**Índice de Contenido**

[1 Introducción 6](#_Toc414384011)

[1.1 Justificación 7](#_Toc414384012)

[1.2 Objetivos 8](#_Toc414384013)

[1.2.1 Objetivo General 8](#_Toc414384014)

[1.2.2 Objetivos Específicos 8](#_Toc414384015)

[1.3 Caracterización del área en que participó 9](#_Toc414384016)

[1.3.1 TERACOM.mx 9](#_Toc414384017)

[1.3.1.1 Misión 9](#_Toc414384018)

[1.3.1.2 Visión 9](#_Toc414384019)

[1.4 Problemas a resolver 10](#_Toc414384020)

[1.5 Alcances y limitaciones 10](#_Toc414384021)

[1.5.1 Alcances 10](#_Toc414384022)

[1.5.2 Limitaciones 11](#_Toc414384023)

[2. Fundamento teórico 11](#_Toc414384024)

[2.1 Proceso de uso de los cupones de descuento 11](#_Toc414384025)

[2.2 UML 12](#_Toc414384026)

[2.3 Modelo de software incremental 13](#_Toc414384027)

[2.4 Lenguaje de programación Java 14](#_Toc414384028)

[2.5 HTML 15](#_Toc414384029)

[2.6 XML 15](#_Toc414384030)

[2.7 API REST 16](#_Toc414384031)

[2.8 HTTP 16](#_Toc414384032)

[2.9 CSS 17](#_Toc414384033)

[2.10 Bootstrap 17](#_Toc414384034)

[2.11 NodeJS 17](#_Toc414384035)

[2.12 MongoDB 18](#_Toc414384036)

[2.13 Android 18](#_Toc414384037)

[2.14 NPM 19](#_Toc414384038)

[2.15 JavaScript 19](#_Toc414384039)

[2.16 AJAX 20](#_Toc414384040)

[2.17 AngularJS 20](#_Toc414384041)

[2.18 Android Studio 20](#_Toc414384042)

[2.19 Eclipse 21](#_Toc414384043)

[2.20 jQuery 21](#_Toc414384044)

[2.21 JSON 21](#_Toc414384045)

[2.22 IONIC Framework 22](#_Toc414384046)

[2.23 CORDOVA Framework 22](#_Toc414384047)

[2.24 Socket.IO 23](#_Toc414384048)

[2.25 GitHub 23](#_Toc414384049)

[2.26 Enterprise Architect 23](#_Toc414384050)

[2.27 ProjectLibre 24](#_Toc414384051)

[2.28 Diagramas de Casos de Uso 24](#_Toc414384052)

[2.29 Diagramas de Secuencias 24](#_Toc414384053)

[2.30 Diagramas de Actividades 24](#_Toc414384054)

[2.31 Diagramas Entidad Relación 25](#_Toc414384055)

[3. Procedimiento y descripción de las actividades realizadas 25](#_Toc414384056)

[3.1 Obtención de requerimientos 25](#_Toc414384057)

[3.2 Análisis de requerimientos 25](#_Toc414384058)

[3.2.1 Interfaz gráfica 26](#_Toc414384059)

[3.2.2 Servicios 26](#_Toc414384060)

[3.2.3 Eventos 27](#_Toc414384061)

[3.2.4 Base de datos 27](#_Toc414384062)

[3.2.6 Gestor de paquetes NPM 28](#_Toc414384063)

[3.3 Modelado de la Aplicación 29](#_Toc414384064)

[3.4 Diseño de la aplicación 30](#_Toc414384065)

[3.5 Desarrollo de la aplicación 30](#_Toc414384066)

[3.6 Pruebas e implementación 33](#_Toc414384067)

[3.6.1 Pruebas 33](#_Toc414384068)

[3.6.2 Implementación 35](#_Toc414384069)

[4. Resultados, planos, gráficas, prototipos y programas 36](#_Toc414384070)

[4.1 Obtención de requerimientos 36](#_Toc414384071)

[4.1.1 Documento de requerimientos 36](#_Toc414384072)

[4.1.2 Características de los usuarios 36](#_Toc414384073)

[4.1.3 Requisitos específicos 37](#_Toc414384074)

[4.1.4 Requisitos funcionales 37](#_Toc414384075)

[4.1.4.1 Registro al sistema de la cuponera como usuario “promotor” 37](#_Toc414384076)

[4.1.4.2 Registro al sistema de la cuponera como usuario “visor” 38](#_Toc414384077)

[4.1.4.3 Registro de cupones 38](#_Toc414384078)

[4.1.4.4 Restricción de las medidas del cupón 38](#_Toc414384079)

[4.1.4.5 Vista de inicio 38](#_Toc414384080)

[4.1.4.6 Vista de cupones 39](#_Toc414384081)

[4.1.4.7 Botón Favorito 39](#_Toc414384082)

[4.1.4.8 Botón Comentar 39](#_Toc414384083)

[4.1.4.9 Botón Buscar 40](#_Toc414384084)

[4.1.4.10 Vista Favoritos 40](#_Toc414384085)

[4.1.4.11 Vista Buscar 40](#_Toc414384086)

[4.1.4.12 Edición de datos de usuario 40](#_Toc414384087)

[4.1.4.13 Restricción comentarios de cupón 41](#_Toc414384088)

[4.1.4.14 Restricción lista de favoritos 41](#_Toc414384089)

[4.1.4.15 Restricción vista de cupones 41](#_Toc414384090)

[4.1.4.16 Eliminación de cupones 41](#_Toc414384091)

[4.1.4.17 Formato de imágenes para cupones permitidos 42](#_Toc414384092)

[4.1.4.18 Inicio de Sesión 42](#_Toc414384093)

[4.1.4.19 Administración de usuarios 42](#_Toc414384094)

[4.1.4.20 Funcionalidad uniforme 43](#_Toc414384095)

[4.2 Modelado del sistema 43](#_Toc414384096)

[4.2.1 Diagramas de casos de uso 43](#_Toc414384097)

[4.2.2 Diagramas de secuencias 46](#_Toc414384098)

[4.2.3 Diagrama Entidad Relación 54](#_Toc414384099)

[4.3 Desarrollo del sistema 56](#_Toc414384100)

[4.3.1 Convenciones de nombres 56](#_Toc414384101)

[4.3.2 Vistas de la Aplicación Web. 57](#_Toc414384102)

[4.3.3 Vistas de la Aplicación Móvil. 57](#_Toc414384103)

[Recomendaciones y Conclusiones 58](#_Toc414384104)

[Referencias bibliográficas y virtuales 60](#_Toc414384105)

**Índice de figuras**

**No se encuentran elementos de tabla de ilustraciones.**

# 1 Introducción

Son cada vez más las personas que utilizan dispositivos móviles en la vida diaria con el propósito de hacer más eficiente su manera de vivir, haciendo más sencillas tareas como ir al banco para realizar alguna transacción, ir a una tienda de libros e incluso conseguir boletos de avión. Actualmente existen aplicaciones para realizar todo este tipo de tareas, en los primeros dispositivos móviles estaban enfocadas en mejorar la productividad personal como eran alarmas, calculadoras, calendarios, etc.

El aumento de la accesibilidad a internet ha sido causa importante por la cual el uso de aplicaciones móviles ha aumentado considerablemente. Es por eso que desde un teléfono celular podemos ver los lugares más cercanos a partir de nuestra ubicación y esto se logra a través de internet.

Cuando hablamos de lugares en los cuales podemos consumir algún producto o servicio por lo general pensamos también en el consumo y su precio. Los cupones de descuento son una estrategia de ventas para la mayoría de los establecimientos comerciales que los puede llevar a más público, obtener nuevos clientes y aumentar su cartera de clientes futuros. Como vemos la promoción y divulgación de un negocio por medio de cupones de descuento es una tarea rentable y que trae beneficios para los negocios y para los consumidores.

Debido a lo anterior, desarrollar una aplicación móvil que pueda tener los cupones de descuento de los mejores lugares a los cuales ir a comer, o te indique el mejor lugar para comprar ropa, o para adquirir productos de uso diario o para regalar, se vuelve en una aplicación realmente indispensable en un dispositivo móvil.

## Justificación

Debido a la mayor accesibilidad a internet, y a que en la actualidad el uso de un dispositivo móvil se ha incrementado en la población, se tiene la oportunidad de desarrollar una aplicación móvil que contenga cupones de descuento de las tiendas que ofrecen productos o servicios que estén más cercanas al usuario a partir de la ubicación del dispositivo móvil.

Al desarrollar una aplicación móvil que contenga cupones de descuento de asentamientos comerciales el usuario podrá ver los comercios que estén más cerca de su ubicación actual para obtener un beneficio en cuanto al costo de los productos o servicios que necesita. Así mismo los comercios pueden apreciar un beneficio al publicar sus cupones de descuento para después los usuarios puedan asistir a sus negocios para obtener los productos y servicios que ofrece dicho negocio.

La aplicación de cupones de descuento hasta el momento es la única aplicación que cumple con estos objetivos dentro del país puesto que no hay alguna otra aplicación que promocione cupones de descuento de los las empresas, locales comerciales, tiendas.

## Objetivos

### Objetivo General

* Desarrollar una aplicación móvil la cual muestra cupones de descuento de las tiendas más cercanas a partir de la ubicación geográfica del usuario.

### Objetivos Específicos

* Realizar el análisis de los requerimientos para el desarrollo de la aplicación tanto móvil como web.
* Desarrollar la interfaz de la aplicación móvil la cual mostrara los cupones existentes.
* Desarrollar la interfaz que permita a las tiendas subir sus cupones.
* Desarrollar la interfaz que permita a las tiendas registrarse, para poder subir los cupones.
* Realizar pruebas de la aplicación web en conjunto con la aplicación móvil.

## 1.3 Caracterización del área en que participó

### 1.3.1 TERACOM.mx

TERACOM.mx desarrolla e implementa servicios web para PyMES que hacen de la experiencia en el trabajo, la mejor. Con soluciones como e-commerce, e-shop, web pages, web apps, y web portals, consigue que hagas lo que quieres hacer con y para tus clientes y colaboradores de forma rápida, sencilla y a un costo competitivo.

#### 1.3.1.1 Misión

Proveer aplicaciones móviles y servicios web sofisticados a la medida de PyMEs y ONGs de México y Latinoamérica para optimizar sus funciones y contribuir al desarrollo de la competitividad a través de consultoría personalizada.

#### 1.3.1.2 Visión

Ser una empresa reconocida y recomendada por profesionales, empresas, organizaciones e instituciones, por la calidad de los servicios web y soluciones informáticas que implementamos a través de metodologías y técnicas que dan valor al negocio de nuestros clientes.

## Problemas a resolver

Las empresas, locales comerciales, tiendas departamentales o cualquier establecimiento comercial ya sea pequeño, mediano o grande, tienen como actividad prioritaria buscar herramientas para obtener consumidores de los productos o servicios que ofrece y así llegar a más consumidores, pero ¿qué pasa cuando los consumidores adquieren preferencia por algún establecimiento o marca comercial y no saben si dicha marca se encuentra cerca de su ubicación? o cuando un establecimiento nuevo necesita hacerse conocer a los consumidores o ¿qué pasa cuando un consumidor requiere de algún producto o servicio específico y no sabe conde buscar? La aplicación promociona los cupones de descuento de los establecimientos comerciales para que el usuario puedan adquirir los productos o servicios según la actividad comercial de estos establecimientos, tomando en cuenta la ubicación geográfica del usuario para que así pueda visualizar la ubicación de establecimiento.

## 1.5 Alcances y limitaciones

### 1.5.1 Alcances

Por una parte se desarrollará la aplicación móvil la cual permitirá a los usuarios observar los distintos cupones de descuento de las tiendas más cercanas por medio de la ubicación geográfica del usuario.

Por otra parte se desarrollará la aplicación web en donde las tiendas participantes deberán registrarse para poder así subir sus cupones participantes, para que posteriormente los usuarios puedan observarlos en la app móvil y poder así canjearlos.

### 1.5.2 Limitaciones

La principal fue el desconocimiento del funcionamiento de algunas de las herramientas empleadas en el desarrollo de la aplicación. Se volvió necesario el investigar sobre dichas herramientas para tener un panorama más claro y poder aplicarlas a la aplicación.

# 2. Fundamento teórico

## 2.1 Proceso de uso de los cupones de descuento

La compra mediante cupones de descuento se basa en el principio de que el cliente al comprar un producto pueden obtener un mejor precio.

En la actualidad muchos establecimientos comerciales utilizan cupones publicitarios o códigos de descuento para atraer a más clientes a su negocio o para extender su cartera de clientes, por lo que dichos cupones les ha sido de gran ayuda para obtener el éxito esperado en sus ventas. Los cupones de descuento o publicitarios son hechos de papel o mediante códigos. Estos cupones de papel o códigos tienen dos características importantes: el descuento del producto o servicio específico y la fecha de caducidad de la promoción. Estas dos características hacen que los cupones no puedan extenderse durante mucho tiempo y que el descuento al precio del producto está marcado en el cupón. Una vez que el cupón es presentado en el establecimiento comercial, el usuario recibe el beneficio marcado según el cupón.

Para las empresas interesadas en llegar a más público, obtener nuevos clientes y aumentar su cartera de clientes futuros es una buena opción el uso de cupones de descuento. Es importante revisar las condiciones de uso como el período de tiempo, porcentajes de beneficio para los usuarios (hasta el 50% del producto rebajado), etc.

## 2.2 UML

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

 El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no específica en sí mismo qué metodología o proceso usar. UML sirve para hacer modelos que permitan:

* Visualizar como es un sistema o como queremos que sea.
* Especificar la estructura y/o comportamiento de un sistema.
* Hacer una plantilla que guíe la construcción de los sistemas
* Documentar las decisiones que hemos tomado.

## 2.3 Modelo de software incremental

Alfredo Weitsenfeld (Weitsenfeld, 2007) dice que es un desarrollo inicial de la arquitectura completa del sistema, seguido de incrementos y versiones parciales del mismo. Cada incremento tiene su ciclo parcial de vida; agrega funcionalidad mejora el sistema. Conforme se completa cada etapa, se verifica e integra la versión con las demás versiones ya completadas del sistema. Durante cada incremento, el sistema se evalúa con respecto al desarrollo de las versiones futuras.

Las actividades se dividen en procesos y subprocesos, dando lugar al término software Factory. Para que la secuencia de desarrollo sea exitosa, es esencial definir etapas que no requieran cambiar los resultados anteriores al agregar nuevas.

Por lo tanto, es importante comprender al inicio los requisitos completos del sistema, algo que normalmente es muy difícil de lograr.

Las siguientes son algunas creencias del modelo incremental:

* La administración de proyectos es más fácil de lograr en incrementos más pequeños.
* Es más fácil comprender y probar incrementos de funcionalidad más pequeños.
* La funcionalidad inicial se desarrolla más temprano, logrando resultados de inversión en menor tiempo.
* Hay más probabilidad de satisfacer el cambio en los requisitos de usuario mediante incrementos del software en el tiempo, que si fueran planeados todos a la vez en un mismo periodo.

Ian Sommerville (Sommerville, 2005) dice que en un proceso de desarrollo incremental los clientes identifican a grandes rasgos, los servicios que proporcionará el sistema. Identifican los servicios que son más y menos importantes. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema. La asignación de servicios a los incrementos depende de la prioridad del servicio con los servicios de prioridad más alta entregados primero.

Una vez que los incrementos del sistema se han identificado, los requerimientos para los servicios que se van a entregar en el primer incremento se definen a detalle, y éste se desarrolla. Durante el desarrollo, se puede llevar a cabo un análisis adicional de requerimientos para los requerimientos posteriores, pero no se aceptan cambios en los requerimientos para el incremento actual.

Una vez que un incremento de completa y entrega, los clientes pueden ponerlo en servicio. Esto significa que tienen una entrega temprana de parte de la funcionalidad del sistema.

## 2.4 Lenguaje de programación Java

Este lenguaje orientado a objetos es compilado generalmente en un bytecode, pero también es posible crear código de máquina nativo. Este bytecode se compila por lo general en el momento de ejecutarse, en forma de código nativo listo para la ejecución, y también existen dispositivos de hardware capaces de ejecutarlos directamente.

La sintaxis de Java es similar a la utilizada por lenguajes C y C++, siendo más sencilla porque elimina herramientas de bajo nivel, tal es el caso de los punteros (Guglielmetti, n.d.).

Durante varios años la biblioteca de clases de Java, requerida para ejecutar los programas realizados en tal lenguaje no estaba publicada como software libre, mientras que el compilador y la máquina virtual conforme a las especificaciones del *Java Community Process* habían sido licenciados bajo GNU (*General Public License)*.

La mayoría de las tecnologías Java fueron liberadas bajo esta licencia libre elaborada por *Free Software Foundation* entre noviembre de 2006 y mayo de 2007, ante la presión de la comunidad de desarrolladores de software libre.

## 2.5 HTML

HTML, siglas de HyperText Markup Language (Lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que, en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, audio y video. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación (HTML5, 2013).

## 2.6 XML

XML, siglas en inglés de *Xtensible Markup Language* ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el *World Wide Web* *Consortium* (W3C) utilizado para almacenar datos en forma legible. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

## 2.7 API REST

[REST, REpresentational State Transfer](http://es.wikipedia.org/wiki/Representational_State_Transfer), es un tipo de arquitectura de desarrollo web que se apoya totalmente en el estándar HTTP.

REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es increíblemente más simple y convencional que otras alternativas que se han usado en los últimos diez años como SOAP y XML-RPC.

REST se definió en el 2000 por Roy Fielding, coautor principal también de la especificación HTTP. Podríamos considerar REST como un framework para construir aplicaciones web respetando HTTP.

Por lo tanto REST es el tipo de arquitectura más natural y estándar para crear APIs para servicios orientados a Internet.

## 2.8 HTTP

*Hypertext Transfer Protocol* o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

HTTP fue desarrollado por el *World Wide Web Consortium* y la Internet *Engineering Task Force*, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1.

HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse (Descripción de la tecnología, 2013).

## 2.9 CSS

*Cascading Style Sheets*. Son hojas de estilo para agregar formato y diseño a una página web (CSS, 2013).

## 2.10 Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como responsive design o diseño adaptativo.

El beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda. Lo que se usa con más frecuencia y que a mi opinión personal me gusta más es el uso de media queries, que es un módulo de CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de la pantalla y si trabajas las dimensiones de tu contenido en porcentajes, puedes tener una web muy fluida capaz de adaptarse a casi cualquier tamaño de forma automática.

## 2.11 NodeJS

NodeJS es un entorno de programación en la capa del servidor basado en el lenguaje de programación [ECMAScript](http://es.wikipedia.org/wiki/ECMAScript" \o "ECMAScript), asíncrono, con [I/O](http://es.wikipedia.org/wiki/I/O) de datos en una [arquitectura orientada a eventos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_dirigida_por_eventos) y basado en el motor [V8](http://es.wikipedia.org/wiki/V8_(motor_JavaScript)) de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, [servidores web](http://es.wikipedia.org/wiki/Servidor_web). Fue creado por [Ryan Dahl](http://es.wikipedia.org/w/index.php?title=Ryan_Dahl&action=edit&redlink=1" \o "Ryan Dahl (aún no redactado)) en 2009 y su evolución está apadrinada por la empresa [Joyent](http://es.wikipedia.org/w/index.php?title=Joyent&action=edit&redlink=1" \o "Joyent (aún no redactado)).

## 2.12 MongoDB

MongoDB (que proviene de «humongous») es la base de datos NoSQL líder y permite a las empresas ser más ágiles y escalables. Organizaciones de todos los tamaños están usando MongoDB para crear nuevos tipos de aplicaciones, mejorar la experiencia del cliente, acelerar el tiempo de comercialización y reducir costes.

Es una base de datos ágil que permite a los esquemas cambiar rápidamente cuando las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta.

MongoDB ha sido creado para brindar escalabilidad, rendimiento y gran disponibilidad, escalando de una implantación de servidor único a grandes arquitecturas complejas de centros multidatos. MongoDB brinda un elevado rendimiento, tanto para lectura como para escritura, potenciando la computación en memoria (in-memory). La replicación nativa de MongoDB y la tolerancia a fallos automática ofrece fiabilidad a nivel empresarial y flexibilidad operativa.

## 2.13 Android

Android es un [sistema operativo](http://es.wikipedia.org/wiki/Sistema_operativo) basado en el kernel de [Linux](http://es.wikipedia.org/wiki/Linux). Fue diseñado principalmente para [dispositivos móviles](http://es.wikipedia.org/wiki/Dispositivo_m%C3%B3vil) con [pantalla táctil](http://es.wikipedia.org/wiki/Pantalla_t%C3%A1ctil), como [teléfonos inteligentes](http://es.wikipedia.org/wiki/Tel%C3%A9fonos_inteligentes) o [tablets](http://es.wikipedia.org/wiki/Tableta_(computadora)" \o "Tableta (computadora)); y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por [Android Inc.](http://es.wikipedia.org/wiki/Android_Inc." \o "Android Inc.), empresa que [Google](http://es.wikipedia.org/wiki/Google) respaldó económicamente y más tarde, en [2005](http://es.wikipedia.org/wiki/2005), compró. Android fue presentado en [2007](http://es.wikipedia.org/wiki/2007) junto la fundación del [Open Handset Alliance](http://es.wikipedia.org/wiki/Open_Handset_Alliance) (un consorcio de compañías de [hardware](http://es.wikipedia.org/wiki/Hardware), [software](http://es.wikipedia.org/wiki/Software) y [telecomunicaciones](http://es.wikipedia.org/wiki/Telecomunicaciones)) para avanzar en los estándares abiertos de los dispositivos móviles.

## 2.14 NPM

En NodeJS el código se organiza por medio de módulos. Son como los paquetes o librerías de otros lenguajes como Java. Por su parte, NPM es el nombre del gestor de paquetes (package manager) que usamos en Node JS.

El gestor de paquetes npm, no obstante, es un poquito distinto a otros gestores de paquetes que podemos conocer, porque los instala localmente en los proyectos. Es decir, al descargarse un módulo, se agrega a un proyecto local, que es el que lo tendrá disponible para incluir. Aunque cabe decir que también existe la posibilidad de instalar los paquetes de manera global en nuestro sistema.

## 2.15 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente dinámico.

Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo (JavaScript, 2013).

## 2.16 AJAX

AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas web sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. (AJAX, 2013)

## 2.17 AngularJS

AngularJS es un [framework](http://es.wikipedia.org/wiki/Framework" \o "Framework) de JavaScript de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto), mantenido por [Google](http://es.wikipedia.org/wiki/Google), que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de [Modelo Vista Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador) (MVC), en un esfuerzo para hacer que el desarrollo y las [pruebas](http://es.wikipedia.org/wiki/Pruebas_de_software) sean más fáciles.

La biblioteca lee el [HTML](http://es.wikipedia.org/wiki/HTML) que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticas o dinámicas.

## 2.18 Android Studio

Android Studio es un [entorno de desarrollo integrado](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) (IDE) para la plataforma [Android](http://es.wikipedia.org/wiki/Android" \o "Android). Fue anunciado por Ellie Powers el 16 de mayo de 2013. Android Studio está disponible para desarrolladores para probarlo gratuitamente. Basado en IntelliJ Idea de JetBrains, está diseñado específicamente para desarrollar para Android. Está disponible para descargar para Windows, Mac OS X y Linux.

## 2.19 Eclipse

Eclipse es un [programa informático](http://es.wikipedia.org/wiki/Programa_inform%C3%A1tico) compuesto por un conjunto de herramientas de programación de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar [entornos de desarrollo integrados](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado) (del inglés IDE), como el IDE de [Java](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

## 2.20 jQuery

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica el recorrido del documento HTML, manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido. jQuery está diseñado para cambiar la forma en que se escribe JavaScript (jQuery, 2013).

## 2.21 JSON

JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON.

En JavaScript, un texto JSON se puede analizar fácilmente usando la función eval(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web (JSON, 2013).

## 2.22 IONIC Framework

Ionic Framework es de Open Source para el desarrollo de aplicaciones móviles, que utilizan tecnologías web y se integra completamente con AngularJS. Fue desarrollado por la empresa Driftyco.

Ventajas:

* Animaciones aceleradas por hardware
* Mínima manipulación del DOM
* No usa JQuery
* Fácil adaptación del diseño

## 2.23 CORDOVA Framework

[Apache Cordova](https://cordova.apache.org/) es un **framework** de licencia libre que cuenta con muchas Apis de diversos dispositivos móviles para desarrollar aplicaciones nativas dentro de un **smartphone**. Cada vez está tomando más énfasis en el mundo de los programadores y es que para el desarrollo de las aplicaciones se utilizan las tecnologías web HTML, CSS y **JavaScript**.

Una de las grandes peculiaridades de este entorno de trabajo es la posibilidad de desarrollar para iOS, Android y demás sistemas operativos sin la necesidad de programar en sus lenguajes nativos (Java, **Objetive-C**, etc.).

## 2.24 Socket.IO

Socket.IO es una librería que nos permite manejar eventos en tiempo real mediante una conexión TCP y todo ello en JavaScript. Es realmente potente y podemos hacer todo tipo de aplicaciones en tiempo real.

## 2.25 GitHub

Es una plataforma de desarrollo colaborativo de software (forja), alojando proyectos utilizando el sistema de control de versiones Git almacenado en un repositorio ubicado en la nube (Internet). Se creó un repositorio privado donde se alojaría la aplicación durante el desarrollo, este vinculado al IDE NetBeans, donde se verían reflejados cada una de las actualizaciones en las versiones del software. (Paramio, 2011)

## 2.26 Enterprise Architect

Es una herramienta desarrollada como auxiliar en el diseño y análisis UML de software, cubriendo el desarrollo del mismo desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. (Enterprise Architect, 2012)

## 2.27 ProjectLibre

Es un [software de administración de proyectos](http://es.wikipedia.org/wiki/Software_de_administraci%C3%B3n_de_proyectos) (ProjectLibre, 2012), fue lanzado en agosto de 2012. Entre sus características se encuentran:

* Creación de diagramas de Gantt
* Creación de gráficos PERT
* Informes de uso de tareas.

## 2.28 Diagramas de Casos de Uso

Define una notación gráfica para representar casos de uso en los que los diferentes actores del sistema interactuarán.

## 2.29 Diagramas de Secuencias

Define de una forma más detallada la interacción de los usuarios en el sistema, así como el flujo de las tareas o procesos que se ejecutan dentro de este, podría considerarse como una evolución que conjunta lo modelado en los diagramas de actividades junto con los casos de uso.

## 2.30 Diagramas de Actividades

También conocido como Diagramas de Flujo, es la representación gráfica de los algoritmos o procesos dentro de un negocio.

## 2.31 Diagramas Entidad Relación

Diagramas en los cuales se abstraen las categorías y las propiedades de las tablas que interactúan en un sistema, así como su relación con otras tablas. La realización de estos diagramas es de gran utilidad, ya que facilitan la tarea de la creación de la base de datos.

# 3. Procedimiento y descripción de las actividades realizadas

Ya que la aplicación es proyecto de software, las actividades que fueron realizadas se basaron en las fases que comprende el ciclo de vida del software y a continuación se detalla cada una de ellas. (Sommerville, 2005)

## 3.1 Obtención de requerimientos

Para obtener la lista de requerimientos sobre las que se basaría la aplicación, fue necesario acudir con el líder de proyecto en TERACOM.mx para realizar una serie de entrevistas y poder así entender las necesidades de los módulos a desarrollar.

A la par que iban obteniendo las necesidades que tenía que cumplir la aplicación, dichas necesidades se iban plasmando en el documento de requerimientos tanto funcionales como no funcionales.

## 3.2 Análisis de requerimientos

Cómo dicta el ciclo de vida del software, el análisis de los requerimientos es una de las partes más importantes de este proceso. Entre más completo sea el análisis de los requerimientos, existe una menor probabilidad de realizar cambios durante el desarrollo o procesos de re-ingeniería. Adicionalmente, cuando se hace un análisis a profundidad, es posible reducir el tiempo de codificación al haber contemplado las situaciones que pudieran presentarse previamente.

La aplicación se basó en el tipo de arquitectura Representational State Transfer que es la Transferencia de Estado Representacional (REST), la cual nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que entienda HTTP, por lo que es más simple y convencional que otras alternativas como SOAP y así dicha arquitectura contar con la capacidad de responder adecuadamente los requerimientos del resto de la aplicación.

### 3.2.1 Interfaz gráfica

Para la parte visual de la aplicación se decidió utilizar los frameworks Ionic y Bootstrap, tanto para la aplicación móvil y para la aplicación web respectivamente, también se determinó que el uso de estas herramientas sería lo ideal ya que permitiría la estandarización de toda la aplicación de una manera rápida y sencilla, además que mediante el uso de esta herramienta estaríamos asegurando que la aplicación tuviera un Diseño Responsivo además que brindaría al usuario una interacción agradable con la aplicación.

### 3.2.2 Servicios

Los servicios llevan a cabo las tareas con la base de datos, es decir, se procesan las reglas del negocio.

Al ser una aplicación con arquitectura API REST es necesario llevar a cabo la comunicación con la interfaz gráfica por medio de servicios NodeJs y AngujarJs, ya que al ser una herramienta Javascript que trabaja con eventos del lado del servidor, facilita las transacciones realizadas con la base de datos la cual se realiza con sintaxis JSON.

Se encontró que sería necesario también utilizar algunas herramientas que mejoran el desempeño de la aplicación y aumentan la seguridad, como es el caso de Socket.io para el manejo de las peticiones a los servicios y AngularJs para el uso de objetos similares a un mapeo de base de datos en Hibernate Framework.

### 3.2.3 Eventos

NodeJs es el encargado del manejo y gestión de las peticiones así como el manejo de comportamientos que ocurren en el sistema, además de que nos ayuda en la determinación del flujo de la aplicación. Los eventos están relacionados con la interfaz gráfica y están a la espera de que ocurra alguna acción para atenderla con lo cual le brinda una mayor escalabilidad a la aplicación.

### 3.2.4 Base de datos

Actualmente encontramos que en la mayoría de aplicaciones los datos se almacenan en sistemas de bases de datos relacionados llamados RDBMS (Manejadores de sistemas de datos relacionales) como Oracle y MySQL, que utilizan un esquema de tablas con columnas para identificar cada campo y definir el tipo de dato que contendrá. Además de que para acceder a los datos se debe utilizar un lenguaje de consultas estructurado llamado SQL.

Para esta aplicación que requiere de mayor escalabilidad sería complicado utilizar una base de datos como Oracle o MySQL por lo que se decidió utilizar una base de datos no relacional como MongoDB que es de código abierto y es una gran alternativa además de que utiliza de sintaxis JSON para hacer la comunicación con la aplicación y realizar las tareas más comunes con una base de datos como inserciones, actualizaciones o eliminación de registros.

Unas de las principales características de MongoDB es que está diseñado para operar con grandes cantidades de datos de manera rápida. Es por eso que almacena la información en archivos lo que utiliza la memoria como un mecanismo de persistencia.

### 3.2.6 Gestor de paquetes NPM

Los frameworks que se utilizaron para el desarrollo de la aplicación, a su vez requerían que ciertos paquetes estuvieran instalados en el proyecto para que pudieran funcionar. Realizar manualmente ésta gestión habría significado varias horas de tiempo invertido y no se estaría cumpliendo con los requerimientos funcionales de escalabilidad y flexibilidad pues si se decidiera cambiar posteriormente la versión de alguno de los frameworks podría ser necesario actualizar también los paquetes de cada uno.

En lugar de realizar la gestión de paquetes de forma manual, se delegó ésta tarea a otro framework de nombre NPM, el cual tiene ventajas al encontrarse en este caso, ya que permite la descarga automática de las dependencias a través de la configuración de un archivo JSON (package.json).

Una de las características que fue de mayor ayuda para el desarrollo es que NPM provee una manera sencilla para distribuir código a través de un repositorio global. Este gestor de paquetes es la principal herramienta para manejar dependencias o módulos en NodeJs.

## 3.3 Modelado de la Aplicación

De acuerdo a las fases del ciclo de vida del software, una vez que se completó la obtención de requerimientos y el análisis del mismo, se procedió a modelar los módulos a desarrollar.

Es en este paso del proceso en donde se muestra a nivel aplicación lo que se obtuvo en los requerimientos, los cuales tienen como funcionalidad el plasmar de una manera comprensible para el desarrollador que es lo que se va a hacer, además de que dichos diagramas forman parte importante para la documentación final de la Aplicación en general.

Para este proceso se usó al UML (Lenguaje Unificado de Modelado), el cual es un lenguaje que permite representar a través de una serie de diagramas, cada uno de los requerimientos del software, de ésta manera y al ser un lenguaje unificado, se reduce la ambigüedad y ayuda a entender mejor la forma en cómo se ha de resolver un problema. En el caso de este desarrollo fue necesario debido a que la aplicación se desarrolló por etapas teniendo así un modelado por cada una de las etapas. En caso de que no se hubiera usado un estándar, habría sido realmente compleja la tarea de entender el funcionamiento de la aplicación.

De acuerdo a las reglas que dicta el UML se modelaron las siguientes funcionalidades en el sistema:

* **Diagramas de Actividades.**
* **Diagramas de Casos de Uso.**
* **Diagramas de Secuencia.**
* **Diagramas entidad relación.**

El diagrama entidad relación es una herramienta sumamente útil para representar las entidades que componen a un sistema de comunicación ya que además permite apreciar sus relaciones y las propiedades de éstos. Es muy parecido al diagrama de clases, sin embargo es en éste en donde se puede apreciar cada una de las tablas, los campos, sus tipos de datos, las llaves primarias, foráneas, y a los desarrolladores les otorga un panorama completo de la aplicación.

## 3.4 Diseño de la aplicación

Se realizaron interfaces prototipos sin funcionalidad alguna, únicamente con el fin de tener la aprobación del usuario y así pasar a la siguiente fase del ciclo de vida del software, fase de desarrollo.

Debido a la metodología incremental que se llevó a cabo, estuvo en constantes revisiones por parte del usuario que para ésta aplicación es el líder de proyecto de TERACOM.mx, para luego ser sometido a modificaciones de acuerdo a sus indicaciones, hasta que se alcanzaron los resultados esperados y se cubrieron los requerimientos redactados.

## 3.5 Desarrollo de la aplicación

Cuando se obtuvo el visto bueno del diseño por parte del líder del proyecto, se procedió a la codificación o desarrollo de la aplicación, para esto fue necesario preparar inicialmente el ambiente de desarrollo apoyado de algunas herramientas como Eclipse, MongoDB y GitHub, esta última fue utilizada para el manejo de versiones de código, facilitando la subida de cambios al proyecto, además de su revisión en cuento a la codificación y arquitectura, por parte del líder del proyecto.

Durante esta parte del ciclo de vida del software la aplicación se dividió por etapas que fueron desarrolladas en el siguiente orden:

1. **Alta de Cupones:** Al ser la aplicación un catálogo de cupones que los usuarios irán dando de alta en el sistema, es necesario un módulo donde los usuarios registren los cupones de descuento, por eso es necesario contar con todos las características del cupón. Para que un cupón pueda ser registrado son necesarios datos como la duración del cupón, es decir, el periodo en que será vigente la promoción, la imagen que será la del cupón donde muestre a detalle el producto o servicio que promociona el cupón y a que tienda permanece, ya que un usuario igualmente previamente registrado, puede registrar un cupón.
2. **Visualización de Cupones:** La función principal de la aplicación es la visualización de los cupones de descuento que se han registrado, es por ello que es necesario un módulo específico para el desarrollo de la interfaz que mostrará los cupones ya que requiere de ver las imágenes de los cupones las cuales pueden extenderse para poder ser vistas a detalle, además de que cada cupón puede tener comentarios por pate de los usuarios.

Los cupones deben mostrarse por imagen del cupón, además de la vigencia y la ubicación del establecimiento que promociona el cupón más cercano según el requerimiento que se ha establecido anteriormente.

1. **Gestión de Usuarios:** Dentro de la aplicación los usuarios son quienes realizan todas las tareas que se llevan a cabo con los cupones es por eso que es necesario desarrollar un módulo que gestione a los usuarios desde el momento en que un usuario se registra hasta que elige un cupón para su visualización. Es importante saber que para registrar un cupón se necesita estar previamente registrado en la aplicación y cubrir ciertos requisitos. Los usuarios pueden registrarse ellos mismos, con para lo cual la aplicación le mostrará un formulario con los datos requeridos para su registro. La ubicación de su local comercial o establecimiento es un dato indispensable ya que cuando registre un cupón esté tomara la ubicación de referencia para que un usuario que se ha interesado en dicho cupón pueda localizar el establecimiento y así hacer válida la promoción del cupón.

Bajo estos requerimientos surgen también los usuarios que no se registran, los cuales sólo pueden ver cualquiera de los cupones que se muestran en la aplicación, estos usuarios no pueden registrar cupones ya que no están registrados en el sistema.

Para cumplir con los requerimientos es necesario que un módulo que gestione todos los usuarios y las sesiones que se generan por medio de los usuarios, tomando en cuenta la seguridad requerida en el sistema, la concurrencia y eficiencia de la aplicación y las reglas del negocio.

1. **Geolocalización:** Dentro de la aplicación también es necesario el uso de la ubicación geográfica ya que cada usuario que registra cupones establece dentro de un mapa la localización de su negocio al cual acudir para realizar el canje de los cupones de descuento, es por ello que el módulo de geolocalización es importante dentro del funcionamiento de la aplicación. Además de que actualmente la localización geográfica de lugares y establecimientos se ha vuelto una tarea indispensable en aplicaciones de este tipo, lo que las hace aún más atractivas para los usuarios.

Se hizo uso de todas las herramientas y metodologías descritas en la parte del análisis:

1. **Metodologías**
   1. Iterativo incremental
   2. API REST
2. **Herramientas**
   * NodeJS
   * Ionic Framework
   * AngularJs
   * MongoDB
   * HTML5
   * CSS
   * JavaScript
   * JQuery Framework
   * JSON Framework
   * Ajax
   * Socket.io

Para complementar el desarrollo y evitar las ambigüedades, se continuó con el uso de las convenciones de desarrollo establecidas al inicio del proyecto para nombrar las tablas, clases, variables, archivos, controladores, funciones, procedimientos y vistas.

## 3.6 Pruebas e implementación

### 3.6.1 Pruebas

En primer lugar para esta etapa del ciclo de vida del software, fue necesario realizar pruebas unitarias de cada uno de las etapas desarrolladas, esto nos ayuda a corroborar que la aplicación cumple con cada uno de los requerimientos por los que fueron desarrollados correctamente. Estas pruebas unitarias fueron aplicadas cada una de las etapas del desarrollo de la aplicación, desde la alta de cupones, hasta la gestión de los usuarios.

Una vez que cada uno de la aplicación pasó las pruebas unitarias, se procedió a realizar las pruebas de integración, estas pruebas no son otra cosa más que verificar que los módulos pudieran integrarse sin alterar ningún funcionamiento, o bien causar alguna anormalidad en el flujo general de la aplicación, o alterar la funcionalidad de los módulos que ya estaban probados previamente.

Finalmente una vez que se completaron con éxito todas las pruebas de sistema se procedió a las pruebas de Usuario, que son las más importantes ya que son aquellas donde el usuario evalúa el sistema y prueba que realmente el sistema haga lo que él solicito en los requerimientos, dichas pruebas van acompañadas de una matriz de pruebas que realiza el usuario, para evaluar todos los posibles casos que él requiere dentro del sistema.

Cabe mencionar que durante todo el desarrollo de la aplicación se tuvieron tres ambientes donde se tenía la aplicación instalada: Ambiente de Desarrollo, Ambiente de Calidad y Ambiente de Producción.

El Ambiente de Desarrollo se utilizó como bien su nombre lo dice para llevar acabo el desarrollo y todas las pruebas de nuevo código en la aplicación. El ambiente de Calidad es un entorno controlado donde no se permitía subir cambios a menos que estos estuvieran probados en el ambiente de desarrollo y el usuario haya aprobado dichos cambios.

Finalmente el Ambiente de Producción que es el entorno que se tiene en funcionamiento y por obvias razones la migración de código a este entorno deberá ser en horas poco recurrentes y con la aprobación del Líder del Proyecto.

Ahora bien, para llevar a cabo las pruebas de usuario se utilizó el ambiente de calidad que como se mencionó es un ambiente controlado y libre de errores, además se utilizó la herramienta WebIssues para llevar el control de los Issues, que no son otra cosa más que los defectos encontrados en la aplicación.

Cada Issue reportado debería de contener la siguiente información:

**Estado**:

* **Activo:** el issue se reporta con este estado inicial.
* **Resuelto:** el issue está corregido y listo para que se vuelva a realizar prueba
* **Concluido:** el issue está corregido y revisado, se reporta que el issue no presenta problemas a la aplicación

**Razón:** Corregido**,** Obsoleto**,** Duplicado**,** Así fue diseñado**,** No es posible su reproducción**,** Prueba fallida.

Se acordó con el líder de Proyecto que se realizarían 3 ciclos de prueba.

### 3.6.2 Implementación

Una vez que se concluyeron los tres ciclos de prueba con éxito, se procedió a la elaboración de los manuales de Usuario.

Amazon Web Services (AWS) ofrece recursos y servicios en la nube bajo demanda, es decir, se puede ejecutar un servidor en el cual se puede iniciar sesión y así configurar, proteger y ejecutar como si fuera un servidor que se tiene físicamente. Es por eso que se decidió comprar los servicios de AWS para alojar la aplicación, con lo cual el ambiente de producción fuera a la medida de las necesidades de la aplicación.

El manual de Usuario contiene toda la información acerca de cómo manejar el Sistema, tanto como por el lado del administrador como por el lado del Usuario, cabe mencionar que un aspecto importante de este manual es una redacción clara y concisa para que cualquier persona que lo lea pueda hacer uso del Sistema sin problemas, además de que dicha redacción está apoyada por una gran cantidad de imágenes que facilitan la comprensibilidad del mismo.

# Resultados, planos, gráficas, prototipos y programas

## 4.1 Obtención de requerimientos

### 4.1.1 Documento de requerimientos

La aplicación web para la promoción de cupones será un producto diseñado para trabajar en entornos web y en dispositivos móviles, lo que permitirá su utilización de forma rápida y eficaz, además de que requiere una integración entre la interfaz web y la interfaz para dispositivos móviles.

### 4.1.2 Características de los usuarios

|  |  |
| --- | --- |
| Tipo de usuario | Usuarios no registrados |
| Actividades | Puede acceder a la vista principal de los cupones y puede buscar un establecimiento en el mapa. |

|  |  |
| --- | --- |
| Tipo de usuario | Usuario “Promotor” |
| Actividades | Puede registrar cupones y acceder a todos los aparados del sistema, excepto a la administración de los usuarios. |

|  |  |
| --- | --- |
| Tipo de usuario | Usuario “Visor” |
| Actividades | Puede acceder a todos los apartados del sistema pero no puede registrar cupones, excepto a la administración de los usuarios. |

|  |  |
| --- | --- |
| Tipo de usuario | Usuario administrador |
| Actividades | Puede acceder a todos los apartados del sistema, además de que tiene acceso a la vista de administración de usuarios “promotores”. |

### Requisitos específicos

##### Interfaces de usuario

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF 01 | | |
| Nombre de requisito | Todas las páginas web deben ser uniformes en su contenido, es decir, deben contener el logo y la misma gama de colores en cada una de ellas. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

##### Interfaces de software

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF 02 | | |
| Nombre de requisito | El sistema se integrará completamente a la aplicación móvil en cuanto a su funcionamiento y al diseño antes de la fase de implementación. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

### Requisitos funcionales

#### Registro al sistema de la cuponera como usuario “promotor”

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 01 | | |
| Nombre de requisito | Registro al sistema de la cuponera como usuario “promotor”. | | |
| Descripción | Cualquier usuario puede registrarse al sistema de la cuponera como usuario “promotor” siempre y cuando cumpla con llenando los campos que deben ser, nombre(s), apellido paterno, apellido materno, edad, usuario, contraseña, tipo de negocio que tiene, nombre del establecimiento, dirección o direcciones del establecimiento y puesto que ocupa el usuario dentro del establecimiento. | | |
| Actores | Cualquier usuario | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Registro al sistema de la cuponera como usuario “visor”

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 02 | | |
| Nombre de requisito | Registro al sistema de la cuponera como usuario “visor”. | | |
| Descripción | Cualquier usuario puede registrarse al sistema de la cuponera como usuario “visor” siempre y cuando cumpla con llenando los campos que deben ser, nombre(s), apellido paterno, apellido materno, edad, usuario, contraseña e intereses. | | |
| Actores | Cualquier usuario | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Registro de cupones

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 03 | | |
| Nombre de requisito | Registro de cupones | | |
| Descripción | Los usuarios registrados en el sistema pueden registrar cupones para promocionar su establecimiento, por lo que es necesario para registrar un cupón la subida de la imagen del cupón así mismo el periodo de validez del cupón. | | |
| Actores | registrados | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Restricción de las medidas del cupón

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 04 | | |
| Nombre de requisito | Restricción de las medidas del cupón | | |
| Descripción | Los cupones que se registran por usuario deben ser de 300px de alto por 300px de ancho. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Vista de inicio

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 05 | | |
| Nombre de requisito | Vista de inicio | | |
| Descripción | Al inicio de la aplicación cualquier usuario debe de mostrar la lista de cupones que se han registrado. | | |
| Actores | Cualquier usuario | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Vista de cupones

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 06 | | |
| Nombre de requisito | Vista por cupones | | |
| Descripción | Los cupones pueden expandirse para mejorar su visualización, en la parte inferior debe contener los botones Favorito, Comentar, Buscar. | | |
| Actores | Cualquier usuario. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Botón Favorito

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 07 | | |
| Nombre de requisito | Botón Favorito | | |
| Descripción | El botón favorito que se encuentra en la parte inferior de la imagen del cupón, al momento de ser presionado debe agregar al botón a una lista de favoritos y así se guarden sus preferencias. | | |
| Actores | Usuarios registrados. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Botón Comentar

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 08 | | |
| Nombre de requisito | Botón Comentar | | |
| Descripción | El botón comentar que se encuentra en la parte inferior de la imagen del cupón, al momento de ser presionado debe agregar un comentario para el cupón. | | |
| Actores | Usuarios registrados. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Botón Buscar

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 09 | | |
| Nombre de requisito | Botón Buscar | | |
| Descripción | El botón Buscar que se encuentra en la parte inferior de la imagen del cupón, al momento de ser presionado debe de llevar a la interfaz con el mapa ubicando al negocio al que pertenece el cupón. | | |
| Actores | Cualquier usuario | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Vista Favoritos

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 010 | | |
| Nombre de requisito | Vista Favoritos | | |
| Descripción | La vista de Favoritos muestra la lista de cupones que el usuario agregó como favoritos y que además la fecha de validez del cupón es posterior a la fecha actual. | | |
| Actores | Usuarios registrados | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Vista Buscar

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 11 | | |
| Nombre de requisito | Vista Buscar | | |
| Descripción | La vista Buscar muestra el mapa ubicando a los negocios más cercanos tomando en cuenta la ubicación actual del usuario. | | |
| Actores | Cualquier usuario. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Edición de datos de usuario

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 12 | | |
| Nombre de requisito | Edición de datos del usuario. | | |
| Descripción | Los usuarios registrados pueden acceder una vista donde puedan editar sus datos. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Restricción comentarios de cupón

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 13 | | |
| Nombre de requisito | Restricción comentarios de cupón. | | |
| Descripción | La longitud de los comentarios no debe exceder los 140 caracteres. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Restricción lista de favoritos

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 14 | | |
| Nombre de requisito | Restricción en lista de favoritos. | | |
| Descripción | La lista de favoritos debe mostrar los cupones que el usuario ha agregado a tal lista y que además la fecha de validez del cupón sea posterior a la fecha actual. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Restricción vista de cupones

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 15 | | |
| Nombre de requisito | Restricción en vista de cupones. | | |
| Descripción | Sólo deben mostrarse los cupones en los que la fecha de validez del cupón sea posterior a la fecha actual. | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Eliminación de cupones

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 16 | | |
| Nombre de requisito | Eliminación de cupones | | |
| Descripción | Los cupones se eliminarán automáticamente siempre y cuando la fecha de validez de cupón haya expirado, de otro modo cuando el usuario decida eliminar un cupón. | | |
| Actores | Sistema | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Formato de imágenes para cupones permitidos

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 17 | | |
| Nombre de requisito | Formato de imagen para cupones permitidos. | | |
| Descripción | Los formatos de imagen aceptados para la Anexo de cualquier tipo de archivo son: JPEG Y PNG | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Inicio de Sesión

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 18 | | |
| Nombre de requisito | Inicio de sesión | | |
| Descripción | Todos los usuarios registrados pueden acceder a su sesión ingresando los datos de: usuario y contraseña, y la pantalla de inicio es la que se requirió anteriormente, en caso de que no esté registrado se agregará el error y un vínculo para que el usuario se registre. | | |
| Actores | Usuarios registrados | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Administración de usuarios

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 19 | | |
| Nombre de requisito | Administración de usuarios. | | |
| Descripción | Todos los usuarios registrados como “promotores” podrán verse desde una vista a la que sólo un usuario tendrá acceso y que será el administrador. | | |
| Actores | Usuario administrador | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

#### Funcionalidad uniforme

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF 20 | | |
| Nombre de requisito | Funcionalidad uniforme | | |
| Descripción | Todos los requerimientos deben ser visibles y funcionales en entorno web y móvil. | | |
| Actores | Usuario administrador | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Entrevista a líder de proyecto. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

## 4.2 Modelado del sistema

Los requerimientos que se obtuvieron fueron modelados en diferentes diagramas, tanto de casos de uso, como de secuencias y de clases con ayuda del software Enterprise Architect.

### 4.2.1 Diagramas de casos de uso

La figura 4.1 se muestra el proceso para que un usuario registrado o no registrado pueda ver los cupones, además de los comentarios de los cupones y ver el mapa de ubicación de cupones, con la validez de fecha por parte del sistema y la ubicación en el mapa de las direcciones de los usuarios proveedores.



Figura 4.1 Diagrama de Casos de Uso Cuponera App - Usuarios

La figura 4.2 se muestra el proceso para que un usuario “visor” pueda evaluar y comentar un cupón, además, pueda agregarlo a su lista de favoritos.

Figura 4.2 Diagrama de Casos de Uso Cuponera App - Visor

La figura 4.3 se muestra el proceso para que un usuario “proveedor” pueda registrar un cupón, registrar ubicaciones, evaluar y comentar un cupón además de que pueda agregarlo a su lista de favoritos.



Figura 4.3 Diagrama de Casos de Uso Cuponera App - Proveedor

La figura 4.4 se muestra el proceso para que un usuario “administrador” pueda llevar a cabo la administración de los usuarios, tomando en cuenta las tareas de eliminar usuario y eliminar cupones.



Figura 4.4 Diagrama de Casos de Uso Cuponera App - Administrador

### 4.2.2 Diagramas de secuencias

A partir de la Figura 4.5, se describe la secuencia que deben realizar para poder interactuar con los diferentes usos de la aplicación, tanto para los usuarios no registrados y registrador como son, los usuarios visor y proveedor.

En la figura 4.5 se describe el proceso que se lleva acabo para que un usuario pueda ingresar al sistema.

En la figura 4.6 se describe el proceso por el cual se debe de pasar para que el usuario pueda iniciar sesión, según sea un usuario proveedor, visor o administrador.



Figura 4.5 Diagrama de Secuencia Cuponera App – Registrar Usuario

La Figura 4.7 describe el proceso para que un usuario registrado o no registrado pueda ver la lista de cupones y para que el sistema verifique la fecha de validez de los cupones, con lo cual toma puede agregarlo a la lista de cupones o eliminar el cupón.



Figura 4.6 Diagrama de Secuencia Cuponera App – Inicia Sesión

La Figura 4.8 describe el proceso para que un usuario registrado o no registrado pueda ver el mapa donde se muestra la ubicación de los proveedores de los cupones registrados.



Figura 4.7 Diagrama de Secuencia Cuponera App – Lista de Cupones

La Figura 4.9 describe el proceso para que un usuario registrado pueda agregar un cupón a la lista de favoritos además de que pueda evaluar o/y comentar un cupón.



Figura 4.8 Diagrama de Secuencia Cuponera App – Ver Mapa

La Figura 4.10 describe el proceso para que un usuario registrado pueda visualizar su lista de los cupones que ha marcado como favoritos.



Figura 4.9 Diagrama de Secuencia Cuponera App – Comentar, evaluar y agregar a Favoritos

En la Figura 4.11 se describe el proceso con el cual un usuario proveedor puede registrar ubicaciones en el mapa.



Figura 4.10 Diagrama de Secuencia Cuponera App – Lista de Favoritos

En la Figura 4.12 se describe el proceso con el cual un usuario proveedor puede registrar un cupón.



Figura 4.11 Diagrama de Secuencia Cuponera App – Agregar Ubicaciones

En la figura 4.13 se muestra el proceso que se debe de seguir para que el usuario administrador pueda eliminar un usuario y ver la lista de usuarios registrados en sistema.



Figura 4.12 Diagrama de Secuencia Cuponera App – Registrar Cupón

### 4.2.3 Diagrama Entidad Relación

En la figura 4.14 se muestra el diagrama entidad relación que representa la base de datos que se utilizó para esta aplicación.

En la figura 4.15 se muestra el diagrama JSON con el cual se establece la estructura con la cual serán guardados los documentos en la base de datos de MongoDB.



Figura 4.13 Diagrama de Secuencia Cuponera App – Administración de Usuarios

## Desarrollo del sistema

Es en ésta etapa en donde se realizó la codificación de la aplicación, sin embargo no fueron las únicas tareas que se realizaron.

### 4.3.1 Convenciones de nombres

Con la finalidad de mantener un orden en el desarrollo de la aplicación y de evitar problemas posteriores derivados del nombre de los componentes, se optó por usar convenciones para los nombres de clases, métodos, paquetes, carpetas, variables y objetos.

**Clases**

* El nombre comienza con mayúscula
* Los nombres deben ser sustantivos.
* Cuando un nombre conste de varias palabras, se escribirán una tras otra, sin solución de continuidad, comenzando cada palabra por mayúscula.

**Métodos**

* El nombre debe ser un verbo, ejemplo: correr, obtener, traer, llevar, partir, etc. Se debe escribir en minúscula y si es compuesto de varias palabras se debe iniciar la siguiente palabra con mayúscula.
* Si el nombre es compuesto por varias palabras, la primera palabra es un verbo y la segunda un adjetivo.

**Paquetes**

* Los nombres de los paquetes deben usar solo minúsculas.

**Carpetas**

* Inicia con mayúsculas

**Variables**

* El nombre comienza con minúscula.
* cuando un nombre conste de varias palabras, se escribirán una tras otra, sin solución de continuidad, comenzando cada palabra por mayúscula.

**Objetos**

* Iniciar con el nombre del clase de la cual se instancio y continuar con la palabra OBJ.

### 4.3.2 Vistas de la Aplicación Web.

A continuación se muestran las pantallas correspondientes a la aplicación web, y que responden a los requerimientos que se obtuvieron.

### 4.3.3 Vistas de la Aplicación Móvil.

A continuación se muestran las pantallas correspondientes a la aplicación móvil, y que responden a los requerimientos que se obtuvieron.

## Recomendaciones y Conclusiones

El que tuviera la oportunidad de liberar mi residencia profesional con este proyecto, pienso que fue un gran aporte hacia mi perfil como profesional, ya que en el tiempo que estuve desarrollando éste proyecto pude aprender cosas muy importantes acerca de mi carrera que quizá se vieron en los salones de clase, pero nunca se les dio el enfoque a un proyecto real como en este caso.

Una de las cosas más importantes que pude aprender durante mi estadía en este proyecto es que no todo se trata de ir directamente a programar, ya que como bien nos lo muestran en las asignaturas relacionadas con la Ingeniería de Software, tenemos muchas herramientas como ingenieros para hacer un análisis de todo lo que nos solicita nuestro usuario, y modelar todas sus necesidades en diagramas y procesos que nos faciliten plasmar dichas necesidades en un producto final.

La comunicación con el usuario en este proyecto considero que fue de suma importancia, ya que se realizaron diversas entrevistas las cuales nos facilitaron la obtención de requerimientos, y con esto se definió con lujo de detalle que era lo que usuario requería visualizar en el sistema y las actividades que debería de poder llevar acabo en el mismo, después de realizar varios ciclos de entrevistas con el usuario finalmente se obtuvieron todas las necesidades del sistema, dichas necesidades posteriormente se tuvieron que redactar en forma de requerimientos funcionales y no funcionales, detallando que actividad cubriría cada requerimiento y el grado de importancia del mismo. Una vez que se tuvo el documento de requerimientos completo, se prosiguió a presentar dicho documento con el usuario para que este nos diera su firma de consentimiento de lo que se iba a desarrollar, esto con el fin de evitar problemas de desarrollar cosas que no se pidieron y tener algo con que justificar todo lo que se desarrolló. Esto en lo particular me pareció muy importante porque muchas veces nosotros como recién egresados aceptamos todo lo que el cliente nos pide, y como no tenemos un documento de requerimientos que nos respalde, y por ende tenemos que hacer lo que se nos pide aunque este fuera de lo que se acordó de palabra.

Otra cosa que me llevo importante de este proyecto, es todo lo que se tiene que hacer y entregar antes de poder liberar un sistema, como lo son manuales de instalación, manuales técnicos y lo más importante pruebas. Dentro de las pruebas aprendí la importancia de documentar y llevar un adecuado control de las mismas.

Además de todo esto me llevo muchos otros aspectos importantes sobre la correcta administración de un proyecto como lo son: la elaboración de diagramas, el uso de software que ayuda a la administración de tiempos y control de errores, la importancia de la documentación tanto del sistema como de todo lo que se habla en relación al proyecto y lo más importante la comunicación con el usuario y la elaboración de un correcto análisis antes de desarrollar.

Finalmente en el aspecto técnico me llevo muchas cosas, aprendí el uso de nuevas tecnologías como lo son los Frameworks y el desarrollo en dispositivos móviles, que hoy en día facilitan y optimizan mucho el desarrollo de proyectos, además de poder reafirmar y aplicar mis conocimientos en relación a otras tecnologías que ya conocía que también se utilizaron dentro de este proyecto.

## Referencias bibliográficas y virtuales

*AJAX*. (2013). (jQuery) Recuperado el 11 de 10 de 2013, de https://api.jquery.com/jQuery.ajax/

Álvarez, M. Á. (08 de 07 de 2002). *JSP*. Recuperado el 07 de 10 de 2013, de http://www.desarrolloweb.com/articulos/831.php

*CSS*. (08 de 11 de 2013). (Wikipedia) Recuperado el 24 de 10 de 2013, de http://es.wikipedia.org/wiki/Hojas\_de\_estilo\_en\_cascada

*Enterprise Architect*. (2012). (Sparxsystems) Recuperado el 24 de 10 de 2013, de http://www.sparxsystems.com.ar/products/ea.html

*Frameworks*. (14 de 11 de 2013). (Wikipedia) Recuperado el 24 de 10 de 2013, de http://es.wikipedia.org/wiki/Framework

Guglielmetti, M. (s.f.). *Definición de Java*. (MASTERMAGAZINE) Recuperado el 09 de 10 de 2013, de http://www.mastermagazine.info/termino/5470.php

*HTML5*. (15 de 05 de 2013). (Wikipedia) Recuperado el 24 de 10 de 2013, de http://es.wikipedia.org/wiki/HTML5#Diferencias\_entre\_HTML5\_y\_HTML4.2FXHTML

*JavaScript*. (07 de 07 de 2013). (Wikipedia) Recuperado el 24 de 10 de 2013, de http://es.wikipedia.org/wiki/JavaScript

Johnson, M. (1998). *GNU*. (Grulic) Recuperado el 24 de 10 de 2013, de http://www.grulic.org.ar/~mdione/www-grulic/trunk/linux.html

*jQuery*. (05 de 11 de 2013). (The jQuery Foundation) Recuperado el 2013, de http://jquery.com/

*JSON*. (05 de 10 de 2013). (Wikipedia) Recuperado el 05 de 10 de 2013, de http://es.wikipedia.org/wiki/JSON

*Lenguaje de Modelado Unificado (UML)*. (s.f.). (IBM) Recuperado el 05 de 11 de 2012, de http://www-01.ibm.com/software/rational/uml/

Paramio, C. (25 de 05 de 2011). *GitHub contrlador de versiones*. (Gebetadev) Recuperado el 10 de 24 de 2013, de http://www.genbetadev.com/sistemas-de-control-de-versiones/conociendo-github-el-servicio-donde-alojar-tus-repositorios-git-como-el-nuestro

*ProjectLibre*. (2012). (LibreEsfera) Recuperado el 11 de 10 de 2013, de http://www.libresfera.com/projectlibre-nueva-alternativa-a-microsoft-project.html

Sommerville, I. (2005). *Ingeniería de Software.* España: Pearson Educación.

*Spring*. (14 de 08 de 2013). (Wikipedia) Recuperado el 07 de 10 de 2013, de http://es.wikipedia.org/wiki/Spring\_Framework

*Web*. (16 de 03 de 2013). (Wikipedia) Obtenido de http://es.wikipedia.org/wiki/Sitio\_web

Weitsenfeld, A. (2007). *Ingeniería de software orientada a objetos con UML Java e internet.* México: Thomson.