Tecnológico de Monterrey

# Proyecto Integrador

## Cámara de Comercio del Canadá en México

## Manual Técnico

## Aplicación Móvil

Tecnológico de Monterrey

# Índice

Tecnológico de Monterrey

# Introducción

Este manual tiene el objetivo que un posible equipo futuro de desarrolladores pueda instalar y administrar la aplicación web y la aplicación móvil desarrollada específicamente para la Cámara de Comercio del Canadá en México, describiendo los pasos necesarios para que cualquier persona con cierto conocimiento en sistemas pueda hacerlo, describiendo los antecedentes detalladamente.

La aplicación móvil CANCHAM, fue desarrollada usando el framework open source llamado Ionic, construido usando HTML, CSS y Javascript para el desarrollo de aplicaciones híbridas para dispositivos móviles. Otra ventaja de esta, es que utiliza Angular, para el desarrollo de código dinámico y Cordova para el uso de los sensores del dispositivo (cámara, GPS, agenda de contactos, etc).

Las aplicaciones híbridas, permiten que con un código único web podamos generar aplicaciones para todas las plataformas: Android, iOS, etc., creando una aplicación nativa.

Tecnológico de Monterrey

## Instalación de Ionic

Para el buen funcionamiento y mantenimiento de la aplicación, se necesitan instalar las dependencias que pide este framework, Ionic. Puede que algunas ya las tengan instaladas, pero detallaremos paso a paso, incluso para saber que versión tenemos.
Entre los requerimientos técnicos el Sistema Operativo, puede ser: ya se Windows, Mac OS x, Linux, etc.
Los pasos son los siguientes:

1. Antes que nada Ionic necesita *Node.js* y *npm*, para instalarlo simplemente podemos descargarlo e instalarlo desde su Web: https://nodejs.org/en/
   - Para verificar la instalación de *node* se teclea desde la terminal `node -v`
   - Para verificar la instalación de *npm* se teclea desde la terminal `npm -v`

```
MacBook-Pro-de-Krisliz:~ KrislizAl$ node -v
v6.10.0
MacBook-Pro-de-Krisliz:~ KrislizAl$ npm -v
3.10.10
```

2. Instalar *git*, que podemos descargarlo en su página http://git.scm.com/downloads
   - Para comprobar la versión instalada teclear desde la terminar: `git --version`
3. Instalaremos Bower para administrar las dependencias de nuestra librería, este esta construido sobre Node.js. Importante para el desarrollo del front-end.
   - Ya que tenemos pm instalado, podemos teclear desde la terminal para instalarlo:

```
npm install bower -g
```

   - Para verificar su correcta instalación y versión `bower -v`
4. Instalar gulp, que ayuda a Ionic a convertir scss a css. Para instalarlo tecleamos desde la terminal

```
npm install gulp -g
```

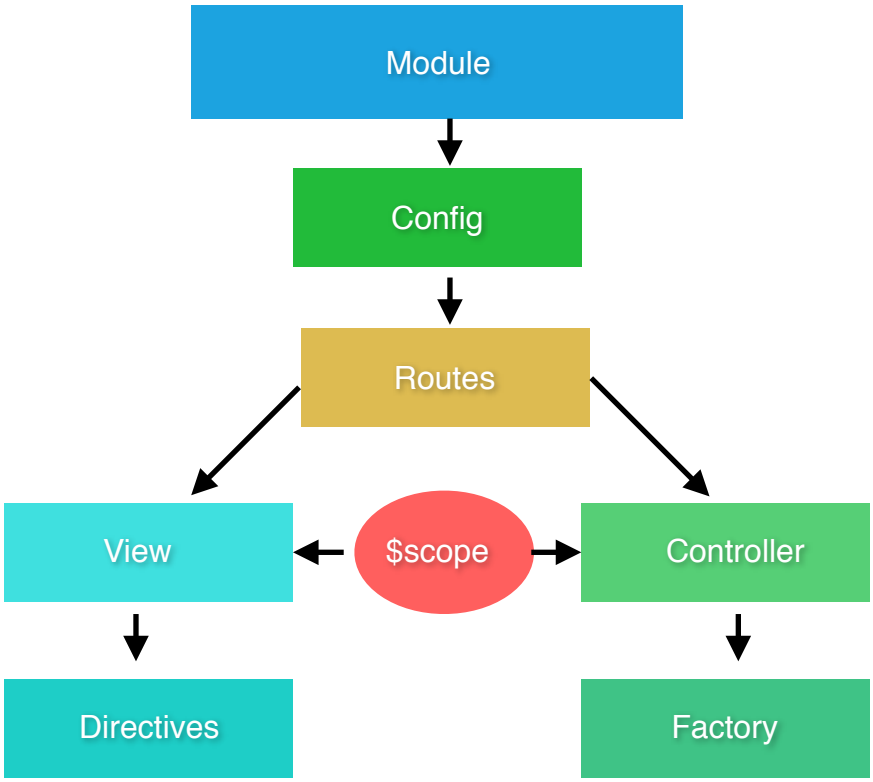
   Para verificar su correcta instalación y versión `gulp -v`
5. Y finalmente, lo más importante Instalar Ionic y Cordova, que se puede hacer mediante el siguiente comando.

```
npm install -g cordova ionic
```

Tecnológico de Monterrey

## Arquitectura de la aplicación

Como ya se menciono anteriormente este framework Ionic, usa Angular que a su vez usa el patrón *Vista-Controlador (View-Controller*), donde los controladores están asociados a las vistas y se encargan de proporcionar los datos y funcionalidad a la aplicación. Así que la arquitectura que se manejo para este proyecto fue la siguiente.




Donde:

- *Controller:* Obtienen datos, ya sea de uno o más *Servicios o Factorias*, enviándolo a una vista (template), a través de la variable $scope
- *View:* Como su nombre dice, es la visualización de la pantalla (o una parte), también obtienen los datos a través de la variable $scope.
- *Config & Routes:* Es el enlace de los controladores con las vistas o templates correspondientes
- *Directives:* Permiten crear y usar componentes con aspecto y comportamiento personalizado.

Tecnológico de Monterrey

## Estructura del directorio

La estructura de nuestro proyecto esta, de la siguiente manera, con su respectiva función

```
CanChamApp/                          Folder que contiene todas las dependencias del proyecto
    ├─ resources/                    Recursos específicos de las plataformas (Assets únicos).
    |    ├─ android/                 Recursos específicos de Android.
    |    |    ├─ icon                Imagen del ícono de la app.
    |    |    └─ splash              Imagen que aparece al entrar a la app.
    |    └─ ios/                     Recursos específicos de iOS.
    |    |    ├─ icon                Imagen del ícono de la app.
    |    |    └─ splash              Imagen que aparece al entrar a la app.
    ├─ scss/                         Código SCSS que será compilado a la carpeta www/css/.
    ├─ www/                          Código fuente PRINCIPAL de nuestra app web: HTML, CSS, Js, img, etc.
    |    ├─ css/                     Hojas de estilo que se usen en la aplicación.
    |    ├─ img/                     Donde se almacenan la imagenes de nuestro proyecto.
    |    ├─ js/                      Contendrá todo el código JavaScript de la aplicación.
    |    ├─ lib/                     Librerías que use la app. Viene incluido todo el código de la librería Ionic.
    |    |    ├─ angular             El core AngularJS framework
    |    |    ├─ angular-animate     Permite las animaciones en la app
    |    |    ├─ angular-pdf-viewer  Permite la visualización de los archivos PDF
    |    |    ├─ angular-sanitize    Proporciona funcionalidad para sanitizar el HTML.
    |    |    ├─ angular-ui-router   Gestiona las transiciones entre los estados de la aplicación.
    |    |    ├─ ionic
    |    |    ├─ ngCordova
    |    |    └─ pdfjs-dist          Portable Document Format (PDF). Parsea y rendeerea PDF's-
    |    └─ templates/               Almacena las plantillas o vistas de la aplicación.
    ├─ plugins/                      Plugins o módulos instalados para la app.
    |    ├─ cordova-plugin-splashscreen/
    |    ├─ cordova-plugin-calendar/
    |    ├─ cordova-plugin-statusbar/
    |    ├─ cordova-plugin-googleplus/
    |    └─ cordova-plugin-inappbrowser/
    ├─ platforms/                    Plataformas para las que esta desarrollada el proyecto.
    |    ├─ android/                 Folder que contiene lo necesario para que funcione en Android.
    |    |    ├─ libs
    |    |    ├─ platform_www
    |    |    ├─ res
    |    |    ├─ src
    |    |    ├─ cordova
    |    |    ├─ assets
    |    |    └─ CordovaLib
    |    └─ ios/                     Folder que contiene lo necesario para que funcione en iOS.
    |    |    ├─ www
    |    |    ├─ build
    |    |    ├─ platform_www
    |    |    ├─ cordova
    |    |    ├─ CordovaLib
    |    |    └─ CanChamApp
    ├─ hooks/                        Añade scripts que se ejecutarán cuando se produzcan determinados eventos.
    └─ node_modules/
```

## Funcionalidad del Sistema

### Casos de Uso

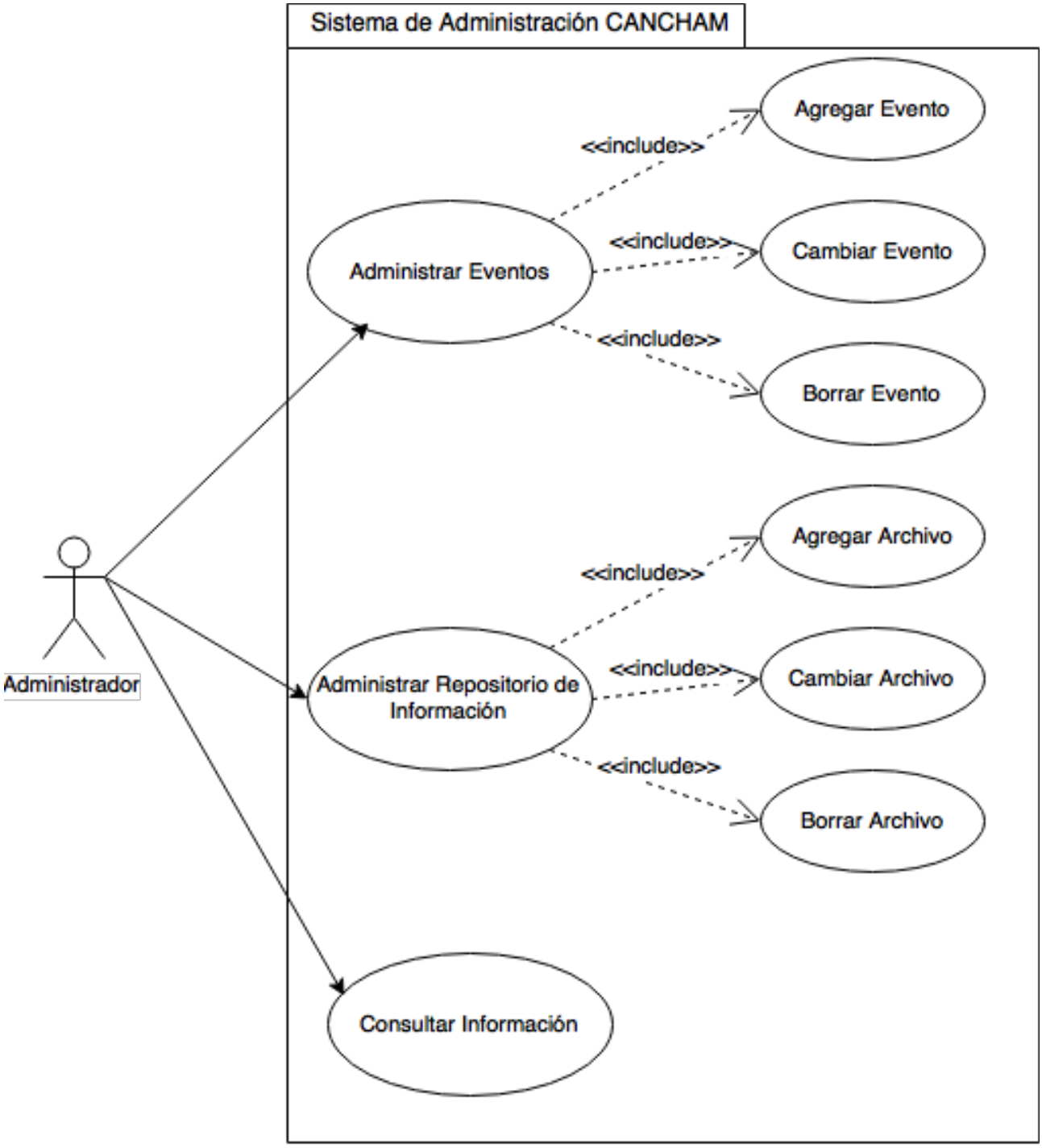



**Imagen. Diagrama UML de casos de uso del sistema de administración**

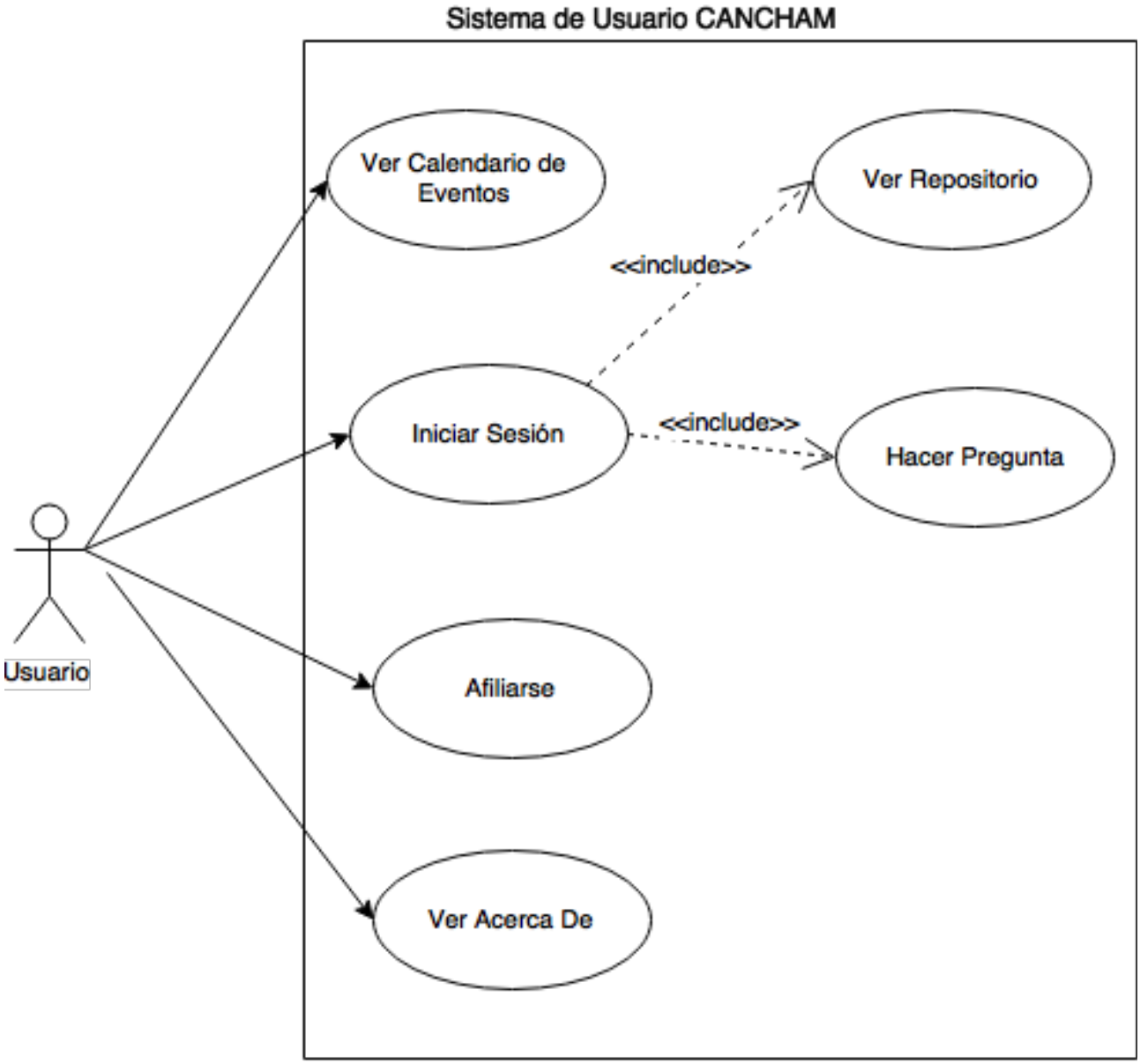



**Imagen. Diagrama UML de casos de uso del sistema de usuario**

Tecnológico de Monterrey

## Casos de Uso Detallados

A continuación mostraremos los casos de uso detallados de la aplicación web CANCHAM, que es la página de administrador, la interfaz que ayuda a manejar la aplicación móvil CANCHAM.

| **CASO DE USO: Agregar un nuevo evento** | |
|---|---|
| **Actores:** Administrador<br>**Objetivo:** Registrar en el sistema un nuevo evento con los campos requeridos en la aplicación web<br>**Precondiciones:**<br>1. Ser administrador de la aplicación web<br>**Postcondiciones:**<br>1. El evento se agregará a la base de datos.<br>**Descripción:** El administrador tiene la capacidad de Agregar un Evento, llenando los campos necesarios. Esto le ayudará a que en la aplicación se veo reflejado este nuevo evento, donde permite a los usuarios poder agregarlos a su calendario | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El administrador da click en la sección *Agregar Evento.*<br>2. Se llenan TODOS los campos que se solicitan en la página<br>3. Al final se da click en el botón *Agregar Evento.* | 4. El sistema agrega en la base de datos un nuevo evento.<br>5. El sistema actualizará la página web, de igual manera permite, si se desea agregar de nuevo un evento.<br>6. Los cambios se ven reflejados en la aplicación móvil CANCHAM |
| **Alternativas.**<br>2. El administrador no llena todos los campos, lo que hará una falta de información al evento | |

Tecnológico de Monterrey

| CASO DE USO: Eliminar evento | |
|---|---|
| **Actores:** Administrador<br>**Objetivo:** Dar de baja del sistema y base de datos, un evento para que no exista confunsiones en la aplicación móvil CANCHAM<br>**Precondiciones:**<br>1. Ser administrador de la aplicación web<br>**Postcondiciones:**<br>1. El evento se verá eliminado de la base de datos, por lo que no se verá en la aplicación móvil CANCHAM<br>**Descripción:** El administrador tiene la capacidad de borrar un evento si así lo desea, lo que en dado caso, se reflejará en la aplicación móvil CANCHAM, evitando al usuario que se confunda, si este llegase a cancelarse | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El administrador da click en la sección *Borrar Evento.*<br>2. Se busca el evento que se desea eliminarse, dentro de la lista de evento<br>3. Al final se da click en el botón *Eliminar Evento.* | 4. El sistema dará de baja en la base de datos el evento seleccionado.<br>5. El sistema actualizará la página web, de igual manera para verificar que se haya borrado exitosamente<br>6. Los cambios se ven reflejados en la aplicación móvil CANCHAM |
| **Alternativas.**<br>3. El evento ya ha sido eliminado por lo que se encuentra en la lista | |

| CASO DE USO: Agregar un nuevo archivo | |
|---|---|
| **Actores:** Administrador<br>**Objetivo:** Registrar en el sistema un nuevo archivo para que se vea reflejado en la sección de Repositorio de la aplicación móvil<br>**Precondiciones:**<br>1. Ser administrador de la aplicación web<br>**Postcondiciones:**<br>1. El archivo se agregará a la base de datos.<br>**Descripción:** El administrador tiene la capacidad de Agregar un Archivo, de cualquier formato, teniendo la posibilidad de elegir a que evento estará disponible dicho archivo. | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |

| CASO DE USO: Agregar un nuevo archivo | |
|---|---|
| 1. El administrador da click en la sección *Agregar Archivo.*<br>2. Selecciona a que Evento desea agregarlos de los ya existentes<br>3. Da click en el botón *Upload File* y elige el archivo deseado | 4. El sistema agrega en la base de datos un nuevo archivo,<br>5. El sistema actualizará la página web, de igual manera permite, si se desea agregar de el archivo de manera exitosa a dicho evento.<br>6. Los cambios se ven reflejados en la aplicación móvil CANCHAM |
| **Alternativas.**<br>3. No se encuentra el evento deseado por lo que no se puede subir el archivo, antes de seleccionar a que evento | |

| CASO DE USO: Eliminar archivo | |
|---|---|
| **Actores:** Administrador<br>**Objetivo:** Dar de baja del sistema un archivo, que estuvo ahí previamente para eliminarlo de la sección de Repositorio de la aplicación móvil CANCHAM<br>**Precondiciones:**<br>1. Ser administrador de la aplicación web<br>**Postcondiciones:**<br>1. El archivo se eliminará de la base de datos.<br>**Descripción:** El administrador tiene la capacidad de Eliminar un Archivo, de cualquier evento, ayudando así a no confundir al usuario en la sección de Repositorio. | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El administrador da click en la sección *Borrar Archivo.*<br>2. Selecciona que archivo de los ya existentes, se desea eliminar<br>3. Da click en el botón *Eliminar Archivo* | 4. El sistema da de baja la base de datos el archivo deseado,<br>5. El sistema actualizará la página web, de igual manera se verá reflejado los cambios.<br>6. Los cambios se ven reflejados en la aplicación móvil CANCHAM en la secciónn de Repositorio. |

| CASO DE USO: Eliminar archivo |
|---|
| **Alternativas.**<br>3. No se encuentra el archivo deseado por lo que el sistema queda igual |

| CASO DE USO: Consultar Información | |
|---|---|
| **Actores:** Administrador<br>**Objetivo:** Localizar en la aplicación web, toda la información, ya sea el repositorio, eventos y la más importante preguntas al expositor<br>**Precondiciones:**<br>1. Ser administrador de la aplicación web<br>**Postcondiciones:**<br>1. El administrador podrá visualizar las opciones deseadas.<br>**Descripción:** El administrador tiene la capacidad de visualizar información de la aplicación y modificarla. En esta parte, la que tiene más importancia es el manejo de preguntas al expositor | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El administrador da click en la sección *Consultar Información.*<br>2. Selecciona la opción que desea visualizar.<br>3. Por ejemplo Preguntar al Expositor y da click en *Submit* | 4. El sistema desplegará todas las preguntas en tiempo real hechas por los usuarios en tiempo real<br>5. Tiene la posibilidad de eliminar preguntas, solo dando click en el botón que dice Borrar del lado derecho |
| **Alternativas.**<br>3. Puede seleccionar también para consultar información de Eventos | |

Ahora mostraremos los casos de uso detallados de la aplicación móvil CANCHAM,

Tecnológico de Monterrey

| CASO DE USO: Ver Calendario de Eventos | |
|---|---|
| **Actores:** Usuario<br>**Objetivo:** Visualizar los eventos próximos de CANCHAM y agendarlos al celular<br>**Precondiciones:**<br>1. Ser usuario y contar con dispositivo móvil (Android, iOS)<br>**Postcondiciones:**<br>1. El usuario tendrá agendado en su celular el evento<br>**Descripción:** El usuario, al interactuar con la app, tiene la posibilidad de visualizar los próximos eventos de CANCHAM y agregarlos a la agenda del celular | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El usuario selecciona la parte que dice *Calendario de Eventos.*<br><br>*3.* Una vez visto los eventos, el usuario selecciona que evento quiere guardar en su agenda.<br>4. Da click en Añadir al Calendario | 2. El sistema despliega una nueva ventana , donde el usuario puede visualizar los eventos próximos<br><br>5. El sistema abrirá el calendario respectivamente del celular y almacenará el evento para recibir una notificación días próximos al evento |
| **Alternativas.**<br>2. No hay eventos disponibles, | |

| CASO DE USO: Iniciar Sesión | |
|---|---|
| **Actores:** Usuario<br>**Objetivo:** Iniciar Sesión en el sistema para poder visualizar el menu completo.<br>**Precondiciones:**<br>1. Ser usuario y contar con dispositivo móvil (Android, iOS)<br>2. Contar con una cuenta ya sea Gmail o Facebook.<br>**Descripción:** El usuario, tiene la posibilidad de iniciar sesión con alguna de sus cuentas personales, para poder visualizar el menú completo | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |

| CASO DE USO: Iniciar Sesión | |
|---|---|
| 1. El usuario selecciona la parte que dice *Iniciar Sesión.* | 2. El sistema despliega una nueva ventana , donde el usuario puede seleccionar con que cuenta desea entrar (Facebook o Gmail)<br><br>3. El sistema despliega el nuevo menú que incluye la sección de repositorio y preguntas al expositor, al igual que es un menú personalizado, ya que puede verse de igual manera su nombre y foto. |
| **Alternativas.**<br>2. No cuenta con ninguna cuenta personal. | |

| CASO DE USO: Ver Repositorio | |
|---|---|
| **Actores:** Usuario<br>**Objetivo:** Visualizar los archivos disponibles del evento y descargarlos al celular<br>**Precondiciones:**<br>1. Ser usuario y contar con dispositivo móvil (Android, iOS)<br>2. El usuario DEBE de ya haber iniciado sesión con alguna de sus cuentas<br>**Descripción:** El usuario, tiene la posibilidad de visualizar los archivos del eventos y descargarlos a su celular, | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El usuario selecciona la parte que dice *Repositorio.*<br><br>3. Una vez visto por el usuario, este elige cual desea descargar y da click en Visualizar Archivo. | 2. El sistema despliega una nueva ventana , donde se visualizan todos los archivos disponibles con su nombre y botón de descarga<br><br>4. El sistema le da a elegir el usuario que app desea para visualizar el archivo (p.e iBooks, Dropbox, etc.) |
| **Alternativas.**<br>— | |

| CASO DE USO: Hacer Pregunta | |
|---|---|
| **Actores:** Usuario<br>**Objetivo:** Hacer preguntas al expositor del evento en tiempo real.<br>**Precondiciones:**<br>1. Ser usuario y contar con dispositivo móvil (Android, iOS)<br>2. El usuario DEBE de ya haber iniciado sesión con alguna de sus cuentas<br>**Descripción:** Durante el evento, el usuario tiene la posibilidad que mediante la aplicación hacer una pregunta y el moderador es el que decide cuales serán contestadas | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El usuario selecciona la parte que dice *Preguntas al expositor*<br><br>3. Una vez escrita la pregunta, el usuario da click en el botón Enviar. | 2. El sistema despliega una nueva ventana , donde se podrá escribir una pregunta, para luego enviarla<br><br>4. El sistema lanzará una notificación de que la pregunta fue enviada y esta se verá reflejada en la página del administrador |
| **Alternativas.**<br>— | |

| CASO DE USO: Afiliarse | |
|---|---|
| **Actores:** Usuario<br>**Objetivo:** Obtener información para poder solicitar una membresía con CANCHAM<br>**Precondiciones:**<br>1. Ser usuario y contar con dispositivo móvil (Android, iOS)<br>**Descripción:** Esta sección permite al usuario interesado en solicitar una membresía, obtener información más detallada | |
| **Curso típico de los eventos** | |
| **Acciones del actor** | **Respuesta del sistema** |
| 1. El usuario selecciona la parte que dice *Afíliate*<br><br>3. El usuario DEBE de llenar todos los campos y luego dar click en el botón solicitar | 2. El sistema despliega una nueva ventana con un formulario, que el usuario debe completar.<br><br>4. El sistema lanzará una notificación de solicitud enviada exitosamente. |

Tecnológico de Monterrey

| CASO DE USO: Afiliarse |
| --- |
| **Alternativas.**<br>— |

## Diagrama Entidad-Relación

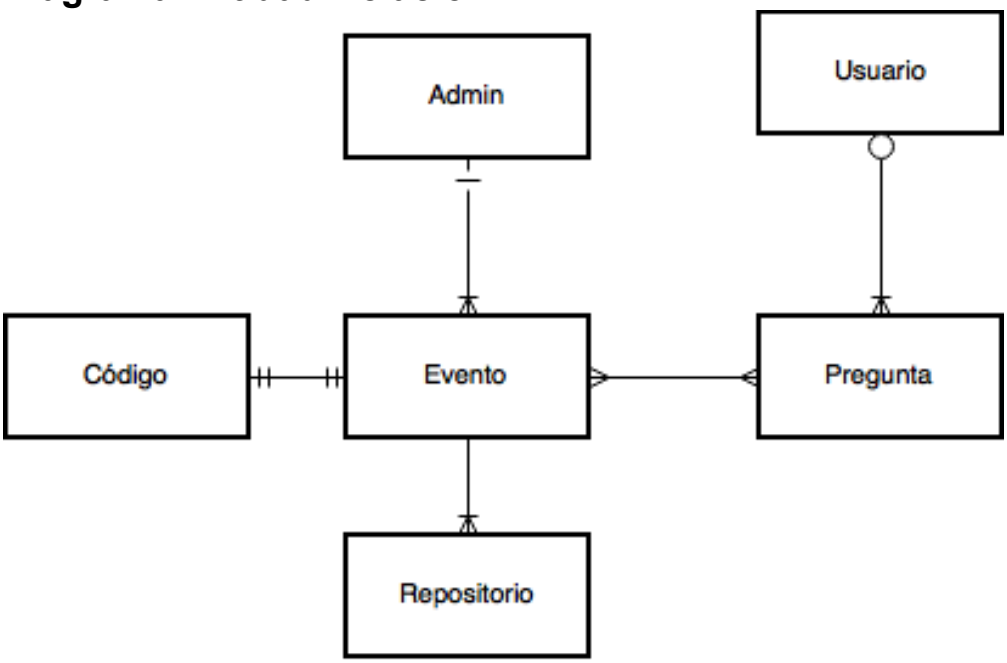



**Imagen. Diagrama E-R de la Base de Datos**

## Diccionario de datos

### Nombre de la tabla: admin

| Campo | Tipo | Null | Llave |
| --- | --- | --- | --- |
| usuario | varchar(15) | NO | PRIMARY |
| passwd | varchar(10) | YES | |

### Nombre de la tabla: codigos

| Campo | Tipo | Null | Llave |
| --- | --- | --- | --- |
| id | char(15) | NO | PRIMARY |
| nombre_evento | char(100) | NO | |

### Nombre de la tabla: eventos

| Campo | Tipo | Null | Llave |
| --- | --- | --- | --- |
| id | int(11) | NO | PRIMARY |

| Campo | Tipo | Null | Llave |
|---|---|---|---|
| nombre | char(100) | NO | |
| fecha | datetime | YES | |
| direccion | char(150) | NO | |
| descripcion | char(150) | NO | |

**Nombre de la tabla: preguntasExpositor**

| Campo | Tipo | Null | Llave |
|---|---|---|---|
| id | int(11) | NO | PRIMARY |
| pregunta | char(100) | NO | |

**Nombre de la tabla: repositorio**

| Campo | Tipo | Null | Llave |
|---|---|---|---|
| id | int(11) | NO | PRIMARY |
| nombre_archivo | char(100) | NO | |
| liga_archivo | char(200) | NO | |

**Nombre de la tabla: usuario**

| Campo | Tipo | Null | Llave |
|---|---|---|---|
| id | int(11) | NO | PRIMARY |
| nombre | char(100) | NO | |
| apellido_paterno | char(50) | NO | |
| apellido_materno | char(50) | NO | |
| mail | char(50) | NO | |
| pass | char(20) | NO | |

## Archivos principales

Estos archivos Javascript, son los que le dan una correcta funcionalidad en su mayoría y se ven reflejado en los archivos .html, ya que al igual que los templates, estos controlan las rutas, paso de datos y servicios.

- app.js : Mapeo de rutas para poder desplegar las vistas y que controlador usa dicha vista.
- controllers.js: Como dice su nombre, controla el paso de los datos, almacenando en la variable $scope que muestra los datos recibidos, dandole formato adecuado.
- services.js:

A continuación los detallaremos y daremos un ejemplo en base a nuestro proyecto

**Archivo: app.js**

Lo importante de este archivo es la siguiente línea

```
angular.module('starter', ['ionic']
```

Esta crea un aplicación AngularJs llamada 'starter', que incluye toda la funcionalidad de Ionic. Además permite a la aplicación, saber que ruta tomar y desplegar la vista correspondiente, así como el controlador. Este archivo tiene la siguiente estructura

```
.config(function($stateProvider, $urlRouterProvider) {
  $stateProvider

  .state('app', {
    //cache: false,
    url: '/app',
    abstract: true,
    templateUrl: 'templates/menu.html',
    controller: 'LogCtrl'
  })

  .state('app.home', {
    url: '/inicio',
    views: {
      'menuContent': {
        templateUrl: 'templates/inicio.html'
      }
    }
  })

  .state('app.homeFb', {
    url: '/inicio-loginFb',
    views: {
      'menuContent': {
        templateUrl: 'templates/inicio_loginFb.html'
      }
    }
  })

  .state('app.homeGg', {
    url: '/inicio-loginGg',
    views: {
      'menuContent': {
        templateUrl: 'templates/inicio_loginGg.html'
      }
    }
  })

// ..more routes
..

// if none of the above states are matched, use this as the fallback
  $urlRouterProvider.otherwise('/app/inicio');
});
```

Y su ruta es la siguiente `CanChamApp/www/js/app.js`

**Archivo: controllers.js**
Al mostrar una página de la aplicación móvil, se llama como mencionamos anteriormente a un controlador, entonces el controlador envía los datos necesarios a la plantilla a través de la variable `$scope`, que es un objeto que contiene los datos definidos por el controlador usado para construir la vista. Este archivo tiene la siguiente estructura

```
.controller('LogCtrl', function($scope, $state, $q, UserService,
$ionicActionSheet, $ionicLoading, $ionicModal, $timeout, $rootScope,
$location, $http, $ionicHistory) {

…

// Triggered in the login modal to close it
  $scope.closeLogin = function() {
    $scope.modal.hide();
  };

  // Open the login modal
  $scope.login = function() {
    $scope.modal.show();
    //$state.go('app.loginmail');
  };

$scope.data = {};

  $scope.doLogin = function() {
      LoginService.loginUser($scope.data.username,
$scope.data.password).success(function(data) {
          $state.go('tab.dash');
      }).error(function(data) {
          var alertPopup = $ionicPopup.alert({
              title: 'Login failed!',
              template: 'Please check your credentials!'
          });
      });
  }

// ..more controllers
..
```

Y su ruta es la siguiente `CanChamApp/www/js/controllers.js`

**Archivo: services.js**
Conocidos como Servicios o Factories, es la capa de datos de la aplicación, proporcionándolos desde un almacenamiento local o desde un servicio externo. Y los factories son más versátiles, ya que pueden devolver lo que queramos, ejecutando la función que los define.
Este archivo tiene la siguiente estructura

```
angular.module('starter.services', [])


.service('UserService', function() {

//for the purpose of this example I will store user data on ionic local
storage but you should save it on a database

  var setUser = function(user_data) {
    window.localStorage.starter_facebook_user =
JSON.stringify(user_data);
  };

  var getUser = function(){
    return JSON.parse(window.localStorage.starter_facebook_user || '{}');
  };

// ..more code

.factory('Request', function($http) {

         var getJSON = '';

         return {
         getInfo: function() {
         return $http.jsonp(getJSON);
         }
         };

})

.factory('Events', function($q, $ionicPlatform, $cordovaCalendar, $http,
$ionicLoading) {

  var incrementDate = function (date, amount) {
      var tmpDate = new Date(date);
      tmpDate.setDate(tmpDate.getDate() + amount)
      return tmpDate;
  };

  var incrementHour = function(date, amount) {
    var tmpDate = new Date(date);
    tmpDate.setHours(tmpDate.getHours() + amount);
    return tmpDate;
  };

// ..more code
```

Y su ruta es la siguiente `CanChamApp/www/js/services.js`