



Proyecto Final: Proyectos de desarrollos para dispositivos móviles.

Nombre de la APP: VIVA MÉXICO

Integrantes:

Rodrigo Solana Chao A01129839

Rayniery R. Grullón Ozuna A01182791

<http://ravnosmerol.wix.com/vivamexico>



Tourism Board

Introducción

Viva México es una red social que permite compartir fotos de lugares del país, etiquetarlas con hashtags para que otros usuarios pueden buscar por hashtags imágenes existentes. Además los usuarios pueden comentar las publicaciones, lo cual genera interacción entre los usuarios y sentido de comunidad. Las imágenes pueden ser reportadas como ofensivas para que sean revisadas por un administrador y éste determina borrarlas o no.

Definición del problema

Actualmente en internet existen comunidades para toda clase de personas y actividades, pero los creadores de **Viva México** no hemos encontrado un lugar exclusivamente dedicado para personas que amen conocer y viajar por México.

Esta app es una red social dedicada al intercambio de fotos de cualquier lugar de México, así un turista que desee ver imágenes verdaderas de un lugar al que esté pensando visitar podrá buscarlas y ver fotos reales tomadas por personas que hayan estado ahí previamente. Así el turista sabrá cómo es el lugar realmente y qué esperar de éste. Porque un problema que a veces los turistas se encuentran al estar investigando sobre algún lugar es que la única información disponible es de los dueños o comerciantes del lugar, por lo que no pueden encontrar una opinión de terceros.

Viva México intenta llenar ese vacío de información y fotos reales al promover la compartición de imágenes y comentarios.

Identificación de empresas

Este proyecto puede ser auspiciado por la Secretaría de Turismo del país.

Librerías y paquetes utilizados

En la realización de la app utilizamos las siguientes librerías:

ionic: un framework de desarrollo de aplicaciones móviles híbridas.

ngCordova: una colección de extensiones de AngularJS que facilita el uso de Cordova desde aplicaciones Angular.

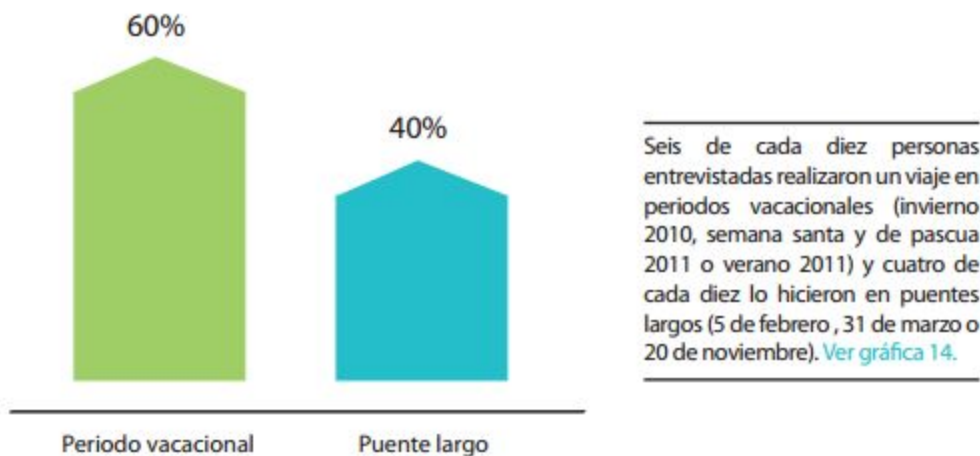
Estudio de mercado de factibilidad de la aplicación

Hay 119 millones 530 mil 753 habitantes en México (INEGI) y en el primer bimestre del 2016 se registraron 5.6 millones de turistas internacionales (INEGI).

Solamente tomando en cuenta el turismo de gente que habita en México, la siguiente gráfica fue tomada de un estudio realizado por Datatur - Secretaría de Turismo.

FLUJO DE TURISTAS POR PERIODO VACACIONAL O PUENTE LARGO

GRÁFICA 14



Como se puede observar es mucha la gente que sale de vacaciones ya sea en periodo vacacional o puente largo.

En la gráfica 54 se puede observar la diferencia de cómo la gente decidió el destino para viajar, lo que nos muestra muy poco porcentaje de internet VS recomendación, eso se debe a que actualmente no hay ninguna aplicación que permita conocer otros lugares de México sin tener que salir, para así tomar una decisión de dónde viajar.

Pero para eso llega VIVA MÉXICO ya que en la actualidad las personas que cuentan con un teléfono celular incluye al 87% de la población mexicana de acuerdo con IAB, estamos hablando de 95 millones 624 mil 602 personas que cuentan con un dispositivo móvil.

PRINCIPALES DESTINOS VISITADOS

GRÁFICA 25

PRINCIPALES DESTINOS VISITADOS

El Distrito Federal fue el lugar preferido para viajar, seguido de Quintana de Roo y de Veracruz.

[Ver gráfica 25.](#)



Como se puede observar la mayoría de las personas decidió vacacionar en lugares como el DF, Quintana Roo, Veracruz... pero qué hay de los demás destinos que también cuentan con una gran variedad de lugares para el turismo y que casi no son tan conocidos?

Por ejemplo, pueblos mágicos, lugares que son casi Vírgenes... hay mucho que explorar en México que incluso no hay necesidad de salir del país para disfrutar de cosas nuevas, la idea es ver, comentar y publicar destinos, para que la gente siga viajando y apoyando al turismo en México.

Conclusiones

Es demasiado fácil realizar aplicaciones móviles con apoyo del framework de ionic, es muy amigable y la documentación y/o manual es de gran apoyo.

En general utilizamos los puntos vistos en clase, generamos los html para las vistas o templates, así mismo como los JS para la aplicación, los controladores y los servicios utilizados para poder realizar las operaciones en el servidor.

Generamos los servicios.php de las operaciones CRUD de diferentes maneras con POST y GET dependiendo de lo que requería en ese momento la aplicación.

En la siguiente imagen es un ejemplo de un servicio que captura toda la información de una persona mediante un GET[ID]:

```
<?php
header('Access-Control-Allow-Origin: *');
//configuracion de la base de datos
$db_name = 'pddm_1129839';
$hostname = 'localhost';
$username = '1129839_user';
$password = '1129839';
//capturamos el flujo de entrada JSON que viene en el request HTTP
//la funcion get_contents(".") nos permite obtener el contenido (body) del request HTTP.
$jsonData = file_get_contents("php://input");
//deserealizacion para convertir un string en formato JSON a un objeto Alumno
$data = json_decode($jsonData);

//conexion a la base de datos
$conexion = new PDO("mysql:host=$hostname;dbname=$db_name", $username, $password);
//or die ("Error al conectarse a la base de datos");
//hacemos el query
$sql = "select * from Personas where idPersonas = " . $data->id;
$q = $conexion->prepare($sql);
// echo "limit: ". $data->startIndex;
// echo "\n";

// $result = $q->execute(array(":startIndex"=>$data->startIndex));
$result = $q->execute();

if ($result) {
    echo json_encode(array_merge(array('respCode' => 1), array('result'=>$q->fetchAll( PDO::FETCH_ASSOC ))));
} else {
    echo json_encode(array("result" => 0));
}
//cerramos conexion
$conexion = NULL;
?>
```

Implementamos la base de datos con sus respectivas llaves primarias y en algunas tablas pusimos restricciones como DELETE ON CASCADE, para asegurar la integridad de los datos en la base...

Aquí se muestra como se creó la tabla de Comentario con DELETE ON CASCADE su llave primaria y foreign keys:

```
| Comentario | CREATE TABLE `Comentario` (  
  `idComent` int(11) NOT NULL AUTO_INCREMENT,  
  `idPersonas` int(11) DEFAULT NULL,  
  `idPublicacion` int(10) unsigned NOT NULL,  
  `texto` varchar(250) NOT NULL,  
  `votos` int(11) DEFAULT NULL,  
  `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`idComent`),  
  KEY `par_ind` (`idPublicacion`),  
  CONSTRAINT `Comentario_ibfk_1` FOREIGN KEY (`idPublicacion`) REFERENCES `Media` (`id`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1 |
```

En las operaciones CRUD también realizamos no solo sentencias sencillas de mysql, sino tambien queries anidados y sentencias JOIN.

La siguiente sentencia es del servicio de reporte_delete.php el cual borra datos de 2 tablas, de igual manera para asegurar la integridad en la base de datos, sirve de manera similar al Delete On Cascade.

```
$sql = "delete m, r from Reporte r join Media m on r.idMedia = m.id where r.idMedia = ". $data->idMedia;
```

El siguiente query es de igual manera un tipo join para regresar datos de 2 tablas diferentes.

```
$sql = 'SELECT coment.texto, coment.idPublicacion, coment.fecha, pers.imagen, pers.Usuario  
FROM Comentario coment, Personas pers where coment.idPersonas = pers.idPersonas  
and coment.idPublicacion =' . $idMedia. ' order by coment.fecha';
```

Creemos que en este proyecto se cumplieron los objetivos personales, los cuales eran terminar una aplicación móvil con todos los servicios necesarios para su correcto funcionamiento y una buena colaboración en equipo.

Problemas encontrados durante el desarrollo del proyecto ninguno que en la documentación o buscando en internet no pudiéramos solucionar.

Referencias:

<http://www.inegi.org.mx/>

[http://www.datatur.sectur.gob.mx/Documentos%20Publicaciones/EncuestaFlujoTurism%20\(1\).pdf](http://www.datatur.sectur.gob.mx/Documentos%20Publicaciones/EncuestaFlujoTurism%20(1).pdf)