

IC-8041 TP2
DOCUMENTACIÓN EXTERNA
LESLIE BECERRA
BRAULIO ALPIZAR
DAVID CHAVERRI

ITCR
APLICACIONES MÓVILES IC

Contenido

Resumen ejecutivo	2
Descripción del problema	3
Requerimientos de la herramienta	4
Funcionalidades de la aplicación móvil	4
Requerimientos no funcionales de la aplicación móvil.....	5
Funcionalidades del Backend	6
Requerimientos no funcionales del Backend	7
Descripción de la arquitectura	8
Descripción detallada	9
Diagrama de Clases.....	9
Diagrama de Componentes	10
Diagrama de Despliegue	11
Descripción de los Web Services	12
Interacción con sistemas externos	14
Wireframes.....	15
Descripción de pantallas.....	16
Patrones de diseño aplicados	17

Resumen ejecutivo

En el siguiente documento se presenta la documentación de la aplicación de manera general.

La aplicación consiste en una app tanto para la plataforma Android como IOS que le permite a los usuarios de un gimnasio realizar diversas funcionalidades como acceder a la información de progreso, obtener información nutricional acorde al programa de ejercicios, información acerca de la ejecución correcta, además recibir noticias acerca de promociones, eventos e información del gimnasio .

Esta App se desarrollará bajo el Framework Titanium ya que permite el desarrollo híbrido para las plataformas Android y IOS, además de proporcionar diversas herramientas entre ellas un SDK.

La arquitectura de la aplicación está compuesta por un Frontend y un Backend los cuales utilizan diversos patrones entre los cuales están: MVC, Cliente-Servidor, Thin Client.

A nivel del Backend algunas de las funciones que éste realizará para el funcionamiento de la aplicación son las siguientes: crear, eliminar y editar usuarios de tipo administrador e instructor, limitar al usuario público tener acceso únicamente a la siguiente información: Nombre del gimnasio, descripción, misión, visión, encargado, información de contacto (email y número de teléfono), horario de atención y la ubicación de las instalaciones...

Para obtener la información necesaria para el funcionamiento la aplicación y el Backend deberá interactuar con sistemas externos (API's) los cuales le proporcionarán la dicha información, los sistemas externos con los que interactúa son los siguientes: google maps y nutrición.

Descripción del problema

Esta aplicación se desarrolla a partir del problema que presentan los gimnasios en la actualidad a la hora de brindar el servicio, puesto normalmente los usuarios no pueden tener acceso de manera inmediata a la información acerca del progreso que tienen durante el tiempo que llevan entrenando, además una parte fundamental del entrenamiento sin importar la plan de trabajo que tengan es la alimentación que en muchos o la gran mayoría de gimnasios se deja por fuera y no existen planes o recomendaciones acerca de la alimentación correspondiente a los objetivos que se desean alcanzar.

Por otra parte un otro problema que poseen los gimnasios es que al ser muy alta la cantidad de personas que asisten a los instructores se les dificulta poder atender a tantas personas al mismo tiempo que tienen dudas acerca de la correcta ejecución de un determinado ejercicio.

Además de esto por el mismo modelo de negocio que poseen los gimnasios es difícil comunicar a sus clientes acerca de promociones, información de general, fecha de vencimiento de su membresía entre otras cosas.

Requerimientos de la herramienta

Se divide en los siguientes dos

Funcionalidades de la aplicación móvil

- La aplicación deberá tener un tipo de usuario el Usuario cliente.
- La aplicación deberá permitir la autenticación únicamente al usuario de tipo cliente.
- La aplicación deberá mostrar cómo información general el Nombre del gimnasio, descripción, misión, visión, encargado, información de contacto (email y número de teléfono), horario de atención y la ubicación de las instalaciones.
- La aplicación deberá permitir la edición del perfil de usuario tipo cliente, se permitirá modificar los siguientes datos: Ocupación, teléfono, email y una fotografía.
- La aplicación deberá permitir al usuario de tipo cliente seleccionar la rutina que desee realizar.
- La aplicación deberá permitir al usuario de tipo cliente seleccionar el ejercicio que desee realizar según cada rutina.
- La aplicación deberá permitir al usuario de tipo cliente visualizar las instrucciones de cada ejercicio.
- La aplicación deberá mostrar al usuario cliente avisos agregados por el administrador del gimnasio.
- La aplicación deberá mostrar al usuario los eventos que tiene el gimnasio durante la semana.
- La aplicación deberá recordar al usuario la próxima fecha de pago.

Requerimientos no funcionales de la aplicación móvil

Confiabilidad

- El sistema tendrá un sistema de autenticación basado en un carnet y una contraseña para mantener privado el perfil de cada usuario.

Rendimiento

- El tiempo de respuesta de las solicitudes al Backend deberá no ser mayor a 10 segundos.

Mantenimiento

- La aplicación deberá tener documentación interna que describa las funcionalidades de cada clase, así como los métodos usados, esto para facilitar el mantenimiento a la aplicación.
- La aplicación deberá tener documentación que describa las decisiones de diseño tomadas con respecto a manejo de interfaz, lógica y datos, esto para facilitar el mantenimiento a la aplicación.

Usabilidad

- La autenticación del usuario de tipo cliente se realizará mediante un número de carnet y una contraseña suministrada por el gimnasio.
- El sistema debe mostrar las instrucciones de cada ejercicio mediante texto que explique el objetivo del ejercicio, los pasos a seguir e imágenes representativas.
- El sistema debe mostrar al usuario la ubicación de las instalaciones mediante un mapa o mediante una aplicación de navegación vehicular.
- La aplicación deberá mostrar al usuario los eventos que tiene el gimnasio durante la semana mediante un horario semanal ordenado por días (Columnas) y horas (Filas).
- El tiempo de aprendizaje sobre el uso de la aplicación para los clientes no debe superar los diez minutos.

Reglas del negocio

- Cada usuario de tipo cliente puede tener asociada más de una rutina.

Funcionalidades del Backend

- El sistema deberá tener dos tipos de usuario: Usuario administrador y usuario instructor.
- La aplicación deberá permitir crear, eliminar y editar usuarios de tipo administrador e instructor.
- El sistema deberá permitir la autenticación de los usuarios.
- El sistema deberá permitir al usuario de tipo administrador e instructor pueden crear, mientras que sólo el de tipo administrador puede editar y eliminar clientes del gimnasio, cada cliente tendrá los siguientes datos: Nombre, fecha de nacimiento, sexo, número de cédula, fotografía, fecha de ingreso al gimnasio, tipo de pago (Mensual, bimensual, trimestral, semestral y anual), ocupación, email y teléfono, lesiones y alguna descripción especial.
- El sistema debe permitir al usuario de tipo administrador crear, editar y eliminar eventos del gimnasio, cada evento tiene los siguientes datos: Nombre, descripción, día y hora.
- El sistema debe permitir al usuario de tipo administrador enviar avisos a los clientes del gimnasio, cada aviso tiene la siguiente información: Nombre, descripción, fecha y hora.
- El sistema debe permitir al usuario de tipo administrador consultar los clientes del gimnasio de acuerdo a los siguientes parámetros: Nombre, apellidos, morosidad y fecha de ingreso.
- El sistema debe permitir al usuario de tipo instructor registrar cada cierto tiempo (Mes, semana) las medidas del cliente del gimnasio (medidas corporales en cm).
- El sistema deberá permitir al usuario de tipo instructor crear, editar y eliminar rutinas, cada rutina tiene los siguientes datos: Nombre, objetivo y descripción.
- El sistema deberá permitir al usuario de tipo instructor crear, editar y eliminar ejercicios, cada ejercicio tiene los siguientes datos: Nombre, descripción e instrucciones.
- El sistema deberá permitir al usuario de tipo instructor añadir y eliminar ejercicios a una rutina específica.
- El sistema deberá permitir a ambos usuario visualizar la información del registro de entrada de los clientes al gimnasio.
- El sistema deberá permitir a ambos tipos de usuario registrar la entrada de un cliente al gimnasio, ese registro tendrá los siguientes datos: carnet, fecha y hora.

Requerimientos no funcionales del Backend

Confiabilidad

- El sistema deberá tener un sistema de autenticación basado en un nombre de usuario y una contraseña para permitir solo usuarios autorizados.
- El sistema deberá validar cada uno de los datos en los formularios con el objetivo de evitar que el usuario registre información no válida para un campo específico.

Rendimiento

- El sistema deberá tener un tiempo de respuesta no mayor 10 segundos para recibir y enviar los datos.
- El tiempo de respuesta de las solicitudes con los API's externos no deberá de ser mayor a 10 segundos.

Mantenimiento

- La aplicación deberá tener documentación que describa las decisiones de diseño tomadas con respecto a manejo de interfaz, lógica y datos, esto para facilitar el mantenimiento a la aplicación.
- La aplicación deberá tener documentación interna que describa las funcionalidades de cada clase, así como los métodos usados, esto para facilitar el mantenimiento a la aplicación.

Usabilidad

- La autenticación el usuario de tipo cliente se realizará mediante un nombre de usuario y una contraseña suministrada por el gimnasio.
- La aplicación deberá mostrar al usuario los eventos que tiene el gimnasio durante la semana mediante un horario semanal ordenado por días (Columnas) y horas (Filas).
- La aplicación deberá mostrar un formulario para la creación de los eventos y avisos.
- El tiempo de aprendizaje sobre el uso de la aplicación para los clientes no debe superar los diez minutos.

Descripción de la arquitectura

Los patrones de arquitectura utilizados en la aplicación son los siguientes:

- **Thin Client:**
La aplicación se basa en arquitectura Thin Client puesto que el procesamiento es realizado en mayor parte en el Backend.
- **Cliente-Servidor:**
La aplicación utiliza una arquitectura Cliente-Servidor donde la app se comunica con un con el Backend (Servidor) para la solicitud de datos.
- **MVC**
La aplicación utilizará el patrón Modelo-Vista-Controlador bajo el Framework Alloy de Titanium SDK el cual lo implementa de la siguiente manera:
 - **Modelo:** Alloy utiliza objetos de tipo *Backbone and Collection* para proveer los datos requeridos por la aplicación.
 - **Vista:** Utilizando *XML markup and Titanium Stylesheets (TSS)* proveen los diversos componentes gráficos para una aplicación.
 - **Controlador:** Utilizando *Alloy API and Titanium API* se realiza la interacción con el modelo y la vista.

Diagrama de Clases

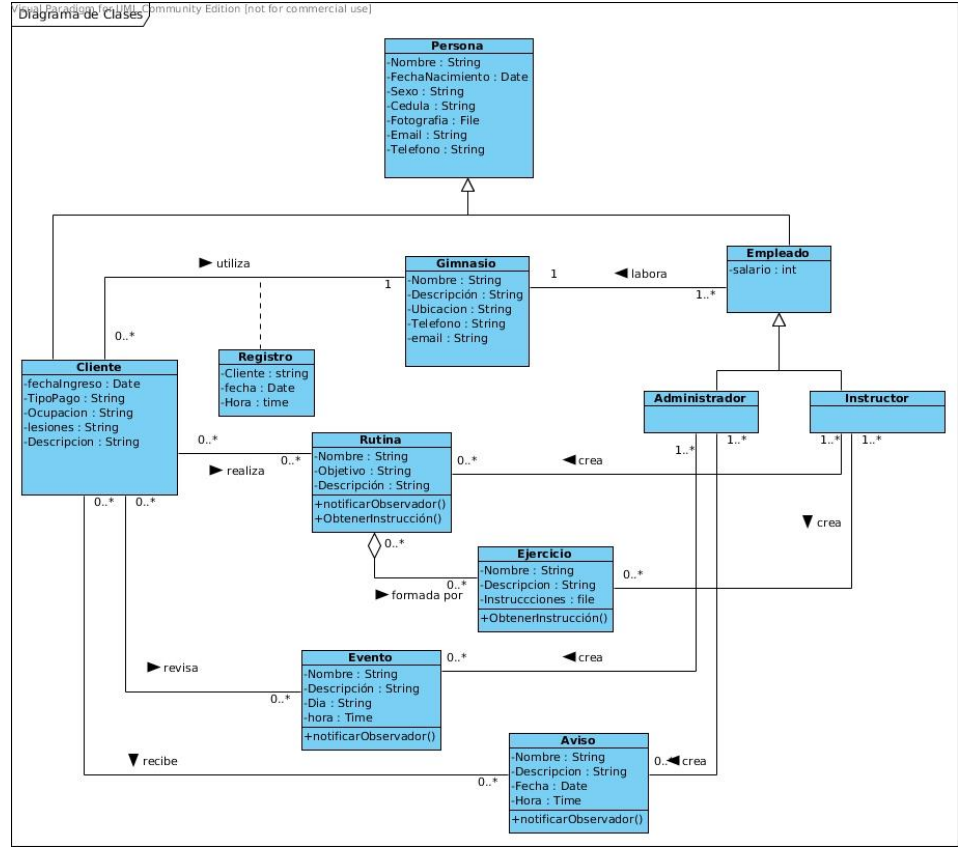


Diagrama de Componentes

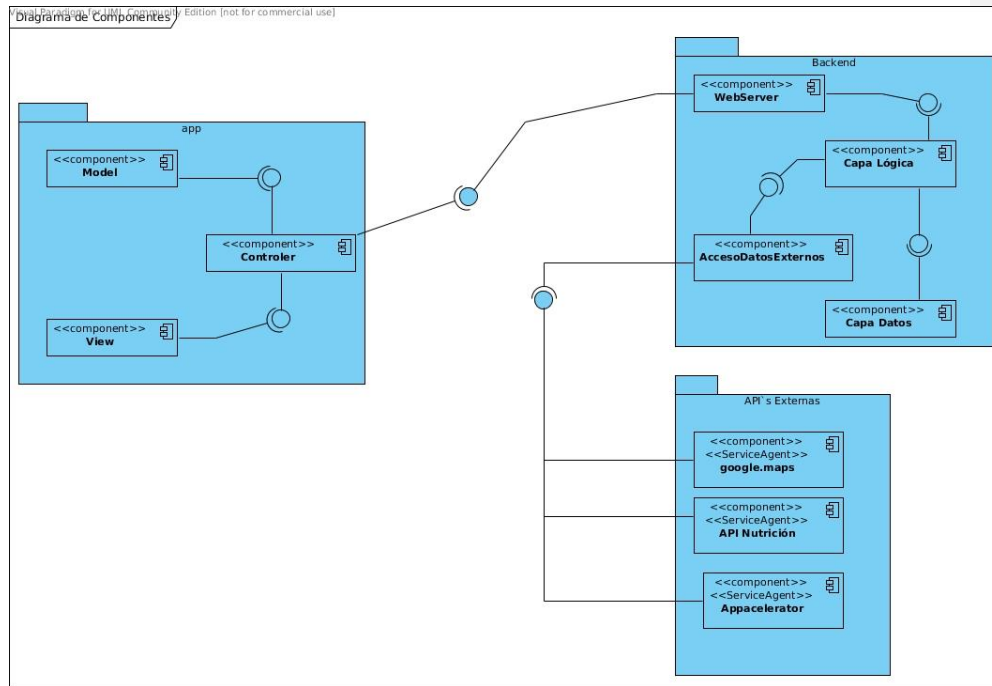
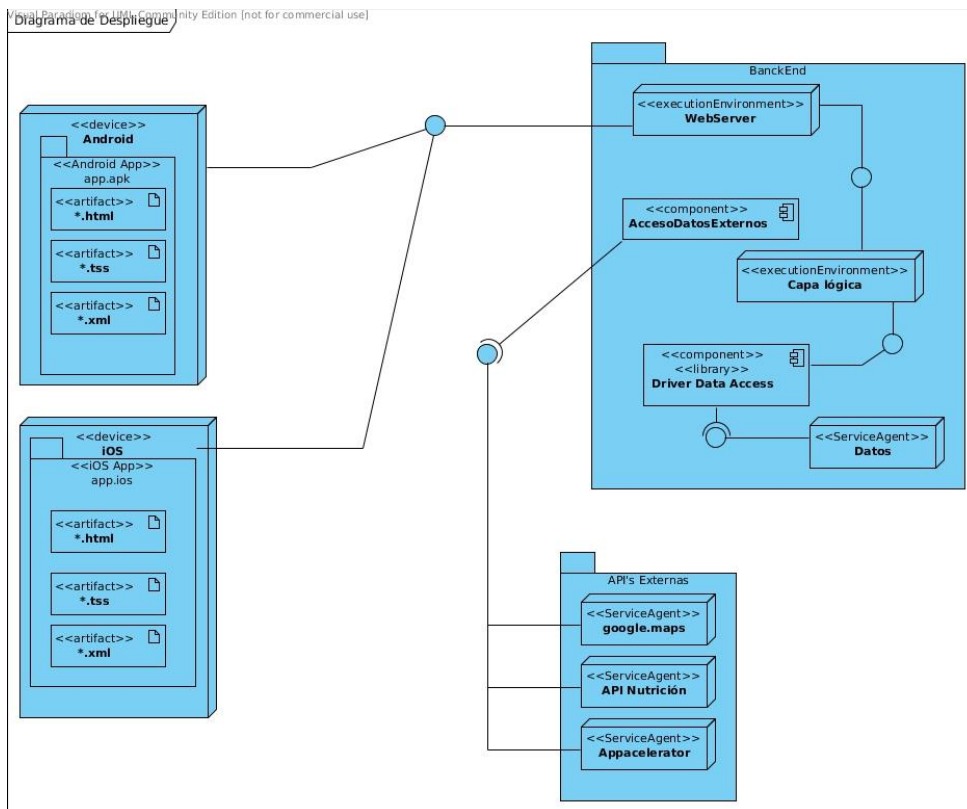


Diagrama de Despliegue



Descripción de los Web Services

Descripción	URL	Params	Método HTTP
UpdatePerfil Se utiliza para actualizar la información del perfil de usuario	http://gymservicio.cloudapp.net/Service1.svc/json/UpdatePerfil	int Id_Usuario string Telefono string Ocupacion string Email	GET
Rutina Se utiliza para obtener las rutinas y sus ejercicios respectivos	http://gymservicio.cloudapp.net/Service1.svc/json/Rutina	int a * (Id Cliente)	GET
Perfil Se utiliza para obtener la información del usuario	http://gymservicio.cloudapp.net/Service1.svc/json/Perfil	int NUMERO_CLIENTE	GET
Login Se utiliza para determinar si el usuario tiene acceso a la aplicación	http://gymservicio.cloudapp.net/Service1.svc/json/Login	string a (UserName) string b (Password)	GET

<p>Horario</p> <p>Se utiliza para obtener las actividades que se realizan determinado día de la semana</p>	http://gymservicio.cloudapp.net/Service1.svc/json/Horario	string valor (Dia de la semana)	GET
<p>GetMedidas</p> <p>Se utiliza para obtener las medidas anatómicas del cliente</p>	http://gymservicio.cloudapp.net/Service1.svc/json/GetMedidas	int Id_Usuario	GET
<p>GetNotificaciones</p> <p>Se utiliza para obtener las notificaciones enviadas por el administrador, además del estado de la membresía con respecto a la fecha de pago.</p>	http://gymservicio.cloudapp.net/Service1.svc/json/GetNotificaciones	int Id_Usuario	GET

Interacción con sistemas externos

Para el funcionamiento de la aplicación se necesitará la interacción con otros sistemas, que este caso serán API's necesarias tanto para obtener los datos necesarios así como para proporcionar diferentes servicios al usuario.

Google maps: Con el objetivo de mostrar la ubicación del gimnasio de una manera más práctica y sencilla se muestra un componente de mapa obtenido mediante el API de google Maps.

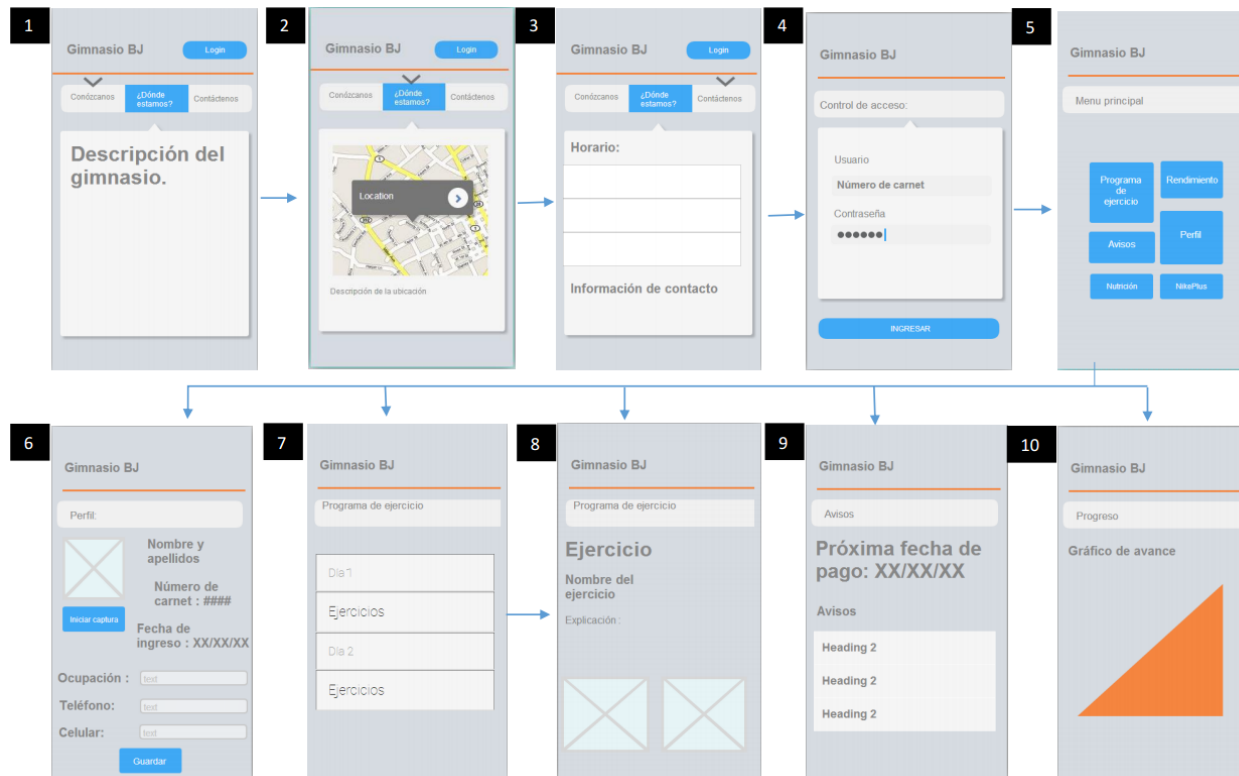
Nutrición: La alimentación es uno de los aspectos más importantes cuando una persona realiza actividad física y más si el objetivo es obtener masa muscular o bajar de peso, ante esto se consume información nutricional de alimentos, lo cual permite a los usuarios poder consultar información como cantidad de grasa y calorías de distintas presentaciones de un alimento.

Titanium App Accelerator: Este API es provisto por el framework, para acceder a ciertas funcionalidades necesarias en este tipo de desarrollo híbrido, tales como manejo de hardware del teléfono, así como la creación de clientes HTTP para el acceso de API's.

Facebook: El usuario puede compartir en su cuenta de Facebook información relacionada con la rutina y los ejercicios que realiza.

Wireframes

Pantallas



Descripción de pantallas

1. En esta pantalla se hace el uso de tabs con los cuales se puede navegar hasta la pantalla 3, en la específicamente se muestra una descripción del gimnasio con la misión y visión del mismo así como información relevante acerca de instructores, etc.
2. En esta pantalla se muestra por medio de un mapa proporcionado por google maps la ubicación del dispositivo móvil con respecto al gimnasio.
3. En esta pantalla se muestra el horario del gimnasio, con las respectivas clases que se imparten de manera diaria (Spinning, Aeróbicos, Pilates, etc.), también se muestra el número de teléfono y correo electrónico para el contacto directo con el gimnasio.
4. En esta pantalla se muestra un formulario de login para el cliente en donde debe proporcionar su número de carnet y contraseña para acceder a la demás funciones de la aplicación.
5. En esta pantalla se muestra un menú con las distintas funcionalidades del programa.
6. En esta pantalla se muestra la información del perfil del usuario en donde puede editar la ocupación, teléfono y el celular y además de poder agregar una foto.
7. En esta pantalla se muestra el programa de ejercicios perteneciente a cada usuario subdivido por día de ejercicios.
8. En esta pantalla se muestra de manera detallada cada uno de los ejercicios especificados en la pantalla anterior con una descripción del cómo se debe hacer y además unas imágenes ilustrativas
9. En esta pantalla se mostrarán avisos del gimnasio como tal, ya sean cambio de horarios u otro tipo, además se mostrará la fecha próxima de pago.
10. En esta pantalla se mostrará un gráfico con el progreso del usuario a través del tiempo con respecto a los cambios de rutinas efectuados.

Patrones de diseño aplicados

Patrones de despliegue de datos

La aplicación cuenta con ListViews donde se muestran datos, algunos de estos ListViews implementan el patrón Master & Detail.

Se cuenta con navegación primaria y secundaria, en la parte primaria se tiene un SpringBoard, el cual está compuesto por imágenes y texto para indicar la acción de cada elemento, una vez seleccionado un elemento se entra en la navegación secundaria la cual tiene controles para retroceder.

Además para la navegación en la aplicación se cuenta con un menú desplegable lateral ubicado en el lado izquierdo, este permite un acceso rápido y fácil a ciertos elementos de la aplicación.

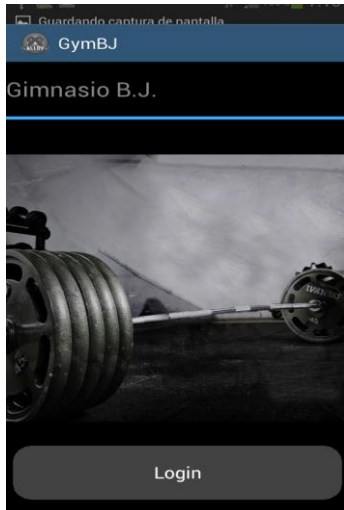
La aplicación de metáfora se da mediante el uso de Google Maps, lo cual permite al usuario tener una referencia geográfica de la ubicación del gimnasio, mediante el uso de relieves y simbologías usadas en los mapas cartográficos.

Comentado [Ibb1]:

En los patrones de herramientas se implementó un generador de gráficos el cual implementa una función que toma las medidas anatómicas del usuario y genera gráficos de barras comparativos para indicar si existe progreso en los objetivos deseados por el cliente. Otra implementación del patrón herramienta está presente en el reproductor usado para mostrar al usuario un video ilustrativo de un determinado ejercicio, las herramientas presentes en el reproductor son los controles de "Reproducir", "Pausar", "Adelantar" y "Retroceder".

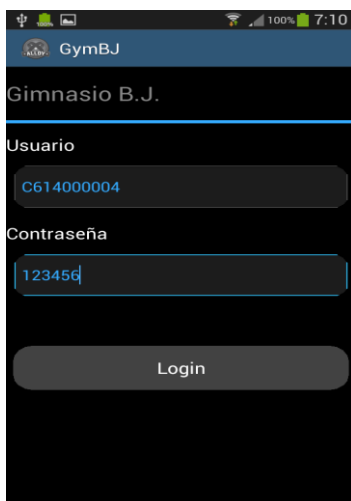
Manual de usuario

1.



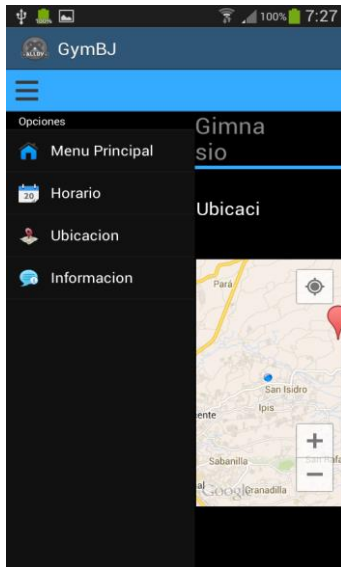
Esta es la pantalla inicial en donde el usuario registrado en el gimnasio se va a logear, mediante el botón de Login ubicado en la parte inferior de la pantalla.

2.



En esta pantalla el usuario realiza el login mediante su nombre de usuario y contraseña; en caso de que estos sean correctos se le permitirá acceso a la aplicación en caso contrario no lo dejará ingresar.

3.



En esta pantalla se muestra el menú lateral en donde el usuario puede hacer uso de las funciones básicas de la aplicaciones y también seleccionar la opción de menú principal donde se encuentran las demás funcionalidades.

4.



Esta pantalla corresponde al icono de información del menú lateral en donde puede observar la información del gimnasio.

5.



Esta pantalla corresponde con el icono de ubicación del menú lateral en donde muestra un mapa con la posición actual de usuario con respecto al gimnasio.

6.



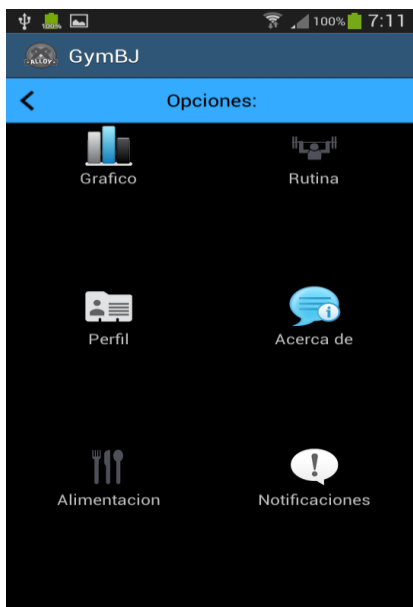
Esta pantalla corresponde con el icono de horario del menú lateral en donde el usuario puede ver por día cuales son las clases y eventos que se realizan.

7.



Esta pantalla corresponde con el día seleccionado por el usuario en la pantalla número 6 en donde muestra la clases y eventos del día.

8.



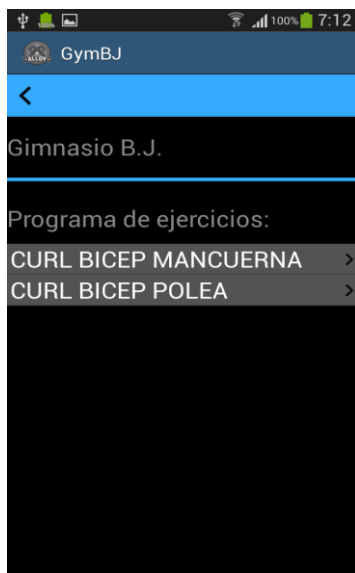
Esta pantalla corresponde con el icono de menú principal ubicado en el menú lateral de inicio; aquí se muestran las distintas funcionalidades del programa.

9.



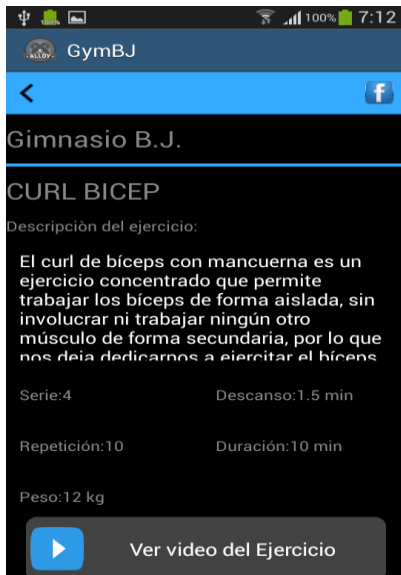
Esta pantalla corresponde con el icono de gráfico del menú principal en donde el usuario puede graficar sus últimas 2 medidas y ver su progreso en el gimnasio

10.



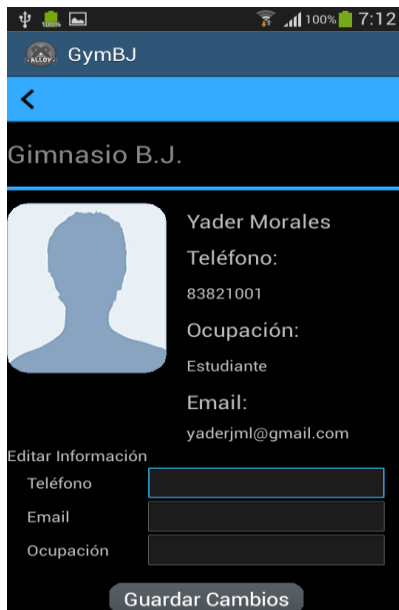
Esta pantalla corresponde al icono de rutina del menú principal en donde se muestran los ejercicios de la rutina de un cliente. En donde puede seleccionar un ejercicio y ver sus detalles.

11.



Esta pantalla corresponde al ejercicio seleccionado en la pantalla anterior en donde se muestra la información del ejercicio, un video y puede compartirlo

12.



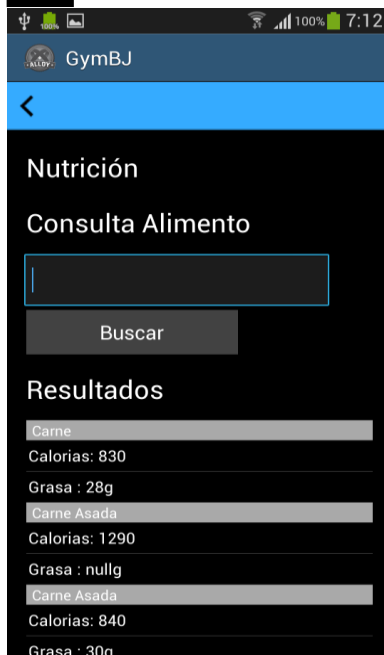
Esta pantalla corresponde con el icono de perfil del menú principal en donde el usuario puede ver su información principal y editar los datos.

13.



Esta pantalla corresponde en el acerca de del menú principal en donde se muestra la información de la aplicación como tal.

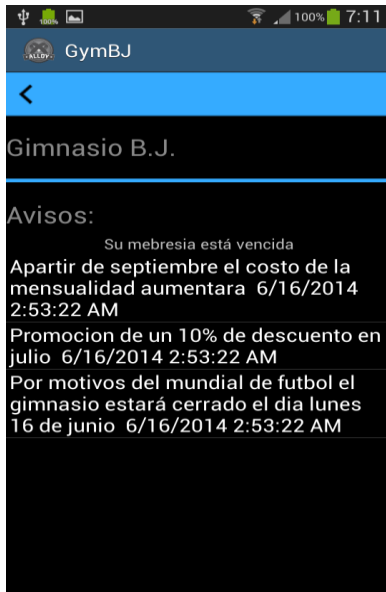
14.



Esta pantalla corresponde con el icono de Alimentación del menú principal en donde el usuario puede consultar un alimento y ver sus calorías y grasas.

15.

25



Esta pantalla corresponde con el icono de notificaciones del menú principal en donde se pueden ver los avisos importantes para el cliente