

A dark blue vertical bar runs down the left side of the page. A blue arrow-shaped graphic points to the right, containing the date. In the bottom left corner, there are several thin, curved lines in dark blue and light grey.

12-10-2014

Instituto Tecnológico de Costa Rica

Documento de diseño de “Prueba Manejo
Móvil”

Ney Rojas Jiménez
Gustavo Vargas Vargas
Manuel Arguedas Sandí

Contenido

Descripción del app	2
Funcionalidades	3
Requerimientos funcionales	3
Usuario:.....	3
Administrador:.....	4
Requerimientos no funcionales	4
UX/UI	5
Descripción de diseño de alto nivel	17
Diagrama de arquitectura	17
Explicación del diagrama de arquitectura	17
Patrones de diseño utilizados	18
• Dos patrones de despliegue de datos	18
• Tres patrones de diseño para búsqueda	18
• Un patrón de diseño de filtro y uno de ordenamiento	18
• Dos patrones de diseño para navegación: primaria y secundaria	18
• Dos patrones de diseño para formularios	18
• Un patrón de herramientas.....	18
Descripción detallada del diseño	19
Diagrama de clases	19
Diagrama de base de datos	20
Descripción de los web services.....	21
URL	21
Métodos HTTP específicos para la aplicación móvil	21
GET	21
POST.....	21
Métodos HTTP específicos para la aplicación web	22
GET	22
POST.....	22
Métodos HTTP específicos compartidos.....	23
GET	23
Interacción con sistemas externos	23
Hardware a utilizar	23

Descripción del app

Para una persona, es importante realizar los trámites para obtener su licencia de conducir, el problema es que muchas veces, la parte teórica suele ser complicada ya que no existe ningún método para estudiar interactivamente. Por lo que muchas veces es difícil para las personas que no están acostumbradas a leer manuales y aprender de ellos, estudiar los distintos conceptos y tenerlos presentes.

De este modo surge la idea de implementar "Prueba manejo móvil" para ayudar a que las personas a aprender y practicar para su prueba teórica de manejo. Esta aplicación ofrece distintas herramientas para promover los distintos tipos de aprendizaje (sea visual, por descubrimiento o quinesésico).

Funcionalidades

Requerimientos funcionales

Usuario:

1. Test teórico rápido: En esta funcionalidad el usuario podrá realizar una prueba teórica que consta de 40 preguntas que son elegidas de forma aleatoria, el usuario contestará una a la vez seleccionando la respuesta correcta de una lista de tres posibles opciones. Al finalizar la prueba se presenta el resultado de preguntas correctas y será almacenado en el historial del usuario.
2. Test dinámico de manejo: En esta funcionalidad el usuario realizará un test de manejo dinámico, utilizando su celular como si fuera un volante. Además se presentan dos botones en pantalla que simulan el acelerador y el freno del carro, en este test se evalúan las habilidades del usuario para interpretar señales visuales como de viraje, alto, ceda, reducción de velocidad, curva peligrosa, etc. Este test cuenta con 20 preguntas seleccionadas de forma aleatoria y de igual manera el resultado de la evaluación será almacenado en el historial.
3. Consultar tema del manual dentro de una pregunta: Cuando el usuario este realizando el test y no sabe la respuesta de la pregunta, pero desea conocer en que sección del manual se encuentra, podrá seleccionar "ver sección del manual" que muestra la información referente a esa pregunta.
4. Login Facebook o Twitter: Los usuarios tendrán la posibilidad de registrarse utilizando estas redes sociales.
5. Lista de lugares para realizar la prueba teórica: Dentro de la aplicación el usuario podrá consultar la lista de lugares donde puede realizar la prueba teórica de manejo y la información básica de teléfonos, nombre y dirección.
6. Búsqueda de lugares por texto clave: También el usuario podrá colocar una palabra clave para buscar un lugar en específico como nombre del lugar, entre otros, y se le mostrará la lista de lugares que tiene coincidencia con esta información.
7. Mostrar ubicación de lugares para prueba teórica por GoogleMaps: Cuando el usuario consulta un lugar puede ver la ubicación del mismo utilizando GoogleMaps.
8. Mostrar lista de temas dentro de manual: Se le desplegará al usuario la lista de los temas del manual de manejo para que consulte uno en específico.
9. Búsqueda de una sección del manual usando una palabra clave: El usuario podrá realizar búsquedas de las secciones en el manual usando un texto clave.
10. Búsqueda de un tema utilizando una palabra clave: El usuario podrá realizar búsquedas de temas utilizando una palabra clave que se encuentre en el cuerpo de un tema en específico.
11. Historial de resultados de pruebas: En la opción de historial se mostrar los resultados de pruebas anteriores, donde el usuario observara su progreso de aprendizaje dentro de la aplicación. Se muestra el historial de las pruebas teóricas y dinámicas en una tabla.

12. Realizar sugerencia: El usuario puede realizar una sugerencia sobre como mejorar la aplicación, reportar un error en alguna pregunta, entre otros.

Administrador:

1. Agregar y eliminar lugares para realizar la prueba de manejo: El administrador tiene la opción de agregar o eliminar lugares donde se realizan las pruebas, así como su información y ubicación geográfica.
2. Agregar y eliminar manuales de manejo y contenido: El administrador podrá crear y agregarle contenido a los manuales de manejo de la aplicación, así como la posibilidad de borrar contenido de ellos. Los manuales se dividen por países.
3. Agregar y eliminar preguntas teóricas: Se podrán agregar nuevas preguntas con sus respuesta correcta e incorrectas. Además de eliminar la preguntas.
4. Agregar y eliminar preguntas dinámicas: Se podrá agregar o eliminar preguntas dinámicas. Cada pregunta cuenta con la interacción correcta que debe realizar el usuario.

Requerimientos no funcionales

1. Seguridad en la comunicación: la comunicación que realiza la aplicación con los App-Backend se realizan mediante el protocolo TCP/IP, para que esta comunicación sea más segura se utilizará una llave única de identificación privada, para que el servidor comprenda que es una conexión segura.
2. Formato de comunicación: El formato de comunicación que se utiliza para obtener los datos deseados por el usuario ante alguna consulta, es por medio de archivos JSON.
3. Comunicación asincrónica: La comunicación de la aplicación con el servidor es asincrónica, por lo que la aplicación únicamente se sincroniza cuando el usuario solicita cargar una sección específica.
4. Android: la aplicación es diseñada para que pueda funcionar en el sistema operativo Android 4.1 Jelly Bean.
5. Lenguaje de la aplicación: El lenguaje principal de la aplicación es el español, para un futuro se implementará el inglés para obtener más mercado.
6. Soporte: se proporcionará soporte a la aplicación cada 3 meses, con el fin de mantener mejoras en la aplicación que serán beneficiosas para los usuarios de la aplicación.

UX/UI

Se presentarán las siguientes pantallas:



Figura 1. Ventana de entrada



Figura 2. Ventana principal de la aplicación



Figura 3. Ventana de inicio del test teórico



Figura 4. Ventana del test teórico cuando está en una pregunta



Figura 5. Ventana de inicio del test dinámico



Figura 6. Ventana del test dinámico cuando está en una pregunta



Figura 7. Consultar manual por texto



Figura 8. Consultar manual por temas



Figura 9. Consultar manual por sección

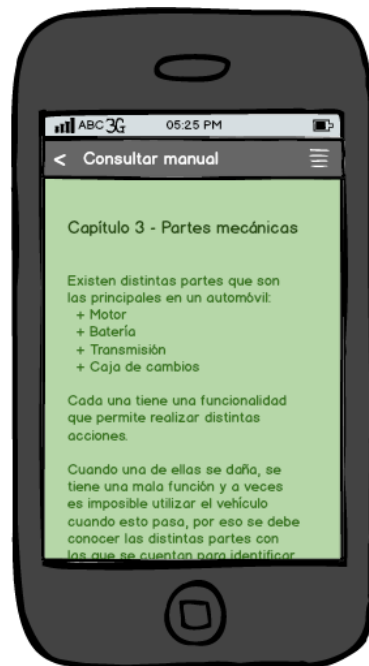


Figura 10. Ventana que muestra la información del manual



Figura 11. Lista de lugares donde se pueden realizar las pruebas



Figura 12. Búsqueda de algún lugar para realizar la prueba



Figura 13. Ventana que muestra un lugar en específico y su información

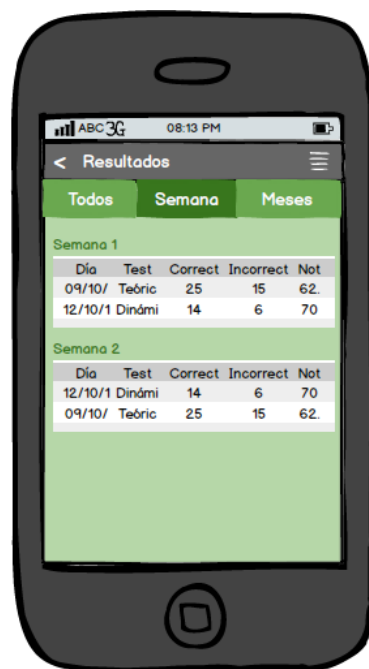


Figura 14. Ventana que muestra el historial de resultados (aquí filtrados por semana)



Figura 15. Ventana que muestra ¿cómo se realiza una sugerencia?



Figura 16. Ventana que muestra el funcionamiento del menú de opciones

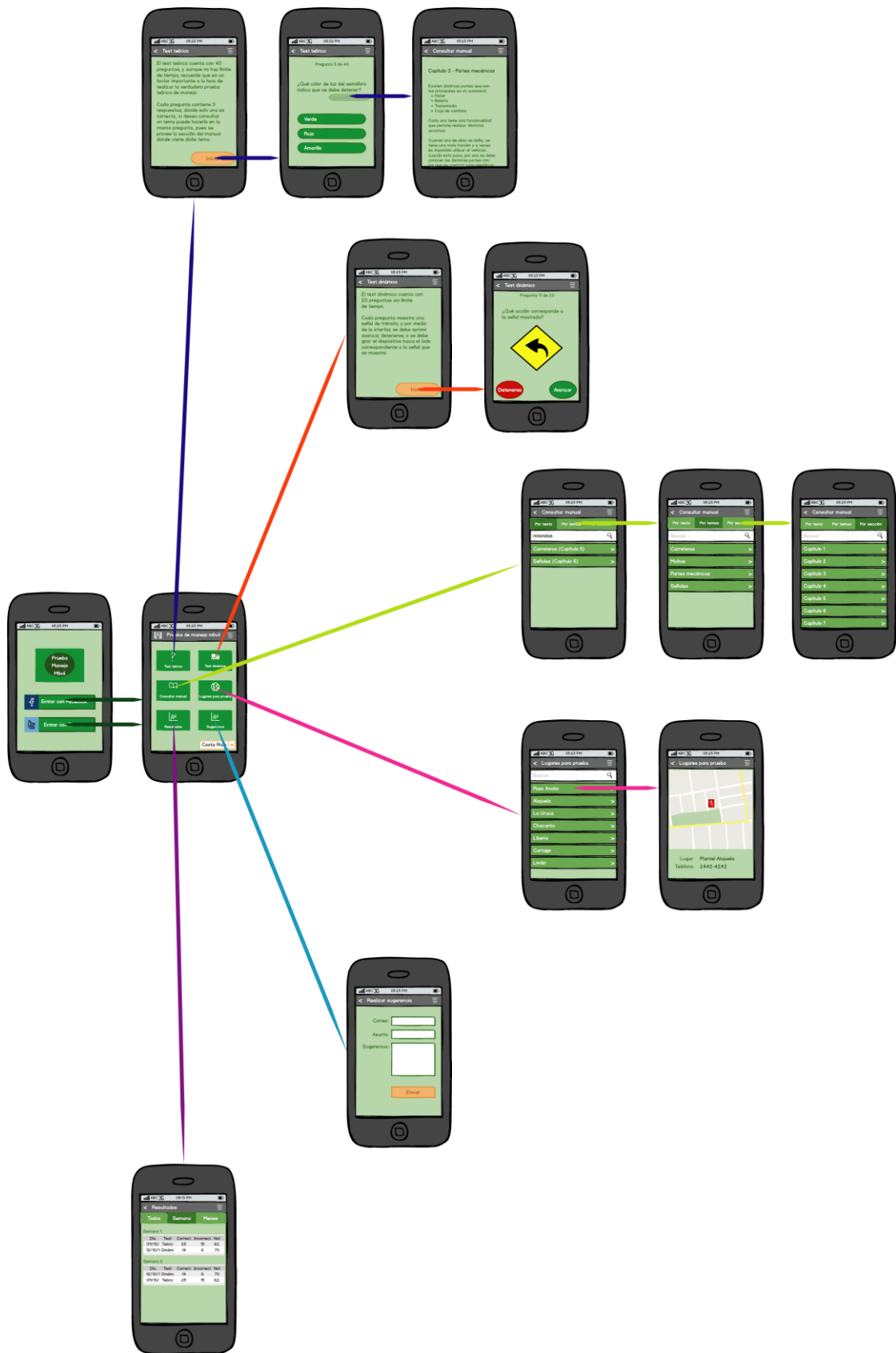


Figura 17. Diagrama de navegación de la aplicación.

A continuación se muestran los mockups de la aplicación web:

The mockup shows a web browser window titled "A Web Page". The address bar contains "http://". The main content area features a large square placeholder with a diagonal 'X' on the left. To the right of the placeholder is the heading "Iniciar sesión". Below the heading are two input fields: "Nombre de usuario" and "Contraseña".

Figura 18. Diagrama del inicio de sesión de la aplicación web.

The mockup shows a web browser window titled "A Web Page". The address bar contains "http://". The main content area is divided into a sidebar on the left and a main panel on the right. The sidebar contains five buttons: "Manuales", "Secciones", "Subsecciones", "Sucursales", and "Salir". The main panel has the heading "Manuales". Below the heading is the section "Manuales existentes", which includes a dropdown menu showing "Costa Rica" and an "Eliminar" button. Below this is the section "Agregar Manual", which includes a "Nombre:" label, an input field, a "País:" label, a dropdown menu showing "Costa Rica", and a "Crear" button.

Figura 19. Diagrama de la sección de manuales de la aplicación web

A Web Page

http://

Manuales

Secciones

Subsecciones

Sucursales

Salir

Secciones

Secciones existentes

Capítulo 1 ▼ Eliminar

Agregar Seccion

Nombre:

Manual: Costa Rica ▼

Crear

Figura 20. Diagrama de la sección de la aplicación web para eliminar o agregar secciones al manual de manejo.

A Web Page

http://

Manuales

Secciones

Subsecciones

Sucursales

Salir

Subsecciones

Subsecciones existentes

Motores ▼ Eliminar

Agregar subseccion

Nombre:

Descripción:

Crear

Figura 20. Diagrama de la sección de la aplicación web para eliminar o agregar subsecciones al manual de manejo.

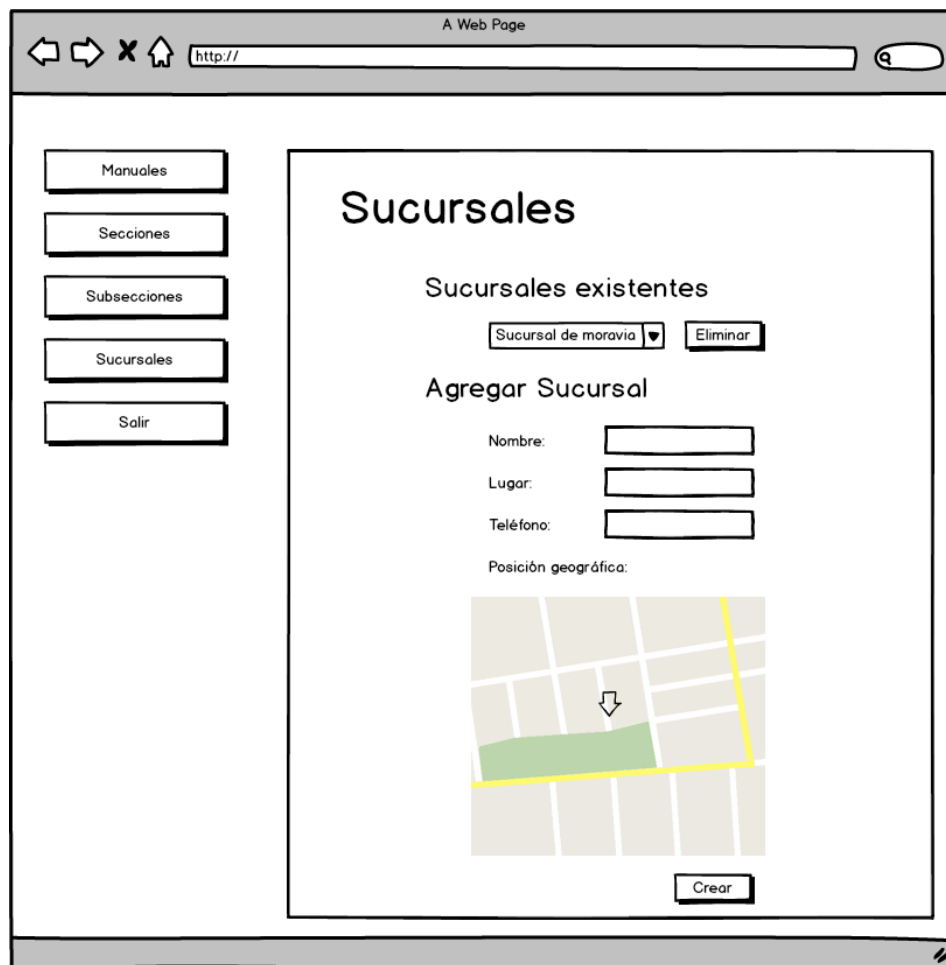


Figura 20. Diagrama de la sección de la aplicación web para eliminar o agregar sucursales.

Descripción de diseño de alto nivel

En la siguiente sección se describe detalladamente la arquitectura de la aplicación:

Diagrama de arquitectura

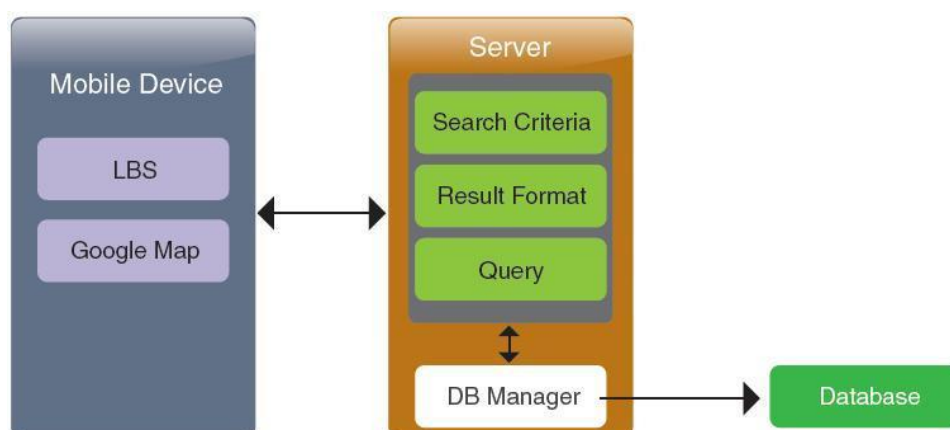


Figura 18. Diagrama general de la arquitectura utilizada en la aplicación.

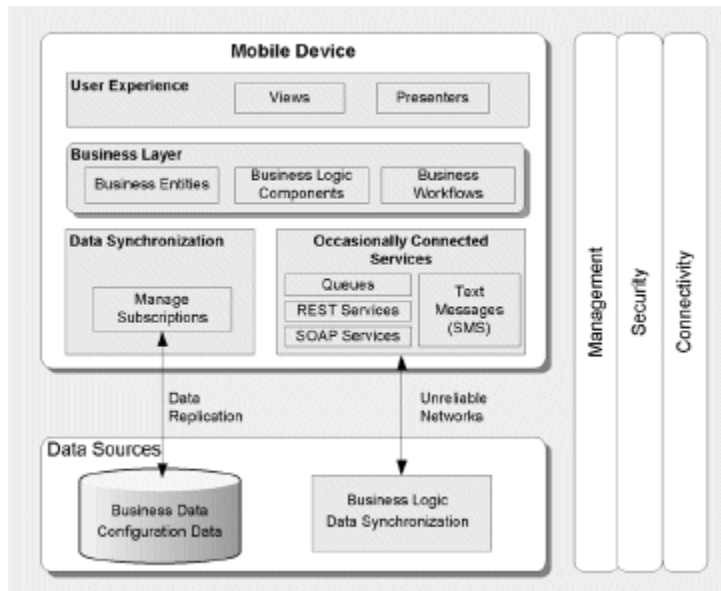


Figura 19. Diagrama específico de la arquitectura Mobile.

Explicación del diagrama de arquitectura

La arquitectura que se propone utilizar es la Mobile, debido a que la aplicación maneja lógica en el servidor y en el dispositivo móvil. El servidor almacena toda la información de los manuales de manejo y realiza búsquedas de las preguntas de forma aleatoria. También el

dispositivo móvil tiene la lógica de controlar dispositivos de hardware y almacenar datos al momento de solicitar un lote de preguntas.

Patrones de diseño utilizados

• Dos patrones de despliegue de datos

- Tablas sencillas: Los resultados de las pruebas teóricas y dinámicas, son mostradas en forma de tablas.
- Master-detail: Este patrón será aplicado en la parte de mostrar la lista de las secciones del manual y otras secciones donde sea necesario desplegar grandes conjuntos de datos.

• Tres patrones de diseño para búsqueda

- Búsqueda explícita: Esta búsqueda es utilizada para buscar una sucursal donde hacer el example teórico.
- Búsqueda con autocompletado: se ofrece la opción de autocompletado, para facilitar la búsqueda a los usuarios.
- Búsqueda con alcances: cuando se busca algo en el manual, se pueden realizar los alcances por sección o por temas.

• Un patrón de diseño de filtro y uno de ordenamiento

- Filtro en pantalla: En la parte de historial, se puede filtrar el contenido por semana, por mes o todos.
- Ordenamiento: Los temas y las secciones del manual se ordenan por orden alfabética

• Dos patrones de diseño para navegación: primaria y secundaria

- Springboard: En la sección principal de la aplicación se muestran las opciones predefinidas que tiene la aplicación.
- Secundaria: Cada opción mostrada en el springboard tiene su distinta funcionalidad y elementos diferentes dentro de ellas.

• Dos patrones de diseño para formularios

- Formulario de ingreso: La aplicación contará con opciones sencillas para registrarse con Facebook y Twitter.
- Formulario de sugerencia: La aplicación cuenta con una sección donde se puede hacer una sugerencia. La sugerencia recibe los siguientes datos: correo, el asunto y la descripción de la sugerencia o pregunta.

• Un patrón de herramientas

- Menú de opciones: Aparte del menú principal de opciones, se maneja un menú secundario de navegación que se activa con el botón de opciones de Android.

Descripción detallada del diseño

Diagrama de clases

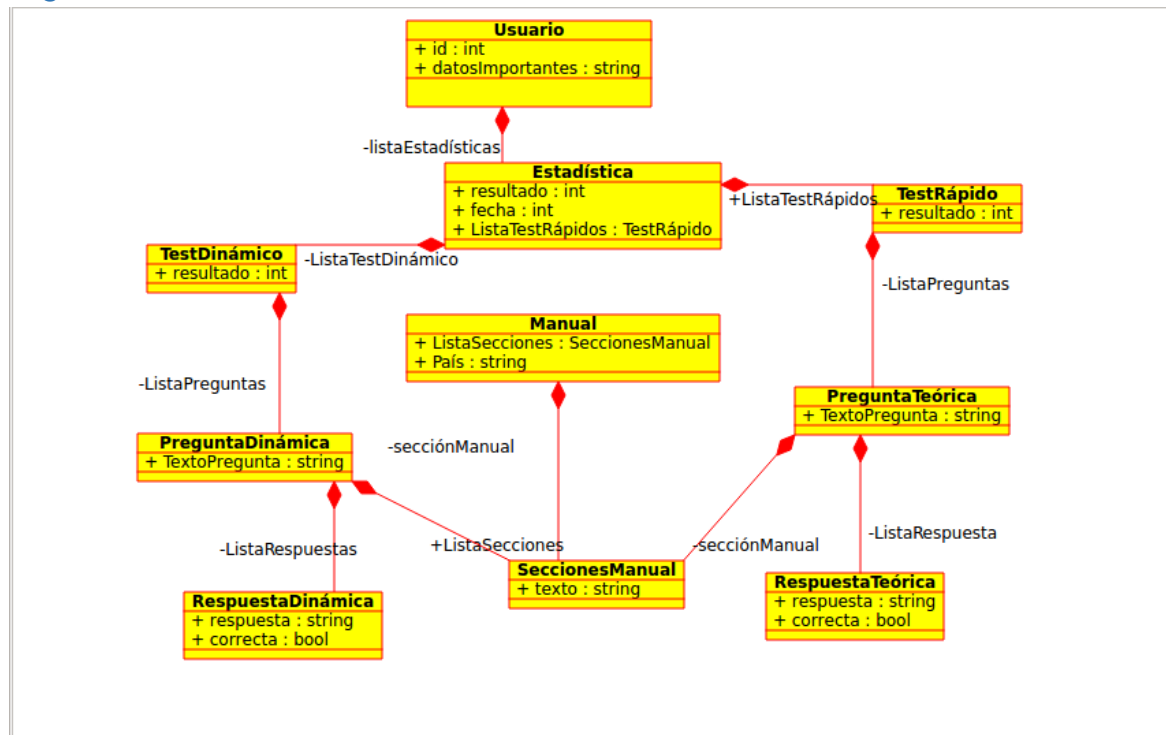


Figura 19. Diagrama dominio de la aplicación

Diagrama de base de datos

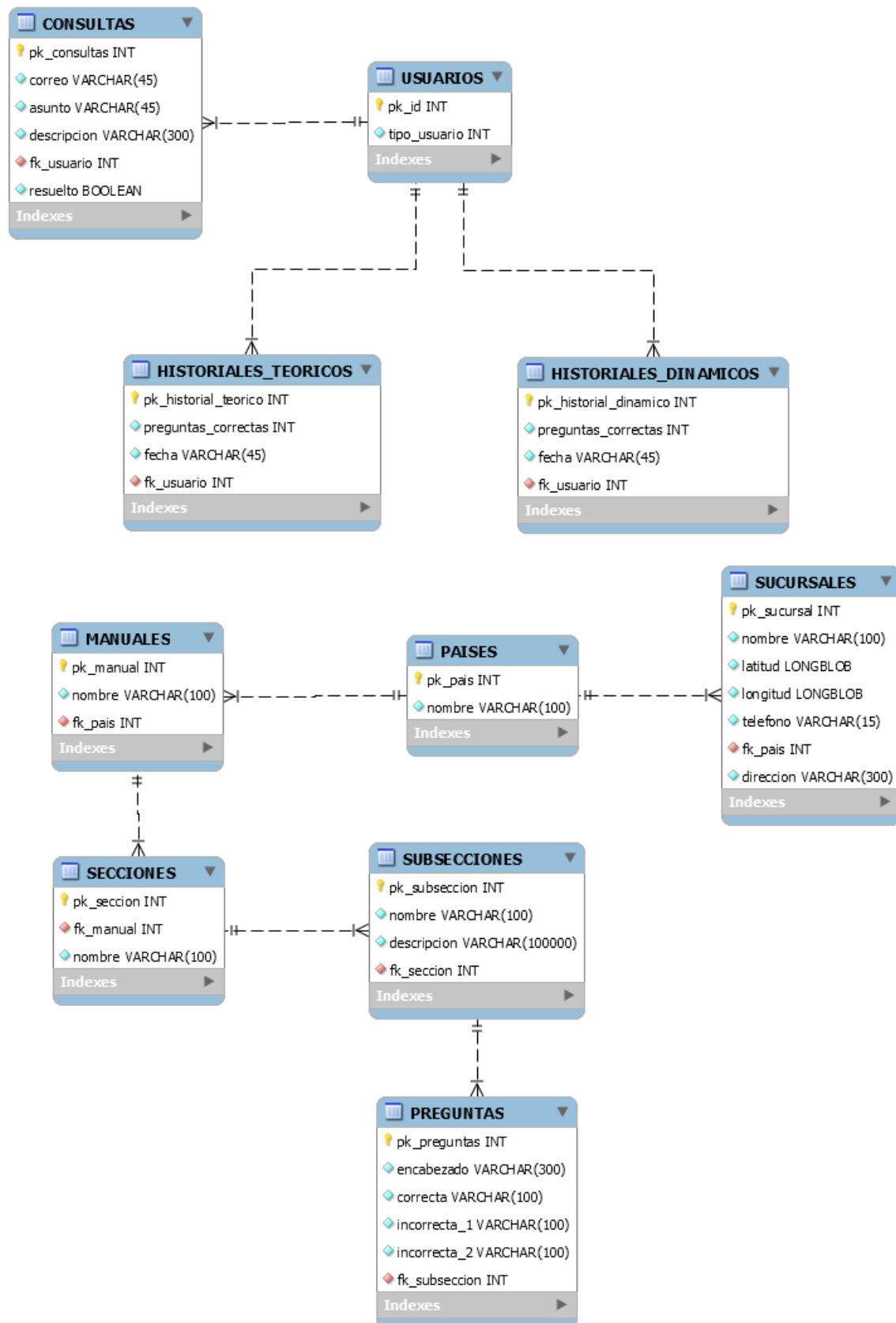


Figura 20. Diagrama de bases de datos para Prueba Manejo Móvil.

Descripción de los web services

Los web services serán implementados utilizando la tecnología de NodeJS, que utiliza un lenguaje de programación en Javascript. Cuando la aplicación móvil intente comunicarse con el servidor, primero tendrá que identificarse utilizando un usuario válido de Facebook o Twitter, de lo contrario se le niega el acceso a la información web. La comunicación con el servidor se hace por medio de archivos en formato JSON.

La aplicación cuenta con una página web, que es utilizada para agregar nuevo contenido a la aplicación, como por ejemplo nuevas preguntas o bien información del manual de manejo. Para poder ingresar a la página web se debe de autenticar con un usuario administrador único.

URL

El dominio fue obtenido de GoDaddy y la dirección reservada es la siguiente: "www.pruebamanejomovil.com".

Métodos HTTP específicos para la aplicación móvil

GET

- **ObtenerPreguntasTeoricas():** Este método lo que hace es devolver un json con la información de las cuarenta preguntas. En este json se encuentran las preguntas, las respuestas incorrectas y la respuesta correcta.
- **ObtenerPreguntasDinamicas():** Este método lo que hace es devolver un json con la información de las veinte preguntas del examen dinámico. En este json se encuentran las preguntas y la respuesta correcta, así como la dirección de la imagen que se muestra en cada pregunta.
- **IdentificarseAplicacion(int idUsuario):** Este método verifica que el usuario que está intentando ingresar, ya se encuentre registrado en la base de datos.
- **GetHistorial(int idUsuario, int Filtrado):** Método que devuelve la información de todo el historial de la persona, usando un formato de json. El filtrado es utilizado para seleccionar el tipo de filtrado para traer todo, el de hace un mes o solamente el del día actual.

POST

- **RegistrarseRedes(int metodoRegistro, int idUsuario):** Registra el usuario usando una red social (Facebook o Twitter). Método de registro lo que hace es decir cual red social se utilizó.
- **EscribirResultadoPruebaTeorica(json informacion):** En el json que se envía, contiene la información de quien es la persona, la cantidad de preguntas correctas, la hora y la fecha. Luego agrega al historial de las pruebas teóricas, la información enviada de la aplicación.

- `EscribirResultadoPruebaDinamica(json informacion)`: En el json que se envía, contiene la información de quien es la persona, la cantidad de preguntas correctas, la hora y la fecha. Luego agrega al historial de las pruebas dinámicas, la información enviada de la aplicación.
- `EscribirSugerencia(json informacion)`: El json que recibe este método es usado para obtener la información que se envió desde la aplicación. Contiene el correo de la persona, el asunto, identificador del usuario y la descripción del mismo.

Métodos HTTP específicos para la aplicación web

GET

- `IdentificarseWeb(string nombreUsuario, string Contraseña)`: Para ingresar a la página web de la aplicación, primero se tiene que identificar con un nombre de usuario y contraseña únicos.

POST

- `CrearManual(int idPais, string nombreManual)`: método que se encarga de crear un nuevo manual en el servidor, dependiendo del país que se seleccionó. No puede haber más de un manual por país.
- `CrearSeccion(string nombreSeccion, int idManual)`: Método que crea una sección en el servidor. Se tiene que especificar el id del manual a la cual va a pertenecer.
- `CrearSubseccion(json informacion)`: Método que crea una subsección completa en el servidor. La información que se envía es el id de la sección, el nombre de la subsección y el texto de la misma.
- `CrearNuevaSucursal(json informacion)`: Método que agrega una nueva localización de sucursal al servidor. La información que almacena el json es la latitud, la longitud y el id de país.
- `CrearPreguntaTeorica(json informacion)`: Método que crea una pregunta teórica. La información que lleva el json es la descripción de la pregunta, la respuesta correcta, las dos incorrectas y el id de la subsección a la cual pertenece.
- `CrearPreguntaDinamica(json informacion)`: Método que crea una pregunta dinámica. La información que lleva el json es la imagen de la pregunta y el int de la respuesta correcta (la acción en el teléfono).

Métodos HTTP específicos compartidos

GET

- `ObtenerSecciones(int idPais)`: Este método devuelve un json con el nombre de todas las secciones del manual para el país del usuario registrado.
- `ObtenerSubSecciones(int idSeccion)`: Este método devuelve un json con el nombre de las subsecciones de una sección.
- `ObtenerSubseccionCompleta(int idSubseccion)`: Devuelve la información completa de una sección del manual.
- `ObtenerPaises()`: Devuelve un json con todos los países registrados en la base de datos del servidor.
- `ObtenerLugaresSucursales(int idPais)`: Devuelve un json con las posiciones de las sucursales registradas en un país.

Interacción con sistemas externos

Con respecto a sistemas externos, utilizaremos el API de Facebook y de GoogleMaps, estos se usan en distintas funcionalidades:

1. API Facebook: Facebook proporcionará el medio para acceder a la aplicación con el login de Facebook y obtener la información completa del usuario.
2. API Google Maps: se utiliza para señalar la posición del lugar donde se puede realizar las pruebas teóricas.
3. API Twitter: Twitter proporcionará el medio para acceder a la aplicación con el login de Twitter y obtener la información completa del usuario.

Hardware a utilizar

La aplicación utilizará el acelerómetro para contestar las preguntas dinámicas que ofrece una mejor experiencia al usuario, también se utilizará el vibrador para indicar en el test dinámico cuando se realiza una acción incorrecta (cuando falla el usuario).