértices del grafo, y se utilizan los caminos conseguidos como parte de los nuevos caminos [14].

3.6.2.2. Procedimiento de algoritmo de Dijsktra

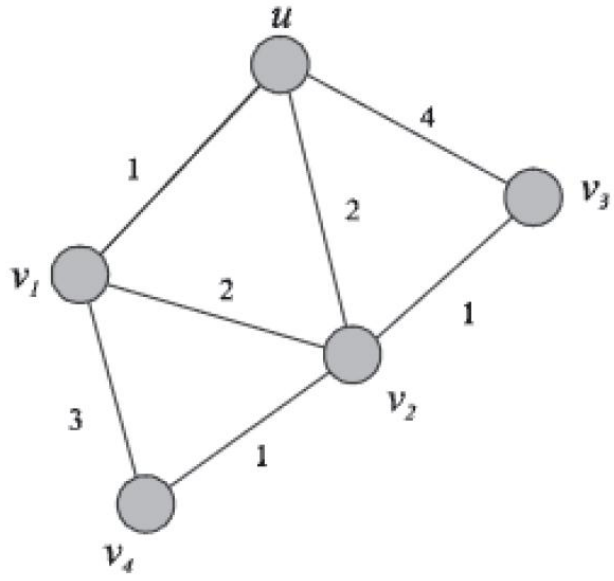


Figura 11. Escenario de una red [15]

Para el algoritmo del diagrama anterior, cada nodo v del grafo $G(V, E)$ tiene una etiqueta asociada $L(v)$, esta etiqueta indica la menor distancia conocida para ir desde un nodo fijado u hasta este nodo. Inicialmente, el valor de $L(v)$ corresponde al peso $w(u, v)$ de la arista que une los nodos u y v, en el caso de que esta arista exista, siendo $L(v) = \infty$; en caso contrario (desconocimiento de las distancias), el algoritmo funciona creando un conjunto de nodos $T[\ ]V$, para los cuales se ha obtenido hasta ese momento el camino más corto desde u hasta ellos. Al final del algoritmo, $L(v)$ contiene el costo del camino más corto para ir desde u hasta v.

En cada iteración el algoritmo añade un nuevo nodo en la lista T. Esto se consigue escogiendo un nodo v' que todavía no pertenezca a T y que tenga una etiqueta $L(v')$ mínima. En otras palabras, se escoge el nodo v' más cercano a u y externo a la lista T. Una vez hecho esto, se actualizan las etiquetas de los nodos sobre los que incide v', de manera que se hace un nuevo cálculo de las distancias de u a estos nodos y se añade este nodo v' a T. El proceso se repite hasta que todos los nodos

del grafo se encuentren en la lista (Procedimiento tomado de [15]).

3.6.3. Módulo de procesamiento y reconocimiento facial

En caso de que se realice la consulta por primera vez, este módulo es el encargado de tomar la fotografía del rostro del usuario y enviarla al servidor.

En caso de que se retome una consulta este módulo se encarga de tomar una nueva fotografía y compararla con las fotografías que se encuentran en el servidor, si encuentra una coincidencia devolverá el nombre de la foto al módulo de ruteo para que este calcule una nueva ruta óptima. Para realizar la comparación de imágenes se encuentra en el servidor un archivo llamado picturehandler.php que es el que se encarga de todo este proceso.

Como se explicó en el punto 2.3.1.1. para realizar esto se utiliza el **método eigenfaces**.

Figura 12. Esquema E/S del Módulo de procesamiento y reconocimiento facial

3.6.4. Casos de uso del sistema

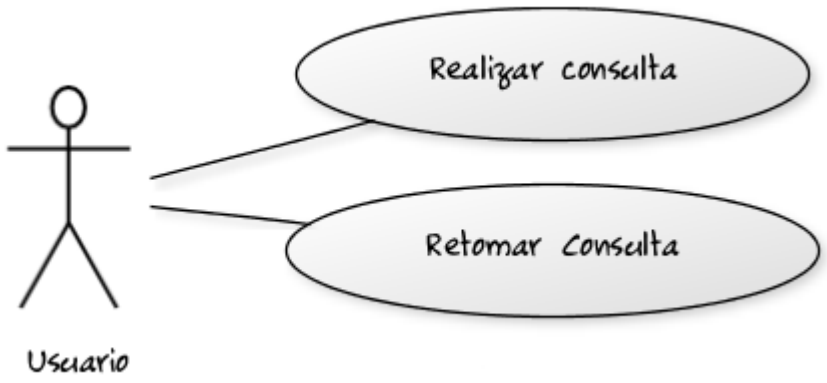


Figura 13. Casos de uso del sistema

3.6.5. Diagrama de interacción

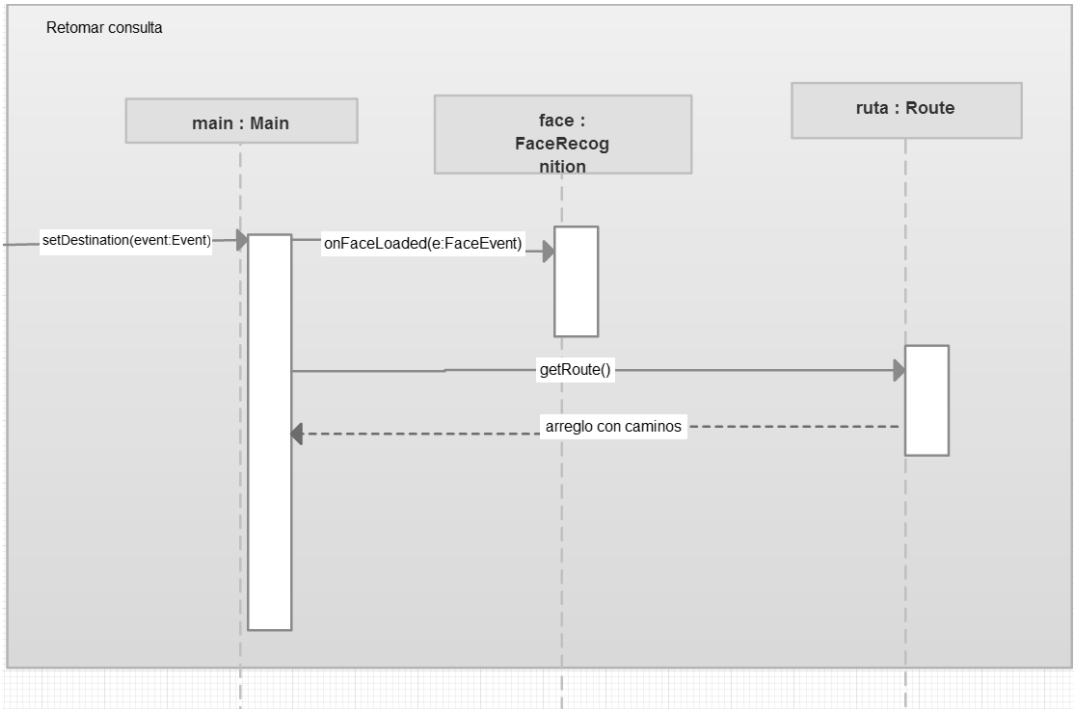


Figura 14. Diagrama de interacción

3.6.6. Diagrama de clases

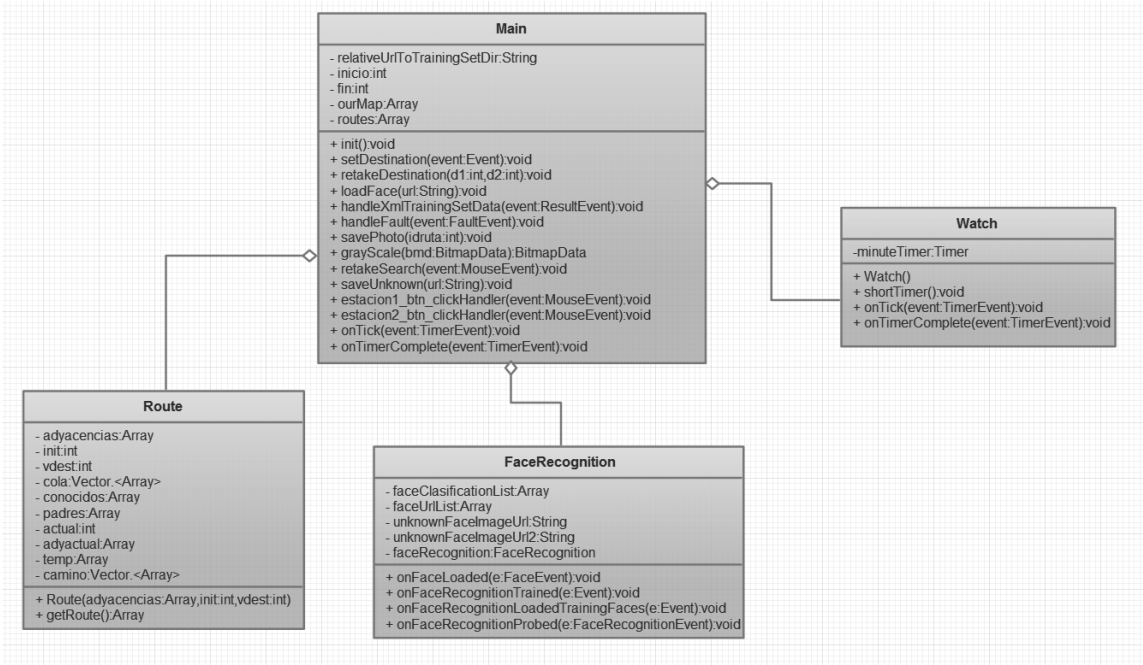


Figura 15. Diagrama de clases

3.7. Diseño de interfaces

El sistema presentará una única pantalla en la que el usuario tendrá disponibles todas las opciones del sistema. A continuación se describen los componentes y las métricas que se siguieron para la creación de la interfaz.

3.7.1. Descripción de componentes

3.7.1.1. Mapa

Se encuentra ubicado de tal manera que ocupe el mayor espacio posible en pantalla, en un inicio se muestra en su totalidad para dar una vista de todos los puntos disponibles del campus. Por medio de la funcionalidad de zoom este se puede mover y enfocar en un punto específico. Muestra un pin en el lugar donde se encuentra ubicado el usuario y las rutas son mostradas en color blanco para hacer contraste con el resto de colores.

Es representado por una imagen de vectores, de tal forma que al hacer zoom en este no se muestre pixelado o con algún defecto.

3.7.1.2. Lista de destinos

Ubicada en el lado derecho de la pantalla concordando con el brazo derecho del usuario, favorable para usuarios diestros sin impedir un fácil uso para usuarios zurdos si el sistema se muestra en una pantalla táctil. Se encuentra optimizada para

uso en pantallas táctiles, muestra un fondo negro y letras legibles en blanco con un icono en el lado izquierdo del nombre de cada ubicación que ayuden a dar una idea general al usuario de que tipo es determinada ubicación (facultad, laboratorio, restaurant, etc.).

3.7.1.3. Botón “Retomar Consulta”

Ubicado en el extremo superior izquierdo de la pantalla de tal manera que no interfiera con componentes del mapa, tiene el mismo formato que la lista de destinos.

Figura 16. Componentes de la interfaz del sistema

3.7.2. Principios usados para el desarrollo de la interfaz.

Existen principios relevantes para el diseño e implementación de la interfaz del usuario, para este sistema se tomaron en cuenta las siguientes:

- **Anticipación:** Con este sistema nos hemos anticipado a las necesidades del usuario, colocando de manera visible todos los componentes y ubicándolos en lugares estratégicos de la pantalla, de esta manera el usuario puede acceder a todas las funciones de manera rápida y sin esfuerzo.
- **Autonomía y Curva de aprendizaje:** Se diseñó y desarrollo la interfaz para que se muestre simple, de tal manera que el usuario no necesite instrucciones o capacitación para poder utilizarlo de manera correcta.
- **Uso de metáforas:** Se muestra la interfaz de tal forma que sea intuitivo, que de la sensación de encontrarse revisando el plano de un mapa.
- **Legibilidad:** Se utilizó colores y tamaños de fuentes grandes de tal manera que exista contraste con otros componentes de la interfaz y facilitar la lectura del mismo.

3.8. Diseño de plan de pruebas

Para realizar las del sistema se involucrara a 10 usuarios, 5 estudiantes de ESPOL familiarizados con el campus Prosperina y 5 que no lo conocen, por medio de esto se realizaran las siguientes pruebas en el sistema:

- Tiempo que toma el usuario para realizar una consulta sin proporcionarles una explicación previa de cómo se utiliza, con esto se determinara que tan intuitivo es el sistema.
- Velocidad de respuesta del sistema al realizar una consulta.
- Velocidad de comparación de imágenes y presentación de respuesta al retomar consulta con diferente número de imágenes en el servidor.

Dependiendo de los datos obtenidos se llenara el siguiente formulario:

	Tiempo que se toma para realizar una consulta (segundos)	Velocidad de respuesta del sistema en proporcionar respuesta a una consulta (segundos)	Velocidad de respuesta al retomar consulta (segundos)
Usuario 1			
Usuario 2			
Usuario 3			
Usuario 4			
Usuario 5			
Usuario 6			
Usuario 7			
Usuario 8			
Usuario 9			
Usuario 10			
Promedio			

Tabla 1. Formulario de pruebas del sistema

CAPÍTULO 4

4. Implementación y pruebas del sistema

En este capítulo se expone los métodos utilizados para la implementación del sistema y los resultados obtenidos del plan de pruebas con un respectivo análisis de los mismos.

4.1. Implementación del sistema

A continuación se explica que métodos fueron usados para el desarrollo de los módulos que conforman el sistema.

4.1.1. Implementación del módulo de búsqueda

El módulo de búsqueda fue desarrollado de tal forma que le envíe los datos necesarios al módulo de ruteo para que este pueda realizar los cálculos correspondientes para hallar la ruta desde un punto inicial hasta un punto final.

Lo que realiza este módulo es obtener el id un ítem seleccionado de una lista al momento de realizar la consulta, transformarlo de tipo de dato String a int y enviarlo al módulo de ruteo como punto de destino, el punto inicial lo da el usuario o es configurado

dependiendo de qué manera se utilizara el sistema, ya sea en móviles o en kioscos.

En la lista los ítems son todos los puntos registrados del campus, ubicaciones donde el usuario puede dirigirse y el id de cada ubicación es el número que identifica su nodo correspondiente en el grafo bidireccional que representa al mapa del campus (Anexo 1) (Anexo 2).

Figura 17. Lista de ubicaciones del campus

Una vez que el usuario ha elegido el punto al que desea dirigirse el sistema capturara su rostro y lo guardara en el servidor en formato

.jpeg y en escala de grises con el id del destino elegido como nombre de archive para ser usado posteriormente por el módulo de reconocimiento facial si es necesario.

4.1.2. Implementación del módulo de ruteo

Al iniciar el proyecto este módulo estaba siendo implementado con ayuda de Google Maps (Anexo 3), pero esta herramienta tenía información incompleta del campus, por este motivo decidimos implementar nuestro propio método de ruteo. Este método está basado en el algoritmo de Dijsktra explicado en el punto 3.6.2.1.

Para que este módulo funcione se necesitó definir una matriz de dos dimensiones de 45x45 donde 45 es el número de destinos disponibles en el mapa (Anexo 2) y el número de nodos en el grafo (Anexo1), en el contenido de la matriz se define 1 como ruta disponible y -1 como infinito o sin ruta.

El algoritmo de Dijsktra implementado en el módulo se encarga de buscar la mejor ruta sumando el peso de las distancias de los posibles caminos de un punto inicial a un punto final, para este caso todos los pesos son 1 lo que hará que el algoritmo entregue como resultado el camino que para llegar al punto pase por el menor

número de puntos en el mapa en forma de arreglo de puntos, luego se recorre este arreglo para mostrar las rutas de un punto a otro ya definidas, las cuales en conjunto conforman la ruta optima de un punto a otro.

4.1.3. Implementación del módulo de reconocimiento facial

El módulo de reconocimiento facial fue desarrollado con la ayuda de la librería de reconocimiento facial de Oscar Wicha en AS3 usando Eingenfaces [12], método explicado en 2.3.1.1.

La función de este módulo es la de reconocer al usuario que ha retomado una consulta hecha anteriormente y presentarle una ruta hacia donde se dirigía desde el nuevo punto en el que se encuentra.

Para esto se realizó una función que tome una nueva fotografía en escala de grises y la compare con las imágenes de consultas realizadas guardadas en el servidor, una vez comparada retorne el nombre de la imagen con mayor coincidencia, esta información es enviada a una función que se encarga de separar el nombre de la extensión de archivo y transformarla a tipo de datos int, este valor representa el lugar al que se dirigía, esta información más la del

punto donde se encuentra se envía al módulo de ruteo el cual muestra la nueva ruta en el mapa.

4.2. Adaptación del hardware

Esta prueba fue realizada en dispositivos móviles como tablets y celulares y laptops, en el caso de dispositivos móviles para que el sistema pueda funcionar de manera eficiente es necesario realizar ajustes en la resolución de la interfaz del sistema y deshabilitar el paso de configuración inicial para kioscos y el módulo de reconocimiento facial puesto que no es una característica útil al utilizarla en estos dispositivos, además debe remplazarse el listado customizado por un combobox que pueda ser fácilmente manejado por navegadores de estos dispositivos y así no ocupar espacio extra en la pantalla.

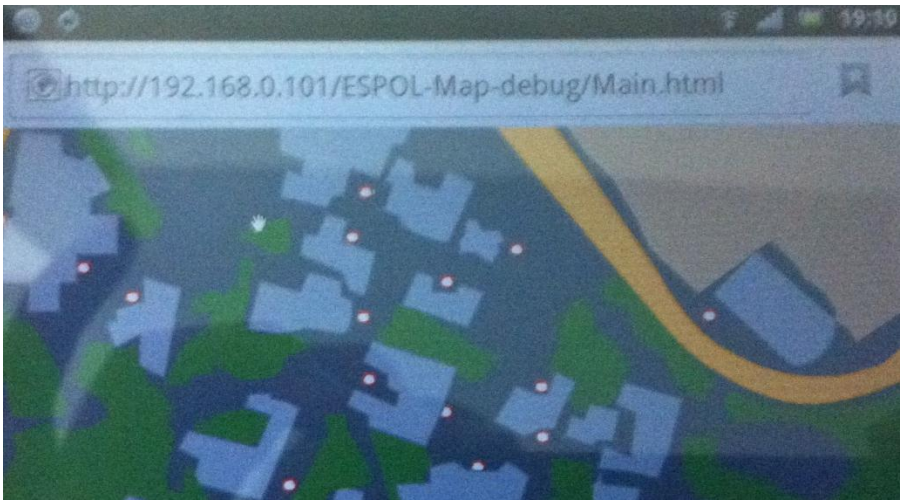


Figura 18. Pantalla del sistema funcionando en un dispositivo móvil

Para el caso de computadores portátiles o kioscos se usa la interfaz original del sistema, la cual aprovecha el espacio extra en pantalla con un menú más detallado y diseñado especialmente para pantallas táctiles (Anexo 4).

4.3. Pruebas y resultados del sistema

Realizadas las pruebas del sistema con 10 usuarios, 5 estudiantes de la ESPOL que ya se encuentran familiarizados con el campus y 5 personas que no se encuentran familiarizados con el mismo usando una computadora con una cámara web emulando a un “kiosco” conectada localmente en red de manera inalámbrica a un servidor se obtuvieron los siguientes resultados:

	Tiempo que se toma para realizar una consulta (segundos)	Velocidad de respuesta del sistema en proporcionar respuesta a una consulta (segundos)	Velocidad de respuesta al retomar consulta (segundos)
Usuario 1	17	1	15
Usuario 2	15	1	16
Usuario 3	15	2	20
Usuario 4	12	1	10
Usuario 5	12	1	11
Usuario 6	18	1	11
Usuario 7	16	1	10
Usuario 8	12	2	14
Usuario 9	20	2	22
Usuario 10	11	1	21
Promedio	**14.8**	**1.3**	**15**

Tabla 2. Resultados de pruebas realizadas al sistema con 10 usuarios

4.4. Análisis de resultados

Analizando a los resultados obtenidos de 10 usuarios utilizando el sistema hemos llegado a las siguientes conclusiones:

- Ningún usuario excepto uno tuvo mayores problemas al utilizar el sistema, una vez que se les explico para que sirve sin dar más detalles ellos lograron realizar su consulta de manera satisfactoria, lo cual indica que se ha logrado una interfaz intuitiva.
- El sistema entrega una respuesta casi inmediata en la realización de una nueva consulta.

- El sistema tarda un poco en entregar una respuesta si u un usuario retoma la una consulta, la causa de esto es el procesamiento y comparación de imágenes, mientras mayor sea el número de imágenes en el servidor mayor será el tiempo de respuesta, otra causa es la velocidad de procesamiento del servidor.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Una vez terminado nuestro proyecto de grado y llevado a cabo los objetivos planteados podemos concluir:

1) Existen muchos algoritmos para la búsqueda de la ruta más corta, en el desarrollo del Módulo de ruteo originalmente se utilizó el método A* para la búsqueda de la mejor ruta, pero luego se decidió utilizar el método de Dijsktra que en comparación es el más practico ya que brinda facilidad en la definición de los destinos en el mapa, lo cual se realiza promedio de una matriz de adyacencias.
2) Al usar Flash Builder como IDE para desarrollar el sistema tuvimos acceso a muchas funcionalidades con otros productos Adobe lo que nos dio una demostración de la integración que existe entre todos ellos, como es el caso de Adobe Illustrator que nos permitió realizar el plano completo de la ESPOL y exportarlo de manera fácil a código .fxg, código que fue usado en el sistema con la finalidad de no tener problemas de pixelación al realizar zoom en el plano.
3) Flex es una excelente herramienta para desarrollar interfaces intuitivas.

4) Obtuvimos buenos comentarios en el periodo de pruebas por parte de los usuarios señalando la importancia de que se ponga en funcionamiento este sistema y lo mucho que se lo ha necesitado, lo en parte cubre las expectativas planteadas en nuestra justificación del problema.

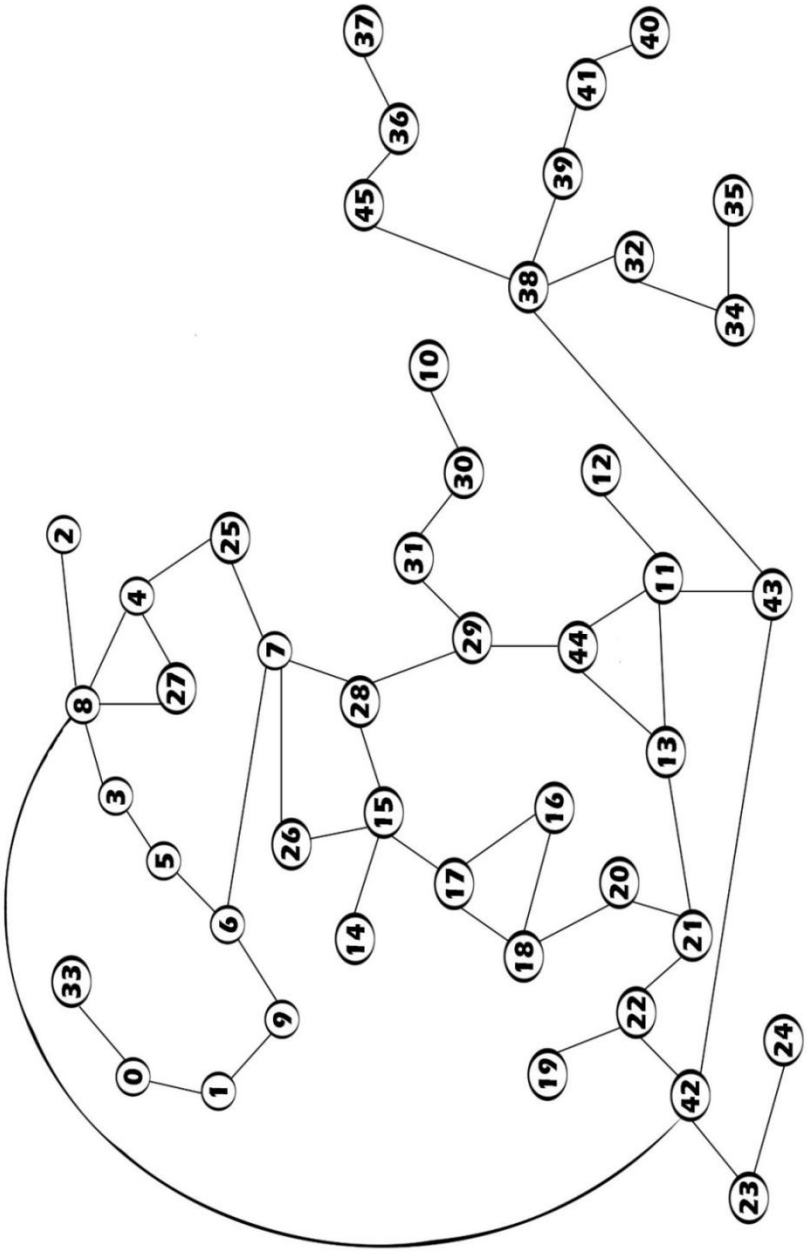
Recomendaciones

A continuación se detallan algunas recomendaciones:

1) Antes de iniciar con el desarrollo de software es necesario analizar y comparar las ventajas y desventajas de frameworks existentes según lo que necesitemos.
2) Recomendamos que el sistema sea utilizado en estaciones ubicadas estratégicamente alrededor del campus Prosperina de la ESPOL pues de esta manera los estudiantes y visitantes podrán acceder a él sin necesidad de tener un móvil con acceso a internet o una laptop.

ANEXOS

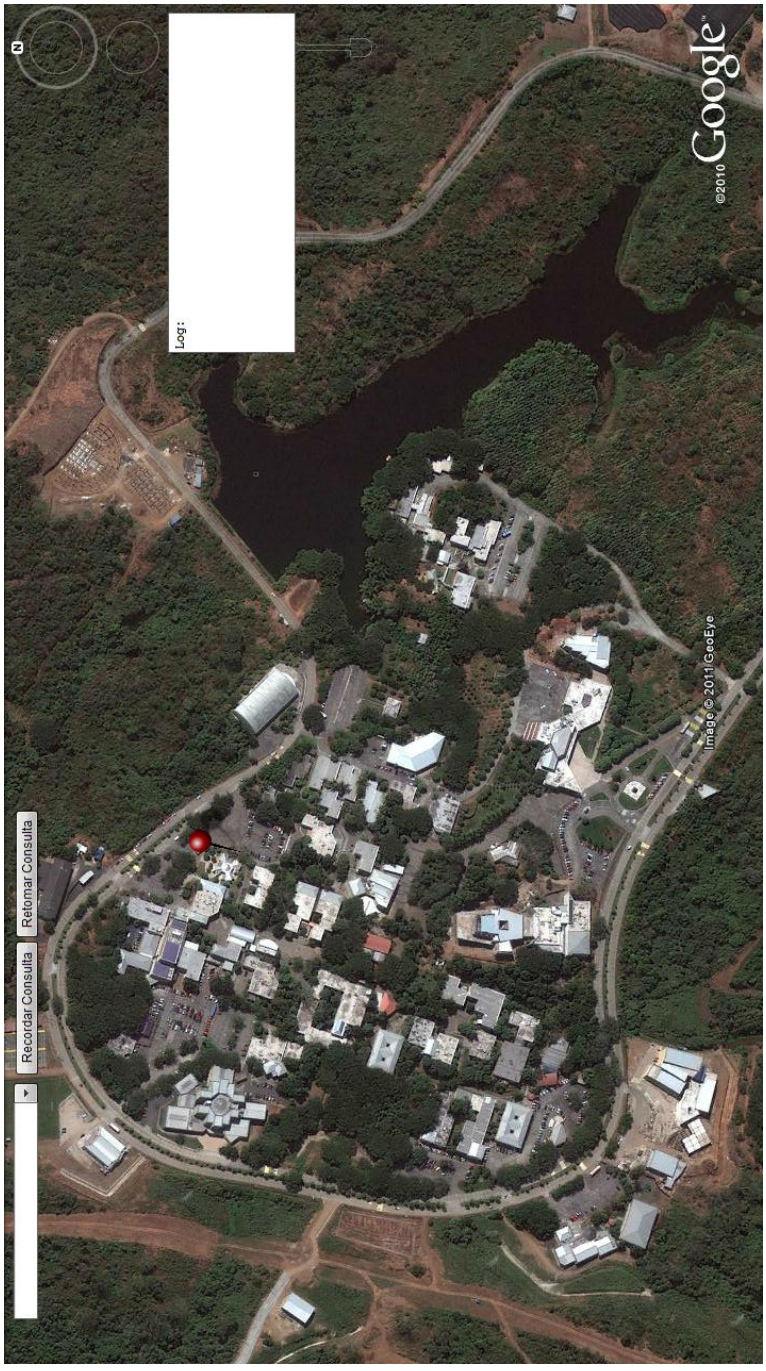
Anexo 1. Grafo bidireccional que representa a los puntos y caminos del campus Prosperina de la ESPOL



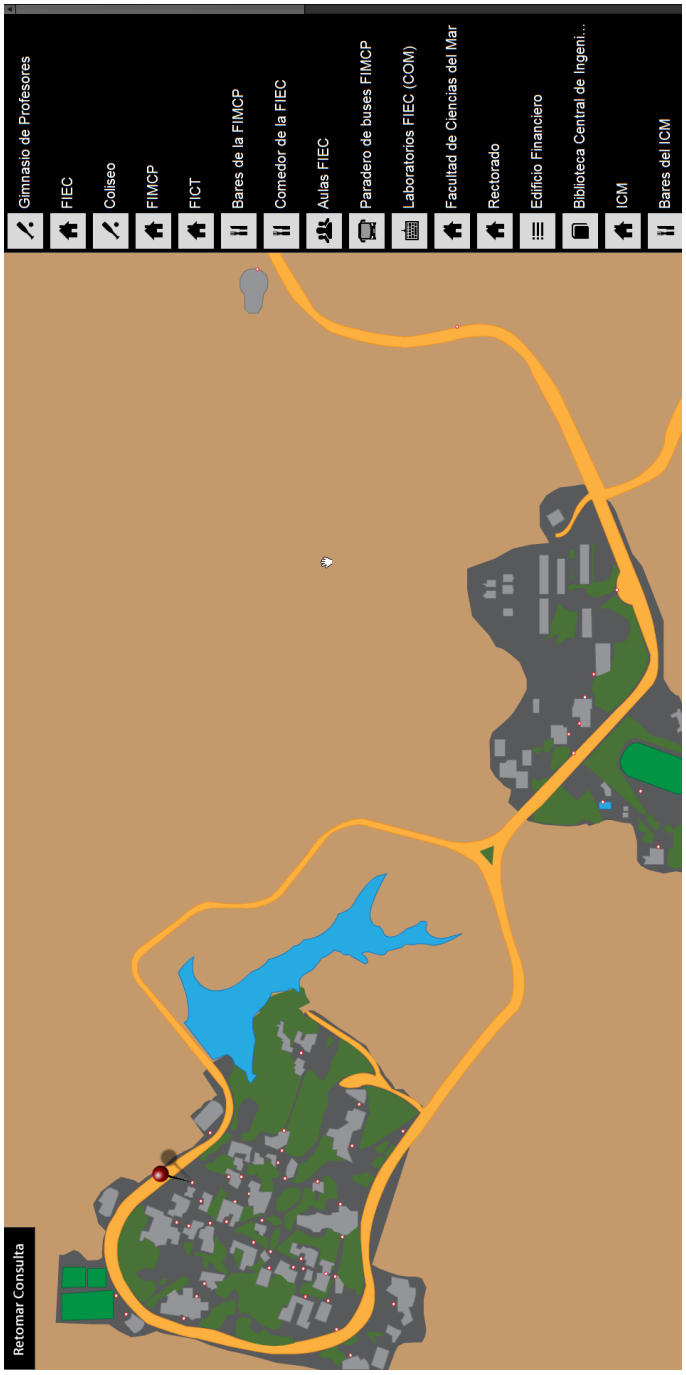
Anexo 2. Tabla de ubicaciones con su respectivo numero de nodo en el grafo del Anexo 1

0	Gimnasio Profesores
1	Fiec (Aulas, Decanato, Subdecanato, Oficinas Profesore
2	Coliseo
3	Facultad de Mecánica FIMCP
4	Faculta de Ciencias de la Tierra FICT
5	Bares de Mecánica
6	Comedor FIEC
7	Aulas FIEC
8	Paradero de Buses FiMCP
9	Laboratorios FIEC (Aulas COM)
10	Facultad de Ciencias del Mar FIMCM
11	Rectorado (Crece, Relex, Vicerrectorado, STA)
12	Financiero
13	Biblioteca Central Ingeniería
14	Instituto de Ciencias Matemáticas ICM
15	Bares ICM
16	Aulas Básica
17	Instituto de Ciencias Física ICF
18	Instituto de Ciencias Químicas ICQ
19	Laboratorios ICQ e ICF
20	Laboratorios y Aulas FEN
21	Comedor FEN
22	Aulas FEN
23	Facultad de Economía y Negocios FEN
24	Aulas Básico 2
25	Aulas FICT
26	Laboratorios FIEC
27	Aulas FIMCP
28	Fepol
29	CSI
30	Comedor Principal
31	Máster Paz
32	Piscina
33	Canchas FIEC
34	Gimnasio Alumnos
35	Canchas Tecnologías
36	CESE (CELEX)
37	Aulas Tecnologías
38	Instituto de Tecnologías
39	COPOL
40	CTI (Parcon)
41	Aulas Pre politécnico
42	Paradero FEN
43	Paradero Rectorado
44	Bopan
45	Facultad de Tecnologías

Anexo 3. Primera versión del sistema



Anexo 4. Versión final del sistema



MANUAL DEL SISTEMA

Realizar consulta

Para realizar una consulta debe seleccionar uno de los destinos ubicados en el extremo derecho de la pantalla, automáticamente el sistema le mostrara el resultado.

Retomar consulta

Si se realizó una consulta previa y desea retomarla desde un nuevo punto, seleccionar el botón “Retomar Consulta” ubicado en la esquina superior derecha, luego de unos segundos el sistema le mostrar el resultado, si este existe.

BIBLIOGRAFÍA

[1] Ingmarrr, Indoor Navigation System - Positioning Demo in Hallway, http://www.youtube.com/watch?v=YhQR9t4n36I/, fecha de consulta julio 2011.

[2] Broadbandsystems, Multi-Sensor Pedestrian Indoor/Outdoor Navigation, http://www.youtube.com/watch?v=ySlgBUP0Vqo&feature=autoplay&list=PL6A658B76F559445A&index=2&playnext=2/, fecha de consulta agosto 2011.

[3] Alegsa, Definición De Tablet PC, http://www.alegsa.com.ar/Dic/tablet%20pc.php, fecha de consulta diciembre 2011].

[4] Samsung, GALAXY Tab 10.1- Samsung Mobile, http://www.samsung.com/global/microsite/galaxytab/10.1/images.html/, fecha de consulta enero 2012.

[5] O. B, Pantallas Táctiles: Capacitivas vs Resistivas, http://www.xatakamovil.com/desarrollo/pantallas-tactiles-capacitivas-vs-resistivas/, fecha de consulta enero 2012.

[6] PlanarSystems, Touch Screens, http://www.planartouch.com/101/select/, fecha de consulta enero 2012.

[7] A. Que Es Java Y Para Que Sirve?, http://firedoz.blogspot.com/2011/04/que-es-java-y-para-que-sirve.html/, fecha de consulta enero 2012.

[8] Oracle, ¿Qué es la tecnología Java y por qué lo necesito?, http://www.java.com/es/download/faq/whatis_java.xml, fecha de consulta mayo 2012.

[9] W3Schools, HTML5 Introduction, http://www.w3schools.com/html5/html5_intro.asp, fecha de consulta mayo 2012.

[10] O. Cortes, Que Es Flex?, http://www.holaflex.com/?p=34/, fecha de consulta enero 2012.

[11] TechTarget, What is Flex?, http://whatis.techtarget.com/, fecha de consulta noviembre 2011.

[12] O. Wicha, Face-recognition-library-as3 - Oskar Wicha's ActionScript 3 Face Recognition Library Using Eigenfaces - Google Project Hosting, http://code.google.com/p/face-recognition-library-as3/, fecha de consulta diciembre 2011.

[13] I. Armengot and M. J. , Análisis Comparativo De Métodos Basados En Subespacios Aplicados Al Reconocimiento De Caras, http://www.uv.es/marjoari/pdf/definitivo.pdf, fecha de consulta diciembre 2011.

[14] G. Sánchez Torrubia and V. Lozano Terrazas, Algoritmo de Dijkstra. Un Tutorial Interactivo, http://bioinfo.uib.es/~joemiro/aenui/procJenui/ProcWeb/actas2001/saalg223.pdf, fecha de consulta mayo 2012.

[15] L. Pedraza, D. López and O. Salcedo, Enrutamiento basado en el algoritmo de Dijkstra para una red de radio cognitiva, http://tecnura.udistrital.edu.co/downloads/revista30/conciencias/Articulo9-30.pdf, fecha de consulta mayo 2012.