



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES

Fernando José Archila Morán

Jorge Raúl Gómez Bosarreyes

Asesorado por el Ing. Edgar Eduardo Santos Sutuj

Guatemala, enero de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

FERNANDO JOSÉ ARCHILA MORÁN

JORGE RAÚL GÓMEZ BOSARREYES

ASESORADO POR EL ING. EDGAR EDUARDO SANTOS SUTUJ

AL CONFERÍRSELES EL TÍTULO DE

INGENIEROS EN CIENCIAS Y SISTEMAS

GUATEMALA, ENERO DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. José Alfredo González Díaz
EXAMINADOR	Ing. Roberto Estuardo Ruiz Cruz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Edgar Josué González Constanza
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES

Tema que nos fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha noviembre de 2013.

Fernando José Archila Morán

Jorge Raúl Gómez Bosarreyes

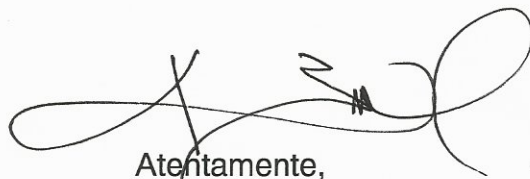
Guatemala, 17 de Septiembre de 2014

**Ingeniero
Carlos Alfredo Azurdia
Revisor de Trabajos de Graduación
Facultad de Ingeniería
Universidad de San Carlos de Guatemala**

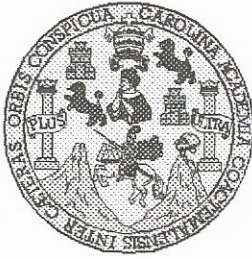
Ingeniero Azurdia:

Me complace saludarle, haciendo referencia al trabajo de graduación titulado "APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES", desarrollado por el estudiante universitario Fernando José Archila Moran con número de carné 200914994 y el estudiante universitario Jorge Raúl Gómez Bosarreyes con número de carné 200714230, que como asesor apruebo el contenido del mismo.

Para su conocimiento y efectos, sin otro particular, me suscribo.



Atentamente,
Ing. Edgar Santos
Asesor



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 3 de Septiembre de 2014

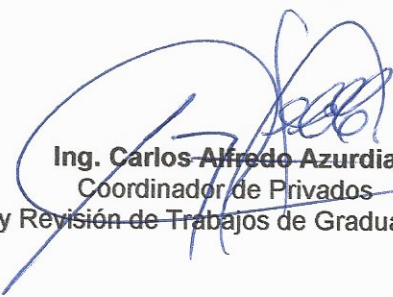
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de los estudiantes **FERNANDO JOSÉ ARCHILA MORAN** con carné **2009-14994**, y **JORGE RAÚL GÓMEZ BOSARREYES** con carné **2007-14230**, titulado: **“APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación “**APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES**”, realizado por los estudiantes **FERNANDO JOSÉ ARCHILA MORÁN** y **JORGE RAÚL GÓMEZ BOSARREYES**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

A handwritten signature in black ink, appearing to read 'Ing. Marlon Antonio Pérez Türk'.

Ing. Marlon Antonio Pérez Türk
Director, Escuela de Ingeniería en Ciencias y Sistemas



Guatemala, 19 de enero 2015



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **APLICACIÓN INTERACTIVA PARA PROMOCIONAR HOTELES**, presentado por los estudiantes universitarios: **Fernando José Archila Morán** y **Jorge Raúl Gómez Bosarreyes**, y después de haber **culminado las revisiones** previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. Murphy Olympo Paiz Recinos
Decano

Guatemala, 22 de enero de 2015

/gdech



ACTO QUE DEDICO A:

Dios	Por guiarme, cuidarme y estar siempre presente en cada una de las batallas a lo largo de la carrera.
Mis padres	Quienes con su amor, sacrificio, comprensión y apoyo, me han educado de forma correcta para ser mejor persona día con día.
Mi abuela	Quien con su amor y cariño incondicional ha cuidado de mí desde muy pequeño.
Universidad San Carlos de Guatemala	Por proporcionarme las herramientas y contribuir a mi formación académica.
Todos los guatemaltecos	Quienes con su arduo trabajo e impuestos brindan oportunidad a miles de jóvenes.

Fernando José Archila Morán

ACTO QUE DEDICO A:

- Dios** Por guiarme toda la vida.
- Mis padres** Jorge Humberto Gómez y Zonia Lisbeth Bosarreys, por su apoyo y esfuerzo durante todos estos años; no habría llegado hasta aquí de no ser por ustedes.
- Mis hermana** Sonia Pamela Gómez Bosarreys, por su cariño y apoyo durante toda mi vida. Por ser un ejemplo a seguir.
- Mis abuelos** María de Jesús Monterroso, Factor Bosarreys y German Gómez Gutiérrez (q.e.p.d.); los recordaré siempre.
- Mi abuela** Rosalía Azurdia, por su apoyo incondicional, cariño y comprensión.

Jorge Raúl Gómez Bosarreys

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	VII
GLOSARIO	IX
RESUMEN.....	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVII
1. MARCO TEÓRICO.....	1
1.1. Teoría que soporta la investigación	1
1.2. Relación con la tecnología escogida.....	4
2. APLICACIONES ACTUALES Y COMPARATIVA.....	7
2.1. Antecedentes	7
2.2. Mercado objetivo.....	8
2.3. <i>Benchmark</i> de la aplicación	8
2.3.1. <i>Booking.com</i>	8
2.3.2. <i>Hotels.com</i>	9
3. PROTOTIPO DE LA APLICACIÓN	11
3.1. Pantalla <i>Splash</i>	11
3.2. Pantalla de inicio	12
3.3. Módulo de servicios.....	13
3.4. Módulo de ubicación.....	14
3.5. Módulo de reservación	16
3.6. Módulo de fotografías.....	17

3.7.	Módulo de selección de fecha <i>Check in</i>	18
3.8.	Módulo de selección de fecha <i>Check out</i>	19
3.9.	Módulo de selección de políticas	20
3.10.	Validaciones y formularios.....	21
3.10.1.	Requisitos de las validaciones	21
3.11.	Formularios	22
3.12.	Diseño intuitivo y usabilidad	23
3.13.	Elementos	24
4.	DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN	25
4.1.	Herramientas	25
4.2.	Hardware	26
4.3.	Software	26
4.4.	MaIncloud	27
4.5.	Elementos (<i>storyboard</i>)	29
4.6.	Desarrollo del módulo de presentación	30
4.6.1.	Herramientas adicionales	31
4.6.2.	Proceso	31
4.7.	Desarrollo del módulo de localización	32
4.7.1.	Herramientas adicionales	32
4.7.2.	Proceso	34
4.8.	Desarrollo del módulo galería.....	38
4.8.1.	Herramientas adicionales	39
4.8.2.	Proceso	39
4.9.	Desarrollo del módulo de servicios	42
4.9.1.	Proceso	43
4.10.	Desarrollo del módulo de reservaciones.....	45
4.10.1.	Proceso	46

4.11.	Desarrollo del módulo de Políticas.....	50
4.12.	Consideraciones de implementación	51
CONCLUSIONES		53
RECOMENDACIONES		55
BIBLIOGRAFÍA... ..		57

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Modelo teórico.....	3
2.	Booking.com	9
3.	Hotels.com	10
4.	Pantalla <i>Splash</i>	12
5.	Pantalla de inicio	13
6.	Pantalla de servicios	14
7.	Pantalla de ubicación	15
8.	Pantalla Reservar.....	16
9.	Pantalla Galería	17
10.	Pantalla <i>Check in</i>	18
11.	Pantalla <i>Check out</i>	19
12.	Pantalla de selección de políticas	20
13.	Navegabilidad	23
14.	Elementos que comprende la aplicación.....	24
15.	Opciones de MacIncloud.....	27
16.	Vista general Xcode	28
17.	Especificaciones técnicas Mac.....	29
18.	<i>Storyboard</i>	30
19.	Opciones de pantalla.....	32
20.	API´s de Google	33
21.	Simple API Access.....	33
22.	Botones de las categorías.....	34
23.	Vista módulo de localización	38

24.	Vista módulo de galería	42
25.	Vista módulo de servicios	45
26.	Vista módulo de reservar	50
27.	Vista políticas del hotel	51

TABLAS

I.	Comparación de las pantallas según sus aplicaciones.....	10
----	--	----

LISTA DE SÍMBOLOS

Símbolo	Significado
Ddr3	<i>Double data rate</i>
Gb	Gigabyte
Px	Pixel

GLOSARIO

<i>API</i>	<i>Application programming interface.</i> Especifica cómo deben de interactuar los componentes de software con los demás.
<i>App</i>	Abreviación para la palabra aplicación.
<i>Benchmark</i>	Conjunto de librerías para desarrollo.
<i>Campo</i>	Espacio para ingreso de información.
<i>Check in</i>	Proceso de registro de entrada a un hotel.
<i>Check out</i>	Proceso de registro de salida de un hotel.
<i>Formulario</i>	Documento digital diseñado para que el usuario ingrese datos.
<i>GPS</i>	<i>Global Positioning System.</i> Sistema de posicionamiento global basado en satélites, que proporciona la ubicación de un dispositivo móvil.
<i>iOS</i>	Sistema operativo para dispositivos móviles de Apple.

<i>MacinCloud</i>	Servicio en línea que permite rentar una computadora con sistema operativo MacOS.
<i>Módulo</i>	Componente funcional de la aplicación.
MVC	Modelo vista controlador es un control de arquitectura de software que separa los datos y la lógica de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos.
Prototipo	Ejemplo de algo que se va a crear, en este caso desarrollar.
Red social	Ambiente para relaciones entre individuos.
SDK	Conjunto de librerías para desarrollo.
<i>Smartphone</i>	Dispositivo móvil inteligente con altas capacidades de procesamiento y mayor conectividad a un teléfono convencional.
<i>Splash screen</i>	Pantalla que se muestra durante la carga de la aplicación.
Storyboard	Representación gráfica de una secuencia de trabajo.
<i>Suite</i>	Habitación más lujosa de un hotel.

<i>Tablet</i>	Dispositivo móvil más liviano que una computadora, pero más potente que un teléfono.
Validar	Comprobar que se cumpla una condición.
ViewController	Clase que provee los fundamentos del modelo vista controlador para todas las aplicaciones basadas en iOS.
<i>Wi-fi</i>	Wireless fidelity. Permite a los dispositivos intercambiar datos de manera inalámbrica.

RESUMEN

Hoy en día existen muchos hoteles y alternativas de alojamiento para los viajeros y turistas. Con el increíble crecimiento e incorporación de la tecnología a las diferentes áreas del comercio y su integración con la información, es necesario adaptarse a las nuevas necesidades de los turistas, así como innovar en el medio a manera de sobresalir.

En el ámbito del turismo, específicamente los negocios hoteleros, es fundamental la publicidad; tomando en cuenta esta necesidad, este trabajo se enfoca en ofrecer un solución moderna, viable e innovadora para promover y publicitar un hotel, apoyados en tecnologías modernas que permiten maximizar el alcance de la publicidad.

La selección de soluciones tecnológicas está fundamentada y basada en las nuevas tecnologías enfocadas y dirigidas a los dispositivos móviles, determinando qué información se debe de presentar y qué operaciones se deben de permitir a través de la aplicación móvil.

OBJETIVOS

General

Colaborar con las tareas de publicidad, *marketing* y mercadotecnia de la industria hotelera, a través de una solución de software que sea accesible para los usuarios a través de un dispositivo móvil.

Específicos

1. Estudiar el entorno hotelero, a manera de conocer el esquema y estrategia de promociones.
2. Construir una solución de software que permita promocionar un hotel específico.
3. Desarrollar una solución de software que colabore con el proceso de registro de reservaciones para un hotel específico.

INTRODUCCIÓN

En el presente trabajo se da a conocer una solución tecnológica prototipo basada en el desarrollo de aplicaciones móviles, la cual tiene como objetivo incrementar y fortalecer el impacto de mercado y publicidad de un hotel. También se detallan tecnologías específicas y todos los requisitos, tanto de software como de hardware, para el desarrollo de la misma; así como también una sección explícita del proceso de desarrollo.

Para el desarrollo de la misma fueron tomados en cuenta aspectos clave que deben de potenciarse al momento de promocionar una institución hotelera, tales como la ubicación, servicios, precios e interacción en las redes sociales.

La aplicación está destinada específicamente a dispositivos con sistema operativo *iOS*; con esta aplicación se pretende mejorar la cantidad de visitantes al hotel, el interés por parte de los viajeros nacionales e internacionales e innovar tecnológicamente y diversificar los procesos para los visitantes.

1. MARCO TEÓRICO

En el presente capítulo se describirá la teoría de investigación utilizada, su relación con el trabajo de investigación, así como las razones de dicha elección y sus características principales.

1.1. Teoría que soporta la investigación

Para el presente trabajo de investigación se propuso escoger una teoría que se adecuará al tema y que sirviera como base para dicha investigación. Es por eso que se eligió la teoría *Task technology fit*. La traducción de la teoría es “Ajuste tecnológico al trabajo” y se enfoca primordialmente en la relación que existe, como su nombre lo dice, entre el trabajo a realizar, y la tecnología utilizada para ejecutarlo.

El objetivo planteado anteriormente de la teoría establece que el éxito de un sistema de información va a estar ligado a la relación entre las tareas y las tecnologías elegidas para su realización. Establece también que debe prestarse mayor atención a los problemas que puedan surgir durante el desarrollo de las tareas, ya que esto ayudará a evitar problemas más adelante.

Otra característica de la tecnología es que está modelada a partir de los requerimientos de los usuarios sobre el sistema, y cómo se relacionan dichos requerimientos con el tipo de tecnología escogida para cumplirlos. Finalmente se debe establecer el impacto de la tecnología en cada requerimiento funcional y no funcional, y establecer si la utilización de dicha herramienta trae un beneficio para el usuario.

La manera en que Task Technology Fit establece las relaciones y los beneficios para los usuarios, es seleccionando cierta cantidad de métricas para calificarlas de acuerdo con qué tanto se aplica dicha métrica al utilizar la tecnología escogida. La calificación se establece asignando un número de uno a siete. Las métricas calificadas por parte de la teoría son las siguientes:

- Calidad
- Localización
- Autorización
- Compatibilidad
- Facilidad de uso
- Tiempos de producción
- Confianza
- Relación con el usuario

En resumen, la teoría necesita de dos fuentes de información:

- Características de las tareas
- Características de las tecnologías

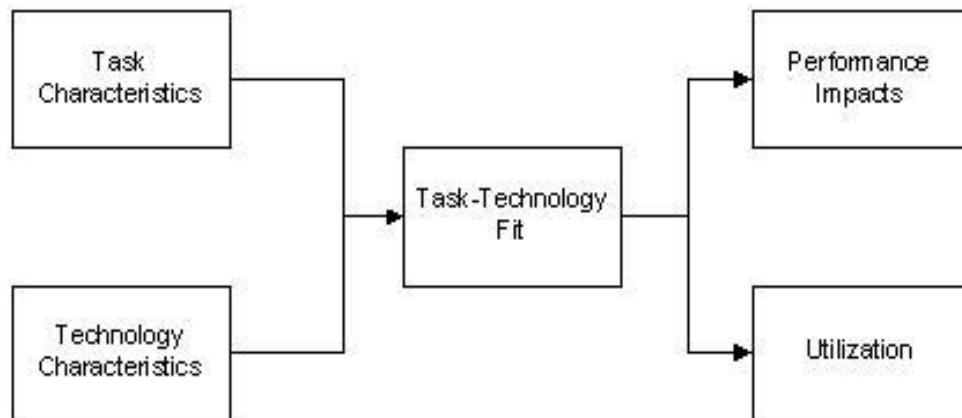
Se pueden establecer relaciones entre las dos fuentes para formar una idea y enumerar métricas para luego establecer dos cosas:

- Impacto de rendimiento
- Utilización
-
-

El impacto se verá en los beneficios directos que se tienen al hacer uso de la tecnología, mientras que la utilización va a darse a través de mediciones, como puede ser qué tanto se utiliza la aplicación, para qué tareas se utiliza y los propósitos de las mismas. Todo esto da como resultado un análisis que reflejará si la utilización de la tecnología trajo un beneficio o una ventaja competitiva sobre otras alternativas.

El modelo teórico se presenta a continuación:

Figura 1. **Modelo teórico**



Source: Goodhue and Thompson, (1995)

Fuente: <http://istheory.byu.edu/wiki/File:Ttf.JPG>. Consulta: 08 de octubre de 2013.

En la imagen se puede apreciar que es necesario conocer las características de las tareas a realizar y la tecnología que se quiere aplicar; luego utilizar la teoría para verificar los impactos que tendrá en la tarea seleccionada, así como el impacto en el usuario y el nivel de utilización que tendrá.

La utilización, según esta teoría, es el comportamiento de utilizar la tecnología para completar tareas. Medidas como la frecuencia de uso o la diversidad de aplicaciones utilizadas en el proceso son tomadas muy en cuenta. El impacto se refiere al enlace entre la teoría de ajuste y las consecuencias sobre la utilización de un sistema. La teoría debe de ser un determinante en cuanto a si el sistema de información es útil e importante, o si da una ventaja competitiva.

1.2. Relación con la tecnología escogida

Inicialmente, el proyecto estará orientado hacia tecnologías móviles, por lo que la teoría se adapta a las necesidades del caso, ya que como se manifestó antes, se considera muy importante la relación entre el usuario y la tecnología para el éxito del proyecto. También se tomó en cuenta que los requerimientos pueden ser revisados constantemente en búsqueda de errores, ya que el proceso de retroalimentación es muy rápido al tener la aplicación de manera inmediata al contacto con usuarios.

Finalmente, es un proyecto que puede ser revisado fácilmente tomando en cuenta el cumplimiento de las métricas establecidas por la teoría, como se explicará más adelante.

Las métricas que define la teoría se deberán tomar muy en cuenta para el desarrollo del proyecto:

- Calidad: esta se medirá de manera constante haciendo revisiones al proyecto terminado; por el tipo de proyecto se puede hacer un despliegue del proyecto periódicamente, y revisar en las herramientas que usarán los usuarios finales de la aplicación.

- Localización: al ser una aplicación móvil, esta métrica debe de ser fundamental, ya que el proyecto debe ser consultado desde cualquier lugar.
- Autorización: en el proyecto se tiene que contar con conexiones y fuentes de información seguras, ya sea localmente o hacia aplicaciones de terceros.
- Compatibilidad: la compatibilidad que se busca es con las diferentes versiones del sistema operativo de iOS.
- Facilidad de uso: por la naturaleza de la aplicación, se debe asegurar que sea fácil de utilizar, lo que facilitará el crecimiento de la misma, a través de buenas calificaciones en la tienda de aplicaciones.
- Tiempos de producción: esta métrica está más orientada al proceso de desarrollo, ya que establece mediciones para ver qué tan eficiente se fue en los tiempos establecidos para realizar la aplicación.
- Confianza: la confianza se dará conforme la aplicación se vaya distribuyendo, siempre tomando en cuenta los comentarios y calificaciones de los usuarios de la aplicación; debe asegurarse el mantener un constante desarrollo sobre la aplicación para mantenerla actualizada.
- Relación con el usuario: esta se mantendrá a través de la tienda propietaria de aplicaciones, tomando los comentarios y calificaciones, y aplicándose dichos cambios de manera rápida.

2. APLICACIONES ACTUALES Y COMPARATIVA

2.1. Antecedentes

Hoy en día existe una gran cantidad de hoteles a nivel local, los cuales se ven en la necesidad de destacar y llegar a un mayor número de clientes.

La mayoría de los hoteles cuenta con políticas de promoción orientada únicamente a medios de comunicación escritos e internet.

Actualmente los grandes hoteles disponen de un sitio web donde publica toda la información referente a los mismos, a excepción de las cadenas hoteleras mundiales, la gran mayoría de los hoteles no cuenta con una aplicación móvil para plataformas iOS, aplicaciones como "Marriot International", "Four Seasons" y "Jumeirah", entre otras; únicamente funcionan proporcionando el servicio de búsqueda de habitaciones a nivel mundial y prereservaciones de las mismas.

También existen aplicaciones como "Booking.com" u "Hotels.com" las cuales proporcionan información de una forma mas general de los hoteles, acordes a una zona elegida por el usuario. Sin embargo estas aplicaciones ofrecen información básica, poco personalizada y con pantallas sobrecargadas de datos, lo cual las hace poco atractivas para el usuario y visualmente deficientes.

2.2. Mercado objetivo

Se tienen identificados dos sectores del mercado a los cuales se beneficiará con el uso y comercialización de la aplicación; estos son: los turistas y las empresas hoteleras.

Realmente se busca impactar a quien a través de su teléfono o *tablet* busca información de interés sobre el hotel, donde también puede realizar una reservación y conocer más del mismo.

Otro sector importante al cual va dirigida la aplicación son las compañías hoteleras del medio, las cuales pueden utilizar la aplicación para promocionar su hotel de manera directa sobre los turistas de cualquier parte del mundo.

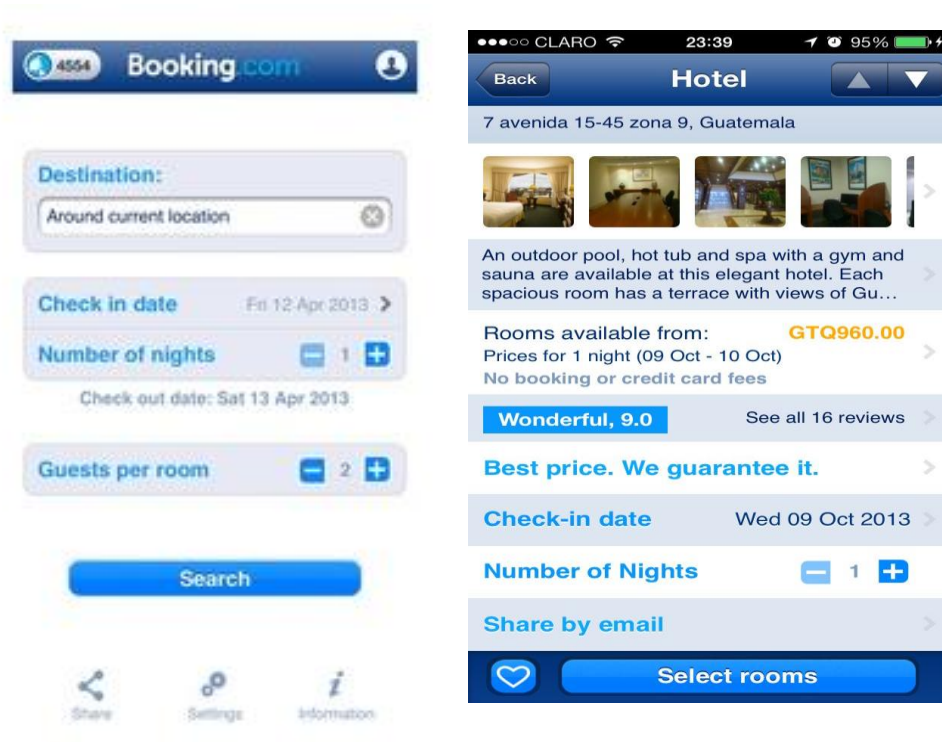
2.3. Benchmark de la aplicación

Se comparó y contrastó la aplicación desarrollada con aplicaciones similares como “Booking.com” y “Hotels.com” en los ámbitos de funciones, pantallas y estructura de la aplicación. Las pantallas que se utilizan se presentan a continuación.

2.3.1. Booking.com

Es una aplicación que proporciona servicios de acomodación de todo tipo de propiedad, desde un cuarto de pensionista hasta apartamentos de lujo a cualquier parte del mundo. Permite seleccionar un lugar, de acuerdo con el precio, y muestra las características del mismo con base en información básica como las fechas de alojamiento, cercanía y número de noches.

Figura 2. **Booking.com**

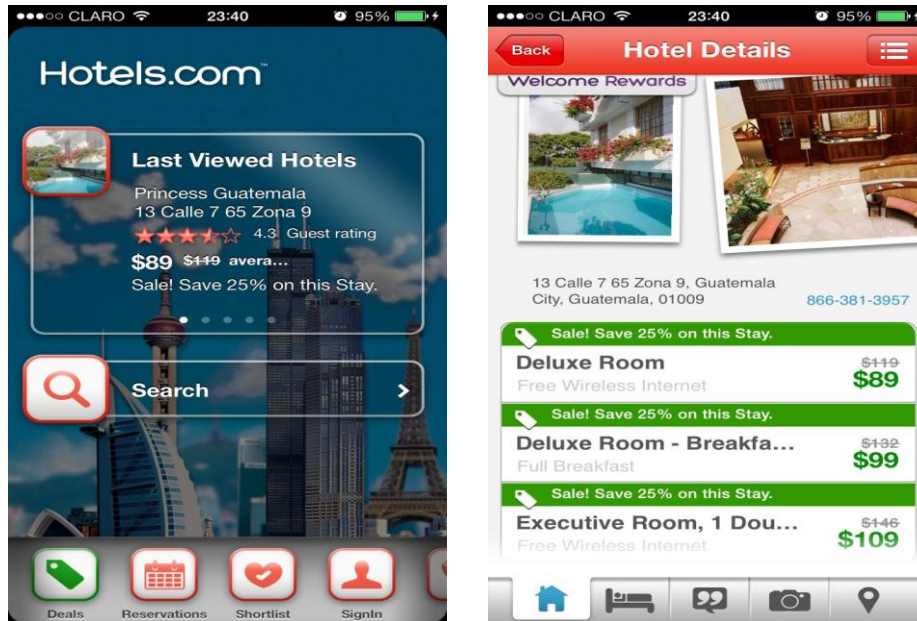


Fuente: elaboración propia utilizando una aplicación móvil.

2.3.2. **Hotels.com**

Es una aplicación que se enfoca en proporcionar el servicio de reservación vía *online* y por teléfono únicamente para hoteles. Permite visualizar los hoteles en su ubicación, tener una lista de los favoritos para seleccionar y comunicarse con cualquiera de ellos, con base en la posición del usuario.

Figura 3. **Hotels.com**



Fuente: elaboración propia utilizando una aplicación móvil.

Tabla I. **Comparación de las pantallas según sus aplicaciones**

Booking.com	Hotels.com	Horizon Suites
Reservar habitación	Reservar y conocer el	Reservación interactiva
Múltiples hoteles	hotel	Oferta y descuentos en
Ubicación en el mapa	Múltiples hoteles	reservación
Pequeña galería de fotos	según la zona	Galería de fotos del hotel
Cálculo de precios (aproximado)	Pequeña galería	Información de servicios
	Precio estimado	Información y políticas del hotel
		Mapa con lugares turísticos

Fuente: elaboración propia.

3. PROTOTIPO DE LA APLICACIÓN

En el presente capítulo se presentará el prototipo de la aplicación. Se mostrarán imágenes sobre lo que se espera obtener al final del desarrollo. Se describe cada uno de los módulos y se hace una descripción sobre el funcionamiento esperado.

Asimismo, se presentará la manera en que el usuario va a interactuar entre las diferentes pantallas de la aplicación, con el objetivo de mantener la usabilidad de dicha aplicación.

Se incluirá además incluirá un diseño limpio previo de la aplicación, para que los usuarios obtengan información rápida y fácil.

3.1. Pantalla *Splash*

La pantalla *Splash* es la primera que aparece al momento de abrir la aplicación es simbólica y aparece mientras cargan las dependencias de la aplicación.

Figura 4. **Pantalla *Splash***



Fuente: elaboración propia.

3.2. Pantalla de inicio

La pantalla de inicio es la que aparece una vez se ha iniciado la aplicación; en ella se incluye una imagen representativa del hotel, el logotipo de la aplicación, un pequeño '*slogan*' y los respectivos botones para compartir en redes sociales y por correo. También se encuentra habilitado el panel de pestañas para desplazarse a las demás módulos de la aplicación.

Figura 5. **Pantalla de inicio**

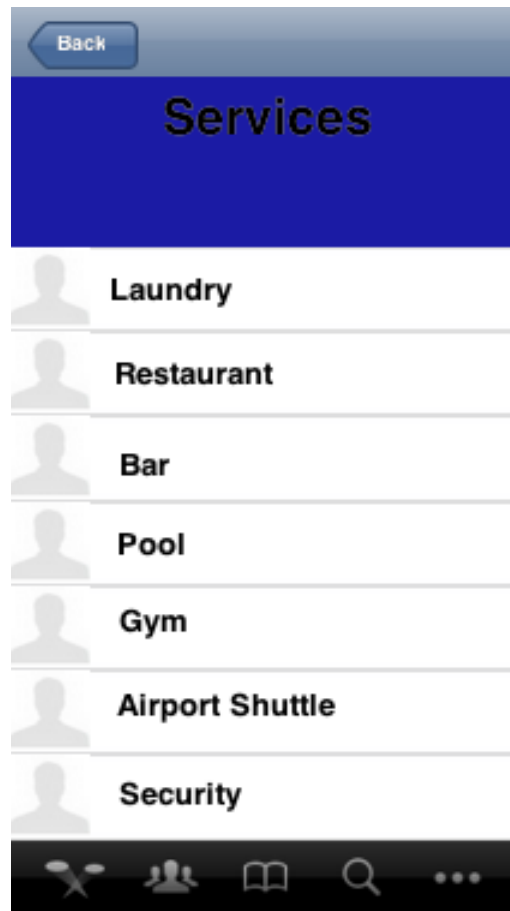


Fuente: elaboración propia.

3.3. Módulo de servicios

Está compuesto por una lista de los servicios que proporciona el hotel que se está promocionando. La lista cuenta con un nombre y una pequeña imagen. Al momento de hacer clic en ella, aparece una pequeña descripción del servicio.

Figura 6. **Pantalla de servicios**



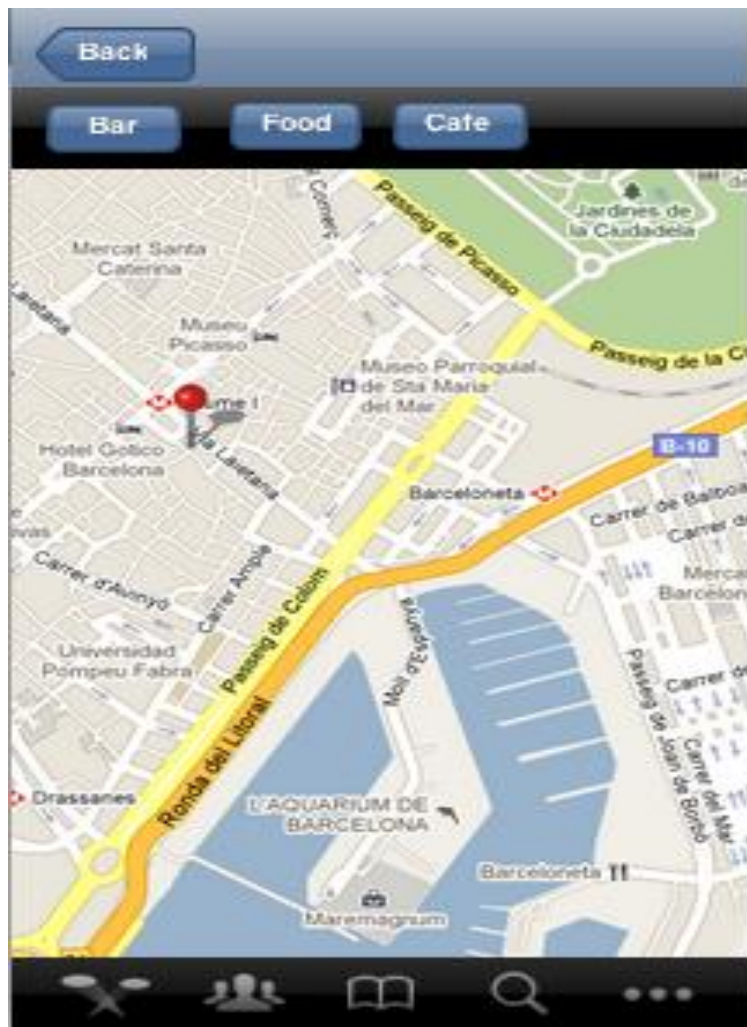
Fuente: elaboración propia.

3.4. **Módulo de ubicación**

En esta pantalla se puede visualizar el mapa con la ubicación del hotel; asimismo, cuenta con botones de diferentes sitios de interés; para obtener puntos cercanos al hotel se debe de presionar cualquiera de los botones según su categoría.

Este módulo también cuenta con la barra inferior para desplazarse a través de la aplicación libremente.

Figura 7. **Pantalla de ubicación**



Fuente: elaboración propia utilizando una aplicación móvil.

3.5. Módulo de reservación

Esta pantalla cuenta con un formulario de reservación donde el usuario debe de ingresar la información personal y básica; así también se seleccionan las fechas de la reservación y el tipo de cuarto. Este módulo también permite el posteo en redes sociales al momento de reservar. Por ultimo, este módulo también cuenta con la barra inferior para desplazarse a través de la aplicación libremente.

Figura 8. **Pantalla Reservar**

The screenshot shows a mobile application interface for reserving a room. The background is a solid blue color. At the top left, there is a white button with a left-pointing arrow and the text "Back". The main title "Reservar" is displayed in white, bold font, with a white button labeled "Reservar" to its right. Below the title, there are three white buttons stacked vertically: "Checkin", "Checkout", and "Room". Underneath these buttons are four white text input fields, each with a "Text" placeholder. The labels "Name", "Email", "phone", and "Skype" are positioned to the left of their respective input fields. At the bottom of the screen, there is a dark navigation bar with five white icons: a home icon, a people icon, a book icon, a magnifying glass icon, and a menu icon.

Fuente: elaboración propia.

3.6. Módulo de fotografías

Esta pantalla cuenta con una serie de fotografías sobre los puntos más destacados del hotel, presentadas de forma de colección o galería. También se cuenta con la barra inferior de navegación para visualizar las diferentes pestañas de la aplicación.

Figura 9. Pantalla Galería



Fuente: elaboración propia.

3.7. Módulo de selección de fecha *Check in*

Este módulo dispone de las barras de navegación de la aplicación, una imagen representativa de la pantalla y un selector de fecha. Por medio de la barra superior se puede seleccionar o cancelar la selección de fecha para el posterior uso en formularios.

Figura 10. Pantalla *Check in*



Fuente: elaboración propia.

3.8. Módulo de selección de fecha *Check out*

Este módulo dispone de las barras de navegación de la aplicación, una imagen representativa de la pantalla y un selector de fecha. Por medio de la barra superior se puede seleccionar o cancelar la selección de fecha para el posterior uso en formularios.

Figura 11. Pantalla *Check out*



Fuente: elaboración propia.

3.9. Módulo de selección de políticas

Esta pantalla muestra las políticas y restricciones por las cuales se basa y rige el hotel que se está promocionando. Cuenta con información que todo usuario debe saber a la hora de querer visitar un hotel.

Figura 12. Pantalla de selección de políticas



Fuente: elaboración propia.

3.10. Validaciones y formularios

A continuación se presentan los criterios de validación y formularios que se encuentran dentro de la aplicación.

3.10.1. Requisitos de las validaciones

- La aplicación cuenta con una serie de validaciones y restricciones que deben cumplirse para asegurar su uso óptimo.
- Conexión a internet: el usuario debe disponer de conexión a la red, ya sea por medio de plan de datos o mediante una red wi-fi.
- GPS activado y con cobertura: el dispositivo debe poseer sistema de posicionamiento global de alta precisión, para el adecuado funcionamiento de la aplicación.
- Correo electrónico configurado en dispositivo: el usuario debe tener configurada alguna cuenta de correo electrónico en su dispositivo.
- Cuenta activa en Facebook y/o Twitter: se debe de disponer de una cuenta en alguna de estas 2 redes sociales para utilizar al máximo la aplicación. Asimismo, estas deben de estar habilitadas en los permisos de usuario, activas en la red.

3.11. Formularios

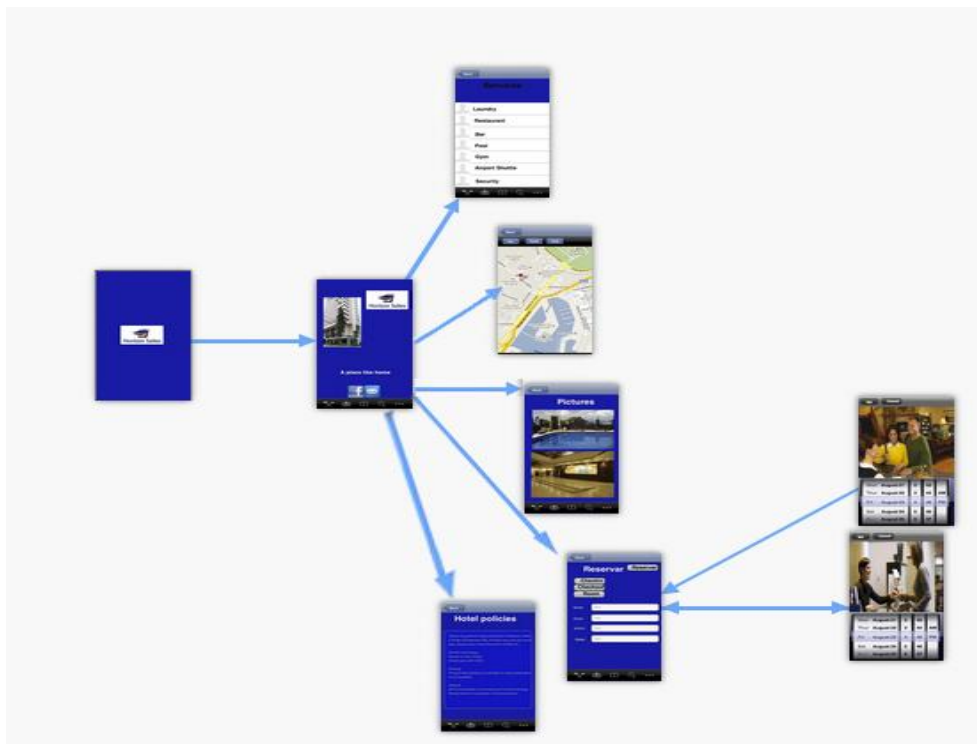
La aplicación cuenta con un único formulario, el cual es utilizado para la obtención de la información necesaria para la reservación del hotel. El formulario consta de los siguientes campos:

- Fecha *Check in*: se selecciona mediante la interfaz una fecha de ingreso al hotel.
- Fecha *Check out*: se selecciona mediante la interfaz la fecha de egreso del hotel.
- Tipo de habitación: se debe seleccionar mediante la interfaz el tipo de habitación que se desea.
- Nombre: el usuario ingresa su nombre con el teclado proporcionado.
- Apellido: el usuario ingresa su apellido con el teclado proporcionado en pantalla.
- Teléfono: se ingresa el número telefónico con el teclado proporcionado en pantalla.
- *Skype*: se ingresa el usuario de *Skype* con el teclado proporcionado en pantalla.
- Email: el usuario ingresa su correo electrónico con el teclado proporcionado en pantalla.

3.12. Diseño intuitivo y usabilidad

El diseño de la aplicación busca la simplicidad y rápida navegabilidad; por esto es que la aplicación presenta un menú que está siempre a la vista y disponible al usuario, para poder navegar libremente hacia cualquier pantalla de la aplicación.

Figura 13. Navegabilidad

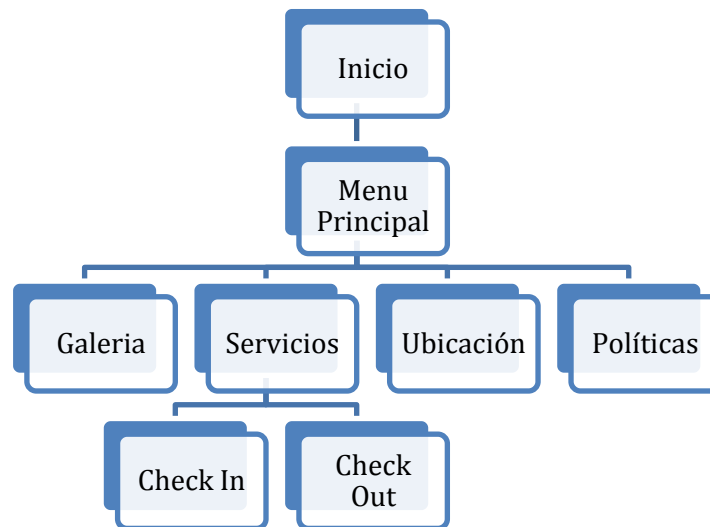


Fuente: <https://www.fluidui.com/editor/live/> Consulta: 10 de noviembre de 2013.

3.13. Elementos

Es necesario tomar en cuenta en el prototipo los diferentes elementos que van a componer la aplicación; asimismo la manera en que estos se relacionan y cómo la aplicación los va a manejar. Es por eso que el siguiente diagrama muestra cómo dichos elementos se relacionan.

Figura 14. Elementos que comprende la aplicación



Fuente: elaboración propia.

El menú de inicio será el punto de partida de la aplicación; luego de mostrar la imagen de carga, se tendrá el nombre del hotel, una imagen y el menú, para recorrer los diferentes módulos de la aplicación. El menú principal será un *tabBarController* para poder mostrarlo en la mayoría de pantallas, y que sea de fácil acceso en todo momento para el usuario. El modelo podrá replicarse de manera sencilla, utilizando el *Storyboard* que provee Xcode, para establecer las diferentes conexiones entre los módulos.

4. DOCUMENTACIÓN Y TUTORIAL DE PROGRAMACIÓN DE LA APLICACIÓN

4.1. Herramientas

Para el ciclo de vida del proyecto de software se utilizaron diferentes herramientas tanto de hardware como de software; estos se detallan a continuación:

- Xcode 4.6.3 (4H1503): IDE utilizado para el desarrollo y construcción de la solución de software.
- Photoshop CS5: aplicación de edición de imágenes utilizada como apoyo para el diseño gráfico.
- MacInCloud: servicio de apoyo en la nube que se utiliza para construir y compilar la solución de software.
- IOS Simulator 6.0 (369.2): emulador oficial de Apple, cuya función es realizar pruebas.
- Google Maps API: sirve para integrar los mapas a la solución de software.

4.2. Hardware

Entre los requisitos de hardware utilizados para desarrollar la solución de software están:

- Computadora Apple MacbookPro Core 2 Duo
- 4 GB RAM DDR3
- 250 GB HD
- Dispositivo con iOS

4.3. Software

Los requisitos de software para desarrollar la solución son:

- Xcode 4.6.3
- Mac OSX 10.7.5
- IOS Simulator 6.0
- Adobe Photoshop CS5
- IOS SDK
- Mac In Cloud
- Dispositivo con IOS v 6.0.1 en adelante
- Github
- Windows Remote Desktop
- Notepad
- Notepad ++
- TextEdit
- Google Maps API
- Microsoft Word 2013

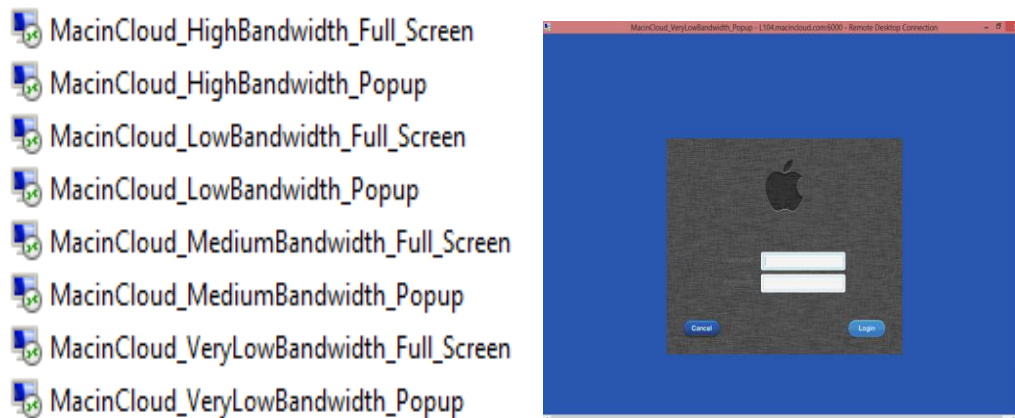
- Microsoft Windows 8
- Itunes 10

4.4. MacIncloud

Para la realización del presente proyecto fue necesario utilizar un servicio de terceros para la utilización de un sistema operativo Mac. Por lo que luego de comparar alternativas, se seleccionó el servicio de *MacInCloud*. Se obtuvo una suscripción con la que se puede ingresar por 5 horas al día.

Para poder conectarse es necesario utilizar la conexión a escritorio remoto de Windows. MacInCloud provee conexiones directas a través de un zip que se envía al correo, el cual tiene varias opciones, dependiendo del ancho de banda.

Figura 15. Opciones de MacIncloud

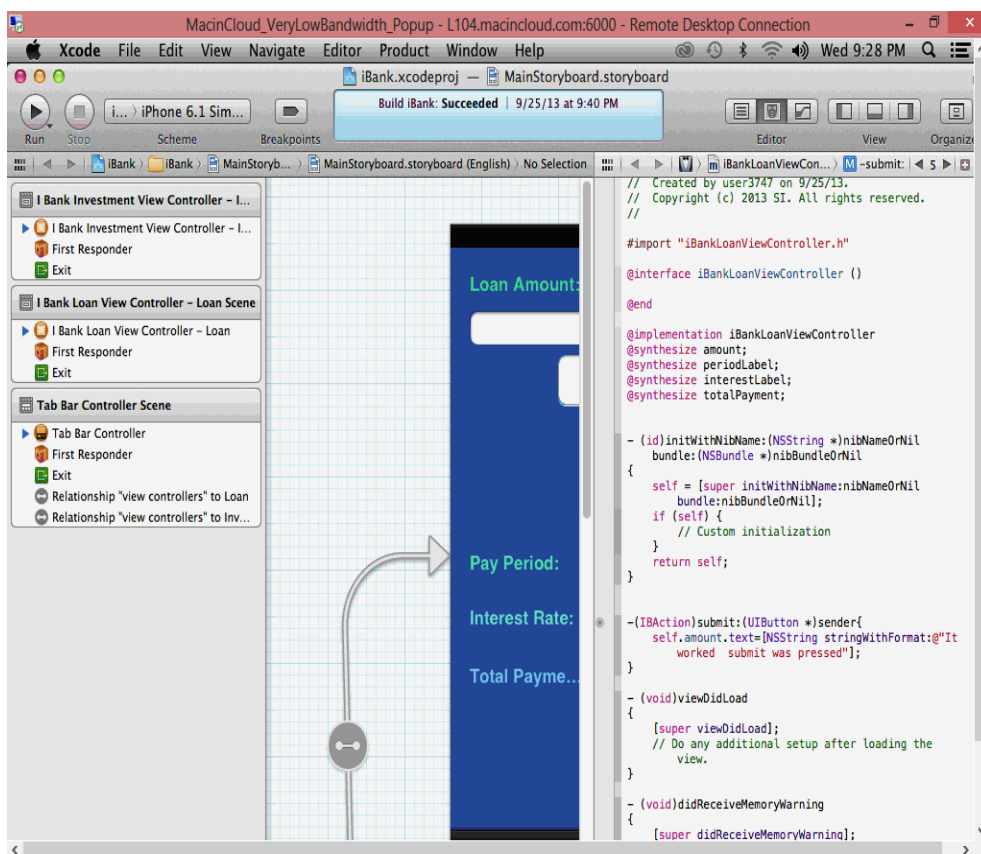


Fuente: elaboración propia.

Al abrir un enlace se muestra el área de ingreso de información del usuario.

Al ingresar, se tendrá una Mac totalmente funcional, con herramientas ya preinstaladas dedicadas al desarrollo de aplicaciones, incluyendo XCode que es el IDE utilizado para el proyecto. Asimismo, cuenta ya con GitHub para el control de versiones.

Figura 16. Vista general Xcode



Fuente: elaboración propia.

Figura 17. **Especificaciones técnicas Mac**



Fuente: elaboración propia.

4.5. Elementos (*storyboard*)

Para la realización del proyecto es necesario tener definidas las relaciones entre los diferentes elementos o módulos que van a componer la aplicación; en este caso se cuenta con:

- Localización
- Galería
- Servicios
- Reservaciones
- Políticas

Xcode ya provee una herramienta gráfica llamada *Storyboards*, que sirve para establecer de relaciones entre los elementos, y así poder visualizar de mejor manera el flujo de las acciones que podrá realizar el usuario. En este caso el *Storyboard* utilizado es el siguiente:

Figura 18. **Storyboard**



Fuente: elaboración propia.

4.6. Desarrollo del módulo de presentación

El objetivo del módulo de presentación es desplegar a los usuarios una agradable vista inicial de la aplicación y proporcionar enlaces a redes sociales del hotel; así también dispone de la capacidad de compartir el hotel por internet. Este módulo consistirá en una imagen distribuida en la pantalla junto con el logotipo y *slogan* del hotel; en la parte inferior cuenta con los botones de enlace para las redes sociales Facebook y Twitter, así como el botón de compartir por email.

4.6.1. Herramientas adicionales

Para el desarrollo de dicho módulo de presentación se utilizaron los *frameworks* de desarrollo:

- UIImageFramework
- UIImageFramework

4.6.2. Proceso

Lo primero que se debe tener es un *ViewController* en el *Storyboard* que va a ser utilizado por el módulo; se agregan los objetos “*UIImageView*”, para las respectivas imágenes. Después se deben agregar los botones para conectar la aplicación con los medios sociales y compartir correo. Lo siguiente será asignarle nueva clase al *ViewController* en el *Storyboard*; estas nuevas clases se llamarán: *HomeViewController*, derivadas de *HomeViewController*.

Posteriormente se agregan y relacionan los botones al archivo de encabezados y se asigna el código a los botones con su respectiva funcionalidad, de la siguiente manera:

```
- (IBAction)fb_button:(id)sender {  
  
    [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"https://www.facebook.com/  
    pages/Holiday-Inn-Zona-10-Guatemala/135752759825855"]];  
}  
  
- (IBAction)tw_Button:(id)sender {  
    [[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"https://twitter.com/  
    HolidayInn"]];  
}
```

Al compilar y ejecutar el código se presentarán las siguientes pantallas según el botón que el usuario elija.

Figura 19. Opciones de pantalla



Fuente: elaboración propia.

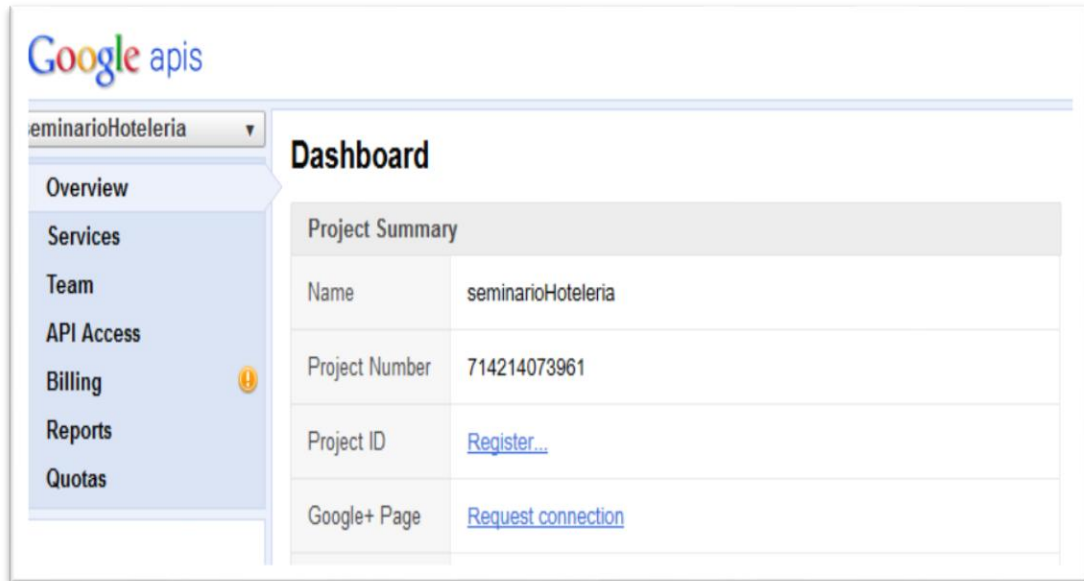
4.7. Desarrollo del módulo de localización

A continuación se detalla el proceso de programación del módulo de localización.

4.7.1. Herramientas adicionales

Para la realización del módulo de localización se utilizará la API de Google Places, para esto fue necesario crear un nuevo proyecto dentro de la consola de API's de Google.

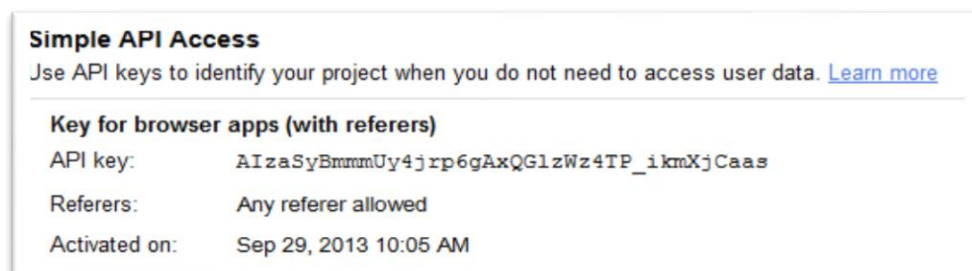
Figura 20. **API's de Google**



Fuente: <https://developers.google.com/places/documentation/>.
Consulta: 10 de noviembre de 2013.

La *API* va a proveer una llave para obtener los datos de los lugares cercanos, dependiendo de la categoría seleccionada.

Figura 21. **Simple API Access**



Fuente: <https://developers.google.com/maps/documentation/javascript/reference> web.
Consulta: 11 de noviembre de 2013.

Los *frameworks* necesarios para que los mapas funcionen correctamente serán:

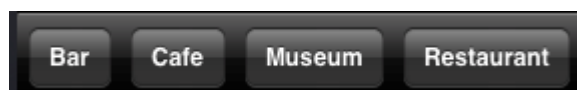
- MapKit.framework
- CoreLocation.framework

4.7.2. Proceso

Lo primero que se debe tener es un *ViewController* en el *Storyboard*, que va a ser utilizado por el módulo; se agrega una barra de herramientas, que contenga los botones con los nombres de las categorías que se van a buscar. En este caso dichos botones serán:

- Bar
- Café
- Museum
- Restaurant
- Park

Figura 22. **Botones de las categorías**



Fuente: elaboración propia.

Debajo de la barra, se agrega el objeto MKMapView desde la librería de objetos. Al ejecutar la aplicación e ir hacia el módulo de localización se podrá ver qué carga el mapa de manera correcta, aunque sin funcionalidad.

Lo siguiente será asignarle nueva clase al *ViewController* en el *Storyboard*; estas nuevas clases se llamarán: *UbicaciónViewController*, derivadas de *UIViewController*.

Ahora es necesario conectar el *MapView* con *UbicacionViewController*, por lo que sosteniendo CTRL se arrastra el *MapView* hacia el archivo, con lo que se tendrá una nueva línea en el código:

```
@property (strong, nonatomic) IBOutlet MKMapView *mapView;
```

Posteriormente se debe agregar el código para obtener la posición actual del usuario, y que esta pueda ser mostrada en el mapa. El código es el siguiente:

```
CLLocationManager *locationManager;  
CLLocationCoordinate2D currentCentre;  
int currenDist;
```

Luego en el archivo *UbicacionViewControler.m* agregar lo siguiente al método *ViewDidLoad*:

```
self.mapView.delegate=self;  
[self.mapView setShowUserLocation:YES];  
locationManager = [[CLLocationManager alloc] init];  
[locationManager setDelegate:self];  
[locationManager setDistanceFilter:kCLLocationDistanceFilterNone];  
[locationManager setDesiredAccuracy:kCLLocationAccuracyBest];
```

El código anterior asegura que se mostrará la posición en el mapa, así como seguimiento de la posición.

Ahora, al ejecutar la aplicación se mostrará una ventana con la solicitud de acceso a la posición del usuario, junto con un pin a dicha posición.

```
- (void)mapView:(MKMapView *)mv didAddAnnotationViews:(NSArray *)views
{
    CLLocationCoordinate2D coordenadasHotel;
    coordenadasHotel = CLLocationCoordinate2DMake(14.633316, -
90.516629);
    MKCoordinateRegion region;
    region =
MKCoordinateRegionMakeWithDistance(coordenadasHotel,500,500);

    [mv setRegion:region animated:YES];
}
```

El código anterior permite hacer el acercamiento; ya que se va a tener una posición estática, y a establecer el área alrededor de las coordenadas del hotel, con una región de acercamiento de 500 m.

La manera en que se obtienen datos desde la API de Google Places es a través de un JSON que devuelve lugares, de acuerdo con las solicitudes del usuario.

Los parámetros que se van a utilizar para la solicitud son los siguientes:

- Llave de la API
- Ubicación
- Radio
- Sensor

En este caso la cadena de solicitud de información es la siguiente:

```
NSString *url = [NSString  
stringWithFormat:@"https://maps.googleapis.com/maps/api/place/search/json?l  
ocation=%f,%f&radius=%@&types=%@&sensor=true&key=%@",  
currentCentre.latitude, currentCentre.longitude, [NSString  
stringWithFormat:@"%i", currenDist], googleType, kGOOGLE_API_KEY];
```

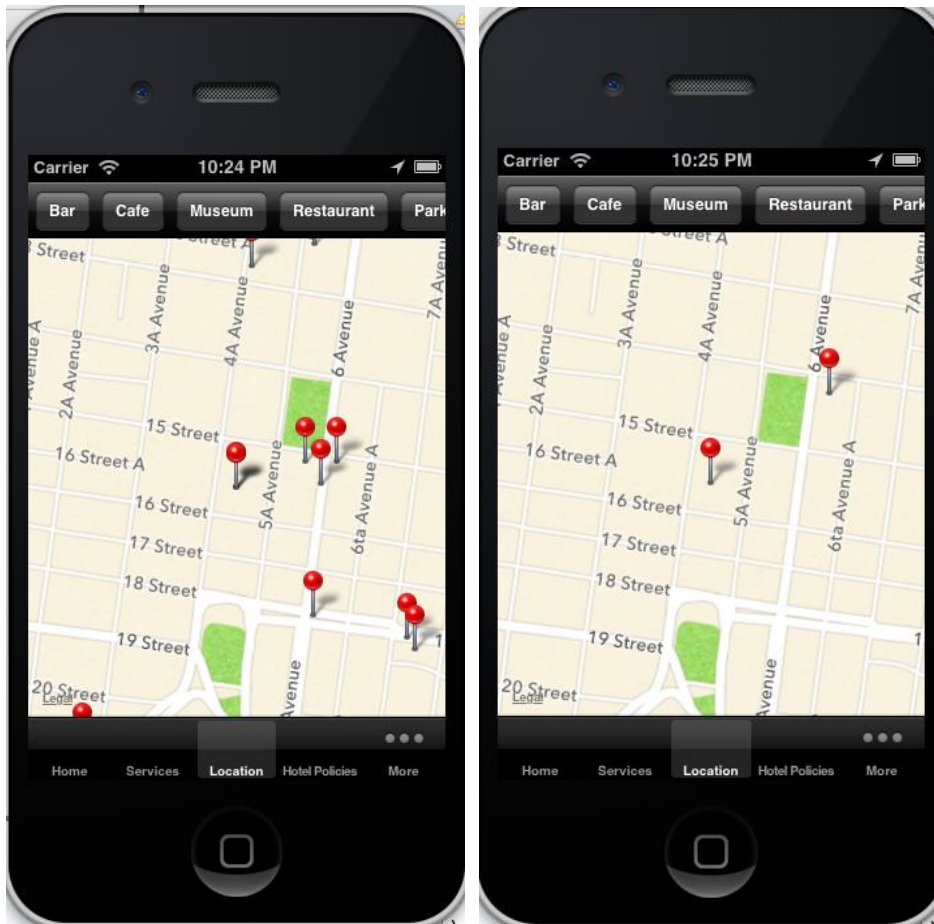
Ya obtenidos los datos de los lugares, es necesario mostrarlos en el mapa:

```
MapPoint *placeObject = [[MapPoint alloc] initWithName:name address:vicinity  
coordinate:placeCoord];  
[self.mapView addAnnotation:placeObject];
```

Toda la información como el nombre y coordenadas, son obtenidas directamente del JSON que devuelve la API.

Luego de ejecutar el código cada vez que se selecciona una categoría, se mostrarán lugares que cumplen con ese filtro.

Figura 23. Vista módulo de localización



Fuente: elaboración propia.

4.8. Desarrollo del módulo galería

El objetivo de la galería es mostrar imágenes de las diferentes áreas y actividades que se realizan en el hotel.

4.8.1. Herramientas adicionales

Para la realización básica del módulo galería, es necesario tener cargadas varias imágenes que se utilizarán para comprobar el funcionamiento del mismo; estas se agregan al proyecto.

4.8.2. Proceso

Lo primero que se tiene que hacer luego de haber cargado las imágenes es que en lugar de un *ViewController*, que es el controlador por defecto, se debe agregar un *UICollectionViewController*.

El *UICollectionViewController* provee de una vista con celdas, las cuales se pueden modificar al gusto; estas serán los contenedores para las imágenes. Se puede modificar el tamaño de la celda de manera gráfica, arrastrando sus bordes, o especificando el tamaño en sus propiedades. En dichas propiedades de la celda se debe especificar un identificador, por lo que se le nombró *CellID*.

El siguiente paso es crear una clase específica para la celda; en este proyecto es *MyCostumCell* derivada de *UICollectionViewCell*, y se le asigna a la celda.

Es necesario hacer la unión entre la celda y la clase que recién se creó, por lo que se arrastra la celda hacia la clase *MyCostumCell.h* para unir las.

Eso crea el siguiente código dentro de la clase:

```
@property (strong, nonatomic) IBOutlet UIImageView *ImageinCell;
```

Se necesita que la clase derivada sea *UICollectionViewController*, asimismo es necesario agregar *UICollectionViewDataSource* y *UICollectionViewDelegate*, con lo que se tendría el siguiente código en la clase *PicViewController.h*:

```
@interface PicViewController: UICollectionViewController
<UICollectionViewDataSource,UICollectionViewDelegate>{
```

Lo siguiente es ir a la clase *PicViewController.m* e importar las clases creadas:

```
#import "PicViewController.h"
#import "MyCostumCell.h"
```

Ahora se tiene que crear un arreglo donde se guardarán los nombres de las imágenes, esto en el archivo *.h*.

```
NSArray *recipePhotos;
```

Este arreglo se utiliza en el archivo *.m* en el método *viewDidLoad*, donde se le asigna los valores, tomando como base los nombres de las imágenes.

```
recipePhotos = [[NSArray alloc
initWithObjects:@"1.JPG",@"2.JPG",@"3.JPG",@"4.jpg",@"5.jpg",@"6.jpg",@"
7.jpg",@"8.jpg",@"9.jpg",@"10.jpg",@"11.jpg", nil];
```

Es necesario tener un método que diga cuántas secciones vamos a tener, para este caso se utilizará una, por lo que este método llamado *numberOfSectionsInCollectionView* retornará un 1.

Asimismo, se tendrá un método que devolverá la cantidad de elementos que contiene el arreglo. Este se llamará *numberOfItemsInSection*.

El código utilizado para mostrar las imágenes es el siguiente:

```
static NSString *cellID=@"CellID";
    MyCostumCell *Mycell= (MyCostumCell *)[collectionView
dequeueReusableCellWithReuseIdentifier:cellID forIndexPath: indexPath];

    Mycell.ImageinCell.image= [UIImage imageNamed:[recipePhotos
objectAtIndex:indexPath.item]];
    return Mycell;
```

Este código obtiene la celda llamada *CellID*. Luego se crea un objeto de la clase para la celda y se inicia.

Finalmente se establece que la imagen de la celda consistirá en cada uno de los elementos del arreglo de imágenes.

Figura 24. Vista módulo de galería



Fuente: elaboración propia.

4.9. Desarrollo del módulo de servicios

El módulo de servicios servirá para mostrar al usuario las diferentes actividades que pueden realizar dentro del hotel; esto se hará a través de una lista con dichas actividades, desde la cual el usuario podrá seleccionar una y obtener más información.

4.9.1. Proceso

Lo primero que se debe tener es un objeto *TableView*, el se arrastra desde la librería de objetos hacia el área de trabajo en el *storyboard*.

Luego se crean las clases para poder manejar el *ViewController*, en nuestro caso se llamará *ServiciosViewController*.

En los encabezados se deberá agregar lo siguiente:

```
#import <UIKit/UIKit.h>
```

```
@interface ServiciosViewController : UIViewController  
<UITableViewDelegate, UITableViewDataSource>
```

En la clase *ServiciosViewController*, se debe primero agregar las definiciones para los arreglos a utilizar; estos serán *TableData* y *thumbnails*.

Al cargar es necesario que se llenen los arreglos, por lo que en el método de *ViewDidLoad* se agrega lo siguiente:

```
tableData = [NSArray arrayWithObjects:  
    @"Restaurante", @"Room Service", @"Bar", @"Salones de  
    Conferencia", @"Piscina Climatizada", @"Gimnasio", @"Lavanderia", @"Venta  
    de Regalos y Revistas", @"Centro de Negocios e Internet", @"Renta de Autos",  
    @"Agencia de Viajes", @"Intercambio de Dinero", @"Transporte Aereopuerto-  
    Hotel", @"Seguridad 24h", @"Cuidados Medicos y Ambulancia", nil];
```



```
thumbnails = [NSArray arrayWithObjects:
```

```
    @"restaurante.jpeg",    @"room_Service.jpeg",    @"bar.jpeg",  
    @"salon.jpeg",        @"piscina.jpeg",        @"gym.jpeg",        @"lavanderia.jpeg",  
    @"regalo.jpeg",        @"internet.jpeg",        @"autos.jpeg",        @"agencia.jpeg",  
    @"intercambio.jpg",    @"transporte.jpeg",    @"seguridad.jpeg",    @"medico.jpeg",  
    nil];
```

Con esto se llenan los arreglos con las imágenes que se quieren mostrar. Es necesario también mostrar la información de cada imagen, por lo que se utiliza el arreglo `tableData`, y el `Thumbnail` para las imágenes:

```
cell.label_nombre_servicio.text = [tableData objectAtIndex:indexPath.row];  
cell.image_servicio.image = [UIImage imageNamed:[thumbnails  
objectAtIndex:indexPath.row]];
```

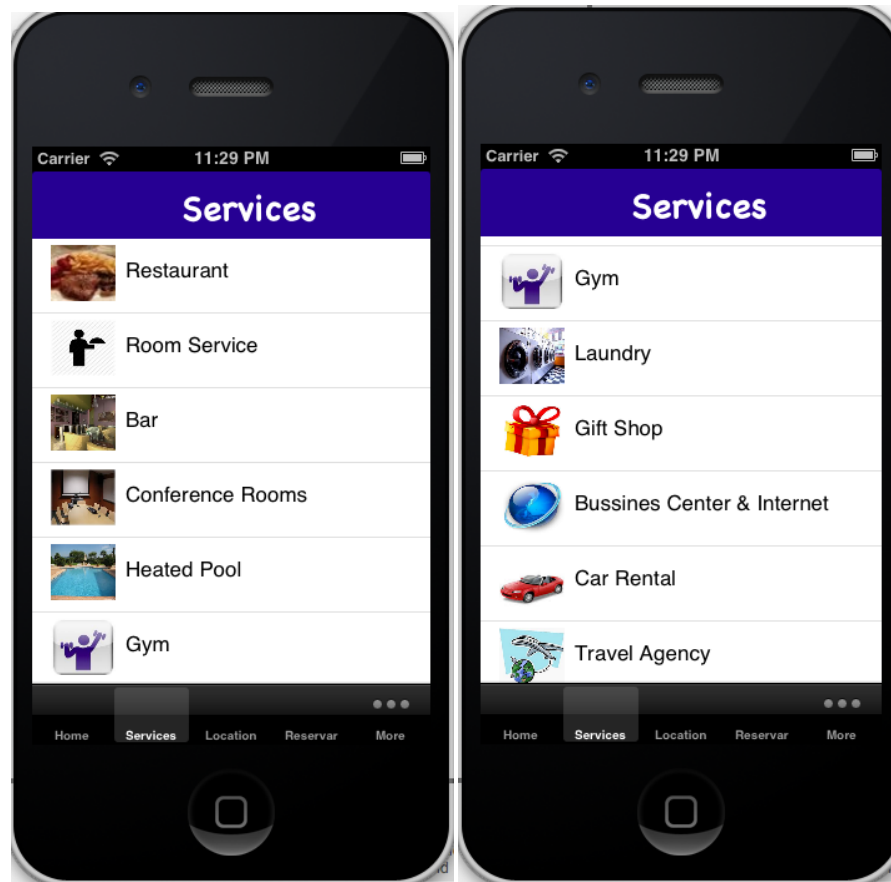
El siguiente paso es modificar la celda a un diseño que se vea mejor. Para esto se crea un nuevo archivo llamado *SimpleTableCell*.

Esta clase se va a modificar visualmente, para que contenga los elementos necesarios. A dicha celda se le agrega un *Imageview* para contener las fotos, y una *label* para los nombres. Para esto se utiliza el código creado anteriormente, pues son los mismos elementos; solo lo visual va a cambiar.

Luego se crea la clase que va a utilizar esa celda, que será *SimpleTableCell.m*. En el encabezado de esa clase se une la etiqueta con la imagen, por lo que se tendrá lo siguiente:

```
@property (strong, nonatomic) IBOutlet UILabel *label_nombre_servicio;  
@property (strong, nonatomic) IBOutlet UIImageView *image_servicio;
```

Figura 25. Vista módulo de servicios



Fuente: elaboración propia.

4.10. Desarrollo del módulo de reservaciones

El módulo de reservaciones tiene como objetivo proveer al usuario de una manera fácil de crear una reservación. Esto se realiza a través de un formulario en el cual se pedirá la información básica del usuario, además de opciones para realizar *Check in*, *Check out* y efectuar una reservación en una fecha específica.

4.10.1. Proceso

Lo primero que se tiene que hacer es agregar los componentes que se van a utilizar, esto es:

- Botones:
 - Reservar
 - *Check in*
 - *Check out*
 - *Room*

- Labels:
 - *First Name*
 - *Last Name*
 - *Phone*
 - *Skype User*
 - *Email*

- TextBox
 - *First name*
 - *Last name*
 - *Area*
 - *Phone*
 - *User*
 - *Mail*

Los elementos que van a transmitir información deberán de ser agregados a la clase *viewController*, que en el presente caso se llama *ReservarViewController*. Esto será lo que contendrá el encabezado:

Declaración de las variables externas, que son utilizadas fuera del *viewController*, pero que son necesarias:

```
extern NSDate *extern_fecha_checkin;
extern NSDate *extern_fecha_checkout;
extern NSInteger extern_room_type;
extern NSInteger indicador_fecha_checkin;
extern NSInteger indicador_fecha_checkout;
extern NSInteger costo;
```

Declaración de las variables que enviarán información:

```
@property (strong, nonatomic) IBOutlet UILabel *label_checkin;
- (IBAction)reservar:(id)sender;
@property (nonatomic, strong) NSString *fecha_checkin;
@property (nonatomic, strong) NSString *fecha_checkout;
@property (strong, nonatomic) IBOutlet UILabel *label_checkout;
@property (strong, nonatomic) IBOutlet UILabel *label_room;
@property (nonatomic, strong) NSString *room_type;
@property (strong, nonatomic) IBOutlet UITextField *textfield_firstname;
@property (strong, nonatomic) IBOutlet UITextField *textfield_lastname;
@property (strong, nonatomic) IBOutlet UITextField *textfield_phone1;
@property (strong, nonatomic) IBOutlet UITextField *textfield_phone2;
@property (strong, nonatomic) IBOutlet UITextField *textfield_skype;
@property (strong, nonatomic) IBOutlet UITextField *textfield_email;
```

En la clase *ReservarViewController* se va a tener lo siguiente: en el método de *ViewDidLoad* deberán tenerse las dos fechas: *Check in* y *Check out*, que se obtendrán de otra vista.

```
if (indicador_fecha_checkin==1) {
    NSDateFormatter *form = [[NSDateFormatter alloc] init];
    [form setDateFormat:@"MM-dd-YYYY"];
    NSString *str = [form stringFromDate:extern_fecha_checkin];
    self.label_checkin.text=str;
}
if (indicador_fecha_checkout==1) {
    NSDateFormatter *form = [[NSDateFormatter alloc] init];
    [form setDateFormat:@"MM-dd-YYYY"];
    NSString *str = [form stringFromDate:extern_fecha_checkout];
    self.label_checkout.text=str;
}
```

El costo que se va a manejar es el siguiente:

```
self.label_room.text=@"Simple";
costo=80;
```

- \$ 80 por habitación simple.

```
self.label_room.text=@"Double";
costo=95;
```

- \$ 95 por doble

```
self.label_room.text=@"Suite";
```

Costo = 115;

- \$115 por suite

Se tendrá un método llamado "Reservar" donde se hará el cálculo del total de la reservación, y luego se mostrará una alerta de confirmación.

```
total = costo* numberOfDays;
```

El mensaje de confirmación es:

```
NSString *mensaje =[NSString stringWithFormat:@"Su reservacion sera  
por %i ", total];
```

```
confirmation = [[UIAlertView alloc] initWithTitle:@"Confirmation"  
message:mensaje delegate:self cancelButtonTitle: @"Cancel"  
otherButtonTitles:@"Confirm", nil];
```

```
[confirmation show];
```

Figura 26. Vista módulo de reservar



Fuente: elaboración propia.

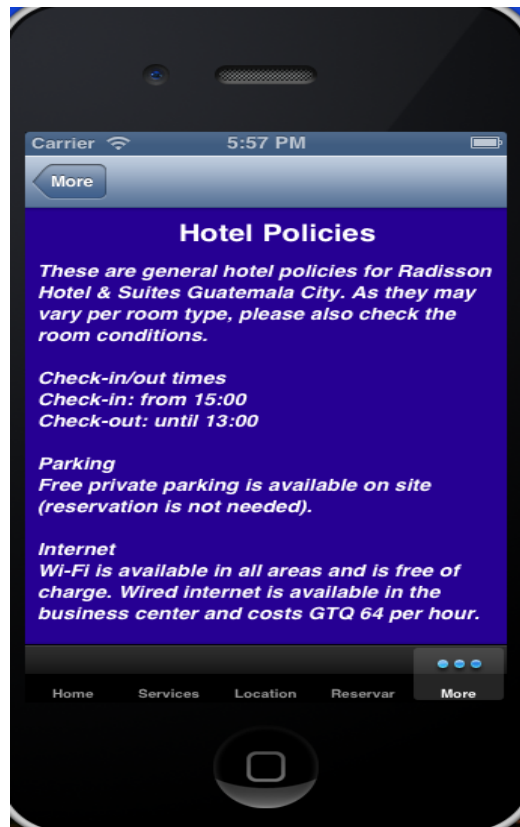
4.11. Desarrollo del módulo de Políticas

El módulo de “Políticas” tiene como objetivo dar a conocer las diferentes políticas de la empresa. La creación de este módulo es únicamente con un *TextView* agregado al *ViewController*, con la siguiente información:

- Horarios de *Check in* y *Check out*
- Información del parqueo
- Información sobre la conexión a internet
- Información sobre cargos a niños

- Información sobre camas extras

Figura 27. Vista políticas del hotel



Fuente: elaboración propia.

4.12. Consideraciones de implementación

Para implementar la solución es necesario disponer de una cuenta de desarrollador de Apple; esta se utiliza para generar el archivo *.ipa*: una vez generado el archivo *.ipa* se procede al envío y validación al *AppStore*, donde de ser aprobada la aplicación es publicada para su distribución: en caso de que sea para uso personal, con el archivo *.ipa* generado se procede a importarlo a la herramienta iTunes, para cargarlo al dispositivo iOS que se desee.

Por la naturaleza de la aplicación, es necesario contar con acceso a las diferentes API disponibles para la obtención de información; en el presente caso se pudo contar con acceso a la *API* de Google Places, la cual provee información de manera gratuita, pero que al incrementar considerablemente la cantidad de usuarios de la aplicación, es necesario realizar un pago anual, dependiendo de dicha cantidad.

CONCLUSIONES

1. A través del desarrollo de la aplicación se pudo verificar el proceso de creación de una herramienta para la publicidad, *marketing* y mercadotecnia de la industria hotelera, y ver cómo una aplicación de este tipo puede beneficiar a dicha industria.
2. Se creó una aplicación accesible a todos los usuarios, tomando en cuenta la calidad de la misma.
3. El entorno hotelero es un espacio con muchas posibilidades para la industria, ya que exige aplicaciones de alta calidad, al tener opciones muy competitivas en el mercado.
4. Se pudo crear una aplicación con el objetivo de promocionar un hotel específico, y así resaltar sus fortalezas y tomar nota para mejoras.
5. La solución que se plantea permite el registro de personas de manera más rápida y eficiente.

RECOMENDACIONES

1. Antes de empezar el desarrollo de la aplicación es necesario diseñar un bosquejo, conocer el lenguaje y realizar pruebas, para que la experiencia de desarrollo sea más agradable.
2. Utilizar dispositivos físicos para realizar las pruebas, ya que si bien es cierto el emulador tiene muchas funciones del dispositivo real, este está limitado en cuanto al uso de otras aplicaciones, conectividad, acelerómetro e interactividad.
3. Construir un repositorio en línea a manera de tener una copia de seguridad en caso de cualquier fallo o error irreversible en el desarrollo.

BIBLIOGRAFÍA

1. D'AMBRA, John; RICE, Ronald E. *Explaining perceived performance of the World Wide Web: uncertainty and the task-technology fit model*. [en línea]. <http://iris.nyit.edu/~kkhoo/Spring2008/Topics/TTF/000ExplainPercieve_Internet.pdf> [Consulta: marzo de 2014].
2. LLOPIS, Glenn. *Telecom industry depends on the hispanic market for future growth of wireless technology and mobile Apps*. [en línea]. <http://www.forbes.com/sites/glennllopis/2013/07/29/telecom-industry-depends-on-the-hispanic-market-for-future-growth-of-wireless-technology-and-mobile-apps/>. [Consulta: 12 de septiembre de 2013].
3. PILONE, Dan; PILONE, Tracey. *Head First iPhone and iPad Development: a learner's guide to creating objective-c applications for the iPhone and iPad*. 2a ed. Estados Unidos: O'Reilly Media, 2011. 674 p. ISBN: 14-493-8782-9.
4. RAY, John. *Sams teach yourself iOS 6 Application Development in 24 Hours*. 4a ed. Estados Unidos: Sams Publishing, 2013. 864 p. ISBN: 06-723-3443-7.
5. RIVERA, Janessa. *Gartner Says Smartphone Sales Accounted for 55 percent of overall mobile phone sales in third quarter of 2013*. [en línea]. <<http://www.gartner.com/newsroom/id/2623415>>. [Consulta: 2 de noviembre de 2013].

