



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador - CENEAC

Yo Culinario: aplicación móvil para la gestión de recomendaciones gastronómicas a nivel mundial

Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por los Bachilleres
Eduardo José, Moreno Correa CI: 19.829.461
Hugo Javier, Rojas Monzón CI: 19.649.326

Tutora:
Profa. Yusneyi Yasmira Carballo Barrera.

Ciudad Universitaria de Caracas, Mayo de 2015.

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR – CENEAC

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado “Yo Culinario: aplicación móvil para la gestión de recomendaciones gastronómicas a nivel mundial” presentado por los Bachilleres Eduardo José Moreno Correo C.I. V-19.829.461 y Hugo Javier Rojas Monzón C.I. V-19.649.326, a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día _____ de _____ de 2015, a las _____ horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en _____ de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de ____ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día ____ de _____ de 2015.

Prof. Yusneyi Yasmira Carballo Barrera

Tutora

Prof. Nestor Mendez

(Jurado)

Prof. Franky Uzcátegui

(Jurado)

Agradecimientos

Agradecemos haber culminado este Trabajo Especial de Grado, que por redundante que suene nos costó mucho esfuerzo dedicación tiempo y planificación para trabajar en equipo y poder solventar los problemas que se ocasionen y avanzar en paralelo las actividades planteadas.

En particular un agradecimiento a Nidal Barake, fundador de Yo Culinario. Gracias a ese incentivo que permitió desarrollar ideas y ejecutar este proyecto de Tesis.

Al equipo de trabajo de La Causa Agencia Móvil en particular al equipo de diseño quienes participaron en el diseño de las vistas dentro de la aplicación que fue parte fundamental de la maquetación de la aplicación móvil.

Un agradecimiento especial a nuestra tutora y amiga Yusneyi Carballo que a pesar de su gran carga de trabajo nos permitió realizar un trabajo de grado bajo su tutela, y nos guio por casi un año en esta difícil labor de trabajo de grado y seminario.

Un saludo y millones de gracias a mis padres Evelyn Monzón y Hugo Rojas junto a mis tíos Luis Monzón y Jazmín Monzón, no hubo un instante en que no sintiera el apoyo y fuerza necesaria para culminar el arduo Trabajo Especial de Grado.

Por último le brindo a mi abuela Raquel Bermúdez este logro, sé que estaría muy orgullosa de mí.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Enseñanza Asistida por Computador – CENEAC

Yo Culinario: aplicación móvil para la gestión de recomendaciones gastronómicas a nivel mundial

Autores:

Hugo Javier Rojas, C.I.: 19.649.326, correo-e: hugo.javierr@gmail.com,
Eduardo José Moreno, C.I.: 19.829.461, correo-e: guayoucv@gmail.com

Tutora: Profa. Yusneyi Carballo Barrera **Correo-e:** yusneyi.carballo@ciens.ucv.ve

Fecha: 08 de Mayo de 2015

RESUMEN

El presente Trabajo Especial de Grado está orientado al desarrollo de una aplicación móvil ideada para los amantes de la gastronomía y el buen comer, así como también a aquellas personas interesadas en conocer sitios de interés, especialmente establecimientos y restaurantes de comida gourmet, localizados en diversas partes del mundo. Esta aplicación permite leer reseñas destacadas de los restaurantes y también ilustrar al usuario con imágenes de alta calidad acerca del sitio; pudiendo incluir si así lo desea una lista personalizada de establecimientos favoritos para un rápido acceso. Para un mayor detalle, la aplicación permite ubicar de manera precisa a través de mapas de Google las coordenadas y geolocalización de dicho establecimiento. Además, si el usuario está interesado en difundir sus experiencias durante la interacción de la aplicación, la misma permite una integración con los servicios nativos del dispositivo para poder compartir a través de todas las aplicaciones instaladas, las fotos y comentarios de interés, permitiendo la divulgación de las reseñas por diversas redes sociales. Cabe destacar que fue necesario el desarrollo de un Sistema Manejador de Contenidos (CMS) que tuviera una conexión con una base de datos desarrollada en MySQL para brindar soporte, fidelidad y consistencia de datos de manera segura y garantizada, que van a servir de insumo a la aplicación móvil mediante el desarrollo de Servicios Web a través de una conexión síncrona con los servicios de AJAX, con el objetivo de desplegar en tiempo real y con un alto nivel de calidad los datos necesarios para que la aplicación pueda funcionar de manera ideal. El desarrollo del CMS fue implementado con el *framework* CakePHP bajo el patrón de diseño Modelo-Vista-Controlador (MVC), mientras que el desarrollo de la aplicación móvil fue construido con el *framework* de aplicaciones móviles y nativas Sencha Touch.

Palabras Clave: Arte Gastronómico, Gestor de contenidos, Aplicación móvil, Servicios nativos, Gestor de recomendaciones, Sistema Manejador de Contenidos.

Índice General

Introducción	I
CAPÍTULO I	1
1. Marco Teórico Conceptual	1
1.1 Definiciones	1
1.2 Herramientas para crear una aplicación híbrida	2
1.3 Componentes de un manejador de contenidos	3
1.4 Ejemplos de aplicaciones similares a Yo Culinario.....	3
1.5 Arquitecturas utilizadas en Yo Culinario	6
1.6 Fases para la creación de una aplicación híbrida.....	9
1.7 Tecnologías para el desarrollo de la aplicación móvil.....	10
1.8 Tecnologías para desarrollar un manejador de contenidos	11
1.9 Otras tecnologías utilizadas	12
1.10 Formato óptimo de imágenes.....	13
CAPÍTULO II	16
2. Marco Metodológico.....	16
2.1 Metodologías Ágiles.....	16
2.2 Metodología SCRUM	17
2.3 Eventos SCRUM.....	18
2.4 Artefactos SCRUM.....	19
2.5 Ventajas y desventajas de la metodología SCRUM	20
CAPÍTULO III	22
3. Desarrollo de la aplicación Yo Culinario.....	22
3.1 Objetivos de la aplicación.....	22
Objetivo General	22
Objetivos Específicos	22
3.2 Alcance de la aplicación.....	22
3.3 Metodología de desarrollo.....	23
3.4 Fase de levantamiento de requerimientos.....	24

3.5	Aplicación de la metodología	26
3.6	Interacción con el cliente, Sprints quincenales y mejoras en la aplicación	68
3.7	Dispositivos utilizados para probar Yo Culinario.....	73
3.8	Pruebas de Aceptación de Yo Culinario	75
3.9	Resultados obtenidos.....	76
	Resultados y Conclusiones	81
	Recomendaciones y Trabajo Futuros	82
	Referencias Bibliográficas	83
	Anexos.....	86
4.	Tecnologías de referencia	86
4.1	Anexo A: Yo Culinario	86
4.2	Anexo B: Tecnologías de desarrollo.....	87

Índice de Figuras

Figura 1 : Pantallas aplicación Degusta.	4
Figura 2: Pantallas aplicación Tangoo.	5
Figura 3: Arquitectura Cliente-Servidor.	6
Figura 4: Modelo Vista Controlador.....	7
Figura 5: Tecnologías utilizadas en AJAX.....	8
Figura 6: Modelo básico de un servicio Web.....	9
Figura 7: Comparación entre formatos de imágenes.....	14
Figura 8: Comparación PNG-8 JPEG.	14
Figura 9: Comparación entre iOS y Android en el 2014.	15
Figura 10: Ciclo SCRUM, artefactos y Sprint.	20
Figura 11: Fases de un Sprint de Yo Culinario.....	25
Figura 12: Vista inicial Yo Culinario.....	31
Figura 13: Vista por establecimientos Yo Culinario.	32
Figura 14: Vista del establecimiento Yo Culinario.	33
Figura 15: Vista del mapa Yo Culinario.....	34
Figura 16: Menús de Yo Culinario.....	35
Figura 17: Esquema de base de datos Yo Culinario.	37
Figura 18: Operaciones CRUD para Ciudades y Establecimientos.	42
Figura 19: Operaciones CRUD para Teléfonos e Imágenes.	42
Figura 20: Sistema de archivos Yo Culinario.	44
Figura 21: Ejemplo Servicio Web de Ciudades.	47
Figura 22: Ejemplo Carrusel documentación de Sencha.....	50
Figura 23: Ejemplo Carrusel editado en Sencha.	51
Figura 24: Ejemplo carrusel de imágenes de ciudades.....	55
Figura 25: Ejemplo carrusel de imágenes de establecimientos.....	55
Figura 26: Ejemplo distancia en establecimientos.....	58
Figura 27: Estructura directorios Sencha Touch en producción.	59
Figura 28: Diagrama de secuencia sistema de caché.....	60
Figura 29: Ventana emergente botón compartir.	67
Figura 30: Vista de ciudades sin categorización.	69
Figura 31: Vista de ciudades categorizadas.....	70
Figura 32: Vista del establecimiento sin carrusel.	70
Figura 33: Vista de establecimientos con carrusel.	71
Figura 34: Vista de ciudades sin carrusel.....	71
Figura 35: Vista de ciudades con carrusel.	72
Figura 36: Vista por establecimientos con 4 íconos.	72
Figura 37: Vista por establecimientos con 2 íconos.	73
Figura 38: Inspector de elementos de Google Chrome.....	74
Figura 39: Resultados de la pregunta 1.	76
Figura 40: Resultados de la pregunta 2.	76
Figura 41: Resultados de la pregunta 3.	77
Figura 42: Resultados de la pregunta 5.	78

Figura 43: Resultados de la pregunta 6.	78
Figura 44: Resultados de la pregunta 7.	79
Figura 45: Resultados de la pregunta 8.	79
Figura 46: Resultados de la pregunta 9.	80
Figura 47: Yo Culinario.	86
Figura 48: Logo Apache Cordova.	87
Figura 49: Sencha Touch.....	88
Figura 50: Logo CakePHP.	90
Figura 51: Logo Xcode.....	91
Figura 52: Logo mapas de Google.....	92
Figura 53: Ejemplo de un mapa de Google.....	92

Índice de Tablas

Tabla 1: Integrantes del equipo SCRUM.....	24
Tabla 2: Pila de productos de Yo Culinario.	30
Tabla 3: Reglas de validación del modelo de base de datos.....	39
Tabla 4: Descripción del módulo de autenticación de CakePHP.....	41
Tabla 5: Plugins utilizados en Yo Culinario.	65
Tabla 6: Cambios realizados por el cliente al presentar iteraciones del producto a realizar.	69
Tabla 7: Dispositivos de prueba IOS.....	73
Tabla 8: Dispositivos de prueba Android.....	74
Tabla 9: Componentes de navegación.....	89

Introducción

El estudio de la relación del hombre con su alimentación y su medio ambiente o entorno es una definición aceptada para el término "Gastronomía". El Gastrónomo por su parte es el profesional que se encarga de esta ciencia. A menudo se piensa erróneamente que el término gastronomía únicamente tiene relación con el arte culinario y la cubertería en torno a una mesa. Sin embargo, ésta es una pequeña parte del campo de estudio de dicha disciplina: no siempre se puede afirmar que un cocinero es un gastrónomo. La gastronomía estudia varios componentes culturales tomando como eje central la comida.

Por otra parte, la constante innovación tecnológica en el vivir diario nos lleva a formar una alianza inconsciente entre nuestras actividades cotidianas con aquellas piezas de software existentes que, eventualmente nos facilitan a concluir una carga considerable de las actividades que realizamos cotidianamente apoyado de esas virtudes que nos ofrecen para completar las tareas con una menor inversión de tiempo.

Los productos desarrollados en este Trabajo Especial de Grado permitirán difundir y compartir por los diversos medios sociales aquellas experiencias en el ambiente culinario que pueden estar a la mano con solamente tener acceso a su aplicación, ofreciendo así, una alternativa para expandir esta información por los diversos canales existentes en las redes de difusión para estar al tanto de aquellas culturas gastronómicas que se practican en cada rincón del planeta.

Este documento está estructurado de la siguiente manera:

En el Primer Capítulo se describen todos aquellos términos teóricos básicos, pero necesarios para comprender todo el dominio que abarca el presente Trabajo de Grado.

El Segundo Capítulo resume de forma precisa la conceptualización de la metodología SCRUM siguiendo a través de la práctica todas las bases teóricas en que se fundamenta dicha metodología con el fin de aprovechar sus numerosas ventajas y sus artefactos desarrollados, e incluso concluir los productos de manera rápida por ser esta, una metodología ágil.

El Tercer Capítulo abarca la identificación del problema, descripción de los objetivos generales y específicos, definición del alcance de los productos, aplicación de la metodología como estrategia para el desarrollo del Trabajo de Grado.

Finalmente, se presentan los resultados obtenidos, conclusiones, recomendaciones y trabajos futuros, referencias bibliográficas y finalmente algunos anexos referentes al Trabajo Especial de Grado.

CAPÍTULO I

1. MARCO TEÓRICO CONCEPTUAL

1.1 DEFINICIONES

- **Sistema Operativo Móvil:** Según (Rivera, 2012) un sistema operativo móvil “es el software que permite la interacción real con los dispositivos que podemos hacer a partir de las capacidades del hardware. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice”; también realiza un énfasis en que una de las cualidades más importantes de los sistemas operativos móviles “es la rapidez con la que en general se desempeña”.
- **Dispositivo Inteligente :** Es un dispositivo electrónico, por lo general conectado a otros dispositivos o redes a través de diferentes protocolos como Bluetooth, Campo de Comunicación Cercano o NFC, Wi-Fi, 3G, etc.; que puede funcionar hasta cierto punto de forma interactiva y autónoma (Wikipedia, 2013).
- **Aplicación Móvil:** Una aplicación móvil es un programa que se puede descargar y al que se puede acceder directamente desde un teléfono o desde algún otro aparato móvil, como por ejemplo una tableta.
- **Aplicaciones Híbridas:** Las aplicaciones híbridas, también denominadas *web Apps*, son aplicaciones optimizadas para que se vean bien a través de las pantallas de los dispositivos móviles. Las aplicaciones híbridas no se desarrollan en el lenguaje de programación nativo del sistema, sin embargo, cuando accedemos a ella, la estética es muy similar a las de las Apps nativas.
- **Manejador de Contenido:** Es una aplicación Web desarrollada para crear, editar, manejar, publicar y administrar el contenido de un sitio Web. Un sistema administrador de contenidos consiste de una interfaz que controla una base de datos donde se almacena el contenido de un sitio. Permite administrar todos los contenidos del sitio Web con la conveniencia de utilizar un simple navegador de Internet.
- **Marco de trabajo (*Framework*):** Es un esquema (un esqueleto, patrón o modelo) para el desarrollo y/o la implementación de una aplicación.
- **Formato de una imagen:** el formato de imagen o de archivo se refiere a la estructura de datos en que se guardará el documento, por ejemplo: jpeg, tiff, bmp (Manual del Photoshop, 2010). Cabe acotar que unos formatos de imagen son más livianos u ocupan menos espacio en memoria que otros.

1.2 HERRAMIENTAS PARA CREAR UNA APLICACIÓN HÍBRIDA

Para el desarrollo, maquetación y construcción de una aplicación híbrida se necesitan como mínimo tres componentes fundamentales, los cuales son:

- **Lenguajes de programación del lado del cliente:** Son todos aquellos lenguajes o herramientas de *software* que son directamente procesados e interpretados por un navegador web.
- **Framework orientado al desarrollo de aplicaciones híbridas:** Permiten a los programadores desarrollar aplicaciones para dispositivos móviles utilizando lenguajes del lado del cliente. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado¹ se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema), pero no se tratan tampoco de aplicaciones web (teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema móvil nativo). Estos *frameworks* permiten tener acceso a los elementos de *hardware* propios del dispositivo (como por ejemplo la cámara o el acelerómetro). Herramientas de este tipo como Apache Cordova (anexo 4.2) permiten tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas librerías se conectan al sistema operativo móvil usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas².
- **Dispositivos de prueba:** Estos artefactos permiten ejecutar el producto final mediante el hardware que poseen; proveen la visual mediante la cual se pueden depurar y verificar errores tanto como de lógica de construcción de la aplicación y fallas a nivel de interfaz de los elementos que la componen.

Combinando estos tres componentes fundamentales es posible realizar una aplicación híbrida la cual pueda ejecutarse en un dispositivo sin la necesidad de haber sido desarrollada bajo el lenguaje nativo del sistema operativo móvil en la cual se quiere utilizar.

¹ Renderizar: término usado para referirse al proceso de generar una imagen desde un modelo.

² Interfaz de funciones foráneas: Es un mecanismo por el cual un programa escrito en un lenguaje de programación puede realizar llamadas a funciones o usar los servicios escritos en otro.

1.3 COMPONENTES DE UN MANEJADOR DE CONTENIDOS

Con la finalidad de poder gestionar y alimentar de forma dinámica y con información al día la aplicación Yo Culinario tal como se presentó en el trabajo de investigación (seminario), se optó por la realización de un manejador de contenidos el cual partiendo de sus componentes fundamentales explicados a continuación pueda cumplir con las exigencias planteadas y permita al usuario administrador o editor de contenidos agregar de forma sencilla los contenidos que se desea mostrar a los usuarios finales de la aplicación.

- **Base de Datos:** Una base de datos es un repositorio centralizado de datos lógicamente relacionados, que permite almacenar y organizar hechos o eventos para producir información. Según (Ospina, 2009) la tecnología de base de datos fue desarrollada para superar las limitaciones y problemas de los sistemas de procesamiento de archivos. En los sistemas de procesamiento de base de datos se usa un enfoque centralizado de los datos para lo cual se añade un nuevo componente, el Sistema Manejador de Base de Datos (SMBD).³
- **Manejador de Contenido:** Es una aplicación Web desarrollada para crear, editar, manejar, publicar y administrar el contenido de un sitio Web. Un sistema administrador de contenidos consiste de una interfaz que controla una base de datos donde se almacena el contenido de un sitio.

1.4 EJEMPLOS DE APLICACIONES SIMILARES A YO CULINARIO

Considerando algunos casos de éxito a nivel nacional como internacional, se consideró importante analizar y detallar algunas aplicaciones móviles existentes para evaluar sus virtudes y características así como fallas y módulos faltantes en las mismas, que podrían agregar mayor valor a su producto.

1.4.1 Degusta

Aplicación móvil desarrollada para leer reseñas de establecimientos en Venezuela, donde se puede opinar, recomendar y compartir a través de las redes sociales las experiencias vividas por el usuario. Permite también subir fotos a la aplicación de algún plato en particular para dar un propio enfoque a través de fotografías. Degusta incluye un módulo de búsqueda categorizado por áreas de interés como localización, precio o incluso si aquellos restaurantes poseen puestos de estacionamiento. De igual manera ofrece a los usuarios la experiencia de compartir opiniones acerca de restaurantes que haya conocido dándole la oportunidad de dar una crítica y evaluar a los mismos con una escala de puntuación.

³ Conjunto de datos relacionados entre sí y un conjunto de herramientas de software (y/o hardware) para tener acceso a esos datos. Consiste de un conjunto de programas que son usados para definir, procesar y administrar la BD y sus aplicaciones.

Entre las características más significativas de Degusta encontramos:

- Subir fotos de platos a la aplicación.
- Evaluar, categorizar y comentar restaurantes.
- Filtros de búsqueda mediante parámetros.
- Ubicación mediante mapas de Google.
- Subir restaurantes a la aplicación.
- Agregar restaurantes a favoritos.
- Interacción con redes sociales.
- Puntuación de usuarios según comentarios y calificación.

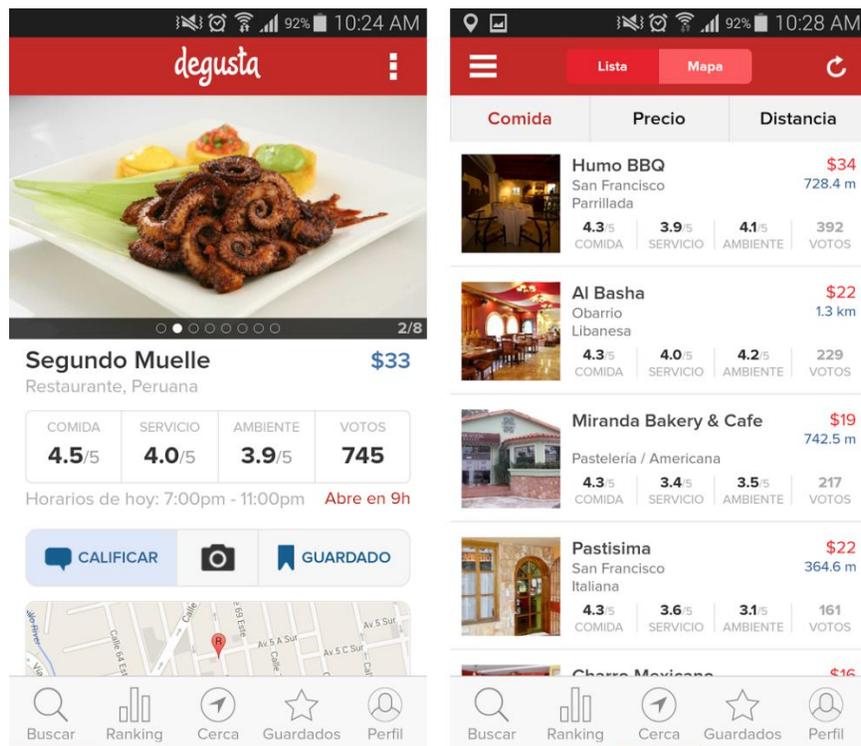


Figura 1 : Pantallas aplicación Degusta.
Fuente: (googleplay, 2014).

1.4.2 Tangoo

Aplicación móvil desarrollada para el sistema operativo iOS, que ofrece locales, restaurantes y bares en distintas ciudades de Canadá de acuerdo a un momento especial que se quiera experimentar y compartir, ya sea con amigos familiares o una pareja.

La característica de mayor impacto y atractivo de la aplicación es la selección detallada y acertada de su contenido multimedia, particularmente las fotos; desde que se inicia la aplicación ya ofrece un abanico de ciudades disponibles para elegir, también incluye numerosos carruseles de fotos por cada establecimiento que permiten al usuario elegir basándose de este poderoso valor agregado que posee Tangoo. Al igual que Degusta, Tangoo permite compartir

en diversas redes sociales como Facebook y Twitter experiencias vividas por los usuarios que quieren dar a conocer sus opiniones personales. Tangoo además permite ubicar los establecimientos a través de los mapas de Google para ofrecerle al usuario una perspectiva geográfica acerca de la ubicación del local.

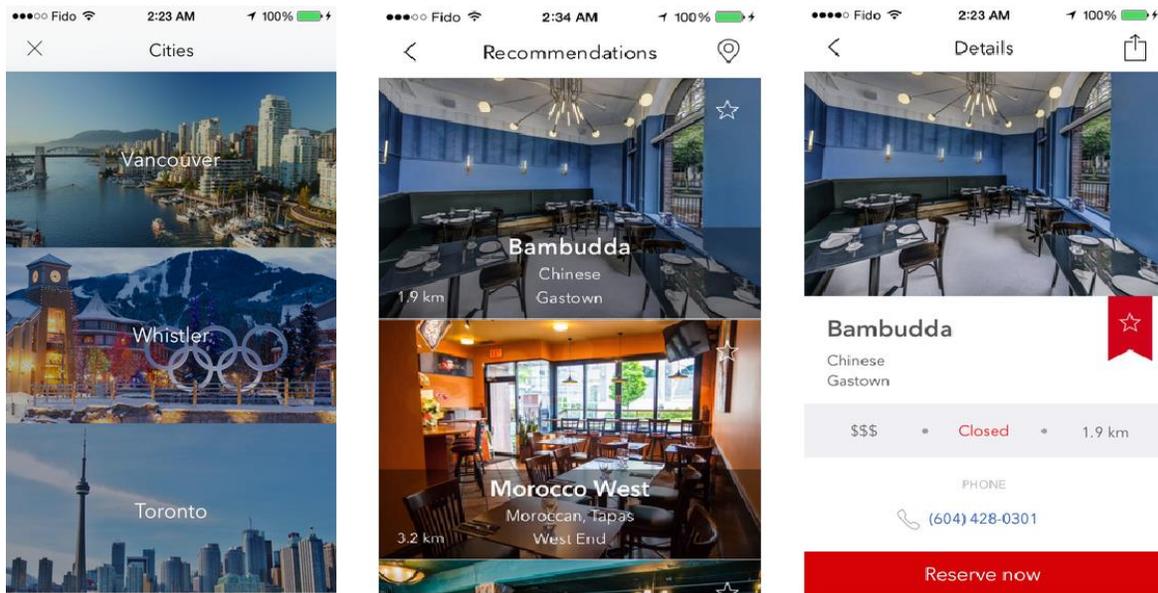


Figura 2: Pantallas aplicación Tangoo.
Fuente: (itunes.apple.com, 2015).

En la Figura 2 se presentan ejemplos de las interfaces de la aplicación Tangoo, se observa fotos de las diversas ciudades disponibles para conocer dentro de la aplicación como un conjunto de establecimientos con sus respectivas fotos representativas, y una vista particular de un establecimiento de ejemplo que sirve como referencia de la descripción del mismo.

1.5 ARQUITECTURAS UTILIZADAS EN YO CULINARIO

1.5.1 Arquitectura Cliente-Servidor

Consiste en términos sencillos en un cliente que realiza peticiones a otro programa (el_servidor) que las responde.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Las aplicaciones normalmente trabajan bajo este esquema debido a que es más fácil alojar los datos y algunas rutinas del lado del servidor, de forma que al realizar las peticiones obtengamos respuestas con los datos esperados sin requerir de procesamiento del lado del cliente, y así se logra distribuir la carga, tener una mayor eficiencia en nuestros programas.



Figura 3: Arquitectura Cliente-Servidor.

1.5.2 Patrón de Arquitectura de Software Modelo-Vista-Controlador (MVC)

Es un patrón de arquitectura que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

En el patrón MVC, las entradas del usuario, los modelos del mundo exterior y la retroalimentación visual son explícitamente separados y manejados por tres tipos de objetos, cada uno especializado para un conjunto de tareas específicas (Junta de Andalucía, S.F.).

- **Modelo:** Es la representación específica de la información con la cual el sistema opera. Por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos habitualmente están alojados en una base de datos relacional.
- **Vista:** Este presenta la data alojada en el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. Maneja el despliegue de los datos para un usuario final.
- **Controlador:** Es el encargado de manejar la lógica del negocio. Este responde a eventos, usualmente acciones del usuario e invoca peticiones al modelo y, probablemente, a la vista.

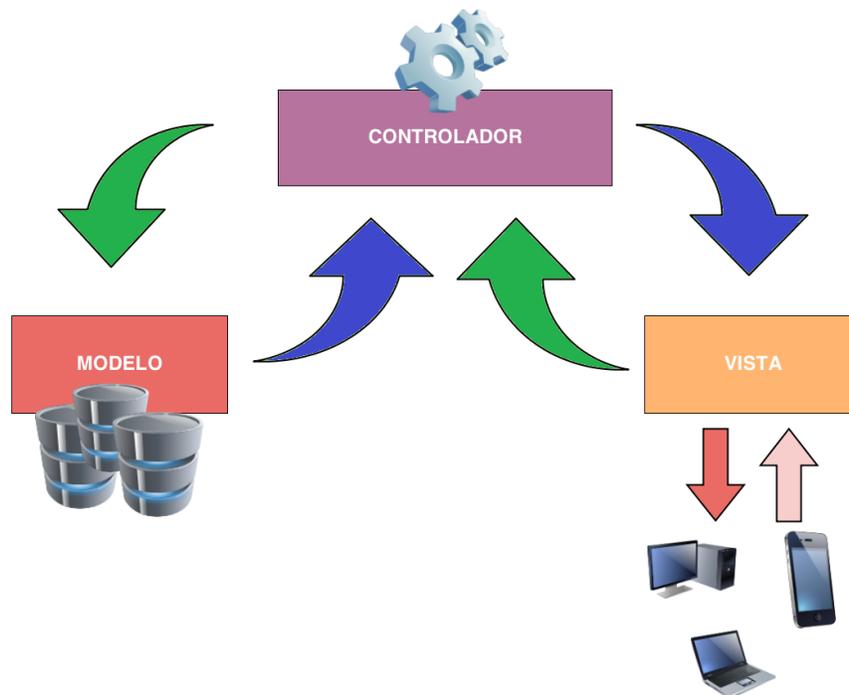


Figura 4: Modelo Vista Controlador.

1.5.3 AJAX

El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". Ajax no es una tecnología en sí misma, se trata de varias tecnologías independientes que se unen de una forma novedosa.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

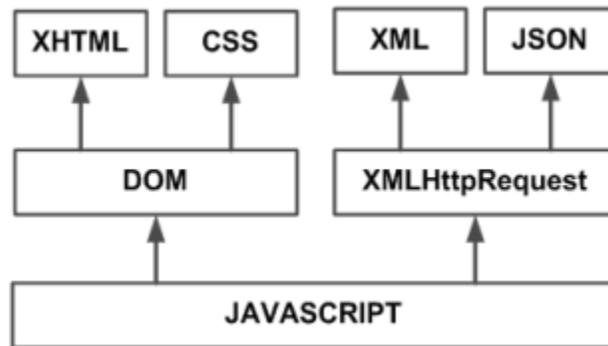


Figura 5: Tecnologías utilizadas en AJAX.
Fuente: (librosweb.es, 2010).

Cuando se combinan tecnologías en el modelo Ajax como en la figura 5, se obtienen ganancias de rendimiento y las aplicaciones funcionan mucho más rápido, ya que las interfaces de usuario se pueden actualizar por partes sin tener que actualizar toda la página completa.

1.5.4 Servicios Web

Los servicios WEB son tecnologías que utilizan un conjunto de protocolos de red y estándares que sirven para intercambiar datos entre aplicaciones, normalmente se encuentran alojados en un servidor al cual se le realiza una petición específica la cual puede o no requerir de parámetros de entrada, y ellos se encargan de retornar la información solicitada, ya sea realizando una consulta a una base de datos, o de manera directa.

Para el intercambio de información mediante servicios web podemos utilizar dos formatos para los datos, el primero es XML el cual es un lenguaje de marcado con ciertas similitudes a HTML, que nos permite agregar la data por etiquetas y en forma de árbol para que de esta manera podamos realizar recorridos y conseguir en las hojas del árbol la información que buscamos.

La segunda forma de recibir los datos es mediante el formato JSON, el cual es un formato de intercambio ligero que al igual que el XML puede ser leído fácilmente y es independiente de la plataforma, está basado en subconjunto de JavaScript por esta razón puede ser convertido a JavaScript rápidamente y por lo tanto es ideal para aplicaciones Web basadas en AJAX, pero la ventaja de JSON sobre el primero es su capacidad de representar la misma información en un formato más ligero que el XML y por lo tanto es más rápido de transportar y consume menos ancho de banda.

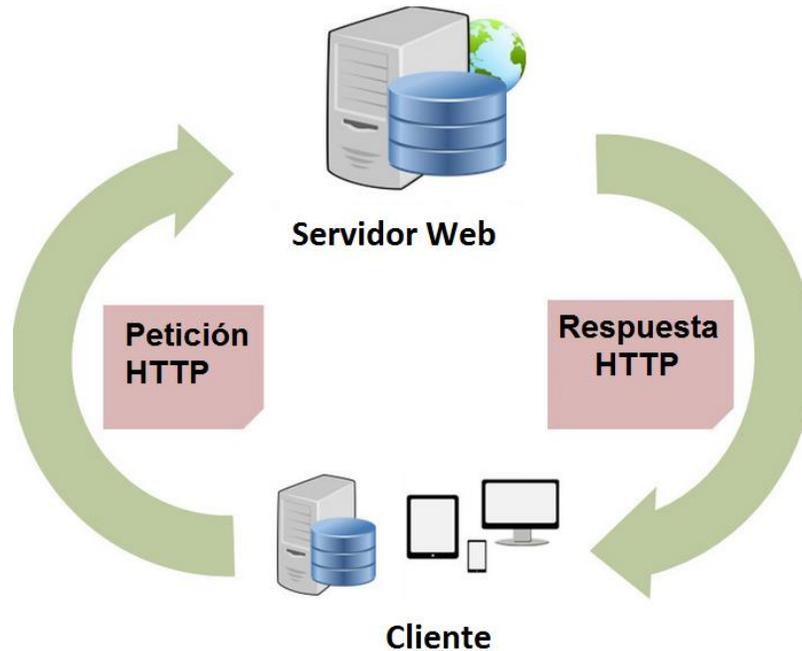


Figura 6: Modelo básico de un servicio Web.
Fuente: (chemaxon.com, 2013).

1.6 FASES PARA LA CREACIÓN DE UNA APLICACIÓN HÍBRIDA

Según el blog en Internet (Netizen, 2014) en su artículo “como-crear-tu-primera-app” es necesario llevar a cabo una serie numerada de pasos para lograr desarrollar una aplicación exitosa, los cuales son definidos a continuación:

1. Identificar una idea o problema: Tener una gran idea es el punto de partida en cada nuevo proyecto. Pero antes de entrar en los detalles de la aplicación, es necesario definir claramente el propósito y la misión de la aplicación. ¿Qué va a hacer? ¿Cuál es su atractivo principal? ¿Qué problema concreto se va a solucionar, o qué actividad se va a mejorar?
2. Dibujar la experiencia de la aplicación: A partir de este momento se le empieza a dar forma a la aplicación. Para ello lo más sencillo es realizar unos bocetos básicos sobre las diferentes funcionalidades de la aplicación. Ya sea con papel y lápiz o utilizando medios digitales como Photoshop, lo importante es dar forma a lo que se tiene en mente, para este paso resulta de utilidad fijarse en otras aplicaciones ya existentes que sirvan como inspiración y modelo del que se puedan extraer ideas.

3. Metodología a utilizar: El desarrollo de una aplicación es un producto de software que otros ya han desarrollado, por tal motivo se puede implementar una metodología que siga un patrón y un conjunto de procesos definidos para la realización de cada uno de los requisitos funcionales y no funcionales del producto de software.
4. Desarrollo de la aplicación: En esta etapa lo más relevante es realizar versiones que puedan ir siendo probadas y evaluadas constantemente, para que de esta manera se puedan ir resolviendo uno a uno los requerimientos planteados.
5. Pruebas de la aplicación: Esta fase es muy importante para garantizar el perfecto funcionamiento de la aplicación. Permitir que el mayor número posible de usuarios interactúe con la aplicación ya que ellos pueden encontrar nuevas necesidades o aportar opiniones que puedan mejorar su funcionalidad. Prueba todas y cada una de las funciones más de una vez de manera que te asegures que funciona correctamente. Apunta en una lista todo aquello que quieras mejorar gracias a la información obtenida durante las pruebas realizadas por ti mismo y por otros usuarios.

1.7 TECNOLOGÍAS PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL

Ya definidos los objetivos y evaluadas el conjunto de funcionalidades en el pasado Trabajo de Investigación Seminario, se han seleccionado el siguiente conjunto de herramientas y tecnologías para llevar a cabo la aplicación móvil:

- **Lenguaje de marcado HTML5:** Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un lenguaje de etiquetado (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.
- **Lenguaje de estilo CSS3:** Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML, CSS3 enfoca su funcionalidad a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado, entre otros. Con la combinación de HTML5 y CSS3 se pueden generar aplicaciones con estilos personalizados logrando así cumplir con los requerimientos de diseño planteados.
- **Lenguaje de programación JavaScript:** JavaScript permite agregar en las páginas web validaciones del lado del cliente y dinamismo para los elementos creados en HTML5, de esta manera se podrá contar con una aplicación que realice tareas dinámicamente en tiempo real, las cuales en su mayoría son iniciadas por peticiones del usuario.
- **Framework de desarrollo móvil Sencha Touch:** Es un *framework* que utiliza los estándares web de HTML5, CSS3 y JavaScript para desarrollar aplicaciones web para dispositivos móviles con la apariencia de ser nativos de los sistemas iOS, Android y BlackBerry. Sencha Touch además provee una extensa variedad de funcionalidades pre-cargadas que nos ayudan a facilitar las tareas más comunes a la hora de programar. La escogencia de esta herramienta va de la mano con la investigación previa realizada ya que en los resultados analizados es el *framework* para desarrollar aplicaciones híbridas con mejores tiempos de respuestas y el único que está pensado para trabajar con dispositivos táctiles incorporando a sus librerías una amplia cantidad de gestos

táctiles. Mediante esta herramienta se podrán desarrollar muchas de las funcionalidades previamente planteadas dando como resultado una aplicación con un alto grado de usabilidad y con tiempos de respuesta óptimos.

- **Framework de empaquetamiento y traducción Apache Cordova:** Una vez desarrollada la aplicación Web es necesario mediante algún mecanismo traducir este código fuente a código que pueda ser interpretado por los diferentes sistemas operativos móviles, en específico para la aplicación “Yo Culinario” los sistemas Android e iOS, esto se logra mediante la herramienta Apache Cordova la cual permite tomar el código desarrollado en Sencha y empaquetarlo de manera que pueda ser interpretado por el sistema operativo que se escoja, esto mediante una serie de librerías que se encargan de hacer llamadas a las funciones nativas de los sistemas. Una ventaja importante de utilizar Apache Cordova es la portabilidad que provee, debido a que las aplicaciones desarrolladas requieren una mínima cantidad de cambios para poder funcionar en diferentes dispositivos y sistemas operativos, tomando en cuenta que el código a generar es básicamente HTML5 con CSS3 y JavaScript. Además de esto, provee la alternativa de hacer llamadas nativas al hardware de los dispositivos para poder interactuar con elementos como por ejemplo la cámara o el acelerómetro; otra gran ventaja es la posibilidad de añadir *plugins*⁴ o herramientas de terceros que permitan realizar funcionalidades comunes, fomentando con esto la reutilización de código y la facilidad para resolver algunas tareas en específico.

1.8 TECNOLOGÍAS PARA DESARROLLAR UN MANEJADOR DE CONTENIDOS

- **CakePHP:** es un framework libre, de código abierto, para el desarrollo rápido de aplicaciones, estructurado bajo el patrón de arquitectura MVC. Con CakePHP el desarrollo web ya no es monótono porque ofrece las herramientas para escribir el código necesario. Cake se caracteriza por su sencilla rapidez y fácil uso, además de ser intuitivo y fácil de aprender heredando la sintaxis y semántica del lenguaje de programación PHP. Por otro lado ofrece soporte de operaciones estándares basado en el ORM⁵ que provee el *framework*.
- **Sistema manejador de base de datos MySQL:** Es un sistema de gestión de bases de datos relacional, basado en la política de código abierto, cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones; de manera que mediante el mismo podamos consultar el contenido alojado en un repositorio de datos con lo cual se pueda desplegar la información obtenida en las diferentes interfaces de la aplicación (Nieto, 2007). Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. MySQL en particular es un sistema de

⁴ Plugin: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de librerías.

⁵ ORM es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador. ORM relaciona las tablas de la Base de Datos en Objetos.

manejador de bases de datos bastante robusto y con una documentación amplia, esto debido a ser un software de código abierto.

- **jQuery:** Las tendencias en la programación siempre se van al lado de buscar herramientas o tecnologías que permitan facilitar el trabajo, y engloba de manera directa funciones o procedimiento que sean tediosos a la hora de ser realizado por los programadores (Desarrollo Web, 2009). jQuery es una librería JavaScript que permite realizar tareas que serían mucho más largas y difíciles de realizar en el lenguaje nativo, como tal no es una tecnología, pero su facilidad y eficiencia lo ha convertido en casi un estándar para la programación con JavaScript en el lado del cliente; tanto así que es la herramienta más usada a la hora de agregar dinamismo a los contenido realizados con HTML y CSS. jQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX.
- **jQuery UI:** jQuery UI es una biblioteca de componentes que permite implementar componentes diversos para generar interfaces de usuario y efectos visuales en páginas web, además de otras funcionalidades básicas para crear aplicaciones web enriquecidas. Como su propio nombre indica, está basado en el popular framework JavaScript y su desarrollo está desarrollado con las librerías de jQuery.

1.9 OTRAS TECNOLOGÍAS UTILIZADAS

Con las herramientas nombradas anteriormente es posible la creación de una aplicación híbrida que cumpla con las funcionalidades planteadas en el trabajo de seminario, no obstante para poder resolver con mayor facilidad y rapidez las mismas se optó por utilizar dos herramientas las cuales permiten a los programadores ahorrarse una serie de tareas las cuales pueden llegar a ser bastante engorrosas y con una complejidad en tiempo considerable.

- **Mapas de Google (Anexo 2.6):** Están disponibles tanto para web como para el desarrollo de aplicaciones en dispositivos móviles; ofrecen una completa plataforma para interactuar con los mapas agregando marcadores en cualquier ubicación del mapa y marcar rutas entre ellos. Además cuentan con una precisión en cuanto a coordenadas bastante elevadas y la cantidad de información y puntos de interés que podemos encontrar en ellos es casi inigualable. Yo Culinario cuenta con un módulo de localización de establecimientos para el cual utilizar los mapas de Google resulta conveniente ya que se puede resolver esta funcionalidad valiéndonos de esta herramienta.
- **Plugins para Apache Cordova:** Los *plugins* son la mejor forma de extender funcionalidad en Cordova permitiendo la ejecución de código nativo invocado desde JavaScript. A través de ellos podemos no solamente acceder a funcionalidades nativas del sistema, sino también delegar tareas a funciones predefinidas. El punto más importante de esto es que Apache Cordova nos permite instalar e insertar directamente en el código fuente *plugins* desarrollados por terceros los cuales pueden resolver tareas como compartir contenido en las aplicaciones instaladas en los dispositivos, acceder a la cámara, abrir dentro de la aplicación un navegador interno, entre otros; teniendo como resultado poder realizar tareas que tendrían un alto grado de complejidad de manera más sencilla. Cabe destacar que existe una comunidad amplia de programadores que se

dedican a crear estos repositorios de acceso público y cada día es más sencillo encontrar *plugins* para tareas más específicas.

1.10 FORMATO ÓPTIMO DE IMÁGENES

Las imágenes digitales se pueden guardar en distintos formatos. Cada uno se corresponde con una extensión específica del archivo que lo contiene. Los más utilizados en la actualidad son: BMP, GIF, JPG, TIF y PNG.

Para la aplicación Yo Culinario se deben tener las imágenes almacenadas en un formato que permita una rápida descarga ya que estas estarán alojadas en un servidor y se debe establecer una conexión para poder visualizarlas en los dispositivos. Tomando esto como premisa se evaluaron los formatos siguientes:

- **GIF** (*GraphicsInterchangeFormat* = Formato de Intercambio Gráfico): Utilizan un máximo de 256 colores, y son recomendables para dibujos con grandes áreas de un mismo color o de tonos no continuos. También si se muestra texto. Suelen utilizarse con gran frecuencia, ya que permiten contener transparencias y animación. En cambio, no están recomendados para fotografías, ya que se perderían colores, y al no tener áreas de color continuo, el archivo final sería mayor que por ejemplo un JPG.
- **JPG-JPEG** (*JointPhotographicExpertsGroup*=Grupo de Expertos Fotográficos Unidos): Estas imágenes pueden contener millones de colores, en un archivo comprimido de tamaño razonable. Por ejemplo, las imágenes que obtenemos de una cámara digital suelen estar en este formato. Por tanto, son especialmente indicadas para fotografías, o gráficos complejos, obteniendo mejores resultados que el GIF.
- **PNG** (*Portable Network Graphic* = Gráfico portable para la red): Se trata de un formato de compresión sin pérdida. Tiene varias versiones:
 - **PNG 8** es un formato de 256 colores muy similar al GIF, que en teoría obtiene archivos algo menores. También admite transparencias.
 - **PNG 24 y PNG 36**. Es un formato de color verdadero (24 o 36 bits), lo que hace que sea un archivo de tamaño algo mayor. Admite canal de transparencia alfa, lo que quiere decir que puede obtener distintos niveles de transparencia, a diferencia de PNG 8 o GIF que pueden ser totalmente transparentes o no.

desarrolloweb.com		Formatos gráficos para páginas web	
GIF	<ul style="list-style-type: none"> - Compresión sin pérdida - Comprime bien los dibujos - Paleta de colores variable - Hasta 256 colores - Permite transparencia - Permite animación - Alta compatibilidad <p>OPTIMIZACIÓN:</p> <ul style="list-style-type: none"> - Reducir paleta de colores 	JPG	<ul style="list-style-type: none"> - Compresión con pérdida - Comprime bien las fotos - Paleta de color real - Hasta 16 Millones colores - Sin transparencia - Sin animación - Alta compatibilidad <p>OPTIMIZACIÓN:</p> <ul style="list-style-type: none"> - Alterar calidad de la imagen
PNG	<ul style="list-style-type: none"> - Compresión sin pérdida - Comprime bien los dibujos - Paleta de colores variable - Hasta millones de colores - Permite transparencia - Sin animación - Menor compatibilidad <p>OPTIMIZACIÓN:</p> <ul style="list-style-type: none"> - Reducir paleta y más 		

Figura 7: Comparación entre formatos de imágenes.
Fuente: (desarrolloweb.com, 2001).

Una vez evaluados los diferentes tipos de formatos existentes observados en la figura 7, PNG-8 figura como la mejor opción debido a la menor cantidad en espacio que ocupan este tipo de archivos manteniendo también un alto contraste, lo cual permite no perder el atractivo de las imágenes que se quieren reflejar en la aplicación y obteniendo como ganancia mejores tiempos de respuestas ya que se optimiza la carga de datos.

En la figura 8 podemos observar una imagen JPEG transformada a PNG-8, donde se puede notar que la pérdida de calidad y nitidez en la imagen es mínima.

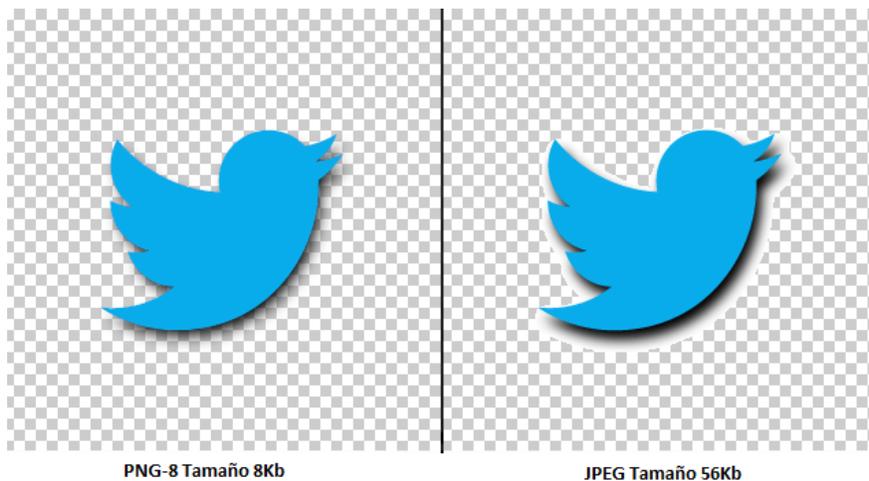


Figura 8: Comparación PNG-8 JPEG.

Con estas estas tecnologías se puede lograr una aplicación híbrida que funcione correctamente bajo los dos sistemas operativos más importantes de la actualidad según el autor (Merper, 2014) en su artículo con el título “ios-y-android-rentabilidad-contra-cuota-mercado-gráfico” acotan que en el mercado actual estos dos sistemas operativos tienen una penetración de hasta el 89% a nivel mundial, lo que quiere decir que entre ambos son líderes absolutos en el mundo de los dispositivos inteligentes. Con lo cual podemos avalar la decisión de realizar la aplicación Yo Culinario para estos dos sistemas, y de esta manera poder garantizar que estará disponible para el mayor número de usuarios en el mundo entero.

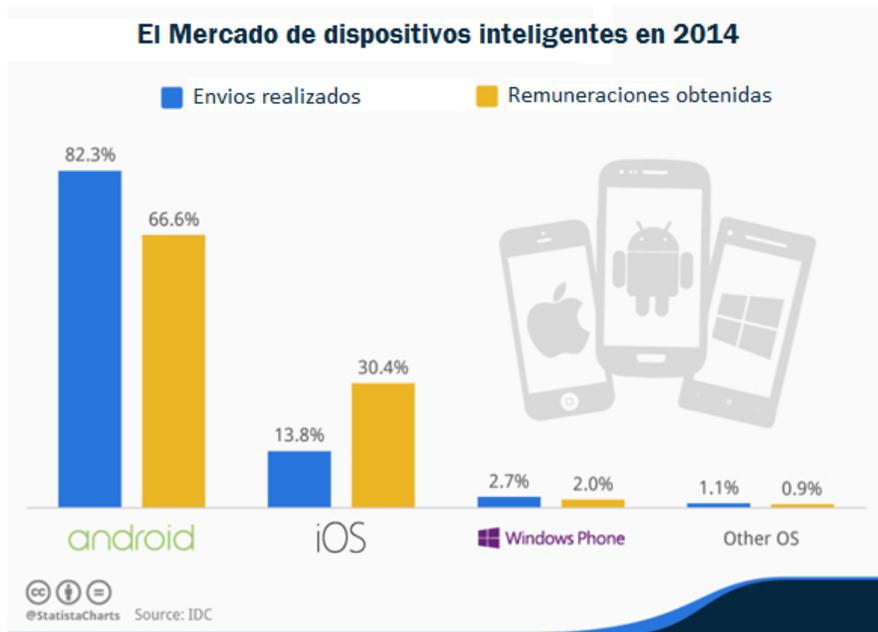


Figura 9: Comparación entre iOS y Android en el 2014.
Fuente:(hipertextual.com, 2014).

En la figura 9 podemos observar una gráfica comparativa de los sistemas operativos en dos puntos: el porcentaje de mercado a nivel global durante 2014 basado en el número de envíos, y el mismo porcentaje en 2014 basado en la cantidad de beneficios obtenidos, Android domina el mercado mundial con un 82,3% de los envíos, mientras que iOS se queda solamente con un 13,8%, y dejando el resto de las migajas para otros sistemas operativos. Sin embargo, en términos de beneficios, iOS consigue un 30,4% de beneficios totales del mercado, mientras que Android se queda “sólo” con un 66,6% de los mismos.

CAPÍTULO II

2. MARCO METODOLÓGICO

La realización de un producto de software puede ser una actividad compleja, que comprenda un número elevado de etapas teniendo en cada una de ellas diferentes grados de dificultad, lo que trae como consecuencia la dificultad de organización y lineamientos que deben seguir los desarrolladores del producto, de manera que la actividad a realizar se convierte en caótica y utilice mayor número de recursos de los necesarios, lo cual lo conlleva a un desarrollo ineficiente.

Por esta razón es vital la utilización de una metodología de software que permita además de obtener un esquema de trabajo y una serie de lineamientos a seguir, agilizar los procesos de desarrollo trabajando de manera más estructurada para obtener mejores resultados y beneficios en el producto final.

En este capítulo se describe la metodología de desarrollo a utilizar, tomando como base y pensando siempre en la aplicación a desarrollar, una metodología ágil, en particular la metodología SCRUM, para que mediante esta se dé cumplimiento al desarrollo de la aplicación a proponer.

2.1 METODOLOGÍAS ÁGILES

Un proceso es ágil cuando el desarrollo de software es incremental (entregas pequeñas de software, con ciclos rápidos), cooperativo (cliente y desarrolladores trabajan juntos constantemente con una cercana comunicación), sencillo (el método en sí mismo es fácil de aprender y modificar, bien documentado), y adaptable (permite realizar cambios de último momento).

La idea fundamental de las metodologías ágiles, es mostrar versiones parciales funcionales al cliente en intervalos cortos de tiempo, de manera que se pueda ir evaluando y sugiriendo cambios en el producto a medida se va desarrollando.

Las metodologías ágiles aparecen para corregir las fallas y problemas que se presentan al utilizar las metodologías de desarrollo tradicionales. Esta metodología tiene la capacidad de dar respuestas muchos más rápidas y accesibles a los problemas que surgen al realizar cambios en los productos de software que se desarrollan, además tiene como punto a favor para este tópico, que integra de manera directa y constante una comunicación entre los clientes y los desarrolladores, para que luego de cada interacción se pueda evaluar las prioridades del proyecto logrando con estos cumplir de manera más eficientes los objetivos del proyecto planteados (Garzas, Javier Garzas, 2013).

2.2 METODOLOGÍA SCRUM

Es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. SCRUM no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar (Clariss, 2008).

El marco de trabajo SCRUM consiste en los Equipos SCRUM, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el correcto uso y posterior éxito de esta metodología.

SCRUM se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

2.2.1 Participantes

1. **Representante o Dueño del Producto:** es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos SCRUM e individuos. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto. La gestión de la Lista del Producto incluye:
 - Expresar claramente los elementos de la Lista del Producto.
 - Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible.
 - Optimizar el valor del trabajo desempeñado por el equipo de desarrollo.
 - Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y, asegurar que el equipo de desarrollo entienda los elementos de la Lista del Producto al nivel necesario.
 - El Dueño de Producto podría hacer el trabajo anterior, o delegar en el equipo de desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.
2. **Facilitador:** es el responsable de asegurar que la dinámica es entendida y adoptada. Se aseguran que el equipo trabaja ajustándose a la teoría, prácticas y reglas de trabajo. Es un líder que está al servicio del equipo. El facilitador ayuda a todos a modificar estas interacciones para maximizar el valor desarrollado.
3. **Equipo de desarrollo:** Son los encargados de desarrollar y cumplir lo que les asigna el dueño del producto. Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del equipo de desarrollo.
Características del Equipo de Desarrollo:

- Son auto organizados. Nadie indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en incrementos de funcionalidad potencialmente desplegados.
 - Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto.
 - Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.
4. **Cliente:** Recibe el producto y puede influir en el proceso, entregando sus ideas o comentarios respecto al desarrollo (SCRUM.org, 2013).

Servicios del facilitador al dueño del producto:

- Encontrar técnicas para gestionar la Lista de Producto de manera efectiva.
- Entender la planificación del producto en un entorno empírico.
- Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor.

Servicios del Facilitador al Equipo de Desarrollo:

- Guiar al Equipo de Desarrollo en ser auto organizado y multifuncional.
- Ayudar al Equipo de Desarrollo a crear productos de alto valor.
- Eliminar impedimentos para el progreso del proyecto.
- Guiar a los miembros en el entorno de organizaciones en las que SCRUM aún no ha sido adoptado y entendido por completo.

Servicios del Facilitador a la Organización:

- Liderar y guiar a la organización en la adopción de SCRUM.
- Planificar las implementaciones de SCRUM en la organización.
- Motivar cambios que incrementen la productividad del Equipo SCRUM.
- Trabajar con otros SCRUM Masters para incrementar la efectividad de la aplicación de SCRUM en la organización.

2.3 EVENTOS SCRUM

En SCRUM existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas. Todos los eventos son bloques de tiempo, de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

2.3.1 El Sprint

El corazón de SCRUM es el Sprint, es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Durante el Sprint:

- No se realizan cambios que puedan afectar al objetivo del Sprint.
- Los objetivos de calidad no disminuyen.
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante (Ken Schwaber, 2013).

Un Sprint puede ser cancelado antes de que el bloque de tiempo llegue a su fin. Solo el dueño de producto tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de los interesados, del equipo de desarrollo o del facilitador.

2.4 ARTEFACTOS SCRUM

- **Pila de Productos:** Es una lista de deseos sobre las funcionalidades del producto. Es elaborado por el dueño del producto y las funciones están priorizadas según lo que es más y menos importante para el negocio. El objetivo es que el dueño del producto responda la pregunta “¿Qué hay que hacer?”.
- **Pila de iteración Sprint:** Es un subconjunto de tareas de la pila del producto, que son seleccionados por el equipo para realizar durante el Sprint sobre el que se va a trabajar. El equipo establece la duración de cada Sprint.
- **Reunión de planificación:** Esta reunión se hace al comienzo de cada Sprint y se define cómo se va a enfocar el proyecto que viene de la pila de producto las etapas y los plazos.
- **Reunión Breve:** Es una reunión breve que se realiza a diario mientras dura el periodo de Sprint. Se responden individualmente tres preguntas: ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Qué ayuda necesito? El SCRUM Master debe tratar de solucionar los problemas u obstáculos que se presenten.
- **Revisión:** Se revisa el sprint terminado, y ya debería haber un avance claro y tangible para presentárselo al cliente.
- **Fase de Revisión:** El equipo revisa los objetivos cumplidos del Sprint terminado. Se anota lo bueno y lo malo, para no volver a repetir los errores. Esta etapa sirve para implementar mejoras desde el punto de vista del proceso del desarrollo.

Scrum the framework

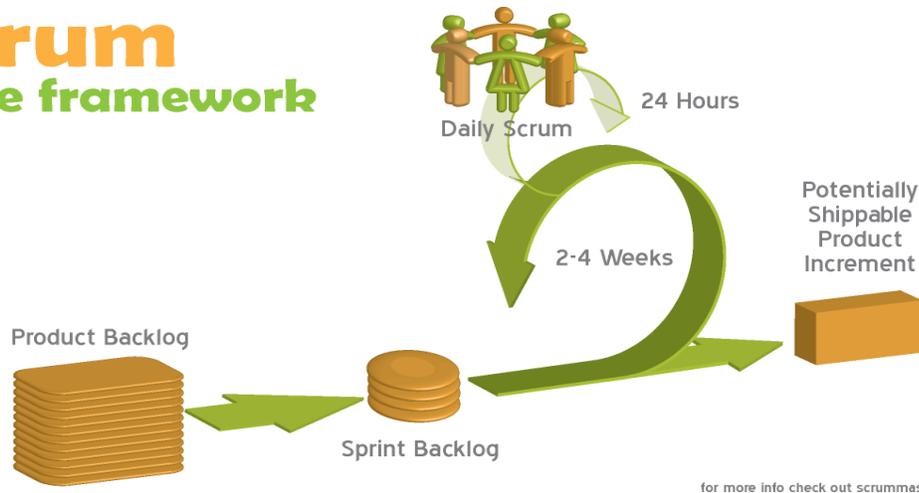


Figura 10: Ciclo SCRUM, artefactos y Sprint.
Fuente: (diegosalama.com, 2010).

2.5 VENTAJAS Y DESVENTAJAS DE LA METODOLOGÍA SCRUM

Tras analizar la metodología SCRUM, se procede a enumerar y analizar las ventajas y desventajas encontradas, porque a pesar de ser una herramienta bastante útil para agilizar la realización de los proyectos de software la misma pueda traer consigo alguno inconvenientes a tomar en cuenta (Ortí, 2013).

- La primera y la más relevante, es que estas metodologías ofrecen una rápida respuesta a cambios de requisitos a lo largo del desarrollo del proyecto gracias a su proceso iterativo, es tan importante realizar una buena recolecta de requisitos, como después poder modificarlos evitando grandes pérdidas en cuanto a costes, motivación, tiempo, entre otros. El cliente, si quiere colaborar, puede observar cómo va avanzando el proyecto, y por supuesto, opinar sobre su evolución gracias a las numerosas reuniones que realiza el equipo con el cliente. Esto le da tranquilidad.
- Uniendo las dos anteriores, se puede deducir que al utilizar estas metodologías, los cambios que quiera realizar el cliente van a tener un menor impacto, ya que se va a entregar en un pequeño intervalo de tiempo un pequeño “trozo” del proyecto al cliente, y si éste quiere cambiarlo, solo se habrá perdido unas semanas de trabajo. Con las metodologías tradicionales las entregas al cliente se realizaban tras la realización de una gran parte del proyecto, eso quiere decir que el equipo ha estado trabajando meses para que luego un mínimo cambio que quiera realizar el cliente, conlleve la pérdida de todo ese trabajo.
- Importancia de la simplicidad al eliminar trabajo innecesario.
- Utilizando esta metodología se obtiene una mayor productividad, ya que se motiva de manera directa al equipo de trabajo.
- Es más sencillo realizar predicciones de tiempos, debido a que se conoce la velocidad media del equipo por iteración, y es más fácil estimar cuándo se dispondrá de una funcionalidad determinada.

Pero claro, aunque las ventajas sean muy llamativas, estas metodologías también presentan inconvenientes que hay que asumir cuando se decide trabajar con ellas. Estos son:

- Falta de documentación del diseño. Al no haber documentación es el código (junto con sus comentarios) lo que se toma como documentación.
- Toma como premisa que el cliente está fuertemente involucrado en el desarrollo, y revisa frecuentemente el avance de las funcionalidades desarrolladas, pero en realidad los clientes participan poco y evitan dedicar tiempo y recursos para revisar pequeños avances en el desarrollo.
- Problemas derivados de la comunicación oral. No hace falta decir que algo que está escrito “no se puede borrar”, en cambio, algo dicho es muy fácil crear ambigüedad.
- Fuerte dependencia de las personas.
- Se requiere de un experto en la metodología que monitorice su cumplimiento.
- Falta de reusabilidad derivada de la falta de documentación.
- Plantea un problema si el desarrollo está restringido por una fecha de entrega y un precio de entrega cerrados por un contrato.
- Problemas derivados del fracaso de los proyectos ágiles. Si un proyecto ágil fracasa no hay documentación o hay muy poca; lo mismo ocurre con el diseño. La comprensión del sistema se queda en las mentes de los desarrolladores.

Se seleccionó SCRUM como metodología de desarrollo principalmente a que es una metodología ágil, logrando el desarrollo rápido y efectivo de software, por medio de iteraciones cortas de desarrollo. A diferencia de otras metodologías ágiles, SCRUM permite enfocarse por adelantado en los aspectos más valiosos y de mayor riesgo del proyecto, minimizando los errores y permitiendo la prevención a los fallos generando robustez al proyecto.

SCRUM es un proceso de la Metodología Ágil que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa. Entre las ventajas se encuentran la productividad, calidad y además se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes están unidos, comunicados y que el cliente tenga información continua del desarrollo de las funcionalidades del proyecto.

Además de las prestaciones que provee SCRUM a la hora de realizar software de mayor calidad, su elección también está basada al fuerte interés del cliente con el producto de manera que el mismo tenga cierto grado de tranquilidad al poder proveer entregas parciales luego de cada iteración.

CAPÍTULO III

3. DESARROLLO DE LA APLICACIÓN YO CULINARIO

3.1 OBJETIVOS DE LA APLICACIÓN

Objetivo General

Desarrollar una aplicación móvil para los sistemas operativos Android y IOS, mediante la cual los amantes de la cocina culinaria puedan ver y compartir contenidos y recomendaciones.

Objetivos Específicos

Con la finalidad de lograr el objetivo general, se establecen los siguientes objetivos específicos:

1. Aplicar la metodología SCRUM para el desarrollo de la solución utilizando un *framework* de desarrollo para aplicaciones (Sencha Touch).
2. Realizar un diseño de base de datos para dar soporte a la aplicación.
3. Diseñar, maquetar e implementar una interfaz gráfica.
4. Construir un manejador de contenidos que permita al administrador agregar, modificar y eliminar información de utilidad para la aplicación.
5. Desarrollar los servicios Web que proveen información entre la base de datos y la aplicación móvil.
6. Implementar un mecanismo que permita difundir información a través de distintos medios.
7. Elaborar un módulo de geolocalización.
8. Realizar pruebas de aceptación tanto en el área de diseño como en el área de desarrollo.

3.2 ALCANCE DE LA APLICACIÓN

En este Trabajo Especial de Grado se ha definido para la aplicación que se desarrolló como solución el siguiente alcance:

1. La aplicación Yo Culinario estará disponible únicamente en los sistemas operativos iOS y Android.
2. La disposición geográfica de cada establecimiento está definida por ciudades.
3. Yo Culinario está desarrollado para funcionar dispositivos inteligentes excluyendo iPad y tabletas.
4. La aplicación consta con un menú de ciudades y otro para los establecimientos favoritos del usuario.
5. Solo se podrá compartir la información de establecimientos en las redes sociales instaladas en el dispositivo.
6. Para cada establecimiento se puede tener entre una y cinco imágenes a lo sumo.
7. Se ubicará por defecto la ciudad más cercana a la localización actual del usuario.

8. La aplicación funcionará exclusivamente en modo vertical (*portrait*).
9. Los establecimientos están categorizados en cuatro rubros distintos: restaurantes, compras, café-pastelería y bares.
10. Por un período de 15 días cada ciudad agregada tendrá un icono distintivo en el menú de ciudades.
11. Añadir un establecimiento a favoritos solo podrá realizarse dentro de la sección representativa del local.
12. Dentro de la sección del establecimiento se ofrece un enlace al sitio Web correspondiente.
13. Es necesaria una conexión a Internet para el correcto funcionamiento de la aplicación.
14. La gestión del administrador de contenidos será manejado por un único usuario.
15. La búsqueda dentro del administrador de contenidos está restringida por establecimientos.

3.3 METODOLOGÍA DE DESARROLLO

Dada las ventajas que ofrece la metodología SCRUM mencionada en el Capítulo II y el conjunto de actividades que se plantearon en el trabajo de investigación (seminario) como objetivos funcionales y no funcionales para el desarrollo del manejador de contenidos y de la aplicación móvil, la estrategia de desarrollo aplicando las bases teórico prácticas que brinda la metodología es la siguiente:

1. Realizar un completo análisis del dominio para así, levantar los requerimientos funcionales y no funcionales para el desarrollo de ambos artefactos.
2. Desarrollar una pila de producto que contengan los productos en pequeñas instancias independientes y modulares para desarrollar con la intervención del dueño del producto, el facilitador y el equipo de desarrollo.
3. Definir un número finito de actividades y componentes independientes a desarrollar dentro de la aplicación.
4. Asignar cada uno de los componentes como un Sprint que formará parte de la pila de producto.
5. Establecer orden de prioridades y tiempos de desarrollo para cada uno de los Sprints, definidos por el facilitador con el equipo de desarrollo.
6. Planificar reuniones breves o diarias entre el facilitador y el equipo de desarrollo con el fin de solucionar aquellos problemas u obstáculos que se presenten.
7. Realizar reuniones de planificación al comienzo de cada sprint para definir los alcances y objetivos a desarrollar.

A continuación el equipo que formó parte de la implementación práctica de la metodología (ver Tabla 1)

Miembro	Rol dentro de la Metodología
Nidal Barake	Cliente
Hugo Rojas	Equipo de Desarrollo
Eduardo Moreno	Equipo de Desarrollo
Gerente General	Dueño del Producto
Director Creativo	Facilitador o SCRUM <i>Master</i>

Tabla 1: Integrantes del equipo SCRUM.

3.4 FASE DE LEVANTAMIENTO DE REQUERIMIENTOS

Es necesario aclarar que SCRUM, más que una metodología de desarrollo *software*, es un método de gestión de proyectos, el cual puede adaptarse a cualquier tipo de proyecto y no únicamente a los de desarrollo de *software*. Aplicado a este último, está basado en el modelo de las metodologías ágiles, incrementales, compuestas por iteraciones y revisiones continuas. El objetivo principal es elevar al máximo la productividad de del equipo de desarrollo y reducir las actividades no orientadas a producir *software* funcional; con lo cual se consiguen resultados en períodos cortos de tiempo.

3.4.1 Desarrollo de Yo Culinario

Para el proyecto Yo Culinario, en un Sprint 0 se planificaron 10 Sprints, cada una con una duración variada dependiendo de la complejidad de la funcionalidad a resolver. Con cada Sprint se buscaba incrementar las funcionalidades agrupadas en módulos de la aplicación. Con la planificación inicial se pudo definir una pila de producto detallada más adelante, con lo cual se logró constituir la base y tiempos estimados para el proyecto.

La finalización de cada Sprint dio como resultado una versión estable del producto, con el incremento de las funcionalidades planificadas, las mismas que eran presentadas al dueño del producto. Para ello, es recomendable que cada requisito planificado se completase en una única iteración (incluyendo pruebas).

3.4.2 Fases para la creación de Yo Culinario

Cada iteración puede entenderse como un pequeño proyecto individual; en cada iteración se repite un proceso de trabajo similar (iterativo) para proporcionar un resultado completo sobre el producto final, así el dueño del producto obtiene los beneficios que se fueron desarrollando de forma incremental.

La ejecución de cada Sprint del proyecto puede dividirse en cinco fases, similares al ciclo de vida del modelo en cascada, como se muestra en la figura 11.

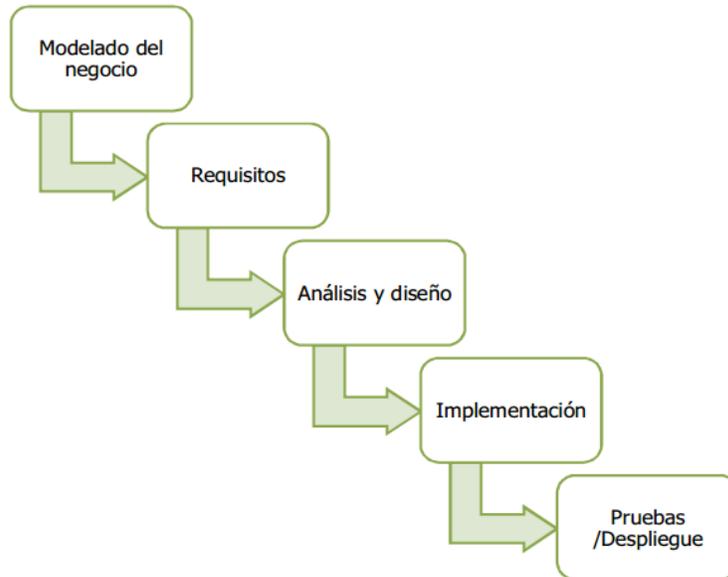


Figura 11: Fases de un Sprint de Yo Culinario.

Modelado del negocio

El modelado del negocio tiene como objetivo el comprender y describir de forma simplificada la realidad de las necesidades a resolver.

Esta fase se llevó a cabo durante el Sprint 0, el cual tiene como objetivo analizar la lógica de la solución, los requerimientos, plantear una arquitectura base y planificar en forma general la solución a desarrollar. Durante el desarrollo de cada Sprint planificado, se realizaron afinamientos y correcciones a los objetivos específicos de cada uno.

Requisitos

Esta fase tiene como propósito especificar las funcionalidades que serán implementadas durante cada Sprint.

En el Sprint 0 se especificaron las funcionalidades de toda la aplicación, mientras que en cada Sprint se analizó de forma detallada los requerimientos específicos, según los objetivos planteados.

Análisis y diseño

En el análisis se intenta descubrir que es lo que realmente se necesita, para llegar a una comprensión adecuada de los requerimientos (¿qué hacer?). El diseño representa las características que permitirán la implementación de los requerimientos en forma efectiva (¿cómo hacerlo?).

Respecto al diseño, en el Sprint 0 se planteó una arquitectura candidata, la misma fue ratificada o modificada, según el análisis de los requerimientos a implementar en cada Sprint.

Implementación

En esta etapa el equipo de desarrollo se encarga de implementar las funcionalidades necesarias, de acuerdo a las especificaciones analizadas y según el diseño planteado.

Esta etapa se pasa por alto en el Sprint 0 debido a que el objetivo de este Sprint es planificar y analizar el proyecto como tal.

Pruebas / Despliegue

La etapa de pruebas tiene como objetivo garantizar el correcto funcionamiento de las funcionalidades desarrolladas en cada uno de los Sprint.

Durante la ejecución del proyecto, la etapa de pruebas se lleva a cabo tanto por el equipo de desarrollo como por los usuarios del producto que forman parte del equipo de trabajo.

Con cada prueba se corrigen fallas y se realiza una versión incremental de cada funcionalidad, para poder lograr un funcionamiento óptimo de cada requerimiento realizado en el Sprint.

Tomando como base estas 5 (cinco) etapas, se logró realizar una pila un desarrollo completo de cada uno de los Sprint, con lo que se logró llevar a cabo la culminación de la aplicación Yo Culinario, siendo la misma revisada constantemente por el dueño del producto y teniendo como consecuencia partes incrementales de la misma.

3.5 APLICACIÓN DE LA METODOLOGÍA

El primer paso a realizar en SCRUM, es definir cada uno de los Sprint que serán realizados y dar un tiempo estimado de cada uno de ellos. Para Yo Culinario cada Sprint representa una funcionalidad del sistema asociándole una prioridad con respecto a su peso correspondiente en el proyecto.

Sin embargo antes de poder empezar a desarrollar y analizar cada Sprint por separado se debe realizar un análisis de levantamiento de requerimientos para la aplicación y se deben establecer los diseños gráficos que servirán de modelo para la implementación de las vistas del producto.

En este orden de ideas, se realizó un Sprint 0 en el cual se enfatizó todo el análisis tanto de la lógica del negocio de la aplicación como también de los diseños gráficos propuestos, con la finalidad de poder tener como resultado una lista de requerimientos funcionales y no funcionales para la aplicación, además de poder contar ya con una pila de productos

organizada que sirva como referencia con la finalidad de ir desarrollando Sprint por Sprint hasta conseguir el producto final.

3.5.1 Sprint 0

En el Sprint 0 se realizó el modelo del negocio, cuyo objetivo es comprender y describir de forma simplificada la realidad de las necesidades a resolver.

En este Sprint participaron todos los miembros del equipo de SCRUM, ya que cada opinión y detalle cuenta, de manera que con esto se pudieran tener definidas cada una de las facetas y detalles con los que contaría la aplicación.

La principal premisa es la necesidad de poder mostrar de forma ordenada, concisa y atractiva a los usuarios la información recolectada por el dueño del producto, referente a establecimientos localizados en diferentes ciudades del mundo.

Se llegó a la conclusión de realizar una aplicación híbrida utilizando el *framework* de desarrollo Sencha Touch y el *framework* de integración y empaquetamiento Apache Cordova, que tendrá una conexión mediante servicios Web en formato JSON con un servidor remoto, con la idea de poder cargar los datos de manera dinámica, entendiendo que los mismos pueden ser actualizados y modificados de manera constante.

Para la gestión de la información se optó como mejor opción desarrollar un manejador de contenidos mediante el cual, el administrador de la aplicación pueda a través de una interfaz intuitiva y accesible realizar operaciones básicas de agregar, modificar, verificar y eliminar los datos que serán mostrados en las diferentes interfaces de la aplicación. Para este tópico y siguiendo el lineamiento que se investigó en el trabajo de seminario se utilizó CakePHP como herramienta de desarrollo.

Como primer punto se hizo un levantamiento de los requerimientos funcionales y no funcionales, tomando como base los ya planteados en el trabajo de investigación del seminario, pero teniendo una gran cantidad de cambios debido a su mayoría a las exigencias presentadas por el dueño del producto. Los requerimientos definidos fueron los siguientes:

Requerimientos Funcionales:

- Realizar un módulo de autenticación para el usuario administrador en el manejador de contenidos.
- Generar botones que permiten al usuario agregar, modificar, visualizar y eliminar tanto ciudades como establecimientos en el manejador de contenidos.
- Implementar en el manejador de contenidos la funcionalidad de cargar entre 1 y 5 imágenes por cada establecimiento.
- Elaborar servicios Web que estén disponibles y retornen la información pertinente a las vistas de la aplicación.
- Construir un módulo de búsqueda en el manejador de contenidos para ubicar de forma más sencilla los establecimientos ya agregados.

- Implementar un servicio que se ejecute en segundo plano que permita cambiar el estatus de una ciudad de nuevo a viejo después de tener más de 15 días en la aplicación.
- Conectar servicios Web con la aplicación.
- Crear carruseles de imágenes para las ciudades y los establecimientos de cada ciudad.
- Configurar un módulo de favoritos para guardar establecimientos.
- Crear un módulo que permita ubicar cada establecimiento en un mapa de Google.
- Cada vez que se ejecute la aplicación debe elegir por defecto la ciudad más cercana a la localización del usuario.
- Guardar de manera interna en el dispositivo información con el fin de poder visualizar la aplicación sin conexión a Internet.
- Desarrollar un botón que permita compartir la información de un establecimiento a través de las redes sociales instaladas en el dispositivo.
- Crear un botón para visitar la página Web de cada establecimiento.
- Identificar en el menú de ciudades si una ciudad fue añadida recientemente, resaltando la misma con una imagen particular.

Requerimientos no Funcionales:

- **Desempeño y escalabilidad**
 La aplicación debe contener un alto contenido gestual, sin que esto retrase el desempeño de la misma, permitiendo a los usuarios interactuar obteniendo tiempos de respuesta lo más eficiente posible.
 La información debe ser obtenida y presentada en la interfaz sin retrasos significativos. Ejemplo: ante una petición del usuario para ver la información de un establecimiento, la misma, no debe tomar un lapso de espera de más de 30 segundos (considerando las limitaciones de ancho de banda existente).
- **Disponibilidad y confiabilidad**
 La aplicación debe acceder a la información que requiera el usuario en cualquier momento. En caso de no ser posible el acceso a la aplicación, se debe notificar al usuario la no disponibilidad del servicio para que intente nuevamente después de un lapso de tiempo.
 Al tener problemas para cargar alguna información solicitada a través de un servicio Web, se debe notificar al usuario del problema.
 Debe responder ante posibles errores que puedan presentarse notificando al usuario lo sucedido.
- **Usabilidad**
 Debe presentar una interfaz intuitiva para los usuarios, de manera que la curva de aprendizaje sea mínima.

Las acciones deben ser similares a las convenciones que se han establecido con el tiempo. Ejemplo: efectos de menú que se trasladan de izquierda a derecha, botones de control general para desplazarse en vistas, entre otros.

- Portabilidad
Está aplicación debe funcionar perfectamente bajo los dos sistemas operativos móviles propuestos, iOS y Android.

Luego de ser establecidos los requerimientos presentados, se dispuso a crear la pila de productos de SCRUM, la cual está constituida por el número del Sprint, una descripción de la actividad, el tiempo estimado y por último la prioridad que tiene dicha actividad.

Cabe destacar que una vez creada la pila de productos se pueden calcular tiempos estimados para el desarrollo del proyecto, dando como consecuencia conocer la disponibilidad en tiempo estimada para poder finalizar el producto y definir los roles y tareas que va a realizar cada recurso del equipo de desarrollo.

Recapitulando las definiciones y conceptos básicos acerca de la metodología SCRUM, a continuación se enumeran una lista de deseos (ver Tabla 2) sobre los requerimientos funcionales de los productos realizados. Con el fin de, enfatizar, detallar y establecer un orden jerárquico y orden de prioridades sobre estos subproductos realizados.

#Sprint	Descripción Actividad	Tiempo Utilizado	Prioridad
Manejador de Contenidos			
1.0	Desarrollar interfaces visuales	16 días	Alta
2..0	Establecer mecanismos de asociación entre las tablas para el soporte de relaciones de pertenencia entre entidades.	2 días	Alta
2.1	Desarrollar un módulo de seguridad que permita la autenticación de los usuarios administradores.	6 días	Alta
2.2	Implementar funciones que permitan realizar operaciones con la Base de Datos (CRUD)	10 días	Alta
3.0	Crear una arquitectura de directorios y renombramiento de ficheros del lado del servidor.	5 días	Alta
4.0	Implementar los Servicios Web en formato JSON por cada modelo de datos en el manejador de contenidos que estén disponibles para la aplicación.	10 días	Alta
4.1	Elaborar un módulo de búsqueda dentro del manejador de contenidos para el rápido acceso de los establecimientos.	3 días	Baja

5.0	Desarrollar un servicio que se ejecute en segundo plano que verifique la cantidad de tiempo que tiene una ciudad en la base de datos.	2 días	Media
Aplicación Móvil			
6.0	Asignación de Video Tutoriales y ejemplos de práctica al equipo de desarrollo para lograr un acercamiento con la tecnología a utilizar.	10 días	Baja
7.0	Desarrollar interfaces visuales de la aplicación Yo Culinario.	22 días	Alta
8.0	Establecer una comunicación con Sencha Touch y el manejador de contenidos para consumir la información mediante los Servicios Web que provee el servidor.	4 días	Alta
8.1	Crear carruseles de imágenes relacionados a las ciudades y establecimientos.	9 días	Media
9.0	Desarrollo de un módulo de establecimientos favoritos.	5 días	Media
9.1	Implementar un módulo de geolocalización que permita calcular distancia entre 2 puntos y determinar ubicación actual.	3 días	Alta
9.2	Desarrollar un módulo que permita el correcto funcionamiento de la aplicación en un estado sin conexión a Internet.	5 días	Media
Framework de integración Apache Cordova			
10.0	Creación del proyecto e integración de <i>plugins</i>	1 día	Media
10.1	Elaborar un módulo que permita compartir información en las redes sociales instaladas en el dispositivo.	7 días	Alta
10.2	Realizar una integración del código fuente de Yo Culinario con el proyecto de Cordova.	1 día	Baja

Tabla 2: Pila de productos de Yo Culinario.

Tomando como referencia la tabla de la pila de productos, se tiene como tiempo estimado para la realización del proyecto un total de 115 días, los cuales pueden o no aumentar teniendo en cuenta la cantidad de iteraciones que quiera realizar el cliente por cada funcionalidad; además se toma un período de alrededor de entre 5 y 10 días para realizar las pruebas de aceptación necesarias para el producto final.

En cuanto a las interfaces de la aplicación y ya teniendo definida lo que es la pila del producto, se evaluaron varios diseños tomando en cuenta y como referencia las aplicaciones gastronómicas nombradas y estudiadas en los antecedentes del trabajo de seminario, llegando con esto a proponer y elegir las siguientes interfaces para Yo Culinario:

Vista Inicial



Figura 12: Vista inicial Yo Culinario.

Es un carrusel de imágenes con una franja en el medio la cual tiene el nombre de la ciudad con una imagen de fondo representativa a la misma. Haciendo desplazamientos laterales tanto a la derecha como a la izquierda se puede cambiar de ciudad, las mismas están ordenadas de manera alfabética y cada una cuenta con una imagen alusiva.

Vista por ciudad



Figura 13: Vista por establecimientos Yo Culinario.

Esta vista tiene un carrusel de imágenes por cada establecimiento dentro de la ciudad seleccionada; cada carrusel contiene entre 1 y 5 imágenes correspondientes al establecimiento, además se agrega el nombre del mismo, el tipo y la distancia con respecto al usuario en kilómetros.

Vista de establecimiento

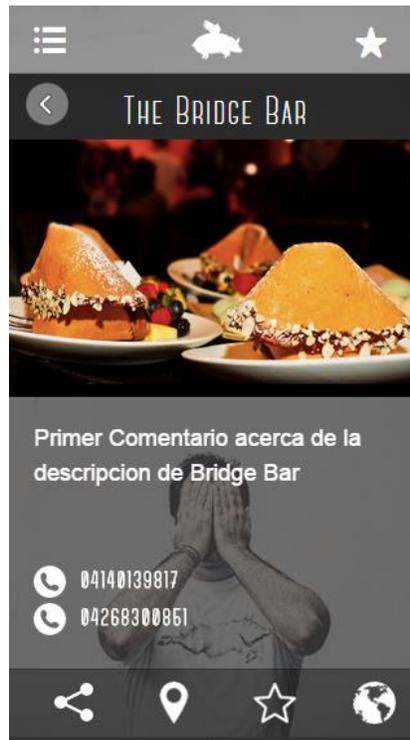


Figura 14: Vista del establecimiento Yo Culinario.

Esta vista describe el establecimiento seleccionado, en ella tenemos varios elementos, primero la descripción o comentario que hace el dueño del producto de su experiencia en el establecimiento.

Segundo se tiene el mismo carrusel de imágenes de la vista anterior, para degustar las imágenes representativas del local; tercero se cuenta con información de los números de teléfonos del establecimiento para poder realizar un contacto directo si existe el interés del usuario.

Por último se cuenta con una barra en la parte inferior donde se puede compartir en las redes sociales información del establecimiento, también se puede añadir a favoritos, visitar su página Web o localizar en un mapa la ubicación de este establecimiento.

Vista en el mapa



Figura 15: Vista del mapa Yo Culinario.

Como se visualiza en la figura 15 se puede ver posicionado el establecimiento en un mapa de Google, teniendo la posibilidad de hacer énfasis en buscar puntos de referencia para poder llegar al mismo.

No obstante, la aplicación también ofrece la posibilidad de abrir la localización en la aplicación de Google Maps, con la cual se obtiene un número más elevado de funcionalidades como por ejemplo establecer una ruta entre la localización y la del establecimiento. Cabe destacar que, si el usuario desea ubicar el establecimiento mediante la aplicación de Google Maps, esta debe estar preinstalada en el dispositivo.

Menús de la aplicación

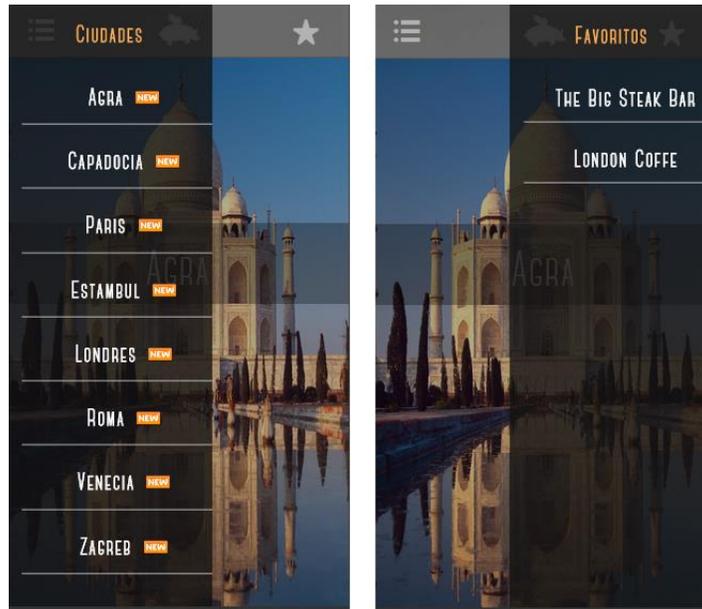


Figura 16: Menús de Yo Culinario.

La aplicación cuenta con dos menús, uno para la lista de ciudades ordenadas alfabéticamente y señalando si la ciudad ha sido añadida recientemente a la aplicación; y el otro es el menú de favoritos donde podemos ir directamente desde cualquier vista de la aplicación a los establecimientos que se tengan marcados.

Una vez evaluadas y aprobadas las interfaces por el equipo se completa el análisis del modelo de negocios de la aplicación con lo cual se empiezan a desarrollar uno a uno los Sprints definidos en la pila de productos.

3.5.2 Sprint 1.0

Requerimiento: Desarrollar interfaces visuales del manejador de contenidos

Número de Sprint: 1.0

Prioridad: Alta

Como punto de partida, el manejador de contenidos es el primero producto a desarrollar dentro de este Trabajo Especial de Grado, y como tópico inicial se estructuró este Sprint como el desarrollo de las interfaces visuales del CMS.

En resumen, se desarrollaron diversas vistas totalmente funcionales las mismas basadas en un esquema común con una gama de colores entre blancos y grises y el color naranja para destacar lo importante. Los detalles de las vistas se muestran a continuación:

Autenticación: Ofrece un una sección que le permite al usuario ingresar con sus credenciales de acceso al manejador.

Ciudades: Es la vista principal del servidor, con la mayor disposición de elementos visuales que incluye, un módulo de búsqueda, un botón de cerrar sesión, y una lista de todas las ciudades agregadas por el usuario administrador con las operaciones CRUD al alcance inmediato. Por especificar más, cada ciudad en sí es un acordeón que permite categorizar los establecimientos según a su categoría, al igual que las ciudades estos establecimientos tienen al alcance las operaciones CRUD identificadas con un icono blanco expresado en una metáfora para reforzar su significado.

Establecimientos: En esta vista se plasma información detallada del establecimiento seleccionado, al igual que en la vista de ciudades están al alcance las operaciones CRUD sobre los establecimientos. Esta vista también ofrece las mismas operaciones que los establecimientos pero para las imágenes y teléfonos respectivos del sitio. Para facilitar la visualización de las imágenes se construyó un carrusel para mostrar las imágenes representativas al establecimiento.

Agregar: Tanto para las ciudades como para los establecimientos se desarrolló una vista sencilla con un formulario con todos los campos pertenecientes al modelo asociado, con el fin de agregar ya sea una ciudad o un sitio al manejador.

Editar: Se hace una instancia de la vista Agregar con la única diferencia que la misma recibirá un id como parámetro con la información ya en los campos del formulario disponible a que el administrador los cambie y se actualice tanto en el manejador como en la Base de Datos.

Servicios Web: Es una interfaz común para todos los Servicios Web que presenta la información del modelo en un formato JSON.

3.5.3 Sprint 2.0

Requerimiento: Establecer mecanismos de definición y asociación entre las tablas para el soporte de relaciones de pertenencia entre entidades

Número de Sprint: 2.0

Prioridad: Alta.

Siguiendo el esquema Modelo Vista Controlador, un Modelo es la representación lógica de los datos, y en CakePHP es una representación de una instancia de una tabla de Base de Datos en un objeto, esto significa que un objeto tiene una correlación directa con una tabla en un esquema de Base de Datos, en un ambiente orientado a objetos la manipulación de los mismos como instancias de una clase ofrecen múltiples ventajas que se pueden considerar de suma importancia. Primeramente, se crea una capa de abstracción para el desarrollador ya que se elimina la barrera de conocer la sintaxis y el lenguaje de un sistema manejador de base de datos específico.

Modelo de Base de Datos propuesto

Una vez comprendido el dominio del problema y con todas las variables necesarias para desarrollar este complejo sistema que involucra un servidor de base de datos un estándar para la conexión entre el cliente y el manejador de contenidos, a continuación se propone la siguiente arquitectura de base de datos la cual dará soporte a todas la solicitudes e interacciones entre los entes mencionados.

- Sistema Manejador de Base de Datos (SMBD): MySQL
- Numero de Tablas: 5
- Nombres de las Tablas:
 1. cities (Ciudades)
 2. establishments (Establecimientos)
 3. users (Usuarios)
 4. pictures (Imágenes)
 5. phones (Teléfonos)
- Tablas relacionadas:
 1. 1 Ciudad tiene muchos Establecimientos.
 2. 1 Establecimiento tiene muchas Imágenes.
 3. 1 Establecimiento tiene muchos teléfonos.

Para seguir la convención de nombres de CakePHP se asignaron los nombres de las tablas como se definieron anteriormente para aprovechar la relación automática entre tablas, usando los plurales como nombre de las mismas. A continuación se muestra en la figura 17 el esquema de base de datos de Yo Culinario.

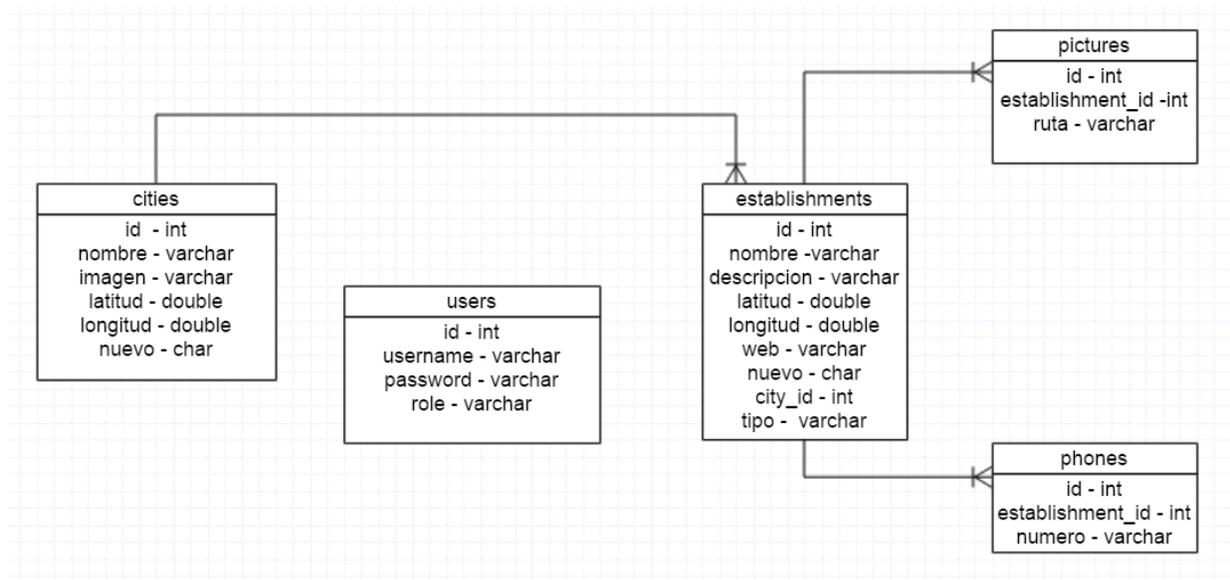


Figura 17: Esquema de base de datos Yo Culinario.
Fuente: (gliffy.com, 2015).

Al igual que se mencionó anteriormente, tanto los nombres de las tablas como de algunos atributos en específicos aquellos que son clave foránea⁶ en otra tabla tienen un nombre en particular para seguir la convención de CakePHP y este pueda ser fácilmente asociado por el *framework* al momento de hacer la relación entre tablas.

Relación entre tablas

Una vez hecho la relación entre tablas de manera física en el sistema manejador de base de datos, en el *framework* CakePHP también hay que hacer una correlación entre tablas para poder hacer operaciones lógicas a nivel de programación y que pueda persistir esa relación entre los modelos. Particularmente, el *framework* provee unas sencillas reglas que facilitan la relación entre modelos de manera rápida y eficaz:

- **hasMany:** Indica que una instancia del modelo A tiene muchas instancias en el modelo B
- **belongsTo:** Indica que una instancia del modelo B pertenece a una única instancia del modelo A
- **Recursive:** Es una definición numérica del nivel de profundidad que el *framework* puede navegar para recuperar datos entre multi-tablas relacionadas.
 - -1: Datos del mismo modelo sin *joins*.
 - 0: Datos del mismo modelo y a los que pertenece (*Belongs To*).
 - 1: Datos del mismo modelo, al que pertenece y a los que está asociado con la propiedad *hasMany*.
 - 2: Datos del mismo modelo, a los asociados y a la relación muchos a muchos.

Validación de Datos

CakePHP ofrece una importante validación de datos antes de interactuar y hacer operaciones directamente contra la Base de Datos, por ser un *framework* es necesario que este incorpore un mecanismo de consistencia y seguridad para garantizar la calidad de la información. Particularmente en CakePHP al definir un modelo como una clase y una descripción detallada de sus atributos, se puede también aplicar reglas de validación en pocas líneas de código antes de realizar operaciones sobre el modelo y así liberar la carga de desarrollo para el programador.

Algunas de las reglas de validación que se usaron en el desarrollo del CMS de Yo Culinario se observan en la Tabla 3.

⁶ Clave foránea: Es una identificación relacional de una tabla que está siendo referenciada por otra.

Tabla	Campo	Regla de Validación	Significado
cities	nombre	'/^[a-z]{4,}\$'/i'	Caracteres alfanuméricos mayores de 4 letras.
cities	imagen	array('extension', array('jpeg', 'png', 'jpg'))	Solo formatos jpeg png y jpg válidas
cities	latitud, longitud	notEmpty	Campos obligatorios no vacíos.
users	username, password	notEmpty	Campos obligatorios no vacíos.
phones	numero	array('minLength', 8)	Al menos 8 dígitos para el teléfono.
establishments	tipo	array('inList', array('Restaurante', 'Bar', 'Compras', 'Café/Pastelería'))	Permite solo elegir una de las opciones desplegadas.
establishments	web	array('url', true)	Validar si es una dirección Web bien formada.

Tabla 3: Reglas de validación del modelo de base de datos.

De esta forma, y con el soporte que ofrece el *framework* la etapa de pruebas se vio reducida una vez que la estructuración entre las tablas, con las correctas definiciones de los atributos y las relaciones entre ellas mediante los operadores soportados por el *framework* se pudo completar el Sprint 2.0 de manera satisfactoria para una presentación limpia y precisa al cliente.

3.5.4 Sprint 2.1

Requerimiento: Desarrollar un módulo de seguridad que permita la autenticación de los usuarios administradores.

Número de Sprint: 2.1

Prioridad: Alta.

Un sistema de autenticación de usuarios es una parte común de muchas aplicaciones web. Por esta razón como requerimiento de la pila de productos fue necesario el desarrollo de un módulo de seguridad que permitiera solo el ingreso de los usuarios administradores a través de credenciales de autorización con el fin de gestionar y editar los contenidos.

En CakePHP existe un módulo sencillo ya desarrollado. La esencia del componente de autenticación es comprobar si el usuario tiene una cuenta con el sitio. De ser así, el componente da acceso completo al sitio.

Cuando los datos de los usuarios viajan vía POST⁷ desde un formulario, el componente de autorización de Cake encripta y codifica automáticamente el contenido de la contraseña en un código *hash*⁸ que el *framework* por configuración brinda para ofrecer medidas de seguridad (Alexander, 2010).

Un ejemplo de una contraseña codificada con un código *hash* podría ser la siguiente:

Nombre de usuario: HugoJavier

Password: HugoJavier

Password con Hash: \$2a\$10\$Sjz5fClhykJTdsOwb6zhDbJdgtg85jpGpmc2cht4f433

Cabe destacar que el módulo de autenticación de usuarios es, sin duda el módulo de mayor importancia en el desarrollo de este Trabajo Especial de Grado, debido que se espera la manipulación de altos volúmenes de datos que, además de ser sensibles y de vital importancia para la retroalimentación con la aplicación móvil, estos deben tener un mecanismo de seguridad con un alto grado de garantía para ofrecer consistencia y confiabilidad en los datos que van a ser desplegados a miles de usuarios que interactúen con la aplicación.

Por defecto, el AuthComponent espera que se tenga una tabla llamada 'users' con campos llamados 'username' y 'password'

Código	Funcionalidad	Ubicación en el proyecto
<pre>var \$components =array('Auth');</pre>	Obliga a autenticar al usuario antes de acceder a las vistas.	Controlador que administra el acceso a los usuarios.
<pre>\$this->Auth->allow('index', ' logout);</pre>	Vistas autorizadas de acceso público.	Controlador que administra a los usuarios, en una función antes de cargar las vistas.
<pre>\$this->redirect (\$this- >Auth>logout());</pre>	Permite cerrar la sesión del proyecto y redirección a la ruta	Es una función dentro del controlador de los usuarios.

⁷ Método de envío de datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición.

⁸ Algoritmo de encriptación para combinar la contraseña original con una cadena aleatoria de longitud finita. Esto se aplica como medida de seguridad.

	base del sitio.	
<pre> if(\$this-> Auth -> user('id')) \$this-> redirect (array('controller '=> 'cities', 'action'=> 'index')); </pre>	Verificación correcta de la autenticación de usuario.	Función del controlador que verifica la autenticidad.

Tabla 4: Descripción del módulo de autenticación de CakePHP.

Una vez desarrollado este módulo con las directivas que se reflejan en la Tabla 4, se procedió a una exhaustiva prueba de rendimiento y seguridad con el fin de encontrar alguna brecha de seguridad que permitiera el acceso no autorizado de algún usuario que no tuviera las credenciales de acceso. Afortunadamente el módulo de autenticación Auth de CakePHP abarcó todo el dominio y no permite estas irregularidades.

3.5.5 Sprint 2.2

Requerimiento: Implementar funciones que permitan realizar operaciones con la Base de Datos (CRUD).

Número de Sprint: 2.2

Prioridad: Alta.

Para que un manejador de contenidos cumpla su objetivo para el cual fue diseñado tiene que soportar al menos 4 operaciones de manipulación de datos, estas operaciones son bien denominadas en el ámbito Web como CRUD por sus siglas en inglés que significan:

Crear (**Create**), Leer (**Retrieve**), Actualizar (**Update**) y Eliminar (**Delete**).

Específicamente, durante el desarrollo de Yo Culinario estas 4 opciones son válidas y de acceso inmediato para operar sobre los distintos elementos que conforman la arquitectura de datos propuesta para este Trabajo Especial de Grado.

Sin embargo, CakePHP en su ORM provee operaciones sobre los modelos de datos que permiten mediante un lenguaje estándar para operar con la misma. Basado en esto, de acuerdo con la pila del producto, durante el desarrollo de las interfaces visuales propuesta se va incorporando mediante iconos intuitivos aquellas acciones disponibles para que el editor de contenidos pueda operar con sistema manejador de contenidos.

En teoría, para el desarrollo de los **CRUD** ciudades, fotos, imágenes y establecimientos se desarrolló un controlador para cada uno de los modelos creados. Luego, por cada controlador se crearon 4 funciones equivalentes a cada vista del manejador denominadas: **add, edit, view y delete**.

Una vez desarrolladas todas las operaciones en el manejador, el equipo de desarrollo en conjunto con el facilitador y el dueño del producto, procedió a realizar las pruebas de aceptación y funcionamiento corrigiendo aquellos errores que se pudieron presentar en el desarrollo de este requerimiento con la finalidad de mostrarle al cliente un paso más del producto final a desarrollar.

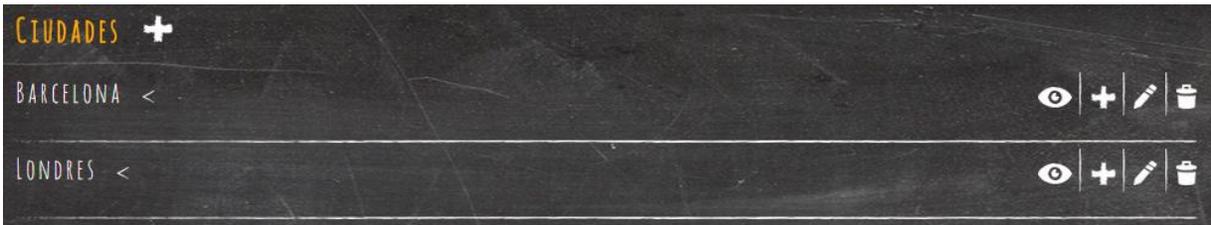


Figura 18: Operaciones CRUD para Ciudades y Establecimientos.

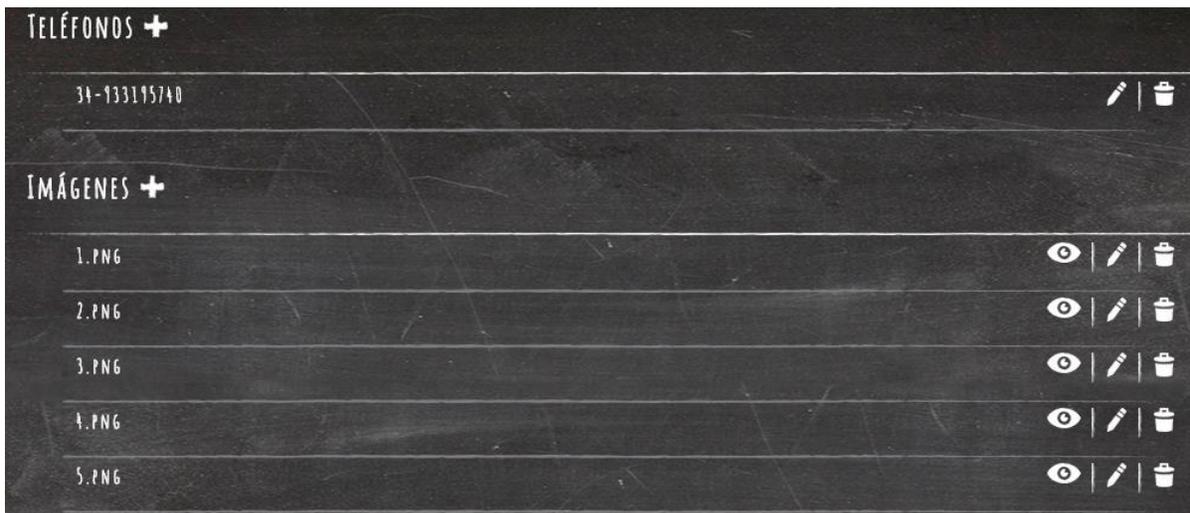


Figura 19: Operaciones CRUD para Teléfonos e Imágenes.

Como se puede observar tanto en la figura 18 como en la figura 19 las operaciones CRUD disponibles para el editor de contenidos se caracterizan por iconos blancos con la metáfora asociada a su funcionamiento.

3.5.6 Sprint 3.0

Requerimiento: Crear una arquitectura de directorios y renombramiento de ficheros del lado del servidor.

Número de Sprint: 3.0

Prioridad: Alta

Yo Culinario entre sus principales necesidades tiene como premisa tener una gran cantidad de imágenes mediante las cuales se pueda reflejar a los usuarios finales una sensación lo más gráfica y detallada posible, de los establecimientos que la misma recomienda.

Con la finalidad de solventar la necesidad antes planteada, y tomando como punto de partida las restricciones impuestas sobre las entidades que componen la aplicación; principalmente las pertenecientes a la ciudades y los establecimientos, las cuales establecen que los mismos deben tener una cantidad finitas de imágenes asociadas y en cuanto a las ciudades que la mismas deben poseer múltiples locales gastronómicos, se dispuso a generar en el sistema manejador de contenidos un sistema de directorios el cual va a permitir a los usuarios administradores olvidarse de la preocupación de ¿cómo? y ¿en dónde? van a estar alojados los archivos que sean cargados y muchos menos como están organizados.

CakePHP cuenta con dos componentes fundamentales que permiten recorrer un sistema de archivos y a partir del mismo, y de ser necesario generar tanto carpetas completas como archivos básicos con sólo especificar los permisos necesarios que se deben tener sobre los directorios. Estos componentes de CakePHP son *Folder* y *File*, y son dos librerías que podemos incluir que permiten realizar un amplio número de operaciones en los sistemas de archivos del lado del servidor.

En Yo Culinario se generó una estructura base, donde cada vez que una ciudad sea ingresada como nueva, se crea una carpeta con el nombre del archivo dentro de una carpeta denominada “webroot” perteneciente al sistema de archivos propio de CakePHP; una vez generada la misma en esta se guarda la imagen por defecto que corresponde a la ciudad y conforme se agreguen nuevos establecimientos gastronómicos a la ciudad se generan nuevas carpetas con el nombre de los mismos, logrando así tener un sistema de archivos que permita recorrer de forma estructurada y ordenada los datos (Ver Figura 20) que luego serán enviados a las vistas de la aplicación móvil.

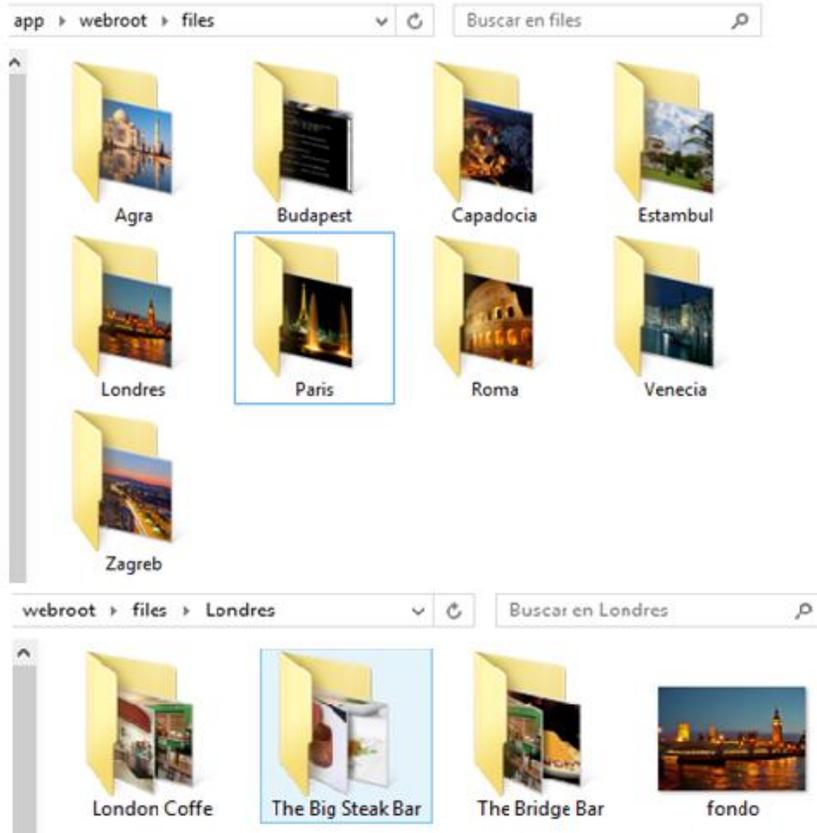


Figura 20: Sistema de archivos Yo Culinario.

Cabe destacar que es necesario proveer de los permisos en las carpetas creadas como en las imágenes cargadas por el usuario administrador, para esto CakePHP provee una instrucción donde se puede especificar claramente que permisos se requieren a la hora de la creación de archivos en el sistema de directorios, en específico y en el caso de generar de un nuevo archivo, la instrucción es la siguiente:

```
$file = new File ('Ruta', <Reescribir>, <Permisos>);
```

Con esta instrucción se genera un nuevo archivo en la ruta que enviemos en el primer parámetro de la función. Con el segundo parámetro le decimos a CakePHP que si el archivo ya existe lo sobrescribe de manera que nos quedemos con la última copia del mismo.

3.5.7 Sprint 4.0

Requerimiento: Implementar los Servicios Web en formato JSON por cada modelo de datos en el manejador de contenidos que estén disponibles para la aplicación.

Número de Sprint: 4.0

Prioridad: Alta.

Un requerimiento fundamental en el desarrollo de este Trabajo Especial de Grado es el desarrollar un mecanismo de comunicación que permita a la aplicación consumir los datos alojados en el servidor de manera rápida, efectiva y confiable.

Una vez investigado en el trabajo de seminario, CakePHP ofrece el soporte estándar RESTful⁹ que no es más que un conjunto de rutinas habilitadas por el *framework* para transformar los datos alojados en los modelos en arreglos asociativos y luego convertir estos arreglos en un formato de texto plano, fácil de trabajar y de contenido liviano para evitar la sobrecarga de la red.

Principalmente en el desarrollo de este Sprint se habilitaron algunas configuraciones y mecanismos para permitir que el servidor responda mediante una petición externa en formato JSON los datos alojados en el modelo solicitado. Estas configuraciones se enumeran a continuación:

Habilitar en el servidor peticiones de todos los orígenes: El proyecto que contiene el manejador de contenidos, contiene en sus directorios un archivo “.htaccess” que es el archivo de configuración distribuida, es un fichero especial, que permite definir diferentes directivas de configuración para cada directorio sin necesidad de editar el archivo de configuración principal de Apache (Wikipedia, 2015). En este archivo se habilitaron las instrucciones:

- Header set Access-Control-Allow-Origin *.
- Header set Access-Control-Allow-Headers "Origin, X-Requested-With, Content-Type, Accept".

En el archivo de configuración del proyecto “routes.php” se agregó la instrucción *Router::parseExtensions()*; la cual se encarga en transformar la data del modelo en JSON, XML y otras extensiones, por defecto el *framework* toma JSON, y *Router::mapResources(array('rest_cities','rest_establishments'))*; para especificar cuáles controladores son los que van a soportar REST.

Para cada controlador creado que soporte REST se agrega la instrucción en el inicio del controlador *public \$components = array ('RequestHandler')*; para habilitar el manejo de las peticiones REST.

⁹ Es un conjunto de principios, o maneras de hacer las cosas, que define la interacción entre distintos componentes, es decir, las reglas que dichos componentes tienen que seguir. El protocolo más usado que cumple esta definición, es el protocolo HTTP.

Hay que desarrollar los controladores que se especificaron en el archivo de configuración `routes.php`, es decir el controlador REST para `cities` y `establishments`. En estos controladores, se obtienen todos los registros mediante una consulta al modelo y luego se serializan esos datos para que puedan ser soportados en formato JSON, es necesaria la serializar para especificar la variable en la vista debe ser serializada a formato JSON y utilizarla como respuesta de la petición (CakePHP.org, s.f.).

El siguiente paso es crear un cliente encargado de capturar las peticiones externas y construir la respuesta que va a ser desplegada en una vista. Es necesario crear un `Socket`¹⁰ para hacer la petición HTTP al servicio REST. CakePHP posee su propia clase `Socket` para este tipo de peticiones. Se agrega al inicio del cliente lo siguiente:

```
App::uses('HttpSocket', 'Network/Http'); dado a que se implementó un mecanismo de seguridad en el manejador de contenidos fue necesario habilitar el cliente para evitar que el manejador solicitar usuario y contraseña con la instrucción $this->Auth->allow(array('request_cities','request_establishments')); ya después se desarrolla cada uno de los servicios mediante funciones en el controlador. Cada función básicamente va a crear un socket dependiendo del modelo que el cliente solicita con los datos serializados y renderizar una vista en común que va a mostrar la información.
```

Por último una vez renderizada la vista se procese a entregar la información al cliente con la instrucción `echo $response_body`.

Una vez culminado los requerimientos del Sprint desarrollado se realizaron pruebas de aceptación y rendimiento probando desde diversas estaciones de trabajo, dispositivos móviles e incluso ya dentro de la aplicación de Sencha Touch para corroborar que la información está completa y en su formato correspondiente. En la siguiente figura se observa un ejemplo del resultado obtenido al realizar una petición al Servicio Web del manejador de contenidos.

¹⁰ Es una clase que permite el envío y recepción de datos bajo el protocolo de transferencia de hipertexto o HTTP.

```

{
  "cities": [
    {
      "City": {
        "id": "12",
        "nombre": "Agra",
        "imagen": "fondo.jpeg",
        "latitud": "23.12350082397461",
        "longitud": "-124.24099731445312",
        "nuevo": "1"
      },
      "Establishment": [
    ]
    },
    {
      "City": {
        "id": "14",
        "nombre": "Capadocia",
        "imagen": "fondo.jpeg",
        "latitud": "-69.24099731445312",
        "longitud": "12.324999809265137",
        "nuevo": "1"
      },
      "Establishment": [
    ]
    }
  ],
}

```

Figura 21: Ejemplo Servicio Web de Ciudades.

3.5.8 Sprint 4.1

Requerimiento: Elaborar un módulo de búsqueda dentro del manejador de contenidos para el rápido acceso de los establecimientos.

Número de Sprint: 4.1

Prioridad: Baja.

El desarrollo de este Sprint está ideado en la escalabilidad que pueda presentar este proyecto a largo plazo, es decir, en algún momento el editor de contenido tendrá como necesidad el buscar de manera rápida y directa algún establecimiento en particular ya sea para verlo, editarlo o eliminarlo. Por esta razón este módulo de búsqueda fue el último en realizarse y de presentarlo en una reunión con el cliente ya que no existe una relación de dependencia entre el manejador y el módulo de búsqueda.

Básicamente, el módulo ubicará mediante una predicción el nombre de los establecimientos que vayan coincidiendo con lo que el administrador coloque en la barra de búsqueda, para el desarrollo del mismo se utilizó una librería complementaria de jQuery denominada jQueryUI previamente definida en el Capítulo I.

El módulo en primera instancia hace una consulta al modelo de los establecimiento para obtener los nombres de los mismo y almacenarlos en un arreglo asociativo, después, con el soporte de JavaScript se crea un arreglo vacío y se le asigna el arreglo asociativo que se extrajo del servidor, una vez el arreglo de JavaScript tenga los nombres con una simple instancia del método autocomplete de la librería jQueryUI ya se encuentra listo para su uso. Por

último una vez que el administrador haya tipado el nombre del establecimiento mediante una acción del formulario de la búsqueda llega al servidor el nombre solicitado y mediante una consulta con el ORM obteniendo el identificador correspondiente a ese nombre se hace un direccionamiento inmediato a la vista de dicho establecimiento.

Dada la baja prioridad del Sprint y su esfuerzo reducido en la implementación, las pruebas con el equipo de desarrollo y el facilitador para su entrega al cliente requirieron poco tiempo puesto que los resultados obtenidos son 2, redirección o no al establecimiento.

3.5.9 Sprint 5.0

Requerimiento: Desarrollar un servicio que se ejecute en segundo plano que verifique la cantidad de tiempo que tiene una ciudad en la base de datos.

Número de Sprint: 5.0

Prioridad: Media.

Una de los requerimientos de la aplicación es la necesidad de informar al usuario si una ciudad ha sido añadida recientemente, con la finalidad de poder llamar la atención del usuario y que el mismo se interese por descubrir las informaciones de establecimientos que se han añadido con esta.

Para esto MySQL provee una funcionalidad denominada “Eventos MySQL”, según (Blanco, 2012) estos no son más que la ejecución planificada de ciertas sentencias que pueden ser llamadas a procedimientos almacenados o simples conjuntos de sentencias SQL. La ventaja de estos eventos es que se pueden configurar para planificarlos, es decir, determinar cuándo se tienen que ejecutar y que intervalo de repetición tienen.

Específicamente en Yo Culinario se definió un evento el cual verifica si la ciudad tiene más de 15 días en la aplicación mediante un campo de la tabla correspondiente que indica la fecha en la que fue creada la ciudad; de ser así el evento se encarga de modificar el valor de la propiedad pertinente que indica si es una ciudad nueva o no lo es.

Con esta herramienta que ofrece MySQL se logra de manera directa resolver el requerimiento antes planteado con una alto índice de efectividad.

Con respecto a las pruebas en este Sprint, se realizaron dos tipos las primeras fueron probadas con un evento que se ejecutará cada 5 minutos para poder verificar si se estaban cumpliendo las reglas impuestas en el mismo, y la segunda fue una prueba en la cual se tuvieron que esperar 15 días para poder verificar la validez del evento con respecto al tiempo asignado. En ambos escenarios los resultados obtenidos fueron satisfactorios tanto para el equipo de desarrollo como para el cliente.

3.5.10 Sprint 6.0

Requerimiento: Asignación de Video Tutoriales y ejemplos de práctica al equipo de desarrollo para lograr un acercamiento con la tecnología a utilizar.

Número de Sprint: 6.0

Prioridad: Baja.

Sencha Touch es una tecnología relativamente nueva creada en 2007 por lo que el porcentaje de desarrollo es aún estrecho, motivado a esto surgió la necesidad de tener un primer acercamiento con la tecnología y esto generó la decisión entre el equipo de desarrollo el facilitador y dueño del producto de incluir como un Sprint el conocimiento y tutoriales de desarrollo con la tecnología.

Se estableció un periodo no mayor a 2 semanas con el fin de observar videos tutoriales para crear una primera aplicación de prueba. En este periodo de tiempo se aprovechó para investigar más a fondo el alcance de la tecnología, pruebas en distintos dispositivos móviles, leer artículos y recomendaciones de cómo trabajar con el *framework* así como también realizar reuniones diarias con el equipo para comentar y debatir sobre beneficios y estimación de tiempos de desarrollo en base a lo previamente investigado.

Videos Tutoriales

Los videos tutoriales que se utilizaron como referencia para empezar fueron los siguientes:

- **Video2brain:**<https://www.video2brain.com/mx/cursos/sencha-touch>
- **Tuts+:**<http://code.tutsplus.com/tutorials/getting-started-with-sencha-touch-2--mobile-12049>

Documentación Sencha Touch

Una de las principales virtudes que ofrece Sencha es su extensa documentación desarrollada por el mismo equipo de trabajo que construyó el *framework*, aunado a su muy activa comunidad, permite la interacción entre desarrolladores que realizan preguntas o módulos de código con los miembros del equipo de Sencha, para tener una comunicación en ambos sentidos de preguntas y respuestas.

Para formar parte de esta comunidad al igual que la mayoría de las comunidades solo es necesario estar registrado a la página de Sencha. Por otro lado, la documentación de Sencha Touch es de criterio público y esta accesible a cualquier persona que desee leer acerca de la misma.

A diferencia de la documentación de CakePHP, donde son artículos escritos completos acerca de algún tópico, en Sencha la documentación es dinámica e interactiva es decir, es una estructura de una página web con un listado de todas las funcionalidades y herramientas que provee la tecnología con la diferencia de que está enmarcada en 4 renglones principales y 2 opcionales apoyado siempre de una barra de búsqueda de acceso inmediato a algún componente.

El siguiente ejemplo denota la versatilidad y facilidad que provee dicha documentación usando como modelo el componente carrusel.



The screenshot shows the Sencha documentation for the `Ext.carousel.Carousel` component. At the top, there is a gear icon, the component name `Ext.carousel.Carousel`, and the xtype `carousel`. Below this, there are statistics for various categories: Configs (56), Properties (0), Methods (187), Events (32), CSS Vars (3), and CSS Mixins (1). A navigation bar contains links for `new Ext.carousel.Carousel`, `add`, `getAnimation`, `getAt`, `getFlex`, and `getFloatingCls`. A 'Select C' button is also present. The main content area displays the following code snippet:

```
Ext.create('Ext.Carousel', {
    fullscreen: true,

    defaults: {
        styleHtmlContent: true,
        centered: true,
    },
    items: [
        {
            html: ' archive        | 2/26/2015 10:50 AM | File folder   |        |
|  deltas         | 2/26/2015 10:50 AM | File folder   |        |
|  resources      | 2/26/2015 10:50 AM | File folder   |        |
|  app.js         | 2/6/2015 9:36 AM   | JS File       | 561 KB |
|  app.json       | 2/6/2015 9:36 AM   | JSON File     | 1 KB   |
|  build.xml      | 1/27/2015 12:23 AM | XML File      | 1 KB   |
|  cache.appcache | 2/6/2015 9:36 AM   | APPCACHE File | 1 KB   |
|  index.html     | 2/6/2015 9:36 AM   | HTML File     | 10 KB  |

Figura 27: Estructura directorios Sencha Touch en producción.

Los archivos que se observan en la figura 27 son los correspondientes al modo de producción de Sencha Touch, Sencha transforma los componentes y módulos a una reducción mínima donde solo son incluidos las clases que son necesitadas únicamente con sus dependencias, esto genera como consecuencia un código fuente mucho más compacto y optimizado, para ser más adelante incluido en el proyecto y luego pasar por un proceso de compilación para obtener los resultados buscados.

También es importante considerar los modos de carga y ejecución de la aplicación en sus diversos ambientes, mientras que en modo de desarrollo las clases y dependencias son cargadas dinámicamente mientras se van necesitando para su ejecución, en el modo de producción estas clases y todo el código fuente que requiera la librería para poder cumplir con las instrucciones escritas por los desarrolladores son procesadas, optimizadas y empaquetadas en el archivo “app.js” obteniendo así como ventaja un código fuente con mejores tiempos de respuesta y una buena experiencia de usuario, ya que toda las mejoras en cuanto a recursos del *framework* son automatizadas.

Por el modo en que trabaja el *framework* denominado SPA (*Single Page Application*) todo el contenido se encuentra alojado en el archivo index.html es decir que en esa única vista en formato HTML se encuentran todas las rutinas, vistas, controladores y modelos asociados a la aplicación desarrollada.

Para poder trabajar sin conexión, una aplicación únicamente necesita de un archivo de manifiesto, el cual indica al navegador que ficheros debe almacenar en la caché local. El contenido del manifiesto puede ser tan simple como un listado de archivos. Una vez que el navegador ha descargado y almacenado los archivos (HTML, CSS, imágenes, JavaScript, etc.), el navegador hace uso de estos ficheros, incluso cuando el usuario actualiza la página en su

navegador. El archivo de manifiesto es lo que le indica al navegador cuando y qué tiene que almacenar en su caché, y qué tiene que traerse de la Web (Itzgarro, 2012), el archivo de manifiesto en Sencha se denomina “cache.appcache” (ver figura 27); un archivo de manifiesto puede incluir tres categorías: CACHE, NETWORK y FALLBACK.

- **CACHE:** esta es la sección predeterminada para las entradas. Los archivos incluidos en esta sección (o inmediatamente después de CACHE MANIFEST) se almacenarán en caché explícitamente después de descargarse por primera vez.
- **NETWORK:** los archivos incluidos en esta sección son recursos permitidos que requieren conexión al servidor. En todas las solicitudes enviadas a estos recursos se omite la caché, incluso si el usuario está trabajando sin conexión.
- **FALLBACK:** se trata de una sección opcional en la que se especifican páginas alternativas en caso de no poder acceder a un recurso. La primera URI<sup>13</sup> corresponde al recurso y la segunda, a la página alternativa. Ambas URI deben estar relacionadas y tener el mismo origen que el archivo de manifiesto. Se pueden utilizar carácter comodín.

Así, el manifiesto va a guardar en caché el archivo index.html que equivale a todo el proyecto en sí y la ruta a los Servicios Web que están alojados en el manejador de contenidos, con estas 3 directivas es suficiente para que se pueda utilizar la aplicación sin conexión consumiendo los ficheros almacenados en caché. En cuanto al almacenamiento de las imágenes, estas se van a guardar de manera automática en la caché siempre y cuando quede espacio en la misma, el tamaño de la caché varía entre 5 y 10 Mb dependiendo de los navegadores. En la Figura 28 se describe un modelo gráfico de secuencia que ilustra la petición entre el cliente y el servidor utilizando un sistema de caché.

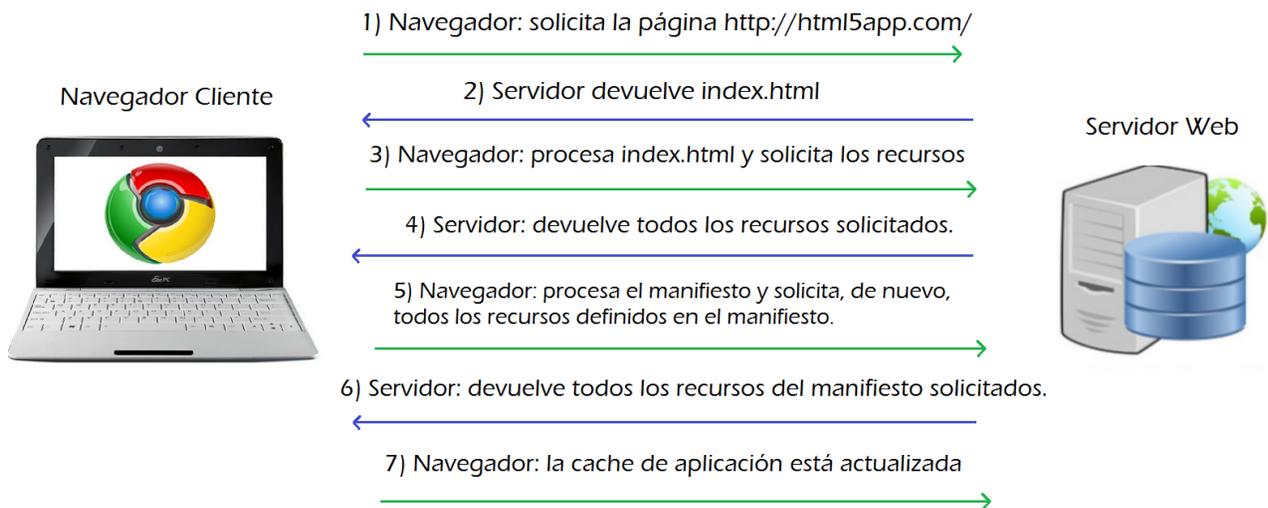


Figura 28: Diagrama de secuencia sistema de caché.  
Fuente: (HTML5app.com, S.F.).

<sup>13</sup> Identificador de recursos uniforme de manera unívoca en una red.

Ahora, el navegador está listo y la caché contiene los ficheros indicados en el manifiesto. Si el manifiesto no ha cambiado, y la página se recarga, ocurre lo siguiente:

1. **Navegador:** vuelve a solicitar la página “http://html5app.com/”.
2. **Navegador:** detecta que tiene una copia local de index.html y la sirve de manera local.
3. **Navegador:** procesa la página index.html y los recursos existentes en la caché se sirven de manera local.
4. **Navegador:** solicita de nuevo el manifiesto al servidor.
5. **Servidor:** devuelve un código 304 indicando que no ha cambiado nada en el manifiesto.

Basado en lo descrito en este Sprint, y con los conocimientos teóricos obtenidos se asegura que la aplicación Yo Culinario pueda operar sin conexión a Internet siempre y cuando la primera vez haya guardado todos los recursos descritos en el manifiesto y estos hayan sido almacenados en la caché del explorador. Para su validación, el equipo SCRUM realizó pruebas sin conexión con la aplicación probando diversas combinaciones de estados de la aplicación, en todas estas fases el resultado fue positivo.

### 3.5.17 Sprint 10.0

**Requerimiento: Creación del proyecto e integración de *plugins*.**

**Número de Sprint:** 10.0

**Prioridad:** Media.

Para poder cumplir con los requerimientos previamente establecidos en el Sprint 0 se investigaron en específicos tres herramientas fundamentales, una de ellas es Apache Cordova, la cual se utilizaría para generar el proyecto que será desplegado en los sistemas operativos móviles quien funcionará como puente entre los algoritmos escritos en Sencha Touch y las interfaces nativas de los sistemas elegidos.

En específico y ampliamente definido en el anexo 4.2.1 Apache Cordova es un conjunto de librerías que permiten a los desarrolladores de aplicaciones móviles acceder a las funciones nativas de los dispositivos inteligentes, tales como la cámara o el *GPS* desde el lenguaje de programación JavaScript. Combinado con un *framework* de desarrollo tal como jQuery Mobile, Kendo UI o Sencha Touch provee la capacidad de generar un código fuente multiplataforma utilizando solamente los lenguajes bases HTML5, CSS3 y JavaScript.

Al utilizar las librerías de Cordova las aplicaciones pueden ser desarrolladas sin la necesidad de utilizar los lenguajes nativos de los diferentes sistemas operativos móviles existentes.

Una ventaja importante de utilizar Apache Cordova es la portabilidad que provee, debido a que las aplicaciones desarrolladas requieren una mínima cantidad de cambios para poder funcionar en diferentes dispositivos y sistemas operativos, tomando en cuenta que el código a generar es básicamente HTML5 con CSS3 y JavaScript.

## Crear Aplicaciones usando Apache Cordova

Para crear aplicaciones, Cordova utiliza un cliente llamado Cordova CLI mediante el cual se agregan los comandos necesarios para proveer las configuraciones particulares de cada sistema operativo en el que se desee un correcto funcionamiento del código fuente.

Una vez instalado el cliente Cordova CLI y realizado el código fuente, ya sea utilizando un *framework*, o escribiendo el algoritmo en los lenguajes base HTML5 CCS3 y JavaScript; se procede a generar el proyecto introduciendo las primitivas que reconocerá el sistema para la creación del mismo. Cordova permite elegir la plataforma para la cual se desee generar el proyecto, ya que para cada plataforma debe crear y empaquetar diferentes archivos de configuración con los cuales la aplicación tendrá un correcto funcionamiento en el sistema operativo seleccionado, esto sin la necesidad de realizar modificaciones en el código base de la aplicación.

El comando mediante el cual Cordova permite generar una aplicación desde cero es el siguiente: `Cordova create <Directorio> <Identificador> <Título>`

En donde el primer parámetro es la ruta o directorio donde se va a generar la carpeta que contendrá al proyecto completo.

El segundo parámetro es el identificador único del proyecto, el cual más adelante será tomado como referencia al momento de hacer la carga del proyecto a las diferentes tiendas de los sistemas operativos para los cuales se va a generar el mismo. Cabe destacar que este parámetro es opcional y el mismo puede ser editado en los archivos de configuración del proyecto.

Por último el tercer parámetro es el título o nombre que va a tomar la aplicación, este último también es opcional y puede ser modificado luego en los archivos de configuración.

Al ejecutar el comando anterior Cordova CLI se encarga de crear una carpeta en el directorio indicado la cual contiene un código base para el proyecto aún sin tener las especificaciones necesarias para las plataformas que se deseen agregar.

Es importante tomar en cuenta que para la aplicación Yo Culinario se utilizaron las plataformas iOS y Android las cuales fueron definidas en el alcance del Trabajo Especial de Grado.

Cabe destacar que el proceso anteriormente descrito es igual de válido para generar el proyecto base para ambas plataformas, pero para poder crear las configuraciones necesarias tanto para Android como para iOS se tienen ciertos criterios que varían, además se necesitan diferentes herramientas y entornos de desarrollo que permitan y soporten las creación de estos proyectos.

### Creación de una aplicación para la plataforma Android

Ya creado el proyecto base, Cordova CLI tiene la opción de agregar la plataforma o sistema operativo para el cual se van a crear las configuraciones necesarias, además de la estructura y sistema de archivos que el proyecto nativo requiere.

En el caso particular de Android es necesario el siguiente comando:

```
$ cordova platform add android
```

Una vez ejecutado el comando se genera en el directorio especificado el proyecto con el sistema de archivos y las configuraciones propias de un proyecto Android.

Entre los directorios creados en el sistema de archivos del proyecto se encuentra la carpeta “assets/www” la cual contendrá en un principio el código fuente de la aplicación junto con un archivo denominado “Cordova.js” el cual viene incluido en dicha carpeta en el momento de su creación. Este archivo contiene todas las librerías para hacer las llamadas a las funcionalidades nativas del sistema en el código fuente y en sí nos proporciona la capacidad de realizar peticiones a los recursos del dispositivo.

Una vez copiado los archivos pertinentes al proyecto, Cordova CLI permite construir una versión de prueba la cual se puede ejecutar de dos diferentes maneras, la primera es en un dispositivo virtual con Android y la segunda mediante la conexión USB de algún dispositivo Android. Esto es posible siempre y cuando se tenga instalado el SDK<sup>14</sup> de Android en el computador donde se estén realizando las pruebas.

Para Yo Culinario se utilizó la segunda opción al momento de hacer pruebas en cuanto a la plataforma Android se refiere, esto debido a la facilidad y mayor grado de accesibilidad que otras herramientas proveen con respecto a Cordova CLI.

La opción escogida fue el IDE <sup>15</sup>Eclipse, ya que dicho editor tiene la capacidad de gestionar de manera más directa y con una interfaz bastante adaptada al entorno de los desarrolladores, los proyectos del sistema operativo Android. Además, provee la facilidad de importar y exportar los proyectos, y la compilación se puede realizar mediante una interfaz gráfica teniendo una ventana donde se muestran tanto los errores como los eventos que se están desarrollando en tiempo real dentro de los dispositivos de prueba.

Juntando todas estas herramientas y realizando la serie de pasos antes detallada, se puede construir desde cero una aplicación que corra sin ningún tipo de problemas con cualquier dispositivo Android; cabe destacar que la visual del ejecutable generado va de la mano con el código fuente, no obstante algunos elementos pueden variar en cuanto a cómo los vemos dependiendo de cada plataforma, ya que Cordova trabaja directamente con el motor visual del sistema operativo en el cual se esté ejecutando.

---

<sup>14</sup> (*Software Development Kit* - Kit de desarrollo de software o devkit). Un SDK es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo o similar.

<sup>15</sup> *Integrated development environment* que está referido a un entorno de desarrollo.

## **Creación de una aplicación para la plataforma IOS**

De manera similar a la plataforma Android, Cordova CLI proporciona los comandos necesarios para crear un proyecto con las configuraciones requeridas para funcionar correctamente bajo el sistema operativo móvil IOS; teniendo que realizarse un proceso muy similar al de la plataforma anterior pero utilizando las herramientas propias de la plataforma Apple.

Para comenzar el proceso se necesita tener un computador con un sistema operativo MAC OS, ya que las aplicaciones para dispositivos Apple solo pueden ser compiladas y probadas utilizando este entorno, debido a que todos los productos de este proveedor están fuertemente acoplados; lo que genera como desventaja la necesidad de poseer un equipo con estas características.

Una vez instalado Cordova CLI en el equipo se necesita el siguiente comando para crear un proyecto con un sistema de archivos propio de un proyecto para dispositivos IOS:

```
$ cordova platform add ios
```

Con el comando anterior se crea el proyecto al cual se le añadirá el código base para más tarde realizar su compilación y ejecución.

Como artefacto para realizar la compilación y las pruebas en los dispositivos, el sistema MAC OS ofrece una herramienta preinstalada llamada XCODE (Anexo 4.2.5); la cual no es más que un IDE con la capacidad de gestionar dispositivos periféricos.

Una vez exportado el código a XCODE, se pueden realizar las pruebas de dos maneras; la primera es mediante la compilación en dispositivos virtuales, teniendo la capacidad de poder seleccionar cuál de los diferentes modelos existentes en el mercado simular, y la otra manera sería conectando un dispositivo físico y haciendo la compilación de manera que sea posible visualizar los resultados mediante una conexión USB en el mismo.

## **Personalización de una aplicación creada en Apache Cordova**

Al finalizar el proceso de exportación de los proyectos creados a los diferentes IDE, se requiere que la aplicación de alguna manera tenga características propias y distintivas para que los usuarios puedan recordarlas y recomendarlas.

Para dar solución al punto anterior tanto iOS como Android proveen dos personalizaciones de la aplicación de gran relevancia, ellos son el ícono que tendrá la aplicación cuando queramos acceder a ella desde el dispositivo una vez esté instalada, y una pantalla de carga la cual se muestre únicamente al abrir la misma, dando tiempo así que puedan ser cargados todos los módulos y componentes necesarios para tener una correcta ejecución.

En cuanto a los íconos los proyectos Android usando el IDE Eclipse tienen una gran ventaja, debido a que para ellos solo debe proporcionarse una imagen y este se encarga de hacer la redimensión de la misma para verse bien y adaptado a todos los diferentes tamaños de dispositivos actuales. En cuanto IOS se refiere no es tan sencillo, ya que es necesario generar

la misma imagen en diversos tamaños según la versión del sistema operativo, cada uno representando un dispositivo diferente de los ofrecidos por el proveedor.

Teniendo como referencia los íconos, el *Splash* o pantalla de carga de la aplicación también debe ser una imagen la cual se debe generarse en diferentes tamaños con la finalidad de adaptarse de manera idónea en las diferentes pantallas. En este tópico no existe diferencia alguna entre ambas plataformas ya que cada una requiere de una cantidad diferente de dimensiones de imagen para su correcta implementación.

### **Integración de *Plugins***

Algunas aplicaciones requieren de herramientas en específico para poder realizar o cumplir con los requerimientos planteados en la fase de análisis, algunos de estos requerimientos necesitan utilizar dispositivos de *hardware* propios del dispositivo, para esto Apache Cordova implementó un sistema de *plugins*, los cual no son más que herramientas de terceros que podemos descargar y añadir a proyectos, para que mediante instrucciones JavaScript, permitan a los desarrolladores acceder a estos dispositivos nativos.

En Yo Culinario era necesario utilizar varios de estos plugins, debido a que el mismo requiere varias interacciones con el dispositivo, por lo cual se presenta una tabla (ver Tabla 5) con los *plugins* instalados y su funcionalidad.

| Nombre Plugin       | Funcionalidad                                                                                                                                  | Sistema Operativo              |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| Network-information | Con este complemento podemos obtener la información de la disponibilidad de la conexión a Internet del dispositivo.                            | iOS y Android                  |
| Geolocation         | Se utiliza para obtener los permisos necesarios para poder acceder al GPS del dispositivo y con esto obtener la posición geográfica del mismo. | iOS y Android                  |
| Dialogs             | El mismo se utiliza para poder mostrar ventanas emergentes en la aplicación.                                                                   | iOS y Android                  |
| Splahscreen         | Permite mostrar una imagen previa mientras cargan los componentes de la aplicación.                                                            | iOS y Android<br>iOS y Android |
| InAppBrowser        | Con esta herramienta se puede abrir un navegador interno dentro de la aplicación.                                                              | iOS y Android                  |
| Social Sharing      | Este plugin se utiliza para poder compartir la información deseada en las redes sociales instaladas en el dispositivo.                         | iOS y Android                  |
| Status Bar          | Funciona solamente en iOS, y se utiliza para obtener permisos sobre la barra de estado de los dispositivos iOS.                                | iOS                            |

Tabla 5: Plugins utilizados en Yo Culinario.

Es importante destacar que para la instalación de estas herramientas se accede mediante Cordova CLI a repositorios en la red, en los cuales se encuentran las especificaciones de cada una de estas y ejemplos de su uso. El siguiente comando ejemplifica la incorporación de un *plugin* al proyecto.

```
$ cordova plugin add org.apache.Cordova.inappbrowser
```

### 3.5.18 Sprint 10.1

**Requerimiento: Elaborar un módulo que permita compartir información en las redes sociales instaladas en el dispositivo.**

**Número de Sprint:** 10.1

**Prioridad:** Alta.

En la gran mayoría de las aplicaciones móviles en la actualidad podemos encontrar un botón que nos permita compartir contenidos de interés en las redes sociales, esto tiene un significado bastante lógico, desde el punto de vista del usuario interesa dar a conocer a amigos y familiares tópicos de interés que encontremos en las diferentes fuentes que se consulten. De parte de las marcas, siempre es importante dar a conocer su producto por todos los canales existentes, ya que mediante el mismo se produce un auge masivo de la marca teniendo un resultado lógico de ampliar el mercado y llegar a todos los rincones a través de la red.

Teniendo en cuenta lo expresado anteriormente Yo Culinario no es la excepción, para la marca surge la necesidad de llegar a los usuarios por la mayor cantidad de medios posibles, y para los usuarios la de compartir el contenido que sea de mayor agrado en la aplicación a través de sus redes sociales.

Específicamente, la aplicación permite compartir información de establecimientos con una imagen respectiva del local en cuestión en todas las redes sociales que se tengan instaladas en el dispositivo en donde se esté ejecutando.

Para ahondar un poco más en el tema y según la opinión de (importancia.org, s.f.) Una red social no es más que una enorme sala de Intercambio de Información donde los usuarios comparten con otros todas las actividades que realizan, así como Contenidos Multimedia o bien simplemente textos, estando en contacto constante y con actualizaciones en tiempo real, realizando intercambios de información y dando lugar a debates y comentarios sobre un contenido en particular.

En el trabajo de investigación realizado previamente, se optó por compartir los contenidos únicamente en las redes Twitter y Facebook, al ser las dos anteriores las de mayor auge en la actualidad; no obstante, se logró conseguir que la aplicación se comunicará y estuviera en la capacidad de compartir contenidos con todas las redes instaladas en el dispositivo, esto en gran parte a una herramienta de tercero o *plugin* el cual fue desarrollado para ser integrado con cualquier aplicación desarrollada bajo Apache Cordova. El mismo permite detectar las aplicaciones instaladas en el dispositivo y hacer una conexión para compartir información; además de esto da la posibilidad no solo de elegir cualquier red social instalada, sino también

decidir qué información se desea compartir teniendo como añadido poder agregar contenidos multimedia a lo que se quiere publicar en las redes sociales de cada uno de los usuarios de la aplicación.

Para solventar la necesidad se creó un botón en la vista perteneciente al establecimiento con una metáfora bastante conocida en el mundo de las aplicaciones, el cual al ser presionado muestra la lista de las aplicaciones instaladas en las cuales se pueden compartir la información antes visualizada.

Como punto de interés el *plugin* utilizado se denomina “SocialSharing” cabe destacar que el mismo funciona de la misma manera en ambas plataformas móviles iOS y Android pero esta es una de las virtudes que ofrece desarrollar aplicaciones con Apache Cordova.

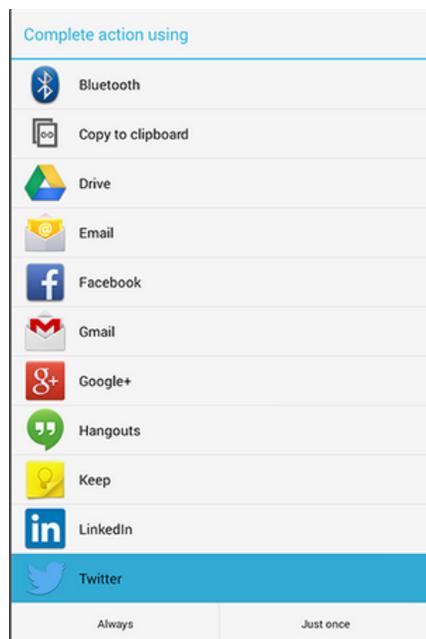


Figura 29: Ventana emergente botón compartir.

La figura 29 es un ejemplo de una instancia a los servicios nativos del dispositivo para listar las aplicaciones disponibles para compartir en Yo Culinario.

### 3.5.19 Sprint 10.2

**Requerimiento: Realizar una integración del código fuente de Yo Culinario con el proyecto de Cordova.**

**Número de Sprint:** 10.2

**Prioridad:** Baja.

Una vez terminada la creación e instalación de todos los *plugins* necesarios para el proyecto, se procede a realizar la integración del código fuente escrito bajo el *framework* Sencha Touch;

para esto es necesario ubicarnos en la carpeta generada al crear el proyecto “www/assets”. Enviaremos todos los archivos pertenecientes al proyecto a este directorio, con lo cual los editores de texto utilizados tomarán los archivos bases de aquí y podrán compilar el proyecto tomando como base los archivos incluidos en esta carpeta.

Esta carpeta debe contener a los sumo dos archivos primordiales, el primero es “Cordova.js” el cual es la compilación de todas las librerías de Apache Cordova y el mismo es el que permite la ejecución del código escrito en HTML en los dispositivos. El segundo es denominado normalmente “index.html” y es la vista principal de la cual va a partir la aplicación, esta debe contener una implementación de lo que se quiere que se visualice una vez sea cargada y desplegada la aplicación en el dispositivo.

Una vez terminado este Sprint se realizaron varias pruebas en diferentes dispositivos, con la finalidad de encontrar errores tanto de lógica como de visual de la aplicación; obteniendo como resultado mejorar el rendimiento en algunas partes de la misma y optimizando cada componente visual.

### 3.6 INTERACCIÓN CON EL CLIENTE, SPRINTS QUINCENALES Y MEJORAS EN LA APLICACIÓN

Un factor de vital importancia durante la aplicación de la metodología SCRUM son las repetidas interacciones con el cliente donde cabe destacar que el mismo es Nidal Barake fundador de Yo Culinario y que a su vez es nuestro Jefe en la organización y director corporativo, siendo así el escenario ideal para llevar a cabo la metodología gracias a la facilidad de realizar iteraciones que forman parte del ciclo de vida de SCRUM.

El fin de llevar a cabo dichas iteraciones con el cliente es ir aprobando de manera modular aquellos Sprints o conjunto de estos que integrados, son un subconjunto del producto total a desarrollar con la idea de mostrar avances parciales al cliente y este pueda hacer una evaluación objetiva sobre el producto en construcción, que a su vez, puede ser evaluado, analizado y cambiado de ser necesario con anticipación para reducir el impacto debido al cambio, permitiendo que el nuevo elemento se pueda integrar con total perfección al producto en construcción.

Una tabla de iteraciones entre el equipo de desarrollo, facilitador y cliente expresa de manera sencilla y comprensible el proceso de las reuniones del cliente con el grupo de trabajo. Un ejemplo sería el siguiente en la Tabla 6.

| Nro. Reunión | Artefactos y modelos entregados | Producto                | Sprints involucrados |
|--------------|---------------------------------|-------------------------|----------------------|
| 1            | CRUD ciudades                   | Manejador de Contenidos | 2                    |
| 2            | CRUD establecimientos           | Manejador de Contenidos | 2                    |
| 3            | Autenticación de usuarios       | Manejador de Contenidos | 2                    |

|    |                                            |                         |       |
|----|--------------------------------------------|-------------------------|-------|
| 4  | Módulo de búsqueda                         | Manejador de Contenidos | 4     |
| 5  | Servicios Web                              | Manejador de Contenidos | 4     |
| 6  | Menú de ciudades, carrusel de ciudades     | Aplicación              | 7     |
| 7  | Carrusel de establecimientos               | Aplicación              | 8     |
| 8  | Vista establecimiento                      | Aplicación              | 7 y 8 |
| 9  | Menú favoritos                             | Aplicación              | 9     |
| 10 | Integración mapas de Google                | Aplicación              | 9     |
| 11 | Integración código fuente en iOS y Android | Apache Cordova          | 10    |

Tabla 6: Cambios realizados por el cliente al presentar iteraciones del producto a realizar.

Algunos de los cambios realizados tanto en el manejador de contenidos como en la aplicación se muestran en las siguientes imágenes:

### 3.6.1 Manejador de Contenidos

**Propuesta #1:** Vista Principal del manejador donde se observan todas las ciudades organizadas con sus respectivas operaciones CRUD.

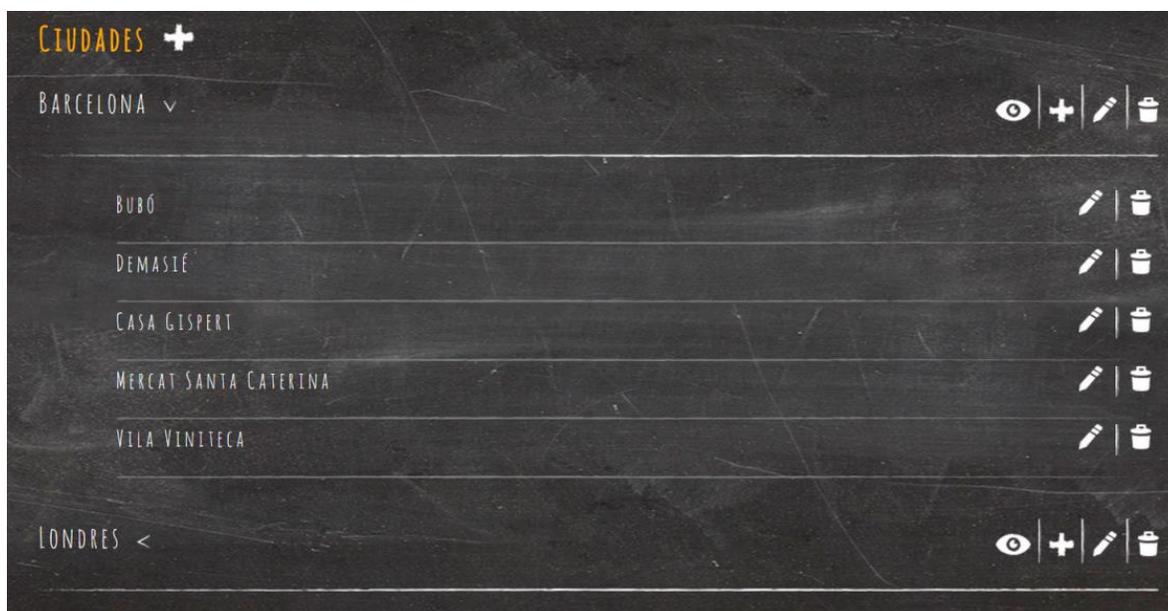


Figura 30: Vista de ciudades sin categorización.

**Cambio realizado:** Agregar un factor que categorice los establecimientos por su tipo.



Figura 31: Vista de ciudades categorizadas.

**Propuesta #2:** Desarrollar una vista que permita ver en detalle la información del establecimiento justo a sus números telefónicos y sus fotos correspondientes.



Figura 32: Vista del establecimiento sin carrusel.

**Cambio realizado:** Incorporar un carrusel de imágenes pertenecientes a establecimientos que permita al editor de contenidos visualizar de manera rápida y agradable a la vista para una mejor gestión de las imágenes.



Figura 33: Vista de establecimientos con carrusel.

### 3.6.2 Aplicación Móvil

**Propuesta #1:** Desarrollar una vista inicial de la aplicación donde se visualicen todas las ciudades correspondientes a la aplicación con una disposición horizontal.



Figura 34: Vista de ciudades sin carrusel.

**Cambio realizado:** Después de una reunión con el cliente, se propuso realizar un carrusel en pantalla completa donde se pudiera apreciar cada ciudad con más detalle, brindando un valor agregado a la aplicación, ya que pueden apreciarse mejor las imágenes y esto es un factor de promoción de las localidades.

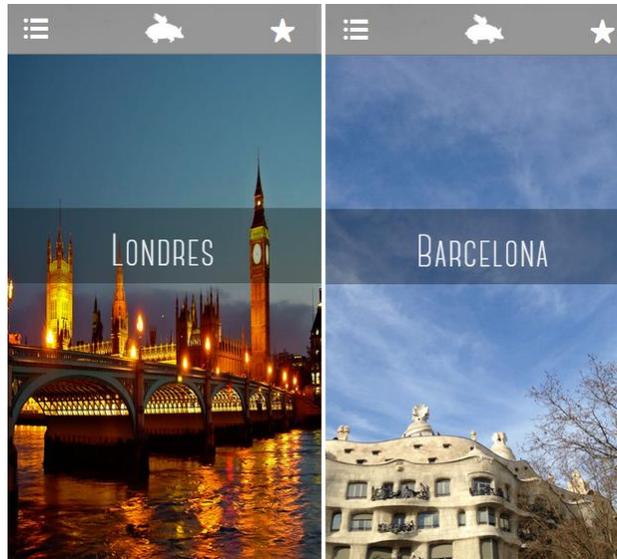


Figura 35: Vista de ciudades con carrusel.

**Propuesta #2:** En la vista del establecimiento se desarrolló una barra inferior con múltiples atajos que permiten al usuario realizar interacciones con la aplicación, por ejemplo: compartir en con las redes sociales, ubicar en el mapa al establecimiento, agregar a favoritos o simplemente visitar la página del establecimiento.



Figura 36: Vista por establecimientos con 4 íconos.

**Cambio realizado:** El cliente optó por hacer algunas modificaciones e incorporar las secciones de la barra inferior como parte de la información del establecimiento dejando solamente la sección de compartir y favoritos en la barra.



Figura 37: Vista por establecimientos con 2 íconos.

### 3.7 DISPOSITIVOS UTILIZADOS PARA PROBAR YO CULINARIO

Yo Culinario es una aplicación híbrida que funciona bajo los sistemas operativos iOS y Android, por lo tanto para su constante desarrollo se requirieron diversos dispositivos que soporten ambos sistemas. Tener acceso a estos dispositivos permitió realizar pruebas continuas para detectar errores en la fase de desarrollo y poder observar cómo estaba siendo construida la aplicación.

Se utilizaron los siguientes dispositivos de la plataforma iOS (ver Tablas 7 y 8):

| Dispositivos iOS | Tamaño de pantalla en píxeles | Densidad de pixel |
|------------------|-------------------------------|-------------------|
| IPhone 4s        | 320x480                       | 2 dpi             |
| IPhone 5s        | 360x568                       | 2 dpi             |

Tabla 7: Dispositivos de prueba IOS.

Dispositivos de prueba Android:

| Dispositivos Android     | Tamaño de pantalla en píxeles | Densidad de pixel <sup>16</sup> |
|--------------------------|-------------------------------|---------------------------------|
| Samsung Galaxy SIII      | 360x640                       | 2 dpi                           |
| Samsung Galaxy S4        | 360x640                       | 3 dpi                           |
| Samsung Galaxy S5        | 360x640                       | 3 dpi                           |
| Samsung Galaxy SIII mini | 320x533                       | 1.5 dpi                         |
| HTC Evo 3D               | 360x640                       | 1.5 dpi                         |

Tabla 8: Dispositivos de prueba Android.

Adicional a estos dispositivos de prueba se utilizó el inspector de desarrollo del navegador Google Chrome (Ver Figura 38) el cual es una herramienta que permite depurar JavaScript, inspeccionar el DOM o analizar el tiempo de ejecución de cada función para optimizar el rendimiento. Adicional a esto el mismo permite simular tamaños de pantalla de dispositivos, lo que genera como ventaja poder simular el comportamiento visual de la aplicación en cualquier tamaño de pantalla, con el fin de ajustar detalles en dispositivos específicos.



Figura 38: Inspector de elementos de Google Chrome.

---

<sup>16</sup> Densidad de pixel (dpi) es una medida de resolución de vídeo relacionada al tamaño del monitor en pulgadas y del número total de píxeles en la dirección horizontal y en la dirección vertical.

### 3.8 PRUEBAS DE ACEPTACIÓN DE YO CULINARIO

Para definir la correcta funcionalidad de la aplicación se realizó una encuesta a un grupo de personas aleatorias con la finalidad de que las mismas interactuaran con las interfaces y las funcionalidades pertenecientes a Yo Culinario dando a conocer sus opiniones y recomendaciones.

Se utilizó para esta tarea formularios de Google, enviándolos a los encuestados un enlace con las preguntas y teniendo resultados anónimos de los cuales se obtuvieron una serie de estadísticas bastante interesantes y a su vez de gran utilidad.

Las preguntas de la encuesta están basadas en las funcionalidades que posee Yo Culinario y destacando que es una aplicación híbrida, se quiso obtener una comparación de la experiencia de usuario con aplicaciones nativas que conozcan previamente.

Las preguntas realizadas en el cuestionario fueron las siguientes:

1. ¿Te pareció intuitiva la navegación dentro de la aplicación?
2. ¿El contenido multimedia de la aplicación fue de tu agrado?
3. Basado en la situación de la red en el territorio nacional, ¿cómo te pareció la velocidad de carga y ejecución de la aplicación?
4. ¿Qué fue lo que más impacto causó dentro de la aplicación?
5. La aplicación cuenta con las siguientes funcionalidades: guardar en Favoritos, compartir con todas las redes sociales de tu teléfono, mapas de Google y llamar a los establecimientos de manera directa. ¿Encontraste todos los módulos mientras utilizaste la aplicación?
6. ¿Cuál de las categorías de los establecimientos te llamó más la atención?
7. ¿Qué tan importante es para ti que la aplicación pueda seguir funcionando sin conexión a Internet?
8. ¿Recomendarías la aplicación basado en su contenido de comida gastronómica?
9. ¿Durante la interacción con Yo Culinario percibiste que estabas en una página Web?

En las preguntas 1, 2, 3 y 7 se utilizó una escala entre 1 y 5 para medir el nivel de satisfacción del usuario partiendo que el número 1 es totalmente en desacuerdo y el 5 es totalmente de acuerdo.

Para las preguntas 5, 8 y 9 se realizó una pregunta de selección simple donde el encuestado simplemente debía elegir entre sí y no; en cuanto a las preguntas 4 y 6 las respuestas eran específicas de la aplicación y se debía interactuar con la misma para realizar una respuesta concreta.

### 3.9 RESULTADOS OBTENIDOS

Pregunta 1:



Figura 39: Resultados de la pregunta 1.

Pregunta 2:



Figura 40: Resultados de la pregunta 2.

**Pregunta 3:**

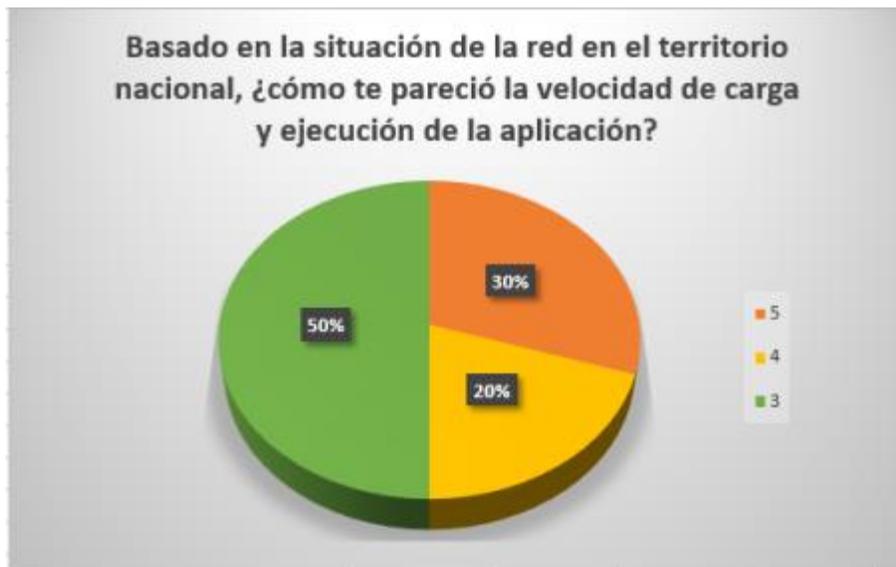


Figura 41: Resultados de la pregunta 3.

**Pregunta 4:**

La pregunta está basada en la interacción directa de los usuarios con la aplicación, por ende no se pueden cuantificar y mostrar en un gráfico los resultados.

La conclusión directa con esta pregunta, es que la mayor parte de los usuarios se sienten atraídos por el contenido multimedia que tiene la aplicación, en particular con la gran cantidad de imágenes de establecimientos con las que cuenta.

**Pregunta 5:**

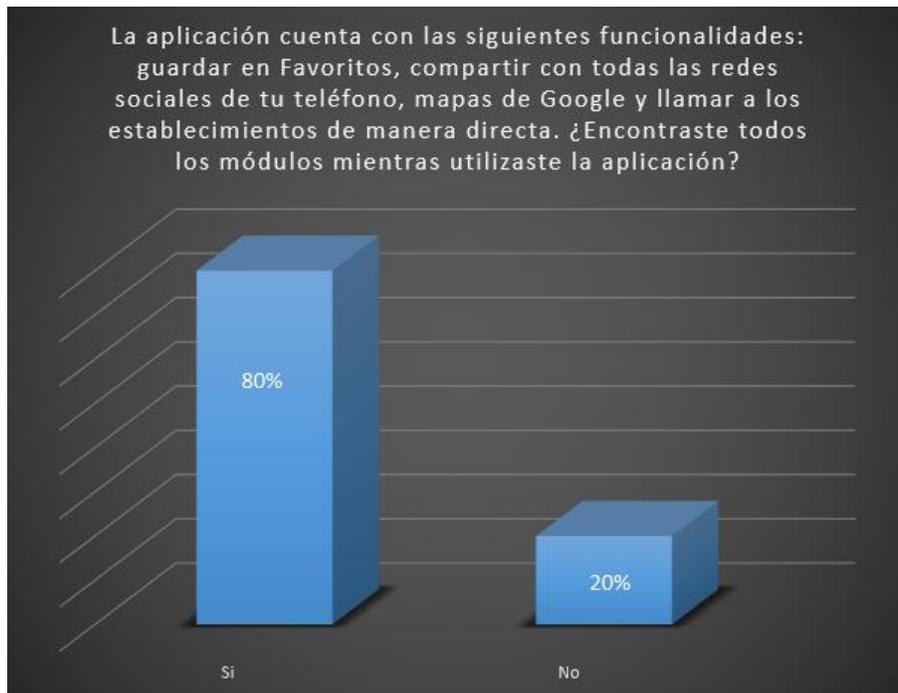


Figura 42: Resultados de la pregunta 5.

**Pregunta 6:**



Figura 43: Resultados de la pregunta 6.

**Pregunta 7:**



Figura 44: Resultados de la pregunta 7.

**Pregunta 8:**



Figura 45: Resultados de la pregunta 8.

**Pregunta 9:**

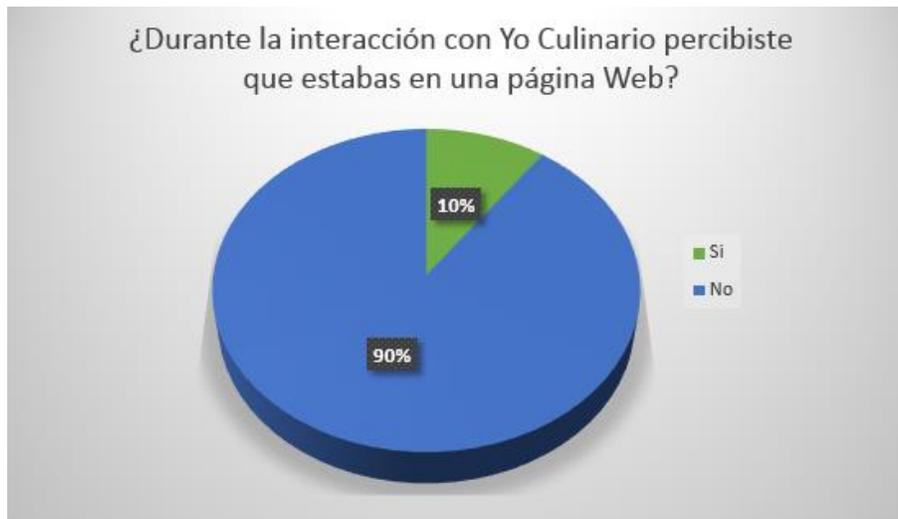


Figura 46: Resultados de la pregunta 9.

Esta pregunta en particular tenía una relevancia destacada, debido a que Yo Culinario es una aplicación híbrida lo que quiere decir básicamente que es una Web desplegada en un dispositivo inteligente. Obtener un resultado de un 90% de los usuarios expresando que no se notó en lo más mínimo que estaba dentro de una página Web, es un agregado adicional para los resultados de la investigación, logrando con esto realizar un trabajo lo más óptimo posible y siendo de mucho agrado para el grupo de desarrollo.

En cuanto al resto de las preguntas realizadas se obtuvieron resultados positivos, sin mayores detalles para resaltar.

## Resultados y Conclusiones

---

Finalizado este Trabajo Especial de Grado, tenemos un manejador de contenidos que permite gestionar toda la información referente al dominio del problema (arte gastronómico) por una parte, y por otra una aplicación móvil híbrida desarrollada en un *framework* de última generación que además de soportar la integración con múltiples plataformas móviles también ofrece diversos módulos tecnológicos que integrados permite a Yo Culinario competir contra las últimas aplicaciones del mercado sin nada que envidiar e incluso, servir modelo para construir nuevas aplicaciones.

La decisión de seleccionar una metodología ágil particularmente SCRUM, tomando en cuenta que sus artefactos, modelos y ciclo de vida se adaptaba de manera ideal a nuestro equipo de trabajo, fue factor decisivo para el desarrollo tanto del manejador de contenidos como de la aplicación móvil, desarrollando además de los productos, otros elementos que forman parte también de la ingeniería de software como una pila de producto que puede funcionar como manual de documentación para verificación y reutilización de requerimientos modelos y actividades.

Los objetivos específicos de este Trabajo Especial de Grado se cumplieron satisfactoriamente, estos en conjunto permitieron que se lograra el objetivo general del mismo, teniendo así 2 piezas de software funcionales que permiten trabajar de manera paralela e independiente sobre dominios diferentes pero que tienen una relación semántica entre sí, permitiendo asociarlos y en conjunto lograr solventar el planteamiento del problema que surgía por la necesidad de poseer información al instante para aquellos amantes del mundo gastronómico sobre aquellos sitios de interés ubicado en todos los rincones del planeta.

Según lo afirmado en los párrafos anteriores y en base a los resultados favorables obtenidos en las encuestas, podemos decir que se cumplieron los objetivos planteados al comienzo de este TEG, y atribuimos en gran parte el logro de estos a la planificación, y aplicación de inicio a fin de la metodología ágil SCRUM, obteniendo excelentes resultados por su implementación.

En el ámbito personal, llevar a cabo de inicio a fin el seguimiento, coordinación y desarrollo de un proyecto tan complejo y extenso permitió adquirir nuevas destrezas y habilidades tanto a nivel de programación con el *framework* Sencha Touch como habilidades en la gestión y desarrollo de proyectos a nivel corporativo y con un equipo de trabajo integrado.

## Recomendaciones y Trabajo Futuros

---

Yo Culinario es una aplicación que dejó muchas enseñanzas para los desarrolladores, como fue mencionado anteriormente se utilizó el *framework* Sencha Touch para su desarrollo el cual a pesar de tener un gran número de ventajas también tiende a ser bastante confuso, por lo que es recomendable hacer una buena inducción en el mismo antes de empezar a desarrollar en él, o realizar cursos con los cuales se puede entender de manera efectiva, para poder explotar al máximo los beneficios que provee.

Otra recomendación muy importante es comprender que en Sencha todo lo creado dinámicamente debe ser destruido, ya que en sus últimas versiones aún no incorpora un recolector de basura; por lo cual es recomendable para el correcto y óptimo funcionamiento de las aplicaciones tener suma precaución con este detalle, es lógico saber que la memoria creada en tiempo de ejecución y no eliminada puede causar que las aplicaciones se ralenticen.

Como punto importante y sobre todo para las aplicaciones híbridas, es comprender que estas aún están un paso por debajo de las aplicaciones nativas en cuanto a funcionamiento con los componentes internos de los dispositivos, por lo cual es recomendable evaluar que tanto se desea la conexión de la aplicación con los medios externos como la cámara o el GPS ya que si la aplicación tiene una dependencia directa de los mismos, utilizar o desarrollar una aplicación híbrida puede no ser la mejor opción.

Para trabajos futuros y como recomendación directa se pueden realizar algoritmos en el servidor que permitan modificar las imágenes a formato PNG-8, para evitar tener que utilizar herramientas dedicadas para cumplir con este requerimiento.

Otro módulo de importancia que se puede agregar, es permitir al usuario trazar rutas dentro del mismo mapa de la aplicación, para lograr con esto obtener una mejor experiencia de usuario.

Por ser una aplicación con alcance mundial, y tomando como premisa el inglés como idioma universal y además ser un factor que favorece la accesibilidad dentro de la aplicación, es el incorporar el idioma inglés a la misma favoreciendo a aquellos lectores y potenciales usuarios de la aplicación a su descarga de las tiendas virtuales..

También se podría considerar añadir una herramienta de búsqueda de establecimientos específicos en la aplicación, debido a que los mismos se irán incrementando cada vez más, puede resultar útil añadir este tipo de funcionalidad.

Por último y esto tomando en consideración los beneficios que se obtienen al realizar aplicaciones híbridas con Apache Cordova y Sencha Touch, es expandir la aplicación para otros sistemas operativos menos populares como BlackBerry o Windows Phone teniendo como resultado poder llegar a un número aún mayor de usuarios a nivel mundial.

## Referencias Bibliográficas

---

- Alexander, A. (21 de Diciembre de 2010). *AlvinAlexander.com*. Obtenido de <http://alvinalexander.com/php/cakephp-login-auth-user-username-id-authcomponent>
- Alvarado, P. (14 de Julio de 2014). *Ipadizate*. Obtenido de <http://www.ipadizate.es/2014/07/20/xcode-93212/>
- Barake, N. (Marzo de 2010). *Yo Culinario*. Obtenido de <http://www.yoculinario.com/index.php/yo-culinario/>
- Blanco, A. (4 de Octubre de 2012). *Adictos al Trabajo*. Obtenido de <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=eventosMysql>
- CakePHP.org*. (s.f.). Obtenido de <http://api.cakephp.org/2.3/class-JsonView.html>
- Caldas, R. (16 de Septiembre de 2013). *rolandocaldas.com*. Obtenido de <https://rolandocaldas.com/html5/localstorage-en-html5>
- Claris, P. (2008). *SOFTENG*. Obtenido de <http://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>
- CokeBook 1.3*. (2012). Obtenido de <http://book.cakephp.org/1.3/es/The-Manual/Core-Components/Authentication.html>
- CookBook CakePHP*. (2012). Obtenido de <http://book.cakephp.org/1.3/es/The-Manual/Developing-with-CakePHP/Models.html>
- CoookBoke 2.0*. (2015 de Abril de 2015). Obtenido de <http://book.cakephp.org/2.0/en/core-libraries/components/authentication.html>
- Cruz, A. (2014). *Aplicación para el Reconocimiento de Obras de Arte: La Ciudad Universitaria de Caracas en tu Dispositivo Móvil*. Caracas.
- Desarrollo Web*. (Mayo de 2009). Obtenido de <http://www.desarrolloweb.com/manuales/manual-jquery.html>
- Docs Oracle*. (2011). Obtenido de [http://docs.oracle.com/cd/E24842\\_01/html/E23286/secfile-60.html](http://docs.oracle.com/cd/E24842_01/html/E23286/secfile-60.html)
- Docs.Sencha*. (21 de Octubre de 2014). Obtenido de <http://docs-origin.sencha.com/touch/2.4/2.4.1-apidocs/>
- EcuRed*. (s.f.). Obtenido de [http://www.ecured.cu/index.php/Eclipse,\\_entorno\\_de\\_desarrollo\\_integrado](http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado)
- Garzas, J. (Abril de 2012). *JavierGarzas.com*. Obtenido de <http://www.javiergarzas.com/2012/04/herramientas-scrum.html>
- Garzas, J. (Marzo de 2013). *Javier Garzas*. Obtenido de <http://www.javiergarzas.com/metodologias-agiles>

goicochea, a. (15 de Marzo de 2013). *Tecnologías de la informática y la estrategia*. Obtenido de <http://anibalgoicochea.com/2013/03/15/phonegap-una-solucion-para-el-desarrollo-de-aplicaciones-moviles-multiplataforma/>

*importancia.org*. (s.f.). Obtenido de <http://www.importancia.org/redes-sociales.php>

Itzgarro, A. (2012). *arkaitzgarro.com*. Obtenido de <http://www.arkaitzgarro.com/html5/capitulo-9.html>

*Junta de Andalucía*. (S.F.). Obtenido de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>

Ken Schwaber, J. S. (Julio de 2013). *Scrumguides*. Obtenido de <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100>

*Manual del Photoshop*. (2010). Obtenido de <http://www.manualdephotoshop.com/manualdephotoshop/formatos/>

Meléndez, C. (11 de Octubre de 2012). *Maestros del Web*. Obtenido de <http://www.maestrosdelweb.com/aplicaciones-moviles-sencha-touch/>

Merper, F. (2 de Diciembre de 2014). *Hipertextual*. Obtenido de <http://hipertextual.com/archivo/2014/12/ios-y-android-rentabilidad-contra-cuota-mercado-grafico/>

*Netizen*. (20 de Noviembre de 2014). Obtenido de <http://blog.netizen-online.es/como-crear-tu-primera-app/>

Ortí, C. (1 de Junio de 2013). *wordpress*. Obtenido de <https://al095668.wordpress.com>

Ospina, M. (3 de Septiembre de 2009). *ciens.ucv Base de Datos*. Obtenido de <http://www.ciens.ucv.ve:8080/genasig/sites/bases-de-datos/archivos/GuiaTeoricaTema1BaseDatos.pdf>

Rivera, A. (17 de Enero de 2012). *PCWorld*. Obtenido de <http://www.pcworld.com.mx/Articulos/20734.htm>

*safariBooksOnline*. (s.f.). Obtenido de <https://www.safaribooksonline.com/library/view/hands-on-sencha-touch/9781449366513/ch09.html>

*SCRUM.org*. (2013). Obtenido de <https://www.scrum.org/Resources/scrum-glossary>

*Sencha docs*. (21 de Octubre de 2014). Obtenido de <http://docs-origin.sencha.com/touch/2.4/2.4.1-apidocs/#!/api/Ext.Component>

*Sencha docs*. (25 de Marzo de 2015). Obtenido de [http://docs.sencha.com/touch/2.4/core\\_concepts/components.html](http://docs.sencha.com/touch/2.4/core_concepts/components.html)

*Sun Earth Tools*. (s.f.). Obtenido de <http://www.sunearthtools.com/es/tools/distance.php>

Tedeschi, N. (s.f.). *Microsoft Developer Network*. Obtenido de <https://msdn.microsoft.com/es-es/library/bb972248.aspx>

Toapanta, K., Vergara, M., & Campaña, M. (s.f.). *espe.edu*. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/5899/1/AC-SIS-ESPE-034427.pdf>

*Wikipedia*. (6 de Noviembre de 2013). Obtenido de [http://es.wikipedia.org/wiki/Dispositivo\\_inteligente](http://es.wikipedia.org/wiki/Dispositivo_inteligente)

*Wikipedia*. (11 de Abril de 2015). Obtenido de <https://es.wikipedia.org/wiki/.htaccess>

Yusuf, M. (Diciembre de 2013). *miftyisbored.com*. Obtenido de <http://miftyisbored.com/complete-restful-service-client-cakephp-tutorial/>

### 4. TECNOLOGÍAS DE REFERENCIA

#### 4.1 ANEXO A: YO CULINARIO

Yo Culinario es la plataforma mediante la cual Nidal Barake, gastrónomo y escritor, comparte sus experiencias de viajes de gastronomía y reseñas de restaurantes con su audiencia a través de diversos canales como redes sociales, blog, publicaciones digitales e impresas, y participación en medios audiovisuales. Las personas que consumen esta información y siguen Yo Culinario, tienen gusto por la buena mesa y persiguen tendencias globales de gastronomía y buen vivir. En pocos años, Yo Culinario se ha convertido en referencia en América Latina para conseguir una buena experiencia culinaria y saber dónde comer en cada ciudad. Adicionalmente, Nidal Barake es co-fundador de gasTROPop, primer restaurante itinerante de Venezuela y referencia regional de innovación y creatividad (Barake, 2010).



Figura 47: Yo Culinario.  
Fuente: (yoculinario.com, 2010).

## 4.2 ANEXO B: TECNOLOGÍAS DE DESARROLLO

### 4.2.1 Apache Cordova

Apache Cordova es un *framework* gratuito y de código abierto que permite crear aplicaciones móviles multiplataforma a partir de tecnologías web como HTML5, CSS3 y JavaScript. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema), pero no se tratan tampoco de aplicaciones web (teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo).

Apache Cordova maneja librerías que permiten tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas librerías se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas en JavaScript.

Además permite el desarrollo ya sea ejecutando las aplicaciones en nuestro navegador web, sin tener que utilizar un simulador dedicado a esta tarea, y brinda la posibilidad de soportar funciones sobre *frameworks* como Sencha Touch o jQuery Mobile.

Apache Cordova permite crear actualmente aplicaciones móviles para: iPhone, Android, Windows Phone, BlackBerry, BlackBerry 10, webOS, Symbian y Bada (goicochea, 2013).

Los recursos del dispositivo que podemos acceder con Apache Cordova son los siguientes:

- Acelerómetro.
- Cámara.
- Contactos.
- Sistema de Ficheros.
- Geolocalización.
- Media.
- Red.
- Notificaciones (alertas, sonido, vibración).
- Almacenamiento.



Figura 48: Logo Apache Cordova.  
Fuente: (Cordova.apache.org, 2012).

## 4.2.2 Sencha Touch

Sencha Touch es un *framework* bajo licencia libre que utiliza los estándares web de HTML5, CSS3 y JavaScript para desarrollar aplicaciones web para dispositivos móviles con la apariencia de ser nativos de los sistemas IOS, Android y BlackBerry.

Sencha Touch produce un efecto que compite entre las aplicaciones nativas ofreciendo una experiencia de alto rendimiento que los usuarios esperan cuando se utilizan aplicaciones móviles.

Tiene como principal ventaja ser el marco de trabajo con mayor rapidez de respuesta actualmente, su grado de fluidez (gestualmente hablando) es óptimo y permite trabajar con una serie de complementos ya implementados y reutilizables que son de gran utilidad a la hora de realizar aplicaciones móviles (Meléndez, 2012).

Sencha trabaja bajo el modelo de programación MVC y tiene un grado bastante alto de documentación tanto en la web como en la bibliografía existente; otro punto fuerte es el soporte multiplataforma lo cual ya viene siendo un común denominador para los desarrolladores de estas herramientas.

Sencha Touch está específicamente desarrollado para aprovechar la mayor potencia y flexibilidad de HTML5, CSS3 y JavaScript, haciendo uso de los nuevos elementos disponibles en la nueva especificación del estándar para utilizar video, audio, guardar datos offline, así como una librería con múltiples *widgets* de usuario y control para los eventos táctiles de los dispositivos móviles con efectos muy agradables gracias a la capa de CSS3 que incorpora.



Figura 49: Sencha Touch.  
Fuente:(sencha.com, S.F.).

Sencha Touch está desarrollado en componentes es decir, sus librerías y código fuente (API) son un conjunto de subclases que heredan de la clase Ext.Component los cuales ofrecen las siguientes ventajas:

- Auto carga propia en una página usando una plantilla de su interfaz.
- Operadores de configuración, métodos y manejo de eventos.
- Manejo de funciones que capturan eventos predeterminados (*listeners*).
- Los Componentes son creados e instanciados de la misma manera que un objeto es instanciado a partir de una clase definida usando la sintaxis Ext.create.
- La instancia a un componente es directa y es optimizada para dispositivos de diversas características.
- Ambiente de referencia de componentes entre vistas.

Una de las principales ventajas de Sencha es la facilidad que le brinda al desarrollador el uso de herramientas poderosas y a su vez de fácil manipulación para crear los elementos que van a formar parte de la aplicación; 'xtype' es la instancia usada en Sencha Touch para hacer referencia a los mismos dentro de la aplicación. Alguno de los componentes más usados de Sencha se muestra a continuación (ver Tabla 9):

| Clase        | Objeto         | Clase        | Objeto         | Clase        | Objeto         |
|--------------|----------------|--------------|----------------|--------------|----------------|
| Ext.Toolbar  | xtype:toolbar  | Ext.Carousel | xtype:carousel | Ext.Map      | xtype:map      |
| Ext.Button   | xtype:button   | Ext.List     | xtype:list     | Ext.Img      | xtype:image    |
| Ext.TitleBar | xtype:titlebar | Ext.Menu     | xtype:menu     | Ext.TabPanel | xtype:tabpanel |

Tabla 9: Componentes de navegación.

### 4.2.3 CAKE PHP

CakePHP es un *framework* libre, de código abierto, para el desarrollo rápido de aplicaciones para PHP estructurado bajo el patrón de arquitectura MVC. Es una estructura fundamental para ayudar a los programadores a crear aplicaciones web. Su objetivo principal es permitir trabajar de forma estructurada, rápida y sin pérdida de flexibilidad.

Con CakePHP el desarrollo web ya no es monótono porque ofrece las herramientas para escribir el código necesario: la lógica específica de la aplicación.

Se caracteriza por su sencilla rapidez y fácil uso además de ser intuitivo y fácil de aprender ya que se hereda la sintaxis y semántica del lenguaje de programación PHP aprovechando por otro lado el ORM que provee el *framework*.

Aquí una lista de las características que provee esta herramienta de desarrollo:

- Licencia flexible.
- Compatibilidad con las últimas versiones de PHP (5.2.6 o superior).
- Soporte de aplicación (Scaffolding).
- Contiene CRUD para la interacción con las base de datos.
- Generación automática de código.
- Arquitectura MVC.
- Funciones de validación para los Modelos.
- Componentes de Email, Cookie, Seguridad, Sesión y otros.
- Integración de plantillas y *Helpers* de desarrollo.
- Soporte de AJAX y JSON.
- Manejo de rutas relativas y absolutas.



Figura 50: Logo CakePHP.  
Fuente: (cakephp.org, S.F.).

#### 4.2.4 Eclipse

Eclipse es un programa informático de licencia libre, compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Dispone de un editor de texto con resaltado de sintaxis donde puedes ver el contenido del fichero en el que estás trabajando, además contiene una lista de tareas y otros módulos similares; la compilación se realiza tiempo real y tiene integración con Apache Ant.

Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plugins. Asimismo, a través de estos plugins libremente disponibles es posible añadir un sistema de control de versiones o elementos que permitan trabajar diferentes tipos de proyectos, como lo son por ejemplos proyectos para la plataforma Android.

#### 4.2.5 Xcode

Introducido en octubre de 2003 junto a la versión 10.3 de Mac OS X, Xcode es el entorno de desarrollo integrado (IDE) creado por Apple que trabaja de manera conjunta con *Interface Builder* (herramienta gráfica para crear interfaces de usuario) y que se puede descargar de manera gratuita desde la Mac App Store. Este conjunto de herramientas además incluye una colección de compiladores del proyecto GNU (GCC) y puede compilar código C, C++,

Objective-C, Objective-C++, AppleScript y Java gracias a una gran gama de modelos de programación.

La aplicación incluye el IDE de Xcode, el compilador, instrumentos, simulador de iOS, el SDK de los últimos sistemas operativos iOS y OS X y muchas características más al alcance de los desarrolladores. Entre otras cosas, cuenta con una amplia variedad de herramientas innovadoras para crear aplicaciones, el editor profesional se mantiene enfocado en el código, se ha simplificado la interfaz para hacer que sea mucho más rápido y fácil de usar, así como varias herramientas para analizar el rendimiento visual.

Xcode ofrece a los desarrolladores todo lo necesario para crear aplicaciones para Mac, iPad e iPhone. La última versión cuenta con un diseño de interfaz de usuario que unifica la codificación, pruebas y depuración dentro de una única ventana (Alvarado, 2014).

Entre las características más destacables de Xcode hay que destacar la tecnología que permite distribuir el proceso de construcción a partir de código fuente entre varios ordenadores.



Figura 51: Logo Xcode.  
Fuente ([itunes.apple.com](https://itunes.apple.com), 2015).

## 4.2.6 Mapas de Google

Los mapas de Google consisten en un servidor de aplicaciones de mapas que pueden ser incrustadas en una página web mediante JavaScript o en una aplicación para Android. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes (Cruz, 2014).



Figura 52: Logo mapas de Google.  
Fuente: (scriptcase.net, S.F.).

Los mapas de Google permiten hacer clic en cualquier parte para que se muestre más información según la ubicación en la que nos encontremos, en general los siguientes elementos están presentes en ellos:

- **Cuadro de búsqueda:** Busca lugares, negocios, intersecciones, direcciones etc.
- **Resultados de búsqueda:** Mediante marcadores para obtener información.
- **Compartir enlaces:** Puedes compartir la URL de tu mapa en cualquier momento.
- **Regla para medir distancias:** Te ayudará a entender la escala del mapa y las distancias entre ubicaciones en distintas magnitudes.
- **Recorridos virtuales:** Permite realizar recorridos virtuales por las calles, ciudades, plazas, etc.
- **Contenido multimedia:** Permite incluir elementos multimedia (enlaces a páginas web, imágenes, videos, etc.) dentro de las explicaciones emergentes de marcas de posiciones y zonas.
- **Acercamientos y alejamientos:** Para mostrar el mapa.

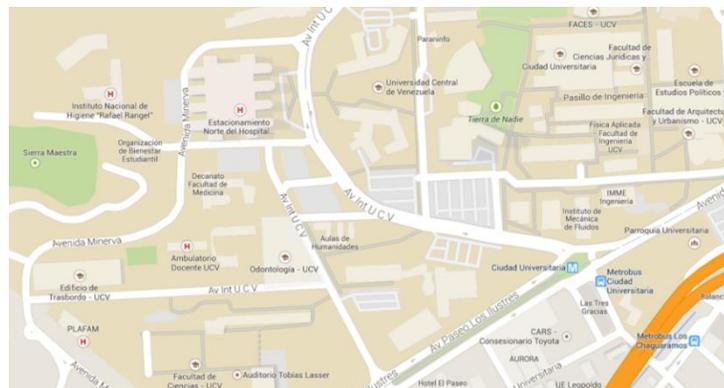


Figura 53: Ejemplo de un mapa de Google.