

**Instituto Tecnológico de Costa Rica  
Escuela de Ingeniería en Computación**

Q'Filo: una plataforma que permite ordenar  
comida mediante la web

Trabajo final para optar por el título de Ingeniería en  
Computación con el grado académico de  
Bachillerato

Victor Rojas Sancho

San Carlos Noviembre, 2012

# Tabla de Contenidos

Tabla de Contenidos .....	2
Resumen .....	3
Descripción del problema .....	4
Contexto del proyecto .....	4
Quehacer de la empresa .....	4
Antecedentes del proyecto.....	5
Descripción del problema .....	5
Patrocinadores - Stakeholders.....	6
Perspectivas, supuestos y dependencias del producto .....	6
Requerimientos no funcionales .....	6
Análisis de los Riesgos.....	7
Objetivos y Alcances del sistema .....	9
General.....	9
Específicos.....	9
Alcances del sistema .....	9
Tecnologías utilizadas .....	13
Solución implementada .....	14
Modelo de Diseño.....	14
Arquitectura conceptual de la solución .....	14
Componentes y servicios .....	15
Servidor Web .....	15
Base de datos .....	15
Lenguajes de programación:.....	15
OpenStreetMaps.....	16
Facebook API.....	16
Interfaces de usuario .....	17
Diseño de base de datos .....	25
Conclusiones y comentarios .....	26
Documentos y productos entregados.....	26
Experiencias adquiridas .....	27
Oportunidades de mejoras .....	27
Crecimiento de usuarios .....	29
Referencias.....	31

# Resumen

Las tecnologías y las redes de comunicación como Internet han venido en ascenso en los últimos años. Este ascenso ha impulsado el comercio electrónico que consiste en la compra y venta de productos y servicios a través de dichos medios.

En países desarrollados y ciudades importantes es común acceder a la web para realizar transacciones, comprar accesorios, ropa, dispositivos electrónicos e incluso para ordenar comida. Se obtienen grandes beneficios del comercio electrónico como la capacidad de ver detalladamente la información de un producto, comparar, comprar y vender todo a través de un dispositivo electrónico como la computadora, sin tener que desplazarse hasta una tienda.

La Zona Norte de Costa Rica es una región con acceso a la tecnología, computadoras, dispositivos móviles e internet, a pesar de estos muchas personas de la región no tienen una cultura tecnológica que le permita aprovechar al máximo estas herramientas, se junta a esto la falta de plataformas de comercio electrónico en la región.

Con la idea generar una cultura tecnológica en la región Avantica Technologies ha ideado una aplicación web "Q'Filo", que es una plataforma que permite ordenar comida mediante la web a los restaurantes de la zona que se encuentren registrados, esta plataforma tiene características de interfaces intuitivas para el usuario y un flujo entendible para poder ser utilizada por la mayor cantidad de personas posible.

**Palabras clave:** sitio web, ordenar comida, plataforma web, comercio electrónico.

# Descripción del problema.

## Contexto del proyecto

### Quehacer de la empresa

La empresa se especializa en desarrollo de software a la medida. Está dividida en Departamento de Finanzas, de Recursos Humanos, Ventas y Mercadeo, Producción y Soporte (TI). La práctica se realiza en el departamento de desarrollo, que es una subdivisión del departamento de Producción. En la Figura 1. se muestra el organigrama de Avantica Technologies.

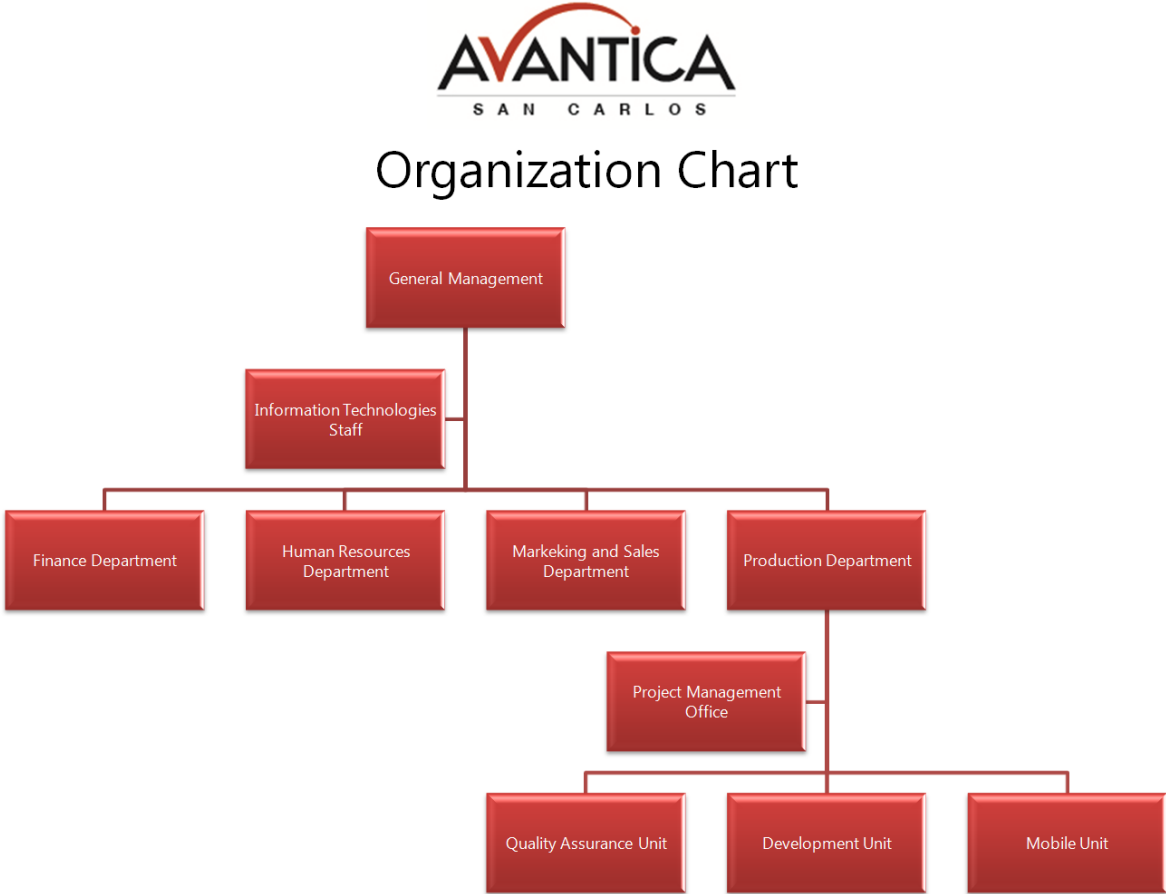


Figura 1: Organigrama de Avantica Technologies

## **Antecedentes del proyecto**

En 1970 empezó a surgir en el mundo el comercio electrónico, esta nueva herramienta abrió un abanico de posibilidades en el uso de internet y medios electrónicos, en la actualidad las ciudades más importantes del mundo y donde la tecnología es de uso diario, es muy común que la personas realicen sus compras a través de internet, artículos personales, contenidos digitales e incluso comida.

## **Descripción del problema**

El problema es que en el cantón de San Carlos existe una barrera cultural que frena el progreso tecnológico de esta zona, se afirma que la barrera es cultural debido que los recursos tecnológicos están presentes y la población tiene acceso a ellos. Este problema evita que se realice inversión de tecnología para la masificación de servicios, afectando esto directamente a al desarrollo y por lo tanto a la población directamente.

Como empresa, Avantica San Carlos adoptó el reto de romper esta barrera, e invertir estratégicamente tanto en cultura como en tecnología, y así lograr una transición progresiva hasta lograr el objetivo de convertir San Carlos en una ciudad de Tecnología.

El primer paso es crear una plataforma que permita realizar pedidos a restaurantes por medios electrónicos. Una plataforma cuyo objetivo no es tener lo último en tecnología sino más bien empezar a moldear la cultura de la comunidad. Se pretende reemplazar las llamadas telefónicas, y los tiempos largos de espera por líneas ocupadas o esperando que la persona que atiende el teléfono menciona el menú.

## **Patrocinadores - Stakeholders**

- Rodrigo Vargas: Gerente de Producción - Se encargará de velar por la ejecución del proyecto y asegurarse de que las personas involucradas tengan el equipo que se requiere para su ejecución.
- Erick Vargas: Arquitecto de Software - Es el líder de la sección de Investigación Desarrollo, y será el encargado de darle seguimiento puntual y técnico al proyecto, de apoyar al estudiante en la ejecución del mismo y de encaminarlo y ayudarlo cuando sea necesario.
- Víctor Rojas: Desarrollador - Desarrollo de software siguiendo estándares de industria y mejores prácticas, y será encargado del desarrollo de la plataforma web para cliente.
- Manrique Varela: Desarrollador - Desarrollo de software siguiendo estándares de industria y mejores prácticas, y será encargado del desarrollo de la plataforma web para restaurantes.

## **Perspectivas, supuestos y dependencias del producto**

- La principal perspectiva es que el producto llegue a ser aceptado y usado por los clientes para que esta plataforma se convierta en una herramienta útil.
- Se cuenta con la arquitectura para publicar la plataforma web mediante los servicios de Avantica Technologies. Además contamos con accesibilidad de los usuarios hacia el Internet mediante dispositivos electrónicos.
- La accesibilidad de la plataforma dependerá del servicio de registro de Facebook ya que es esencial para el manejo de usuarios. Así como la disposición de los restaurantes en utilizar el servicio.

## **Requerimientos no funcionales**

- La plataforma tendrá un tiempo de respuesta aceptable a no mayor de 3.5s por cada petición.
- La aplicación contará con protocolos de seguridad como sesiones encriptadas y contraseñas SHA256, filtrados en formularios contra XSS.
- La interfaz será limpia y agradable para el usuario utilizando colores cafés claros.
- El idioma empleado será únicamente español.
- Las notificaciones de las órdenes deben de ser en tiempo real.

# Análisis de los Riesgos

Los riesgos se han clasificado de acuerdo a su impacto y se ha utilizado la siguiente escala:

- Despreciable: con un valor 1
- Marginal: con un valor 2
- Critico: con un valor 3
- Catastrófico: con un valor 4

## Riesgo número 1: Aceptación de los usuarios

<b>Nombre</b>	Aceptación de los usuarios
<b>Categoría</b>	Usuarios
<b>Causas</b>	Falta de cultura tecnológica y miedo por realizar pedidos en línea.
<b>Impacto</b>	3
<b>Probabilidad de ocurrencia</b>	0.25
<b>Exposición</b>	0.75
<b>Estrategia de evasión</b>	Propaganda y compartir mediante red sociales.
<b>Estrategia de mitigación</b>	Propaganda y compartir mediante red sociales.
<b>Estrategia de contingencia</b>	Propaganda y compartir mediante red sociales.

## Riesgo número 2: Disposición de los restaurantes

<b>Nombre</b>	Disposición de los restaurantes
<b>Categoría</b>	Usuarios
<b>Causas</b>	Indisposición de los restaurantes y/o falta de equipo para la utilización del servicio.
<b>Impacto</b>	3
<b>Probabilidad de ocurrencia</b>	0.25
<b>Exposición</b>	0.75
<b>Estrategia de evasión</b>	Dar a conocer y proponer a restaurantes el servicio. Hacerles constar que cuenta con el respaldo de Avantica Technologies.

<b>Estrategia de mitigación</b>	Dar a conocer y proponer a restaurantes el servicio. Hacerles constar que cuenta con el respaldo de Avantica Technologies.
<b>Estrategia de contingencia</b>	Dar a conocer y proponer a restaurantes el servicio. Hacerles constar que cuenta con el respaldo de Avantica Technologies.

**Riesgo número 3: Acceso a la red**

<b>Nombre</b>	Acceso a la red
<b>Categoría</b>	Comunicaciones
<b>Causas</b>	Fallo en la red
<b>Impacto</b>	2
<b>Probabilidad de ocurrencia</b>	0.10
<b>Exposición</b>	0.20
<b>Estrategia de evasión</b>	Verificar el acceso a la red dentro de la aplicación.
<b>Estrategia de mitigación</b>	Por cada transacción verificar su finalización para informar al usuario.
<b>Estrategia de contingencia</b>	Informar al usuario de que ocurrió un fallo.



# Objetivos y Alcances del sistema

## General

Crear una plataforma web para ordenar comida, que permita generar una cultura tecnológica en el distrito de Ciudad Quesada.

## Específicos

- Diseñar e implementar la base de datos para la plataforma.
- Crear la interfaz gráfica de la aplicación web utilizando HTML/CSS.
- Desarrollar la funcionalidad de la aplicación en el lenguaje PHP, utilizando el modelo MVC con el framework codeigniter.
- Implementar medidas de seguridad contra SQL injection y JavaScript injection.

## Alcances del sistema

<b>Identificación:</b>	C001
<b>Nombre:</b>	Inicio de sesión.
<b>Descripción:</b>	El sistema debe permitir el inicio de sesión mediante la red social Facebook.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C002
<b>Nombre:</b>	Lista de restaurantes.
<b>Descripción:</b>	Los usuarios podrán ver la lista de restaurantes registrados en qfilo.com
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C003
<b>Nombre:</b>	Búsqueda de restaurantes.
<b>Descripción:</b>	Los usuarios tendrán la posibilidad de hacer búsquedas de restaurantes mediante el nombre respectivo o su categoría.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C004
<b>Nombre:</b>	Menú
<b>Descripción:</b>	Los usuarios podrán ver la lista del menú que dispone un restaurante.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C005
<b>Nombre:</b>	Información del restaurante
<b>Descripción:</b>	Los usuarios podrán ver la información correspondiente a un restaurante. Por ejemplo horario, teléfono y dirección.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C006
<b>Nombre:</b>	Pedir
<b>Descripción:</b>	Los usuarios tendrán la opción de realizar pedidos a partir del menú de un restaurante e indicando hora de entrega, teléfono y dirección de la ubicación de cliente y algún comentario.
<b>Categoría</b>	Cliente

<b>Identificación:</b>	C007
<b>Nombre:</b>	Direcciones
<b>Descripción:</b>	El usuario tendrá la opción de editar y guardar un conjunto de direcciones para luego ser escogidas en sus pedidos. Indicando un alias, dirección, detalles y teléfono.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C008
<b>Nombre:</b>	Perfil del cliente.
<b>Descripción:</b>	El usuario tendrá la opción de editar su perfil, tales como nombre, dirección, correo y teléfono.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	C009
<b>Nombre:</b>	Pedidos
<b>Descripción:</b>	Los usuarios tendrán la opción de ver el estado actual de los pedidos realizados. Tales como pendientes, atendidos, entregados.
<b>Categoría</b>	Cliente.

<b>Identificación:</b>	R001
<b>Nombre:</b>	Inicio de sesión
<b>Descripción:</b>	Los restaurantes tendrán la opción de iniciar sesión mediante su usuario y contraseña.
<b>Categoría</b>	Restaurante.

<b>Identificación:</b>	R002
<b>Nombre:</b>	Menú.
<b>Descripción:</b>	Los restaurantes tendrán la opción de agregar editar y eliminar su menú. Además estarán agrupados mediante categorías.
<b>Categoría</b>	Restaurantes.

<b>Identificación:</b>	R003
<b>Nombre:</b>	Ordenar Menú.
<b>Descripción:</b>	Los restaurantes tendrán la opción de acomodar las categorías del menú en la forma que serán mostrados al usuario.
<b>Categoría</b>	Restaurante.

<b>Identificación:</b>	R004
<b>Nombre:</b>	Pedidos.
<b>Descripción:</b>	Los restaurantes serán notificados por nuevos pedidos.
<b>Categoría</b>	Restaurante.

<b>Identificación:</b>	R005
<b>Nombre:</b>	Cambiar estado a pedidos.
<b>Descripción:</b>	Los restaurantes tendrán la opción de cambiar el estado de un pedido el cual por defecto será Pendiente, y podrá ser cambiado a Atendido y/o Entregado.
<b>Categoría</b>	Restaurante.

<b>Identificación:</b>	R006
<b>Nombre:</b>	Perfil
<b>Descripción:</b>	Los restaurantes tendrán la opción de modificar su perfil el cual será nombre, dirección, correo, teléfono, categoría, horario, logo.
<b>Categoría</b>	Restaurante.

<b>Identificación:</b>	R007
<b>Nombre:</b>	Contraseña
<b>Descripción:</b>	Los restaurantes tendrán la opción de cambiar la contraseña para acceder al servicio de qfilo.com
<b>Categoría</b>	Restaurante.

## Tecnologías utilizadas

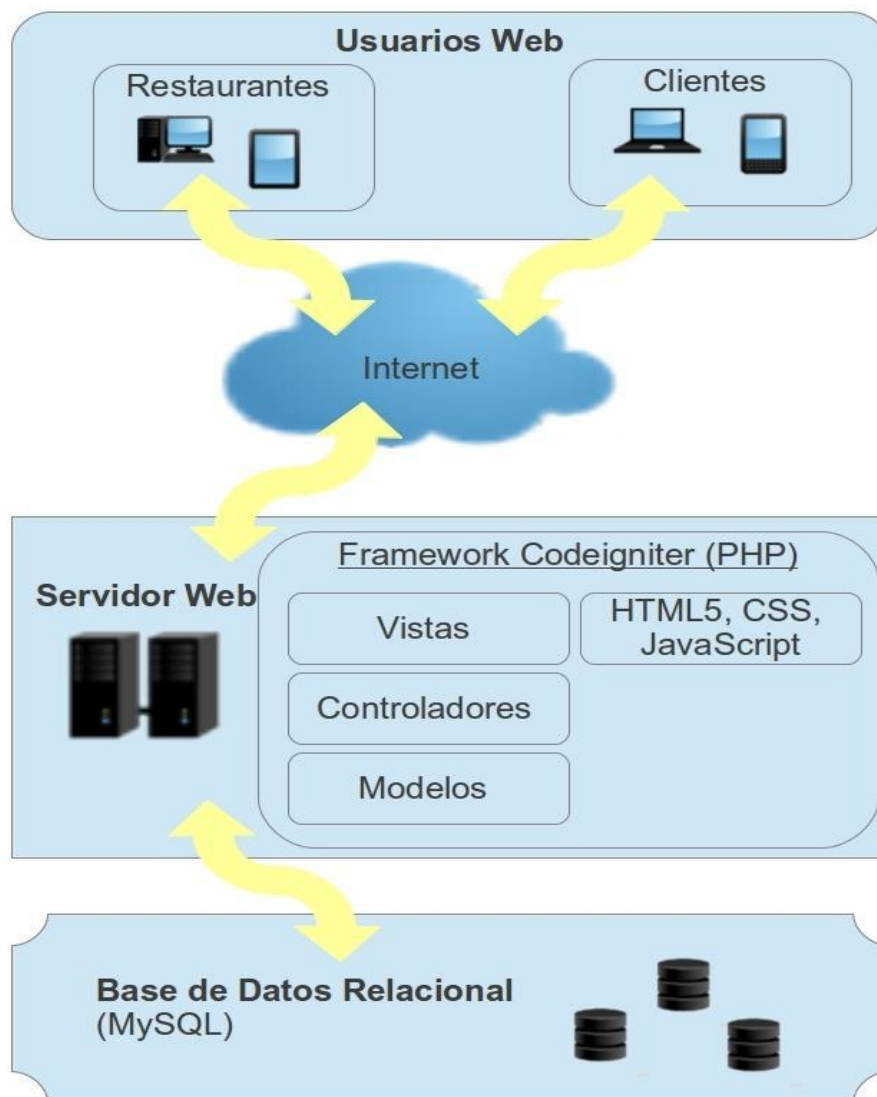
- MySQL - Base de datos a utilizar con InnoDB como gestor de almacenamiento.
- PHP - Lenguaje de programación del servidor.
- Codeigniter - Framework de desarrollo para PHP que integra el modelo MVC.
- Facebook SDK - Librería en PHP para el manejo de la sesión con Facebook.
- HTML y CSS - Lenguajes para desplegar información con estilo al usuario y restaurantes.
- JavaScript / JQuery - Lenguaje de programación para brindar contenido dinámico.
- OpenStreetMap - Librería en JavaScript para el despliegue de mapas.
- Balsamiq Mockups - Herramienta de diseño de bocetos para proyectos de software.
- MySQL Workbench - Herramienta UML utilizada para el diseño de base de datos.

# Solución implementada.

## Modelo de Diseño

### Arquitectura conceptual de la solución

En la *Figura 2*. Se muestra el diagrama de la arquitectura de qfilo.com en el cual representa a nivel general los componentes de la aplicación que serán explicados en la siguiente sección.



**Figura 2:** Diagrama de la arquitectura.

## Componentes y servicios

En esta sección se describe detalladamente todos los componentes y servicios de la aplicación tal a como lo vimos en el diagrama de la arquitectura en la sección previa.

### Servidor Web

Este proyecto utiliza la infraestructura LAMP, un conjunto de herramientas en un solo paquete, las cuales incluyen como servidor web Apache, es código abierto y está disponible para plataformas Linux, Windows, Macintosh (Dougherty, 2001).

### Base de datos

Se utiliza MySQL, porque es una base de datos libre y muy robusta, con muchas herramientas de desarrollo como PhpMyAdmin o el mismo MySQL WorkBench. Además de que viene integrada con la infraestructura LAMP.

### Lenguajes de programación:

Como lenguaje principal se usa PHP para el manejo de la lógica de la aplicación junto con CodeIgniter, un entorno de trabajo (FrameWork) que nos permite desarrollar aplicaciones php con la metodología MVC (Modelo, Vista y Controlador) el cual es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos: Modelo, Vista y Controlador respectivamente (Codeigniter, n.d.).

El proyecto está constituido por una serie de modelos para consultar e insertar informaciones de restaurantes, de sus menús y órdenes, además de los clientes como datos, sus pedidos realizados y direcciones de entrega. Mediante el framework codeigniter se utiliza un modelo de consultas a bases de datos llamado Active Record que consiste en la utilización de patrones para obtener, insertar y modificar los datos de manera tal que sean independientes de la base de datos que se esté utilizando, este sistema se encarga de generar los queries correspondientes a la base de datos en uso, además de aplicarles filtros de escape para así generar consultas más seguras (Codeigniter, n.d.).

Las vistas del proyecto son páginas web predefinidas que mediante datos recibidos o de entrada genera información funcional para los usuarios. En una página web se aplican cerca de 3 vistas generalmente, las cuales las constituyen el encabezado, el contenido y el pie de página por lo cual el más cambiante es el contenido.

Por último los controladores se encargan de conectar las vistas con los modelos, consultando información de los modelos, la procesa y la envía a las vistas para ser mostradas y viceversa, obteniendo información de las vistas administrada por los usuarios, luego procesarlos en enviarla a los modelos para ser almacenada.

Además de PHP se utiliza los siguientes lenguajes:

*HTML*: Este es el lenguaje utilizado en las vistas de la aplicación, la cual representa páginas web que serán vistas por el usuario final: el Cliente o el Restaurante. Además con html5 se utiliza componentes de geolocalización para determinar la ubicación geográfica de los usuarios de la aplicación (Berjon, R., Leithead, et al. 2012).

*CSS*: Lenguaje de programación utilizado para brindar color y diseño a las páginas web, además para que la aplicación quedara uniforme en cuanto a su estilo visual se crearon algunas plantillas (World Wide Web Consortium. n.d.)

*JavaScript*: Lenguaje incluido en la interfaz de usuario para generar páginas dinámicas (Crockford, D. 2001), la utilización de este lenguaje está complementado con la implementación de la librería JQuery que básicamente nos da grandes facilidades como programadores a la hora de utilizar metodologías Ajax, facilita la selección y búsqueda de elementos en la página web además permite generar animaciones y personalizar la interfaz con algunos “widgets” como datepicker, botones e inputs (The jQuery Project. n.d.).

En cuanto a Ajax es utilizado en casos en los que se debe cargar información solicitada al servidor y no se desea recargar la página por completo, sino sólo una parte (Garrett, J. J., 2005).

## **OpenStreetMaps**

Es otra librería JavaScript utilizada para desplegar un mapa en HTML a partir de una posición geográfica (OpenStreetMap Wiki., 2012)

El despliegue de mapas en esta aplicación se utiliza para que el usuario cliente pueda especificar gráficamente la ubicación en la que se encuentra que es donde desea que llegue su pedido, con esto el usuario restaurante puede visualizar la ubicación del cliente con más precisión.

Esta librería es fácil de utilizar, primeramente se debe importar un archivo JavaScript que es el que contiene las funciones de la librería, luego y para el caso de este proyecto lo que se utiliza de la librería son las funciones de crear el mapa, crear un punto geográfico y la de agregar un marker, además se utilizar el evento de “click” que le permite al usuario cambiar la ubicación del marker.

Se utilizó OpenStreetMaps ya que en es el más completo en cuanto a carreteras y direcciones se refiere la región, incluso por encima de Google maps y Bing maps.

## **Facebook API**

La implementación de esta librería permite el registro de clientes con su cuenta de Facebook esto que con el objetivo evitarle al usuario llenar un formulario para registrarse (Facebook., n.d.)

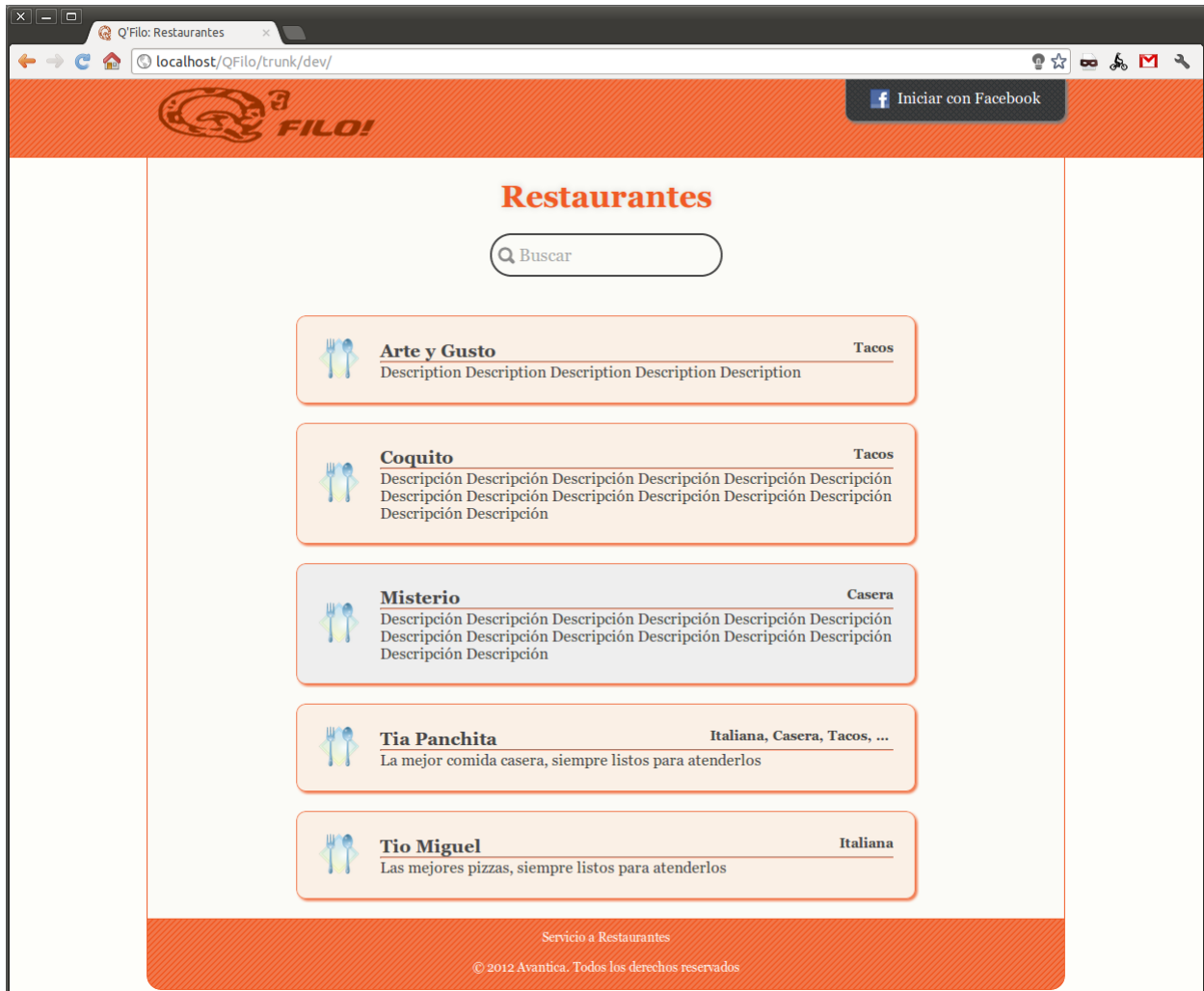
Para implementar esta librería fue necesario crear una aplicación en “facebook developers”, instalar CURL que es un complemento de apache, además de descargar la librería de PHP y agregarla al proyecto.



## Interfaces de usuario

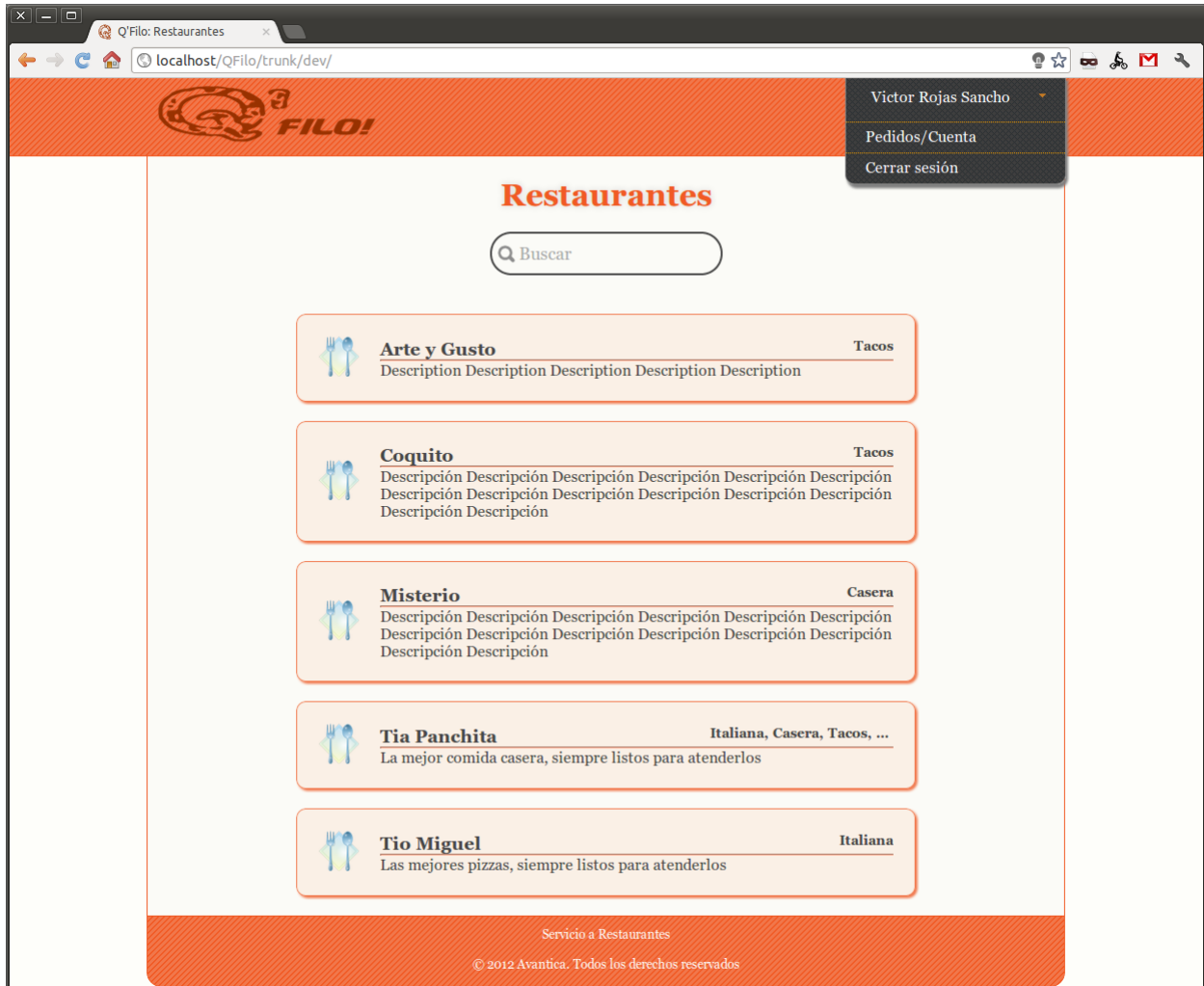
Se incluyen las capturas de pantalla tomadas a las interfaces de usuario (Clientes).

En la *Figura 3*. Se muestra la página de inicio, donde se muestra la lista de restaurantes registrados en la página, también contiene un buscador que le permite al usuario filtrar la lista de restaurantes ya sea por su nombre o por su categoría. En la parte superior se encuentra la opción de iniciar sesión con Facebook. El usuario puede hacer “click” en cualquiera de los restaurantes, esta acción lo redirigirá al menú de ese restaurante como se muestra en la *Figura 5*.



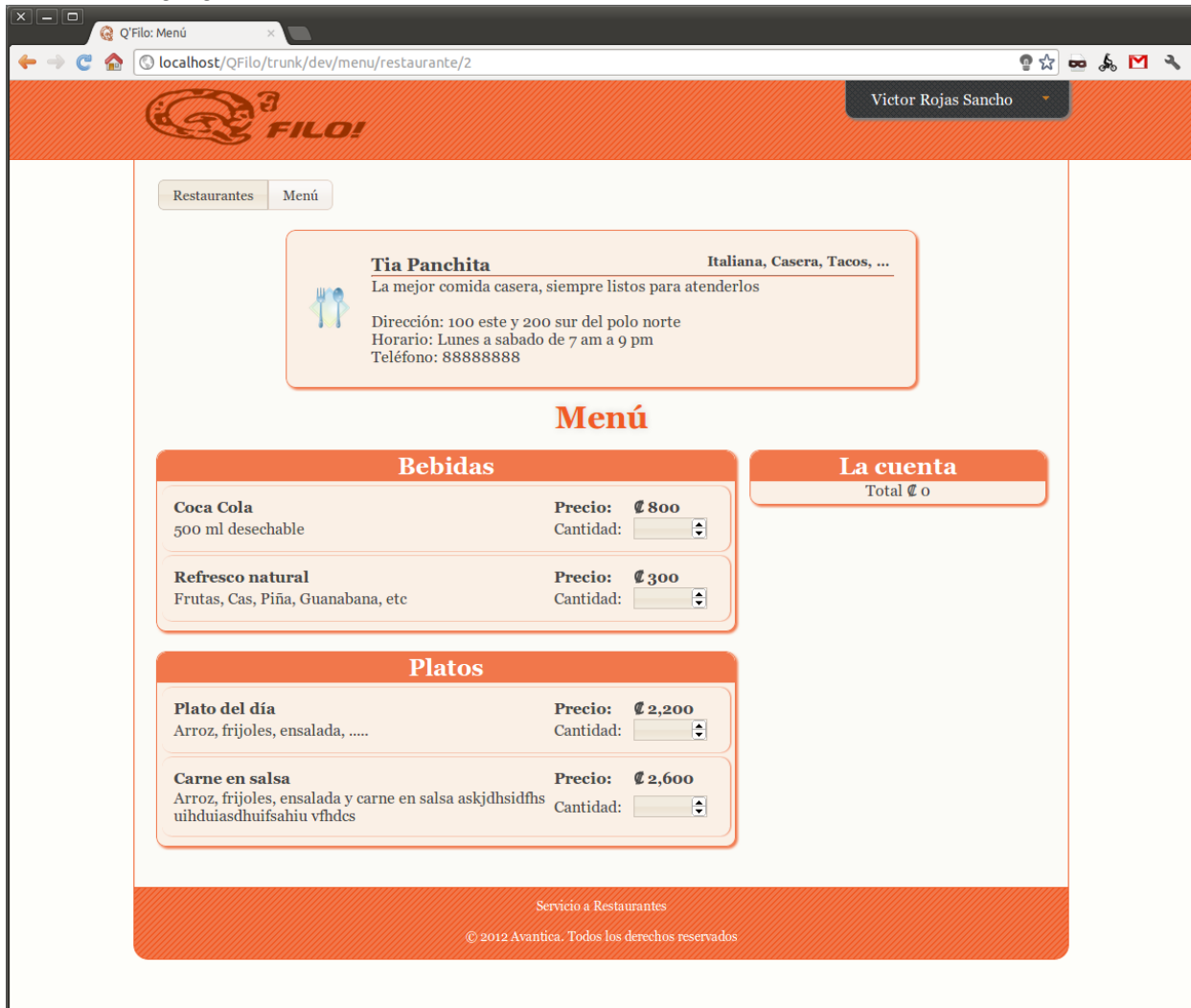
**Figura 3:** Página inicial.

En la *Figura 4*, se muestra la página de inicio una vez que el usuario ha iniciado sesión, donde lo que cambia es que en lugar de estar la opción de iniciar sesión, está el nombre del usuario, si este hace “click” sobre su nombre se desplegará dos opciones, la de ir a la cuenta y la de cerrar sesión.



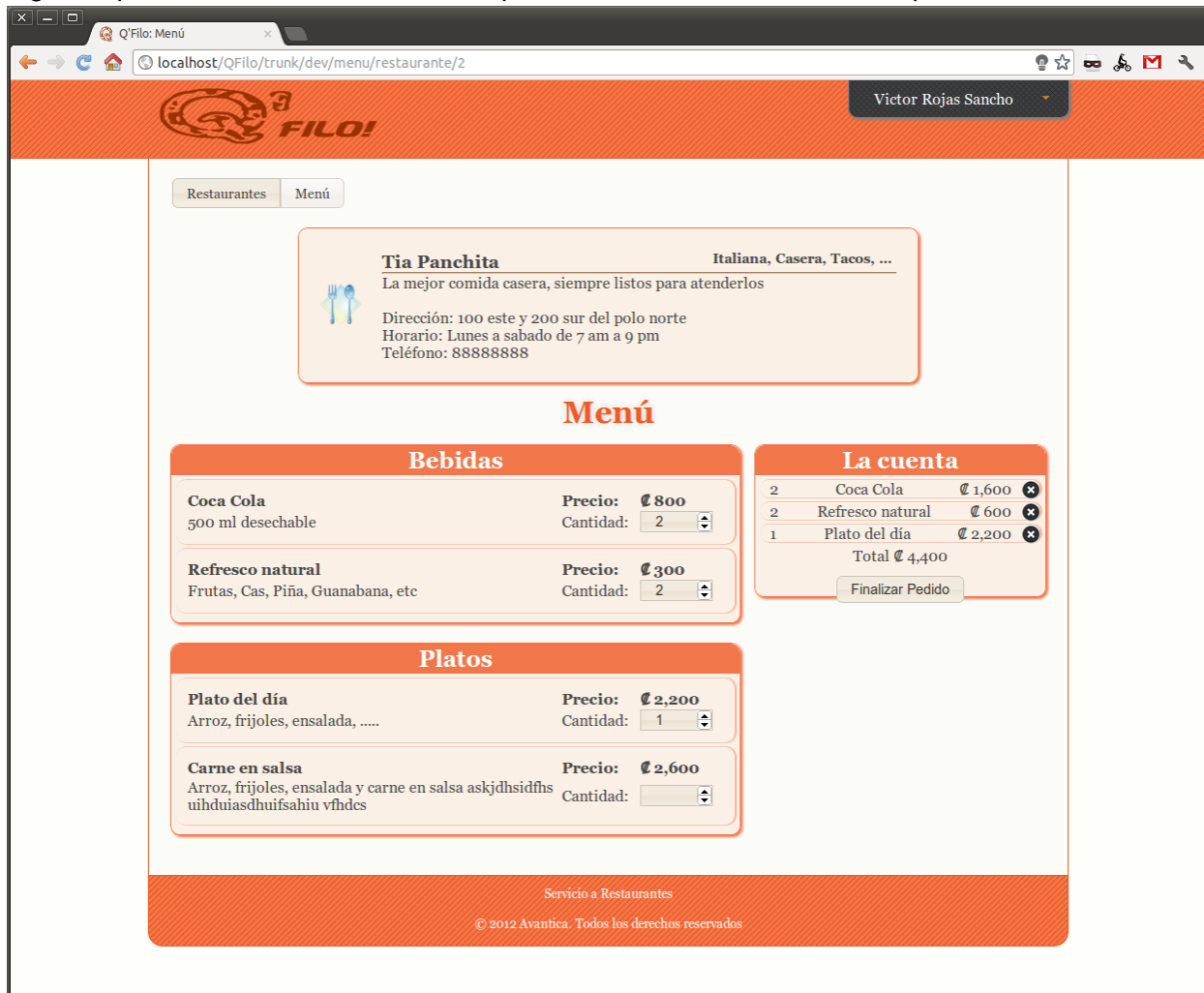
**Figura 4:** Página inicial con una sesión de usuario abierta.

En la *Figura 5*, se muestra el menú de un restaurante, en la parte superior está la información importante del restaurante, como su dirección y su horario de atención. En la parte inferior están las categorías creadas por el restaurante y los productos que cada una contiene, el usuario tiene la posibilidad de seleccionar la cantidad que desea comprar de cada pedido, y esto se irá agregando a la cuenta del usuario.



**Figura 5:** Menú de un restaurante.

En la *Figura 6*, se muestra el menú de un restaurante una vez que el usuario ha seleccionado algunos productos, el usuario tiene la posibilidad de continuar con su pedido o de cancelarlo.



**Figura 6:** Menú de un restaurante con productos seleccionados.

Una vez que el usuario ha decidido continuar con su pedido, será dirigido al paso final del pedido como se muestra en la *Figura 7*. donde se muestra la información de los productos que ha seleccionado, el usuario puede decidir si desea ir al restaurante a recoger su pedido o si desea que el pedido sea enviado por servicio express, si selecciona el servicio express se le sumará los costos del envío y el usuario debe indicar cuál es la dirección a la que debe ser enviado el pedido. Además existe una sección de comentarios, donde el usuario tiene la posibilidad de agregar algún detalle adicional a su pedido. Una vez realizado el pedido el sistema notificará al restaurante.

The screenshot shows a web browser window with the URL `localhost/QFilo/trunk/dev/pedido`. The page title is "Pedido" and the user is logged in as "Victor Rojas Sancho". The navigation menu includes "Restaurantes", "Menú", and "Pedido".

### Tu pedido en Tia Panchita

Cantidad	Producto	Precio
2	Coca Cola	₡ 1,600
2	Refresco natural	₡ 600
1	Plato del día	₡ 2,200
-	Costo del envío	₡ 1,000
Total		₡ 5,400

### Entrega

Voy al restaurante

Servicio Express

Dirección:

300m norte y 150m este del AyA en Cedral.

[Agregar dirección](#)

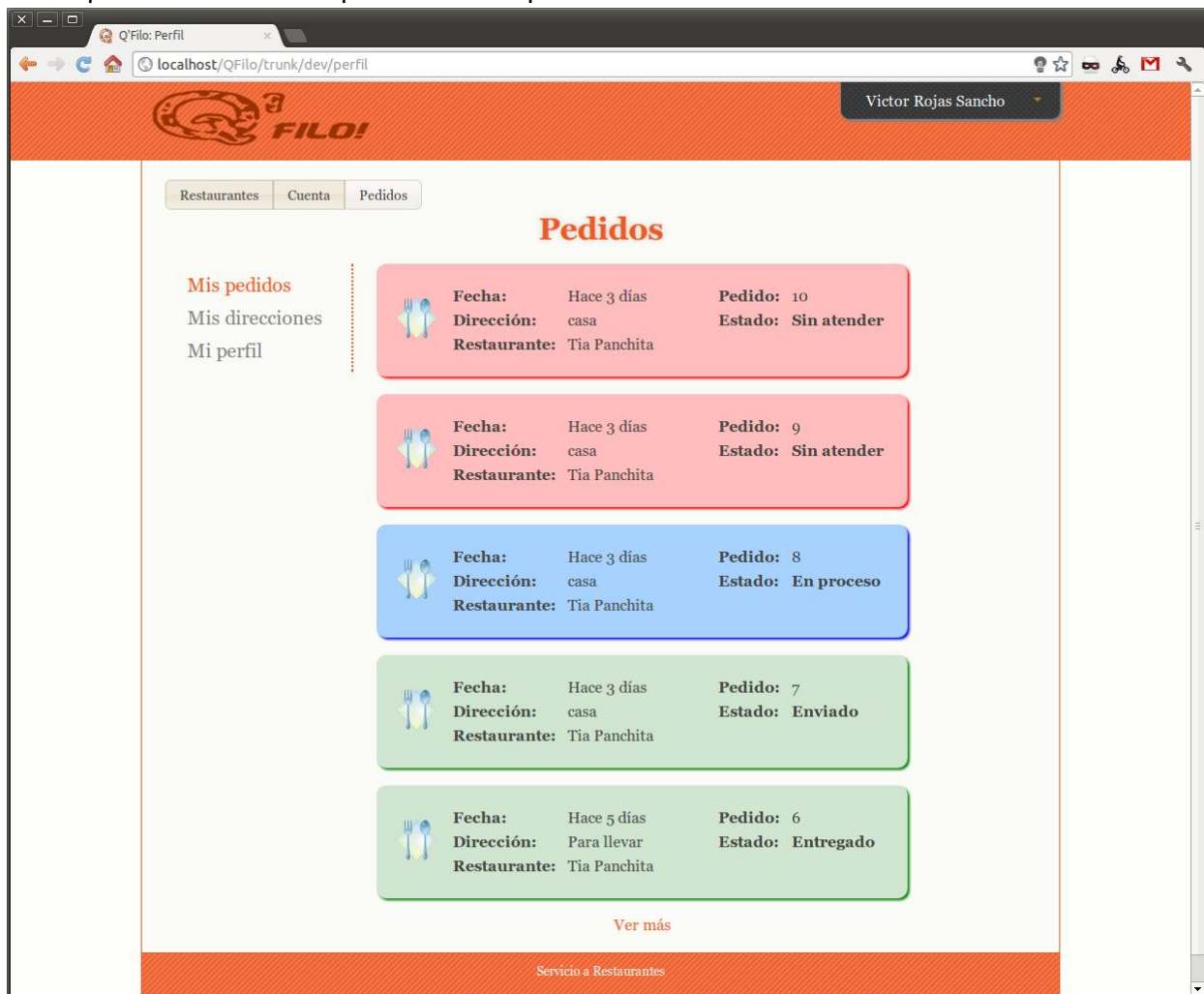
### Comentarios

e.g. Deseo la hamburguesa sin tomate.

Servicio a Restaurantes

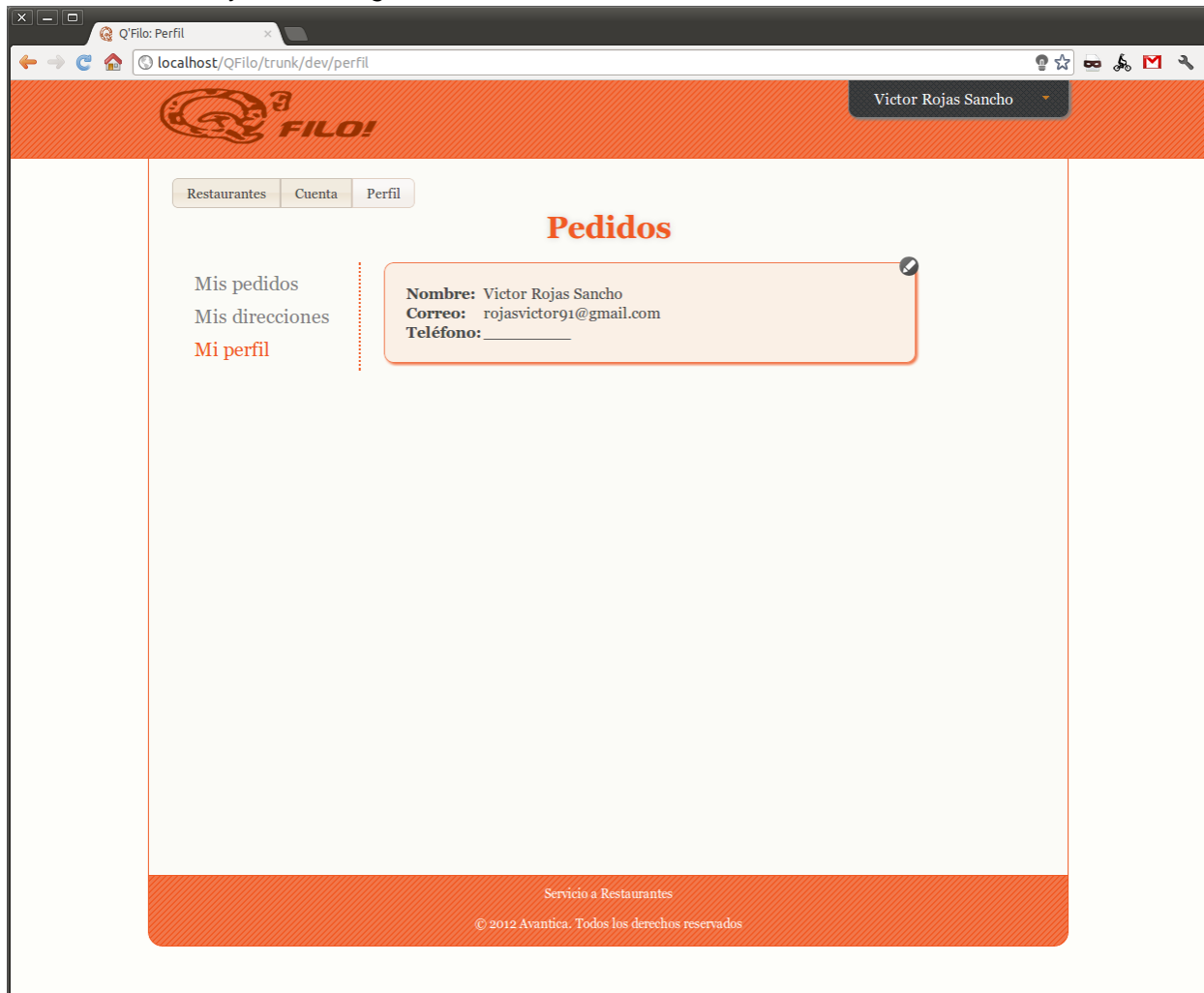
**Figura 7:** Realizar pedido.

En la *Figura 8*. se muestra la sección de pedidos del usuario, donde este podrá observar el historial de sus pedidos, del más reciente al más antiguo, cada pedido tiene información importante como el número de pedido, fecha y hora a la que fue enviado, dirección elegida, restaurante y el estado del pedido que puede ser pendiente, en proceso o entregado, de esta forma podrá monitorear el proceso de su pedido.



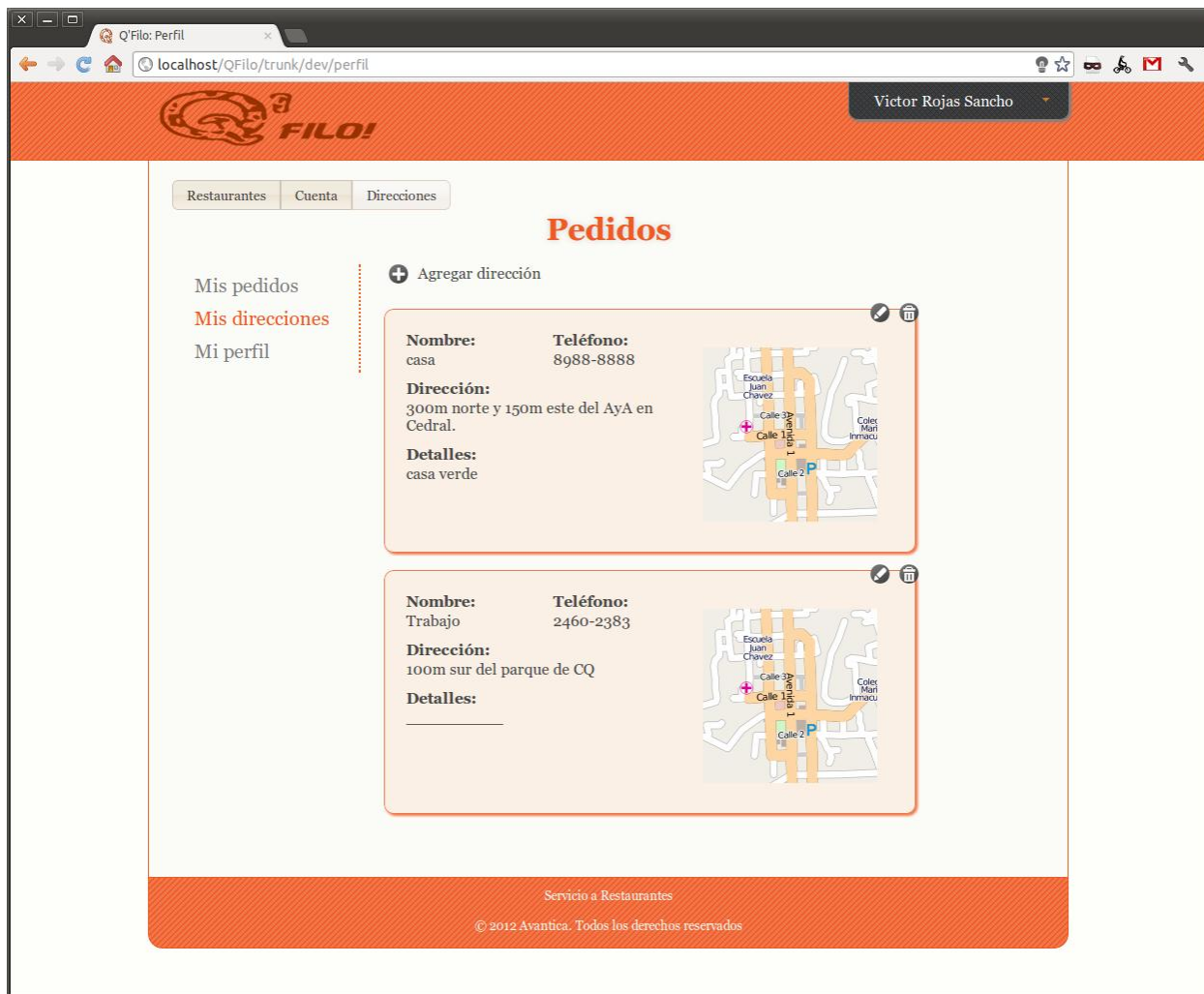
**Figura 8:** Pedidos de usuario.

También está la sección de perfil, donde el usuario podrá editar su información básica como el nombre, teléfono y correo. *Figura 9.*



**Figura 9:** Perfil del cliente.

Por último está la sección de direcciones como se muestra en la *Figura 10*. donde el usuario administra sus direcciones, puede agregar, eliminar o modificar las direcciones. La información de las direcciones es un nombre, la dirección, y de forma opcional los detalles, un teléfono y las coordenadas geográficas.

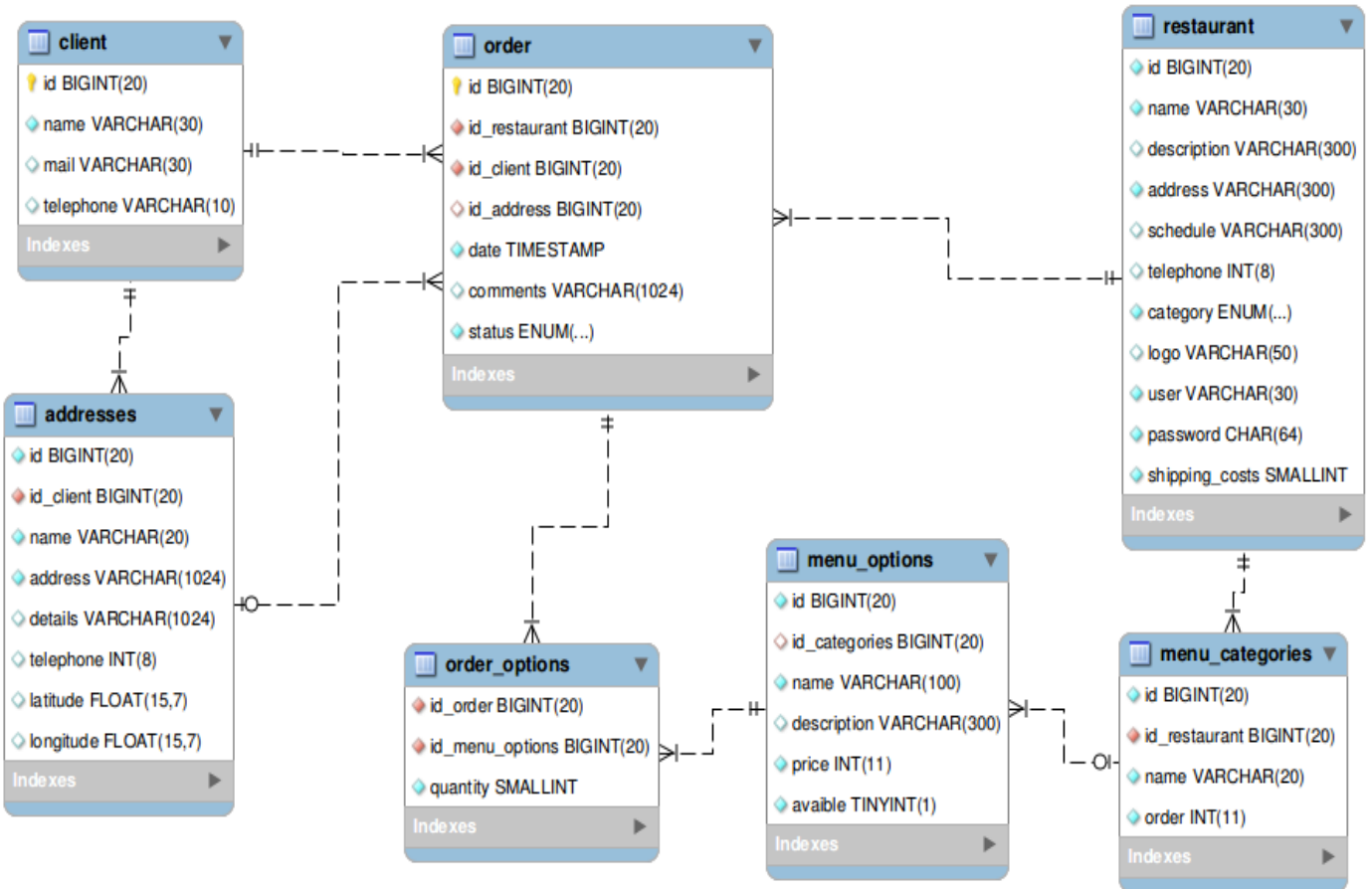


**Figura 10:** Direcciones del usuario.



## Diseño de base de datos

En la *Figura 11*. se muestra el diagrama de la base de datos, utilizando MySQL WorkBench



**Figura 11:** Diagrama de Base de Datos.

# Conclusiones y comentarios

Todos los objetivos tanto de diseño como funcionales fueron completados exitosamente y aparte de ello se pudo integrar la retroalimentación proporcionada de cada reunión.

Esta retroalimentación fue sugerida por las personas con más experiencia en la empresa permitiendo que la aplicación fuese más dinámica e intuitiva para los usuarios, esto mediante elementos de interfaz que mejora la organización de la página y representan el flujo de la aplicación de una forma clara y concisa.

## Documentos y productos entregados

Se realizaron dos documentos en el transcurso del proyecto, se fueron actualizando conforme avanzaron las etapas de desarrollo. Los documentos son el StoryBoard y el Plan de Trabajo. Documentos suficientes para la realización de nuestro proyecto. En ellos se incluyen los bocetos, funcionalidades flujo y eventos del sistema además de la repartición de tareas en el tiempo y por persona o estudiante.

Ambos documentos fueron completados con éxito, y nos enseña una nueva forma de realizar proyectos de software mediante el uso de mockups o bocetos.

A continuación en la *tabla 1* se muestran los productos entregados con su nivel de porcentaje y comentarios:

<b>Módulo</b>	<b>Estado</b>	<b>Observaciones y mejoras</b>
Inicio de sesión con Facebook	Completado 100%	
Direcciones	Completado 100%	
Pedidos	Completado 100%	
Perfil	Completado 100%	

**Tabla 1:** Módulos de productos entregados

## **Experiencias adquiridas**

El ambiente laboral es muy bueno, la mayoría de las personas son egresadas del TEC por lo que la transición y el ajuste no fueron tan difíciles ni tan lentos, ya que la mayoría son personas que uno conoce a través de los años de estudio.

El ajustarse a un horario es un aspecto importante ya que las tareas asignadas tienen que ser realizadas dentro del tiempo establecido y no se pueden dejar para después, como podía hacerse mientras se estudia.

La relación jerárquica no es tan vertical, por lo que facilita que los colaboradores puedan dar sus ideas y opiniones, lo que hace que los colaboradores puedan expresar sus molestias y si es posible encontrarle solución de manera expedita.

Una ventaja es la oportunidad de trabajar en una empresa local, que realiza proyectos tanto para empresas nacionales como internacionales con altos estándares de calidad lo cual nos permite desarrollarnos profesionalmente.

## **Oportunidades de mejoras**

La finalidad de este proyecto es ser funcional lo más antes posible, tener un producto en ambiente de producción, ósea utilizado por los clientes finales, por lo cual no se entró a muchos detalles y características que lo convirtiera en un proyecto de largo alcance, por eso se generó un producto simple, que cumpliera con lo esencial. Y por ello fue diseñado de tal manera que se pudiera agregar más características en cada menú de opciones, editar el flujo de la aplicación o agregar nuevas características que lo convierten en una mejor herramienta por eso a continuación se mencionan una lista de mejoras, y nuevas características que considero sería bueno agregar a futuro:

### **Editor de pedidos:**

Como mejora o nueva característica consiste en permitir al cliente editar sus pedidos enviados desde la posibilidad de cancelarlos completamente hasta agregar o eliminar nuevos productos y editar comentarios siempre y cuando el pedido tenga un estado de pendiente, porque sería un problema si ya se estuviera procesando. Por lo cual como sugerencia debería de bloquearse el pedido si el usuario lo está editando para que así es restaurante se percate de dicho evento y no lo procese hasta que haya sido modificado o pues eliminado.

**Búsquedas de pedidos:**

Una nueva característica más útil para los restaurantes que a los clientes es la de búsquedas con ciertos criterios de los pedidos u órdenes, lo cual le permite observar la cantidad de órdenes generadas en un cierto periodo de tiempo, con toda la información correspondientes, como la cantidad de ventas por producto, lo más solicitado y así. De la misma manera se pueden hacer búsquedas de órdenes por producto para determinar su promedio de ventas e inclusive para determinar qué posiciones geográficas es donde se concentran los clientes para así hacer campañas en lugares de poco consumo y utilización de la aplicación.

**Sistemas de pago:**

Otra nueva característica para agregar al proyecto es la de incluir sistemas de pago lo cual brinda facilidad de los cobros de los pedidos. Un aspecto importante a sugerir es que se efectuará el cobro una vez que el pedido entre en estado de procesado o entregado dado a que se puede que el cliente edite o elimine su orden.

Entre los sistemas de pago podríamos incluir desde lo más conocido como Paypal, tarjetas de crédito y débito hasta utilizar nuevos sistemas como Titicupon, o cobros por sistemas de telefonía celular prepago o postpago.

**Especificar hora de entrega:**

Otra característica es especificar la hora en que desea el cliente recibir sus pedidos, siempre y cuando se encuentre en el horario de atención del restaurante. Esto con el fin de facilitarle a los clientes organizarse con sus tiempos y quehaceres. (e.g) un cliente puede que solo tenga acceso a internet desde su trabajo y en su hogar no disponga por lo tanto podría generar el pedido desde su lugar de trabajo.

Cómo manejar esto en el lado del restaurante e informar al cliente de su pedido?

Hay varias soluciones que se pueden implementar, desde agregar un nuevo estado como "escuchado" lo cual significa que el restaurante observó el pedido y lo procesará cuando deba y así poder informar al cliente que han observado su pedido. Otra forma es mostrar en el tablero de los pedidos del restaurante solo los que tengan la hora de entrega próxima a una hora por ejemplo, los demás pedidos estarían ocultos esperando llegar a esa hora para ser mostrado.

Otra solución es la combinación de ambos.

**Reservaciones:**

Consiste en ampliar el sistema, permitir que no solo para sea express o para llevar sino también para ir a comer en el restaurante algo como una reservación. El cual el cliente ya tendría su orden y agregaría su hora de llegada y se le cobraría cuando se entrega la orden o se procesa, por ahora no hay nada nuevo, el problema radica en el manejo de espacios del restaurante por lo cual se debe agregar características como número de personas que llegarán y un manejo de los espacios del restaurante.

### **Horario de atención:**

Una funcionalidad de gran importancia para la mejora de la aplicación es la de horario de atención, donde el restaurante pueda manejar más detalladamente su horario de atención, incluyendo fines de semana y días festivos, de esta manera cuando un usuario desea realizar un pedido al restaurante, el sistema le indicará si el restaurante se encuentra disponible para atender su pedido o no. Esta funcionalidad evitará que se realicen pedidos fuera del horario de atención, ya que en la actualidad el usuario debe verificar por cuenta propia si el restaurante se encuentra en el horario de atención.

Las anteriores fueron una lista de sugerencias a implementar a un futuro, pensadas por nuestra creatividad y experiencia sin embargo estamos dispuestos a escuchar nuevas características por parte de los clientes y restaurantes para implementarlas en el proyecto.

### **Crecimiento de usuarios**

Actualmente el proyecto se limita a un área de Ciudad Quesada y como objetivo se tiene llegar a cubrir toda la Zona Norte y posteriormente llegar a cubrir nuestro país. Con ello se generarían más restaurantes y más usuarios utilizando la aplicación y llegamos al problema de la escalabilidad que consiste en tener la capacidad de administrar y trabajar con cientos de usuarios conectados y utilizando la aplicación sin presentar problemas como lentitud y consistencia de datos por ejemplo esperar mucho para ver el menú de un restaurante. Pero para ello existen soluciones a implementar al proyecto, por eso a continuación se presentan diferentes los tipos de escalabilidad y cómo nos benefician en el proyecto, además se explica qué cambios se necesitan aplicar en el proyecto actual para conseguir dichos beneficios.

**La escalabilidad** es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. (Marco de Desarrollo de la Junta de Andalucía, s. f.)

#### **Escalabilidad Vertical**

El escalar verticalmente o escalar hacia arriba, significa el añadir más recursos a un solo nodo en particular dentro de un sistema, tal como el añadir memoria o un disco duro más rápido a una computadora. (Marco de Desarrollo de la Junta de Andalucía, s. f.)

El costo sobre el proyecto sería mínimo, y nulo para el código de la aplicación, las nuevos dispositivos de mayor capacidad implican mayor inversión en dinero y conforme vaya aumentando el nivel de usuarios llega el punto que se necesita nuevo hardware y por ende nuevos costos económicos. El rendimiento del hardware llega a un punto límite de capacidad de procesamiento siendo esta la principal limitante de la escalabilidad vertical

Si se adquiere un nuevo servidor de mayor capacidad el proyecto requiere de una exportación completa de la base de datos aparte de la aplicación.

## **Escalabilidad horizontal**

La escalabilidad horizontal, significa agregar más nodos a un sistema, tal como añadir una computadora nueva a un programa de aplicación para espejo. (Marco de Desarrollo de la Junta de Andalucía, s. f.)

Con este tipo de escalabilidad no existe el limitante de procesamiento que se da en la escalabilidad vertical ya que se pueden agregar más nodos al sistema.

El problema que surge con este tipo de escalabilidad es que el proyecto requiere de un nuevo diseño que implica cambios tanto a nivel de código como estructura de la base de datos.

Cambios a nivel de código:

El principal cambio es el desarrollo de un distribuidor de cargas que consiste en una aplicación que recibe las peticiones http y las distribuye entre los diferentes nodos.

Además puede implicar la descomposición de la aplicación por módulos específicos que serán incluidos en nodos por aparte con el fin de recibir solamente un tipo de petición.

Cambios a nivel de base de datos:

Los cambios surgen en tener una base de datos distribuida o en espejos para de igual forma balancear las peticiones y consultas. Generalmente se utiliza una configuración en el sistema gestor de base de datos para obtener este tipo de base de datos.

# Referencias

Berjon, R., Leithead, T., Navara, E. D., O'Connor, E., & Pfeiffer, S. (11 October 2012). *HTML5: A vocabulary and associated APIs for HTML and XHTML*. Retrieved Octubre 23, 2012, from <http://dev.w3.org/html5/spec/single-page.html>

Codeigniter. (n.d.). *User Guide*. Retrieved Octubre 23, 2012, from [http://codeigniter.com/user\\_guide/index.html](http://codeigniter.com/user_guide/index.html)

Crockford, D. (2001). *JavaScript: The World's Most Misunderstood Programming Language*. Retrieved Octubre 23, 2012, from <http://www.crockford.com/javascript/javascript.html>

Dougherty, D. (2001, Enero 26). *LAMP: The Open Source Web Platform*. Retrieved Octubre 23, 2012, from <http://onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>

Facebook. (n.d.). *Developers Facebook*. Retrieved Octubre 23, 2012, from <https://developers.facebook.com/>

Garrett, J. J. (February 18, 2005). *Ajax: A New Approach to Web Applications*. Retrieved Octubre 23, 2012, from <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>

Marco de Desarrollo de la Junta de Andalucía (s. f.). *Conceptos sobre la escalabilidad*. Recuperado de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/220>

OpenStreetMap Wiki. (October 5, 2012). *OpenLayers*, . Retrieved October 23, 2012 from <http://wiki.openstreetmap.org/w/index.php?title=OpenLayers&oldid=817035>

The jQuery Project. (n.d.). *jQuery: The write less, do more, JavaScript library*. Retrieved Octubre 23, 2012, from <http://jquery.com/>