



Proyecto

WikiFut

Desarrollo de Aplicaciones Para Dispositivos Móviles

Elaborado por:

Miguel Ángel Ceballos Yate - 2259619

Cristhian Leonardo Albarracin Zapata - 1968253

Sebastián Gómez Agudelo - 2259474

Stiven Henao Aricapa - 2259603

Víctor Manuel Hernández Ortiz - 2259520

Juan David Valencia Montalvo - 2160103

Tina María Torres Tascón - 2259729

Docente:

Áskar Rodríguez, MIGUEL ÁNGEL

Sede Tuluá

Marzo 2025

Índice

Introducción	3
Propósito del Proyecto	3
Objetivos del Proyecto	3
Objetivo General:	
Desarrollar una aplicación que permita a los usuarios seguir equipos de fútbol, consultar información detallada sobre partidos y ligas, y recibir notificaciones personalizadas, utilizando APIs externas para obtener datos en tiempo real.	3
Objetivos Específicos:	3
Problema o Necesidad a Solventar	4
Alcance del Proyecto	4
Stakeholders y Usuarios Finales	5
Requerimientos Funcionales	6
Enlaces	20

Introducción

Este documento describe el proyecto de desarrollo de una aplicación móvil y/o web que permite a los usuarios seguir y obtener información detallada sobre equipos de fútbol, partidos, ligas y estadísticas en tiempo real.

Propósito del Proyecto

El propósito de este proyecto es desarrollar una aplicación que:

- Proporcione a los usuarios acceso rápido y confiable a información actualizada sobre equipos, partidos y ligas de fútbol.
- Permita a los usuarios personalizar su experiencia seleccionando equipos favoritos y recibiendo notificaciones sobre partidos importantes.
- Utilice APIs externas para obtener datos en tiempo real y garantizar la precisión de la información.
- Ofrezca una interfaz intuitiva y fácil de usar para mejorar la experiencia del usuario.

Objetivos del Proyecto

Objetivo General:

Desarrollar una aplicación que permita a los usuarios seguir equipos de fútbol, consultar información detallada sobre partidos y ligas, y recibir notificaciones personalizadas, utilizando APIs externas para obtener datos en tiempo real.

Objetivos Específicos:

- Implementar un sistema de registro y autenticación de usuarios seguro y eficiente.
- Permitir a los usuarios seleccionar y gestionar una lista de equipos favoritos.

- Proporcionar notificaciones en tiempo real sobre partidos de equipos favoritos.
- Mostrar listados de partidos del día, ligas disponibles y detalles de partidos históricos y futuros.
- Garantizar la integración con APIs externas para obtener datos actualizados.

Problema o Necesidad a Solventar

Actualmente, los aficionados al fútbol enfrentan dificultades para acceder a información centralizada y actualizada sobre sus equipos y partidos favoritos.

Alcance del Proyecto

El proyecto incluye:

- Desarrollo de un sistema de registro y autenticación de usuarios.
- Integración con la API-Football para obtener datos sobre equipos, partidos y ligas.
- Implementación de funcionalidades como selección de equipos favoritos, notificaciones de partidos y consulta de ligas y partidos.
- Diseño de una interfaz de usuario intuitiva y responsive.

El proyecto no incluye:

- Desarrollo de APIs externas (se utilizarán APIs existentes).
- Mantenimiento de servidores externos o bases de datos de terceros.

Beneficios Esperados

- Centralización de información deportiva en una sola plataforma.
- Personalización de la experiencia del usuario.

- Acceso a datos en tiempo real y actualizados mediante APIs confiables.
- Mejora en la experiencia del usuario gracias a una interfaz intuitiva.

Stakeholders y Usuarios Finales

Stakeholders:

- Equipo de desarrollo.
- Proveedores de APIs (ejemplo: API-Football).
- Posibles patrocinadores o inversores.

Usuarios Finales:

- Aficionados al fútbol que deseen seguir a sus equipos favoritos.
- Usuarios que buscan información actualizada sobre partidos y ligas.

Requerimientos Funcionales

Código:	RF-01
Título:	Registro de Usuario
Descripción:	<p>La aplicación debe permitir a los nuevos usuarios registrarse creando una cuenta personal. Para ello, deberán proporcionar un nombre, correo electrónico y contraseña, asegurando el cumplimiento de las políticas de seguridad establecidas. El sistema validará que el correo electrónico no esté registrado previamente y que la contraseña cumpla con los requisitos de seguridad. Una vez completado el registro, se enviará un correo de confirmación con un enlace para verificar la cuenta antes de permitir el acceso. Este proceso será gestionado a través de la API de Usuarios, garantizando la seguridad de la información mediante encriptación de contraseñas.</p>
Entrada:	<ul style="list-style-type: none">● Nombre (<i>String</i>)● Correo electrónico (<i>String</i>)● Contraseña (<i>String</i>).
Proceso:	<ol style="list-style-type: none">1. El usuario accede a la pantalla de registro.2. Ingresa su nombre, correo electrónico y contraseña en el formulario.3. La aplicación valida:<ul style="list-style-type: none">● Que el correo no esté registrado en la base de datos.● Que la contraseña cumpla con los criterios de seguridad.4. Si los datos son válidos:<ul style="list-style-type: none">● Se almacena la información del usuario en la base de datos con la contraseña encriptada.● Se envía un correo de confirmación con un enlace para activar la cuenta.5. Si el usuario hace clic en el enlace de confirmación, su cuenta se activa y podrá iniciar sesión.6. Si el registro falla, se muestra un mensaje de error indicando el motivo.
Salida:	<ul style="list-style-type: none">● Cuenta creada y correo de confirmación enviado.● Mensaje de error en caso de datos inválidos o correo ya registrado.● Notificación de activación exitosa tras la confirmación del correo.
Precondiciones:	<ul style="list-style-type: none">● El usuario debe proporcionar un correo electrónico válido.

	<ul style="list-style-type: none"> ● La API de Usuarios y el servicio de correo deben estar operativos.
Postcondiciones:	<ul style="list-style-type: none"> ● Si el usuario confirma su correo, podrá iniciar sesión en la aplicación. ● Si no confirma su cuenta, no podrá autenticarse hasta que complete la activación. ● Los datos del usuario quedan almacenados en la base de datos.
Reglas de negocio:	<ul style="list-style-type: none"> ● La contraseña debe cumplir con los siguientes requisitos: <ul style="list-style-type: none"> ○ Mínimo 8 caracteres. ○ Al menos una letra mayúscula y un número. ● Si un usuario intenta registrarse con un correo ya existente, se debe rechazar el registro y mostrar un mensaje de error. ● La cuenta debe confirmarse en un plazo de 24 horas o el enlace de activación expirará. ● Se debe permitir el reenvío del correo de activación en caso de que el usuario lo solicite.
Prioridad:	Alta (Esencial)
Fuente:	Requerimientos del cliente
Referencias a Componentes	Módulo de Autenticación, API de Usuarios, Base de Datos

Código:	RF-02
Título:	Inicio de Sesión de Usuario
Descripción:	La aplicación debe permitir a los usuarios autenticarse en el sistema mediante su correo electrónico y contraseña para acceder a su cuenta personal. El sistema validará las credenciales ingresadas consultando la base de datos y, en caso de ser correctas, otorgará acceso al usuario. Si las credenciales son incorrectas, se mostrará un mensaje de error indicando la causa del fallo. Para mejorar la seguridad, se implementará un bloqueo de cuenta tras 5 intentos fallidos, y el usuario podrá solicitar un enlace de recuperación de contraseña en caso de olvido. La autenticación se gestionará mediante la API de Usuarios, garantizando seguridad y compatibilidad con futuras integraciones, como autenticación en dos pasos.
Entrada:	Correo electrónico (<i>String</i>) Contraseña (<i>String</i>)
Proceso:	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de inicio de sesión. Ingresa su correo electrónico y contraseña en el formulario de autenticación. 2. La aplicación envía las credenciales a la API de Usuarios. 3. El sistema verifica las credenciales en la base de datos: <ul style="list-style-type: none"> • Si son correctas, se genera un token de sesión y se concede acceso. • Si son incorrectas, se muestra un mensaje de error. 4. Si se superan los 5 intentos fallidos, la cuenta se bloquea y se envía un correo con instrucciones para su desbloqueo.
Salida:	<ul style="list-style-type: none"> • Acceso concedido, con token de sesión válido. • Mensaje de error en caso de credenciales incorrectas. • Notificación de cuenta bloqueada tras superar el límite de intentos.
Precondiciones:	<ul style="list-style-type: none"> • El usuario debe estar previamente registrado en el sistema. • La API de Usuarios y la base de datos deben estar operativas.
Postcondiciones:	<ul style="list-style-type: none"> • Si la autenticación es exitosa, el usuario es redirigido a la pantalla principal de la aplicación. • Si la autenticación falla, el usuario permanece en la pantalla de inicio de sesión con un mensaje de error. • Si la cuenta es bloqueada, el usuario no podrá acceder hasta desbloquearla a través del enlace de recuperación.
Reglas de negocio:	<ul style="list-style-type: none"> • Se debe bloquear la cuenta tras 5 intentos fallidos. • El usuario podrá recuperar su cuenta a través de un correo

	<p>de recuperación con un enlace para restablecer la contraseña.</p> <ul style="list-style-type: none"> • La sesión del usuario permanecerá activa por un tiempo determinado según las políticas de seguridad. • Se debe permitir cerrar sesión manualmente desde la configuración.
Prioridad:	Alta (Esencial)
Fuente:	Requerimientos del cliente
Referencias a Componentes	Módulo de Autenticación, API de Usuarios, Base de Datos

Código:	RF-03
Título:	Selección de Equipos Favoritos
Descripción:	<p>La aplicación debe permitir a los usuarios seleccionar y almacenar una lista personalizada de equipos favoritos en su cuenta, brindándoles acceso rápido a información detallada de cada equipo, incluyendo su nombre, país, liga en la que compite, estadio (nombre, capacidad, ubicación e imagen si está disponible) y estadísticas relevantes de la temporada actual, como posición en la liga, partidos ganados y perdidos, goles a favor y en contra, y rendimiento en casa y fuera. Los usuarios podrán agregar o eliminar equipos de su lista de favoritos en cualquier momento, y la información se obtendrá en tiempo real desde la API-Football, garantizando actualizaciones dinámicas dentro de la aplicación.</p>
Entrada:	<p>Lista de equipos seleccionados por el usuario (<i>Array<Objeto></i>): Cada objeto en la lista contiene:</p> <ul style="list-style-type: none"> ● ID del equipo (<i>Integer</i>) ● Nombre del equipo (<i>String</i>) ● País (<i>String</i>) ● Liga (<i>String</i>) ● Estadio (Objeto): <ul style="list-style-type: none"> ○ Nombre (<i>String</i>) ○ Capacidad (<i>Integer</i>) ○ Ubicación (<i>String</i>) ○ Imagen (<i>String</i>, opcional) ● Estadísticas de la temporada (Objeto): <ul style="list-style-type: none"> ○ Posición en liga (<i>Integer</i>) ○ Partidos ganados (<i>Integer</i>) ○ Partidos perdidos (<i>Integer</i>) ○ Goles a favor (<i>Integer</i>) ○ Goles en contra (<i>Integer</i>) ○ Rendimiento en casa y fuera (Objeto)
Proceso:	<ol style="list-style-type: none"> 1. El usuario accede a la lista de equipos. 2. Selecciona los equipos que desee seguir. 3. La aplicación guarda la selección en su perfil. 4. Cuando el usuario consulta su lista de favoritos, la aplicación recupera la información actualizada desde la API-Football. 5. La información del equipo favorito se muestra en la interfaz de usuario.
Salida:	<p>Lista de equipos favoritos guardada (<i>Array<Objeto></i>): Cada objeto en la lista contiene:</p>

	<ul style="list-style-type: none"> ● ID del equipo (<i>Integer</i>) ● Nombre del equipo (<i>String</i>) ● País (<i>String</i>) ● Liga (<i>String</i>) ● Estadio (<i>Objeto</i>) ● Estadísticas de la temporada (<i>Objeto</i>)
Precondiciones:	<ul style="list-style-type: none"> ● El usuario debe estar autenticado en la aplicación. ● La API-Football debe estar operativa y accesible.
Postcondiciones:	<ul style="list-style-type: none"> ● El usuario podrá visualizar información actualizada sobre los equipos seleccionados. ● Los datos de los equipos favoritos se mantendrán sincronizados con la API-Football.
Reglas de negocio:	<ul style="list-style-type: none"> ● Un usuario puede seleccionar un máximo de 20 equipos favoritos. ● Si la API-Football no está disponible, la aplicación debe mostrar un mensaje indicando que la información no pudo ser recuperada.
Prioridad:	Media (Esencial)
Fuente:	Requerimientos del cliente
Referencias a Componentes	Módulo de Favoritos, API Football, Base de Datos

Código:	RF-04
Título:	Notificaciones de Partidos de Equipos Favoritos
Descripción:	La aplicación debe notificar a los usuarios sobre los próximos partidos de sus equipos favoritos en función de sus preferencias de notificación. Las alertas deben enviarse antes del inicio del partido, permitiendo al usuario mantenerse informado en tiempo real sobre los encuentros más relevantes de sus equipos seleccionados.
Entrada:	<p>Entrada:</p> <p>Lista de equipos favoritos del usuario (<i>Array<Objeto></i>): Cada objeto en la lista contiene:</p> <ul style="list-style-type: none"> ● ID del equipo (<i>Integer</i>) ● Nombre del equipo (<i>String</i>) ● Liga (<i>String</i>) ● Partidos programados (<i>Array<Objeto></i>): <ul style="list-style-type: none"> ○ ID del partido (<i>Integer</i>) ○ Fecha y hora del partido (<i>DateTime</i>) ○ Equipo local (<i>String</i>) ○ Equipo visitante (<i>String</i>) ○ Ubicación (opcional) (<i>String</i>) ○ Canal de transmisión (opcional) (<i>String</i>) <p>Estado de notificaciones (<i>Boolean: Activas/Inactivas</i>)</p>
Proceso:	<ol style="list-style-type: none"> 1. El sistema consulta automáticamente los partidos programados de los equipos favoritos del usuario a través de la API-Football. 2. Filtra los partidos próximos en función de la fecha y hora actual. 3. Determina el tiempo de anticipación configurado por el usuario para enviar la notificación. 4. Genera una notificación emergente con información del partido. 5. La notificación se envía al dispositivo del usuario antes del inicio del partido.
Salida:	<p>Notificación emergente en el dispositivo del usuario (<i>Objeto</i>), incluyendo:</p> <ul style="list-style-type: none"> ● Título: "Próximo partido de [Equipo Favorito]" ● Mensaje: "[Equipo Local] vs. [Equipo Visitante] – [Fecha y Hora]" ● Ubicación (si está disponible) ● Canal de transmisión (si está disponible)

	<ul style="list-style-type: none"> ● Acción rápida para ver más detalles en la aplicación
Precondiciones:	<ul style="list-style-type: none"> ● El usuario debe haber iniciado sesión en la aplicación. ● El usuario debe haber seleccionado al menos un equipo favorito. ● La aplicación debe tener acceso a la API-Football para obtener la información de los partidos.
Postcondiciones:	<ul style="list-style-type: none"> ● El usuario recibe alertas oportunas sobre los partidos de sus equipos favoritos. ● El usuario puede acceder a más detalles del partido desde la notificación.
Reglas de negocio:	<ul style="list-style-type: none"> ● El usuario puede activar o desactivar las notificaciones en la configuración de la aplicación. ● Se pueden establecer diferentes niveles de anticipación para las notificaciones (ejemplo: recibir alertas 15 minutos antes del partido o el día anterior). ● La aplicación no enviará notificaciones si el usuario ha desactivado esta opción.
Prioridad:	Media (Esencial)
Fuente:	Requerimientos del cliente
Referencias a Componentes	Módulo de Favoritos, API de Usuarios, API Football, Base de Datos

Código:	RF-05
Título:	Listado de Partidos del Día
Descripción:	La aplicación debe mostrar al usuario la lista de partidos programados para la fecha actual, incluyendo detalles como hora de inicio, equipos involucrados y, si está disponible, el resultado en tiempo real o final.
Entrada:	Fecha actual (<i>Date</i>): Fecha del día en que se realiza la consulta.
Proceso:	<ol style="list-style-type: none"> 1. La aplicación obtiene la fecha actual del sistema. 2. Consulta la base de datos o API externa de eventos deportivos. 3. Filtra los partidos programados que coincidan con la fecha actual. 4. Recupera los detalles del partido, incluyendo: <ul style="list-style-type: none"> • Equipos que disputan el encuentro. • Hora de inicio. • Ubicación (si está disponible). • Estado del partido (programado, en curso, finalizado). • Resultado final (si el partido ya ha concluido). 5. Muestra la lista de partidos en la interfaz de usuario con la información relevante.
Salida:	<p>Lista de partidos del día (<i>Array<Objeto></i>): Cada objeto en el array contiene:</p> <ul style="list-style-type: none"> • Nombre del equipo local (<i>String</i>). • Nombre del equipo visitante (<i>String</i>). • Hora del partido (<i>Time</i>). • Ubicación (<i>String</i>, opcional). • Estado del partido (<i>String</i>: "Programado", "En curso", "Finalizado"). • Resultado (<i>String</i>, opcional si el partido ya ha concluido).
Precondiciones:	<ul style="list-style-type: none"> • La aplicación debe tener conexión a la fuente de datos de los partidos del día. • La API o base de datos deportiva debe estar operativa.
Postcondiciones:	<ul style="list-style-type: none"> • El usuario visualiza la lista de partidos programados para el día actual con información relevante. • Si el partido ya terminó, el usuario puede ver el resultado final (si está disponible en la fuente de datos).
Reglas de negocio:	<ul style="list-style-type: none"> • No se mostrarán partidos de fechas pasadas o futuras en esta vista. • Si no hay partidos programados para el día actual, se debe

	mostrar un mensaje informativo ("No hay partidos programados para hoy").
Prioridad:	Alta
Fuente:	Requerimientos del cliente
Referencias a Componentes	Módulo de Favoritos, API de Usuarios, API Football, Base de Datos

Código:	RF-06
Título:	Consulta de Ligas Disponibles
Descripción:	La aplicación debe proporcionar una interfaz que liste todas las ligas disponibles de manera clara y organizada. Cada liga debe mostrarse con su nombre, logotipo y detalles clave, como el país de origen y la temporada en curso. Además, la aplicación debe permitir filtrar y buscar ligas por nombre, región o categoría para facilitar la navegación.
Entrada:	Ninguna
Proceso:	<ol style="list-style-type: none"> 1. El usuario accede a la sección de ligas desde el menú de navegación. 2. La aplicación envía una solicitud al servidor para obtener la lista de ligas. 3. Se reciben y procesan los datos de las ligas disponibles. 4. La aplicación muestra la lista de ligas con la siguiente información: <ul style="list-style-type: none"> • Nombre de la liga • Logotipo de la liga • País de origen • Temporada en curso 5. Se habilitan opciones de búsqueda y filtrado para facilitar la navegación: <ul style="list-style-type: none"> • Búsqueda por nombre de liga • Filtrado por país o categoría 6. Si el número de ligas es grande, se implementa paginación para mejorar la experiencia del usuario. 7. Si no hay ligas disponibles, se muestra un mensaje informativo al usuario ("No hay ligas disponibles en este momento").
Salida:	<p>Lista de ligas disponibles (<i>Array<Objeto></i>), donde cada objeto contiene:</p> <ul style="list-style-type: none"> • Nombre de la liga (<i>String</i>) • Logotipo de la liga (<i>String</i> - URL de la imagen) • País de origen (<i>String</i>) • Temporada en curso (<i>String</i>)
Precondiciones:	La aplicación debe tener conexión con la API o base de datos que almacena la información sobre las ligas.
Postcondiciones:	<ul style="list-style-type: none"> • El usuario visualiza la lista de ligas disponibles con su información relevante. • El usuario puede filtrar o buscar ligas según su preferencia.

	<ul style="list-style-type: none"> • El usuario puede seleccionar una liga para ver más detalles o sus equipos asociados.
Reglas de negocio:	<ul style="list-style-type: none"> • Se debe permitir la búsqueda de ligas por nombre. • Se debe permitir el filtrado por país o categoría.
Prioridad:	Baja (<i>condicional</i>)
Fuente:	Requerimientos del cliente
Referencias a componentes:	API Football.

Código:	RF-07
Título:	Consulta de Partidos de un Equipo
Descripción:	La aplicación debe mostrar los partidos jugados y programados de un equipo con detalles como fecha, hora, estadio, marcador (si aplica) y eventos clave. Debe permitir filtrar por competición, fecha o condición de local/visitante y ofrecer notificaciones para partidos próximos.
Entrada:	Selección de un equipo. (<i>Object</i>) El usuario elige un equipo desde el buscador, una lista de equipos o a través de una liga.
Proceso:	<ol style="list-style-type: none"> 1. El usuario selecciona un equipo desde el buscador, la lista de equipos o dentro de una liga. 2. La aplicación envía una solicitud al servidor para obtener los partidos jugados y programados del equipo seleccionado. 3. Se reciben y procesan los datos, organizándolos en dos secciones: <ul style="list-style-type: none"> ● Partidos jugados: Se muestran con el marcador final y eventos clave. ● Partidos programados: Se presentan con la fecha, hora y competición correspondiente. 4. Se habilitan opciones de filtrado para mejorar la experiencia del usuario: <ul style="list-style-type: none"> ● Filtrar por temporada o torneo. ● Filtrar por partidos de local o visitante. ● Filtrar por fechas específicas. 5. Se ofrece la opción de activar notificaciones para partidos próximos del equipo seleccionado. 6. Si no hay partidos registrados para el equipo, se muestra un mensaje informativo ("No hay partidos registrados para este equipo").
Salida:	<p>Lista de partidos del equipo (<i>Array<Objeto></i>), donde cada objeto contiene:</p> <ul style="list-style-type: none"> ● Fecha del partido (<i>Date</i>) ● Hora del partido (<i>Time</i>) ● Nombre del equipo local (<i>String</i>) ● Nombre del equipo visitante (<i>String</i>) ● Estadio (<i>String, opcional</i>) ● Competición (<i>String</i>) ● Estado del partido (<i>String: "Programado", "En curso", "Finalizado"</i>)

	<ul style="list-style-type: none"> ● Marcador final (<i>String, opcional si el partido ha terminado</i>) ● Eventos clave (<i>Array<Object>, opcional, si el partido ya se jugó</i>)
Precondiciones:	Deben existir datos sobre los partidos del equipo en la API o base de datos.
Postcondiciones:	<ul style="list-style-type: none"> ● El usuario puede visualizar el historial de partidos del equipo seleccionado. ● El usuario puede aplicar filtros para encontrar partidos específicos. ● El usuario puede activar notificaciones para partidos futuros.
Reglas de negocio:	<ul style="list-style-type: none"> ● Se debe permitir filtrar por temporada o torneo. ● Se debe permitir filtrar por condición de local o visitante.
Prioridad:	Media (<i>Esencial</i>)
Fuente:	Requerimientos del cliente
Referencias a componentes:	API Football

Enlaces

- Enlace Prototipo en Figma

[WikiFut – Figma](#)

- Enlace Tablero Trello

[Mi tablero de Trello](#)

- Enlace Repositorio GitHub

[StivenHenao/WikiFut: WikiFut is an Android application developed in Kotlin and Jetpack Compose that displays real-time football information using the API-Football and MVVM architecture.](#)