Indice

Introducción

El proyecto que se describe en este documento se realizó con el fin de cumplir con el requerimiento de prácticas profesionales, para la carrera Ingeniería en Sistemas de Información.

Aquí se detalla el proyecto llamado “Proyecto Introductorio: mobile.quepasa.com”, de la empresa Quepasa.com de México, que tiene como finalidad el desarrollar los conocimeintos tanto teóricos como técnicos de alumnos próximos a egresar de carreras referentes a las Tecnologías de la Información.

En el presente documento se muestra el desarrollo de una aplicación móvil para el sitio de internet “quepasa.com”, red social enfocada a el público latino en todo el mundo.

Este proyecto abarcó todos los aspectos que se toman en cuenta en la página oficial del sitio, incluyendo el análisis del problema, el desarrollo de la base de datos, programación de manejadores de información, y la presentación de los datos al usuario final.

El tiempo proyectado para la realización de la aplicación fue de tres meses, incluyéndose además del desarrollo del proyecto, capacitaciones impartidas por personas especializadas en cada área de la empresa.

Descripción del área donde se llevó a cabo la práctica

El proyecto no fue desarrollado en un área específica de la empresa, ya que fue una aplicación independiente del sitio oficial. Por esta razón, los asistentes al curso participamos en todas las áreas del desarrollo, con la asesoría de especialistas en éstas.

El desarrollo se llevó a cabo por etapas, con tiempo tentativamente definido, comenzando con pláticas de capacitación en las distintas áreas y tecnología utilizadas por la empresa, impartidas por personas que laboran en éstas áreas.

Después se trabajó en cada etapa del desarrollo, basándose en la teoría adquirida durante la primera etapa del proyecto.

Justificación

Se desarrolló este proyecto debido al constante crecimiento de la tecnología móvil, celulares y teléfonos inteligentes con acceso a internet y una gran cantidad de aplicaciones, por ello la empresa Quepasa.com considera aprovechar estos cambios en la tecnología para su beneficio.

Una aplicación de este tipo permitiría a la empresa satisfacer a sus usuarios ofreciéndoles nuevas alternativas para utilizar los servicios a los que tienen acceso en el sitio, lo cual representa un beneficio tanto para los usuarios como para la empresa, que se vería beneficiada al representarles esto una nueva oportunidad de obtener mayor presencia en el mercado y mantenerse en la preferencia de los usuarios.

Objetivos

Objetivo general

Desarrollar una aplicación móvil sencilla, que pueda ser desarrollada por personas con poca experiencia en el desarrollo de aplicaciones de software.

Objetivos específicos

- Desarrollar una aplicación que sea amigable para el usuario.
- Desarrollar una aplicación que sea segura y confiable para el usuario.
- Satisfacer las necesidades de los usuarios de dispositivos móviles.
- Tener más presencia en el mercado.
- Desarrollar una aplicación que sea adaptable a cambios en el futuro, como agregar nuevas funcionalidades.

Problemas planteados para resolverlos

Actualmente el sitio Quepasa.com no cuenta con una aplicación de software que pueda ser utilizada en dispositivos móviles. Al no tener una aplicación de este tipo, la empresa pierde usuarios, puesto que no les ofrece alternativas distintas de acceder al sitio y utilizar sus servicios.

Alcances y limitaciones de la solución

Esta aplicación será desarrollada por personas que no tienen experiencia en el desarrollo de software, lo cual implica que sea sencilla y no cuente con gran parte de los servicios que el sitio ofrece.

Las tecnologías a utilizar no requieren del pago de licencia, por lo que el acceso a éstas no representa una limitante para el desarrollo de la aplicación.

Fundamento teórico

Se explican a continuación los conceptos fundamentales, y las tecnologías, con su descripción, utilizadas para cada uno de los módulos del proyecto.

Base de datos

La mayoría de las aplicaciones web, sobre todo una como la aplicación en cuestión, tienen como requerimiento imprescindible contar con una base de datos, por el simple hecho de tener la necesidad de almacenar información, sin importar la cantidad de ésta.

Para este proyecto se tomó como base el gestor MySQL, en el que fue creada la base de datos principal para el proyecto.

Se aplicaron conceptos tales como bases de datos relacionales, utilización de índices para acelerar las búsquedas en las tablas, así como también utilización de técnicas para encriptación de claves de acceso.

Todas las tablas y sus relaciones fueron creadas de acuerdo a las necesidades especificadas en la etapa de análisis de requerimientos, después se fueron modificando algunas de ellas por ciertas cuestiones que surgieron durante el desarrollo, así como también en algunos casos fue necesario crear nuevas tablas, por nuevas necesidades de la aplicación.

“Back-end”

Es la parte del software que procesa las peticiones hechas por el “front-end” y que interactúa con la base de datos, realizando las consultas o modificaciones según sea el caso. Cada uno de estos posibles casos se denomina “end-point”, un punto final o destino del que se obtendrá un resultado [2].

Esta parte de las aplicaciones web se denomina “API”, “application programming interface”, por sus siglas en inglés (interfaz de programación de aplicaciones), la cual, dependiendo del “endpoint” requerido, se encarga de procesar los datos de entrada, en caso de ser necesarios, realizar las operaciones necesarias con estos datos y regresar un resultado a quien realizó la petición [4].

En el caso en cuestión, este módulo fue denominado api.quepasa.com, que es una aplicación basada en PHP, MySQL, y la metodología “MVC”, modeo-vista-controlador, utilizando “Zend Framework” para tal fin.

PHP fue utilizado para la creación de los manejadores de la información, el llamado a la base de datos, el procesamiento de estos datos, y la respuesta enviada al “front-end”. En cuanto a MySQL, se crearon dentro de la base de datos “stored procedures” (procedimientos almacenados), que son archivos almacenados en la base de datos con instrucciones definidas, con el fin de minimizar la elaboración de consultas.

El modelo MVC se aplica en este módulo del proyecto:
Controlador.- Estos se encargan de recibir las peticiones hechas por el “front-end”, para después llamar al manejador indicado enviándole los parámetros requeridos según la petición. Al recibir estos la respuesta de los manejadores, envían esta respuesta a la vista, que se encargará de darle el formato necesario a los datos. Una vez recibidos los datos en su respectivo formato, son enviados como la respuesta, en este caso, al “front-end”.
Modelo.- Se encarga de la manipulación y formato de los datos, y de las consultas tanto de lectura como escritura con la base de datos. Para esto, hace llamados a los procedimientos almacenados de la base de datos, enviándoles los parámetros necesarios según la petición hecha, y al recibir la respuesta, se encarga de organizar estos datos para después enviralos de vuelta al controlador.

Vista.- Estos son requeridos por los controladores después que la petición ha sido procesada por los manejadores; las vistas tienen como finalidad darle el formato necesario a los datos para que sean enviados a quien hace la petición. En el caso en cuestión, los formatos utilizados fueron XML y JSON, siendo este último el utilizado por el “front-end” para mostrarlos al usuario [1].

En la figura 1 se muestra el funcionamiento de la metodología MVC:

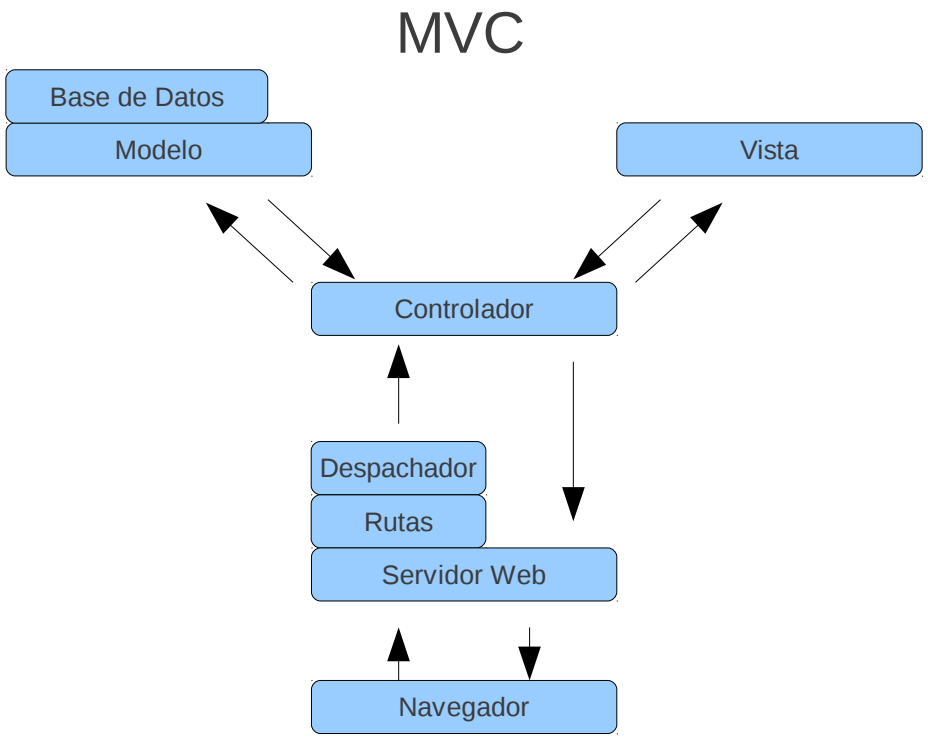

Figura 1. Diagrama de la metodología MVC.

“Front-end”

Su función es la presentación de los datos al usuario de una manera amigable, entendible, sencilla y manipulable, haciendo uso de tecnologías tales como PHP, HTML y Javascript (jQuery, Ajax, JSON), y librerías especiales de PHP, en este caso Smarty y SmartyGettext [2].

Para el proyecto en cuestión, este módulo fue llamado mobile.quepasa.com.

Se utiliza PHP para el llamado a las diferentes plantillas de que se compone el sitio; estas plantillas son el código HTML que será interpretado por el navegador, y Javascript es utilizado para la construcción y funcionalidad de las páginas que el navegador interpretará, a partir de las plantillas antes definidas.

Con la utilización de jQuery, librería de Javascript, es posible manipular estructuras HTML muy eficientemente y con gran variedad de opciones, tanto para funcionamiento interno como para la vista mostrada al usuario, permitiendo elaborar una interfaz atractiva para el usuario.

La interacción entre el “front-end” y el “back-end” se lleva a cabo por medio de peticiones AJAX, “Asynchronous Javascript And Xml”, por sus siglas en inglés (Javascript asíncrono y Xml), la cual es una técnica de desarrollo web para la creación de aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, mientras se mantiene una comunicación asíncrona con el servidor en segundo plano, lo que permite interactuar con la aplicación sin necesidad de recargar la página [5].

En este proyecto, se utilizó la técnica AJAX, sólo que en vez de recibir respuestas en formato XML, se utilizó JSON como el formato por defecto para recibir estas respuestas. JSON es un formato para representar datos, de manera similar a la que se representan en XML, pero JSON es más ligero y sencillo de interpretar para Javascript.

En cuanto a la metodología de trabajo, en la empresa se trabaja por medio de la asignación de tareas a los participantes en el desarrollo de un proyecto. Es aquí donde se aplica la tecnología SVN, o subversion, en la que un proyecto se encuentra almacenado en un servidor central, y cada participante descarga el proyecto completo en su computadora. De esta manera, se cuenta con una

copia del proyecto, y en ésta se comenzará a trabajar. Al finalizar las tareas asignadas, éstas serán cargadas al servidor principal, que, utilizando la tecnología SVN, se encargará de adherir los nuevos cambios al proyecto principal, teniendo así una nueva versión del proyecto. Esto permite tener un mayor control sobre los cambios efectuados, y además permite el trabajo en sincronía sobre diferentes partes del mismo proyecto [6].

SVN ofrece diferentes funcionalidades para el control de versiones de proyecto, tales como la verificación con versiones anteriores, diferencias entre el proyecto local de cada usuario con el proyecto que se encuentra en el servidor central, carga y actualización de archivos, entre otras [6].

En resumen, el usuario interactúa directamente con el “front-end”, el cual, dependiendo de las peticiones del usuario, utilizando AJAX hace el llamado al “endpoint”, o destino requerido del “back-end” (API); aquí, se hace la consulta a la base de datos, ya sea para lectura o escritura, se obtienen los datos requeridos y se procesa esta información para ser devuelta al “front-end” en formato JSON, y poder manipular esta representación de datos para mostrarla al usuario de manera amigable y entendible.

Desarrollo del proyecto

La solución propuesta fue el desarrollo de dos módulos distintos, pero que trabajen en conjunto para presentar la información al usuario y realizar las distintas peticiones hechas por éste.

Estas dos aplicaciones son llamadas “back-end” y “front-end”, explicadas posteriormente. Con la interacción de estas aplicaciones, se intenta obtener la más rápida, eficaz y segura experiencia para el usuario al navegar por el sitio en cuestión.

Las tecnologías utilizadas para el desarrollo del proyecto son:

- Linux
- Apache
- MySQL
- PHP
- Zend Framework
- Javascript (jQuery, Ajax, JSON)

La finalidad y función dentro del proyecto de cada una de estas tecnologías se explicará más adelante.

Los requerimientos de la aplicación se definen a continuación:

- El usuario debe poder accesar a su cuenta en quepasa.com,
- ver su perfil,
- cambiar su estado,
- ver las actividades de sus amigos y
- comentar en actividades. Esta función será desarrollada en caso de que los objetivos anteriores sean terminados antes de el límite de tiempo definido para el desarrollo del proyecto.

El proyecto se realizó en las siguientes etapas:

1. Etapa de capacitación
2. Análisis de requerimientos
3. Estructuración de base de datos
4. Implementación de base de datos
5. Programación de clases, manejadores, entidades
6. Programación de vistas
7. Integración de vistas y peticiones con Javascript

Durante todo el desarrollo del proyecto se contó con el asesoramiento de especialistas en las diferentes áreas de la empresa, apoyándonos en cuestiones de programación, estructuración de la base de datos, métodos de consulta y diversas funciones aplicadas al proyecto, como la encriptación de claves.

A continuación se explican cada una de las etapas del proyecto:

1.- *Etapa de capacitación*

El inicio del curso fue con pláticas de capacitación en las diversas tecnologías utilizadas por la empresa para el desarrollo y mantenimiento del sitio quepasa.com, y que serían las mismas que nosotros utilizaríamos para el desarrollo de nuestro proyecto.

Después de la plática introductoria a cada tecnología realizamos las instalaciones y configuraciones necesarias para cada una de éstas en la computadora que se nos fue asignada. El objetivo era familiarizarnos con las funciones, comandos, capacidades, entre otras características de cada una de éstas tecnologías, para así poder empezar con el desarrollo de nuestra aplicación.

Simultáneamente con estas pláticas trabajamos en un proyecto de prueba, en el

que vimos algunas de las funciones que utilizaríamos en el desarrollo del proyecto principal y cómo aplicar lo visto en los cursos, debido a que esta prueba fue diseñada con una estructura muy parecida a la del proyecto que nosotros realizaríamos.

2.- *Análisis de requerimientos*

Al terminar con la etapa de capacitación, como en el desarrollo de todo proyecto, se realizó un análisis de requerimientos, en el cual, se definieron las funcionalidades y capacidades del proyecto en cuestión, qué se requiere para cumplir con estas funcionalidades correcta y eficazmente, y cual sería la función de cada una de nuestras tecnologías en el proceso de desarrollo.

De esta manera, se definieron cuáles serían las entidades a representar, que en este caso, se tomó en cuenta el usuario que utiliza la aplicación, y cómo interactúa este con el sistema, además de otras entidades que sería necesario representar, tales como las actividades generadas por un usuario, los comentarios que los usuarios realizan sobre actividades, y también las relaciones entre estas entidades.

Después del análisis general, se continuó con el análisis específico a la base de datos. Para ello, tomando como base lo obtenido del análisis general, se definieron las tablas que integrarían la base de datos, y como éstas estarían estructuradas, con el fin de obtener la mejor representación de las entidades anteriormente mencionadas.

La estructura de cada tabla fue definida de acuerdo a las necesidades observadas. En las tablas donde fue necesario, se aplicaron técnicas como la utilización de llaves primarias, principalmente en las tablas que representaban a las entidades, llaves foráneas, para definir la relaciones necesarias entre las tablas que lo requerían, y además de utilizar una propiedad para el acelerameinto de búsquedas, llamado índices, el cual toma como referencia un

campo definido sobre el cual empezar la búsqueda, para agilizar las consultas a la base de datos en el caso de que se cuente con grandes cantidades de datos, como en el proyecto en cuestión, el cual está pensado para almacenar usuarios, actividades, comentarios, entre otro tipo de información necesaria.

Algunos ejemplos de estas tablas se muestran en la figura 2.

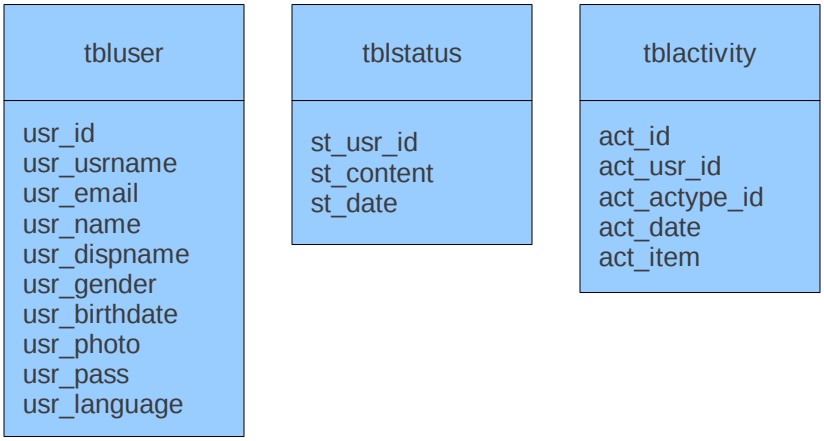

Figura 2. Ejemplos de algunas de las tablas creadas.

Al quedar definida la base de datos, se analizó la manera en que se consultarían los datos almacenados. En este proceso se contó aún más con el apoyo de los asesores, ya que se tenía pensado utilizar tecnologías y técnicas con las que nosotros como practicantes no estábamos familiarizados.

Como se mencionó antes, la consulta a la base de datos se realizó mediante una aplicación denominada API, la cual, al recibir los parámetros necesarios vía URL, verificaría que los datos recibidos sean los correctos para tal petición, realizaría la petición a la base de datos, ya sea de lectura o escritura, y tomaría la respuesta de la base de datos para darle el formato necesario según el tipo de petición, que, en el caso de la aplicación en cuestión, fueron respuestas en formato JSON, utilizando la técnica AJAX.

3.- *Estructuración de base de datos*

Esta etapa comprende la creación de la base de datos principal utilizada por el sistema, de acuerdo a lo definido en el análisis de requerimientos.

Durante la realización de este proceso, aparecieron nuevos cuestionamientos sobre la funcionalidad y necesidades de la base de datos para cumplir correctamente con sus objetivos, por lo que fue necesario realizar ciertos cambios en la estructura de la base de datos.

Además de las tablas que representaban a las entidades principales, usuarios, actividades y comentarios, fue necesario la creación de tablas de apoyo, como la tabla preferencias de usuario, que era un complemento de la tabla usuario.

La tabla usuario almacena los datos principales de cada usuario, para almacenar las preferencias del usuario en cuanto a permisos, se creó esta nueva tabla. Otra tabla complementaria fue la tabla de amigos, en la cual se almacenan todas las relaciones de cada usuario con sus respectivos amigos.

Para la tabla actividades, se crearon complementos tales como la tabla tipo de actividad, de la cual depende la tabla actividades, ya que de esta toma la información del tipo de actividad.

Se creó también la tabla contenido de actividad, ya que la tabla actividad sólo almacena los identificadores y fecha de creación de actividad. El contenido de la actividad es consultado en otra tabla. El mismo procedimiento fue aplicado para la tabla comentarios.

Una tabla esencial fue la tabla de usuarios que se encuentran en sesión, utilizada para almacenar a los usuarios cada vez que accesan al sistema. Una vez que finalizan la sesión, son eliminados de esta tabla. Además, aquí se almacena el código de identificación del usuario en sesión, asignado al usuario

cada vez que accesa al sistema. Esto con la finalidad de controlar las acciones que sólo pueden ser realizadas por usuarios en sesión, y para la verificación de preferencias de permiso.

En la figura 3 se muestra el diagrama de la base de datos definitiva:

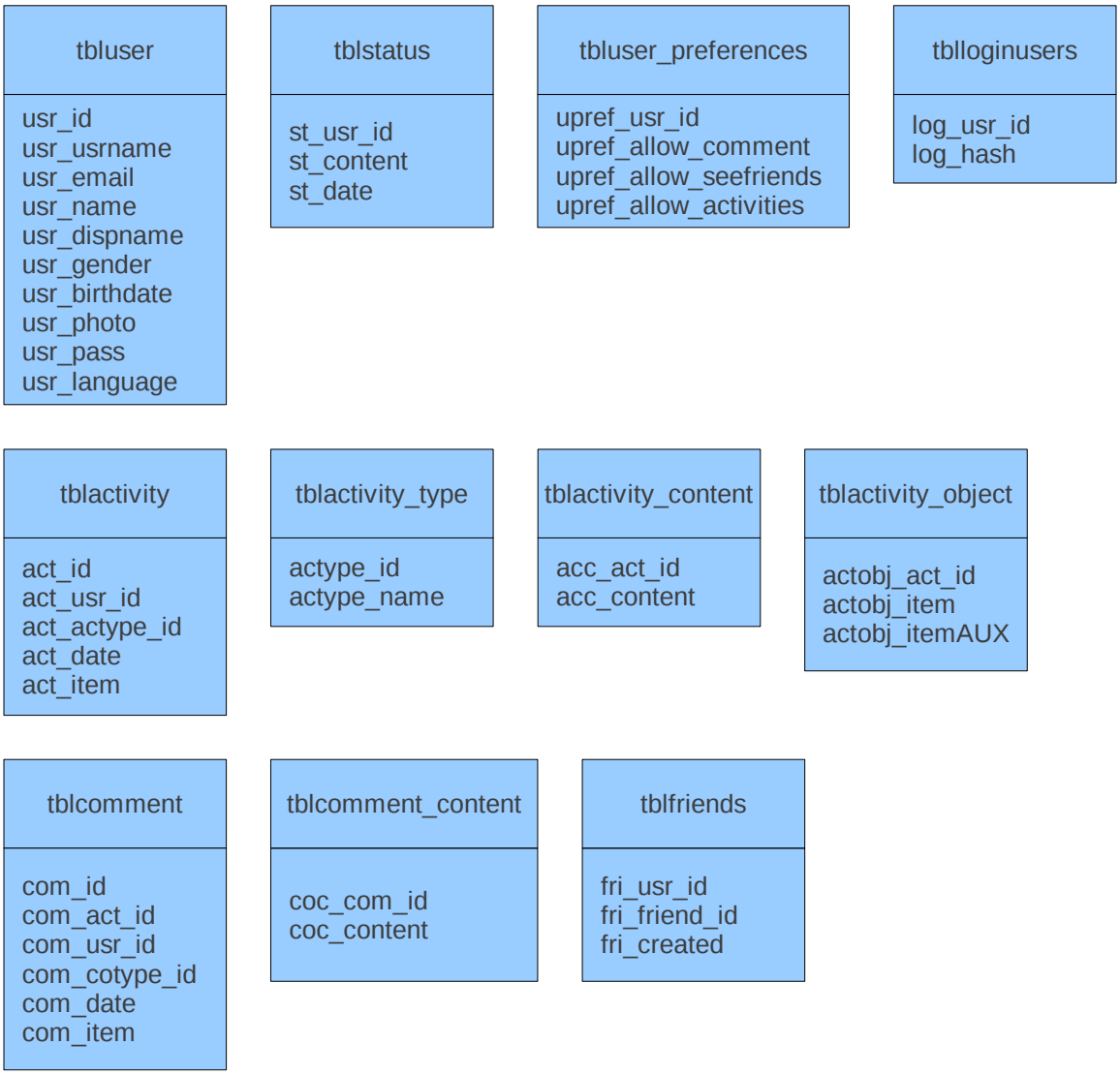

Figura 3. Tablas de la base de datos definitiva.

4.- *Implementación de base de datos*

Se crearon aquí los primeros SP's (procedimientos almacenados, por sus siglas en inglés) para realizar las consultas necesarias a la base de datos. La utilización de éstos permite realizar consultas más complejas sin necesidad de crear demasiadas sentencias desde la petición. Estos SP's pueden realizar varias consultas y devolver cada uno de los diferentes resultados, simplemente construyendo la sentencia para llamarlos y agregando los parámetros necesarios.

En esta etapa también fue necesario modificar ciertas tablas de la base de datos, ya que al momento de crear los SP's surgieron necesidades en la estructura de las tablas que no fueron consideradas durante el análisis de la base de datos.

Los primeros SP's creados fueron los de consulta de usuarios, el cual realizaba la consulta y obtenía los datos del usuario requerido, utilizando como referencia un identificador. Los SP's de consulta y cambio de estado de usuario, en donde empezaban a utilizarse conexiones de escritura en la base de datos. Además, para estos fue necesario el uso de un código para verificar que el usuario que realizaba tal consulta realmente tenía permisos para tal acción.

Los SP's de inicio y finalización de sesión, utilizados para agregar o eliminar usuarios a la tabla de usuarios en sesión. El SP de inicio de sesión, además de agregar al usuario a la tabla de usuarios en sesión, le asigna a cada usuario un código, utilizado para verificar que el usuario se encuentra en sesión activa. Esto debido a que ciertas funcionalidades de la aplicación, como agregar comentarios en actividades, cambio de estado, entre otras con permisos restringidos, requieren que el usuario solicitante se encuentre en sesión.

Las funciones de consulta al perfil de usuarios, actividad en particular, o el acceso al panel de control de cada usuario, requieren de consultas a la base de

datos que obtengan una o varias actividades generadas junto con los comentarios hechos en cada una de éstas actividades, según sea el caso.

Para cumplir con estas funcionalidades, fue necesario crear SP's que obtuvieran tal información de la base de datos. Esta es la parte del proceso de implementación de la base de datos más complicado, ya que fue necesario realizar sentencias muy complicadas para obtener estos datos.

La figura 4 muestra un ejemplo de un procedimiento almacenado:

```
DELIMITER ;;

DROP PROCEDURE IF EXISTS usr_sp_select_getUser_20110725;;
CREATE PROCEDURE usr_sp_select_getUser_20110725(numId INTEGER, myid INTEGER)

BEGIN

    IF numId >0 THEN

        SET @sql = CONCAT('SELECT usr_id,usr_name,usr_dispname,usr_gender,
(SELECT usr_fn_select_getAge_20110803(',numId,')) AS age,usr_photo
FROM mobiledb.tbluser WHERE usr_id =',numId);
        PREPARE STMT FROM @sql;
        EXECUTE STMT;
        DEALLOCATE PREPARE STMT;


    END IF;
    END;;

DELIMITER ;
```

Figura 4. Ejemplo de procedimeinto almacenado.

La funcionalidad de agregar comentarios sobre actividades fue pospuesta por decisión del asesor del proyecto, debido a que no estaba considerada dentro del proyecto definitivo, razón por la cual no se desarrolló el SP necesario desde

un principio. Esta funcionalidad, al igual que el cambio de estado, requiere de que el usuario solocitante esté en sesión. Son necesarios parámetros tales como el identificador del usuario, su código de sesión, el identificador de la actividad y el contenido del comentario.

Fueron necesarias correcciones a los SP de obtener actividades de un usuario y actividades de amigos, ya que la respuesta recibida no era la esperada. También se corrigieron los SP de iniciar y finalizar sesión, debido a cuestiones de seguridad de navegación.

5.- *Programación de clases, manejadores, entidades*
En esta etapa comienza la utilización del lenguaje PHP, para la programación del manejo de peticiones al API, tomando como base el modelo MVC.

Primero fue la estructuración básica de los controladores, ya sea para usuario, actividad, API, controladores de errores, o comentario. En cada uno de éstos, se define el nombre de los puntos de destino que tendrá el API, denominados acciones. Utilizando el nombre del controlador y el nombre de cada una de estas acciones, se harán las consultas al API para obtener de ella la información requerida dependiendo de la petición del usuario.

El controlador del API tiene como acción la de obtener el estado del API. Esto para verificar si el API se encuentra disponible para ser utilizada.

El controlador usuario cuenta con las acciones de obtener usuario, verificar estado del usuario, actualizar estado, iniciar sesión, finalizar sesión, obtener amigos, obtener actividades y obtener actividades generadas por amigos del usuario.

El controlador de actividad tiene como única acción la de obtener actividad. Esto debido a la sencillez de la aplicación. No fue necesario agregar más

acciones a este controlador.

El controlador de comentario, desarrollado al final, cuenta con la acción de agregar comentario en actividad.

Otros controladores utilizados fueron los controladores de errores, los cuales se utilizan en casos en los que ocurre un error al consultar el API, para enviar un aviso al usuario de que se ha presentado un error, sin que afecte esto su experiencia de navegación en la aplicación.

Se continuó con la programación de manejadores, los cuales serían llamados por los controladores por medio de las acciones.

Los manejadores construidos fueron el de usuario, actividad, estado de usuario y comentario. Cada manejador tiene funciones específicas según la entidad que representa. Dentro de estas funciones, se realiza la verificación de los parámetros recibidos, verificación de permisos en caso de ser necesario, estructuración de las sentencias de consulta a la base de datos, realización de esta consulta, y manipulación y estructuración de la respuesta recibida para ser devuelta al controlador.

El manejador de usuario se encarga de todas las acciones que requieran manipular información referente a usuarios, tales como obtener el identificador del usuario en sesión, obtener los datos de un usuario, iniciar y finalizar sesión, y obtener los amigos de un usuario.

El manejador de actividad tiene como fin la consulta y manipulación de información que se refiera a actividades, como la obtención de una actividad en específico, obtener las actividades generadas por un usuario, u obtener las actividades generadas por los amigos de un usuario en particular.

El manejador de estado de usuario es el encargado de la obtención y actualización del estado de un usuario. Realiza la estructuración de la sentencia para consultar la base de datos, y manipula la respuesta recibida para estructurarla y enviarla al controlador.

El último manejador creado fue el de comentario, cuyo único objetivo es la funcionalidad de agregar comentario sobre una actividad específica, tomando como parámetros para tal fin el identificador del usuario, su código de sesión, el identificador de la actividad y el contenido del comentario.

La programación de entidades incluye el desarrollo de clases con el nombre de cada entidad definida en el análisis de requerimientos. Entre ellas está la entidad usuario, actividad y estado de usuario. Estas incluyen métodos para obtener y asignar valores a sus propiedades definidas.

Otra entidad muy importante es la entidad resultado, que es la clase que define cómo deberán ser todas las respuestas devueltas por el API. Esta define una estructura estándar, que deberá ser respetada por cada punto destino de la aplicación.

La estructura a seguir es básicamente un arreglo con un campo que almacena un identificador de resultado, utilizado por el “front-end” para el desarrollo de sus procesos; un identificador de resultado HTTP; y un campo que contiene el resultado de la consulta, ya sea que haya sido exitosa o no, en cuyo caso deberá contener una representación vacía del objeto a recibir.

6.- *Programación de vistas*

Esta etapa se incluye aún dentro del modelo MVC. Las vistas se refieren a la forma en que se presentarán los datos como respuesta.

El API cuenta con dos tipos de formato para responder a las peticiones. El

formato a utilizar dependerá de un parámetro enviado al API dentro de cada petición.

El formato de la vista se llevará a cabo por un auxiliar invocado por el controlador después de que haya recibido la respuesta del manejador. Dependiendo del formato requerido, se llamará al auxiliar para dar formato XML o JSON, este último es el utilizado por el “front-end” para el desarrollo de sus procesos.

Los auxiliares reciben la información en forma de objeto, al cual se le aplica una estructuración, según sea el caso, y lo envían de vuelta al controlador, quien se encarga de presentar la información en el formato requerido.

7.- *Integración de vistas y peticiones con Javascript*

Como se mencionó antes, la aplicación está compuesta por un “back-end”, explicado anteriormente, y un “front-end”, el cual hace las peticiones al “back-end” y presenta la información requerida al usuario de una forma amigable y entendible. Es el “front-end” el encargado de hacer que el usuario tenga una buena experiencia al utilizar la aplicación.

Para esta parte del proyecto se utilizó PHP, librerías especiales de este lenguaje, como Smarty y Smarty-Gettext, Javascript y jQuery, librería de Javascript.

El usuario, al iniciar la aplicación, es dirigido, por medio de PHP, a la página de inicio de sesión. Aquí, al acceder a su cuenta, se crea en PHP una variable de sesión, donde se almacenan los datos del usuario mientras éste se encuentre en sesión activa.

La figura 5 muestra la pantalla de inicio de sesión:

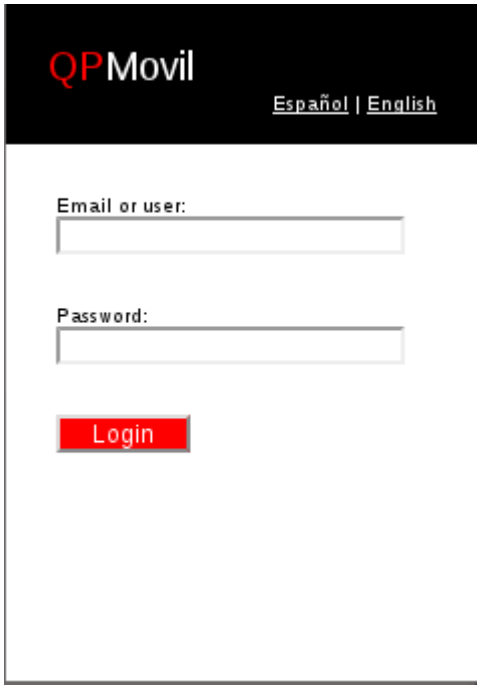

Figura 5. Página de inicio de sesión.

Después es dirigido a la página de panel de control, la página principal donde tiene acceso a la mayoría de las funcionalidades de la aplicación, la cual se muestra en la figura 6.

Básicamente, el “front-end” funciona mediante la construcción de archivos HTML a partir de plantillas ya definidas para cada página, esto se logra utilizando Javascript y jQuery, que permiten la manipulación de estructuras HTML de una manera sencilla.

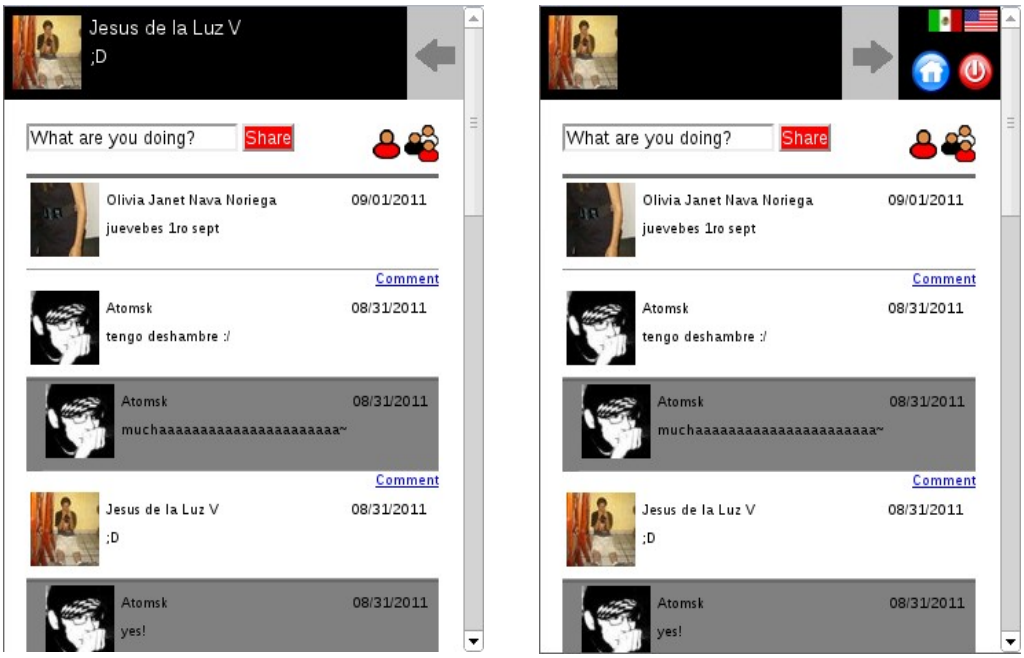

Figura 6. Página panel de control.

Entre las plantillas creadas, existen las plantillas de encabezado y pie de página, las cuales son comunes para todas las páginas, a excepción de la página de inicio de sesión, que no utiliza la plantilla de encabezado.

El despliegue de las páginas se realiza mediante el llamado a archivos PHP para cada página, ya sea para ingresar al panel de control, ver el perfil de algún usuario, o ver una actividad en específico.

Estos archivos PHP hacen el llamado a las plantillas necesarias: primero a la plantilla de encabezado, que siempre muestra la información del usuario en sesión; luego a cada plantilla según sea la página requerida; y por último a la

plantilla de pie de página, en la que se incluyen todos los archivos Javascript que serán requeridos por la página en cuestión.

La aplicación ofrece al usuario el cambio de idioma, español o inglés, para lo cual se utilizó Smarty-Gettext. Esta librería permite agregar a las plantillas palabras clave, o cadenas de palabras, las cuales pueden ser sustituidas por otras definidas en archivos de traducción que almacenan la clave y su respectiva cadena según el lenguaje que el usuario elija.

El diseño de la aplicación se realizó mediante el uso de archivos CSS, con los cuales se agregó el estilo a las plantillas, utilizando propiedades tales como identificadores y clases, que permiten agregar, mediante CSS, propiedades como tamaño de letra, color de fondo, entre otros, a los elementos incluidos en archivos HTML.

De manera similar a como trabaja CSS, jQuery permite manipular estos elementos, para agregarles efectos como animación, o reutilizar código HTML definido en las plantillas con funciones tales como la clonación de elementos, lo que resultó de mucha utilidad en páginas como el perfil de un usuario o el panel de control, en las que se muestran una serie de actividades generadas por ellos mismos o los amigos de estos.

La página de perfil de usuario se muestra en la figura 7:

Figura 7. Página perfil de usuario.

Estas actividades tienen la misma estructura, por lo que se utilizó un objeto definido en una plantilla y se clonó tantas veces como fue necesario.

Lo mismo ocurre con la página de amigos de un usuario. En esta se muestra un resúmen del perfil de cada uno de los amigos del usuario en cuestión, y cada uno de estos tiene la misma estructura, por lo que se aplica el mismo procedimiento.

La figura 8 muestra la página de amigos de usuario:

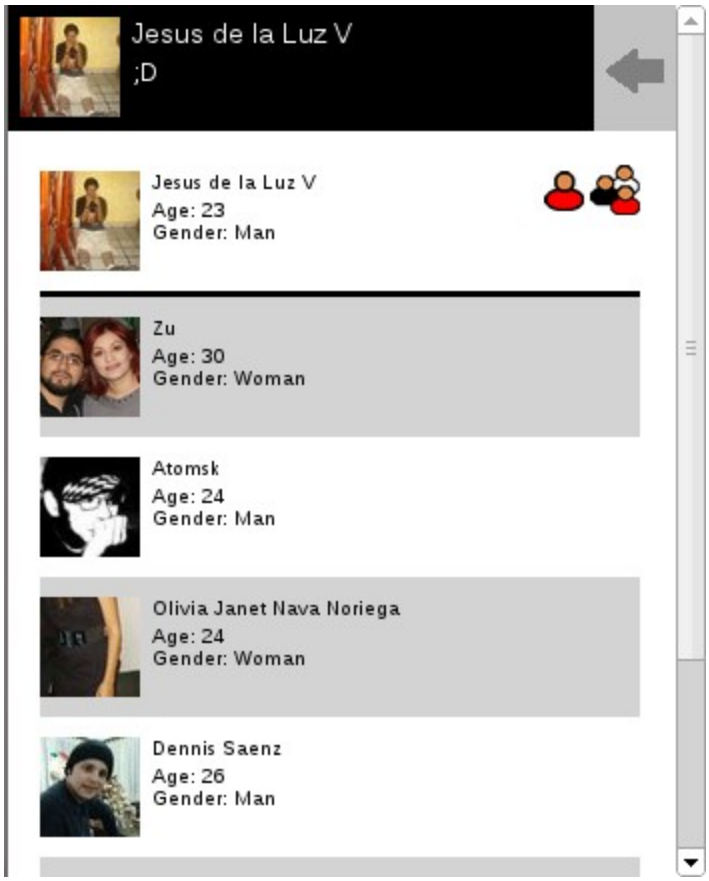

Figura 8. Página amigos de usuario.

Las peticiones al API se realizan utilizando la técnica AJAX con JSON, esto es, se construyen peticiones URL utilizando Javascript paa hacer la petición al API, y se recibe un objeto JSON a partir del cual se iniciará la presentación de los datos recibidos al usuario.

Una última funcionalidad agregada a la aplicación fue la actualización automática cada cierto tiempo definido, para evitar al usuario el tener que refrescar la página para verificar si ocurrió algún cambio en sus actividades o las de sus amigos.

Resultados obtenidos

El resultado fue una aplicación sencilla pero funcional. Los requerimientos iniciales del sistema fueron cumplidos en tiempo y forma.

El usuario de la aplicación tiene varias opciones a realizar, tales como verificar su estado y ver su perfil; consultar perfiles de otros usuarios, incluyendo actividades de éstos; ver su lista de amigos y la lista de amigos de otros usuarios; cambiar su estado, así como comentar en actividades realizadas por él mismo o sus amigos. Y funciones de utilería, como el cambio de idioma, haciendo el uso de la aplicación más cómodo sin importar el idioma de preferencia de cada usuario.

El contar con este tipo de aplicaciones en la actualidad, permite a las empresas tener mayor presencia en el mercado y mantenerse en la preferencia de los usuarios, debido al rápido desarrollo de la tecnología y a la creciente tendencia en el uso de dispositivos móviles para tener acceso a los servicios ofrecidos en diversos sitios.

Estas aplicaciones, además de ser sencillas y amigables, deben siempre conservar la identidad del sitio principal, lo que incluye ofrecer, en lo posible, las funcionalidades a las que el usuario tiene acceso en el sitio, así como respetar la imagen que se muestra a los usuarios en el sitio principal (logo, colores, estilos, etcétera).

Se cumplió con el objetivo principal, el cual era desarrollar las habilidades en programación de los participantes, así como el conocimiento de diferentes tecnologías y la experiencia de trabajar en un proyecto de este tipo.

Conclusiones y recomendaciones

El desarrollo de software no sólo requiere de tener un amplio conocimiento en algún lenguaje en particular, o varios de ellos. Es necesario tener el conocimiento de que existen tecnologías distintas con las que se pueden lograr resultados satisfactorios, y que requieren de una menor cantidad de esfuerzo.

El análisis de requerimientos es una parte esencial en el desarrollo de todo proyecto. Aquí debemos aplicar todas nuestras capacidades analíticas para tomar las mejores decisiones en cuanto a las tecnologías que se utilizarán, tomando en cuenta los requerimientos, el costo de aplicar tales tecnologías, la magnitud del proyecto, el tiempo estimado para la realización del proyecto, entre otros factores que influyen en todo el ciclo de vida de un proyecto.

Algo que debe tomarse en cuenta en todo proyecto, es que no es posible planear un proyecto a la perfección desde un principio. A medida que se avanza en el desarrollo aparecen ciertas cuestiones o imprevistos inevitables. Por más definido que esté el plan de desarrollo de un proyecto, este debe ser flexible para readaptaciones sobre la marcha, no sólo que sea fácil corregir algún problema, sino también, en caso de que se encuentre o surja una nueva tecnología o técnica que mejore los procesos, el proyecto pueda adaptarse a esta nueva opción.

Sobre todo proyectos de software deben tener esa flexibilidad, debido al constante desarrollo de nuevas y mejores tecnologías de información.

Creo que la programación para Web debería ser un tema obligatorio en la carrera, debido a la demanda de este tipo de aplicaciones tanto para empresas desarrolladoras de software como para todo negocio de cualquier giro, las peticiones de aplicaciones son normalmente para entornos Web.

Retroalimentación

Fortalezas

Durante el desarrollo del proyecto, encontré que, gracias a los conocimientos obtenidos durante mi carrera profesional, me fue fácil entender ciertos conceptos necesarios para participar en la realización de esta aplicación, a pesar de tener muy poca experiencia no sólo en el ramo laboral, sino en el manejo de las tecnologías que fueron utilizadas.

Las prácticas realizadas en la escuela y la utilización básica de lenguajes de programación y otras tecnologías, como gestores de bases de datos, me permitieron comprender mejor el objetivo del proyecto, como sería su funcionalidad, los requerimientos del sistema, y la estructura básica de un proyecto.

Debilidades

También, debido a la falta de conocimiento de programación en ambientes Web, se me presentaron ciertas dificultades sobre conceptos o manejo de tecnologías que durante mi carrera nunca me vi en la necesidad de aprender.

Oportunidades

Fueron de mucha utilidad conocimientos como el análisis de proyectos, y todo lo que incluye el proceso de desarrollo de un proyecto, además de la habilidad para adaptarme a nuevos entornos y la facilidad para adquirir nuevos conocimientos, el hábito de la investigación, y el razonamiento analítico de diferentes situaciones.

Referencias

[1] Wikipedia
<http://es.wikipedia.org/wiki/Modelo_Vista_Controlador> [11 de Septiembre, 2011]

[2] Wikipedia
<http://es.wikipedia.org/wiki/Front-end_y_back-end> [11 de Septiembre, 2011]

[3] Wikipedia
<http://es.wikipedia.org/wiki/JSON> [11 de Septiembre, 2011]

[4] Wikipedia
<http://es.wikipedia.org/wiki/Interfaz_de_programaci
%C3%B3n_de_aplicaciones> [11 de Septiembre, 2011]

[5] Wikipedia
<http://es.wikipedia.org/wiki/AJAX> [11 de Septiembre, 2011]

[6] Wikipedia
<http://es.wikipedia.org/wiki/Subversion> [11 de Septiembre, 2011]

Proyecto Introductorio: mobile.quepasa.com
Pláticas introductorias, del 30 de Mayo al 5 de Junio, 2011