ESCUELA POLITECNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERIA EN ELECTRÓNICA Y TELECOMUNICACIONES

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA

“IMPLEMENTACION DE CONTROL DE ACCESO Y MONITORIZACION PARA PERSONA CON DISCAPACIDAD MEDIANTE UN DISPOSITIVO MOVIL”

JUAN FERNANDO VILLALBA GALLARDO

Sangolquí – Ecuador

2013

CERTIFICACIÓN

Certificamos que el Sr. JUAN FERNANDO VILLALBA GALLARDO, ha elaborado el proyecto de grado titulado "**IMPLEMENTACION DE CONTROL DE ACCESO Y MONITORIZACION PARA PERSONA CON DISCAPACIDAD MEDIANTE UN DISPOSITIVO MOVIL**", para la obtención del título de Ingeniería Electrónica, Especialidad Telecomunicaciones, bajo nuestra dirección.

Atentamente,

Ing. Fabián Sáenz	Ing. Carlos Romero
DIRECTOR	**CODIRECTOR**

RESUMEN

El presente proyecto tiene como objetivo la implementación de un control de acceso y monitorización para una persona con discapacidad mediante un dispositivo móvil a través de una intranet para que pueda ser controlado remotamente dentro de una red. Se provee entregar una solución innovadora para ayuda de las personas en cuanto al control de luces y apertura de compuertas con la verificación de una cámara IP que permita al usuario mantener el control de acceso a sus domicilios.

De esta manera utilizando la red de sus hogares puede realizarlo con la mayor comodidad, seguridad, y sin mayores dificultades pues no hace necesario de esfuerzo físico para al encender luminosidad o puertas a su conocidos, pues las personas con discapacidad motriz tendrá la facilidad de realizarlo remotamente, al igual que una persona con la habilidad de realizarlo presencialmente puede utilizarlo de igual manera.

El sistema de control y monitorización permitirá mantener gracias a la conexión de red todo el control de acceso mediante el uso de un servidor web embebido que utiliza protocolos de conexión a páginas web e internet, así como también el almacenaje de su propia web para control. Para mantener el control de luces y compuertas se utilizará un controlador que activará las respectivas salidas para procesar órdenes de ejecución del usuario, y posteriormente activar la etapa de potencia que controlará los actuadores finales para el resultado esperado.

AGRADECIMIENTO

Los resultados de este proyecto de investigación son gracias al producto de la investigación, perseverancia y fortaleza del aprendizaje diario con la constancia de aprender lo desconocido en producto de crear una idea que facilite la vida de las personas. Conocimientos e ideales que han sido inculcados a lo largo de mi vida universitaria por parte de mis profesores para alcanzar mis deseos profesionales. A mi director y codirector quienes han demostrado que cualquier idea es posible desarrollar con una fuerte dedicación y curiosidad constante por aprender; gracias a esto ha sido posible implementar este proyecto.

Juan Fernando Villalba Gallardo

DEDICATORIA

Dedico el presente proyecto de tesis a Dios quien en todos los momentos ha iluminado mi pensamiento e inspiración para continuar con la meta que me he propuesto. A mis padres quienes siempre ha estado al pendiente de mi bienestar, con su sacrificio diario y apoyo me han enseñado a salir adelante en todo momento, depositando su confianza y fe en mi para ser una persona ética y de valores. A mi familia que en todo momento han demostrado su apoyo e incentivando mis deseos a lo largo de mi vida. A mi novia quien con su determinación ha sabido impulsarme con paciencia y constancia a finalizar este proyecto, denotando mis capacidades para el crecimiento diario. A mis profesores quienes han formado en mi una persona de lucha constante ante la presencia de conocimientos tecnológicos que día a día se descubren en virtud de aprendizaje, y crecimiento profesional. Este proyecto de investigación es para todos ustedes.

Juan Fernando Villalba Gallardo.

PRÓLOGO

Hoy en día muchas de las personas especiales con un tipo de discapacidad motriz requieren de un cuidado especial al momento de garantizar el ingreso a sus conocidos a sus domicilios ya que no tienen la facilidad de movilizarse hasta sus puertas para permitir la apertura del mismo, mucho menos con el uso de llaves. Es por eso que se realiza el presente trabajo de investigación con el fin de fomentar el uso de un dispositivo móvil para que a través del uso de la red domiciliaria mediante el uso de una conexión Wifi dentro del área local para permitir a la persona monitorear su domicilio y garantizar la apertura de acceso a personas de interés y conocidas.

El presente trabajo está organizado en 5 capítulos, en los cuales se tratan los aspectos referentes al proyecto.

En el Capítulo 1, se hace una introducción al problema que se pretende solucionar con el presente trabajo, además se muestra un marco teórico referente a sistemas web embebidos, lenguaje HTML para el diseño de páginas web, uso de equipos móviles.

En el Capítulo 2, se detalla los diferentes sistemas operativos para móviles, así como también los conceptos para el diseño de aplicativos nativos dentro de un equipo móvil que para este caso será el IPhone.

En el capítulo 3, se definen los equipos a utilizar, como es el servidor embebido, el cual constan los pasos para la configuración del mismo. El diseño del controlador con sus puertos, así como también la etapa de potencia para el control de luces y accesos. Además del acoplamiento de los circuitos para la presentación total del sistema. Se define la interfaz de programación para el dispositivo móvil IPhone.

En el capítulo 4, se realizan pruebas de conexión de ambos aplicativos, tanto el aplicativo Web, así como también el aplicativo interno instalado en el móvil IPhone para diferenciar el tipo de uso que implica dentro del sistema.

Finalmente en el Capítulo 5 se encuentran las conclusiones y recomendaciones que de han deducido a lo largo del desarrollo del presente trabajo de investigación.

Adicional en la sección anexos se muestra el manual de usuario del sistema implementado.

INDICE

CAPITULO I INTRODUCCION

1.1 ANTECEDENTES 1

1.2 JUSTIFICACION 2

1.3 OBJETIVOS 3

1.3.1 Objetivo General. 3

1.3.2 Objetivos Específicos. 3

1.4 SERVIDOR WEB 4

1.4.1 Definición. 4

1.4.2 Componentes de un servidor. 6

1.4.3 Servidores Embebidos. 7

1.4.3.1 Estructura. 8

1.4.3.2 Funcionamiento. 10

1.5 PAGINAS WEB 12

1.5.1 Definición. 12

1.5.2 Estructura de una página Web. 14

1.5.2.1 Formatos de texto. 16

1.5.2.2 Links a otras páginas. 18

1.5.2.3 Tablas. 19

1.5.2.4 Ubicación de Imágenes. 21

1.5.2.5 Formularios. 23

1.5.3 Medios de Transmisión. 26

1.6 CAMARAS IP 28

1.6.1 Definición. 28

1.6.1.1 Cámara IP Fijas. 28

1.6.1.2 Cámaras IP Móviles. 29

1.6.1.3 Cámaras IP Infrarrojas. 29

1.6.2 Características. 30

1.6.3 Funcionamiento. 32

1.6.3.1 Aplicaciones. 33

1.7 DISPOSITIVO MÓVIL SMARTPHONE 34

1.7.1 Definición. 34

1.7.2 Características. 36

1.7.3 Protocolos de conexión a la WEB. 37

CAPITULO 2 INTRODUCCION AL SISTEMA OPERATIVO DE MOVILES

2.1 CONCEPTOS GENERALES..... 39
2.1.1 Sistema Operativo Android. 40
2.1.1.1 Arquitectura del Sistema Operativo Android.....44
2.1.2 Windows Mobile. 48
2.1.3 IOS de IPHONE. 49
2.1.3.1 Frameworks. 50
2.1.3.2 Creación de aplicaciones.....51
2.2 KIT DE DESARROLLO DE SOFTWARE 53
2.2.1 Herramientas para iPhone. 53
2.2.1.1 Plataforma XCODE.....54
2.2.1.2 Plataforma INTERFACE BUILDER. 57
2.2.1.3 IPhone simulador. 59
2.3 PROGRAMACIÓN EN OBJECT C. 60
2.3.1 Programación a Objetos. 61
2.3.2 Declaración de variables. 62
2.3.3 Interfaces. 63
2.3.4 Implementación. 66
2.3.5 Objetos de XCODE. 67
2.3.5.1 UIWebView. 68
2.3.5.2 Alertviews. 70
2.3.5.3 Seguimientos. 71
2.3.6 Controladores. 72
2.3.7 Propiedades. 75
2.3.8 Instancia de métodos. 80
2.4 COMPATIBILIDAD..... 84

CAPITULO 3 DESARROLLO DEL SISTEMA DE CONTROL DE ACCESOS Y MONITOREO.

3.1 HARDWARE 85

3.1.1 Equipo Servidor de aplicación Web. 87

3.1.1.1 Criterio de selección. 88

3.1.1.2 Descripción. 90

3.1.1.3 Esquema de conexión del Servidor. 91

3.1.2 Cámaras IP. 94

3.1.2.1 Dispositivo de selección. 94

3.1.2.2 Esquema de conexión. 97

3.1.2.3 Conexiones disponibles. 98

3.1.3 Dispositivo móvil utilizado. 99

3.1.3.1 Características técnicas. 100

3.1.3.2 Criterios de selección. 102

3.1.4 Circuito de Control. 102

3.1.4.1 Microcontrolador utilizado. 103

3.1.5 Etapa de Potencia. 106

3.1.5.1 Optoacoplador MOC3031. 106

3.1.5.2 Triac BT136. 107

3.1.6 Diseño del circuito de control de Acceso. 109

3.2 DISEÑO DEL SOFTWARE. 112

3.2.1 Diseño y Programación de la aplicación Web. 113

3.2.1.1 Ingreso Web. 114

3.2.1.2 Control de Accesos y Monitorización. 115

3.2.2 Especificaciones y requerimientos. 120

3.2.2.1 Diagrama de Ejecución. 120

3.2.2.2 Diseño del servidor. 122

3.2.3 Archivo SPD. 124

3.2.4 Archivo SPB. 125

3.2.5 Compilación y carga del Archivo SPB. 126

3.2.6 Diseño de aplicación gráfica en INTERFACE BUILDER XCODE. 129

3.3 DISEÑO Y PROGRAMACIÓN EN LA PLATAFORMA XCODE 135

3.3.1 Inicio del archivo de implementación. 137

3.3.1.1 Encendido de iluminación. 138

3.3.1.2 Acceso a cámaras y advertencia. 139

3.3.1.3 Apertura de compuertas. 140

3.4 COSTO DEL PROYECTO. 142

CAPITULO 4 PRUEBAS Y RESULTADOS EXPERIMENTALES

4.1 PRUEBAS Y ANALISIS DE COMUNICACIÓN 144
4.1.1 Pruebas de ingreso al sistema. 144
4.1.2 Verificación de entrada..... 146
4.1.3 Ingreso al sistema. 148
4.2 PRUEBAS Y RESULTADOS DEL SISTEMA..... 150
4.2.1 Pruebas de visualización de cámara..... 150
4.2.2 Pruebas de activación de iluminación. 151
4.2.3 Pruebas de apertura de compuerta..... 152

CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES..... 157
5.2 RECOMENDACIONES 159

REFERENCIAS BIBLIOGRÁFICAS 165

ÍNDICE DE FIGURAS

Figura. 1.1. Servidor Web.....4
Figura. 1.2. Esquema Aplicativo del Servidor Embebido.....8
Figura. 1.3. Composición de un Sistema Embebido.....11
Figura. 1.4. Estilo de Página Web. 13
Figura. 1.5. Ejemplo básico de una Página Web.....15
Figura. 1.6. Tabla en página Web. 19
Figura. 1.7. Imagen en Página Web. 22
Figura. 1.8. Cámara IP fija. 28
Figura. 1.9. Cámara IP móvil. 29
Figura. 1.10. Camara IP infrarroja. 29
Figura. 1.11. Smartphones 35
Figura. 1.12. Acceso Web en un móvil. 38

Figura. 2.1. Niveles de Programación de IPhone. 49
Figura. 2.2. Plataforma XCODE. 55
Figura. 2.3. Ventana de Proyecto de XCODE. 57
Figura. 2.4. Interfaz de Interface Builder. 58
Figura. 2.5. Simulador de IPHONE.....59
Figura. 2.6. Estructura de una Clase en Object C. 62
Figura. 2.7. Interfaz de la clase. 64
Figura. 2.8. Interfaz con Protocolos Formales. 65
Figura. 2.9. Implementación de la interfaz.....67
Figura. 2.10. Elemento UIWebView.....68
Figura. 2.11. Elemento UIAlertView.....70
Figura. 2.12. StoryBoard.....71
Figura. 2.13. Archivo de Interfaz .H 73
Figura. 2.14. Pantalla de un UIWebView. (Vista Web) 74
Figura. 2.15. Archivo de Implementación con Atributos. 75
Figura. 2.16. Archivo de Implementación de la clase. 77

Figura. 2.17. Archivo de implementación .M 78
Figura. 2.18. Instancia de variables de implementación.....79
Figura. 2.19. Implementación Gráfica en XCODE.....83
Figura. 2.20. Compatibilidad del software de IOS. 84

Figura. 3.1. Diagrama de bloques del Servidor. 86
Figura. 3.2. Diagrama de bloques de Control de Potencia.....86
Figura. 3.3. Servidor Embebido SITEPLAYER.....88
Figura. 3.4. Puerto MAC-JACK con filtros. 91
Figura. 3.5. Estructura del Mac-Jack y RJ-45.....92
Figura. 3.6. Puerto y Conector RJ45 92
Figura. 3.7. Diagrama de Bloques de conexión a Router o Módem.....93
Figura. 3.8. Cámara IP D-Link. 95
Figura. 3.9. Conexión de cámara IP de tipo cableada.....95
Figura. 3.10. Conexión de cámara IP de forma inalámbrica. 96
Figura. 3.11. Parte posterior de una Cámara IP.....97
Figura. 3.12. Conexión de Cámara mediante RJ45 97
Figura. 3.13. Conexión de la cámara de manera inalámbrica. 98
Figura. 3.14. Dispositivo IPhone de Apple.....99
Figura. 3.15. PIC18F2550 103
Figura. 3.16. Características del PIC18F2550.....104
Figura. 3.17. Descripción de los pines del PIC18F2550.....104
Figura. 3.18. Diagrama de conexión del PIC18f2550.....105
Figura. 3.19. MOC3031 106
Figura. 3.20. Triac BT136. 107
Figura. 3.21. Circuito de Potencia. 109
Figura. 3.22. Diagrama de Conexión del SitePlayer.....110
Figura. 3.23. Conexión entre SitePlayer y Controlador. 110
Figura. 3.24. Diagrama del Controlador y Circuito de Potencia. 111
Figura. 3.25. Pantalla de Presentación.....114
Figura. 3.26. Contraseña de ingreso al sistema. 115
Figura. 3.27. Aplicación WEB 116

Figura. 3.28. Encedido de luz. 117
Figura. 3.29. Apagado de Luz. 117
Figura. 3.30. Acceso a la Cámara IP. 117
Figura. 3.31. Monitoreo de la Imagen de la Cámara. 118
Figura. 3.32. Compuerta Nro 1. 119
Figura. 3.33. Compuerta Nro 2. 119
Figura. 3.34. Diagrama de ejecución del sistema.....121
Figura. 3.35. Archivo de configuración SPD_SITEPLAYER.....122
Figura. 3.36. Pantalla de compilación del SiteLinker. 126
Figura. 3.37. Ingreso de la IP correspondiente a SITEPLAYER, ...127
Figura. 3.38. Estado de la descarga de la configuración al SITEPLAYER.....128
Figura. 3.39. Descarga completada al Servidor SITEPLAYER. 128
Figura. 3.40. Copia de configuración en dispositivo SITEPLAYER. 129
Figura. 3.41. Diagrama de flujo del proceso del aplicativo. 130
Figura. 3.42. Portada Principal de la Aplicación móvil.....131
Figura. 3.43. Evento de acceso a Indicaciones. 132
Figura. 3.44. Indicaciones de uso del aplicativo móvil.....132
Figura. 3.45. Indicaciones para el ingreso al aplicativo. 133
Figura. 3.46. Entrada a 2 áreas de control. 134
Figura. 3.47. Componentes de cada área de control. 134
Figura. 3.48. Archivo de cabecera del Área #1. 135
Figura. 3.49. Archivo de cabecera del Área #2. 135
Figura. 3.50. Acción de los objetos del Aplicativo. 136
Figura. 3.51. Inicialización de variables y presentación. 137
Figura. 3.52. Encendido y apagado de iluminación. 138
Figura. 3.53. Acceso a Cámara IP y Advertencia. 139
Figura. 3.54. Código para apertura de compuertas. 140
Figura. 3.55. StoryBoard del aplicativo. 141

Figura. 4.1. Comparativa de ingresos al Aplicativo. 145
Figura. 4.2. Tráfico de comunicación a SITEPLAYER. 145
Figura. 4.3. Contraseña de ingreso al aplicativo. 146

Figura. 4.4. Tráfico de datos en Contraseña. 147
Figura. 4.5. Indicaciones previas al uso del aplicativo móvil. 147
Figura. 4.6. Áreas disponibles en el aplicativo móvil. 148
Figura. 4.7. Áreas disponibles en el aplicativo WEB. 149
Figura. 4.8. Tráfico de ingreso al sistema.....149
Figura. 4.9. Monitorización de Cámaras. 150
Figura. 4.10. Activación de luminosidad desde IPhone. 151
Figura. 4.11. Activación de luminosidad a través del Aplicativo Web.....152
Figura. 4.12. Apertura de compuertas a través de IPhone. 152
Figura. 4.13. Apertura de compuerta desde aplicativo Web.....153

ÍNDICE DE TABLAS

Tabla 1.1. Estructura de una Página Web.....14
Tabla 1.2. Tipos de Cabecera HTML.....16
Tabla 1.3 Tipos de Texto en HTML. 16
Tabla 1.4. Hipervínculos y Enlaces. 18
Tabla 1.5. Código fuente de la Tabla en HTML. 21
Tabla 1.6. Código fuente de Imagen Insertada. 22
Tabla 1.7. Formulario Web 24
Tabla 1.8. Formulario Web Compuesto.....26
Tabla 2.1. Middleware como etapa de integración 41
Tabla 2.2. Middleware en etapa de Aplicación 42
Tabla 2.3. Características del S.O. Android. 47
Tabla 2.4. Archivos de XCODE. 82
Tabla 3.1. Descripción de los pines de SITEPLAYER.....91
Tabla 3.2. Normas para Cable Cruzado. 93
Tabla 3.3. Descripción de los pines del PIC18F2550. 105
Tabla 3.4. Identificación de Componente TRIAC. 108
Tabla 3.5. Tabla de datos de SitePlayer.....125

GLOSARIO

FIRMWARE: Bloque de herramientas de programa para programación y configuración de software mediante instrucciones lógicas para interpretación de dispositivos electrónicos. Es de derecho de fabricante.

WIFI: Tecnología de telecomunicaciones basada en el Estándar IEEE 802.11 para comunicaciones inalámbricas entre dispositivos electrónicos.

LENGUAJE ENSAMBLADOR: Lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores, y otros circuitos integrados programables, implementa una representación simbólica de los códigos de máquina binarios.

MICROCONTROLADOR: Circuito digital que se encuentra conformado por una unidad central de procesamiento CPU, memoria flash y puertos de entrada y salida para periféricos.

COMPILADOR: Es el programa informático que traduce un lenguaje de programación escrito a otro generando un equivalente para la interpretación de una máquina.

LAN: Red de área local utilizada para comunicaciones, usualmente dentro de locales y domicilios.

PIC: Familia de microcontroladores tipo RISC (Conjunto Reducido de instrucciones) fabricados por Microchip Technology.

PERIFERICOS: Son aparatos o dispositivos auxiliares e independientes conectados a la unidad central de procesamiento (CPU), permitan realizar operaciones de entrada/salida (E/S) complementarias al proceso de datos que realiza la CPU.

RJ-45: Interfaz física comúnmente usada para conectar redes de cableado estructurado, de categorías 4, 5, 5E, 6 y 6A. Estructurada por ocho pines o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

ACTUADORES: Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final

MAC: Es el conjunto de mecanismos y protocolos por los que varios dispositivos en una red, como ordenadores, teléfonos móviles, se ponen de acuerdo para compartir un medio de transmisión común (por lo general, un cable eléctrico u óptico, o en comunicaciones inalámbricas el rango de frecuencias asignado a su sistema).

DEPURADOR: Programa usado para probar y depurar (eliminar los errores) de otros programas (el programa "objetivo"). Mediante una técnica que permite gran potencia en su capacidad de detenerse cuando son encontradas condiciones específicas pero típicamente algo más lento que ejecutando el código directamente en el procesador.

GUIs (*Interfaz de uso gráfico*): Programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz, proporciona un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

HTML: Acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), y es el lenguaje de programa que todos navegadores usan para presentar información en internet mediante la abreviatura World Wide Web (WWW).

MEMORIA FLASH: Es una tecnología de almacenamiento, derivada de la memoria EEPROM, que permite lectura y escritura de múltiples posiciones de memoria en la misma operación, mediante impulsos eléctricos, permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM. Es posible borrarla eléctricamente.

TTL: Sigla en inglés de transistor-transistor logic, es decir, "lógica transistor a transistor". Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales.

DHCP (Dynamic Host Configuration Protocol) es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

BAUD RATE (Tasa de Baudios): Es el número de unidades de señal por segundo. Un baudio puede contener varios bits, baudios con los bits por segundo, son conceptos distintos.

CABLE DE PAR TRENZADO (UTP): Medio de conexión usado en telecomunicaciones en el que dos conductores eléctricos aislados son entrelazados para anular las interferencias de fuentes externas y diafonía de los cables adyacentes.

CAPITULO I

INTRODUCCION

CAPITULO I

INTRODUCCION

1.1 ANTECEDENTES

En la actualidad el control de accesos para discapacitados es un área de presente interés dentro de nuestra sociedad ya que para el estado en los últimos años, y debido a las presentes leyes de la constitución ha empezado a tener vigencia en el reglamento de las empresas que se preste atención para encontrar una solución, además si consideramos que estas personas no conocen mucho de lo que se vive su alrededor y que además de la inseguridad que les persiste por no conocerlos se hace factible un objeto de localización de igual manera. Y observando de manera personal, en el Ecuador son varias las personas discapacitadas que solo utilizan una o dos manos para maniobrarse a si mismo para acceder a sus casas y realizar sus actividades sobrellevando sus vidas lo mas normal posible, pero y que sucede con el tema de seguridad domiciliaria para estos individuos, muchos optan por obtener servicios de monitoreo pero incluso así no tienen la seguridad de que sus objetos personales o equipos médicos se encuentren guardados de forma segura.

Para mejorar la situación de estas personas que solo utilizas sus extremidades superiores un dedo o un brazo, es una necesidad el proveer un sistema capaz de interactuar con los usuarios para permitirles el acceso a puertas de manera automática y darles un servicio de seguridad con el monitoreo de sus hogares, y localización en lugares desconocidos dependiendo del dispositivo que utilicen, y de esta manera se benefician con varios servicios en un solo equipo.

La orientación del proyecto es hacia accesos a puertas locales y a la protección de los bienes personales. Dado que en el presente tiempo existen diversos dispositivos móviles con capacidad de una conexión WIFI, no necesariamente un Smartphone, y en un costo no muy elevado se puede utilizar un servicio mencionado en el presente tema de este proyecto, sin tener que adquirir varios equipos para un solo servicio. Además a pesar de no disponer el dispositivo IPhone, a través de la aplicación web podrán disponer del servicio. De esta forma serán beneficiadas las personas que deseen contar con este control de acceso en especial personas discapacitadas que para estas es difícil acceder a lugares incluso dentro de sus hogares.

1.2 JUSTIFICACION

El presente proyecto estará bajo el uso de cámaras IP, un servidor de aplicaciones web, y una aplicación móvil para las personas discapacitadas, para esto se utilizará un IPHONE dado que este Smartphone presenta una buena interactividad con el usuario, y su trabajo es enteramente al uso de un solo dedo para realizar cualquier actividad en el mismo. Con lo cual se logre controlar y monitorear a través de una comunicación inalámbrica, ya sea dentro de una intranet local o ya sea a través de internet el uso de estos servicios, ya que este dispositivo Smartphone es uno de los mejores en su tipo y capaz de prestar mejores servicios que un móvil normal. Todo esto realizado de manera personal para uso en el hogar. La base del desarrollo de la aplicación móvil, estará bajo la plataforma XCODE, INTERFACE BUILDER Y IPHONE SIMULATOR para la aplicación que se implementará en el dispositivo físico, y la interacción con los demás dispositivos. Se realizará una aplicación Web para verificar las cámaras y el control de acceso de puertas de forma remota, en donde esta enviara la señal al puerto de comunicaciones Ethernet del equipo, y mediante un circuito eléctrico abrir las puertas indicadas; todo esto siempre y cuando se verifique en las cámaras, y el administrador decida dar el acceso para garantizar la entrada.

1.3 OBJETIVOS

1.3.1 Objetivo General.

Desarrollar un sistema que permita mantener un control de accesos y monitorear cámaras de manera interactiva en la pantalla de un dispositivo móvil IPhone.

1.3.2 Objetivos Específicos.

- Investigar alternativas tecnológicas utilizadas para el monitoreo y obtención de los datos.
- Consultar Equipos y cámaras ip que brinden los servicios necesarios.
- Conocer, procesar y analizar los datos que entrega el dispositivo móvil IPHONE.
- Diseñar una aplicación Web.
- Diseñar y desarrollar la aplicación en la plataforma XCODE de APPLE.
- Realizar la instalación y pruebas de evaluación requeridas del sistema.

1.4 SERVIDOR WEB

1.4.1 Definición.

Un servidor web es un programa diseñado para procesar conexiones bidireccionales o unidireccionales síncronas y asíncronas, en una aplicación almacenada en un servidor local. Donde los datos recibidos por el cliente que se conecta al servidor es compilado y ejecutado a través de un navegador web, el cual normalmente utiliza un protocolo HTTP de comunicaciones correspondiente a una de las capas del modelo OSI para demostrar la aplicación. El cliente puede acceder a la aplicación WEB mediante una dirección IP o ya sea a través de una dirección de Dominio. A medida que la conexión avanza, utiliza un protocolo de comunicación TCP en el cual realiza la petición de acceso al servidor y este otorga una respuesta de enlace con el equipo.

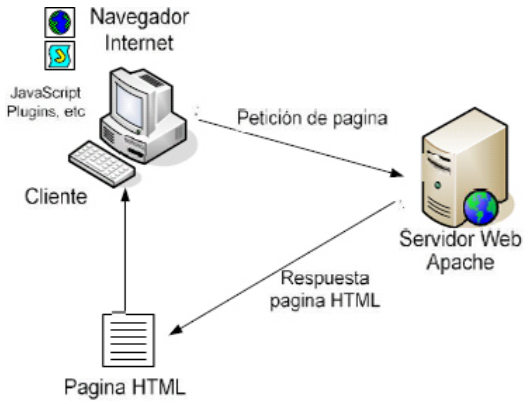


Figura. 1.1. Servidor Web[1]

Para demostrar el proceso de enlace éste realiza una petición HTTP al servidor de dicha dirección (www.midominio.com). El servidor envía única y exclusivamente el código HTML al cliente solicitante mas este no lo interpreta, una vez recibido el código en el equipo, este lo interpreta y lo demuestra en pantalla del navegador.

[1] Imagen obtenida de http://www.mailxmail.com/curso-php-mysql-sql-8/arquitectura-base-datos-web.

Para lograr conseguir esto se hace prescindible interpretar el código HTML, se pueden observar parámetros de texto, colores, imágenes, barras y objetos de la página web. El envío de código HTML no es lo único que se puede entregar por parte de un servidor, se pueden enviar partes de código HTML que son enviadas a aplicaciones Web las mismas que se ejecutan cuando existen pedidos o respuestas de tipo HTTP.

- El cliente web es quien ejecuta los pedidos desde la máquina del usuario. En su mayoría las aplicaciones proporcionadas corresponde a una programación de tipo Java en lo que hace referencia a las *Applets* o a ficheros de *JavaScript,* esos son procesos que ejecutan métodos y funciones de llamada que se muestran en la pagina web. De esta manera el servidor proporciona el código de las aplicaciones al cliente y éste a su vez mediante el navegador las ejecuta.

- Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje JavaScript y Java, aunque pueden añadirse más lenguajes mediante el uso de Plugins.

- El servidor web como función ejecuta la aplicación y una vez que ha sido procesada genera código HTML, el cual es enviado al cliente por medio del protocolo HTTP hacia la máquina que realiza el pedido del mismo.

Las mejores opciones a utilizar son las aplicaciones ejecutadas desde el mismo servidor debido que estas se ejecutan en el mismo y no en el equipo del usuario sin necesidad de ningún recurso agregado como en verdad ocurre en las aplicaciones de ejecución de JavaScript o Java, tan solo se necesita el navegador para ejecutar la aplicación.

1.4.2 Componentes de un servidor.

Comúnmente los servidores Web están constituidos por una serie de componentes muy similar a una PC, procesadores, memorias RAM y almacenamiento en un Disco Duro, la misma que cuenta con un sistema operativo y una serie de aplicaciones y librerías que trabajan en conjunto para generar la aplicación web.

Dependiendo de las necesidades exigidas al servidor se observan estos parámetros, con el objetivo de proveer un servicio como su nombre lo indica. Este equipo a diferencia de un PC normal trabaja de manera constante en lo posible durante 24 horas, 12 meses, y 365 días del año, con un debido mantenimiento de estos con un control de parámetros de funcionamiento en condiciones específicas, como lo son temperatura, niveles de energía, sistema de seguridad interna, que permiten que estos trabajen de manera óptima.

Normalmente los servidores son colocados en paneles atornillados llamados RACKs o armarios en los cuales los equipos son empotrados en un orden específico para conveniencia, estos servidores son llamados Blades o Cuchillas.

Si analizamos los componentes de un servidor serían las siguientes:

- Un procesador de alto rendimiento.
- Memoria Ram para el procesamiento de aplicaciones.
- Tarjetas de red dependiendo del servicio a ofrecer, ya que depende mucho el ancho de banda a ocupar, en todo caso se pueden utilizar dos puertos en 2 tarjetas, una para transferencia de datos, y otra

- para la recepción, utilizando al máximo cada puerto con el objetivo de filtrar la información entrante y la saliente.

- Almacenamiento o Storage que es donde se mantienen instaladas las aplicaciones y se van guardando respaldos de toda la información contenida en los mismos, obteniendo un margen de seguridad y anticipación ante pérdidas de información.

- Un sistema operativo con el cual se administra el acceso y el uso de las aplicaciones, normalmente estos deben contar con parámetros de seguridad elevados para evitar que otros usuarios o terceros participen del acceso al servidor y modifiquen o alteren las características del mismo.

1.4.3 Servidores Embebidos.

Un servidor embebido es un sistema informático de control el cual es diseñado para realizar una aplicación o función en específico. Se utilizan para propósitos generales diferentes a los que se les da a los computadores personales.

Poseen un diseño parecido a las computadoras, como por ejemplo en controles de acceso, cámaras IP, registradoras etc. Sus aplicativos como se mencionan se puede denotar por el siguiente gráfico que demuestra algunas de las aplicaciones:

Figura. 1.2. Esquema Aplicativo del Servidor Embebido.

Normalmente estos sistemas poseen una capacidad reducida y no son tan poderosos como las computadoras, además generalmente están constituidos sobre una plataforma de lenguaje ensamblador del micro controlador que poseen internamente. Para lo cual se utilizan compiladores basados en lenguaje C, C++ o Visual Basic para programarlo y ejecutar una acción en específico.

Para esto los sistemas embebidos utilizan comunicación serial sincrónica para controlar dispositivos periféricos, motivo por el cual estos son un poco mas lentos que los periféricos de los computadores. Los servidores embebidos son aptos para funciones básicas de operación que básicamente consiste en el envío de señales digitales de control y procesos de lectura de tal manera que interactúen para obtener el resultado deseado.

1.4.3.1 Estructura.

El servidor embebido está compuesto por un microprocesador en el cual en la parte de la memoria ROM se almacena un software de inicio el mismo que ya viene pre programado de fábrica para interactuar con la plataforma de programación de la empresa fabricante.

También posee una memoria Flash en la cual se guarda la configuración programada por el usuario para realizar la acción de control sobre el mismo. Tienen limitaciones al usar recursos ya que estos no son equipos tan complejos como computadoras, y son creados en grandes cantidades. Los Servidores Embebidos posee un sistema operativo en tiempo real RTOS. Una característica clave de un RTOS es el nivel de su consistencia respecto a la cantidad de tiempo que se tarda en aceptar y completar la tarea de una aplicación.

Un sistema operativo robusto posee en tiempo real tiene menos fluctuaciones que un sistema operativo ligero en ejecución.

El objetivo no es el diseño de alto rendimiento, sino más bien de una garantía de operación en aplicativo es decir que los tiempos será reducidos.[2]

Un sistema operativo en tiempo real tiene un avanzado algoritmo de programación. permite una mayor flexibilidad en su programación, un sistema operativo en tiempo real es más frecuente dedicado a un conjunto limitado de aplicaciones. Los factores clave en un sistema operativo de tiempo real son una mínima latencia de interrupción y del hilo de conmutación, un sistema operativo en tiempo real se valora más por lo rápido o lo predecible que puede responder que para la cantidad de trabajo que puede realizar en un período determinado de tiempo.

Trabajan de manera simple ya que poseen registros de 8 a 16 bits. El servidor embebido posee líneas de entrada y líneas de salida las mismas que pueden ser controladas por el usuario que lo requieran. Para controlar el dispositivo de hardware se utiliza un software que normalmente viene prediseñado por el fabricante, este es conocido como firmware que es un bloque de instrucciones de programa, que establece la lógica de máquina.

[2] Datos tomados de http://www.chibios.org/dokuwiki/doku.php?id=chibios:articles:rtos_concepts.

Mediante ella se controla los circuitos electrónicos de cualquier dispositivo, convirtiéndose en la interfaz de comunicación que recibe el dispositivo y su electrónica, siendo este el encargado de controlar a ésta última para ejecutar correctamente las sentencias de proceso. Cada uno de los componentes del Servidor Embebido son acoplados y permiten que su construcción sea simple y operen de forma correcta ante parámetros de diseño.

1.4.3.2 Funcionamiento.

El servidor embebido funciona este interactúan los sensores y los actuadores. Para esto el servidor posee líneas de entrada por donde ingresan los requerimientos para ser procesado, y de esta manera mediante las salidas ejecutar un efecto de aplicación sobre un recurso para satisfacer requerimientos.

Las características de estos servidores son comunes para cualquier tipo de servidor disponible, ya que comparten una diversidad de aplicaciones disponibles. Su arquitectura varía de acuerdo a la cantidad de equipos que se vayan a utilizar, ya sea hardware, sensores, u otros periféricos que interactúen con el esquema de funcionamiento del servidor embebido.

El servidor facilita operaciones de control básica en cuanto a una comunicación con un computador o dispositivo de acceso web sea el caso, no recarga gran cantidad de datos ni tampoco hace falta el uso de mucho procesamiento en cuando a demostraciones o gráficos que el dispositivo no lo utiliza. De acuerdo a su estructura este dispositivo mantiene una comunicación serial con periféricos, y de igual manera con una conexión en puerto RJ45 de una interfaz de red que le permite gestionar desde la Web.

En una cantidad limitada de puertos que pueden ser ya sean de entrada o de salida. Es por este motivo que el funcionamiento del mismo dependerá del tipo de aplicativo que se le requiera, tanto así como conexión e interfaz sobre el mismo. En la siguiente figura se puede observar la constitución de uno de estos servidores.

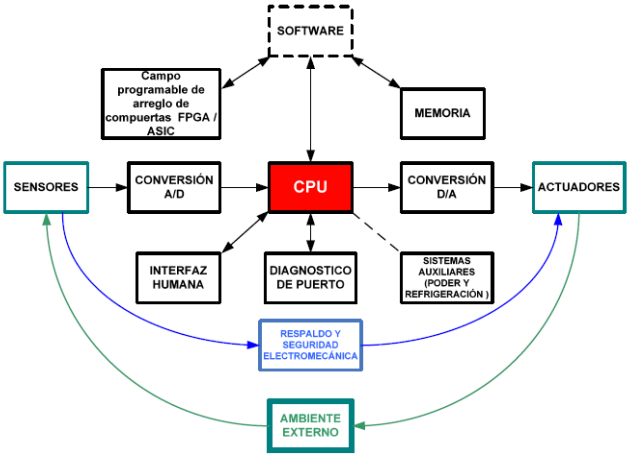


Figura. 1.3. Composición de un Sistema Embebido.[3]

Como se observan los componentes, básicamente el servidor embebido abarca toda la estructura tanto de hardware, puertos, memoria, interfaz de comunicaciones con el único objetivo de interactuar con los elementos externos como sensores y actuadores para un fin determinado. Al igual que un computador, el dispositivo verifica protocolos de conexión, de comunicación y de igual manera de funcionamiento, lo que lo identifica como un dispositivo simple pero de una prestación eficiente. El gráfico es explícito en su estructura y presenta su composición. En el se aprecian las funcionalidades que ofrece en función de sus periféricos y medios de comunicación para permitir el proceso de ejecución, ofreciendo un ambiente operativo óptimo ante las exigencias del diseño y ambiente, manteniendo la relación entre cada uno de ellos.

[3] Imagen tomada de ELECTRICAL & COMPUTER ENGINEERING, "Embedded System Design Issues (the rest of the story), http://www.ece.cmu.edu/~koopman/iccd96/iccd96.html#example

1.5 PAGINAS WEB

1.5.1 Definición.

Las páginas web son estructuras de código que al ser compiladas por un navegador estas muestran un despliegue hacia la pantalla del usuario con información del servidor. Las páginas web son de contenido informativo, y aplicativo, con lo cual el usuario tiene la posibilidad de interactuar con la misma para obtener el servicio otorgado por la misma. Estas pueden contener datos, como lo son texto, imágenes, videos, animaciones, y parámetros de hipervínculos que permiten el acceso a otras páginas web de igual manera.

Son arquitectura está basada en código HTML (HyperText Markup Language) como base de programación. Sin embargo pueden ser diseñadas en varios lenguajes de programación que utilizan diferentes tipos pero en sí, el contexto principal es dar a conocer información al usuario y brindarle un servicio de ser el caso.

La manera en que se utilizan las páginas web como se menciona son de carácter informativo, pero pueden mantener referencia a aplicativos a manera de puerta de enlace para aquello, de manera que el usuario tenga la facilidad de interactuar en ella y obtener un beneficio en general.

Es decir otorgar a través de la página no solo información sino también a un servicio que este requiera, por ejemplo compras online, aplicaciones, gestión remota, supervisión etc. Por ejemplo en el medio existen diferentes páginas web que ofrecen diferentes servicios y presentaciones, por ejemplo se puede apreciar una página web a continuación.

Figura. 1.4. Estilo de Página Web.[4]

Las páginas Web en su codificación contienen parámetros de Cabecera, y Cuerpo de la misma, espacio que es distribuido a lo largo de la plantilla donde se colocan los elementos de la página.

Muchas de estas páginas son almacenadas en Servidores Web los cuales mantienen al frente y en funcionamiento continuo el acceso para la página web, sin embargo existen entidades o usuarios no deseados que interfieren en los servidores, modificando el código de la misma, y proporcionando una imagen equivocada del sitio al que acceden los usuarios.

Es por eso que estas páginas son programadas en diferentes plataformas y formatos que de cierta manera ayudan a prevenir que terceros las modifiquen.

[4] Imagen obtenida de www.apple.com

1.5.2 Estructura de una página Web.

La estructura de una página web básica está codificada por tres parámetros principales que se observan a continuación:

<HTML>.- Es el lenguaje en el que se encuentra escrita la página Web, y es limitada solo a este tipo.

<HEAD>.- Este es el inicio de la página que corresponde a una cabecera la misma que contiene el título (<TITLE>) de la presentación de la página web, la misma que puede ir acompañada por una serie de rasgos que resalte el tema.

Debido que en el código solo se ingresa un solo título en el archivo, este debe ser lo mas atractivo y significativo ya que no se pueden colocar más sentencias dentro del mismo.

<BODY>.- Es el parámetro que encierra todo el contenido de la página, su estructura se muestra en la siguiente tabla:

Tabla 1.1. Estructura de una Página Web

<HTML>	Inicio del archivo
<HEAD>	Inicio de la cabecera
<TITLE>	Titulo del tema
</TITLE>	Finalización del título
</HEAD>	Final de la cabecera
<BODY>	Inicio del contexto
Sentencias HTML	
</BODY>	Finalización del contexto
</HTML>	Final del archivo

Esta es la estructura mínima necesaria para crear una página web, y mantiene cada nivel de acuerdo a la posición y prioridades de programación, que se siguen para tener el orden correcto de los elementos y atributos sobre la misma. Para que una vez compilada se pueda observar en cualquier navegador. Se puede observar en la siguiente figura la construcción de una pagina web básica, ejecutada desde el navegador.

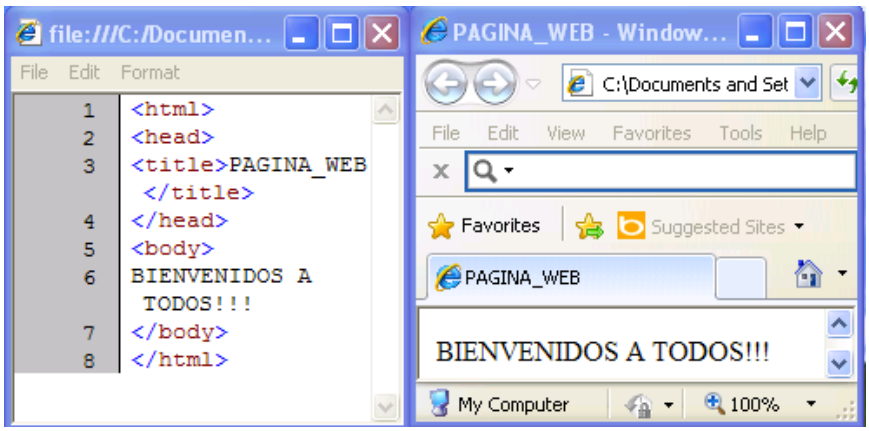


Figura. 1.5. Ejemplo básico de una Página Web.

En la figura se observa el código generado dentro de un archivo básico de tipo *.html*, y en el navegador la interpretación del mismo, a vista de los usuarios que lo acceden. Cabe destacar que en la programación de esta los comandos y etiquetas mantienen un formato simple y fácil de utilizar. Que con el tiempo se pueden ir manejando otros para lograr mejores resultados.

Como se ha destacado en la estructura de la página web, no solo permite observar texto sino, que se pueden realizar cambios en sus atributos, así como también el uso de otros elementos como imágenes, hipervínculos, videos, animaciones, etc. Siempre y cuando sea codificado el código producido en el lenguaje de la página web, para luego pasar a su interpretación.

1.5.2.1 Formatos de texto.

Cabecera de títulos.- Se pueden distinguir, cada título y subtítulo del documento, mediante la aplicación de cabeceras a través de las etiquetas de inicio <Hn> y final </Hn>. Ellas determinan los textos que serán incluidos de forma principal y subalternos. Algunos de los diferentes formatos de texto se pueden apreciar en la tabla como se muestra.

Tabla 1.2. Tipos de Cabecera HTML.[5]

Título	**Título**
Título	**Título**
Título	**Título**
Título	**Título**
Título	**Título**
Título	**Título**

Si se desean modificar otros parámetros de texto se pueden observar los siguientes:

Tabla 1.3 Tipos de Texto en HTML.[6]

Etiqueta	Se ve
<B>Texto en Negrita	**Texto en Negrita**
<I>Itálica</i>	*Itálica*
<B><I>Negrita e Itálica</i>	***Negrita e Itálica***
<U>Subrayado</u>	Subrayado Solo Explorer
<EM>Enfatizado	*Enfatizado*
<STRONG>Fuerte	**Fuerte**
<CODE>Code Texto</code>	`Code Texto`
<CITE> Citation Text</cite>	*Citation Text*
<KBD>Keyboard Text</kbd>	`Keyboard Text`
<SAMP>Sample Text</samp>	`Sample Text`
<TT>Teletype Text</tt>	`Teletype Text`
<VAR>Variable Element Text</var>	*Variable Element Text*
<BIG>Texto grande</big>	Texto grande
<SMALL>Texto pequeño</small>	Texto pequeño
_{Subindice}	Subíndice Solo Explorer
^{Superíndice}	Superíndice Solo Explorer
<BLINK> Texto intermitente</blink>	Texto intermitente Solo Netscape
<STRIKE>Texto tachado</STRIKE>	~~Texto tachado~~

[5] Tabla obtenida de http://platea.pntic.mec.es/~abercian/guiahtml/formatos.htm
[6] Tabla obtenida de http://platea.pntic.mec.es/~abercian/guiahtml/formatos.htm

Se puede modificar el tamaño de la fuente de dos maneras, por medio de un número del 1 al 7, donde mientras más bajo es, el tamaño es reducido. Se consigue con una etiqueta de la siguiente manera: TEXTO . Existe otro método mediante una referencia relativa: TEXTO este formato tiene un tamaño 4 ya que por defecto el tamaño es 3. Si se desea cambiar el tipo de fuente de cualquier texto en la página se debe colocar la siguiente sintaxis: TEXTO , la cual denotaría el texto en tipo Arial. Se pueden especificar colores de texto de la siguiente manera: TEXTO . Como se observa en este caso se pueden colocar varios atributos, separándolos en espacios cada uno de ellos. Para separar y definir párrafos de texto en un formato deseado se utilizan las siguientes etiquetas.

<P> Se utiliza para que los párrafos queden separados por una línea en blanco. Si solo quieres escribir un punto y aparte debes usar <BR>. Las marcas inicial y final son <P> y </P> y al colocarlos se admiten los atributos: ALIGN="left" , ALIGN="right" , ALIGN="center" y ALIGN="justify" que son utilizadas para la justificación del texto en la orientación deseada, ya sea hacia la izquierda que viene por defecto, o hacia la derecha, centro o justificado. Con la etiqueta **<PRE>** el texto insertado entre las marcas <PRE> y </PRE> es visualizado en el navegador con el formato en el que fue escrito en el archivo HTML. Esta etiqueta es muy útil para escribir más de un espacio o para hacer pequeñas tablas.

Se utiliza **
** Este elemento sólo tiene marca inicial e indica un salto de línea, es decir un punto y aparte sin separar el párrafo, esta no posee una etiqueta final que presente el cierre de la misma. También se emplea <HR> Se emplea para representar una línea horizontal, tampoco posee una etiqueta de cierre. Todas estas modificaciones en el texto logran resaltar aún mas las características de la página web, otorgando un contraste significativo que capte la atención del usuario que navega por la misma.

1.5.2.2 Links a otras páginas.

Dentro de las páginas Web se pueden colocar Hiperenlaces los cuales permiten acceder a otras páginas ubicadas en otros puntos de la World wide web. Su formato consiste en una etiqueta de la siguiente manera ***<A href="URL de la página a acceder">Texto del enlace</A>,*** este comando permite el acceso a los contenidos web de otras páginas.

Mediante la aplicación de este atributo se puede colocar la URL de la dirección a la cual se quiere ingresar, y se ejecuta con una acción otorgada por un texto o imagen que lo redireccione.

Es necesario especificar un texto que indique el contenido del archivo con el que se realizará el enlace, normalmente aparece resalta con azul. También se pueden colocar enlaces mediante el acceso con imágenes en la cual se denotará un borde azul que indica el hiperenlace señalado.

<A href="URL de la página a acceder"><IMG src="Imagen.formato"> texto</A> {IMG: Image} {src: Source}

Eso se puede visualizar de mejor manera en el ejemplo siguiente:

Tabla 1.4. Hipervínculos y Enlaces.

ETIQUETA	RESULTADO
<A href= "http://www.bing.com"> Pagina WEB de BING </A>	Pagina WEB DE BING
<A href="http://www.bing.com"><IMG src="bing.gif"></A> Bing	Bing BETA

1.5.2.3 Tablas.

Se pueden desarrollar tablas en el código HTML basta con colocar la sintaxis <table></table>. Estas se definen por fila a fila, a cada celda desde la parte superior izquierda que es en donde esta comienza, y las columnas se van colocando a medida que se vayan ubicando las celdas en las filas de modo automático. Para cada fila se puede indicar mediante el uso de la etiqueta <tr> </tr>. Las etiquetas <th> y <td> con sus correspondientes de cierre, indican filas individuales. Estas etiquetas <th></th> indica celdas se las denota en negrita que es la característica que indica que están como encabezado en la tabla. Y la <td> utilizada en celdas normales.

En la siguiente figura se puede observar una tabla en HTML con varios parámetros que modifican su apariencia:

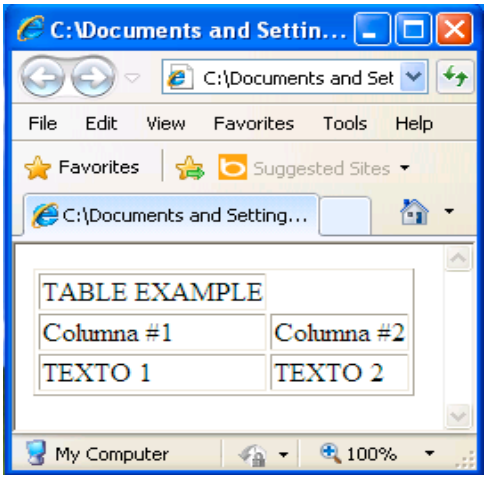


Figura. 1.6. Tabla en página Web.

Dentro de estas se puede hacer que una celda abarque de igual manera filas y columnas, únicamente se deben colocar dentro de la etiquetas mencionadas anteriormente los atributos:

Rowspan = Valor del número de filas que se desean colocar dentro de la celda.

Colspan = Valor del número de columnas que se desean colocar en la celda.

Una tabla puede poseer los siguientes atributos que modifican su apariencia.

Titulación de la tabla.- Que es un parámetro que titula a una tabla, ya sea utilizando texto normal o mediante etiquetas <caption> </caption>. En medio de ellas va el título, sin embargo esto no es necesario, pero si se desea colocarlo.

Se puede alinear el título de manera que lo ubique en la parte superior con la etiqueta *align="top"* , o establecerlo en la parte inferior de la tabla con la etiqueta *align="bottom".* Esta misma etiqueta nos permite alinear las celdas ya sea izquierdo (left), centrado (center), o al derecho (right).

Espaciado.- Se puede modificar el aspecto de la tabla ya sea el espacio entre las celdas o el ancho que existe entre ellas, o incluso el ancho de los bordes. Esto se puede conseguir mediante la etiqueta <table> y los parámetros:

- *Width* = Especifica el valor en pixeles como porcentaje del ancho de la pantalla, es es para dividir en un valor determinado la pantalla para que la tabla abarque esa porción. Se pueden utilizar también para especificar el ancho de las columnas ya sea con cualquiera de las etiquetas <td> o <th>.

- *Cellpadding* = Valor que genera el espaciado en pixeles entre el contenido y el borde de la celda, generalmente este valor es 1 por default.

- *Border* = Que indica el ancho del borde en pixeles. Si el valor de este es cero, indicaría que no existe ningún borde.

- *Cellspacing* = Valor que genera el espaciado en pixeles entre cada celda, generalmente este valor es 2 por default.

Mediante el uso de estos comandos se puede variar las características de presentación de las tablas, tanto en formatos de texto, espaciado, bordes, que marcan un pequeña diferencia para una agradable observación del mismo. Dependerá básicamente de la persona que clase de atributos que se desea, aunque existen casos en los que solo se acepta uno solo para compensar a otro. Para identificar el código contenido en la tabla de la página web de la figura 1.5. Observemos cuales fueron los comandos para dicha tabla:

Tabla 1.5. Código fuente de la Tabla en HTML.

```
<html> <body> <table width= “200” border= “1”>
<tr>
<td colspan=“2”><div aling=“center”>TABLE EXAMPLE </div></td></tr>
<tr>
<td> <div aling=“center”>Columna #1 </div></td>
<td> <div aling=“center”>Columna #2 </div></td></tr>
<tr>
<td> <div aling=“center”>TEXTO 1 </div></td>
<td> <div aling=“center”>TEXTO 2 </div></td>
</tr>
</table></body></html>
```

Esta es la fuente de la tabla contenida en la página web. Varios de sus atributos pueden ser modificados dependiendo de la necesidad y apariencia que le otorgue el usuario. Como se ha observado anteriormente, la estructura básica siempre se mantiene y a medida que se ajustan parámetros se empiezan a utilizar mayores comandos.

1.5.2.4 Ubicación de Imágenes.

Para que una página web capte la atención del usuario, se hace necesario el uso de imágenes, que además demuestren la idea de la página web, sin salir del contexto y objetivo del diseñador. Estas son ubicadas en diferentes lugares de la página, y pueden poseer atributos que generen un cambio, o permitan el acceso a otras páginas, o enlaces relacionadas a la misma. Se puede variar el tamaño de las imágenes, colores, y el formato permitido por las páginas, es muy amplio dentro del código HTML ubicando varias imágenes en una página Web.

Puede resaltar su presentación aún más, para definir los formatos básicos aceptados por los navegadores están las imágenes .png, .jpg, .gif, que dependerá básicamente de la calidad de estos, el formato y tamaño.

Visto esto desde una página web resultaría:

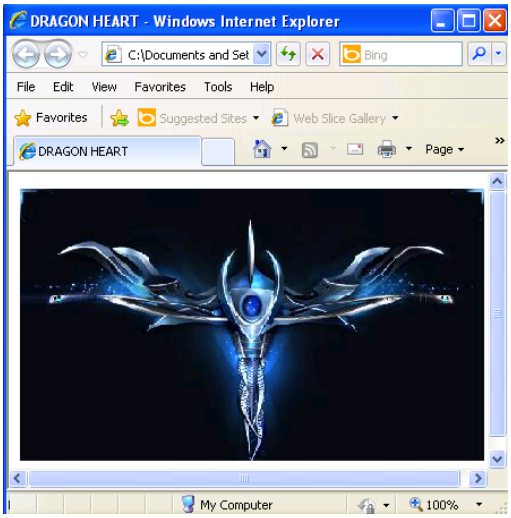


Figura. 1.7. Imagen en Página Web.

En la página web se observa la imagen insertada sin embargo si profundizamos en lo que se halla detrás de ella y miramos el código fuente se verifica lo siguiente a considerar como se observa a continuación:

Tabla 1.6. Código fuente de Imagen Insertada.

```
<html>
<head>
<title>DRAGON HEART</title>
</head>
<body>
<img src="Ubicación de la imagen/Nombre de la
imagen.jpg o el formato que posea" alt="Dragon
Heart" heigh="240" weight="410"/>
</body>
</html>
```

Observamos en las líneas de código la secuencia para la colocación de imágenes, para lo cual se dará en detalle las características de las etiquetas más utilizadas en el acoplamiento de imágenes en las páginas Web con código HTML. Las imágenes pueden interactuar unas con otras, pueden colocarse animaciones de tipo GIF, si se desea en el caso, inclusive el modificar los formatos puede ayudar a que la página no sea muy pesada en su totalidad para la presentación en la Web, muchos formatos pueden ser añadidos no es necesario mantener uno solo. Como complemento se recalca que esto es posible conseguir siempre y cuando se ubiquen las etiquetas con el orden correcto. Incluso es posible como se ha observado colocarles una referencia ya sea de una palabra, o un link a otra página, otorgando acceso a otra página, imagen, etc.

1.5.2.5 Formularios.

Los formularios son parámetros de información ingresada por el usuario para obtener acceso a otros datos o para el ingreso de información en un servidor a través del uso de una página web. Cuando estos campos ha sido llenados son enviados a una URL donde son procesados y almacenados para una vez ingresado, únicamente se verifiquen. Son creados mediante lenguaje HTML y al igual que los mecanismos anteriores, necesita del uso de etiquetas que lo identifiquen, para esto las etiquetas utilizadas son <form> </form> y dentro de estas se receptarán las variables o campos de entrada.

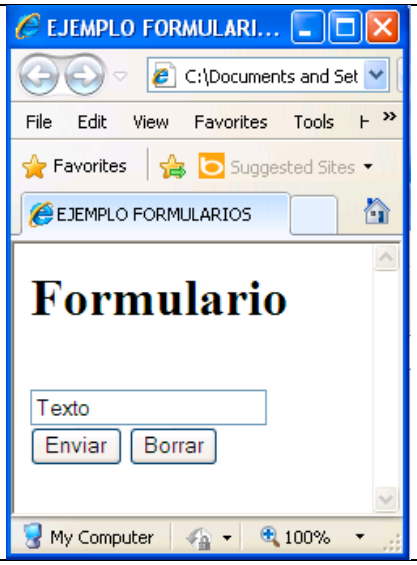
De la misma forma que los parámetros anteriores se pueden modificar los atributos del formulario, esto se puede describir de la siguiente manera:

Method = “ “ Que es un método de transferencia de variables, si es POST se envían por la librería STDIO.Get en la cual lo transmite por la dirección URL.

Action = “ ” Es en donde se especifica el evento que va a tratar con los campos que se van a enviar en el formulario, una dirección URL o un CGI (**Common Gateway Interface**).

Cada uno de ellos hace referencia a un tipo de envío de formulario, que dependerá del método aplicado. Básicamente el objetivo de ellos es ejecutar el envío de formularios para su procesamiento o almacenamiento en cualquiera de ser el caso. Para ilustrar de una manera más útil se puede observar en el siguiente ejemplo:

Tabla 1.7. Formulario Web

CODIGO FUENTE	VISTO EN PAGINA WEB
<html> <head> <title>EJEMPLO FORMULARIOS</title> </head> <body> <H1>Formulario</H1> <Form action = "mailto:Prueba" method= "POST"> <input type="Text" value="Texto"> <input type="submit" value="Enviar"> <input type="reset" value="Borrar"> </Form> </body> </html>	

Al enviarse los formularios se pueden utilizar algunas etiquetas como se demuestra a continuación.

INGRESO DE CAMPOS.- Para el ingreso de variables a los campos solicitados en la página se utiliza <input>. Para esto se pueden elegir entre los siguientes modelos:

- *Image.-* Cuando se haya seleccionado, el campo poseerá el valor dentro del punto de la imagen seleccionada.

- *Checkbox.-* Que es un visto en la casilla seleccionada, dependiendo de las opciones escogidas.

- *Radio.-* Un campo en el que al seleccionar genera un punto circular sobre ella, de igual forma que la anterior escogiendo los parámetros seleccionados.

SELECCIÓN DE CAMPOS.- Es una selección de valores de una lista de elementos, los cuales son contenidos dentro de las etiquetas *<SELECT> elementos</SELECT>.*

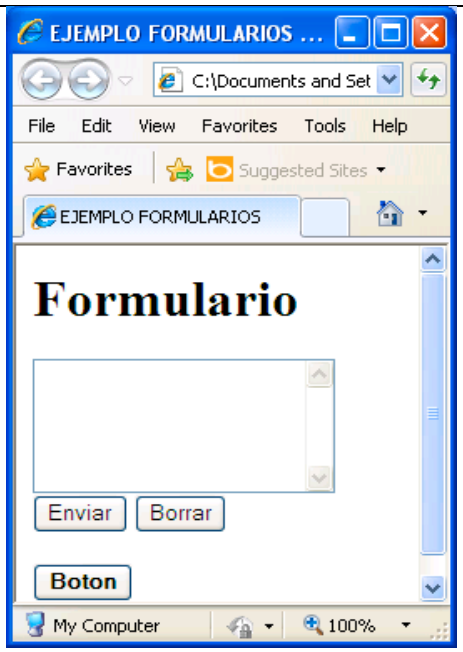
AREA DE TEXTO.- Varias líneas de texto que un usuario que visite la página puede ingresar en forma de comentarios, para lo cual se utilizan las etiquetas *<TEXTAREA> texto </TEXTAREA>.*

BOTONES.- Botones que efectúan el evento de envío de los datos contenidos en los campos de información de un formulario.

Cada uno de ellos puede realizar una acción determinada o interactuar entre sí, depende del proceso a realizar, lo que permite a la página web una gestión de contenidos y objetos de visualización para los usuarios.

En el siguiente ejemplo se puede verificar el uso de algunas modificaciones en los parámetros.

Tabla 1.8. Formulario Web Compuesto

CODIGO FUENTE	VISTO EN PAGINA WEB
<html> <head> <title>EJEMPLO FORMULARIOS</title> </head> <body> <H1>Formulario</H1> <Form action = "mailto:mail" method= "POST"> <textarea cols=20 rows=5 name="Texto">> </textarea> <input type="submit" value="Enviar">> <input type="reset" value="Borrar">> </Form> <button> Boton </button> </body> </html>	

1.5.3 Medios de Transmisión.

Las páginas web dependiendo de la exigencia y la complejidad de la estructura y presentación de la misma pueden utilizar diferentes medios para programarlos, entre las más comunes son en lenguaje HTML y de forma más dinámica con XHTML, que abarca mayores parámetros de programación en el diseño de la página web.

Varias empresas en el mercado utilizan nuevos lenguajes de programación interpretado para la creación de páginas web dinámicas, como lo es el PHP, PYTHON, XMLS, y otras que interactúan con bases de datos PostgreSQL, MYSQL. Los protocolos de conexión utilizados por las páginas web son IP, TCP, HTTP, HTTPS, FTP que son los que comúnmente se aplican la World Wide Web, que lo que las diferencia es el nivel que ocupan en la capa OSI. Ya que estos protocolos cumplen con las siguientes características.

IP (Protocolo de Internet).- Es el protocolo encargado de dirigir los paquetes de datos a través de la web, desde su punto de origen hasta su destino. Este es un protocolo que utiliza direcciones IP para buscar los caminos adecuados para enrutar el tráfico. Es utilizado en servicios de internet.

TCP *(Protocolo de Control de Trasmisión).-* Es un protocolo orientado a la conexión, es decir, se encarga de enviar de manera confiable los archivos desde el origen hacia su destino a través de las rutas designadas por protocolo IP. Los datos son fragmentados en datagramas que son enviadas a su destino para que una vez recibidos todos, estos sean unificados en el paquete original asegurándose que esté completa, y en el orden en que fueron transmitida libre de cualquier error.

HTTP *(Protocolo de transmisión de Hipertexto).-* Este protocolo se lo puede observar en las páginas web (http://), de manera que es el tipo de información de la Web, que sigue un esquema de solicitud y respuesta entre el cliente y el servidor, de manera que en el protocolo que se le solicita este sea correspondido.

HTTPS *(Protocolo Seguro de transmisión de Hipertexto).-* Este protocolo es una versión segura del protocolo HTTP, este tipo de protocolo encripta los datos para su transmisión y recepción, comúnmente es utilizado por entidades importantes como en los Bancos para las transacciones bancarias.

FTP *(Protocolo de trasmisión de archivos).-* Es un protocolo utilizado para la transferencia de archivos, normalmente los Web-Master gestionan sus archivos a través de este protocolo. Por lo tanto una vez determinada la plataforma que se empleará para la programación de la página web, al finalizar, se verificará su funcionamiento con los protocolos mencionados anteriormente.

1.6 CAMARAS IP

1.6.1 Definición.

Las cámaras IP son dispositivos que emiten imágenes a través de la red, sin necesidad de utilizar un computador, trasmitiendo directamente en la intranet o internet. Su funcionalidad y tamaño está dada por su construcción y funcionalidades, además de un software incorporado que ayuda al uso de la misma. Estos dispositivos fueron desarrollados con fines de seguridad y monitoreo ya que de esta manera se puede observar continuamente las actividades dentro de una localidad. Mantienen protocolos de seguridad. Y solo su dueño puede verificarlas. Mediante el uso de las cámaras IP se puede visualizar lo que está pasando en tiempo real, ya que se conecta hacia el internet mediante una dirección IP que es poseedora la cámara. Permiten al usuario ubicar la cámara en una determinada localidad y ver en vivo el video en otro lugar a través del internet. Al existir la incorporación de estos dispositivos en la parte de monitoreo y seguridad, se acoplan a nuevas aplicaciones de Smartphone que interactúan aún mas, mejorando la portabilidad en el manejo y uso, además de ofrecer un mejor control de las actividades del área de acción.

A continuación se pueden observar los tipos de cámaras IP más utilizadas:

1.6.1.1 Cámara IP Fijas.

Figura. 1.8. Cámara IP fija.[7]

[7] Imagen obtenida de http://www.axis.com/es/products/video/camera/fixed/index.htm

Este tipo de cámaras se utilizan para monitorear áreas exactas sin tener la necesidad de mover la posición de la cámara, de tal manera que cubra la superficie de interés en una ubicación estratégica. Su costo no es tan elevado, sin embargo mucho depende la calidad de recepción de la imagen.

1.6.1.2 Cámaras IP Móviles.

Figura. 1.9. Cámara IP móvil.[8]

Este tipo de cámaras poseen una flexibilidad en cobertura de áreas ya que el usuario puede monitorear 360 grados de forma horizontal, y 180 grados verticalmente, lo que le da movilidad y monitoreo móvil. Su costo es un poco elevado y depende principalmente de la calidad de imagen que recepta y movilidad del dispositivo.

1.6.1.3 Cámaras IP Infrarrojas.

Figura. 1.10. Cámara IP infrarroja.[9]

[8] Imagen obtenida de http://www.axis.com/es/products/cam_214/index.htm
[9] Imagen obtenida de http://www.conmutelgdl.mex.tl/152286_Camaras-IP.html

Este tipo de cámaras además de poseer movilidad, también son de modo nocturno, permitiendo a los usuarios monitorear en la noche las áreas deseadas. Su costo es elevado y depende de igual forma la calidad de la imagen y las capacidades para diferenciar objetos en la noche.

Como se ha analizado el acceso a las imágenes de la cámara es restringido, y solo las personas autorizadas pueden observarla y monitorear la actividad. Además se puede garantizar un acceso total y abierto del video en directo si se desea colocar en una página web de la compañía para que cualquier navegador acceda a ella.

Poseen muchas otras funcionalidades dependiendo de la necesidad, además con el pasar de los años se han desarrollado aplicaciones y otros dispositivos que interactúan con este medio.

Los diferentes tipos de cámaras son diseñadas para distintos propósitos, sin embargo el objetivo final es el de monitorear el entorno de desarrollo, para mejorar la seguridad.

1.6.2 Características.

Una cámara IP en la red, utiliza su propio procesador, que es el que le permite procesar y emitir las imágenes de video desde sí misma, comprimiendo el video, y transmitirlo hacia sus destinos. La utilidad de esta función puede estar determinada por las siguientes características:[10]

- Envío de correos electrónicos con las imágenes tomadas.
- Activación mediante movimiento de la imagen.

[10] Datos obtenidos de http://www.lsb.es

- Activación mediante movimiento de sólo una parte de la imagen.

- Creación una máscara en la imagen, para ocultar parte de ella o colocar un logo. O simplemente por adornar.

- Activación a través de otros sensores.

- Control remoto para mover la cámara y apuntar a una zona.

- Programación de una secuencia de movimientos en la propia cámara.

- Posibilidad de guardar y emitir los momentos anteriores a un evento.

- Utilización de diferente cantidad de fotogramas según la importancia de la secuencia. Para conservar ancho de banda.

- Actualización de las funciones por software.

- Las cámaras IP permiten ver en tiempo real qué está pasando en un lugar, aunque esté a miles de kilómetros de distancia. Son cámaras de vídeo de gran calidad que tienen incluido un ordenador a través del que se conectan directamente a Internet.

Una cámara IP (o una cámara de red) es un dispositivo que contiene:

- Una cámara de vídeo de gran calidad, que capta las imágenes.
- Un chip de compresión que prepara las imágenes para ser transmitidas por Internet.
- Un ordenador que se conecta por sí mismo a Internet.

La nitidez de la imagen en las cámaras dependerá del lente que esta posea, así como la resolución de la misma, normalmente existe la posibilidad de que se utilicen cámaras de baja calidad en imagen pero con un monitor de visualización con mejora en la imagen, lo que beneficia en cuanto a uso de ancho de banda si esta es IP, otorgando una capacidad de interacción con múltiples cámaras y optimizando el ancho de banda para estas. Si las cámaras IP pueden ser monitoreadas desde un servidor remoto, aunque ahora estas pueden almacenar el video durante un periodo de tiempo prolongado, con una memoria extendida interna, convirtiéndola en su propio servidor de almacenamiento.

1.6.3 Funcionamiento.

Una cámara IP tiene incorporado un ordenador, pequeño y especializado en ejecutar aplicaciones de red. Por lo tanto, la cámara ip no necesita estar conectada a un PC para funcionar. Esta es una de sus diferencias con las denominadas cámaras web. Tiene su propia dirección IP y se conecta a la red como cualquier otro dispositivo; incorpora el software necesario de servidor de web, servidor o cliente FTP, de correo electrónico y tiene la capacidad de ejecutar pequeños programas o scripts.

También incluye entradas para alarmas y salida de relé. Las cámaras de red más avanzadas pueden equiparse con muchas otras funciones de valor añadido como son la detección de movimiento y la salida de vídeo analógico.

Las cámaras IP incorporan todas las funciones de una cámara de video y añaden más prestaciones. La lente de la cámara enfoca la imagen en el sensor de imagen (**CCD**). Antes de llegar al sensor, la imagen pasa por el filtro óptico que elimina cualquier luz infrarroja y muestra los colores correctos. Las cámaras de día/noche disponen de un filtro de infrarrojos automático, este filtro se coloca delante del **CCD** sólo cuando las condiciones de luz son adecuadas proporcionándonos de esta manera imágenes en color.

Cuando las condiciones de luz bajan este filtro se desplaza y la cámara emite la señal en blanco y negro produciendo más luminosidad y de esta manera podemos iluminar la escena con luz infrarroja y ver en total oscuridad. El sensor de imagen convierte la imagen, que está compuesta por información lumínica, en señales eléctricas. Estas señales eléctricas se encuentran ya en un formato que puede ser comprimido y transferido a través de redes.

Como las cámaras de vídeo convencionales, las cámaras IP gestionan la exposición (el nivel de luz de la imagen), el equilibrio de blancos (el ajuste de los niveles de color), la nitidez de la imagen y otros aspectos de la calidad de la imagen. Estas funciones las lleva a cabo el controlador de cámara y el chip de compresión de vídeo. Las cámaras IP comprimen la imagen digital en una imagen que contiene menos datos para permitir una transferencia más eficiente a través de la Red, cámaras MPEG4.

1.6.3.1 Aplicaciones.

Algunas de las aplicaciones mas frecuentes de las **cámaras IP** son la vigilancia de:[11]

- **Viviendas**, permitiendo monitorear una vivienda propia desde la oficina, desde un hotel, o desde cualquier lugar.

- **Negocios**, permitiendo controlar por ejemplo varias sucursales de una cadena de tiendas, gasolineras.

- **Instalaciones industriales**, almacenes, zonas de aparcamiento, Muelles de descarga, accesos, incluso determinados procesos de maquinaria o medidores.

[11] Obtenido de http://www.lsb.es/camaras_ip.htm

- **Hostelería**, Restauración, Instalaciones deportivas.

- **Lugares Turísticos**, cada día es mas frecuente que Organismos oficiales, como Comunidades Autónomas, Ayuntamientos, promocionen sus zonas turísticas, o lugares emblemáticos de las ciudades, instalaciones deportivas, implementado en sus páginas Web las imágenes procedentes de Cámaras IP estratégicamente situadas en esos lugares.

Esta cámaras otorgan un control total a los usuarios, permiten una monitorización de toda una área de cobertura, manteniendo la seguridad y un riesgo bajo, lo que posibilita aplicativos muy amplios para su uso. Es algo nuevo en el mundo y su conocimiento es limitado aún para los beneficios que se pueden obtener de estos.

1.7 DISPOSITIVO MÓVIL SMARTPHONE

Un dispositivo móvil se puede definir como un aparato electrónico de un tamaño pequeño, con capacidad de procesar la información, ya sea con una conexión continua o una de tipo aleatoria a una red, con la presencia de una memoria limitada que en algunos casos puede ser expandida, diseñado específicamente para llevar a cabo funciones de tipo general.

1.7.1 Definición.

“Es un teléfono inteligente que puede comunicarse a través de Wi-Fi, bluetooth, conexión a internet, envío de mensajería, e-mails. Generalmente se define como dispositivo electrónico de mano que integra la funcionalidad de un teléfono celular, PDA o similar. Generalmente se realiza añadiendo funciones de teléfono a un PDA existente o añadiendo funcionalidades "inteligentes", como las funciones del PDA, en un teléfono celular.

Una característica clave de un Smartphone es que las aplicaciones adicionales pueden ser instaladas en el dispositivo. Las aplicaciones puede ser desarrolladas por el fabricante del dispositivo, por el operador o por cualquier empresa desarrolladora de software**.** Un teléfono móvil Smartphone está diseñado sobre una plataforma móvil normalmente generado por el fabricante, con una capacidad de procesamiento y conectividad avanzada que lo acerca a una comparación de una mini computadora. Los primeros teléfonos Smartphone combinaban funciones de un PDA con teléfono móvil."[12]

Estos equipos incorporan funciones parecidas a la de los ordenadores convencionales, como los son reproductores de video, música como los dispositivos MP3, cámaras digitales, grabadores de video, y posicionamiento global GPS. Convirtiéndolo en un dispositivo de alto rendimiento y capacidad fuera de las características convencionales de un teléfono.

A continuación se muestran algunos dispositivos móviles, cada uno con su respectivo sistema operativo, estos ofrecen servicios y aplicaciones similares para los usuarios finales.

Figura. 1.11. Smartphones[13]

[12] Definición obtenida de http://www.gsmspain.com/glosario/?palabra=SMARTPHONE
[13] Imagen obtenida de http://www.moormobile.com/smartphone-resources/the-top-smartphones-for-2011.html

En los avances recientes de tecnología se incorporan pantallas táctiles de alta resolución que trabajan a base de celdas capacitivas que al tocar la estática de los dedos crea una variación de potencial en la pantalla, y el dispositivo lo procesa e identifica donde fue realizado el toque. Además de este tipo, están aquellos que incorporan mandos de voz lo cual se vuelve más optimo al momento de manejar. Poseen navegadores Web que pueden mostrar las páginas web al igual que en un PC normal, y aun mas en páginas optimizadas para la visualización móvil con una alta velocidad de procesamiento de datos tanto por red Móvil como para red WIFI.

1.7.2 Características.

Los sistemas móviles más comunes operativos (OS) utilizadas por los teléfonos inteligentes modernos incluyen iOS de Apple, Android de Google, Microsoft, Windows Phone, Symbian de Nokia, BlackBerry OS de RIM, y embebidos distribuciones de Linux. Estos sistemas operativos se puede instalar en muchos modelos de teléfonos diferentes, y por lo general cada dispositivo puede recibir varias actualizaciones de software del sistema operativo durante su vida útil.

La distinción entre smartphones y teléfonos de varias capacidades pueden ser vagos y no existe una definición oficial de lo que constituye la diferencia entre ellos. Una de las diferencias más significativas es que las interfaces de aplicaciones avanzadas de programación (API) en los teléfonos inteligentes para el funcionamiento de las aplicaciones de terceros puede permitir que las aplicaciones tengan una mejor integración con el sistema operativo del teléfono y de hardware que es típico de los teléfonos con funciones. En comparación, los teléfonos con funciones más comúnmente se ejecutan en el firmware propietario, con el apoyo de software de terceros a través de plataformas como Java ME. Una complicación adicional en la distinción entre Smartphone es que con el tiempo las capacidades de los nuevos modelos de los teléfonos con funciones puede aumentar hasta superiores a las de los teléfonos que había sido promovido como los teléfonos inteligentes en el pasado.

Las características más comunes de estos dispositivos son en hardware, la posesión de cámaras, reproductores mp3, grabadoras de video, la apertura de documentos de todo tipo, navegadores de internet, entre otras.

La capacidad de conexión al internet en los equipos depende en general de la tecnología en la que hayan sido fabricados. Entre las ultimas tecnologías celulares de conexión se tiene 3G (HSPA) y 4G (LTE) que son las que se están utilizando actualmente. Sus características se definen por un rango de velocidad de conexión que varían entre ellas tanto como de down link y up link.

1.7.3 Protocolos de conexión a la WEB.

Los teléfonos móviles para poder realizar la conexión a la web cuentan con protocolos HTTP, al igual que las computadoras, estos pueden poseer aplicaciones Web en un servidor con aplicaciones Servlets que al generar la conexión otorgan los datos de los formularios contenidos en el dispositivo móvil. Al poseer navegadores, todos trabajan con el lenguaje HTML que es un código general en cualquier página WEB básica. Por lo tanto la visualización es posible en el equipo dependiendo de igual manera en su capacidad de decodificación de la página para su presentación en el móvil incluyendo la resolución de pantalla en el mismo y de igual forma que una computadora los dispositivos se conectan mediante direcciones IP, cada dispositivo posee una dirección MAC que lo identifica.

A comienzos del ingreso del internet en los celulares estos lo realizaban mediante una conexión de tipo WAP (Wireless Application Protocol) que no es más que otro protocolo de conexión a internet de bajas velocidades y con un acceso a contenido básico de internet, soportando código HTML para servicios de información y acceso a contenidos. En la siguiente imagen se aprecia el acceso a contenido Web, que prácticamente es muy parecido al de una computadora.

Figura. 1.12. Acceso Web en un móvil.

El acceso a contenido web y la visualización del mismo dependerá básicamente de los diferentes códigos con los que trabaje el dispositivo, es decir podemos abarcar un amplio rango de páginas codificas en diferentes plataformas las cuales sean compatibles con los protocolos y códigos del dispositivo Smarphone que se trabaje, tanto así como desde un lenguaje HTML hasta un PHP, HTML5 y muchos otros. Lo que posibilita la visualización del contenido en el mismo.

CAPITULO II

INTRODUCCION AL SISTEMA OPERATIVO DE MOVILES

CAPITULO 2

INTRODUCCION AL SISTEMA OPERATIVO DE MOVILES

2.1 CONCEPTOS GENERALES

Con la salida de los teléfonos móviles al mercado en un inicio el único objetivo era el de brindar un servicio de comunicación entre usuarios, por lo cual el software solo permitía la generación de llamadas y mensajes, esto con un sistema propio de cada empresa que manufacture celulares. Sin embargo con la demanda de mayores servicios y aplicaciones, empezaron a utilizar otras mejoras como lo eran cámaras, y visualización de imágenes, video, música.

Lo que provocó que cada empresa comience a realizar estudios en el avance de hardware, lo que causó que se necesite un sistema operativo en los móviles con un manager de aplicaciones. A partir de este la demanda empezó a crecer aún mas convirtiéndose no solo en una competencia por un mejor teléfono en cuanto a hardware, sino también en el avance de su sistema operativo. Lo cual algunas de las empresas con el tiempo descargaron su sistema operativo y contrataron a otras empresas para que lo instalen en sus equipos. Al final de todos los sistemas más reconocidos a nivel mundial son:

Android.- Que es un sistema operativo basado su kernel (Núcleo) basado en Linux desarrollado por los creadores de GOOGLE, este software es libre y posee muchas versiones con características similares, este sistema es utilizado en teléfonos, tablets, simuladores, máquinas virtuales.

IOS de Iphone.- Es un sistema operativo desarrollado por Apple para su dispositivos móviles, este se puede encontrar instalado en teléfonos, tablets, y otros dispositivos portátiles como reproductores Mp3 IPOD de la misma marca.

Windows Mobile.- Es un sistema operativo desarrollado por la corporación Microsoft, para dispositivos móviles, tablets, Palms, y algunos reproductores de música y videos.

Sistemas operativos de Nokia como Symbian o de Blackberry, Motorola, HTC, entre otras marcas empezaron a perder usuarios por la presencia de mejores sistemas que ofrecían mejores servicios y aplicaciones como los mas resaltados anteriormente. Dentro del estudio se toma en cuenta los sistemas operativos más utilizados. Sin embargo, se toma énfasis en el sistema operativo de IPHONE ya que sus recursos son fáciles de utilizar y su arquitectura y librerías para su uso muy amplio.

2.1.1 Sistema Operativo Android.

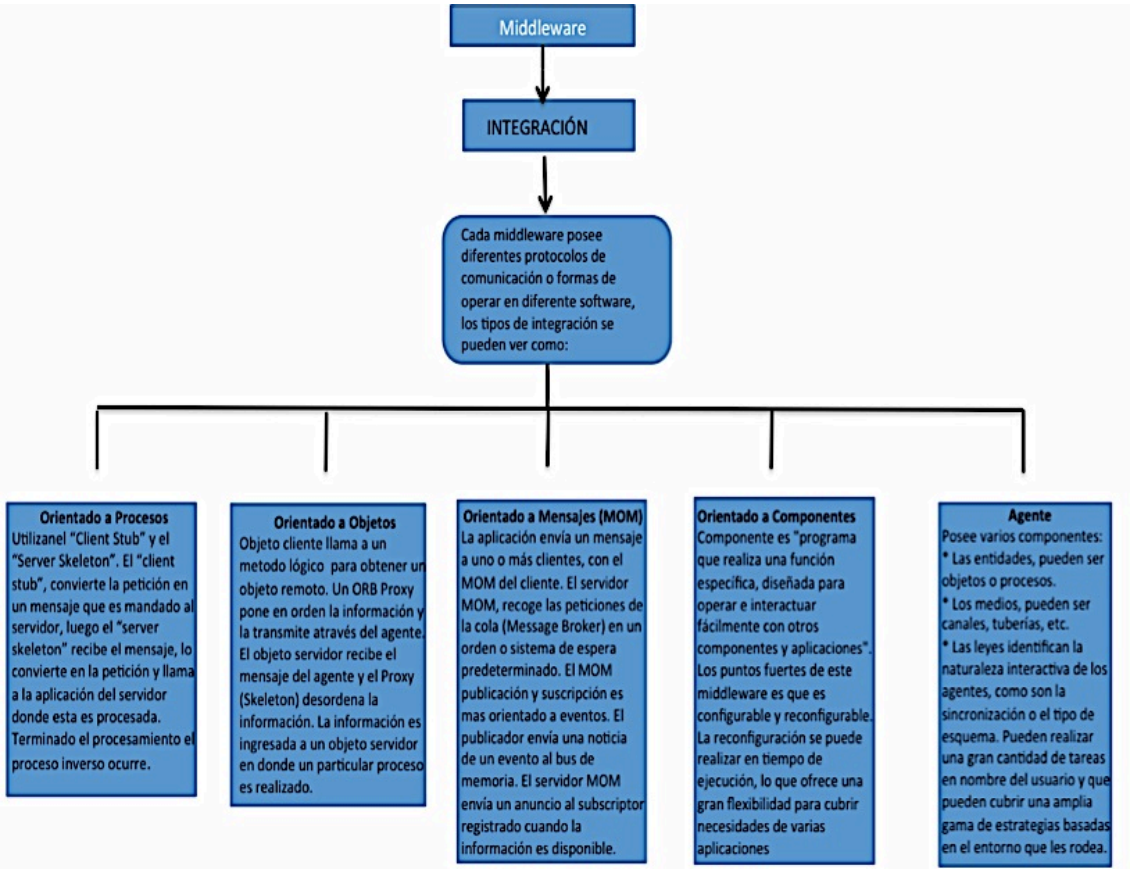
Android es un sistema operativo móvil basado en un núcleo de UNIX, con una interfaz de LINUX modificada para móviles, que junto con aplicaciones middleware, está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, y otros dispositivos. Utiliza aplicaciones de Middleware que consisten en un software que es complementario a una aplicación, capaz de comunicarse y operar con otras aplicaciones, o dispositivos de hardware, o software de otros sistemas operativos, son llamadas así siendo las intermediarias en la parte de comunicación del dispositivo entre aplicaciones.

Estas aplicaciones funcionan como una capa de abstracción de software distribuida, es decir que se sitúa entre las capas de aplicaciones y las capas inferiores del sistema operativo y red.

Abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una interfaz de programación de aplicaciones ***API*** de fácil manejo de aplicaciones distribuidas. [14]

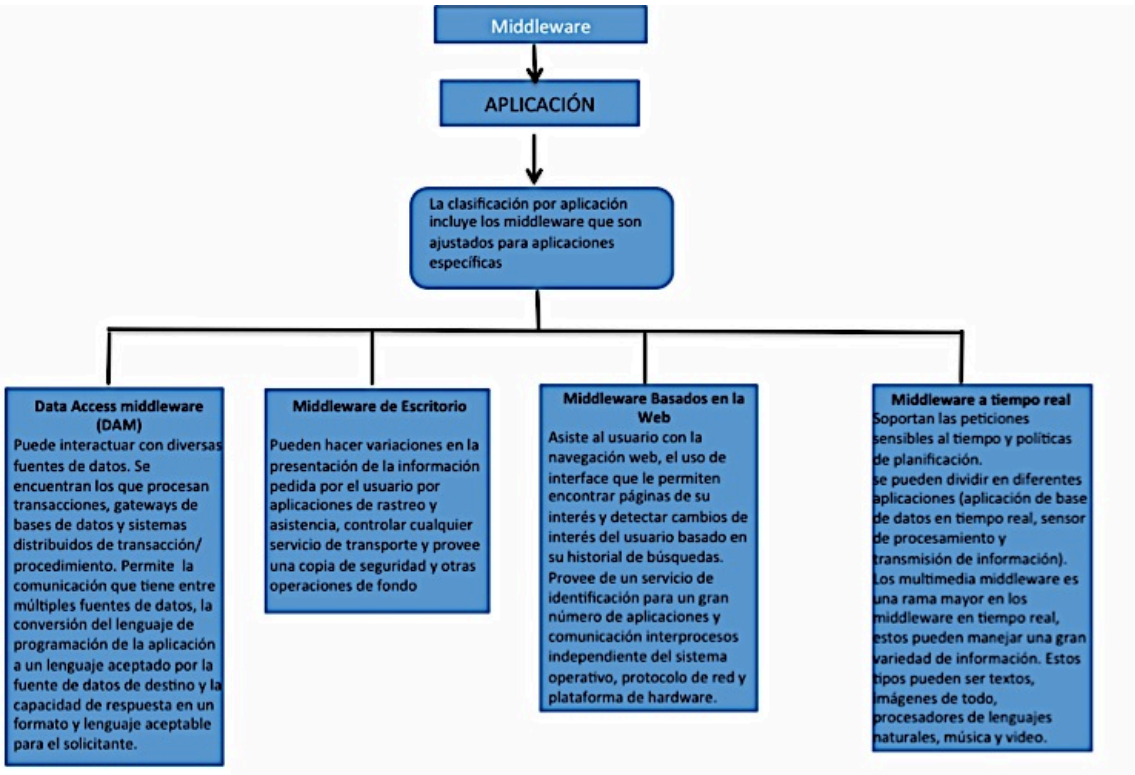
Su compatibilidad con aplicaciones puede contener una amplia gama de recursos, que complementan al sistema operativo al momento de ejecutar aplicaciones de terceros, en ellas se puede manifestar JAVA, o FLASH, que de igual forma se acoplan al sistema operativo de Android.

Tabla 2.1. Middleware como etapa de integración



[14] Extraído de http://www.middleware.org/

Tabla 2.2. Middleware en etapa de Aplicación



Los servicios de middleware proporcionan un conjunto más funcional de la API para permitir una aplicación a:

- Localizar claramente a través de la red, proporcionando así una interacción con otro servicio o aplicación
- Los datos filtrados para que sean utilizables en un ambiente público a través de anonimato de procesos para la protección de la privacidad.
- Ser independiente del servicio de red
- Ser fiable y siempre disponible
- Añadir los atributos complementarios como semántica en comparación con el sistema operativo y servicios de red.

En sus inicios fue desarrollado por Android Inc. con un sistema operativo basado en Linux y orientado a dispositivos móviles, como teléfonos inteligentes y tablets.

"La compañía fue adquirida por la multinacional Google en julio de 2005. El desarrollo de Android continuó de la mano de Google y la Open Handset Alliance, que es una alianza comercial de 84 compañías para desarrollar estándares abiertos para dispositivos móviles. Algunos miembros son Google, HTC, Dell, Intel, Motorola, Qualcomm, Texas Instruments, Samsung, LG, T-Mobile, Nvidia y Wind River Systems.

Estos programas en su mayoría están codificados en lenguaje de programación JAVA, y no es un sistema que se encuentre libre de programas no deseados, virus, spyware etc. Sin embargo todos estos programas son descargados de otros servidores ya que son de libre distribución. Además el código de Android en su mayoría está bajo la licencia Apache, que corresponde una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic." [15]

[15] Extraído de http://www.telcommunity.com/category/internet-y-telecomunicaciones/

2.1.1.1 Arquitectura del Sistema Operativo Android.

El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Aplicaciones: las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Framework de aplicaciones: los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.

- Un amplio conjunto de puntos de vista y extensible que se pueden utilizar para construir una aplicación, incluyendo listas, rejillas, cajas de texto, botones, e incluso un navegador web.
- Los proveedores de contenido que permiten a las aplicaciones acceder a datos de otras aplicaciones (como los contactos), o para compartir sus propios datos.
- Un administrador de recursos, facilitar el acceso a los recursos que sin código tales como cadenas localizadas, gráficos y archivos.
- Un Administrador de notificaciones que permite a todas las aplicaciones mostrar alertas personalizadas en la barra de estado.
- Un gestor de actividad que gestiona el ciclo de vida de las aplicaciones y proporciona una navegación común backstack.

Bibliotecas: Android incluye un conjunto de bibliotecas de C/C++usadas por varios componentes del sistema.

Estas características se exponen a los desarrolladores a través del Framework de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

Sistema de biblioteca de C - una BSD derivada de la implementación de la biblioteca del sistema estándar C , atento a las embebidos basados en dispositivos Linux.

Medios de comunicación - basados en OpenCore PacketVideo, las bibliotecas soportan la reproducción y grabación en varios formatos de audio y de vídeo, así como archivos de imágenes estáticas, incluyendo MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.

Administrador de superficie - gestiona el acceso al subsistema de pantalla y sin problemas 2D y 3D de capas gráficas de múltiples aplicaciones.

Bibliotecas 3D - una implementación basada enOpenGL ES 1.0 API, las bibliotecas utilizara celeración 3D por hardware (donde esté disponible) o el incluido, rasterizador altamente optimizada de software 3D.

FreeType - mapa de bits y la renderización de fuentes vectoriales SQLite - un motor de base de datos relacional potente y ligero disponible para todas las aplicaciones.

Runtime de Android: Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

Núcleo Linux: Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software ejecuta procesos de forma nativa. En el siguiente cuadro se observan las características del sistema operativo.

Tabla 2.3. Características del S.O. Android.

Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE,IDEN, CDMA, EV-DO, UMTS,BLUETOOTOOTH, WI-FI, LTE y WiMAX.
Navegador Web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador obtiene una puntuación de 93/100 en el test Acid3.
Soporte Java	Apesar de que la mayoría de las aplicaciones están escritas en Java, no hay una maquina virtual Dalvik en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dipositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner
Soporte Multimedia	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4),MPEG-4 SP, AMR, AMR-WB(en un contenedor 3GP),AAC, HE-AAC (en contenedores MP4 o 3GP),MP3, MIDI, Ogg Vorbis,WAB, JPEG, PING, GIF y BMP.
Soporte para Streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player
Soporte para Hardware	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2D y 3D.
Entorno de Desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4, 3.5 o 3.6) usando el plugin de Herramientas de Desarrollo de Android.
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o superiores mediante aplicaciones disponibles en el Android Market, por ejemplo PdaNet). Para permitir a un PC usar la conexión 3G del móvil android se podría requerir la instalación de software adicional.

2.1.2 Windows Mobile.

Windows Mobile es un sistema operativo desarrollado por Microsoft, y diseñado para su uso en Smartphones. Se basa en el núcleo del sistema operativo Windows CE que quiere decir Windows Embedded Compact (anteriormente conocida como Windows Embedded CE o Windows CE)es la versión de Windows Embedded para equipos muy pequeños y los sistemas integrados , incluyendo la electrónica de consumo como dispositivos set-top boxes y consolas de juegos . Windows Embedded Compact es un sistema modular en tiempo real del sistema operativo con un núcleo especializado que puede ejecutar en menos de 1 MB de memoria.Viene con el Platform Builder herramienta que puede ser utilizado para agregar módulos a la imagen de instalación para crear una instalación personalizada, en función del dispositivo utilizado.

Windows Embedded Compact está disponible para arquitecturas de procesador ARM , MIPS , SuperH y x86. Microsoft también ha puesto a disposición una versión especializada de Windows Embedded Compact, conocido como Windows Mobile , para su uso en teléfonos móviles. Se trata de una imagen personalizada de Windows Embedded Compact, junto con módulos especializados para su uso en teléfonos móviles. Windows Mobile está disponible en cuatro variantes: Windows Mobile Classic (para Pocket PC), Windows Mobile Standard (para teléfonos inteligentes) y Windows Mobile Professional (para PDA / Pocket PC Phone Edition) y Windows Mobile for Automotive (para los sistemas de comunicación / entretenimiento / información utilizadas en los automóviles).

Las versiones modificadas de Windows Mobile también se utilizaron para centros de medios portátiles.

Windows Mobile ha sido sustituido por Windows Phone 7 , que también está basado en Windows Embedded Compact, pero no es compatible con cualquiera de los productos anteriores.

Con Windows Phone, Microsoft ofrece una nueva interfaz de usuario con su lenguaje de diseño , Metro , integra el sistema operativo con terceros y otros servicios de Microsoft, y establecer los requisitos mínimos de hardware en el que se ejecuta.

2.1.3 IOS de IPHONE.

El IOS de IPhone está conformado por varias capas las mismas que componen un conjunto de servicios.

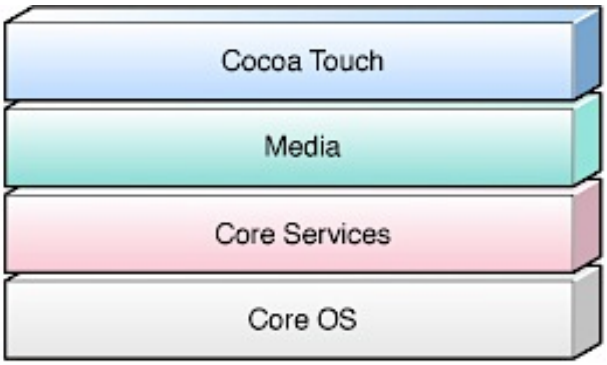


Figura. 2.1. Niveles de Programación de IPhone.

Cada capa de la arquitectura está compuesta por un conjunto de *frameworks. Estos se detallan a continuación:*

- La capa **Core OS** es la base del sistema operativo. Se encarga de realizar la gestión de memoria, el sistema de ficheros, conexión a la red y procesos que interactúan con el hardware.

- **Core Services** nos permite el acceso a los servicios básicos, tales como la agenda, el acceso a la base de datos, preferencias, conexión a servidores y procesamiento de URLs.

- La capa **Media** nos permite la ejecución de tareas multimedia. Entre ella el acceso al Audio, OpenGL, Imágenes y PDF, Animaciones.

- **Cocoa Touch** nos permite acceder al acelerómetro, los eventos y controles táctiles, la jerarquía de vistas, alertas, además gestiona la interacción visual con el usuario.

- **Multitarea**: mientras ejecutamos una aplicación, al pulsar el botón “home” no se cierra sino que pasa a segundo plano, pueden continuar ejecutándose o pueden suspenderse.

- ***Notificaciones Locales*:** completa el sistema de notificaciones push (desde un servidor remoto que nosotros mismos podemos crear).

- **Core Motion**: conjunto de interfaces para acceder a toda la información basada en el movimiento (giroscopio ,motion-based).

- **Data protection**: sistema integrado de encriptación paquetes.

- Soporte para pantalla de alta resolución.

2.1.3.1 Frameworks.

Un **framework** es un conjunto de librerías que nos permite añadir una funcionalidad concreta a nuestra aplicación. Por defecto, cuando creamos un proyecto, tenemos los frameworks esenciales para el funcionamiento básico.

Por ejemplo, para usar bases de datos SQLite usaremos su framework, simplemente con copiar el directorio descargado de su web, aunque en la versión de XCode4 ya viene integrado, también se puede realizar lo mismo con Cocos2D.

Todas ellas permiten obtener un resultado deseado de acuerdo a la especificación de la aplicación y en lo que en ella se vaya a realizar. Básicamente cada herramienta contenida en el paquete de FRAMEWORKS, permiten desarrollar la misma.

- Cocoa Touch: conjunto de frameworks orientados a objetos que permiten el desarrollo de aplicaciones nativas para iOS.

Estas herramientas trabajan en conjunto para obtener el producto final deseado que es el de la creación de una aplicación para Iphone. Cada una de ellas posee un derivativo que consta de varias opciones para su programación.

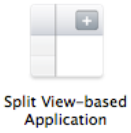
2.1.3.2 Creación de aplicaciones.

En la creación de aplicaciones es necesaria la preparación de un proyecto para esta clase de dispositivo dependiendo de la necesidad y recursos que se vayan a utilizar, se identifican varias herramientas como se detallan a continuación:

- **Cocos** 2d: para crear juegos (Se necesita el framework de Cocos2D).

- Navigation-Based Application **Navigation-based** : presentan la información de forma jerárquica usando múltiples vistas (diseño de pantallas).

- OpenGL Game **Open GL** : igual que Cocos, además de juegos podemos crear cualquier tipo de aplicación que represente imágenes, animaciones o gráficos 3D.

- 

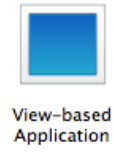
Split View-based : Aplicaciones enfocadas para iPad que muestran más de una vista en la pantalla al mismo tiempo (un control listado y una vista detalle de elementos normalmente).

-

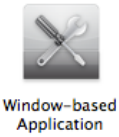
Tab Bar : presentan una interfaz de “radio” que permite al usuario elegir entre varias opciones

-

Utility : aplicaciones que presentan una vista principal y permiten que el usuario acceda a otra vista para realizar personalizaciones básicas

- 

View-based : una vista simple para implementar la interfaz.

- 

Window-based : plantilla universal (cualquier dispositivo con iOS) que sirve de punto de partida con un *delegado* y una ventana. Útil para aplicaciones con jerarquía propia.

Cada una de estas son plantillas para seleccionar el tipo de aplicación que quiere realizar, las mismas que cuentan con diferentes propiedades para que el programador las modifique a su conveniencia a la hora de interactuar con el código, y el usuario final.

2.2 KIT DE DESARROLLO DE SOFTWARE

Para el desarrollo de aplicaciones en los dispositivos móviles, estos cuentan con diferentes plataformas de programación acopladas entre sí para una aplicación de uso final. Se muestra en detalle las herramientas utilizadas para el presente proyecto, para realizar la aplicación en el IPhone.

2.2.1 Herramientas para iPhone.

Para el desarrollo de estos proyectos en el dispositivo Smartphone Iphone, se cuentan con diferentes plataformas de programación, entre ellas están XCODE que es la plataforma de programación oficial de Apple para la creación de aplicaciones móviles. Como se demuestra a continuación el conjunto de herramientas de creación de una aplicación para los dispositivos de Apple:

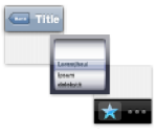
Herramientas.

Xcode

Interface Builder

Frameworks.

Foundation



UIKit

Lenguaje(& Tiempo de ejecución) [textView setStringValue: @“Hello”];

Objeto-C

2.2.1.1 Plataforma XCODE.

Es la plataforma de programación oficial de Apple destinada a la creación de aplicaciones WEB y aplicaciones Nativas para los dispositivos IPHONE. Es un entorno desarrollado integrado, esta se distribuye gratuitamente junto con el sistema operativo del MAC, esta interface trabaja en conjunto con Interface Builder, siendo esta desarrollada en base a NeXT, que es una herramienta gráfica para la creación de interfaces de usuario.

Xcode incluye varios compiladores del proyecto GNU(GCC), que permite la compilación de código C, C++, Objective-C, Objective-C++, Java y AppleScript mediante la aplicación de algunos modelos de programación. Entre las características mas notables de esta plataforma es la capacidad que posee para distribuir el proceso de construcción a partir del código fuente entre varios ordenadores, utilizando el mecanismo Bonjour que es la que permite un servicio de distribución entre máquinas. XCODE proporciona varios compiladores de código, que interpretan tanto la actividad de procesamiento del dispositivo así como también una simulación de la actividad del uso de la memoria del sistema del dispositivo en cuestión.

La versión 4 de este conjunto de herramientas, agrupa tanto la parte de programación en C orientada a objetos, junto con Interface Builder que es un esquema de vista de la pantalla del dispositivo a simular y de igual forma al simulador de dispositivo IOS que es ya una simulación completa virtual de la aplicación dentro de un IPhone. Para la creación de aplicaciones la interfaz demuestra una presentación para un nuevo proyecto, el cual puede estar orientado a los dispositivos móviles IOS o lo que es el sistema operativo del MAC OS X.

A continuación observamos la vista del programa de aplicaciones:

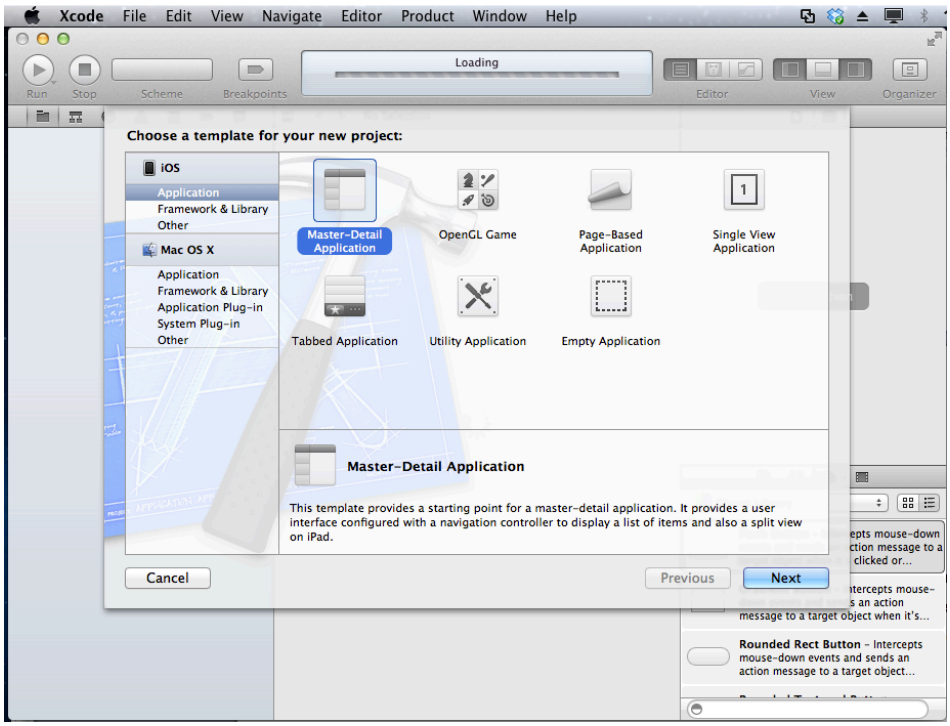


Figura. 2.2. Plataforma XCODE.[16]

En ella como se puede observar existen los mecanismos para realizar la programación de las aplicaciones para IPhone, cada una consta con una descripción corta pero informativa de lo que genera, esto depende básicamente de una plantilla con la que se va a trabajar y a medida que aparezcan mayores requerimientos se procede a añadir efectos y otras plantillas de acceso. Cada una de estas con un conjunto de elementos de control sobre un atributo del dispositivo, es decir se puede definir un conjunto de controladores sobre una vista del dispositivo, o la presencia de una barra con accesos directos que controlen ciertas características que se hayan implementado en el programa, motivo por el cual se dice que son elementos de control.

[16] Imagen extraída de la plataforma de programación de XCODE - APPLE.

Al elegir el prototipo, la plataforma genera los archivos y paquetes necesarios para la compilación, en adición coloca los archivos en diferentes carpetas que señalan a que aspecto del código pertenecen.

Esto permite que el usuario mantenga el control y orden sobre el mismo, lo que al final resulta como una visualización agradable del programa con una parte gráfica con acceso al código. De esta manera es mucho más fácil que otras plataformas el interactuar tanto el software como el hardware, integrándolos a ambos.

Para la creación del aplicativo, se pueden modificar varios parámetros de visualización, se puede cambiar los tipos de orientación permitida por el dispositivo, la versión del software mínima que soportará; en esta las opciones quedan limitadas por la programación tipo hasta esa versión, sin embargo a medida que se utilizan versiones más actuales, la programación puede llegar a una mayor escala.

Cuenta con una división de áreas donde se pueden apreciar el área de carpetas del proyecto, librerías, archivos de cabecera y configuración, donde se pueden declarar clases y objetos al momento de programar.

Cuenta con el área de trabajo que es llamado STORYBOARD, este es el seguimiento de la ubicación de los elementos, que se van colocando las diferentes plantillas a utilizar, así como también los diferentes atributos como botones, barras, contenidos de texto etc. Posee un contenido muy amplio para trabajar, y todas las herramientas del este sistema están a disposición del programador para mayor facilidad en el uso.

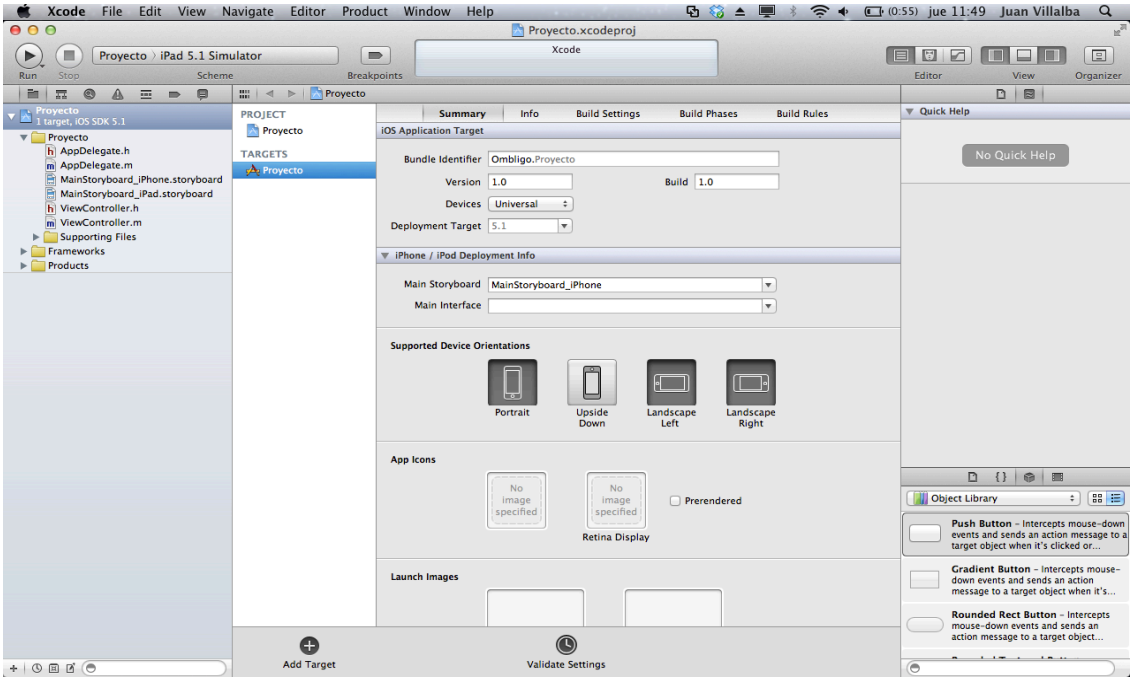


Figura. 2.3. Ventana de Proyecto de XCODE.[17]

Cada uno de estos elementos so fácilmente visualizados en una ventana de contenido en donde se puede interactuar con sus propiedades y capacidades para ejecutar un cambio para la demostración de una actividad en específico. Es una interfaz de fácil manejo para los usuarios, amigable en su disposición y uso de herramientas, una vez familiarizado con el mismo. Se pueden apreciar las posiciones en las que trabajará el dispositivo, no solo para Iphone sino también para Ipad, en él se pueden generar aplicaciones para ambos dispositivos.

2.2.1.2 Plataforma INTERFACE BUILDER.

Es un software de desarrollo de aplicaciones del sistema operativo MACOS X, que forma parte la plataforma XCODE, donde es vinculado en la parte de entorno gráfico, utilizando la secuencia de comandos de XCODE mediante objetos-C con las acciones realizadas en el entorno gráfico de Interface Builder.

[17] Imagen extraída de la plataforma de programación de XCODE - APPLE.

Como resultado de la creación de la interface se crean archivos .nib que es el acrónimo de *NeXT Interface Builder y* mas reciente como archivos .xib. El acrónimo utilizado solo es para identificar el tipo de proyecto sobre la plataforma que se implementa y consta únicamente como característica del compañía que lo diseñó ya que desciende del desarrollo de software de la compañía NeXSTEP. Esta interfaz que forma parte del conjunto de herramientas de XCODE permite visualizar la pantalla de los dispositivos de Apple para ir ubicando los elementos y vistas que se van a utilizar. Esta permite la interactividad con procesos y métodos que sea llamados por estas y que hayan sido codificadas por el programador.

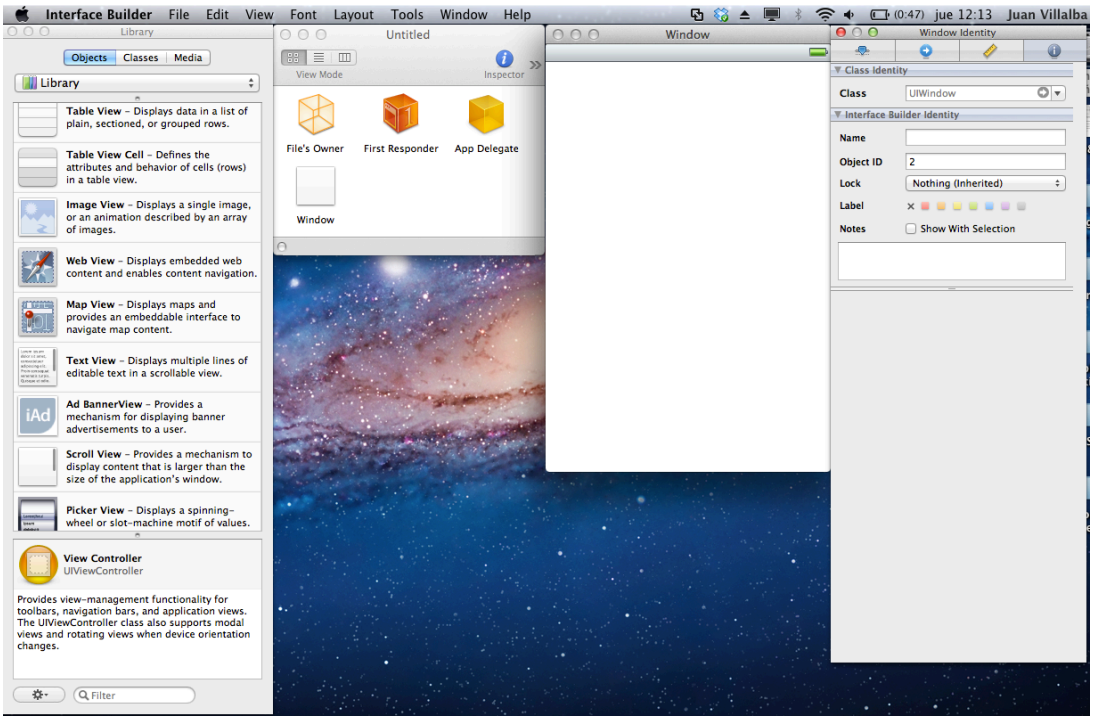


Figura. 2.4. Interfaz de Interface Builder.[18]

En ella se observan de igual forma que en XCODE, las librerías que esta utiliza, elementos, y una plantilla con la vista de la pantalla de un Iphone que es donde se ubican los elementos para interactuar con los métodos.

[18] Imagen extraída de la plataforma de programación de XCODE - APPLE.

Se dispone también de las propiedades de cada elemento para modificar parámetros de presentación y estética sobre los mismos. Mediante esta interfaz se puede interactuar con XCODE donde a su vez se declaran las clases y métodos que utilizarán los objetos de INTERFACEBUILDER para generar los efectos de la aplicación, permite colocar enlaces de las clases determinadas por el usuario con sus objetos.

2.2.1.3 IPhone simulador.

Este es el simulador oficial del dispositivo IPhone el cual enlaza los métodos y objetos generados por XCODE e INTERFACE_BUILDER en una aplicación implementada en el dispositivo virtual. En la imagen se aprecia un ejemplo de dicha simulación:

Figura. 2.5. Simulador de IPHONE.[19]

[19] Imagen extraída de la plataforma de programación de XCODE - APPLE.

Cabe recalcar que la simulación ejecutada funciona de acuerdo a las configuraciones realizadas en XCODE y son simuladas para los requerimientos como lo son:

- Versión de sistema operativo del IOS.
- Dispositivo Utilizado, IPhone, Ipod, Ipad.

Es necesario realizar las pruebas en el simulador antes de poder instalarlas en el dispositivo en sí, ya que en la simulación es posible que este funcione, sin embargo la prueba final se la realiza en el dispositivo. Para lo cual es necesario un certificado de desarrollador otorgado por la empresa Apple para los programadores que la deseen, incluyendo un costo anual como licencia para realizar pruebas de las aplicaciones en los dispositivos.

El simulador trabaja una vez que el código generado en XCODE, se acopla con la interfaz gráfica generada en Interface Builder, de manera que compilados ambos archivos, se unifican para dar continuación la ejecución de la aplicación final en el simulador. En el se pueden realizar las pruebas deseadas. Es un simulador completo que posee parámetros de conexión hacia la red, junto con capacidad de interactuar con contenidos web, por lo tanto es casi lo más cercano al dispositivo físico Iphone con el que posteriormente se pueden realizar las pruebas a través de una aprobación de certificados anteriormente mencionados.

2.3 PROGRAMACIÓN EN OBJECT C.

La programación llamada Object C está orientada a la programación de objetos tal como en cualquier plataforma de compilación de código. Contiene clases y librerías del código en C. Es un pequeño set de extensiones de ANSI C, y los añadidos a C están basados en Smalltalk, uno de los primeros lenguajes orientados a objetos, por lo cual está diseñado para dotar a C de toda la potencia de la orientación a objetos, que se define como una pequeña pero bastante avanzada grupo de extensiones del estándar del lenguaje ANSI C.

Sus aportes al lenguaje C son comúnmente basados en Smalltalk, que es una de los primeros lenguajes de programación orientado a objetos. El Object C es diseñado para dar al lenguaje C una alta capacidad de programación orientada a objetos, realizando esto en una simple pero robusta manera.[20]

Este conjunto de herramientas en el desarrollo de aplicaciones para IPHONE es conocida como Cocoa. La programación en la plataforma está dada normalmente por la creación de archivos de cabecera para generar Clases para la programación a objetos y la utilización de métodos, al igual que son llamados en C. Las clases son creadas en archivos de cabecera en código C, con una sintaxis muy similar, pero debido que esta es una plataforma diferente, el uso de los comandos difiere de otros, sin embargo la definición y estructura de programación es parecida a la de un JAVA.

2.3.1 Programación a Objetos.

Un objeto asocia datos con ciertas operaciones en particular que pueden utilizarse o afectar a los mismos. En la programación en Object-C este tipo de operaciones son llamadas métodos los cuales tienen una incidencia en variables que hayan sido instanciadas dentro de él.

En concreto, un objeto es una estructura formada por la instancia de variables y el grupo de procedimientos o métodos que estén contenidos en una unidad de programación. Los objetos manejados son utilizados por la plataforma XCODE, como características de estructura cada variable instanciada estará dentro del objeto y para obtener un cambio en el estado del objeto únicamente se tiene acceso a el a través de los métodos. Es decir para obtener información de un objeto ya sea para subclases u otros objetos hagan uso de ella, solo podrán realizarlo a través de los métodos. De manera que un objeto solo puede ver los métodos que hayan sido realizados para él, por lo cual no podría ejecutar otros métodos que de otro tipo de objetos.

[20] Obtenido de http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html

Al igual que en las funciones de lenguaje C sus variables locales son ocultadas del resto de código del programa al igual que un objeto lo hace a sus variables y métodos para sí. Para definir cada objeto, se hace posible a través de un identificador para cada tipo de dato que sea de tipo: ID. Este identificador es general para cualquier tipo de objeto sin importar de que clase sea y puede ser utilizado para instanciar objetos o variables para una clase cualquiera, para esto se define:

id ElObjeto;

El id del objeto por lo general es de tipo 'int' entero al igual que en C. El cual corresponde a un puntero a los datos de la estructura del objeto.

Tipo estructura ***clase_objeto *puntero_clase;***

En esta manera se declarará el objeto que mantenga un puntero hacia sí para definirlo, y aplicarlo al método a utilizar. De esta manera se podrá estructurar el esqueleto del programa con las clases junto con sus variables y métodos para que sean utilizadas por el objeto definido.

2.3.2 Declaración de variables.

La declaración de variables en XCODE se declaran similar al lenguaje C con ciertas propiedades que los diferencian. Las clases están definidas de la siguiente manera:

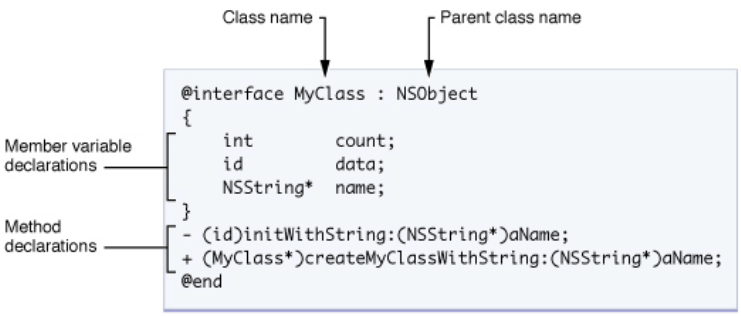


Figura. 2.6. Estructura de una Clase en Object C.

Como se puede apreciar en la imagen se observa como se declara una clase, en definición con el objeto, junto con su conjunto de variables que son instanciadas con el ID que las caracteriza. Y, que una vez definida la clase, el objeto, y los componentes de la misma es necesario declarar los métodos que en sí toman el control de las variables del objeto para generar un cambio en su presentación. Cada una de las variables utilizadas, corroboran un papel fundamental al momento de declarar los métodos, pues estas son parte de cada característica del objeto. Sus formatos y usos van de acuerdo a necesidad de un resultado deseado. Para definirlas existen:

- Caracteres: Se utiliza la etiqueta char que las define como un conjunto de caracteres.
- Enteros: Se utiliza las etiquetas short, int, long.
- Números en coma decimal: float, double, long double
- Booleanos: bool.
- Sin retorno: void.

2.3.3 Interfaces.

Al crear un proyecto en XCODE se crean automáticamente archivos que si se observan en la figura 2.7 podemos apreciar una clase main.m que es la primera en primera instancia se ejecutará. Adicional se crean automáticamente dos archivos de tipo Iphone.xib los mismos que están asociados a los archivos Appdelegate que es el encargado de verificar cuando se hayan registrado cambios en el estado de la aplicación. Esto lo verifica cuando la aplicación se a cargado completamente, para de esta proceder a cargar la interfaz del usuario ya sea en el equipo o simulador. La interfaz de programación dentro de XCODE está dada una plantilla en la cual se van colocando todos los elementos gráficos del aplicativo; el objetivo de este es poder declarar las interfaces para dicha vista mediante la declaración de una clase para lo cual se lo realiza en el archivo de cabecera .h, ya que en este solo contiene declaraciones y no implementaciones.

La interfaz contiene los parámetros de configuración de la presentación del aplicativo de tal manera que para esto utiliza diferentes archivos de configuración, uno para declarar elementos y objetos, y métodos a crear, y otro para poder inicializarlos junto con sus variables e inicializarlas en los métodos para que estos se vean reflejados en un resultado esperado al momento de ejecutarlos.

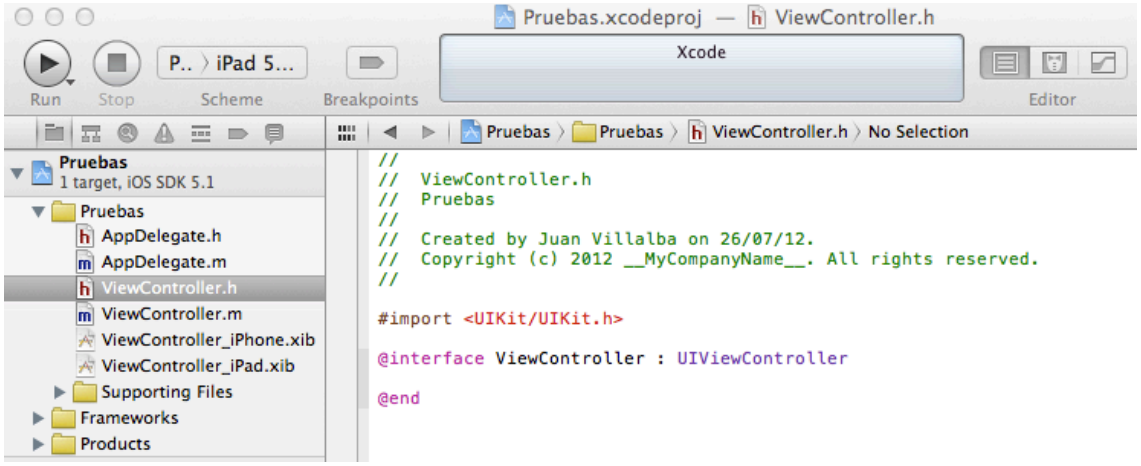


Figura. 2.7. Interfaz de la clase.

Para poder declarar dicha sintaxis de la interfaz de la clase principal en lo que se refiere a Object C, se utiliza las sentencias que se muestran a continuación, la misma que puede hacer referencia a la figura 2.7 para visualizar a ejemplo de la declaración de la interfaz.

@interface (Nombre): NSObject
---- Declaración de métodos----
@end

Adicional algo particular de esto es que proporciona el manejo de la memoria y la previa inicialización del Framework; es decir que carga los objetos y plantillas del proyecto para su modificación. Si se observa la figura 2.7 hace referencia a una clase típica con las propiedades de lo que en este caso utilizará como base un UIViewController que es el controlador del aplicativo gráfico.

La sentencia para declarar es la siguiente:

```
@interface NombreViewController : UIViewController { }
```

En él se definirán los protocolos formales que de forma particular implementara la clase de la siguiente manera:

```
@interface NombreClase <NombreProtocolo1, ... , NombreProtocoloN> {
---Declaración de atributos----
}
--Declaración de métodos y propiedades—
@end
```

En esta segunda instancia se observa la interface con la declaración de atributos y una vez ingresada se procede a declarar los métodos que se implementarán, así como también sus propiedades. Vista esta segunda instancia la podemos verificar en el siguiente gráfico.

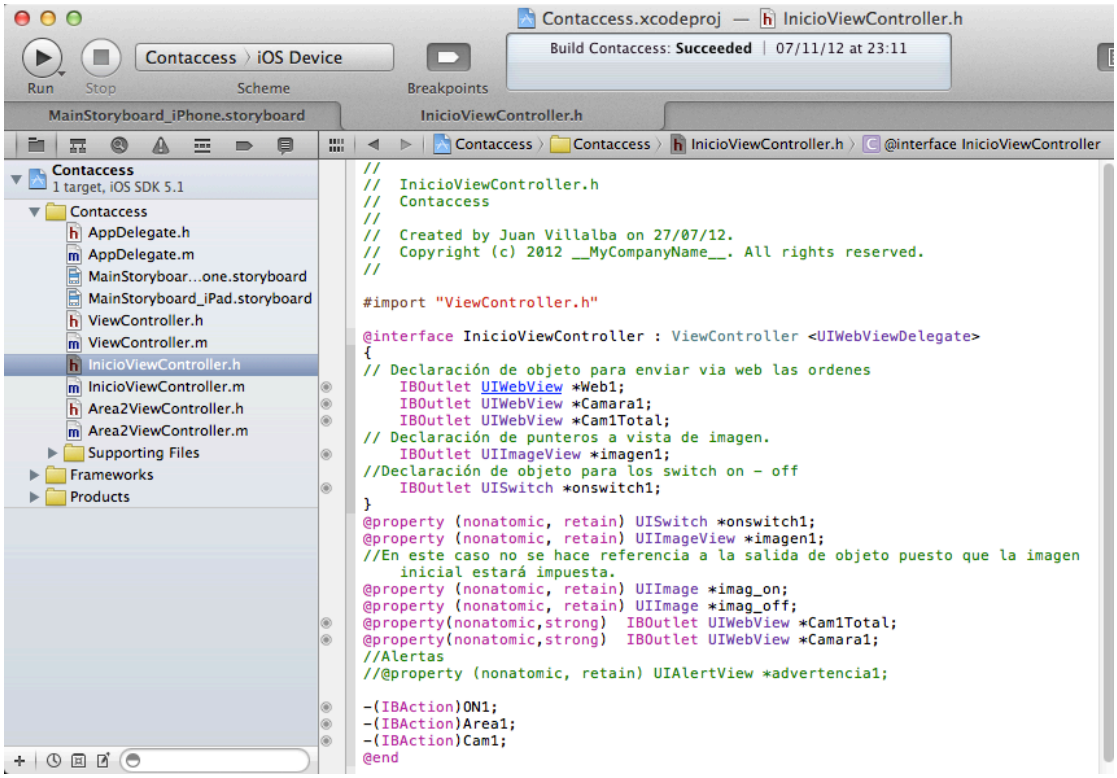


Figura. 2.8. Interfaz con Protocolos Formales.

Adicional como se observa en el gráfico anterior, podemos verificar que estas declaraciones se las realiza en el archivo de cabecera .h que es el archivo donde se declaran la clase y sus objetos. Sin embargo esta es la primera parte de la declaración de la clase principal del programa, para lo cual se hace indispensable otro archivo de configuración en el cual se registran los objetos y los métodos del aplicativo. Para esto se utiliza el archivo de implementación que está representado por el formato .m.

2.3.4 Implementación.

Una vez realizada la interfaz en primera instancia para la declaración de las variables y métodos, es necesario inicializarlos. Para esto se programa el archivo de formato .m que hace referencia a los métodos declarados en el archivo de cabecera. Para lograr que las aplicaciones funcionen de manera adecuada, es necesario la importación de librerías y creación de clases que permiten utilizar métodos e instancias de variables.

Para lo cual se debe crear ficheros que en este caso dependerá del tipo de controlador que se vaya a utilizar, estos controladores. Cabe recalcar que para no existan errores en la compilación cada controlador hace referencia a otro para mantener la relación, esto lo realiza mediante la importación de la interfaz que se ha declarado. Para conseguir esta solo hay que ejecutar #import “El controlador.h“. Así se consiguen que las variables, métodos y referencias se encuentren disponibles. Para declarar la implementación se utiliza la sentencia:

```
#import “CualquierClase.h”
@implementation CualquierClase
@end
```

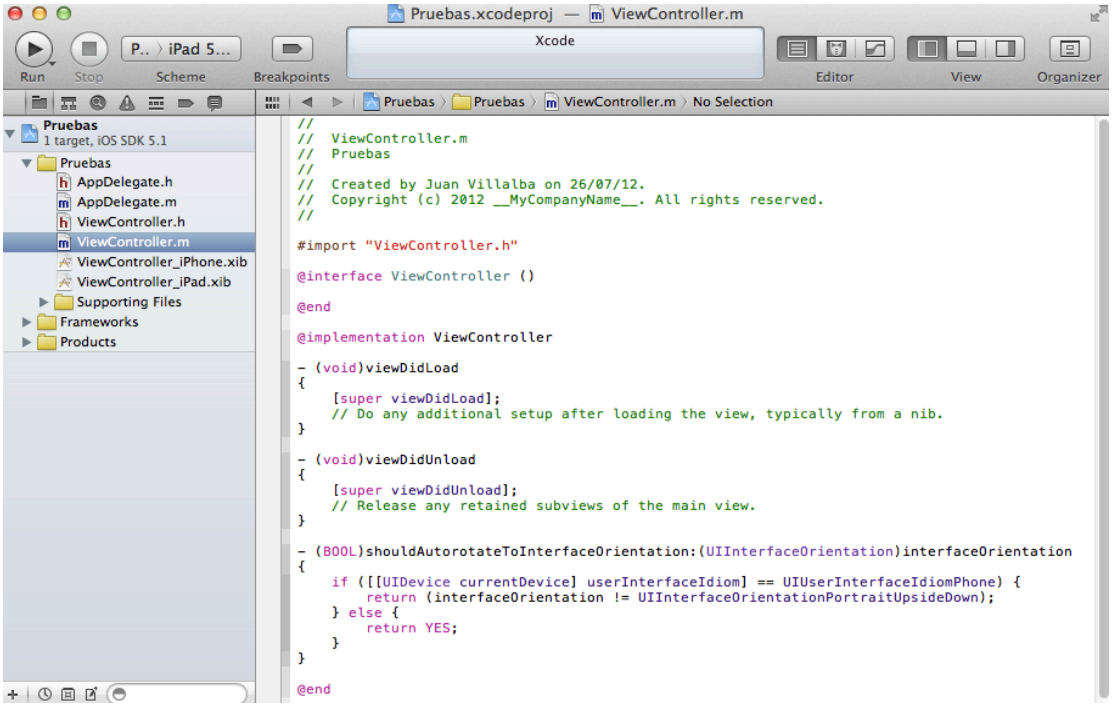


Figura. 2.9. Implementación de la interfaz.

Como se observa en la imagen anterior el archivo de implementación siempre hace referencia al archivo de cabecera .h, de esta manera los métodos declarados en el archivo de implementación se ejecutarán sin error alguno. Puesto que si estos métodos no son declarados inicialmente en el archivo de cabecera, no se puede ejecutar en la interfaz pues cabe recalcar que los únicos métodos que podrían ser inicializados sin declaración en el archivo de cabecera .h son los de tipo void que no retornan ningún valor o referencia a una variable de la clase.

2.3.5 Objetos de XCODE.

Dentro de la plataforma XCODE existen diferentes objetos que se pueden utilizar en la plantilla creada, ya sea para colocar botones, vistas web, imágenes, etc. Dentro estos mencionaremos los más utilizados y los que mas adelante se irán colocando en el proyecto.

Entre los más destacados resaltan los siguientes elementos:

2.3.5.1 UIWebView.[21]

Figura. 2.10. Elemento UIWebView.

Se utiliza la clase UIWebView para incorporar contenido Web Embebido dentro de la aplicación del dispositivo móvil. Para poder realizarlo simplemente se debe crear un objeto UIWebView, para lo cual es necesario unirlo a la ventana, y enviar en código la solicitud de carga de contenido Web.

De igual manera se puede utilizar esta clase para dirigirse a una página web anterior y también dirigirse hacia adelante en el historial de páginas Web, incluso se puede setear algunas propiedades del contenido de forma programable. Se puede utilizar el comando loadRequest: que corresponde al método para empezar a cargar contenido Web en la vista Web, y el método de stopLoading para proceder a detener la carga. De manera que la propiedad para loading es el de identificar si la vista web está en proceso de carga.

[21] Obtenido de http://developer.apple.com/library/ios/#documentation/uikit/reference/UIWebView_Class/Reference/Reference.html%23//apple_ref/occ/instm/UIWebView/

Si se procede a permitir que el usuario se dirija tanto hacia atrás como hacia adelante a través del historial de la Web, entonces se pueden utilizar los métodos de goBack (Regresar) y goForward (Adelante)como acciones para botones del aplicativo. Adicional se pueden utilizar propiedades de canGoBack y canGoForward para deshabilitar los botones cuando el usuario no se puede mover en una dirección. Por default una vista web puede convertir números de teléfono que aparecen en contenido Web en números de teléfono. Cuando el link del teléfono es tocado inmediatamente ejecuta y marca el número. Se puede utilizar la propiedad de detectsPhoneNumbers en opción NO para deshabilitar este comportamiento.

También se pueden identificar propiedades tales como scalesPageToFit para programar el escalamiento de tamaño del contenido Web la primera vez que es desplegada dentro de la vista web. Aunque el usuario puede cambiar la escala usando gestos con los dedos. Utilizando la propiedad delegate con un objeto conformante del protocolo UIWebViewDelegate si se desea rastrear la carga del contenido Web.

Como referencia no se debe colocar un objeto de tipo UIWebView o UITableView dentro de un objeto UIScrollView. Si se lo realiza de la manera anteriormente mencionada, se puede esperar un comportamiento inadecuado del aplicativo a resultado de los eventos producidos al tocar la pantalla puesto que se pueden mezclar los métodos de cada objeto y sean manejados erróneamente. El contenido web de la vista UIWebView no solo se puede utilizar para desplegar contenido Web sino también otros contenidos web de diferente tipo, ya sea PHP, JAVAscript mas no como máquina interna de ejecución JAVA puesto que la plataforma de Apple posee sus propias librerías de código.

2.3.5.2 Alertviews.[22]

Figura. 2.11. Elemento UIAlertView.

El protocolo UIAlertViewDelegate define los métodos delegados de una alerta de un objeto UIalertView debe implementar. El delegado implementa el botón de acción y algunos otros comportamientos personalizados. Algunos de los métodos definidos en este protocolo son opcionales. Si se añade botones de diseño propio o se personaliza el comportamiento de la vista de la alerta, se debe implementar un delegado acorde a este protocolo para poder sostener los mensajes delegados correspondientes. Utilizar la propiedad de delegate de la vista de una alerta permite especificar a uno de los objetos de la aplicación como delegada.

Si se procede a añadir botones con diseño propi en una vista de la alerta, el delegado debe implementar el mensaje alertView:clickedButtonAtIndex: para responder a aquellos botones que se haya clickeado, en otras palabras los botones personalizados no hacen nada. La vista de alerta es ignorada automáticamente después de que el método delegado alertView:clickedButtonAtIndex: es invocada. De manera opcional se puede implementar un método de cancelación de la vista de alerta mediante alertViewCancel que toma posesión de la acción cuando el sistema cancela la vista de alerta.

[22] Obtenido de http://developer.apple.com/library/ios/#documentation/uikit/reference/UIAlertView_Class/UIAlertView/UIAlertView.html#//apple_ref/occ/instp/UIAlertView/delegate

Si el delegado no implementa este método, el comportamiento por default es el de simular el click del usuario sobre el botón de cancelación y cierre de la vista. De igual manera se puede modificar de manera el comportamiento de la presentación y cierre de las señales de alerta utilizando un método de comportamiento personalizado con *Customizing Behavior*.

2.3.5.3 Seguimientos.

Debido que en la plataforma de XCODE en lo que se refiere a la ejecución de presentación de varias ventanas en la aplicación, se hace imprescindible el uso de una secuencia de ejecución de ventanas conocida como SEGUE esta es una opción que permite de manera gráfica el seguimiento a la siguiente secuela del aplicativo, visualizado previamente en el Storyboard que es en el cual se denota paso a paso la secuencia indicada.En el presente gráfico observamos un modelo de StoryBoard que no es mas sino donde se observa la secuencia de ejecución de ventanas, en cada una de ellas se tiene diferentes funcionalidades que a su vez hacen referencia a otras ventanas para que interactúen de manera que el programador se oriente.

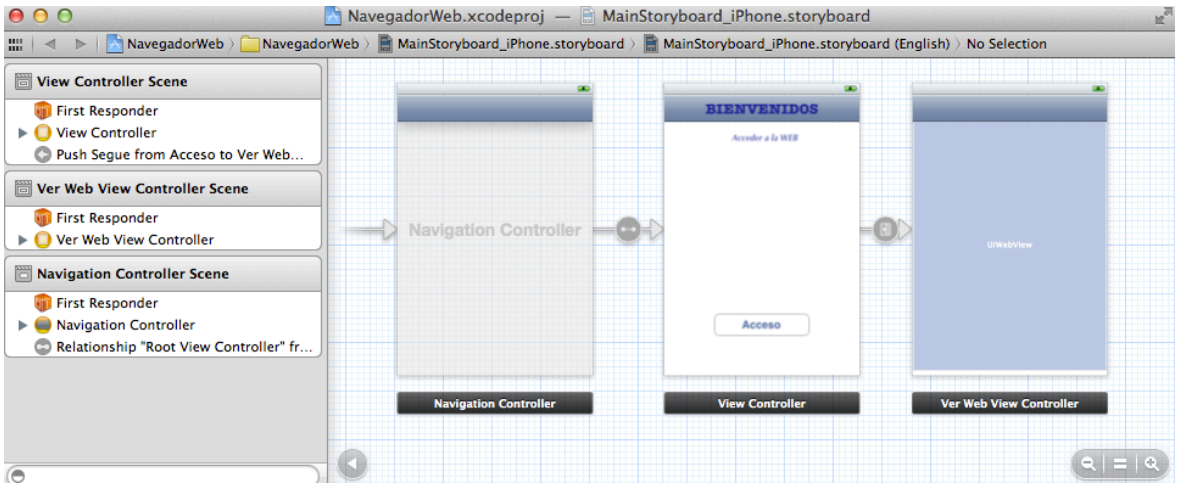


Figura. 2.12. StoryBoard.

Si nos fijamos en el gráfico cada una de las ventanas corresponde a un controlador fijo en la primera se observa un controlador de navegación que no es mas que la referencia para indicar navegación de una ventana a otra. Después se observa el controlador de vista principal que es el ingreso al aplicativo en sí, y de este hacia el controlador de la vista Web. A simple vista con este gráfico ya se puede tener una idea de lo que realiza el programa, que en este caso lo único que realiza es el acceso a una página web desde la orden generada en el botón de ingreso del controlador de vista.

Es por esta razón que mediante esta manera de programación, el programador puede mantener un orden de ejecución ventana a ventana lo que le otorga el completo control de cómo se van a ejecutar los eventos y en la manera que este requiera.

2.3.6 Controladores.

Para la declaración de Variables existen diferentes formatos aceptados por la plataforma XCODE dependiendo de la aplicación y el evento en que se los vaya a utilizar. De tal manera que el uso de la memoria sea la adecuada, y las variables sean utilizadas de manera ya sea temporal o de forma indefinida durante su ejecución.

Para la creación de aplicaciones, en su mayoría se utilizan diferentes tipos de controladores, es decir, si en una aplicación el punto de concentración es la vista del aplicativo se utiliza un controlador, el mismo que corresponde a una clase creada por el programador, en ella se instancian las variables y los métodos que se van a utilizar por los diferentes elementos y atributos que se despliegan en la pantalla del IPhone.

Para describir de mejor manera, se hace referencia a los controladores más usados en la plataforma XCODE; estos son creados a partir del tipo de controlador y es definido por dos archivos de cabecera en formatos .h y .m.

En el archivo de formato .h se define la interfaz en la cual se instancia el objeto sobre el cual se definirán los procesos y eventos dirigidos que se ejecutarán durante la implementación de los métodos.

Para observar de mejor manera se tiene en la siguiente imagen un ejemplo de la ventana proporcionada por XCODE.

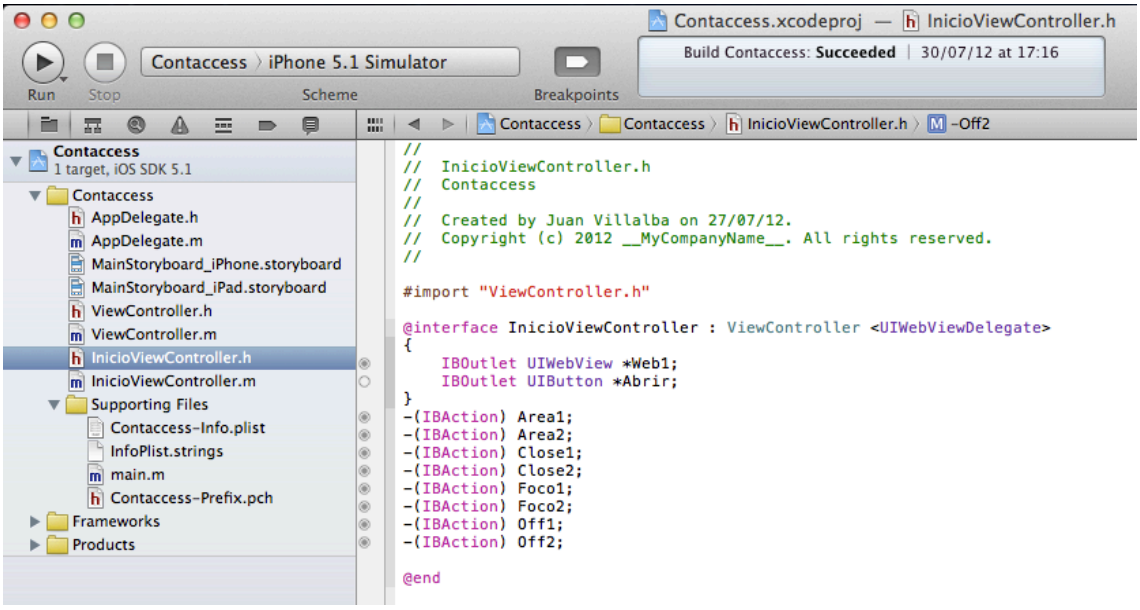


Figura. 2.13. Archivo de Interfaz .H

El archivo de formato .m define las instancias de los métodos, en él se definen las variables de forma que correspondan a los eventos de los objetos sobre los cuales se hará la guía de la interfaz grafica. Una vez que se definan las propiedades de los punteros que apuntan hacia los objetos, y se han definido los métodos, hace imprescindible a los elementos gráficos como botones, áreas de visión, web embebidos, hacer partícipes de la clase correspondiente para su manejo y operación. En la imagen se puede observar en detalle cada instancia.

Como se trata de una página web, si se desea a diferentes botones asignar la misma clase, se lo puede hacer, ya que la instancia ingresada por la interfaz corresponde a un botón, el mismo que puede ejecutar cualquiera de las funciones declaradas.

Figura. 2.14. Pantalla de un UIWebView. (Vista Web)

Las categorías son muy útiles cuando se quiere agregar funcionalidad a una clase, pero no se desea reescribirla o no se cuenta con el código fuente de la clase como cuando esta se encuentra en una biblioteca. Como se observa en el archivo, la nueva clase corresponde a una interfaz de un controlador de vista de aplicación, en la cual se destacan los objetos, que en este caso la clase principal tomará con objeto a controlar es de una VistaWEB la misma que está como delegada.

En esta se utilizarán como referencia los objetos WebView y un Botón para esta clase, la misma que llevará consigo las acciones realizadas por el botón ante una página web, varias de estas funciones pueden generar diferente tipos de ejecución.

2.3.7 Propiedades.

@Property: Esta propiedad permite que los métodos de acceso se generen automáticamente para los atributos de la clase, se pueden utilizar métodos de **setter** y **getter** a manera de sobrecarga siempre que no se utilice las especificaciones de una propiedad por defecto. Para poder diferenciar cual es la más conveniente primero se debe verificar la gestión de memoria.

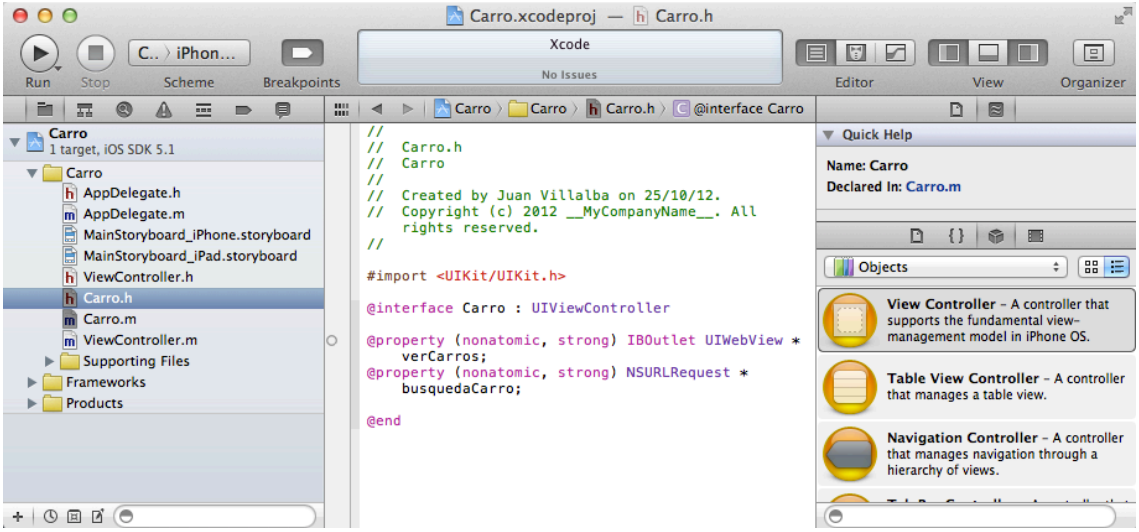


Figura. 2.15. Archivo de Implementación con Atributos.

El método de **setter** se añade por defecto a la propiedad definida como setNombre_variable (set+UCase(nombre_variable)). Para declarar algunas variables se utilizan propiedades, es decir que existen formas de declararlas ya se con un formato como el que se detalla a continuación:

@property tipo nombre; @property (atributos) tipo nombre;

"En la sentencia mostrada se puede notar los atributos, estos denotan a que se quiere hacer referencia de variable de proceso, es decir se puede escoger que tipo de característica tendrá en el proceso.

De tal manera que puede ser una variable de un valor fijo sin ningún tipo de cambio sobre la misma, así como también puede corresponder a una donde tenga proceso de lectura y escritura, de manera que influirá en ejecución del programa en sí. Dichos atributos pueden ser declarados de manera opcional, para lo cual a continuación se pueden denotar algunos que actúan por defecto:" [23]

Métodos para acceder:

- *getter=nombre* → Por defecto, si no se indica nada, el getter tendrá el mismo nombre que el identificador de la propiedad.
- *setter=nombre* → Por defecto, si no se indica nada, el setter tendrá como nombre al identificador de la propiedad precedido de la palabra set. Por ejemplo, si nuestra propiedad se llama Nombre, el setter por defecto sería setNombre.

Escritura o solo lectura:

- readwrite → Es una de las opciones por defecto en la definición de las propiedades, por lo que no hace falta ponerlo, y lo que permite es que se pueda leer y escribir su contenido.

- readonly → Esta opción indica que la propiedad es solo de lectura y no es modificable.

Tipo de asignación:

Esta es una propiedad que utiliza las siguientes opciones que se describen a continuación, por lo que es algo que se suele indicar normalmente al definir las propiedades, ya que por lo general se trata es de trabajar con objetos la mayor parte del tiempo.

[23] https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Chapters/ocProperties.html

- assign → Se realiza la asignación sin invocar a retain. Este tipo se suele utilizar para propiedades de tipos que no son objetos como int.

- retain → Se realiza la asignación invocando a retain para mantener el origen de los atributos de dichas variables.

Este tipo de asignaciones que se detallan a continuación, solo se puede utilizar para propiedades de tipos que son objetos de Objective-C.

- copy → Se le asigna a la propiedad una copia del objeto pasado al setter.

Atomicidad:

- atomic → Es la opción por defecto y lo que hace es bloquear la asignación para que ningún hilo que esté ejecutándose al mismo tiempo, pero no haya sido el que ha invocado al setter, no pueda modificar la propiedad mientras lo está haciendo otro hilo.

- nonatomic → En esta opción no hay bloqueo, por lo que cualquier hilo puede modificar el valor sin tener que esperar a que termine el otro. Esto obviamente puede ser problemático, pero en aplicaciones que solo Tienen un hilo se utiliza esta opción para ahorrar operaciones inútiles que no servirían de nada.

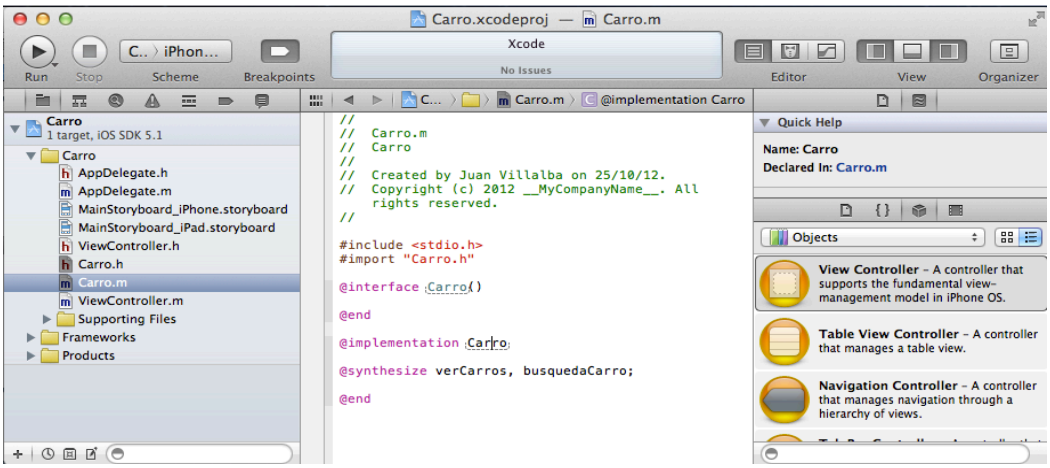


Figura. 2.16. Archivo de Implementación de la clase.

Para poder hacer uso de las propiedades de estas variables en primera instancia, estas deben ser obligadamente inicializadas mediante el uso de **@synthesize** en el archivo **Carros.m** (Que esta corresponde al archivo de implementación de la clase.). Este archivo es el que genera de manera automática tanto **setters** como **getters** en la implementación. Una vez que que se han generado dichos parámetros, se puede especificar en el campo de sintetización el nombre local del acceso; a manera de ejemplo se podría identificar de la siguiente manera:

@synthesize variable u objeto = nombre local

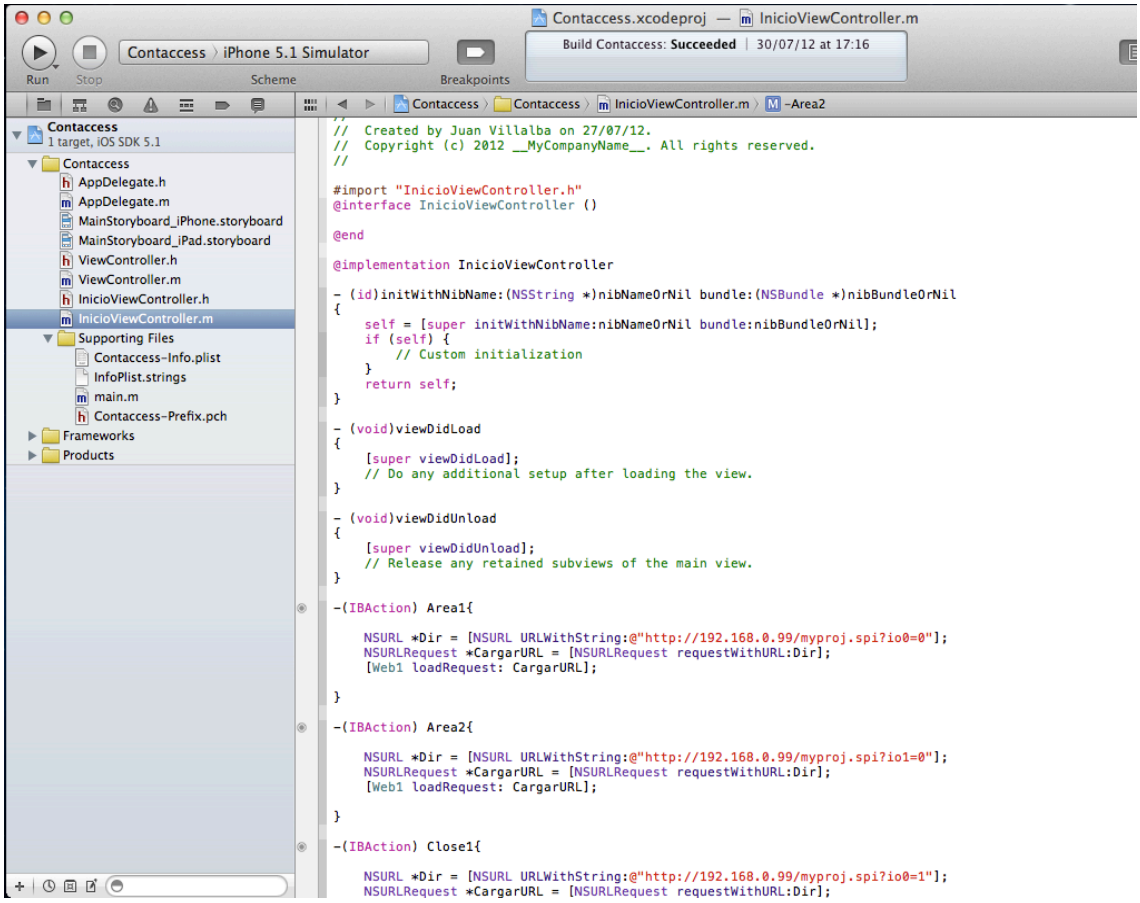


Figura. 2.17. Archivo de implementación .M

En el presente archivo se observa la implementación de la clase, es decir el proceso de operaciones de cada método ingresado en el cual se destaca el acceso a una página Web, para cada acción. En este caso el método aplicado es un método de tipo VOID que no retorna ningún valor o parámetro para otra ejecución.

En el comienzo de la aplicación se instancia el efecto de generación de la vista, o sub_vista en su defecto, que para este caso no corresponde el uso de las mismas pero que queda instanciada por defecto.

Sin embargo si se observan los siguientes métodos, se observa que corresponden a métodos de acción en eventos ejecutados sobre los botones de la aplicación que se hayan designado a ese proceso que en primera instancia están declarados todos en el fichero.

Posteriormente se definirán sobre cada uno de estos objetos. Para definir variables que vayan a ser contenidas en los métodos, que corresponden a punteros de la clase declarada, se los debe declarar como sintetizados en la implementación. Definidas las propiedades en nuestra clase viene la implementación y para ello se utiliza como se mencionó anteriormente:

@synthesize vercarro; @synthesize Tipo= busquedaCarro;

Con @synthesize lo que le estamos indicando al compilador es que si el programador no ha creado los métodos a creación propia, que los cree automáticamente.

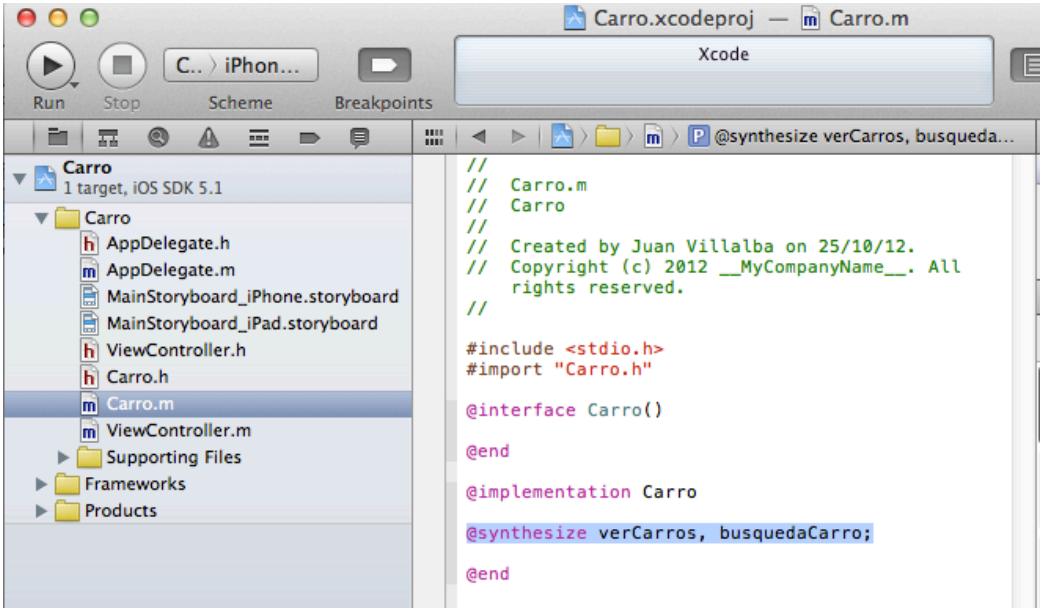


Figura. 2.18. Instancia de variables de implementación.

Existe la opción de sobre-escribir propiedades definidas en clases madres, protocolos (que son el equivalente a las interfaces de Java o C#) o de las categorías, pero tan solo se puede cambiar opción sobre si es de lectura/escritura o solo de lectura no puede ser alterada en la sobre-escritura, el resto tendrán que ser iguales a la definición original.

2.3.8 Instancia de métodos.

Para que se puedan realizar cambios o resultados en parámetros de un objeto de una clase, se utiliza la instancia de métodos que van a la mano con la delegación de una operación requerida por el objeto, y en ocasiones utilizar otro objeto para obtener información para ejecutar otro proceso de un método en particular. Los protocolos declaran métodos que pueden ser implementados por cualquier tipo de clase, y son útiles para cualquiera de las siguientes tres situaciones:

- Para declarar métodos que otros objetos los implementen.
- Para declarar una interface de un objeto que a su vez mantiene su clase.
- Para mantener similaridad entre otras clases que no tienen ningún tipo de relación jerárquica.

Para su declaración simplemente utiliza una lista de declaraciones para el método, que no va atado a ninguna definición de una clase. Esto se puede ver reflejado en un evento de tipo ‘void’ que no realiza ningún tipo de retorno de un valor o resultado para utilizarlo en otro método.

Sino que se ejecuta en un momento fuera de lugar independiente de los procesos que se ejecuten. En la programación de Objetc-C que es la utilizada por la plataforma XCODE no requiere ni es del todo necesario utilizar este tipo de protocolos, sin embargo son opcionales y en algunos casos serán útiles.

Los formatos para definir los métodos al igual que en JAVA se definen 2 tipos entre los que se definen:

Métodos de instancia: Son métodos que solo se pueden invocar desde un objeto que ya ha sido instanciado previamente.

Métodos de clase: que son los que se pueden invocar directamente desde la clase.

En la programación de Object C para definir el tipo de método siempre se lo define con un símbolo al comienzo de la cabecera, de tal manera que a través de este se puede distinguir si es un método de clase o método de instancia.

Para identificarlo de mejor se puede describir de la siguiente manera:

Si se añade (-) al inicio de la instancia del método:

public static void Factorial(int x); En JAVA
+(void) Factorial(int x); En XCODE

Si se añade (+) al inicio de la instancia del método:

public void Factorial(int x); En JAVA
-(void) Factorial(int x); En XCODE

Puesto alloc que devuelve un espacio de memoria reservado para un objeto y a continuación se invoca el método init de ese objeto que lo que hace es inicializar los atributos del objeto pepe al que finalmente se le asignará. Para que se pueda instanciar desde una clase de forma normal se utiliza el parámetro alloc para que al asignar el espacio de memoria al objeto pueda devolverlo en un tipo de variable de la clase.

*Clase *nombre = [clase alloc];*

Adicional se tiene en consideración otro particular que tiene relación con un dealloc, ya que por ciertas circunstancias las propiedades de estos objetos que han sido sintetizados, no permiten invocar un reléase desde un dealloc.

Puesto que es la forma normal en que se trabaja cuando no se utilizan propiedades, para lo cual para poder solventar dicho inconveniente es simplemente asignar un valor de *nil* a la propiedad de dos formas diferentes como se detalla a continuación.

Tipo 1: [self setNombre:nil];
Tipo 2: self.nombre = nil;

De esta manera se puede utilizar el operador punto para poder acceder a los atributos de la clase directamente en vez de utilizar los métodos de **setter** o **getter**, haciendo que el programa proceda con un lenguaje puro y natural.

Donde podemos apreciar este tipo de programación es en su mayoría en la declaración de alertas que se puede verificar en el siguiente código:

Tabla 2.4. Archivos de XCODE.

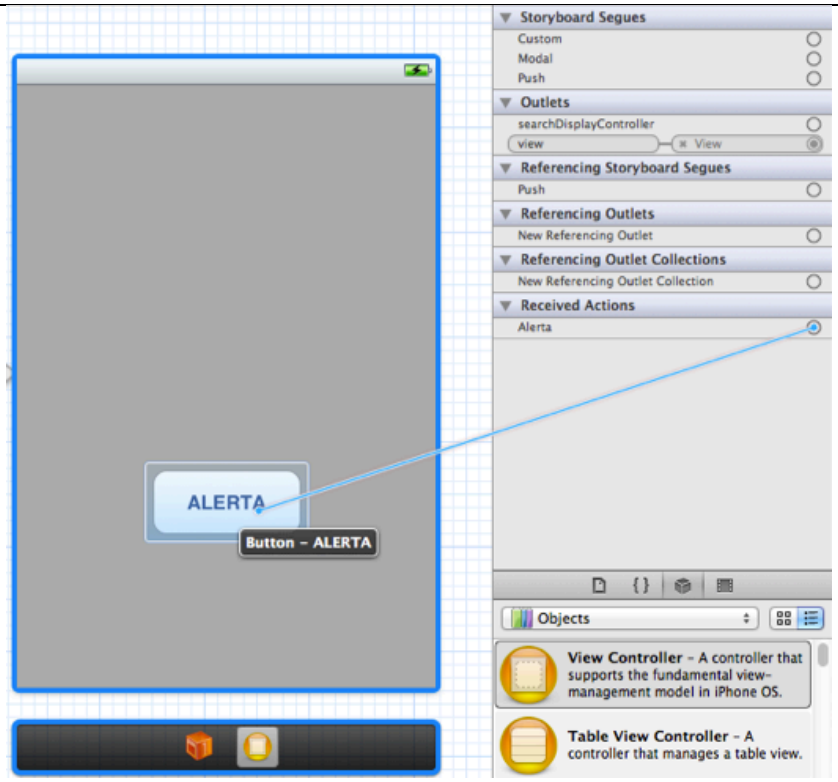
ARCHIVO DE CABECERA .H

`@interface ViewController : UIViewController`
 	
`-(IBAction)Alerta;`
 	
`@end`

ARCHIVO DE IMPLEMENTACIÓN .M

`-(IBAction) Alerta{`
 	
`UIAlertView *alerta1 = [[UIAlertView alloc]`
`initWithTitle:@"BIENVENIDOS"`
`message:@"Esta es una ventana de`
`Alerta"`
`delegate:self`
`cancelButtonTitle:@"Aceptar"`
`otherButtonTitles:nil];`
`[alerta1 show];`
`}`
`@end`

ASIGNACIÓN DEL MÉTODO



RESULTADO

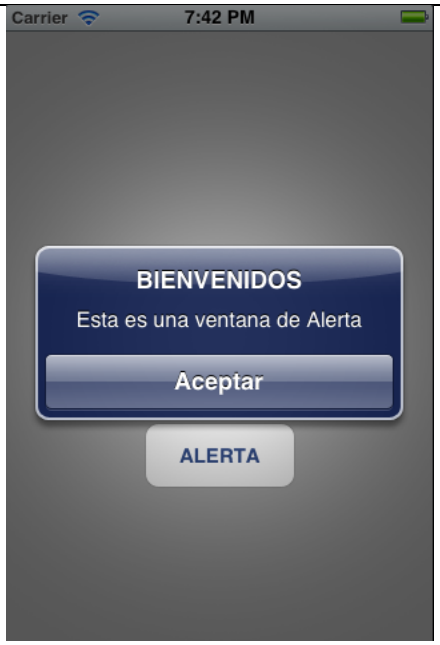


Figura. 2.19. Implementación Gráfica en XCODE.

2.4 COMPATIBILIDAD

La compatibilidad de las aplicaciones realizadas en estas plataformas son solo para Iphone, Ipad, Ipod u otros dispositivos Apple similares, cada uno de ellos ejecuta aplicaciones en ellos. Sin embargo aún dentro de estos no es posible instalar cualquier aplicación ya que las personas encargadas las han desarrollado a partir de una versión de sistema operativo, y por lo tanto estas solo podrán ser cargadas en dispositivos que dispongan de esta versión o superior a ella.

Esto es debido a los recursos, librerías y comportamiento de los dispositivos que son modificados en cada actualización de software, que en muchos casos cambia totalmente la programación del dispositivo, lo que hace necesario actualizar las aplicaciones a este nuevo cambio en los aparatos.

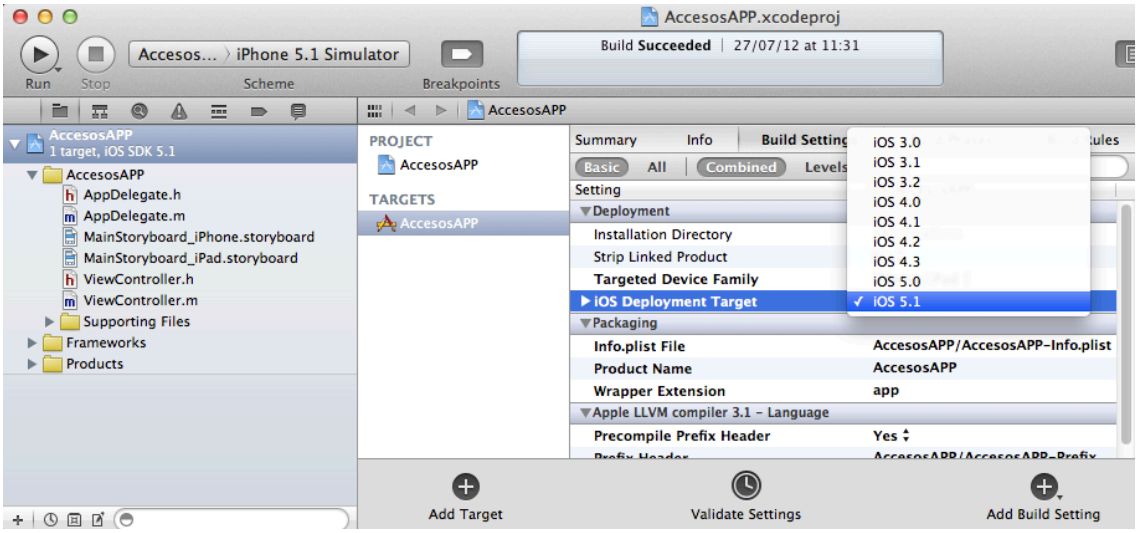


Figura. 2.20. Compatibilidad del software de IOS.

Cabe recalcar que en la parte de compatibilidad no solo contempla la parte de hardware, sino de software también, y al igual que en muchos otros dispositivos, algunos protocolos o Plug-ins no funcionen a causa de una falta de compatibilidad.

Esto es a cargo de las políticas de cada empresa, para el caso de los dispositivos Apple por ejemplo, no se contempla es uso de ActiveX o Flash Player como componentes de accesos embebidos, o de acceso web, inclusive Java, sin embargo para este ultimo si se mantiene vínculos de programación.

Estas pueden ser causas de incomodidad al momento de programar una aplicación al dispositivo para personas que poseen conocimientos avanzados en ellas y quieran acoplarlas a sus aplicaciones. Esto no puede resultar factible siendo una de las desventajas las políticas de uso de software en estos dispositivos.

Y por lo tanto obliga al programador a buscar otra estrategia de creación de código para obtener un resulta igual o aproximado al deseado, para esto en el medio ya se cuentan con protocolos Web avanzados que no solo dependen de un Flash o de ActiveX como en el caso de Windows que en el mercado aún sigue siendo considerado el más utilizado por los usuarios.

Como todo contenido Web puede ser visto desde cualquier dispositivo en código HTML no hace imposible la conexión. Para lo cual es indispensable el uso de herramientas alternativas.

CAPITULO III

DESARROLLO DEL SISTEMA DE CONTROL DE ACCESOS, Y MONITORIZACION.

CAPITULO 3

DESARROLLO DEL SISTEMA DE CONTROL DE ACCESOS, Y MONITOREO.

3.1 HARDWARE

Para la implementación del prototipo de control de accesos y monitoreo se lo ha compuesto por dos componentes como se muestra a continuación:

- **Un servidor embebido.-** Que es el componente que almacenará la página web donde a manera de intranet o internet se pueda tener el acceso al control de puertas y a las cámaras para permitir la apertura de las compuertas, todo esto mediante la comunicación con el dispositivo controlador.

- **Un dispositivo controlador.-** El controlador que está conformado por un circuito electrónico tanto para la parte digital donde se receptan las señales enviadas por el dispositivo móvil hacia el servidor web y este a su vez comunica al controlador para dar paso a la etapa de potencia en la cual se tomará control de la parte eléctrica de las compuertas.

- **Una Aplicación Móvil.-** La aplicación móvil tiene como objeto facilitar el ingreso a la página web a las personas discapacitadas con el uso de una interfaz amigable al usuario que la utilice, de manera básica el ingreso al servidor web para de esta manera tener el manejo de la etapa de control.

La comunicación determinada para el servidor Web está basada en una red cableada con conexión de un cable RJ45 al puerto de un router inalámbrico con conexión a internet, para acceso tanto en intranet, como desde internet. El Dispositivo Servidor Web debe estar alimentado por una fuente de voltaje de 5 [V] mínima, y hasta 9 [V] máxima para su correcto funcionamiento. Las cámaras IP conectadas de manera alámbrica o inalámbrica para la verificación de personal en la entrada, así como también la comunicación con el microcontrolador mediante una conexión serial con el servidor, el cual tiene el control de la apertura de accesos.

Debido que el servidor debe tener continua comunicación con el controlador, las distancias entre los mismos no debe ser tan amplia, por lo tanto se puede considerar que la distancia entre ambos es de máximo 10 metros, pero dado que las compuertas son controladas con un nivel de voltaje superior, esta serán compensadas con la etapa de potencia, en consecuencia la conexión para el controlador estará dada por una alimentación de 5[V] y para la etapa de potencia un voltaje máximo de 110[V].

Dado el estudio del esquema anterior, se procede a identificar de mejor forma el prototipo a ser implementado se hace indispensable un diagrama de bloques para observar el proceso de distribución de los componentes del prototipo.

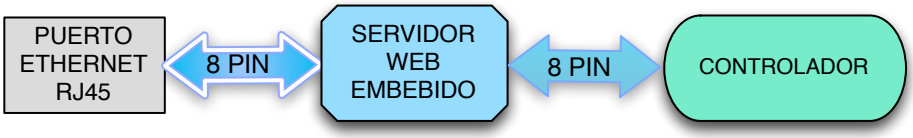


Figura. 3.1. Diagrama de bloques del Servidor.

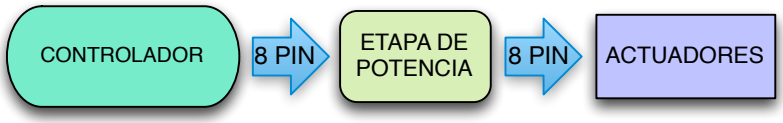


Figura. 3.2. Diagrama de bloques de Control de Potencia.

En el presente diagrama se pueden observar los siguientes componentes que lo conforman. El servidor embebido que es el que almacena la página web de presentación a la aplicación y es controlado a través de la Web, en la cual se realiza el redireccionamiento y verificación de las cámaras IP que se encuentren registradas en el servidor. Para poder enviar y recibir datos con el servidor, este lo realiza mediante una conexión de cable UTP cruzado hacia el Ruteador y a su vez al modem de conexión, por lo tanto la conexión hacia este es solo con comunicación cableada.

La etapa de comunicación entre el servidor y cualquier dispositivo de acceso a la Web lo recepta y envía por el conector hembra MAC-JACK, y después al utilizar la aplicación web contenida en el servidor, este tiene una etapa de comunicación con controlador, de tipo digital o de igual forma serial.
En el controlador se procesa la señal digital y se verifica el numero de compuerta ha ser procesada para su apertura. Una vez escogida, otorga la señal de activación del puerto seleccionado.

Por lo tanto es necesario el uso de una etapa de potencia para las salidas del controlador ya que como se está trabajando con un dispositivo de voltajes y corrientes digital, estas no son suficientes para procesar la tarea solicitada y activar la apertura de la compuerta. Para entender todo este proceso detalladamente se observa el siguiente análisis en cada uno de los elementos que lo conforman.

3.1.1 Equipo Servidor de aplicación Web.

El equipo utilizado para el Servidor Web, es un servidor embebido, el cual consta con una capacidad de almacenamiento para una página web codificada en código HTML, este consta internamente por puertos de entrada y salida, con el objetivo de interactuar con un controlador a través de una página Web.

Para poder almacenar la página web en un dispositivo de este tipo, el almacenamiento mínimo es de unos 40 KB de memoria, con capacidad de procesar imágenes para permitir una interfaz de usuario dinámica y amigable al usuario. Este se conecta directamente con dispositivo de conexión local intranet-internet, modem, de manera alámbrica e inalámbrica para su acceso. Este tipo de conexión es simple y de fácil utilidad, tanto con en parte eléctrica, ya que ambos dispositivos utilizan el mismo voltaje para la transmisión de datos. Para realizar dicha tarea se utilizará un dispositivo **Siteplayer** telnet que es un módulo electrónico que permite controlar dispositivos serie desde una conexión TCP/IP. El módulo incluye un completo servidor web con acceso por contraseña y un puerto serie que es controlado desde la pagina web.

3.1.1.1 Criterio de selección.

Debido al análisis realizado, de los requerimientos de un dispositivo de este tipo, se utilizará un servidor web embebido **SITEPLAYER**. Este servidor cumple en su totalidad con las especificaciones requeridas, es un dispositivo de menor tamaño de hasta 3 cm, y las ventajas que este ofrece son tanto la capacidad de conexión Ethernet, que incluye una memoria Flash de almacenamiento, con procesador de objetos e imágenes, y con la presencia de una interfaz de comunicación en serie con otros dispositivos.

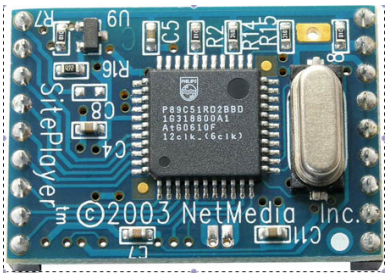


Figura. 3.3. Servidor Embebido SITEPLAYER.

El dispositivo indicado permite realizar una conexión al internet con interacción de entradas y salidas que pueden conectarse a un microcontrolador, dependiendo de la configuración de estos puertos. Es decir que se puede manejar el controlador desde una página Web a través de una conexión de internet, de manera que el dispositivo utiliza los protocolos necesarios para realizar dicha operación.

Por lo tanto si el dispositivo tiene la capacidad de comunicarse con un dispositivo microcontrolador, es debido a una conexión serial que trabaja con señales digitales, lo cual lo hace muy útil para su uso con otros dispositivos electrónicos que trabajan con este tipo de comunicación.

Este dispositivo se puede acoplar a diferentes procesos, lo cual resulta en gran ventaja para abaratar costos de elementos, y otros dispositivos. Para identificar esto se puede describir las siguientes.

- Se puede utilizar en aplicaciones de domótica para controlar accesos, luz, dispositivos electrónicos.

- En aplicaciones de monitoreo para la visualización de cámaras.

- En la utilización de este dispositivo en aplicaciones de tipo médica en el tratamiento y monitoreo externo de pacientes.

- Se puede utilizar en el control de procesos de tipo industrial .

- Para interacción con dispositivos móviles o smartphones.

3.1.1.2 Descripción.

La conexión de este dispositivo se basa en una conexión Ethernet y otra de tipo serial, por lo tanto mantiene una doble conexión entre dispositivos, ya que la una puede trabajar con protocolos web y código HTML, y la otra con otros dispositivos de comunicación digital, ya sea un puerto de entrada y otra de salida. Con este módulo y un conector RJ45 con filtros se puede controlar un dispositivo serie desde cualquier parte del mundo por internet.

Existe una versión montada que incluye el módulo, el conector serie, el conector RJ45 y el alimentador que permite controlar cualquier dispositivo serie desde una red Ethernet. Esta comunicación está basada en el estándar 10BaseT con una corrección automática de polaridad en Hardware. De esta manera en la comunicación del puerto serie llega a una transmisión con un rango de 115200 bits por segundo en interfaz hacia un procesador o controlador.

Y con este esquema se puede conectar directamente a un RJ45 con filtros internos o un puerto de estándar 10BaseT. El dispositivo se conecta a la red a través de DHCP o se puede la realizar la configuración manual de la configuración de la red si lo desea. Trabaja como un sistema de tiempo real para la modificación de páginas web, ya sean parámetros gráficos, etc. Posee una capacidad de almacenamiento de 48 KB, este puede almacenar una página web en código HTML, y por lo cual se puede utilizar cajas de texto o botones de campo que realicen acciones deseadas en el dispositivo.

Por lo tanto soporta protocolos como, ARP, TCP, UDP, ICMP, DHCP de los que se utilizan básicamente para el acceso a la web. Se puede realizar la programación de una aplicación Web en el servidor embebido ya sea mediante C++, JAVA, incluso Visual Basic. Se pueden asumir formatos bit, byte, integer, long, string y gráficos de 768 Bytes para objetos de tipo SiteObjectsTM. De la siguiente imagen se describen el uso de cada uno de los pines del dispositivo SITEPLAYER.

Tabla 3.1. Descripción de los pines de SITEPLAYER

PIN #	NOMBRE	DESCRIPCION
1	LED de Enlace	Indica cuando se ha establecido la conexión.
2	Rx+	Pin positivo del receptor
3	Rx-	Pin negativo del receptor
4	Tx-	Pin negativo del transmisor
5	Tx+	Pin positivo del transmisor
6	Vss	Tierra
7	RXD	Pin de Recepción UART.
8	TXD	Pin de Transmisión UART.
9	VCC	Pin de Alimentación (+5V)
10	Reset	Pin de Reset
11–18	E/S	Pines de entrada o salida (Configurables)

3.1.1.3 Esquema de conexión del Servidor.

Para realizar la conexión del Servidor a la red como se mencionó anteriormente es necesario el uso de un conector RJ45, sin embargo para que la comunicación sea efectuada satisfactoriamente se utiliza un MAC JACK con filtros de acople de impedancia, esto se puede denotar en la siguiente figura.

Figura. 3.4. Puerto MAC-JACK con filtros.

Este puerto es el único que puede funcionar correctamente con la comunicación hacia el dispositivo SITEPLAYER. El conector corresponde a la serie de LF1S022 que cumple con el estándar de la IEEE802.3 correspondiente a la conexión Ethernet. En la presente figura se muestra la estructura interna del puerto del MAC-Jack en acople a los pines del conector RJ-45 para generar la comunicación entre el Servidor y los puertos.

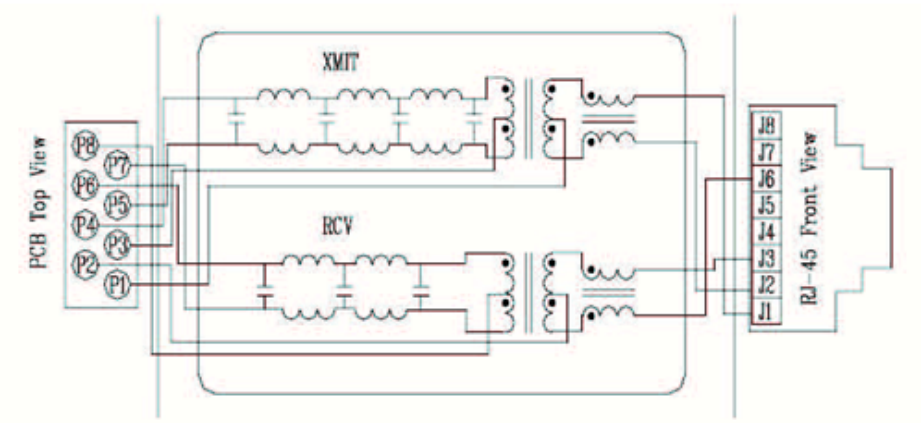


Figura. 3.5. Estructura del Mac-Jack y RJ-45

A continuación se muestra la disposición de los pines de comunicación del puerto RJ45 junto con el conector. Este conector y puerto RJ45 cumplen con un estándar de medidas de igual manera por lo tanto su tamaño y forma no cambiarán.

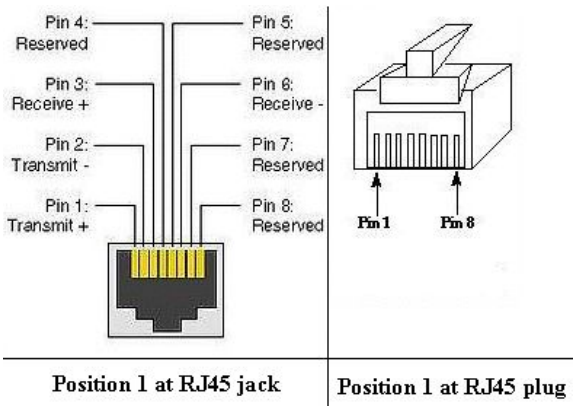


Figura. 3.6. Puerto y Conector RJ45

Como se observó anteriormente los filtros acoplan las señales eléctricas enviadas a través del cable, que en este caso corresponde a un cable cruzado, una vez filtradas serán transmitidas al servidor para su procesamiento. Para definir la configuración del cable que va conectado al servidor, se observan los parámetros y normas de un cable cruzado como se observa a continuación.

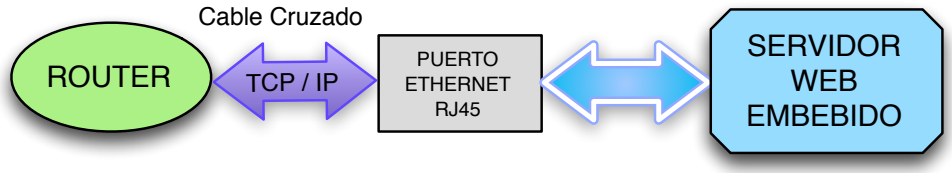


Figura. 3.7. Diagrama de Bloques de conexión a Router o Módem.

Se utiliza este tipo de cable ya que como su nombre, es cruzado, el pin de lectura va conectado al pin de escritura, tanto como de lado del servidor así como también del lado del modem o router al que irá conectado. Esta comunicación es conocida como comunicación bidireccional o full dúplex. Esta norma es la utilizada para la elaboración del cable cruzado se lo realiza con un cable UTP de categoría 5 con 4 pares trenzados los cuales son diferenciados por colores, para lo cual se utiliza una norma de disposición de los cables. Para su elaboración se utilizan dos normas T568A y T568B las cuales conforman el cable cruzado, estas son descritas en disposición de colores como se muestra a continuación:

Tabla 3.2. Normas para Cable Cruzado.

Nro de Pin.	T568A	T568B
1	Blanco-Verde	Blanco-Naranja
2	Verde	Naranja
3	Blanco-Naranja	Blanco-Verde
4	Azul	Azul
5	Blanco-Azul	Blanco-Azul
6	Naranja	Verde
7	Blanco-Café	Blanco-Café
8	Café	Café

En la tabla se observa la disposición de los cables, los pines 1 y 2 son de transmisión, y los pines 3 y 6 son de recepción, por esta razón se dice que es cruzado.

3.1.2 Cámaras IP.

La cámara IP interactuará con el servidor Web y esto lo realizará mediante los protocolos de conexión a internet con el uso de una dirección IP para la cámara donde el servidor desde la página web, a través de su aplicación el usuario redireccionará con un link hacia cada una de las cámaras dispuestas en los sitios donde se ubicara el control de accesos a las puertas.

Este tipo de cámaras se pueden monitorear ya que tienen la capacidad de enviar señales de video y algunas incluso audio, mediante una conexión a un Router o en cualquier dispositivo de una red local como un concentrador o Switch que a su ves debería estar conectados a un modem para su salida a Internet desde cualquier lugar en el mundo.

El equipo tiene la capacidad de envío de imágenes grabadas en secuencias o fotogramas, con formato digital para la recepción en equipos informáticos que se encuentren situados dentro de la red o ya sea fuera de ella. De esta manera se puede otorgar una verificación de todas las acciones que se llevan a cabo en el lugar de evento.

3.1.2.1 Dispositivo de selección.

Para el uso de estos dispositivos se ha seleccionado para el caso una cámara IP de marca D-Link modelo DCS-930 la cual permite una conexión a través de cable de red de área local y a través de la red inalámbrica, de ser requerido el caso.

Figura. 3.8. Cámara IP D-Link.

Para la presentación de imágenes, la cámara lo realiza a través de su propia aplicación a mediante Java para cualquier dispositivo compatible con esta plataforma. Y a través de ActiveX para dispositivos que disponen de este plugin.

Se observa la disposición de los tipos de conexión de la cámara.

Figura. 3.9. Conexión de cámara IP de tipo cableada.

En la imagen mostrada se observa la disposición de la conexión cableada hacia la cámara. Cabe recalcar que la conexión hacia esta se la hace a través del dispositivo móvil a través de la red inalámbrica.

Figura. 3.10. Conexión de cámara IP de forma inalámbrica.

En la presente imagen manteniendo el concepto anterior la conectividad será de forma inalámbrica. Dicha conexión hacia la cámara en cuanto a calidad de la imagen dependerá de la distancia a la cual sea colocada. Para su configuración existe dos formas, la una es accediendo a través de un instalador ubicado en el contenido de la caja de la cámara IP de la empresa fabricante, o ya sea a través del acceso por IP que viene de igual forma preconfigurada de fábrica. Con el objetivo de permitirle su configuración y uso de la misma de una manera fácil.

La cámara seleccionada posee las siguientes características como se muestra a continuación:[24]

- Cámara habilitada con mydlink.
- Tecnología Wireless N para una conexión perfecta desde cualquier punto de la casa.
- Sensor de 1,0 Lux para condiciones de poca luz.
- Envío de alertas por correo electrónico por detección de movimiento.
- Micrófono integrado
- Seguridad inalámbrica con solo tocar un botón gracias a WPS (Wi-Fi Protected Setup).
- Soporte de DNS dinámico para acceder fácilmente a la cámara desde cualquier punto de internet.

[24] http://www.dlinkla.com.ec/home/productos/producto.jsp?idp=1472

3.1.2.2 Esquema de conexión.

El esquema de conexión para el funcionamiento de la cámara es simple y solo es necesario energizar con un adaptador de 12[V], con el cual entra en funcionamiento, se conecta la fuente de poder en la parte trasera, y tiene un led indicador de funcionamiento con el que se puede identificar que se halla encendida. En la siguiente imagen se pueden apreciar los detalles en la parte posterior de la cámara.

Figura. 3.11. Parte posterior de una Cámara IP

Como se puede apreciar, se tiene el puerto de la fuente de poder y el conector del cable de red sobre ella. Esta cámara se puede colocar en cualquier posición ya que posee un soporte para ubicarla en cualquier superficie, atornillándola a ella, y colocándola en la posición deseada. Para el funcionamiento de la cámara se puede lograr de dos maneras, la una cableada y la otra inalámbrica desde un Ruteador. Como se observa en la imagen esta es la forma cableada en la que puede trabajar la cámara IP.

Figura. 3.12. Conexión de Cámara mediante RJ45

En ella se observa cuando el link se ha establecido con el equipo, y existe la comunicación con el mismo, cuando en la parte frontal se verifica el led de color verde con lo cual indica que la cámara se encuentra conectada y transmitiendo información.

Figura. 3.13. Conexión de la cámara de manera inalámbrica.

Y de igual forma se puede apreciar la cámara en su funcionamiento inalámbrico, donde existe de igual manera un led indicador al existir la conexión inalámbrica.

3.1.2.3 Conexiones disponibles.

Las conexiones disponibles dependerán del numero de cámaras ubicadas en los accesos y puertas, o ya sea para seguridad. Para el equipo que se planee utilizar cabe recalcar que puede ser tanto de manera cableada como de manera inalámbrica. Para la conexión cableada de la cámara es necesario el uso de un cable de red con un conector RJ45 que se conecte hacia el Ruteador o modem. Su funcionamiento es simple ya que se lo realiza de la siguiente manera. Para que cada cámara IP conste con una IP designada de forma automática y fija, es necesaria registrar por MAC dentro de la configuración del Ruteador, para esto a cada MAC se le asignará una IP correspondiente al área. Ya sea de forma cableada o inalámbrica el dispositivo siempre les asignará dichas IP a las MAC de las cámaras IP. De mejor manera las conexiones es mejor realizarlas con una IP estática en los equipos para distribuir de mejor manera el rango de IP disponibles.

3.1.3 Dispositivo móvil utilizado.

Para definir el dispositivo móvil utilizado en el sistema de control de accesos, en general sería imprescindible el uso de un equipo que mantenga una conexión de datos de forma inalámbrica o cableada y que posea un navegador en compatibilidad con HTML. Con lo cual estaría satisfecha la necesidad de utilizar una interfaz sobre un dispositivo móvil, sin embargo, debido que el prototipo es orientado hacia el uso de personas con discapacidad motriz, es necesario un manejo más simple de ciertas herramientas de los equipos móviles que reduzcan la cantidad de interactividad y manipulación de teclas u otros campos de modificación.

En otras palabras, mientras menos actividad tenga que realizar la persona con discapacidad más efectivo será el uso del sistema de control. En el medio la cantidad de dispositivos móviles existentes, en su mayoría son difíciles de manejar o poseen sistemas operativos móviles de bajo rendimiento por lo cual se hace imprescindible el uso de móviles con alta capacidad de operación y efectividad. Para la implementación del sistema a través del dispositivo móvil, la única norma es que este dispositivo portátil posea en su configuración y características una conexión a internet y un navegador para realizarlo. Para el presente proyecto se ha seleccionado el dispositivo IPHONE; un teléfono de alta operatividad y rendimiento el cual brinda prestaciones de fácil manejo de aplicaciones y navegación web, con el se pretenderá acceder más adelante mediante una aplicación.

Figura. 3.14. Dispositivo IPhone de Apple.

Debido que el aplicativo es orientado hacia personas discapacitadas, se hace necesario utilizar una aplicación en un dispositivo móvil. Dicha aplicación debe ser programada en una plataforma dicho dispositivo, que en este caso es un IPHONE. La ventaja de utilizar una aplicación para este dispositivo, es que no solo se puede utilizar en este, sino en la mayoría de gama de equipos Apple ya sean IPOD, IPAD, IPHONE que contengan una versión del sistema operativo del IOS 5.1.

3.1.3.1 Características técnicas.

El IPhone es una de los teléfonos inteligentes que posee contenidos multimedia con una conexión a internet, tiene una pantalla táctil de tipo capacitiva y en disponibilidad de muy pocos botones físicos. Se pueden cargar páginas web por medio de una conexión Wifi o mediante la conexión de una red móvil 3G y LTE.

Para demostrar de manera general las características técnicas de este dispositivo móvil se hacen validas las siguientes entre las de mayor consideración para su uso:[25]

Pantalla.

- Pantalla Retina.
- Panorámica Multi-Touch de 3,5 pulgadas (en diagonal).
- Resolución de 960 por 640 píxeles a 326 p/p.
- Relación de contraste de 800:1.
- Cubierta oleófuga resistente a huellas dactilares en las partes frontal y trasera.
- Compatible con la presentación simultánea de múltiples idiomas y grupos de caracteres.

[25] http://support.apple.com/kb/SP587?viewlocale=es_ES&locale=es_ES

TV y vídeo

- Formatos de vídeo compatibles: vídeo H.264 de hasta 720p, 30 fotogramas por segundo, con sonido AAC-LC de hasta 160 Kb/s a 48 kHz y estéreo en los formatos de archivo .m4v, .mp4 y .mov; vídeo MPEG-4 de hasta 2,5 Mb/s, 640 por 480 píxeles, 30 fotogramas por segundo, perfil Simple con sonido AAC-LC de hasta 160 Kb/s a 48 kHz y estéreo en los formatos de archivo .m4v, .mp4 y .mov;

- Motion JPEG (M-JPEG) de hasta 35 Mb/s, 1.280 por 720 píxeles, 30 fotogramas por segundo, sonido en ulaw y estéreo PCM en el formato .avi.

Redes móviles e inalámbricas

- UMTS/HSDPA/HSUPA (850, 900, 1.900, 2.100 MHz).
- GSM/EDGE (850, 900, 1.800, 1.900 MHz).
- Wi-Fi 802.11b/g/n (802.11n solo a 2,4 GHz).
- Bluetooth 2.1 + EDR.

Localización geográfica

- GPS asistido.
- Brújula digital.
- Wi-Fi.
- Telefonía móvil.

Teniendo en consideración las especificaciones técnicas a los parámetros indicados, hacen que este dispositivo tenga características completas para el uso que se le otorgará en la aplicación para uso de los discapacitados. Cabe recalcar que estas especificaciones son muy similares otros productos de la misma marca, por lo tanto no es el único dispositivo con el que se puede utilizar el aplicativo que más adelante se dará a conocer.

3.1.3.2 Criterios de selección.

Este dispositivo se ha seleccionado debido que es uno de los primeros medios portátiles en poseer una pantalla completamente táctil, esto es fácil de diferenciar de otras pantallas que son de tipo resistivas y requieren de presión, a diferencia de esta, el iPhone ofrece una pantalla con variación de voltaje mediante celdas capacitivas en su cristal la cual ofrece funcionalidad y rapidez.

Para poder utilizar un dispositivo móvil que permita el uso de diferentes aplicativos de control en personas con discapacidad motriz, se hace indispensable el uso de una interfaz táctil que mantenga el orden de botones y una interactividad con el usuario que la utiliza. No es imprescindible el uso de este equipo, pero para este caso de estudio e implementación se denota la capacidad de respuesta e interacción que necesita una persona con discapacidad. Es decir que el usuario debe poseer la menor cantidad de complicaciones al momento de utilizar el aplicativo. Y este dispositivo le otorga la intuición al usuario para manejarlo, aún sin tener un manual a su lado, debido que cualquier acceso está al alcance de un click.

3.1.4 Circuito de Control.

Para esta parte del circuito el control se lo realiza hacia puertas y luces del área del usuario, las cuales son controladas por una etapa de control acompañada de una etapa de potencia. Cuando se haya acoplado el servidor embebido al controlador y este a su vez a la etapa de potencia se conseguirá el resultado deseado en la ejecución. Cabe recalcar que la etapa de potencia debe estar a parte de la parte digital para evitar inconvenientes con las grandes cantidades de voltaje que se utiliza para el control de luces y compuertas.

A continuación se realiza el estudio de los componentes seleccionados para ejecutar dichas tareas.

3.1.4.1 Microcontrolador utilizado.

El microcontrolador es un dispositivo que permite la programación lógica de lectura de entradas y salidas para activar o desactivar los componentes del circuito de control de acceso. Para el proyecto se utilizan 4 salidas digitales, que pueden ser aplicadas con una comunicación serial como protocolo de comunicación con el fin de activar y desactivar los actuadores.

Como se ha mencionado anteriormente serán luces y compuertas. Las mismas se verán reflejadas en el dispositivo móvil cuando estas sean activadas. El controlador trabaja con señales digitales, con lo cual se debe manejar rangos entre 0 – 5 [V] de TTL para que este las pueda procesar. Para lo cual se ha seleccionado uno de los microcontroladores conocidos para generar dicha tarea y para este caso será el siguiente:

PIC 16F2550.

Se ha seleccionado dicho dispositivo puesto que cumple con las características deseadas, adicional a esto este permitirá la escalabilidad del sistema para futuras ampliaciones en cuanto a nuevos proyectos de estudio e implementación. Este Chip posee características significativas que al momento de aplicarlo resulta de manera eficiente y práctica al ser implementada en una sistema de control de accesos.

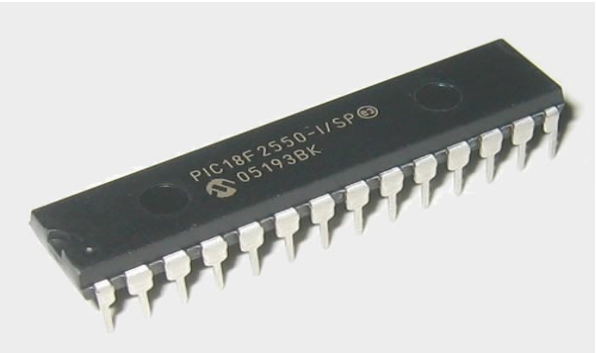


Figura. 3.15. PIC18F2550

Entre las características más notorias de este chip podemos denotar las siguientes.

- Posee una memoria de tipo FLASH la misma que puede ser reprogramada eléctricamente.
- Posee la capacidad de mantener una comunicación serial.

Como características técnicas del dispositivo se observa:.[26]

PARAMETROS	DATOS
Program Memory Type	Flash
Program Memory (KB)	32
CPU Speed (MIPS)	12
RAM Bytes	2,048
Data EEPROM (bytes)	256
Digital Communication Peripherals	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	1 x 8-bit, 3 x 16-bit
ADC	10 ch, 10-bit
Comparators	2
USB (ch, speed, compliance)	1, Full Speed, USB 2.0
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2 to 5.5
Pin Count	28

Figura. 3.16. Características del PIC18F2550

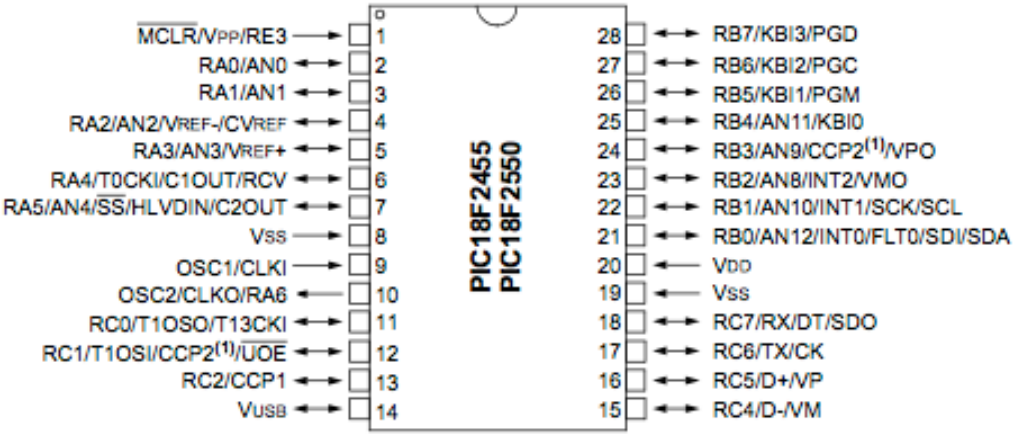


Figura. 3.17. Descripción de los pines del PIC18F2550.

[26] Obtenido de Datasheet PIC18f2550.

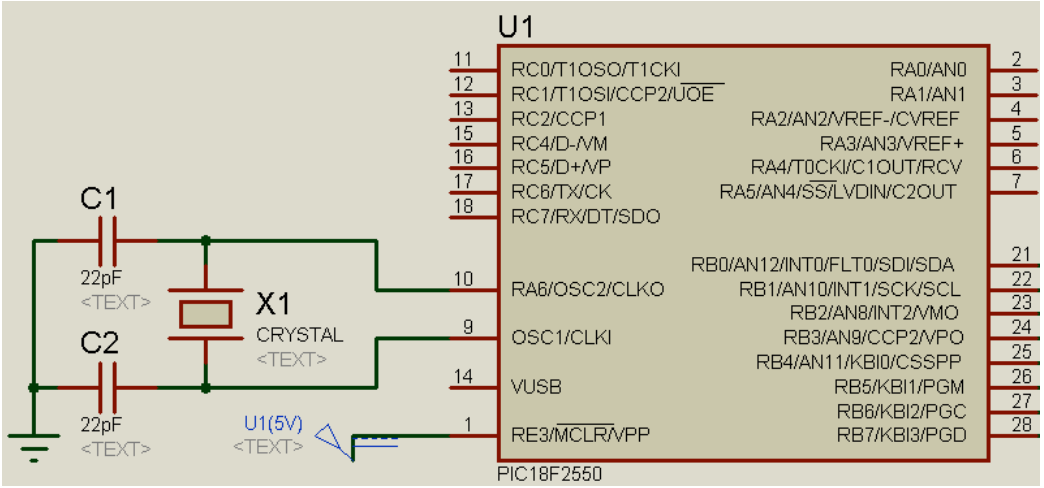


Figura. 3.18. Diagrama de conexión del PIC18f2550.

Tabla 3.3. Descripción de los pines del PIC18F2550.[27]

Pin en el chip	E/S Digitales	Otras	Pin en el chip	Nombre de salidas
0	Si	I2C I/O - SPI SDI	**21**	RB0/AN12/INT0/FLT0/SDI/SDA
1	Si	I2C SCL - SPI SCK	**22**	RB1/AN10/INT1/SCK/SCL
2	Si	-	**23**	RB2/AN8/INT2/VMO
3	Si	-	**24**	RB3/AN9/CCP2/VPO
4	Si	-	**25**	RB4/AN11/KBI0
5	Si	-	**26**	RB5/KBI1/PGM
6	Si	-	**27**	RB6/KBI2/PGC
7	Si	-	**28**	RB7/KBI3/PGD
8	Si	Serial Tx	**17**	RC6/TX/CK
9	Si	Serial Rx - SPI SDO	**18**	RC7/RX/DT/SDO
10	Si	-	**11**	RC0/T1OSO/T13CK
11	Si	PWM	**12**	RC1/T1OSI/CCP2/UOE
12	Si	PWM	**13**	RC2/CCP1
13	Si	-	**2**	RA0/AN0
14	Si	-	**3**	RA1/AN1
15	Si	-	**4**	RA2/AN2/VREF-/CVREF
16	Si	-	**5**	RA3/AN3/VREF+
17	Si	-	**7**	RA5/AN4/SS/HLVDIN/C2OU
Corrida	Solo salida	Led de Ejecución	**6**	RA4/T0CKI/C1OUT/RCV
USB+	-	USB	**16**	RC5/D+/VP
USB-	-	USB	**15**	RC4/D-/VM
Vusb	-	USB	**14**	VUSB
Reset	-	Interruptor de reinicio	**1**	MCLR/VPP/RE3
Vdd (5V)	-	-	**20**	Vdd
Vss (GND)	-	-	**8**	Vss
Vss (GND)	-	-	**19**	Vss
OSC1	-	Quartz	**9**	OSC1/CLKI
OSC2	-	Quartz	**10**	OSC2/CLKO/RA6

[27] Obtenido de Datasheet PIC18f2550.

3.1.5 Etapa de Potencia.

Como se había mencionado anteriormente una vez que se ha seleccionado el controlador a utilizar es necesario acoplarlo a la etapa de potencia la misma que se ha tomado en consideración los siguientes criterios para el diseño. Debido que en todo el circuito se utilizan datos digitales, se procederá a utilizar estas señales de 0 [V] y 5 [V] que provienen del controlador para dar paso al uso de señales de voltaje mayores para activar las compuertas y luces del sistema de control de acceso.

Los focos y compuertas utilizan voltajes de alimentación equivalentes a 110VAC de corriente alterna, con una corriente de 50mA. Es por este motivo que los componentes a ser seleccionados deben ser capaces de soportar estas cantidades de voltaje y corriente mencionadas. Adicional dichos elementos debería soportar estos valores sin mantener un alto recalentamiento ni tampoco sufran desperfectos ya que de ser así podrían afectar al circuito de control. Los elementos seleccionados son los siguientes:

3.1.5.1 Optoacoplador MOC3031.

Es un dispositivo que está compuesto por un diodo de arseniuro de galio, infrarrojo que actúa como emisor ópticamente acoplado a un detector monolítico de silicio donde se verifica su desempeño en un circuito interno de un cruce de cero que conmuta a los Triac en los cruces de cero de las señales alternas.

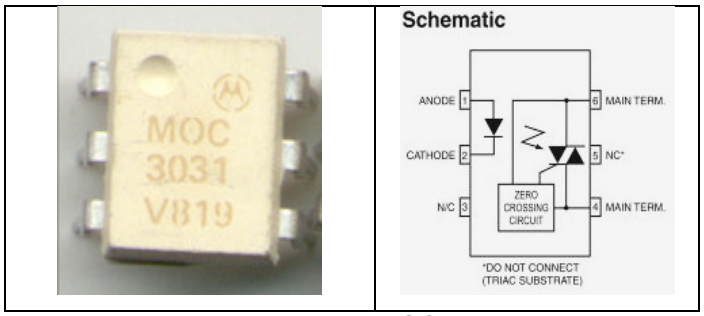


Figura. 3.19. MOC3031

Dado que la utilidad de este elemento radica en su capacidad de cruce por cero, la capacidad para conseguir que el triac se cierre en el momento que este identifica 0[V] permite evitar que potencias elevadas conmuten y se produzcan sobretensiones transitorias, adicional amortigua señales parásitas, picos de tensión y no reacciona bruscamente ante la presencia de luz. Las aplicaciones de control de fase para el uso del MOC3031 van de la mano con el uso de los triac BT136 que es utilizado como tiristor de switcheo bidireccional, para de esta manera permitir el paso de voltajes a manera de regulador.

3.1.5.2 Triac BT136.

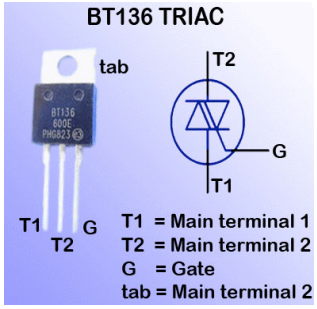


Figura. 3.20. Triac BT136.

Este elemento tiene como utilidad el actuar como un agente regulador de potencia hacia la carga, de manera que este triac se dispara cada medio ciclo cuando la corriente pasa por cero, de esta manera se evitan sobrevoltajes y arcos que son derivadas de la conmutación de las cargas inductivas que se almacenan durante el funcionamiento.

Tabla 3.4. Identificación de Componente TRIAC.[28]

Q 20 04 F 3 1 X

Device Type
Q = Triac or Alternistor

Voltage Rating
20 = 200 V
40 = 400 V
60 = 600 V
80 = 800 V
K0 = 1000 V

Current Rating
X8 = 0.8 A
01 = 1 A
04 = 4 A
06 = 6 A
08 = 8 A
10 = 10 A
12 = 12 A
15 = 15 A
25 = 25 A
30 = 30 A
35 = 35 A
40 = 40 A

Package Type
D = TO-252 (Surface Mount)
E = TO-92 (Isolated)
F = TO-202 (Non-isolated)
J = TO-218X (Isolated)
K = TO-218 (Isolated)
L = TO-220 (Isolated)
N = TO-263 (Surface Mount)
P = *Fastpak* (Isolated)
R = TO-220 (Non-isolated)
V = TO-251 (Non-isolated)
W = TO-218X (Non-isolated)

Special Options
V = 4000 V Isolation
(TO-220 Package Only)

Lead Form Dimensions
TO-202
TO-220
TO-92
TO-218X
TO-218

Gate Variation
DH3 and VH3 = 10mA (Q I, II, III)
3 = 10 mA (Q I, II, III)
H3 = 20mA (Q I, II, III)
4 = 25 mA (Q I, II, III)
H4 = 35 mA (Q I, II, III) *
5 = 50 mA (Q I, II, III)
H5 = 50 mA (Q I, II, III) *
6 = 80 mA (Q I, II, III) *
7 = 100 mA (Q I, II, III) *

* NOTE:
Alternistor device; no Quadrant IV operation

A continuación se muestra la terminología de la etiqueta técnica denotada en los Triac y en algunos transistores como se muestra a continuación.

Q = Triac.

60 = 600 Voltios.
04 = 4 Ampers.
F = Encapsulado de tipo TO-220, sin aislamiento.
4 = 25 mA Variación de corriente de Gate.
1 = Tamaño del tipo TO-220.

Normalmente estos elementos tienen un encapsulado TO-220, que dependiendo de la potencia a disipar y la corriente de trabajo, el valor de la corriente de trabajo RMS es de 4 AMP y voltaje máximo de 600 [V].

[28] Obtenido de Datasheet Triac BT136c.

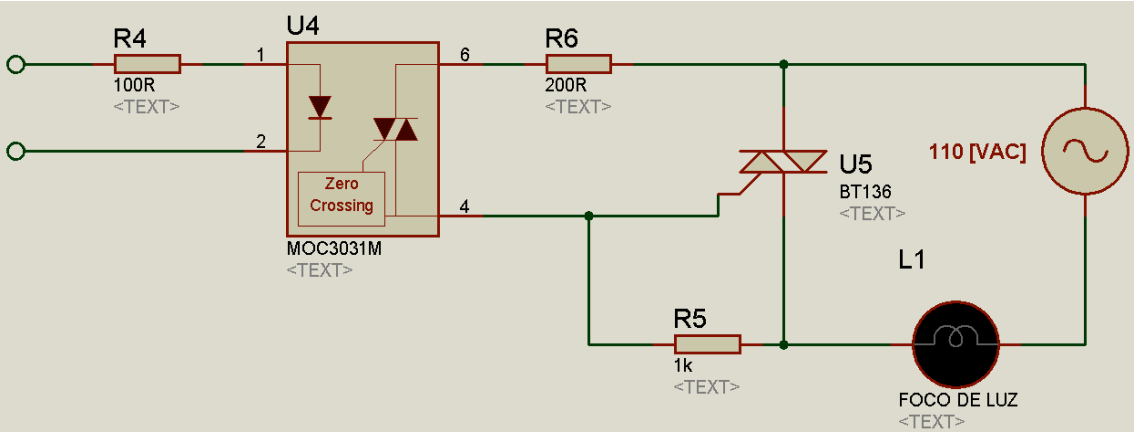


Figura. 3.21. Circuito de Potencia.

Como se había mencionado anteriormente el MOC3031 actúa como aislante del circuito de control de la etapa de potencia, la resistencia de 100 ohms limita la corriente que ingresa la MOC3031 para permitir la activación del mismo con una corriente máxima de 30mA. Este tipo de circuito es aplicado comúnmente en aplicaciones de:

- Accionamiento de motores de corriente alterna.
- Sistemas de control de iluminación.
- Accionamiento de actuadores.

3.1.6 Diseño del circuito de control de Acceso.

A continuación se presentan los diagramas del circuito de control de acceso, sobre este se puede observar el microcontrolador acoplado con la etapa de potencia. Como se había especificado anteriormente el circuito consta de 4 salidas para controlar 2 áreas del prototipo.

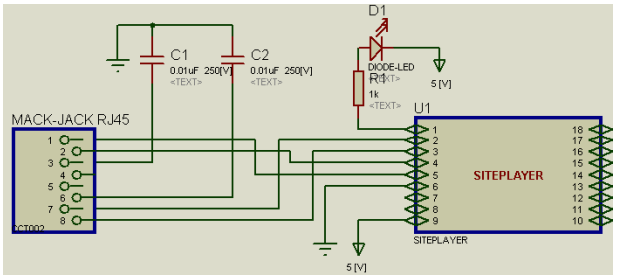


Figura. 3.22. Diagrama de Conexión del SitePlayer

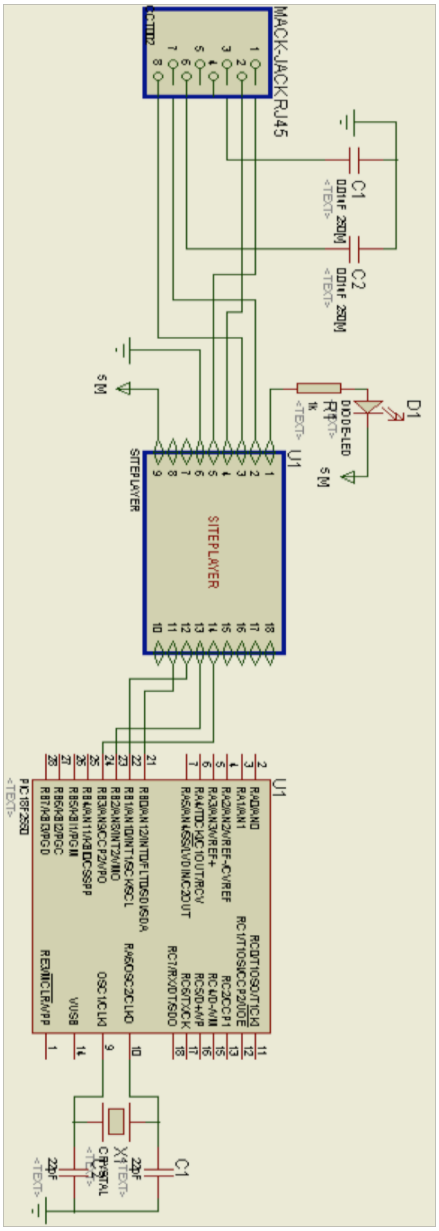


Figura. 3.23. Conexión entre SitePlayer y Controlador.

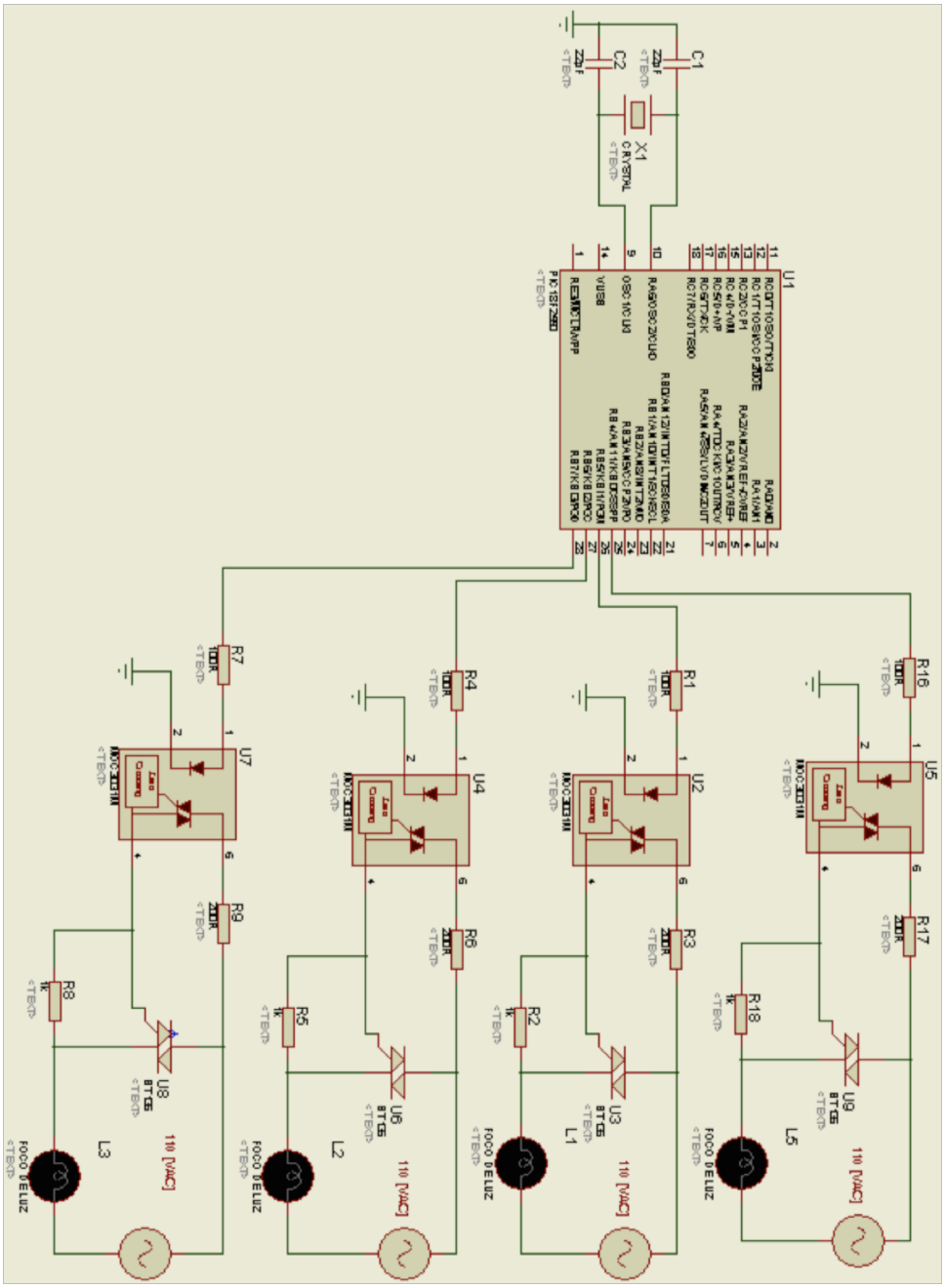


Figura. 3.24. Diagrama del Controlador y Circuito de Potencia.

3.2 DISEÑO DEL SOFTWARE.

Para el diseño del esquema de configuración del servidor web, se cuenta con el diseño gráfico de la misma. Puesto que el objetivo es descartar que el aplicativo sea de uso exclusivo de un solo dispositivo. Mediante el uso de este tipo de servidor web embebido existe la posibilidad de codificar una página web para el acceso universal de cualquier dispositivo móvil o PC que desee acceder al aplicativo desde el acceso a una red definida exclusivamente para el uso del dispositivo ya sea a través de una red inalámbrica o cableada sea el caso.

Para conseguir esto, es necesario realizarlo con cierto cuidado puesto que solo la persona que utiliza y tiene conocimiento de acceso sobre la misma, posea el ingreso a la misma, por lo cual, el método de acceso será a través de una contraseña de acceso que colabora con la seguridad de ingreso a la misma puesto que el portal WEB posee los aplicativos de monitoreo de cámaras y control de acceso de compuertas. Puesto que en el presente proyecto tiene como objetivo de estudio la implementación de el control de accesos y el monitoreo de cámaras, no se prevee estudiar a fondo dichas limitaciones en cuanto a la seguridad a nivel de la red ya sea inalámbrica o cableada.

Debido que la página WEB se encuentra almacenada dentro del servidor; esta debe estar codificada en código de lenguaje HTML para de esta forma garantizar su visualización en cualquier dispositivo móvil que tenga un tipo de conexión a la red, puesto que este lenguaje es el más aceptado a primer nivel, con lo cual se observa la universalidad del uso de la aplicación.

Dicha visualización dependerá del dispositivo y resolución de la ventana en la cual se despliegue, sin embargo esto no tendría ningún tipo de incidencia sobre el aplicativo en sí, puesto que todo el mecanismo se mantiene funcional, a pesar de un cambio de apariencia.

3.2.1 Diseño y Programación de la aplicación Web.

Para el diseño y programación de la página web para uso general del cualquier usuario que no dispone de un dispositivo móvil IPhone que es el dispositivo para el cual se ha diseñado el aplicativo para las personas con discapacidad motriz. El objetivo del aplicativo Web es el de monitorear las cámaras, Luces en caso de ser nocturno el evento y garantizar los accesos a puertas a través de la red. Para poder corresponder a estos requerimientos es necesario utilizar un software que pueda utilizar los lenguajes de páginas Web, en lenguaje HTML, y sea capaz de mantener en la memoria las mismas.

La página web se encuentra codificada en código HTML, para lo cual verificando se puede diseñar la misma a través del uso de un bloc de notas guardado con la extensión .HTM para la creación de la página y en caso de no tener los conocimientos para crearlo a partir de este se lo puede realizar a través de compiladores entre los cuales se pueden utilizar MACROMEDIA DREAMWEAVER, ADOBE DREAMWEAVER, .NET y otros más.

Para conocimiento los archivos de las páginas Web que serán contenidos en el Servidor Embebido no deben sobrepasar la capacidad de memoria del mismo puesto que este solo puede albergar 48Kbytes en su memoria interna, adicional los archivos deben ser colocados con la extensión .HTM que es la que reconoce el dispositivo SITEPLAYER.

El presente diseño de página WEB se lo ha realizado utilizando el software MACROMEDIA DREAMWEAVER.

3.2.1.1 Ingreso Web.

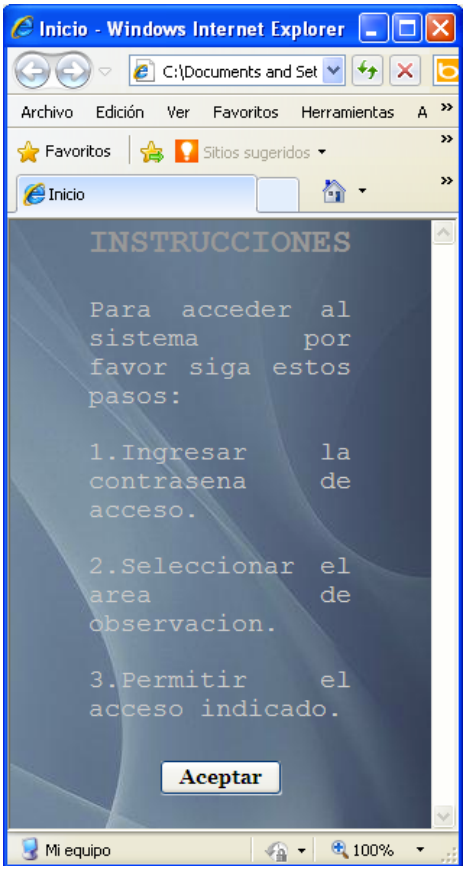


Figura. 3.25. Pantalla de Presentación.

La presentación que se muestra en la imagen es la pantalla de inicial con una introducción al sistema de control de accesos con una breve descripción del uso de este aplicativo Web. Al ingresar este viene acompañado a continuación por el ingreso a través de una clave que le garantice el acceso solo al usuario que lo controle. De esta manera se mantiene un parámetro a consideración la seguridad en el uso de la misma.

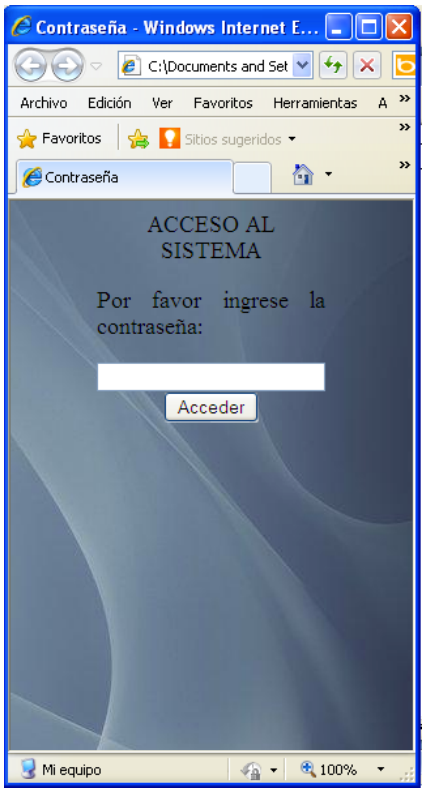


Figura. 3.26. Contraseña de ingreso al sistema.

En esta se verifica un título y un contenido descriptivo del acceso al sistema, para lo cual el usuario debe ingresar la contraseña en la casilla de texto y presionar en aceptar para que este sea redireccionado al aplicativo. Si se coloca de manera errónea la contraseña de ingreso, este inmediatamente retornará a la pantalla de inicio solicitándola nuevamente.

3.2.1.2 Control de Accesos y Monitorización.

Una vez ingresado al sistema de control de acceso y monitorización se puede identificar la imagen de la cámara IP que se encuentra en uso en el área verificada junto con botones que permiten al usuario la apertura de compuertas. Adicional se cuenta con la activación o desactivación de luces.

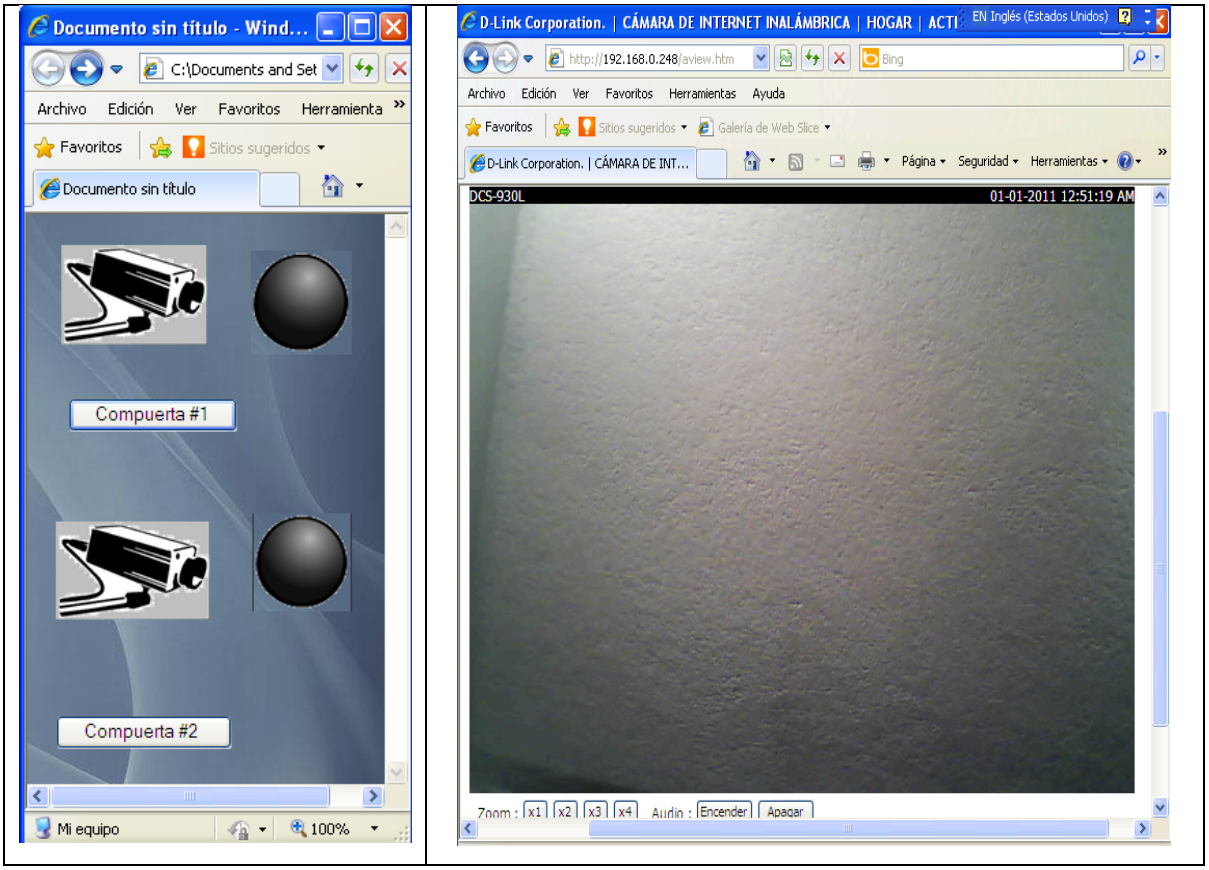


Figura. 3.27. Aplicación WEB

En este menú de selección de opciones se puede seleccionar el área que se desea monitorear y garantizar la apertura de los accesos. En ella se identifican los siguientes parámetros:

- **Activación de Iluminación.**

El motivo por el cual se toma en consideración el control de iluminación en el proyecto, es debido que la apertura de accesos de dará siempre y cuando la persona que utiliza el sistema esté cerciorada que existe una persona conocida en el área. Por consiguiente en horarios nocturnos es necesario el uso y activación de bombillas de iluminación.

El sistema cuenta con la activación de este tipo, y es fácil de verificarlo gracias al color de la esfera que indica que este ha sido activado o se encuentre desactivado.

Figura. 3.28. Encedido de luz.

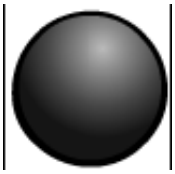


Figura. 3.29. Apagado de Luz.

Para activarlos basta con solo presionar en el área donde se encuentra la esfera, la misma que cambiará de color cuando se accione el evento demostrado.

- **Cámara IP.**

La verificación de la persona en el área deseada se lo realiza a través del uso de cámaras IP, que pueden ser alámbricas o inalámbricas según sea el requerimiento en cuanto a distancias. Utilizando el siguiente botón de identificación, se desplega el acceso a la cámara IP del área solicitada.

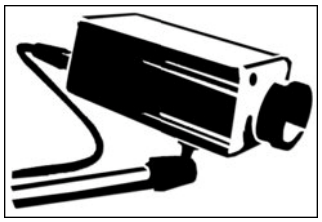


Figura. 3.30. Acceso a la Cámara IP.

Al realizar el click sobre este ícono, automáticamente redireccionará hacia la cámara de vigilancia que corresponde al área para que el usuario pueda identificar a la persona que solicita el acceso. La imagen de la cámara se visualizará de diferentes maneras, es decir en el caso de la cámara DLINK-930L permite un monitoreo en tiempo real de la imagen a través de protocolos JAVA y ACTIVEX, sin embargo también permite visualizarlo a través de imágenes, que si se refresca la página la imagen lo hará de igual manera.

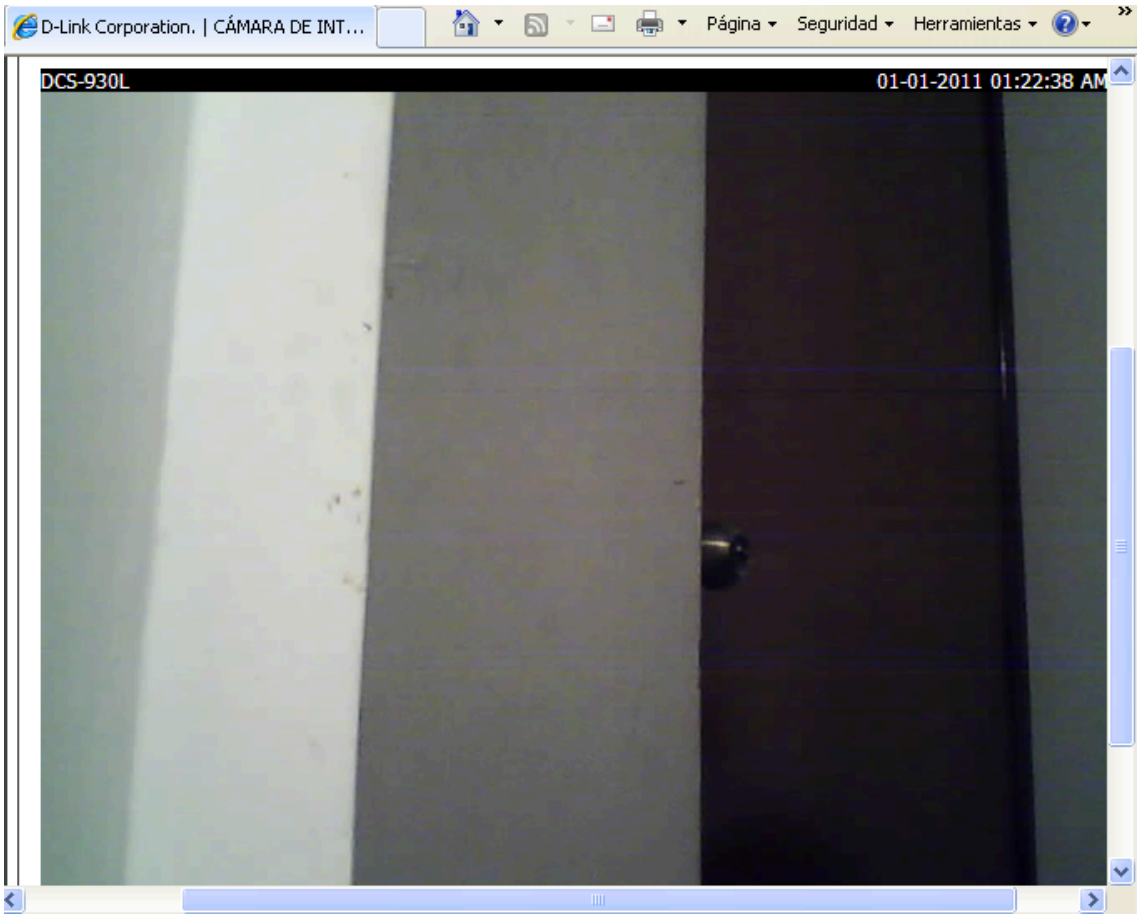


Figura. 3.31. Monitoreo de la Imagen de la Cámara.

Como se observa en la imagen la cámara posee el mecanismo para permitir al usuario del control de acceso verificar quien es la persona que desea ingresar y si este lo desea permitirle el acceso al mismo mediante la apertura de compuerta. De esta forma se mantiene la seguridad en acceso al sitio.

- **Apertura de compuertas.**

Para la apertura de compuertas es necesario tomar en consideración criterios de seguridad al momento de manipular los accesos ya que una acción incorrecta puede conllevar a la apertura de compuerta si ser tomada dicha acción por el consentimiento de la persona que lo utiliza. Para poder prevenir dicho evento se utiliza el siguiente botón que indica la compuerta del área a la que se permitirá el acceso.

Adicional, al momento de intentar ejecutarlo, el aplicativo WEB notifica al usuario si la decisión que está a punto de tomar es la deseada. Si es el caso el usuario puede garantizar la apertura de la compuerta, caso contrario el mismo puede cancelarla.

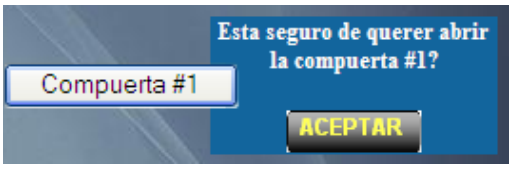


Figura. 3.32. Compuerta Nro 1.

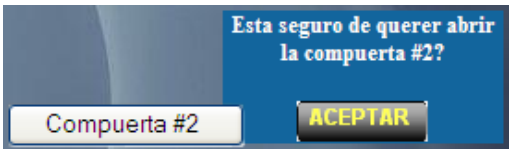


Figura. 3.33. Compuerta Nro 2.

Como se observa en las imágenes para poder garantizar el acceso el sistema siempre que el usuario acepte y ejecute la orden.

3.2.2 Especificaciones y requerimientos.

Una vez que la página Web a utilizar está diseñada con la salidas a utilizar en el dispositivo SITEPLAYER se procederá a cargar la página dentro del servidor. Puesto que el código HTML hace solo referencia a la presentación de la página WEB, se hace imprescindible el uso de una de las herramientas de JAVA para ejecutar métodos.

Para lo cual se utiliza un java script. Mediante el uso de este podemos realizar el envío y recepción de los datos enviados hacia el Servidor Web Embebido, los mismos que posteriormente se verán reflejados en el proceso y activación de la parte de control. Para esto se verifica la estructura de código de dichas funciones:

La primera instancia activa los puertos de salida del Servidor. Para activar las salidas, las líneas de comando envía como parámetro 0, la sentencia se puede ver identificada en el código:

De igual forma que la anterior instancia, para poder realizar el apagado se debe enviar 1 lógico (5 [V]) desde el servidor hacia el controlador.
Todas la sentencias acopladas en un solo párrafo de código se tiene como final el archivo .HTML que contiene la página de presentación del aplicativo WEB junto con sus elementos.

3.2.2.1 Diagrama de Ejecución.

En el presente diagrama se observa el proceso de ejecución desde el Aplicativo Web hasta la ejecución de salidas del controlador hacia la etapa de potencia.

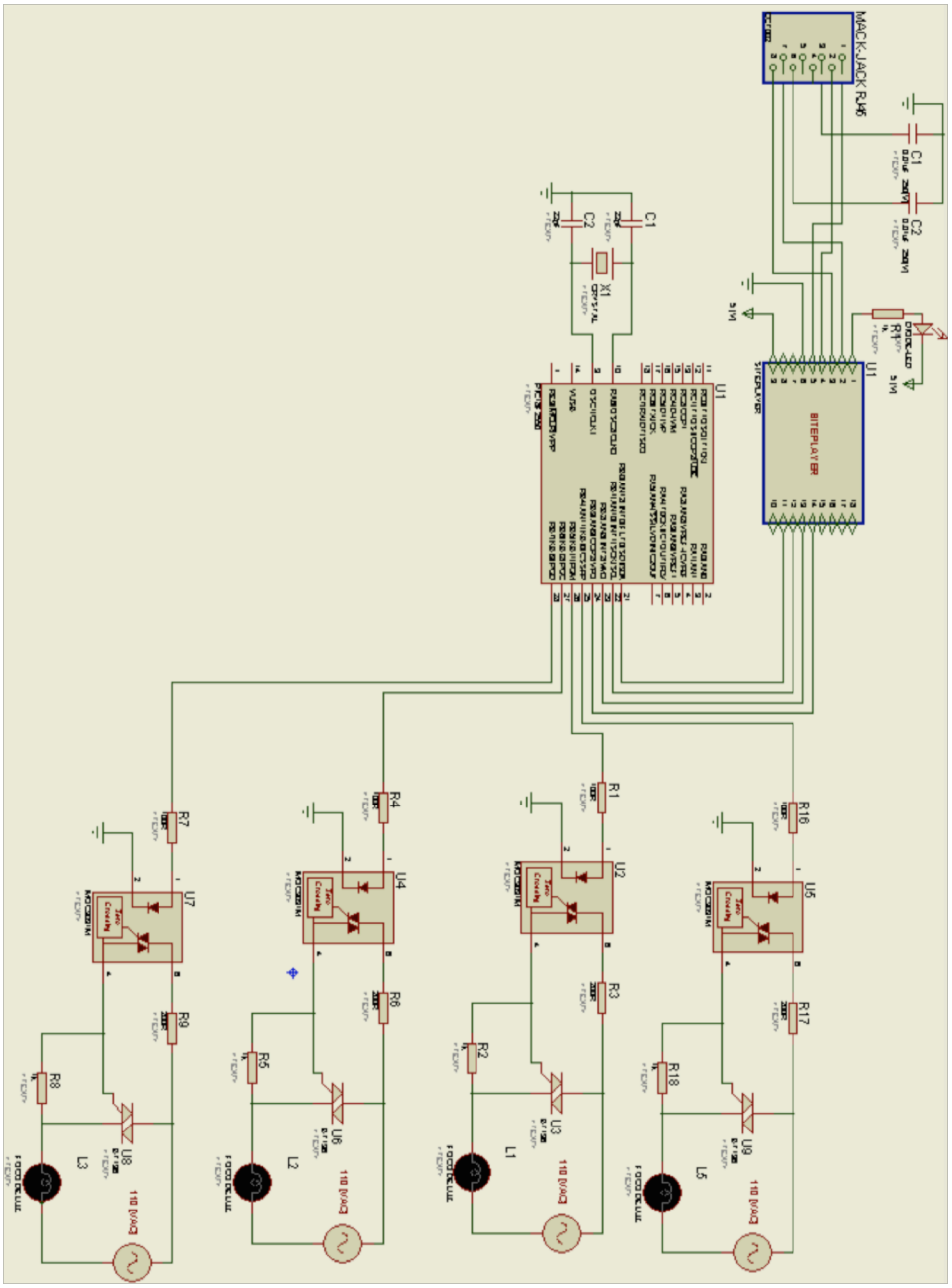


Figura. 3.34. Diagrama de ejecución del sistema

3.2.2.2 Diseño del servidor.

Para realizar la configuración del servidor de acuerdo a los parámetros ya definidos anteriormente se hace necesaria la codificación del mismo sobre 2 archivos de configuración, en los cuales se definen los puertos de entrada y de salida del servidor. Para este caso solo se definen puertos de salida que corresponden a control de luces y compuertas.

El primer archivo de configuración para el dispositivo SITEPLAYER define los puertos y comandos de configuración física del servidor, el mismo que posee un formato de archivo .SPD. El segundo archivo es de formato .SPI y es generado por el compilador del SitePlayer, adicional para poder verificar errores en el acceso a cada página WEB es impresindible la configuración de archivos de este mismo tipo que hagan referencia a la pagina web de ingreso.

En la siguiente imagen se muestra el código para la configuración del dispositivo Siteplayer.

myproj.spd.txt - Bloc de notas

Archivo Edición Formato Ver Ayuda

```
$Devicename "ServidorWeb-Controlador"
$DHCP on
$DownloadPassword ""
$SitePassword ""
$InitialIP "192.168.0.99"
$Sitefile "C:\Documents and Settings\Administrador\Escritorio\servidor Web\myproj.spb"
$Sitepath "C:\Documents and Settings\Administrador\Escritorio\servidor Web\webserver"


    org 0FF19h
COM    ds 128            ;serial port output (only takes up really one byte)

      org 0ff1Ah

baudrate    dw    65406            ;set baudrate to 9600

      org 0ff00h

p1            ds 1                 ; Port 1 all 8 bits

       org 0ff11h

io0           ds 1                 ; Port 1 Bit number 0
io1           ds 1                 ; Port 1 Bit number 1
io2           ds 1                 ; Port 1 Bit number 2
io3           ds 1                 ; Port 1 Bit number 3
io4           ds 1                 ; Port 1 Bit number 4
io5           ds 1                 ; Port 1 Bit number 5
io6           ds 1                 ; Port 1 Bit number 6
io7           ds 1                 ; Port 1 Bit number 7
```

Figura. 3.35. Archivo de configuración SPD_SITEPLAYER

El primer archivo de configuración define las los puertos y comandos de configuración física del servidor, el mismo que posee un formato de archivo .SPD. El segundo archivo es de formato .SPI y es generado por el compilador del SitePlayer, adicional para poder verificar errores en el acceso a cada página WEB es impresindible la configuración de archivos de este mismo tipo que hagan referencia a la página web de ingreso.

Para la creación de dicho archivo, los parámetros que son instanciados se deben especificar mediante definiciones de configuración que el SITEPLAYER utiliza para su funcionamiento y ejecución. Como se observa en la imagen las definiciones son instanciadas con el símbolo de $. En esta no se permiten etiquetas o espacios. Se pueden incluir comentarios que ingresados con un punto y coma (;) precediéndolos. Para la configuración del dispositivo se utilizan las siguientes definiciones, cabe recalcar que para ingresar argumentos, estos son ingresados en corchetes de la siguiente manera:

$Devicename [Nombre] .- En ella se asocia el nombre al Servidor para tener una referencia hacia el mismo, y esta a su vez no puede ser mayor a una logitud de 64 caracteres.

$DHCP [off, on].- En esta declaración se especifica si el módulo debe conectarse mediante DHCP o a través de una IP fija. Si el servidor no se conecta a través de este parámetro, utilizará otra instancia para recibir la IP de conexión.

$Inicialip.- Esta instancia determina una dirección IP para el dispositivo de manera estática cuando el DHCP se encuentra deshabilitado.

$Downloadpassword [contraseña].- Con esta declaración se puede especificar una contraseña para poder descargar una página Web nueva hacia el servidor con el archivo de configuración, por lo tanto se deberá ingresar la contraseña si se desea cargar la página en el SITEPLAYER.

$Sitefile [Direccion de archivo binario SPB].- Con el uso de esta instancia se orienta la creación del archivo de configuración de manera binaria que es creado cuando se ha compilado y descargado el archivo de configuración .SPD en el servidor SITEPLAYER.

$Sitepath [Ruta del archivo de configuración].- Este declaración indica al servidor donde se encuentran los archivos que se deben cargar en el SITEPLAYER. En el se contemplara la carpeta donde se encuentre la página web.

Una vez compilados estos parámetros el usuario podrá ingresar a la http://IP del servidor/inicio.htm en el servidor y sus datos estarán dentro del servidor embebido. Como se ha observado en la figura anterior todos los pines del servidor se han configurado como salidas que se activan en bajo a una comunicación de 9600 que corresponde a una comunicación serial, y en este caso el puerto serial solo envía un byte.

3.2.3 Archivo SPD.

Con esta opción se definen los objetos, que contienen las variables utilizadas por el servidor SITEPLAYER. Dichas variables permitirán al servidor la comunicación y envío de datos u órdenes hacia el controlador. Las variables contenidas en dicho archivo tienen un nombre de pila, un tipo, junto con un tamaño y un valor predeterminado. Dichas variables serán los objetos del SitePlayer o llamados SiteObjects que comunicarán al controlador con el Servidor embebido.

Por lo tanto los objetos que sean declarados deben mantener los nombres de variables utilizadas en el código fuente de las páginas HTML. Los nombres de estos elementos pueden contener 32 caracteres de longitud, un particular es que no son sensibles al uso de mayúsculas o minúsculas con lo cual se pueden utilizar caracteres desde la A hasta la Z, o números desde el 0 hasta el 9, sin importa cual y con la condición de longitud.

Para identificar los objetos Web y el tipo de dato que se va a enviar el Siteplayer puede definir objetos de tipo:

Tabla 3.5. Tabla de datos de SitePlayer.

TIPO	CARACTERÍSTICA
BIT	Es un solo bit determinado.
BYTE	Es un byte o un string de 8 bits.
HEXADECIMAL	Característica 0 hasta F
PALABRA	Que representa hasta 16 bits
DOUBLE	Que representa hasta 32 bits

Se hace imprescindible que cada ubicación de los objetos se encuentre definido, por lo cual existe una directiva que le indica en que lugar de la memoria del servidor serán alojados los objetos, esta está definida por la etiqueta ORG. Para conocimiento el servidor embebido SitePlayer contempla espacios de memoria entre 0000h y 02FFh que son los intervalos definidos en el módulo.

3.2.4 Archivo SPB.

Cuando se instanciado los objetos y han sido utilizados en la definición dentro del archivo de configuración SPD y en la configuración de la página Web, el compilador procede a ensamblar un archivo individual que es que se carga en el módulo SitePlayer mediante la conexión Ethernet con el uso de un PC.

Dicho computador debe estar dentro de la misma red para que se pueda especificar la IP que se configurará en el servidor.

El programa que utiliza el servidor SitePlayer es el llamado SiteLinker, el mismo que es el que permite la compilación del archivo .SPD creando uno nuevo de tipo SPB que debe ser descargado al servidor, como dato adicional, cada vez que se realizan cambios sobre el archivo .SPD y se descarga al módulo SitePlayer, es reemplazado por el nuevo archivo de páginas Web que se haya generado.

3.2.5 Compilación y carga del Archivo SPB.

Cuando se ha ejecutado el programa de compilación SiteLinker, aparece una ventana que permite ejecutar ciertas instrucciones sobre el servidor, una de ellas y la primera es la de abrir el archivo de extensión .SPD que para aquello es necesario abrirlo mediante la opción OPEN y buscar el contenido donde se encuentra dicho archivo.

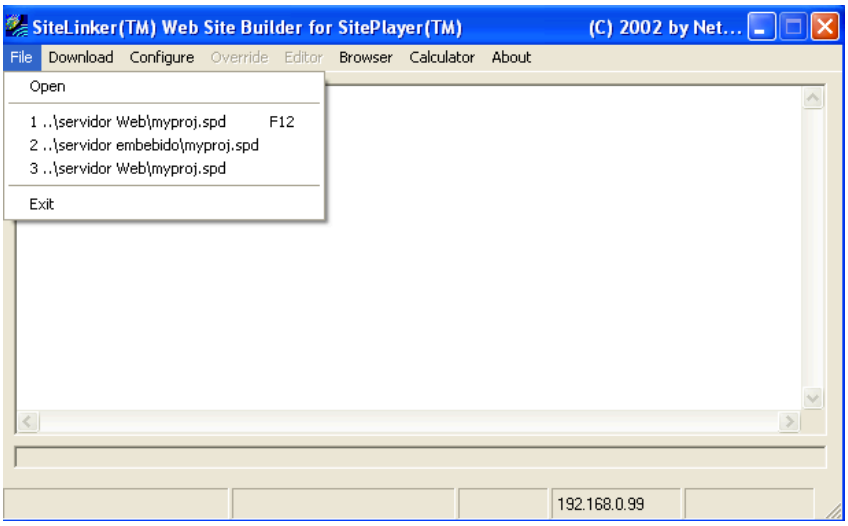


Figura. 3.36. Pantalla de compilación del SiteLinker.

Como se verifica en la imagen una vez que se realizado la apertura del archivo, es necesario descargarlo en el servidor, para lo cual se utiliza la opción del menú DOWNLOAD. Adicional se tienen otras opciones que una vez que es descargado, es necesario compilar el archivo .SPD para que se cree el archivo descargable mediante la opción MAKE DOWNLOAD.

Se puede utilizar una de rápida ejecución como lo es MAKE AND DOWNLOAD, que es la que realiza los procesos en una sola selección. Para que estos pasos funcionen es necesario identificar la IP que es perteneciente al módulo SitePlayer ya que de no ser así el archivo no será descargado y verificará un error sobre sí. Para poder ejecutar esto se utiliza la opción de IP ADDRESS FOR DOWNLOAD, que se encuentra en el menú de CONFIGURE. Para esto se verifica tal como se observa en la imagen.

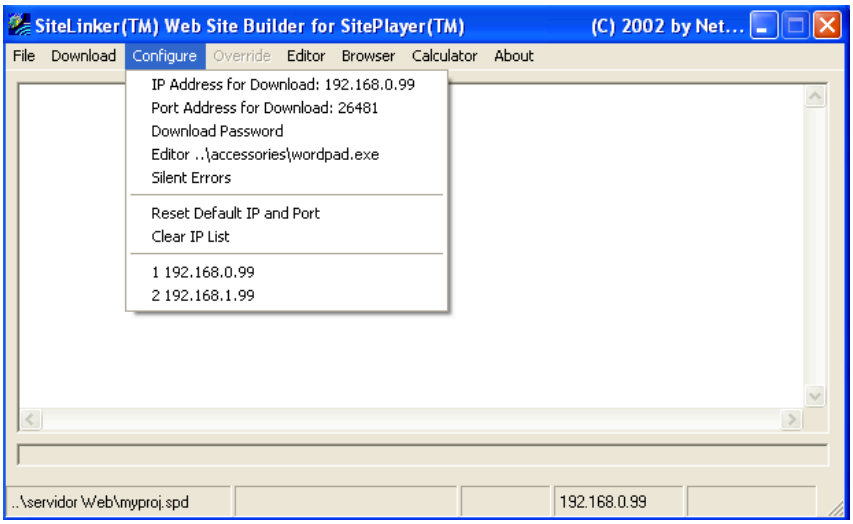


Figura. 3.37. Ingreso de la IP correspondiente a SITEPLAYER,

Esta dirección que se verifica en la imagen corresponde a la dirección IP del servidor y si es necesario su uso a través de un servidor Proxy, sería necesario colocar una excepción sobre el mismo para que permita la comunicación hacia el servidor. El comando DOWNLOADPASSWORD que se observa en el menú, define una contraseña para el SiteLinker, esta es la que bloquea la descarga de configuraciones hacia el módulo pues para proceder con el mismo sería necesario el ingreso de dicha clave.

Cuando el SiteLinker ha comenzado la descarga del archivo de configuración .SPD, se observa información de la descarga, pues después de realizar dicha carga el programa desplegará un mensaje de que se ha cargado el archivo dentro del servidor.

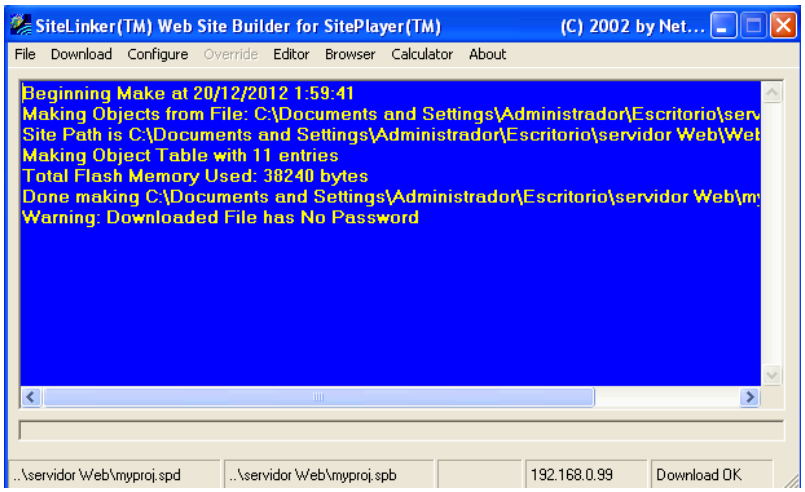


Figura. 3.38. Estado de la descarga de la configuración al SITEPLAYER.

Si todos los parámetros son los correctos, el programa termina la carga y muestra el valor de memoria que se ha utilizado para la carga. En la siguiente imagen se puede apreciar la correcta descarga del archivo hacia el servidor SitePlayer.

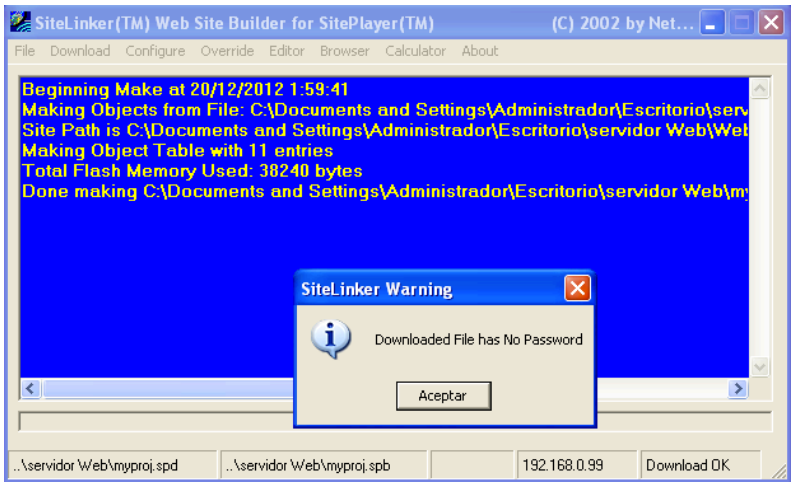


Figura. 3.39. Descarga completada al Servidor SITEPLAYER.

Una vez generados estos archivos, se utiliza el programa de SitePlayer Linker para permitir el acceso a la carpeta donde se encuentra toda la información de la página web para que esta pueda ser cargada en el servidor físicamente.

Este programa copia todo el archivo de configuración, así como tambien la pagina web y su contenido dentro de sí la misma que es enviada mediante una IP preconfigurada en el dispositivo de fábrica. Como acotación a dicha información, esta IP puede ser modificada a través del archivo de configuración del SitePlayer. Es necesario notar que la IP configurada en el dispositivo debe pertenecer a la misma red donde se pretende utilizar. El resultado se observa de la siguiente manera.

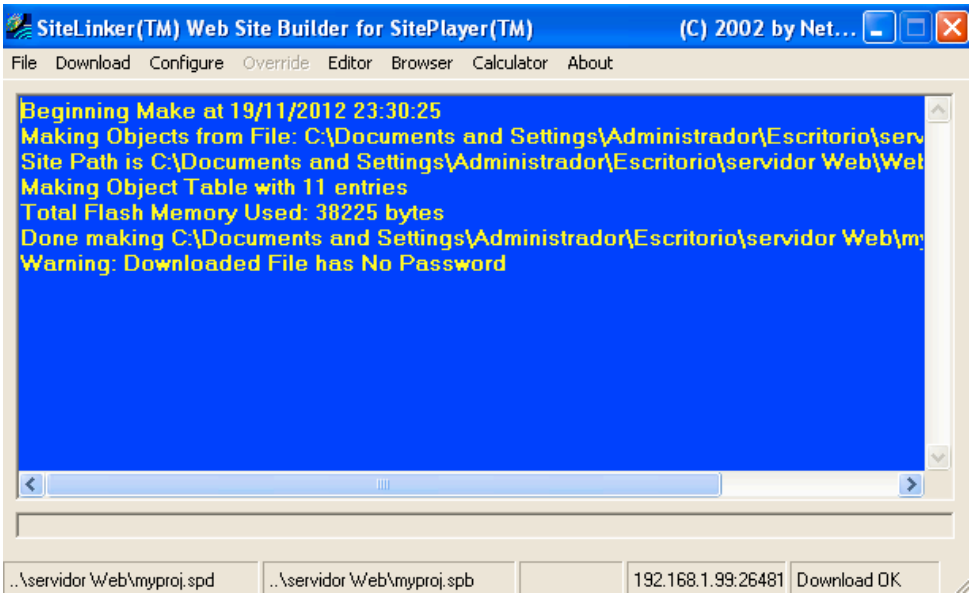


Figura. 3.40. Copia de configuración en dispositivo SITEPLAYER.

3.2.6 Diseño de aplicación gráfica en INTERFACE BUILDER XCODE.

Como se ha observado anteriormente en el diseño anterior era necesario ingresar una IP en el campo para poder acceder al aplicativo, y adicional a eso se debía ingresar una contraseña. Analizándolo quizás para una persona todo ese proceso lo vea como largo y no muy interactivo. Pero para facilitar este proceso a las personas con discapacidad motriz, es necesario otorgar las facilidades para el uso del sistema, por lo cual se realiza el diseño de una aplicación móvil para un IPhone para interactuar de manera fácil y óptima con el usuario.

Dicha interfaz viene otorgada por una aplicación que haya sido compilada y programada en una plataforma compatible con el dispositivo que permita al usuario el menor de indicaciones y a simple vista el usuario pueda manejarlo por iniciativa. Para el diseño de la aplicación se ha comenzado desde la estructura de la aplicación, para lo cual se presenta el siguiente esquema:

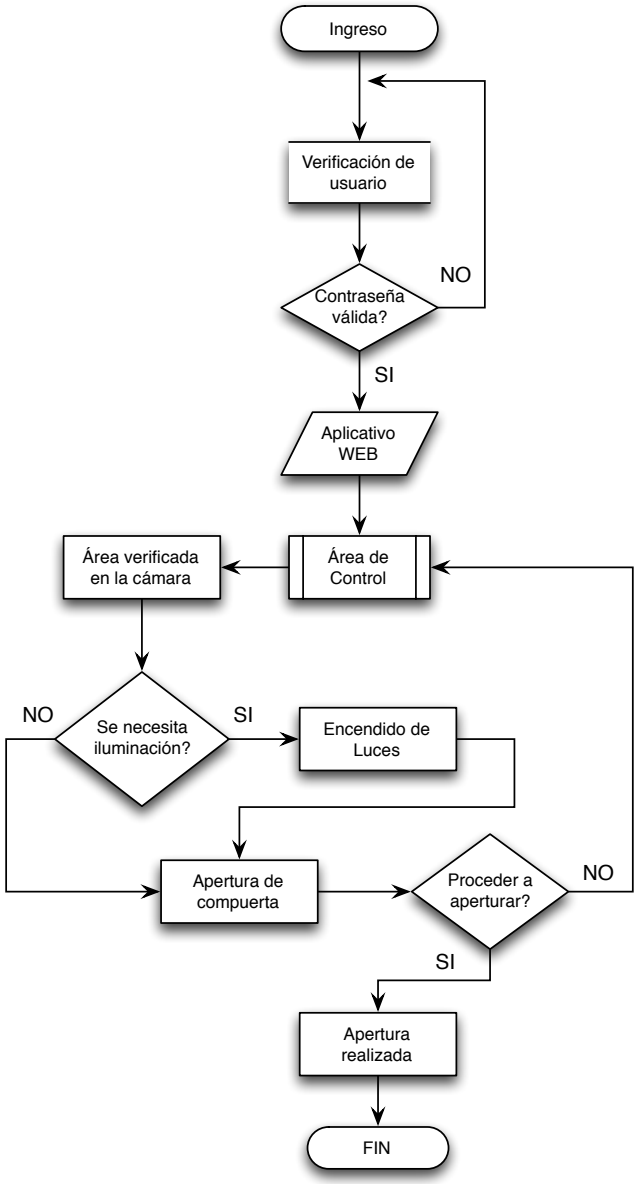


Figura. 3.41. Diagrama de flujo del proceso del aplicativo.

Como se observa en el esquema, como cara principal se tiene la portada de la aplicación la misma que tiene como objetivo informar el nombre del aplicativo, con un título y un botón de ingreso; desde esta se puede ingresar al portal del aplicativo como tal. Después en la siguiente ventana se presenta una descripción general del uso del programa, que de cierta manera ayuda al usuario a orientarse de forma rápida en el manejo del aplicativo.

Figura. 3.42. Portada Principal de la Aplicación móvil.

Contiene un botón de ingreso donde el usuario tiene la posibilidad de continuar hacia el aplicativo de control. Para la programación de dicho evento se tiene el objeto de tipo *UIViewController (Vista del dispositivo)* donde se han colocado el boton, el texto y el fondo de pantalla. El evento colocado en el botón para el ingreso a la siguiente ventana se coloca como *StoryBoard Segues* que es el seguimiento hacia la otra ventana que en este caso son las Indicaciones. Esto se lo realiza manteniendo presionado Control y click sobre el botón.

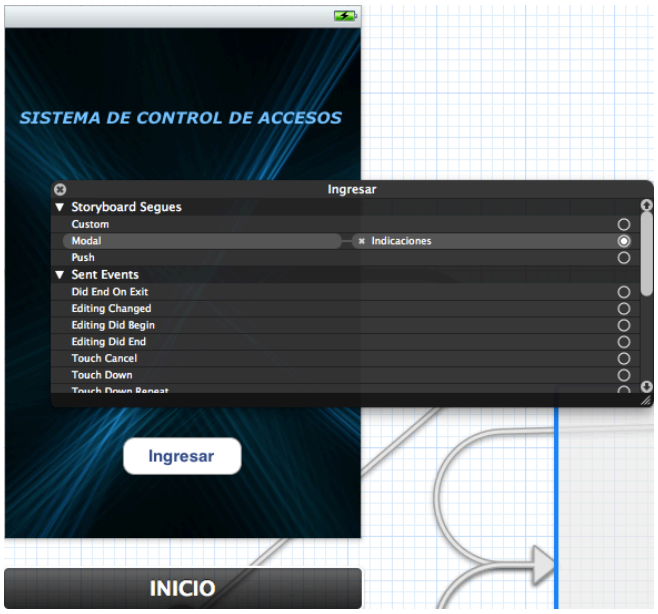


Figura. 3.43. Evento de acceso a Indicaciones.

Se selecciona la opción *modal* en el *Storyboard Segue* pues nada más son estilos de transición hacia la siguiente ventana hacia la siguiente. En la presente el usuario tiene intrucciones para el uso de la misma; posee información de las precauciones a tomar del caso. Una vez que estas sean leidas podrá ir a la aplicación.

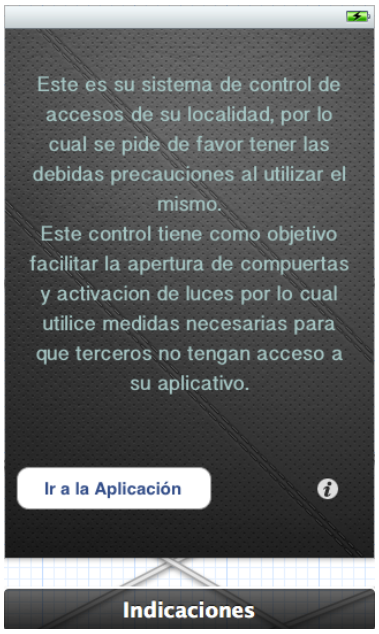


Figura. 3.44. Indicaciones de uso del aplicativo móvil.

En las indicaciones únicamente se utiliza texto para indicar las advertencias y un botón de acceso al aplicativo que hace referencia hacia la siguiente ventana que corresponde a las Indicaciones. En dicho botón se hace referencia al siguiente evento que es hacia la barra de menú donde se contemplan las áreas de control. No existe un contenido complejo para interactuar entre ventanas.

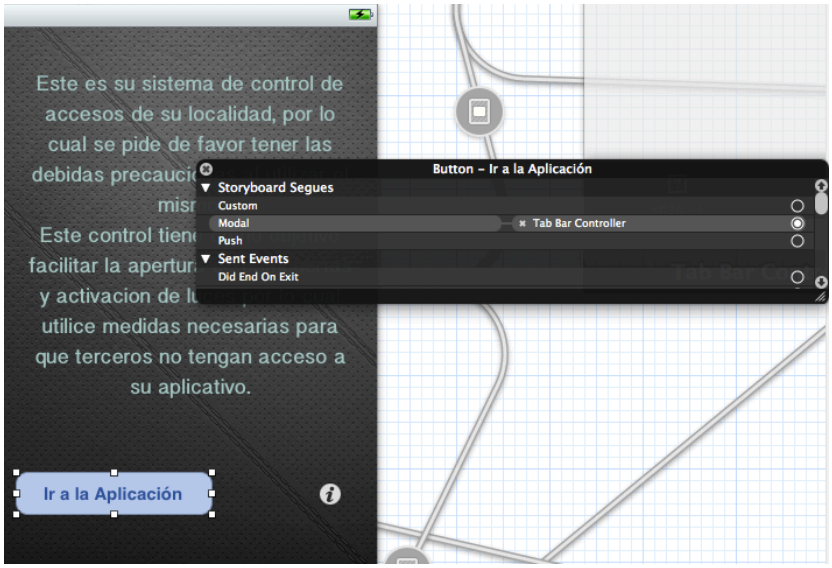


Figura. 3.45. Indicaciones para el ingreso al aplicativo.

Cuando se ha realizado el ingreso por completo a la aplicación de control, en ella se identifica un panel de control, tanto de accesos como del monitoreo de las cámaras en las diferentes áreas de acceso.

El usuario tiene la capacidad de controlar de manera efectiva dichos elementos para interactuar y ejecutar órdenes deseadas. Los eventos ejecutados hasta ahora son la transición de órdenes para el ingreso hacia la ventana principal donde está contenido el aplicativo.

Figura. 3.46. Entrada a 2 áreas de control.

Figura. 3.47. Componentes de cada área de control.

En las imágenes mostradas se hace referencia a un controlador de forma de barra de selección la misma que permite al usuario seleccionar el área que desea controlar. Cada uno de ellos hace referencia a diferentes objetos por lo cual se detalla a continuación la acción de cada uno de ellos. En la imagen que se presenta a continuación se aprecia los siguientes objetos con su interacción en el programa. Cada uno de estos botones y ventanas web realizan la acción indicada. Más adelante se detallarán los métodos para el programa en conjunto para poder identificar de mejor manera el mismo.

3.3 DISEÑO Y PROGRAMACIÓN EN LA PLATAFORMA XCODE

Como se ha observado anteriormente todos los objetos han sido colocados para lo cual es necesario empezar a instanciarlos con sus respectivas etiquetas y sus métodos de ejecución para obtener el resultado esperado. Para esto se definen el archivo de cabecera .H y el archivo de implementación .M que hará referencia a los objetos instanciados y llamará los métodos para su ejecución.

```
//
//  InicioViewController.h
//  Contaccess
//
//  Created by Juan Villalba on 27/07/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "ViewController.h"

@interface InicioViewController : ViewController <UIWebViewDelegate>
{
// Declaración de objeto para enviar via web las ordenes
    IBOutlet UIWebView *Web1;
    IBOutlet UIWebView *Camara1;
    IBOutlet UIWebView *Cam1Total;
// Declaración de punteros a vista de imagen.
    IBOutlet UIImageView *imagen1;
//Declaración de objeto para los switch on - off
    IBOutlet UISwitch *onswitch1;
}
@property (nonatomic, retain) UISwitch *onswitch1;
@property (nonatomic, retain) UIImageView *imagen1;
//En este caso no se hace referencia a la salida de objeto puesto que la imagen inicial estará impuesta.
@property (nonatomic, retain) UIImage *imag_on;
@property (nonatomic, retain) UIImage *imag_off;
//En este se hace referencia a la camara IP del area indicada
@property(nonatomic,strong)  IBOutlet UIWebView *Cam1Total;
@property(nonatomic,strong)  IBOutlet UIWebView *Camara1;

//Metodos Declarados
-(IBAction)ON1;
-(IBAction)Area1;
-(IBAction)Cam1;
@end
```

Figura. 3.48. Archivo de cabecera del Área #1.

```
//
//  Area2ViewController.h
//  Contaccess
//
//  Created by Juan Villalba on 25/09/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "ViewController.h"

@interface Area2ViewController : ViewController <UIWebViewDelegate>
{

    // Declaración de objeto para enviar via web las ordenes
    IBOutlet UIWebView *Web2;
    IBOutlet UIWebView *Camara2;
    IBOutlet UIWebView *Cam2Total;
    // Declaración de punteros a vista de imagen.
    IBOutlet UIImageView *imagen2;

    //Declaración de objeto para los switch on - off
    IBOutlet UISwitch *onswitch2;

}

@property (nonatomic, retain) UISwitch *onswitch2;
@property (nonatomic, retain) UIImageView *imagen2;

//En este caso no se hace referencia a la salida de objeto puesto que la imagen inicial estará impuesta.
@property (nonatomic, retain) UIImage *imag_on;
@property (nonatomic, retain) UIImage *imag_off;
@property(nonatomic,strong)  IBOutlet UIWebView *Cam2Total;
@property(nonatomic,strong)  IBOutlet UIWebView *Camara2;
//Alertas

//@property (nonatomic, retain) UIAlertView *advertencia2;

-(IBAction)ON2;
-(IBAction) Area2;
-(IBAction)Cam2;
-(IBAction)Cam2total;

@end
```

Figura. 3.49. Archivo de cabecera del Área #2.

En la presente imagen se puede apreciar los objetos que han sido declarados en el archivo de cabecera siendo estos tanto para el área #1 como el área 2 de manera similar los que interactuarán con el servidor embebido y con las cámaras.

Figura. 3.50. Acción de los objetos del Aplicativo.

Como se observan las instancias se hace referencia a los objetos declarados en la pantalla principal cada uno de ellos para uso de los métodos que se inicializarán en el archivo de implementación.

Estos archivos de cabecera contienen las configuraciones y métodos para los objetos ubicados en la pantalla. Para un mejor entendimiento se muestra el archivo. Cabe recalcar que tanto para un área como para la otra, el código es similar.

3.3.1 Inicio del archivo de implementación.

```
Contaccess
1 target, iOS SDK 5.1
  Contaccess
    AppDelegate.h
    AppDelegate.m
    MainStory...toryboard
    MainStory...toryboard
    ViewController.h
    ViewController.m
    InicioViewController.h
    InicioViewController.m
    Area2ViewController.h
    Area2ViewController.m
    Supporting Files
  Frameworks
  Products
    Contaccess.app
```

```
//
//  InicioViewController.m
//  Contaccess
//
//  Created by Juan Villalba on 27/07/12.
//  Copyright (c) 2012 __MyCompanyName__. All rights reserved.
//

#import "InicioViewController.h"
@interface InicioViewController ()

@end

@implementation InicioViewController

// Variables de switch

@synthesize onswitch1;
@synthesize imagen1;
@synthesize imag_on,imag_off;
@synthesize Cam1Total;
@synthesize Camara1;

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    // Carga de imagen inicial en portada de Area 1 y Area 2.
    self.imag_on = [UIImage imageNamed:@"bot1.png"];
    self.imag_off = [UIImage imageNamed:@"bot2.png"];

    NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
    NSURLRequest *CargURL = [NSURLRequest requestWithURL:Dire];
    [self.Camara1 loadRequest:CargURL];
    [self.Cam1Total loadRequest:CargURL];

    [super viewDidLoad];
    // Do any additional setup after loading the view.
}
```

Figura. 3.51. Inicialización de variables y presentación.

Como se observa en la imagen se utiliza *@synthesize* para declarar los objetos. Una vez que han sido instanciado cada uno de ellos se procede a detallar la estructura del programa. En el podemos apreciar el método *void* que contempla la carga inicial de la ventana del aplicativo como son las imágenes de los botones, y el acceso y carga de la URL de la cámara.

Todo este método tanto para la un área como para la otra. Cabe recalcar que el programa detalla de por sí como realiza la carga de la Web en el *UIWebView Camara 1* como se había observado en la imagen de la ventana de menú.

Este método únicamente permite inicializar la vista de la ventana de menú, por lo cual es el primer paso para el uso de las prestaciones de la aplicación, por este motivo se muestra a continuación el método de control de luces con el siguiente código.

3.3.1.1 Encendido de iluminación.

```
- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
}

-(IBAction)ON1{
    if (onswitch1.on)
    {
    //En esta se carga la orden de encendido del foco N1.
        NSURL *Dir = [NSURL URLWithString:@"http://192.168.0.99/myproj.spi?io2=0"];
        NSURLRequest *CargarURL = [NSURLRequest requestWithURL:Dir];
        [Web1 loadRequest: CargarURL];

        NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
        NSURLRequest *CargaURL = [NSURLRequest requestWithURL:Dire];
        [Camara1 loadRequest: CargaURL];

    // Carga de imagen referencia del foco N1 en imageview foco1 la imagen1.
        self.imagen1.image = self.imag_on;

        // Crearemos una alerta para mostrar el saludo al usuario.
        UIAlertView *alerta1 = [[UIAlertView alloc]
                                initWithTitle:@"ADVERTENCIA"
message:@"Se ha encendido el foco del Area 1!! Verifica mediante la imagen que esta ha sido encendida sino intentalo nuevamente"
                                delegate:nil
                                cancelButtonTitle:@"ACEPTAR"
                                otherButtonTitles:nil,
                                nil];
        [alerta1 show];
    }
    else
    {
        NSURL *Dir = [NSURL URLWithString:@"http://192.168.0.99/myproj.spi?io2=1"];
        NSURLRequest *CargarURL = [NSURLRequest requestWithURL:Dir];
        [Web1 loadRequest: CargarURL];
        self.imagen1.image = self.imag_off;

        NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
        NSURLRequest *CargaURL = [NSURLRequest requestWithURL:Dire];
        [Camara1 loadRequest: CargaURL];
    }
}
```

Figura. 3.52. Encendido y apagado de iluminación.

En este caso se tiene el código para encendido de luces, que de igual manera corresponde a las direcciones IP configuradas en el servidor SITEPLAYER para enviar las órdenes y cargar las URL determinadas. Adicional como se observa el objeto, corresponde a un Switch el mismo que enciende y apaga. Adicional al ejecutar el Switch un mensaje aparecerá en la pantalla para advertir que luces han sido activadas. La alertas enviadas permiten mantener una pausa en la selección de la opción requerida para que de esta manera el usuario se encuentre seguro de la opción a elegir. Y si caso contrario el usuario cancela la alerta, nuevamente es redireccionado a la ventana del aplicativo sin generar efecto alguno. Cabe recalcar que la cámara y las salidas del Servidor embebido están en constante uso y actualizando.

Las luces se activan con el solo movimiento del Switch, si es encendido cambia la imagen que hace referencia al foco y a su vez envía la URL de control de una de las salidas del Servidor embebido para este a su vez activar luminosidad en un foco físico.

3.3.1.2 Acceso a cámaras y advertencia.

Figura. 3.53. Acceso a Cámara IP y Advertencia.

Como se observa en el acceso a la cámara utiliza un puntero que hace referencia a la página Web del Servidor SITEPLAYER y eventualmente carga el pedido de la URL para poder acceder a la cámara IP. Tanto para el acceso a cámaras como para ejecución de salidas del aplicativo se lo realiza con URL's que contiene el acceso a la cámara o ejecución del servidor y la salida solicitada.

El método que se observa es *Cam1* y es el que permite cargar la página web con la monitorización de la cámara. El método declarado para el *Area1* tiene como objetivo en primera instancia alertar al usuario si se desea realizar la apertura. El usuario tiene la posibilidad de garantizar la operación o de cancelarla dependiendo del criterio y uso de la misma.

Para lo cual al seleccionar el área para la apertura, se ejecutará un segundo método que estará contenido dentro de una segunda alerta para indicar que se ha permitido el acceso al área en gestión.

3.3.1.3 Apertura de compuertas.

```
-(void)alertView:(UIAlertView *) alertView
clickedButtonAtIndex:(NSInteger)ButtonIndex
{
    NSString *title = [alertView buttonTitleAtIndex:ButtonIndex];
    if ([title isEqualToString:@"Cancelar"])
    {
        NSURL *Dir = [NSURL URLWithString:@"http://192.168.0.99/myproj.spi?io0=1"];
        NSURLRequest *CargarURL = [NSURLRequest requestWithURL:Dir];
        [Web1 loadRequest: CargarURL];

        NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
        NSURLRequest *CargaURL = [NSURLRequest requestWithURL:Dire];
        [Camara1 loadRequest: CargaURL];
    }
    else if([title isEqualToString:@"Garantizar"])
    {
        NSURL *Dir = [NSURL URLWithString:@"http://192.168.0.99/myproj.spi?io0=0"];
        NSURLRequest *CargarURL = [NSURLRequest requestWithURL:Dir];
        [Web1 loadRequest: CargarURL];

        UIAlertView *open1 = [[UIAlertView alloc]
                              initWithTitle:@"GARANTIZADA"
                              message:@"El acceso se concedió satisfactoriamente al Area #1"
                              delegate:self
                              cancelButtonTitle:@"Acepto"
                              otherButtonTitles:nil,
                              nil];

        [open1 show];

        NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
        NSURLRequest *CargaURL = [NSURLRequest requestWithURL:Dire];
        [Camara1 loadRequest: CargaURL];


    }

    else if([title isEqualToString:@"Acepto"])
    {
        NSURL *Dir = [NSURL URLWithString:@"http://192.168.0.99/myproj.spi?io0=1"];
        NSURLRequest *CargarURL = [NSURLRequest requestWithURL:Dir];
        [Web1 loadRequest: CargarURL];

        NSURL *Dire = [NSURL URLWithString:@"http://192.168.0.248/image/jpeg.cgi"];
        NSURLRequest *CargaURL = [NSURLRequest requestWithURL:Dire];
        [Camara1 loadRequest: CargaURL];
    }
}
@end
```

Figura. 3.54. Código para apertura de compuertas.

Cada uno de los objetos ejecuta un método, dependiendo si es una variable o si es el resultado de una salida como se observa el tipo *IBOutlet* que hace referencia al contenido Web del objeto *UIWebView*. En este se reflejará la imagen de la cámara tal como se observa en el nombre de la variable.

El seguimiento final de cada uno de estos eventos se identifica por la plantilla llamada StoryBoard que contiene todos los objetos que interactuarán en el aplicativo como se observa en la figura.

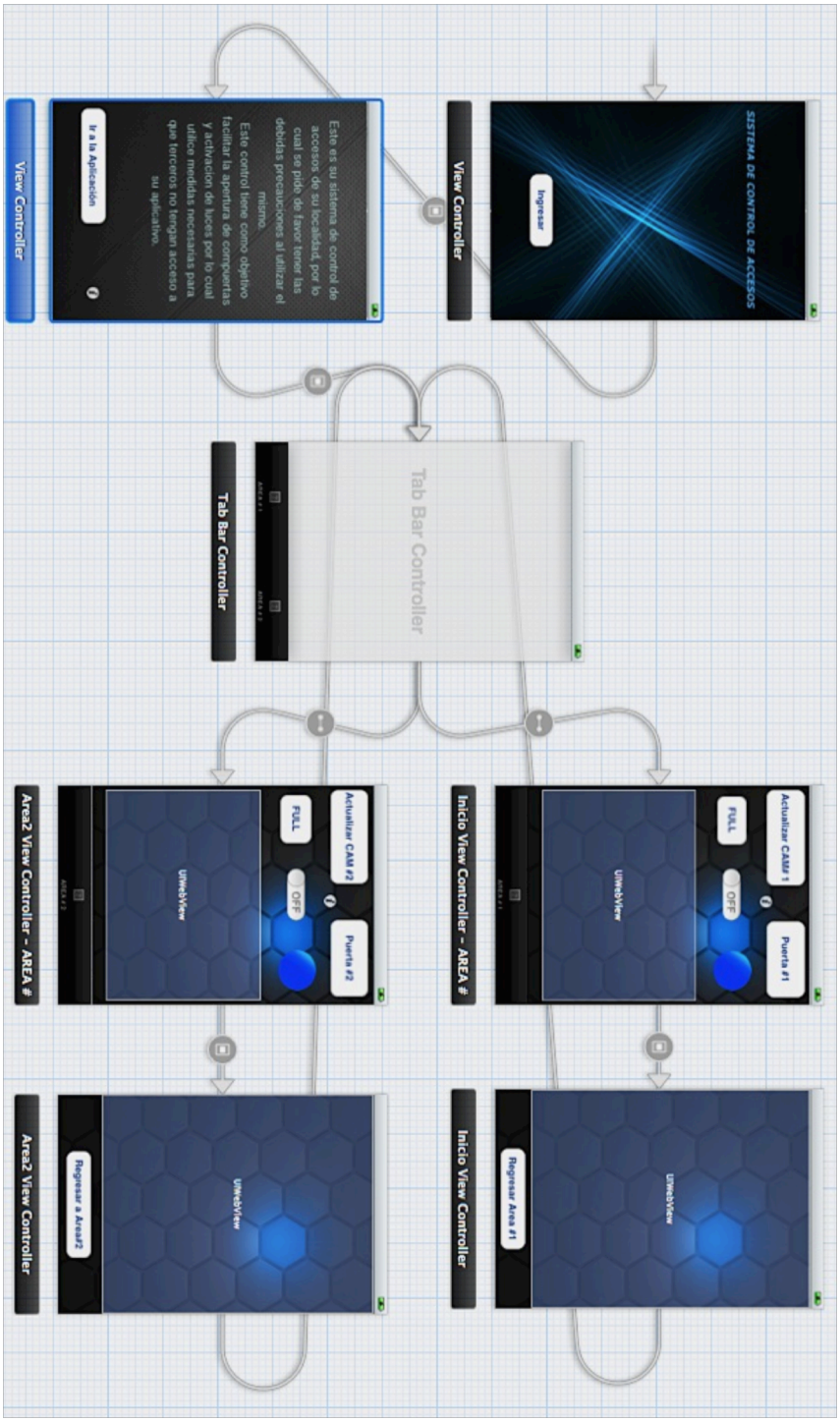


Figura. 3.55. StoryBoard del aplicativo.

En ella se observan flechas de guía que son colocadas por el programador para que la plataforma identifique la secuencia de ejecución de las mismas.

3.4 COSTO DEL PROYECTO.

Para el presente proyecto el costo de materiales utilizados, se presenta a continuación. Cada uno de ellos corresponden al valor comercial indicado.

ELEMENTO	CANTIDAD	VALOR UNITARIO	COSTO
Servidor SITEPLAYER	1	25.00	25.00
Optoacoplador MOC 3031	4	0.30	0.30
Triac BT136	4	0.60	2.40
Resistencia 1Kohm	1	0.03	0.03
Cristal	1	0.12	0.12
Borneras	11	0.15	1.65
PIC 18F2550	1	12.00	12.00
Resistencia 100ohm	4	0.03	0.12
Resistencia 200ohm	8	0.03	0.24
Capacitor 22pF	2	0.03	0.06
Capacitor 0.01uF	1	0.03	0.03
Entrada adaptador	1	0.50	0.50
Mac Jack RJ45	1	3.50	3.50
Baquela	2	5	
		COSTO TOTAL:	50.95

Dichos materiales fueron utilizados para la elaboración de este proyecto.

CAPITULO 4

PRUEBAS Y RESULTADOS EXPERIMENTALES

CAPITULO 4

PRUEBAS Y RESULTADOS EXPERIMENTALES

4.1 PRUEBAS Y ANALISIS DE COMUNICACIÓN

En las siguientes pruebas se verifica conectividad del sistema tanto con el aplicativo Web, así como también el aplicativo móvil. En dichos análisis se verifica estado del enlace y tiempos de conectividad de la misma.

4.1.1 Pruebas de ingreso al sistema.

Para poder proceder con las prueba se utilizarán ambos aplicativos para verificar que ambos funcionen de la misma manera, para lo cual identificaremos que realicen la comunicación de forma óptima hacia el servidor.

El ingreso al sistema a través del navegador se lo realiza con la URL: http://192.168.0.99 que es la IP del servidor. Para lo cual se procederá a ingresar con ambos métodos al servidor Embebido y debería permitir el acceso al aplicativo.

Figura. 4.1. Comparativa de ingresos al Aplicativo.

En esta captura de tráfico utilizando la herramienta WIRESHARK se verifica que al momento de ingresar a través de un navegador existe la comunicación de una máquina con IP: 192.168.0.234 con el servidor embebido correspondiendo a la IP: 192.168.0.99 que es de manera correcta

No.	Time	Source	Destination	Protocol	Length	Info
7	6.160689000	192.168.0.234	192.168.0.99	TCP	78	54422 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=901533769 TSecr=0 SACK_PERM=1
8	6.164003000	192.168.0.99	192.168.0.234	TCP	64	http > 54422 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
9	6.164120000	192.168.0.234	192.168.0.99	TCP	54	54422 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
10	6.164816000	192.168.0.234	192.168.0.99	HTTP	378	GET / HTTP/1.1
11	6.197038000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
12	6.361737000	192.168.0.234	192.168.0.99	TCP	54	54422 > http [ACK] Seq=325 Ack=1281 Win=65535 Len=0
13	6.392843000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
14	6.563011000	192.168.0.234	192.168.0.99	TCP	54	54422 > http [ACK] Seq=325 Ack=2561 Win=65535 Len=0
15	6.566884000	192.168.0.99	192.168.0.234	HTTP	70	Continuation or non-HTTP traffic
16	6.566952000	192.168.0.234	192.168.0.99	TCP	54	54422 > http [ACK] Seq=325 Ack=2578 Win=65535 Len=0
17	6.567292000	192.168.0.234	192.168.0.99	TCP	54	54422 > http [FIN, ACK] Seq=325 Ack=2578 Win=65535 Len=0
18	6.572329000	192.168.0.99	192.168.0.234	TCP	64	http > 54422 [ACK] Seq=2578 Ack=326 Win=1460 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Figura. 4.2. Tráfico de comunicación a SITEPLAYER.

En dicho ingreso se verifican protocolos de conexión de tipo TCP y de tipo HTTP para la visualización del aplicativo Web de manera orientada a la conexión. Si no existe comunicación entre equipos no se puede identificar este tipo de tráfico de datos.

Cabe recalcar que como se ha especificado que la memoria del equipo SITEPLAYER para almacenamiento es mínima, el ancho de banda de conexión de datos de igual forma es mínima, y como es de uso personal dentro de una intranet, el uso de este equipo no es de mayor consumo para todo el ancho de banda disponible. En primera instancia el aplicativo Web ya ha ingresado directamente al servidor, mientras que el aplicativo del IPhone tiene su presentación contenida en la memoria interna del dispositivo móvil, por lo cual aún no se puede verificar la conexión al servidor. Se puede comparar que la pantalla de presentación en el dispositivo móvil es inmediata, mas en el navegador Web toma cierto tiempo hasta que la página se cargue.

4.1.2 Verificación de entrada.

En la siguiente prueba se verifica el ingreso a la siguiente etapa que en este caso solo el servidor Web cuenta con un ingreso por contraseña por seguridad de uso del sistema ya que por parte del dispositivo móvil, se considera que el uso del aplicativo es utilizado por parte del propietario del mismo.

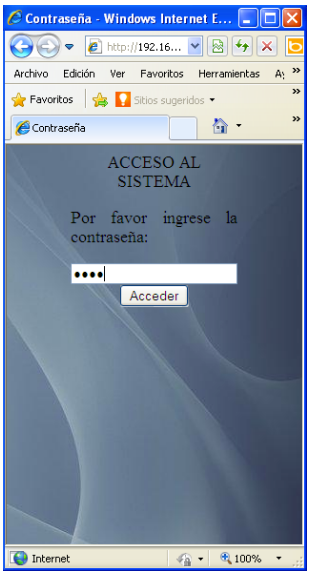


Figura. 4.3. Contraseña de ingreso al aplicativo.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.234	192.168.0.99	TCP	78	54424 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=901660129 TSecr=0 SACK_PERM=1
2	0.003333000	192.168.0.99	192.168.0.234	TCP	64	http > 54424 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.003599000	192.168.0.234	192.168.0.99	TCP	54	54424 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.004817000	192.168.0.234	192.168.0.99	HTTP	424	GET /Contrasena.html HTTP/1.1
5	0.037351000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
6	0.200674000	192.168.0.234	192.168.0.99	TCP	54	54424 > http [ACK] Seq=371 Ack=1281 Win=65535 Len=0
7	0.205622000	192.168.0.99	192.168.0.234	HTTP	120	Continuation or non-HTTP traffic
8	0.205656000	192.168.0.234	192.168.0.99	TCP	54	54424 > http [ACK] Seq=371 Ack=1348 Win=65535 Len=0
9	0.205956000	192.168.0.234	192.168.0.99	TCP	54	54424 > http [FIN, ACK] Seq=371 Ack=1348 Win=65535 Len=0
10	0.211014000	192.168.0.99	192.168.0.234	TCP	64	http > 54424 [ACK] Seq=1348 Ack=372 Win=1460 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Figura. 4.4. Tráfico de datos en Contraseña.

Nuevamente se verifica el aplicativo Web con conexión directa al servidor, en el cual solicita la contraseña de ingreso al sistema, y únicamente colocando los caracteres en el campo y presionando acceder, se puede ingresar al aplicativo contenido en el servidor Web Embebido. Dicha contraseña le otorga seguridad al aplicativo Web, a diferencia que al aplicativo móvil el usuario solo debe ingresar directamente sin problema alguna.

Figura. 4.5. Indicaciones previas al uso del aplicativo móvil.

Adicional cabe recalcar que en este caso, el aplicativo Web es más vulnerable a ataques de red, puesto que tanto URL como configuraciones podrían ser vistas por terceros. A diferencia que el aplicativo móvil mantiene a espaldas las configuraciones y URL que utiliza lo que le convierte en un sistema estable y seguro sin necesidad de una contraseña, su vulnerabilidad radica en la persona que lo utilice.

Como se mencionó anteriormente este es de uso exclusivo del usuario puesto que puede acceder sin permiso alguno al sistema de control.

4.1.3 Ingreso al sistema.

Es este caso ya se puede verificar que ambos aplicativos cuentan con el acceso a la imagen de la cámara IP. Sin embargo el aplicativo móvil recién en este momento se encuentra dentro del Servidor Web con lo que se corroboraría el acceso y funcionamiento al mismo. Se verifica en el aplicativo móvil que en este se ha separado las áreas pues, para la persona que utilice el sistema tendrá varias opciones y facilidades para el control de los actuadores.

De igual forma en el aplicativo Web no se encuentran separadas las áreas y mantiene de la misma manera un control simple de las acciones que se puedan generar al momento de activar salidas en el Servidor.

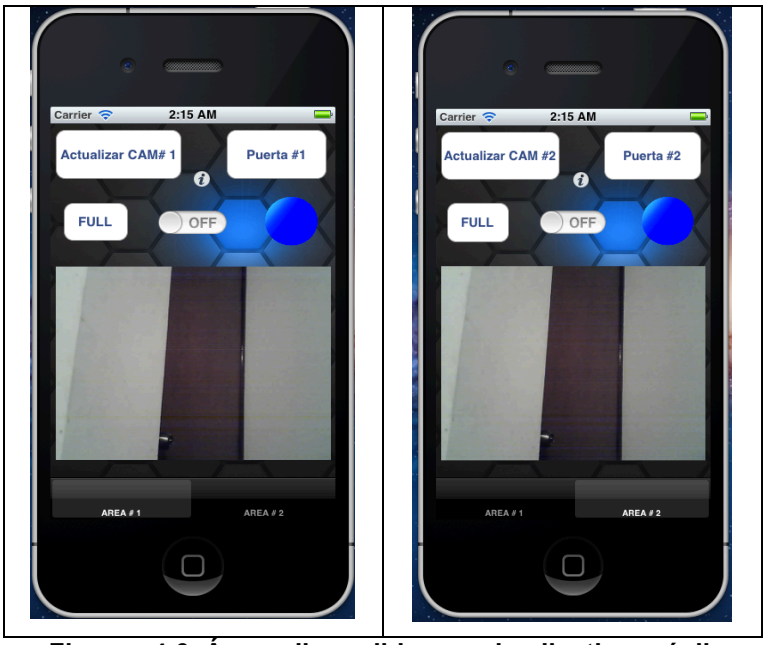


Figura. 4.6. Áreas disponibles en el aplicativo móvil.

Debido al espacio que se considera para distribuir en el móvil no es preciso mantener ambos monitores en una sola pantalla pues estas a su vez tienen en cada una de ellas botones para el accionamiento de luces y compuertas que deben mantenerse dentro de la vista.

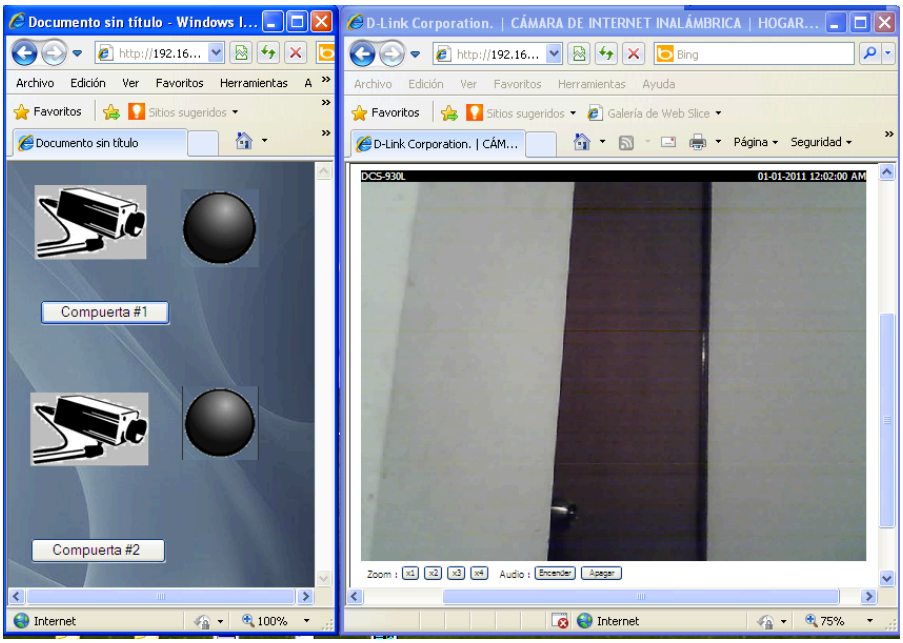


Figura. 4.7. Áreas disponibles en el aplicativo WEB.

No.	Time	Source	Destination	Protocol	Length	Info
5	2.680046000	192.168.0.234	192.168.0.99	TCP	78	54425 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=901735292 TSecr=0 SACK_PERM=1
6	2.683354000	192.168.0.99	192.168.0.234	TCP	64	http > 54425 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
7	2.683433000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
8	2.684031000	192.168.0.234	192.168.0.99	HTTP	432	GET /1234.htm HTTP/1.1
9	2.716473000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
10	2.879291000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=1281 Win=65535 Len=0
11	2.910492000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
12	3.081029000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=2561 Win=65535 Len=0
13	3.112169000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
14	3.204440000	192.168.0.1	224.0.0.12	IGMPv2	60	Membership Report group 224.0.0.12
15	3.282324000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=3841 Win=65535 Len=0
16	3.313540000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
17	3.483586000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=5121 Win=65535 Len=0
18	3.514772000	192.168.0.99	192.168.0.234	HTTP	1334	Continuation or non-HTTP traffic
19	3.683938000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=6401 Win=65535 Len=0
20	3.716070000	192.168.0.99	192.168.0.234	HTTP	991	Continuation or non-HTTP traffic
21	3.716145000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [ACK] Seq=379 Ack=7339 Win=65535 Len=0
22	3.716619000	192.168.0.234	192.168.0.99	TCP	54	54425 > http [FIN, ACK] Seq=379 Ack=7339 Win=65535 Len=0
23	3.721509000	192.168.0.99	192.168.0.234	TCP	64	http > 54425 [ACK] Seq=7339 Ack=380 Win=1460 Len=0 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Figura. 4.8. Tráfico de ingreso al sistema.

4.2 PRUEBAS Y RESULTADOS DEL SISTEMA

A continuación se realizan pruebas de uso del sistema en ambos aplicativos para observar que los resultados sean los deseados y sean los mismos. Para lo cual se procede a validar el funcionamiento de ambas aplicaciones.

4.2.1 Pruebas de visualización de cámara.

En cuanto a la monitorización de la cámara el aplicativo Web y la aplicación móvil mantienen una comunicación continua de tal manera que se mantiene la imagen en el equipo.

Es necesario indicar que en el aplicativo Web este monitor es en tiempo real con protocolos Java y ActiveX, sin embargo para el móvil IPhone este monitor es a través de la imagen de la cámara siendo esta de propiedad intelectual del fabricante, sin tener acceso libre a mediante un protocolo de conexión en recepción de formato de video compatible con el móvil.

Motivo por el cual es factible la actualización de la imagen para mantener cada cierto tiempo la gestión de la cámara a en el móvil.

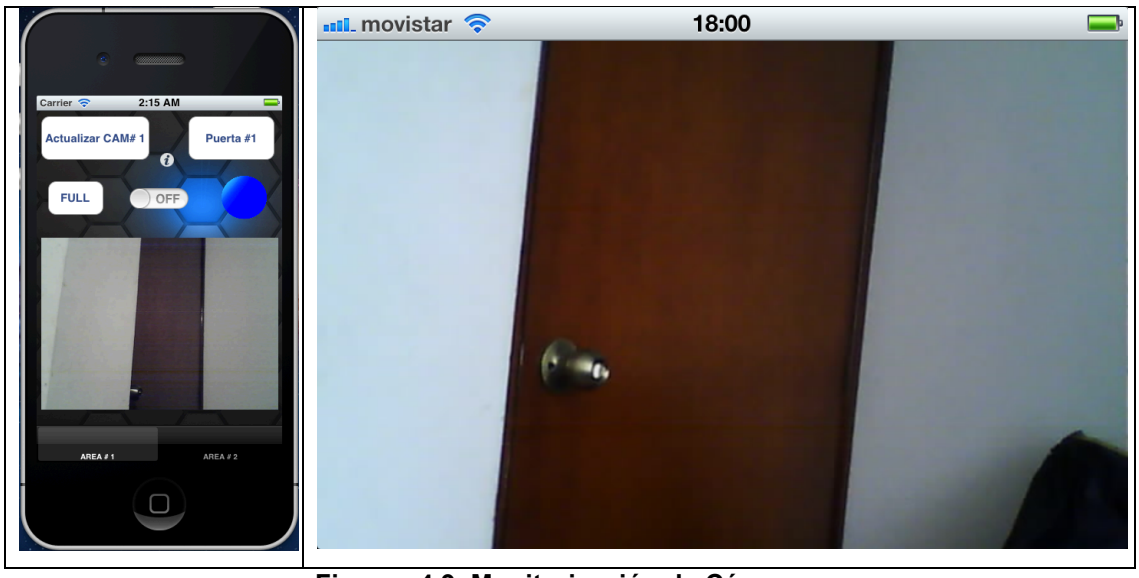


Figura. 4.9. Monitorización de Cámaras.

En la cámara del dispositivo móvil IPhone se puede apreciar la calidad de la imagen, tanto en el área de control, así como también de la opción Full para poder visualizar la misma en pantalla completa. El aplicativo permite al usuario utilizar dichas opciones para monitorizar la misma.

4.2.2 Pruebas de activación de iluminación.

La activación y desactivación de luminosidad en un área dependerá del usuario cual es el área que desea visualizar, ya que esta acción se la ha implementado con el objetivo de que la persona que lo utiliza, al ver dificultad para observar los objetos en la oscuridad encienda luces en el área y pueda verificar de manera óptima la persona en el área.

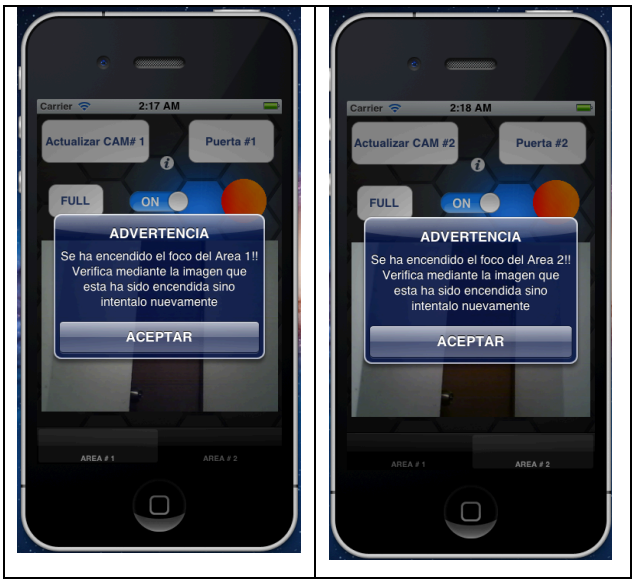


Figura. 4.10. Activación de luminosidad desde IPhone.

Para la activación de iluminación desde el dispositivo móvil únicamente se activa el Switch lo que envía la señal al servidor embebido y una vez ejecutado arroja una advertencia indicando que se ha encendido la luz en el área indicada. De esta manera el usuario está al tanto de lo que sucede con las áreas de monitorización.

En el aplicativo Web se mantiene la comunicación entre el equipo del usuario y el servidor Web, mientras que en el aplicativo móvil únicamente envía las órdenes al Servidor en espera de la tarea ejecutada para notificar al usuario.

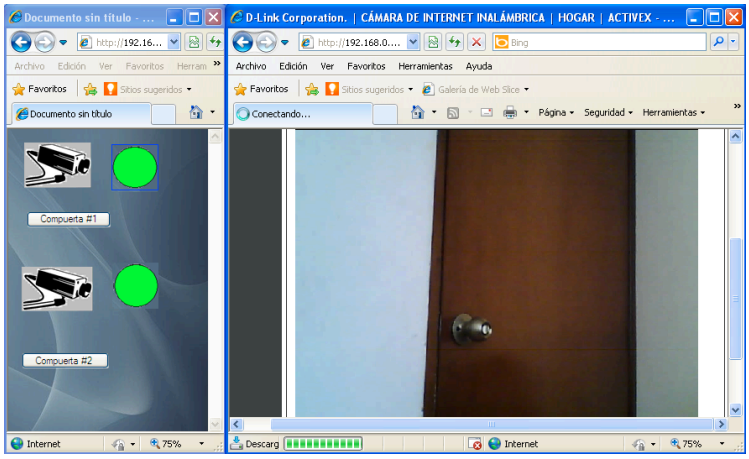


Figura. 4.11. Activación de luminosidad a través del Aplicativo Web.

4.2.3 Pruebas de apertura de compuerta.

Las pruebas realizadas para la apertura de compuertas está dada de manera simple con una advertencia para proceder con la activación del puerto de salida del controlador hacia la etapa de potencia y este a su vez al actuador que hace la apertura de la compuerta. Se realiza la apertura de las mismas, el aplicativo notifica al usuario si desea realizar la apertura, caso contrario cancelarla.

Figura. 4.12. Apertura de compuertas a través de IPhone.

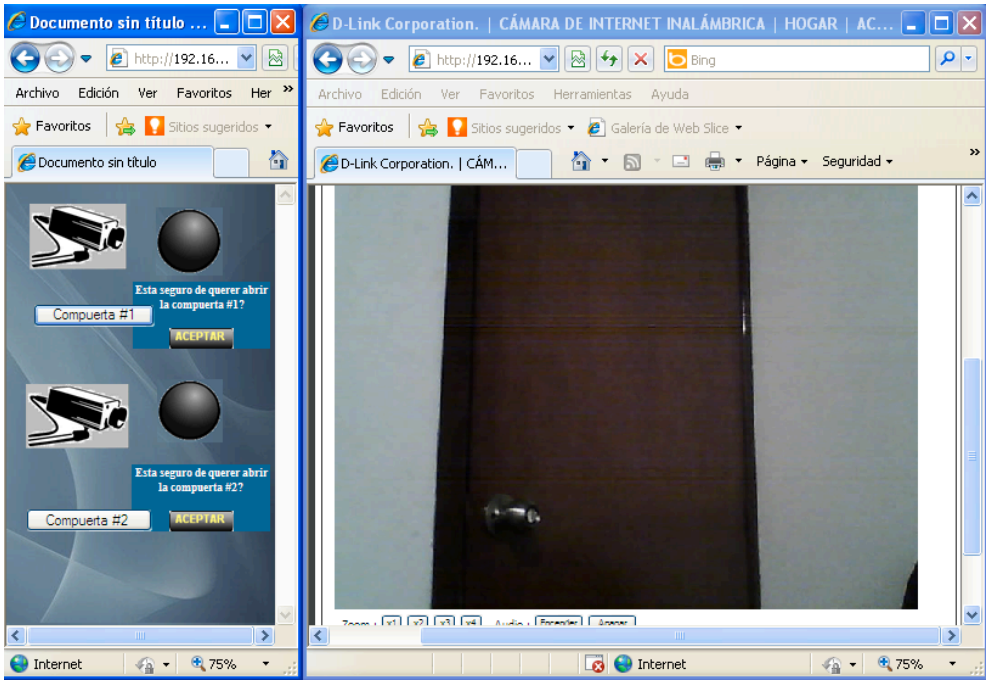


Figura. 4.13. Apertura de compuerta desde aplicativo Web.

Como se han observado en las pruebas ambas interfaces proporcionan un control seguro dentro del entorno donde son aplicadas, es decir tanto de manera móvil como desde un navegador. Cabe recalcar que los tiempos de respuesta son inmediatos pues no requieren de un ancho de bando muy amplio para aplicarlo en la ejecución de órdenes. Sin embargo para el monitoreo de cámaras dependerá de la transmisión y captura de los paquetes en la velocidad y ancho de banda que ocupen del canal. Dicha recepción no debe interferir pues la transmisión en un equipo inalámbrico pues estos hoy en día sobrepasan los 50MBits/seg.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Con el desarrollo del presente trabajo se ha conseguido realizar un sistema de control de accesos y monitorización para uso de la personas y personas especiales que poseen un discapacidad motriz, que brinda confort y seguridad a costos relativamente bajos manteniendo el orden en sus hogares.

- El servidor web embebido SITEPLAYER, posee entradas y salidas de tipo digitales, pero dada las exigencias en el proyecto se utilizó programación lógica que en este caso se lo realizó en un microcontrolador en este caso el PIC18f2550 puesto que se necesitó salidas con tiempos de temporización para los accesos.

- El realizar 2 aplicativos permite mantener la generalidad de uso entre dispositivos, de tal manera que ningún usuario se sienta condicionado al uso de un solo equipo móvil. Sin embargo cabe recalcar que es mucho mejor si se utiliza el equipo indicado que en este caso fue un IPhone.

- Una aplicación nativa en un dispositivo móvil resulta ser más eficiente que un aplicativo Web, pues la navegación a través del mismo es transparente para el usuario, y es gestionado a través de botones y métodos mas no por accesos de páginas web.

- Tanto como el servidor y el controlador, por construcción de circuitos interno, mantienen restricciones de alimentación, tanto en voltaje como en corriente. Con una limitante de 5 [V] y desde 300 hasta 500 mA de corriente para que este funcione de manera óptima, ya que de no ser así pueden existir pérdida de datos y otros errores de ejecución. En cambio si son potencias muy elevadas y estas están contempladas dentro de la parte digital, ella puede causar que se quemen los elementos sensibles, ya sea incluso el módulo SITEPLAYER o el controlador PIC.

- La respuesta de activación en el caso web es un poco más lenta que la enviada directamente del dispositivo móvil. Pues la capacidad de procesamiento de dicho requerimiento atraviesa por protocolos de conexión, mientras que el aplicativo solo ejecuta y envía.

- Cabe recalcar que las personas con discapacidad motriz necesitan las herramientas a la mano pues el hecho de ingresar textos y verificar lecturas es un proceso de cierta dificultad para ellos, para lo cual para optimizar este proceso se realizó el aplicativo móvil que simplemente permite un acceso rápido y eficiente en ejecutar las órdenes del usuario.

5.2 RECOMENDACIONES

- Cuando se trabaje con servidores Web Embebidos tal como el SitePlayer, es recomendable realizar páginas Web que no contengan gráficos o imágenes de gran tamaño o efecto de animaciones que consuman una cantidad considerable o excesiva de la memoria que es solo de 48 KBytes.

- Este trabajo de investigación constituye una base de conocimiento y preparación para futuros proyectos que considero los estudiantes puedan ocuparlo para mejorar y conseguir acciones que puedan facilitar el uso y mejora de la calidad de vida de otras personas, así como lo he considerado con este proyecto.

- Cabe recalcar que es necesario considerar que el sistema funciona a través una red cableada o inalámbrica lo cual resulta limitante en conectividad a cierta distancia, sin embargo considerando el uso, esta solo debe realizarse dentro del área de monitoreo.

- El uso de cámaras Ip garantiza cierta seguridad en la implementación del hogar lo que lo hace un dispositivo de gran ayuda, mucho más aún si es nocturno.

- Se debe tener mucho cuidado con el uso de este sistema ya que como utiliza parámetros de red para ejecutar las salidas tanto en luces como puertas, su uso debe ser exclusivamente personal.

REFERENCIAS BIBLIOGRÁFICAS

[1] Iphone SDK Programming, “A BEGGINERS GUIDE”, James A Brennan, Ed. Mc Graw Hill 2010, Fecha de consulta 02 mayo del 2011.

[2] Beginning IPHONE 3 Development, “Exploring Iphone SDK”, Dave Mark- Jeff LaMarche, APRESS, 2009, Fecha de consulta 14 mayo del 2011.

[3] Cocoa Programming for MacOS , Tercera edición, Aaron Hillegass, 2008, Fecha de consulta 20 de mayo del 2011.

[4] CORE ANIMATION for MacOS and the Iphone “, Bill Dudney ,Ed. Daniel Stanidberg, 2010, Fecha de consulta 25 mayo del 2011.

[5] IPHONE ADVANCED PROJECTS, Dave Mark, Johachim Bondo- Jeff LaMarche, APRESS, 2009, Fecha de consulta 01 junio del 2011.

[5] IPHONE DEVELOPER PROGRAM - WINTER, STANDFORD UNIVERSITY, ItunesU, 2009, Fecha de consulta 02 junio del 2011.

[6] WIKIPEDIA, “Sistema embebido”, http://es.wikipedia.org/wiki/Sistema_embebido, fecha de consulta el 13 de marzo del 2012.

[7] Sistemas Embebidos
http://elserver.forknetar.org/harpo/uch/seminario/escrito//archivos//seminario-uch.pdf, consultado el 15 de marzo del 2012.

[8] ELECTRICAL & COMPUTER ENGINEERING, “Embedded System Design Issues (the rest of the story), http://www.ece.cmu.edu/~koopman/iccd96/iccd96.html#example, fecha de consulta el 15 de marzo del 2012.

[9] SUPERROBOTICA.COM, “Manual de Sistema de Desarrollo SitePlayer”, http://www.superrobotica.com/download/S310265/Manual_Sistema_Desarrollo_SitePlayer_S310265.pdf, fecha de consulta 19 de febrero del 2013.

[10] MICROCHIP, “PIC 18F25XXA Data Sheet”, fecha de consulta el 1 de mayo del 2012.

[11] FAIRCHILD SEMICONDUCTOR, “MOC 3031-M datasheet”, http://www.datasheetcatalog.com/datasheets_pdf/M/O/C/3/MOC3031-M.shtml, fecha de consulta 4 de mayo del 2012.

[12] PHILIPS SEMICONDUCTORS, “BT136 series E”, http://www.datasheetcatalog.com/datasheets_pdf/B/T/1/3/BT136-600E.shtml, fecha de consulta 4 de mayo del 2012.