

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Desarrollo de aplicaciones para dispositivos móviles

II semestre 2014



## Especificación de Diseño

**Estudiantes:**

Jorge Bolaños Solís

Jorge Carvajal Araya

Kenneth Paniagua Díaz

**Profesor:**

Ing. Andrei Fuentes Leiva

13/10/2014

## Tabla de contenido

Propósito .....	3
Requerimientos funcionales .....	3
Gestionar usuario: .....	3
Gestionar estante: .....	3
Agregar libros: .....	3
Ver perfil de libros: .....	3
Buscador de libros: .....	4
Sección de lo más leído: .....	4
Agregar opinión: .....	4
Buscar librerías cercanas: .....	4
Requerimientos funcionales pendientes de revisión .....	5
Requerimientos no funcionales .....	5
Software: .....	5
Hardware: .....	5
Seguridad: .....	5
Descripción de los servicios web .....	6
Métodos del backend: .....	6
UX, UI .....	9
Descripción de alto nivel .....	12
Descripción detallada .....	13
Diagramas de flujo .....	14
Ingresar libro .....	14
Búsqueda sencilla de libro .....	15
Problemas de diseño .....	<b>Error! Bookmark not defined.</b>
Interacción con los sistemas externos .....	<b>Error! Bookmark not defined.</b>
Referencias .....	<b>Error! Bookmark not defined.</b>

## Propósito

### Requerimientos funcionales

#### Gestionar usuario:

El usuario se registra e identifica utilizando el API de Facebook. Una vez que el usuario se identifica, todas sus preferencias se sincronizan en el teléfono según la información que se encuentra en el servidor.

#### Gestionar estante:

Sección de la aplicación donde se muestran los libros favoritos del usuario. En este el usuario puede especificar si alguno de los libros agregados ya ha sido leído y aquellos pendientes. El estante solo puede ser accedido por su dueño. Cada vez que un libro es agregado al estante o es marcado como leído, al usuario se le brinda la opción de compartir en Facebook, para hacerle saber a todos del nuevo logro.

#### Agregar libros:

Al presionar la opción de agregar un nuevo libro, al usuario se le muestra una pantalla en la que se deben llenar los campos: título, cantidad de páginas, autor, fecha de publicación, género, imagen de portada y una previsualización del libro utilizando el API "Google Books".

#### Ver perfil de libros:

Cada libro cuenta con un perfil que contiene: la información ingresada durante su creación, los comentarios que han agregado los lectores del libro, una puntuación calculada como el promedio de las puntuaciones brindadas por los lectores del libro.

El perfil cuenta con el botón recomendar libro, que permite publicar en el muro de Facebook una recomendación de algún libro que puede ser de utilidad para algún amigo.

### Buscador de libros:

Para encontrar aquellos libros en que el usuario está interesado la aplicación cuenta con un buscador que cuenta con el mecanismo de autocompletado y tiene un alcance a nivel de título, género y autor. Además se brindan filtros de la calificación mínima y el año de publicación.

Una vez especificados los parámetros de búsqueda, se despliegan los resultados en una lista ordenados según: el título (alfabéticamente), la calificación (de mayor a menor) o según el año de publicación (más recientes primero). Al presionar alguno de ellos se muestra la sección con el perfil del libro seleccionado.

### Sección de lo más leído:

La sección lo más leído es parte de la sección de búsqueda y se muestra como un carrusel de imágenes de las portadas de los libros más leídos. El usuario puede desplazarse hacia la izquierda o la derecha; al encontrar algún libro de interés, se puede presionar la imagen para ser redirigido al perfil del libro.

### Agregar opinión:

Al ingresar al perfil de un libro, el usuario puede dar la opinión sobre el libro, esta consta de una puntuación de 0 a 5 y un comentario que es opcional. A partir de todas las puntuaciones de los usuarios se calcula el promedio que funciona como puntuación global del libro.

### Buscar librerías cercanas:

Al ingresar a esta sección al usuario se le presenta un mapa que muestra la ubicación del usuario y pines que indican la posición de librerías cercanas, donde el usuario podría adquirir sus libros.

## Requerimientos funcionales pendientes de revisión

- Búsqueda de librerías para buscar libros
- Decir dónde se puede conseguir libro

## Requerimientos no funcionales

### Software:

El backend de la aplicación será alojado en heroku, en una instancia de PostgreSQL. Las tecnologías a utilizar Ruby versión 2.13 en conjunto con el framework Rails versión 4.1.6. Las funcionalidades del backend pueden ser accedidas desde <http://www.geamapps.com/>

El frontend será desarrollado de forma nativa para el sistema operativo Android, el API mínimo para utilizar la aplicación es el 11.

### Hardware:

El desarrollo de la aplicación es multiplataforma. El hardware del dispositivo móvil debe poseer Android, iOS o WindowsMobile y GPS. Es necesaria la conexión a internet para utilizar la aplicación.

### Seguridad:

Las contraseñas de los administradores y operadores en la aplicación en el servidor serán encriptadas utilizando el algoritmo de encriptación por llave RSA.

Para el acceso a los servicios web se comprobará que estos son accedidos por usuarios existentes enviado como parámetro el identificador del usuario

## Descripción de los servicios web

### Métodos del backend:

#### registrar

- Descripción: Método que recibe el número de identificación de un usuario. El sistema verifica que no se haya registrado antes y realiza el registro en la base de datos de ese usuario.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long
- Método HTTP: POST

#### identificar

- Descripción: Método que recibe el número de identificación de un usuario. El sistema verifica que el usuario se haya registrado antes.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long
- Método HTTP: GET

#### agregarAEstante

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe el número de identificación del libro que se quiere agregar al estante. El sistema realiza la relación en la base de datos de libros en el estante.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, pIDLibro de tipo int
- Método HTTP: POST

#### marcarLibroFavorito

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe el número de identificación del libro que se quiere marcar como favorito. El sistema realiza la modificación en la base de datos de libros del estante.

- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, pIDLibro de tipo int
- Método HTTP: POST

#### marcarLibroLeido

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe el número de identificación del libro que se quiere marcar como leído. El sistema realiza la modificación en la base de datos de libros del estante.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, pIDLibro de tipo int
- Método HTTP: POST

#### agregarLibro

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe los datos de título, cantidad de páginas, autor, fecha de publicación, género, imagen de portada y una pre vista del libro utilizando el API “Google Books”. El sistema realiza el registro del libro en la base de datos.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, título, pCantidadPaginas de tipo int, pIdAutor de tipo in, pFechaPublicación de tipo string, pIdGenero de tipo int, plimagenPortada de tipo bytes64 y pIdGoogleBooks de tipo long.
- Método HTTP: POST

#### verPerfilLibro

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe el identificador de un libro para hacer la consulta a la base de datos. Se retorna un string con formato JSon.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, pIDlibro de tipo int.
- Método HTTP: GET

#### buscarLibro

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario y recibe los criterios de búsqueda en un string con formato JSon.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, pCriterios de tipo string.
- Método HTTP: GET

#### obtenerLoMasLeido

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario. Devuelve un string tipo JSon con la información de los libros más leídos.
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long.
- Método HTTP: GET

#### evaluar

- Descripción: Método que recibe el número de identificación de un usuario para verificar la autenticidad del usuario. Toma los datos de evaluación y el comentario opcional, los inserta en la base de datos y retorna un mensaje de éxito (Un boolean true para confirmar éxito, cualquier otra respuesta representa fracaso)
- URL: <http://www.geamapps.com/service.asmx?WSDL>
- Parámetros: pIDUsuario de tipo long, EvaluacionLibro de tipo int, Comentario de tipo string, pCodigoLibro de tipo string
- Método HTTP: POST



## UX, UI



### Funcionalidades

- Identificación con facebook.

### Patrones de diseño

- Formulario de ingreso: La identificación y el registro de usuarios se realizará con el API de facebook. Para registrar un nuevo usuario se le pedirá únicamente un nickname.

### Funcionalidades

- Indicar si ha leído un libro o no.
- Libros favoritos.
- Compartir en facebook.

### Patrones de diseño

- Despliegue de datos: Se utilizará el patrón tablas sin encabezado junto con el master-detail para desplegar la colección de libros del usuario.
- Herramientas contextuales: Al hacer click en el ícono con la fecha hacia abajo se desplegará un menú con las opciones eliminar y marcar como favorito.





### Funcionalidades

- Buscador de libros.
- Recomendación de libros.
- Sección "Lo más leído".

### Patrones de diseño

- Búsqueda: Autocompletado, con alcances (género y autor) y recomendada.
- Ordenamiento: Superior alfabético, por calificación (más alta) y año de publicación (más reciente).
- Filtros: Calificación mínima y año de publicación mínimo.
- Navegación: tab bar (primaria) y carrusel (secundaria).

### Funcionalidades

- Agregar libros (¿Dónde se puede conseguir?).

### Patrones de diseño

- Formulario en varios pasos: En la primer pantalla se pedirá la información básica del libro (ver wireframe) y en la segunda la ubicación de la librería donde se puede conseguir.





### Funcionalidades

- Agregar evaluaciones.

### Patrones de diseño

### Funcionalidades

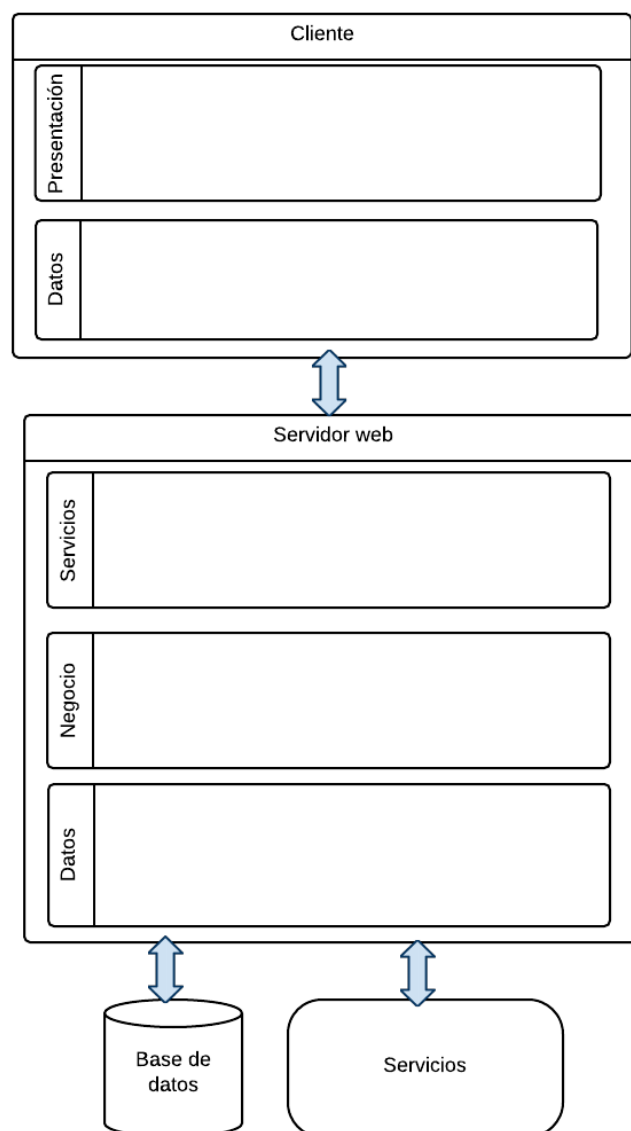
- Ver perfil de libros.

### Patrones de diseño



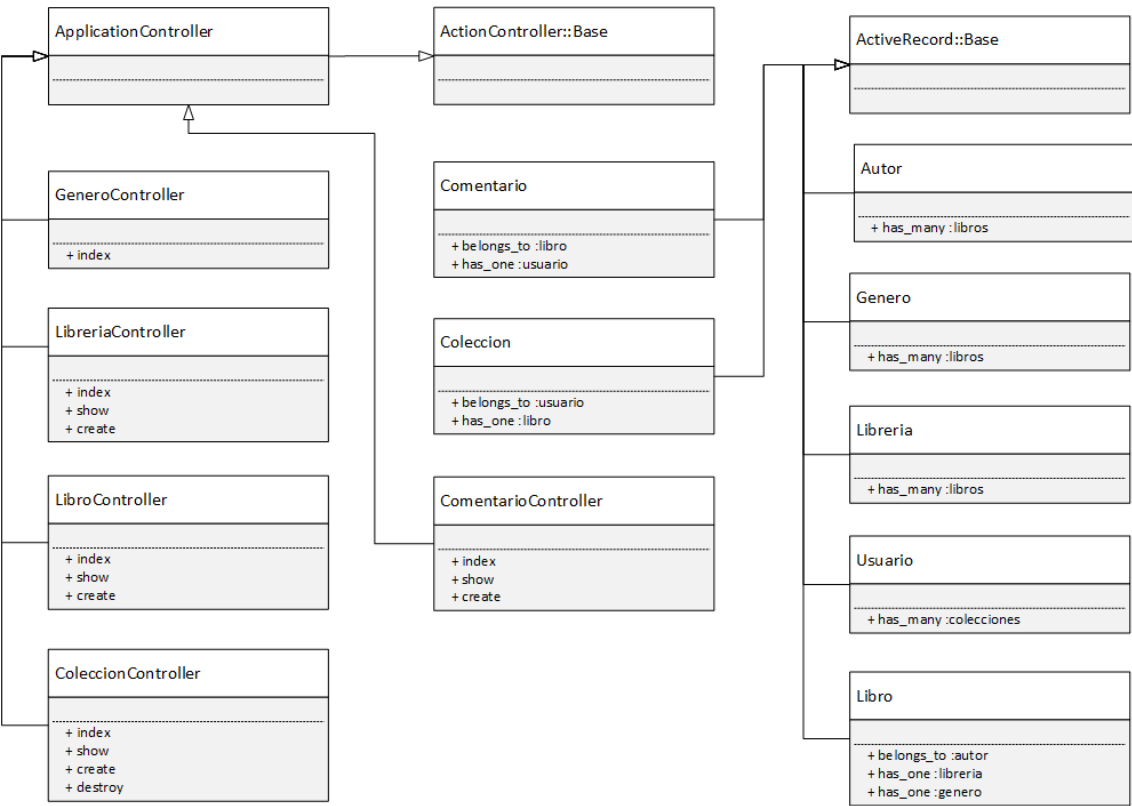
## Descripción de alto nivel

La arquitectura elegida para la aplicación DigiBiblioteca es rich internet application (RIA), debido a que por el lado del cliente sólo se requiere que la aplicación se comunique con los servicios proporcionados por el servidor y los apis externos para luego presentar la información. Por tanto, la implementación de la lógica de negocio, así como los servicios y datos serán implementados del lado del servidor. La interacción entre las capas será estricta; es decir, una capa se puede comunicar únicamente con la capa inmediatamente inferior a ella.

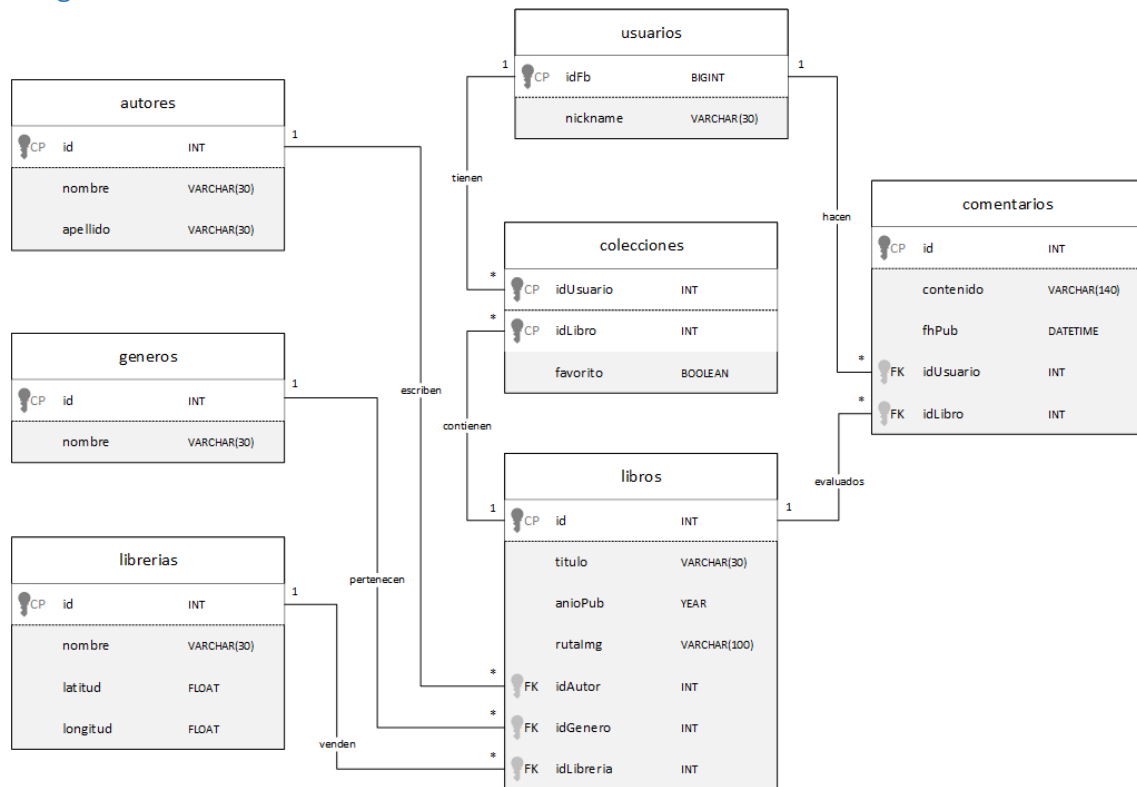


# Descripción detallada

## Diagramas de clase

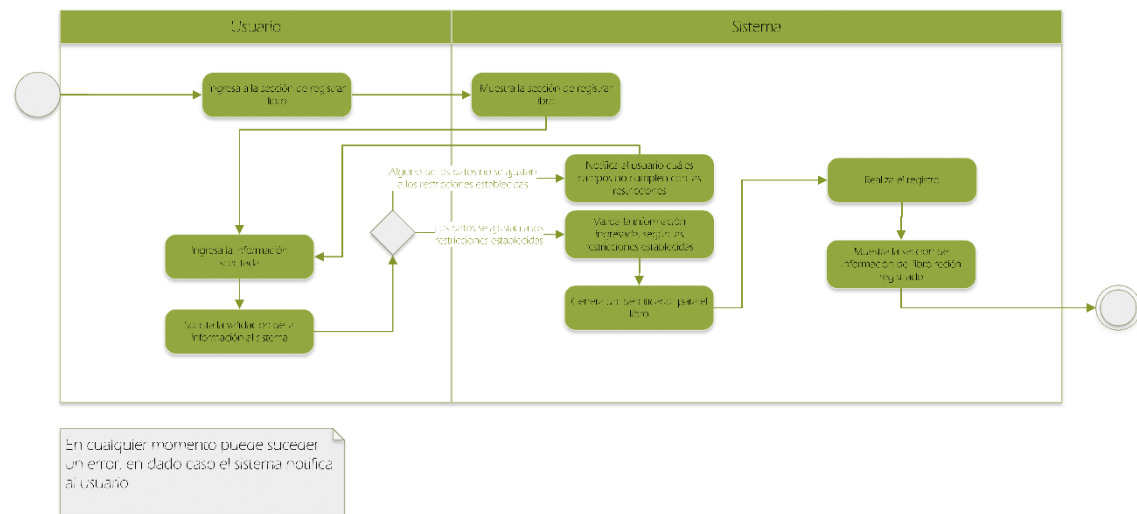


## Diagramas de base de datos

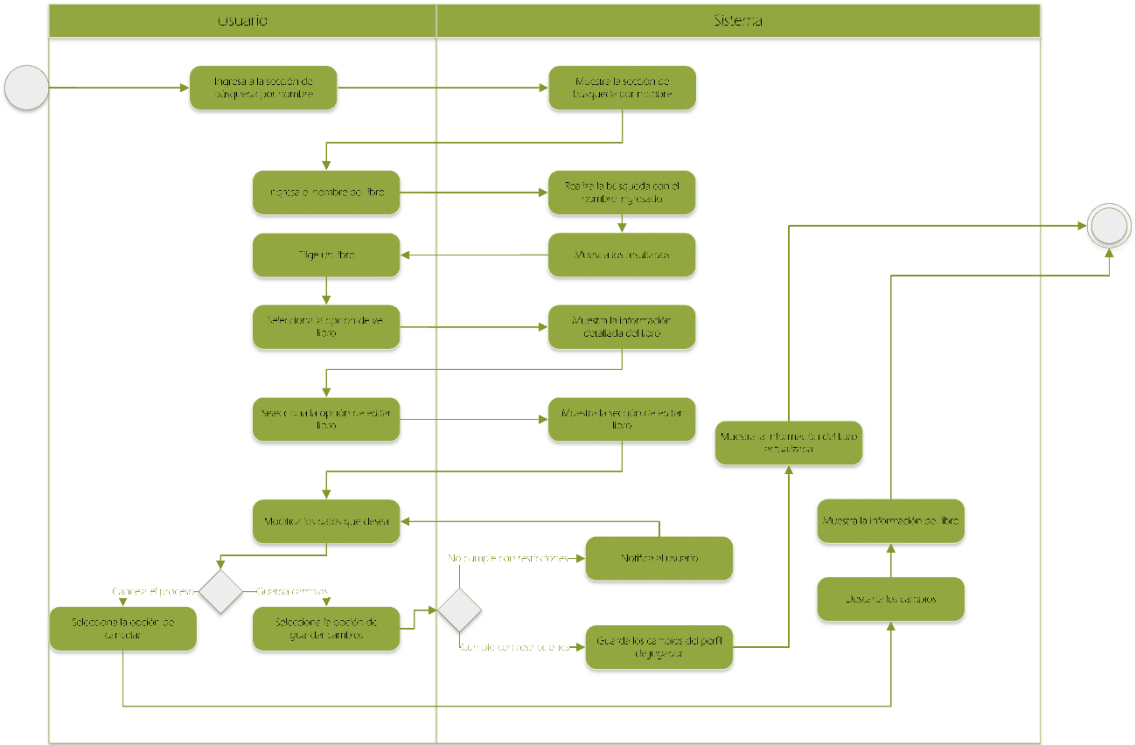


## Diagramas de flujo

### Ingresar libro



Búsqueda sencilla de libro



en cualquier momento puede suceder un error, en dicho caso el sistema notifica al usuario