



MANUAL TÉCNICO

**PROTOTIPO APLICACIÓN MÓVIL DE SERVICIOS DE CONSULTA DEL MÓDULO
ESTUDIANTEL DE CONTROL ACADÉMICO PARA INSTITUCIONES DE
EDUCACIÓN SUPERIOR**

**GENESIS A. GUERRERO MARTINEZ
JOSE LUIS ECHAVEZ MORALES
ANDRES VASQUEZ ZUÑIGA**

**Director
ING. LUIS GARRIDO BARRIOS**

**UNIVERSIDAD DEL MAGDALENA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
SANTA MARTA, D.T.C.H.
2016**

Los derechos de esta obra han sido reservados conforme a la ley por LA UNIVERSIDAD DEL MAGDALENA, por tanto sus textos y gráficos no pueden reproducirse por medio alguno sin previa autorización escrita del autor.

Documento: Manual Técnico Prototipo app móvil del modulo de control académico en IES, caso de estudio Universidad Del Magdalena.

Control de Versiones

Versión	Fecha	Descripción	Autores
1.0	2016	Elaboración del manual técnico	GENESIS GUERRERO MARTINEZ JOSE LUIS ECHAVEZ ANDRES VASQUEZ

Contenido

Introducción	7
Análisis del sistema	7
Análisis de datos	Error! Bookmark not defined.
Requerimientos técnicos	10
Herramientas y tecnologías utilizadas para los servicios web	10
Herramientas y tecnologías utilizadas para la aplicación móvil	Error! Bookmark not defined.
Requerimientos funcionales	11
Requerimientos no funcionales	12
Alcance del sistema	13
Diseño del sistema	13
Descripción general	Error! Bookmark not defined.
Arquitectura del sistema	13
Arquitectura física.....	13
Arquitectura lógica.....	14
Esquema de Seguridad	15
Modelo de casos de uso	16
Descripción general de los actores.....	16
Diagramas del modelo de casos de uso.....	17
Modelo de datos.....	25
Diseño de interfaz.....	30
Implementación	37
Entorno utilizado	Error! Bookmark not defined.
Parte servidor	Error! Bookmark not defined.
Parte cliente	38
Pruebas.....	41
Prueba de servicios web.....	41
Pruebas en dispositivos móviles	43
Despliegue	44
Servidor.....	Error! Bookmark not defined.
Correr el servidor	46
App	49
Optimización para despliegue	49
Publicación en tiendas de aplicaciones	Error! Bookmark not defined.

Lista de figuras

Figura 1 Arquitectura física	14
Figura 2 Arquitectura lógica	14
Figura 3 Funcionamiento token	15
Figura 4 Diagrama casos de uso	17
Figura 5 Login mockup.....	30
Figura 6 Menu mockup.....	30
Figura 7 Notas mockup.....	31
Figura 7 Horario mockup.....	31
Figura 9 Noticias mockup.....	32
Figura 9 Historial mockup	32
Figura 11 Herramientas mockup.....	33
Figura 12 Calculadora mockup.....	33
Figura 13 Ubicaciones mockup	34
Figura 14 Alarma de clase mockup	35
Figura 15 Contactos mockup	36
Figura 16 Estructura de archivos en el servidor.....	37
Figura 17 Estructura de archivos del app.....	39
Figura 18 Login vista final.....	40
Figura 19 Menu vista final.....	40
Figura 20 Horario vista final.....	40
Figura 18 Splash vista final.....	40
Figura 22 Ubicaciones vista final	41
Figura 23 Noticias vista final.....	41
Figura 24 Resultados pruebas API.....	42

Lista de tablas

Tabla 1 Caso de uso: Iniciar sesión.....	18
Tabla 2 Caso de uso: Recordar contraseña	19
Tabla 3 Caso de uso: Cerrar sesión.....	19
Tabla 4 Caso de uso: Consultar perfil.....	19
Tabla 5 Caso de uso: Ver noticias.....	20
Tabla 6 Caso de uso: Consultar horario	20
Tabla 7 Caso de uso: Consultar notas	20
Tabla 8 Caso de uso: Consultar historial	21
Tabla 9 Caso de uso: Actualizar	21
Tabla 10 Caso de uso: Filtrar por asignatura	22
Tabla 11 Caso de uso: Búsqueda rápida	22
Tabla 12 Caso de uso: Listar herramientas	22
Tabla 13 Caso de uso: Calcular notas	23
Tabla 14 Caso de uso: Consultar ubicaciones.....	23

Tabla 15 Caso de uso: Configurar alarma.....	24
Tabla 16 Caso de uso: Ver ayuda	25
Tabla 17 API login.....	25
Tabla 18 API personal	27
Tabla 19 API horario	28
Tabla 20 API notas.....	28
Tabla 21 API historial.....	29
Tabla 22 API Foto.....	29
Tabla 23 Entorno de pruebas android	43
Tabla 24 Entorno de pruebas iOS	44
Tabla 25 Resultado de pruebas funcionales.....	45
Tabla 26 Lista de bugs	46

Introducción

En el presente manual técnico tiene como finalidad describir el diseño del prototipo aplicación móvil de servicios de consulta del módulo estudiantil de control académico para instituciones de educación superior – de ahora en adelante AyRE móvil – tomando como caso de estudio a la Universidad Del Magdalena.

La implementación de AyRE móvil es orientada a los estudiantes como usuarios finales y se basa en una adaptación para teléfonos móviles y tabletas de los servicios de consulta académica que actualmente se ofrecen en su portal web. La solución se apoya en las ultimas tecnologías de desarrollo móvil hibrido para producir dos aplicaciones nativas tanto para Android (desde la versión 4.0.3) como también iOS (desde la versión 7).

Análisis del sistema

Análisis de datos

La información que se maneja en la aplicación esta categorizada en cinco servicios en los que se recoge la información necesaria y oportuna para una correcta visualización de los datos.

Nota: los campos en *cursiva* son opcionales ya que pueden variar entre universidades.

Información académica, personal y otros.

- **Información Personal:**
 - Nombres Completos.
 - Código Estudiantil: (debe ser único, usado para iniciar sesión en la aplicación).
 - Numero de Documento.
 - Sexo

- Edad
- Estrato
- Celular
- Teléfono
- Fecha Nacimiento
- Email.
- Programa académico.
- Promedio Acumulado.
- Estado Académico (nuevo, antiguo).
- Situación académica (normal, readmisión, fbra etc.).
- Foto. (Imagen del Estudiante).
- *Email Institucional.*
- *Ciudad de Origen.*
- *Ciudad de Residencia.*
- *Colegio: (donde terminó la secundaria)*
- *Año terminación décimo.*
- *Valor pensión decimo.*
- *Ano terminación undécimo.*
- *Valor pensión undécimo.*
- *Código del Programa cursado.*
- *Acuerdo de Liquidación.*
- *Readmisiones.*
- **Información académica**
 - Materias cursados normalmente.
 - Créditos cursados normalmente.
 - Materias Homologadas.
 - Créditos Homologados.
 - Materias Vacacionales.
 - Créditos Vacacionales.
 - Materias Perdidas.
 - Créditos Perdidos.
 - *Materias Distancia.*
 - *Créditos Distancia.*
 - *Materias Validaciones.*
 - *Créditos validaciones.*
 - *Materias Reconocidos.*
 - *Créditos Reconocidos.*
- **Requisitos de Grado.**
 - Requisito de Pre practicas.

- Requisito de Practicas.
- Requisito de Grado.
- *Suficiencia de Ingles.*
- *Judicatura Monografía.*
- *Asignaturas Optativas.*
- *Proyecto Académico.*
- ***Paz y salvos***
 - *Departamento de Admisiones.*
 - *Biblioteca.*
 - *Cartera.*
 - *Clínica.*
 - *Laboratorios.*
 - *Anomalías.*

Horario Académico.

- Por cada Asignatura matriculada se necesitan los siguientes datos:
 - Nombre de la Asignatura.
 - Tipo de Asignatura.
 - *Tipo de Registro.*
 - Código de Asignatura.
 - Docente.
 - *Grupo.*
 - Domingo: (*Hora y Lugar*)
 - Lunes. (*Hora y Lugar*)
 - Martes. (*Hora y Lugar*)
 - Miércoles. (*Hora y Lugar*)
 - Jueves. (*Hora y Lugar*)
 - Viernes. (*Hora y Lugar*)
 - Sábado. (*Hora y Lugar*)

Notas.

- Por cada Asignatura matriculada se necesitan los siguientes datos.
 - Nombre de la Asignatura.
 - Código de Asignatura.
 - Nota Corte 1.
 - Nota Corte 2.
 - Nota Corte 3.
 - Nota Final.
 - *Nota Habilitación.*
 - *Nota Definitiva.*

Historial o Sabana de Calificaciones.

- Por cada periodo académico cursado (semestre) se necesitan los siguientes datos.
 - Nombre Periodo
 - Promedio Acumulado.
 - Promedio Semestral.
 - Asignaturas Cursadas:
 - Nombre de la Asignatura.
 - Código de Asignatura.
 - Nota Corte 1.
 - Nota Corte 2.
 - Nota Corte 3.
 - Nota Final.
 - Créditos.
 - *Nota Habilitación.*
 - *Nota Definitiva.*

Requerimientos técnicos

En este apartado se van a listar los requisitos que se deben cumplir para poder hacer un buen uso de este prototipo.

- Se deberá contar con un dispositivo con sistema operativo Android (v4.0.3 o superior) o un iPhone/iPad con iOS(v7 o superior).
- Para los dispositivos Android se requiere como mínimo una capacidad de almacenamiento libre de 5.9 MB y que la terminal posea por lo menos 512MB de memoria RAM.
- Para iPhone/iPad se deberá contar con un espacio libre de almacenamiento de al menos 6.2 MB.
- Se deberá contar con una conexión a internet ya sea vía Wi-Fi o por datos proporcionados por su proveedor de red.
- El dispositivo debe tener GPS activado para usar el modulo de “Ubicaciones del campus”

Herramientas y tecnologías utilizadas para los servicios web

Balsamic Mockups: Es una herramienta para diseñar interfaces gráficas ofrecida por Balsamiq Studios, LLC. Está disponible para los sistemas operativos Mac OS y Windows.

Apache Cordova: Es una plataforma para construir aplicaciones móviles nativas usando html5, css y javascript. Funciona como un contenedor sobre el que corren aplicaciones web escritas en html5, css, javascript, ya que las aplicaciones web no pueden usar funcionalidades nativas del teléfono como Camara, GPS, Acelerómetro, etc. con Apache Cordova se pueden lograr todo esto

y además empaquetar la aplicación en el formato instalador del dispositivo: APK(Android), IPA(IOS), XAP(Windows Phone).

Ionic framework: Es una plataforma de desarrollo la cual funciona sobre *AngularJs* y *Apache Cordova* y provee herramientas y servicios para el desarrollo de aplicaciones móviles híbridas para diferentes sistemas operativos móviles: IOS, Android, Window Phone entre otros, usando tecnologías como CSS, HTML5 y Sass. Para este proyecto se uso en su versión 1.1.1.

Gulp: Es un gestor de tareas para javascript el cual automatiza el flujo de trabajo. Llevando a cabo funciones como la minificación de los archivos javascript, css y hasta html para un mejor funcionamiento en producción. Además ayuda automatizando tareas como:

- Agrupación y minificación de librerías y hojas de estilos.
- Recargar el navegador cuando se hace una modificación al archivo.
- Correr pruebas unitarias.
- Correr análisis de código.
- Compilación de código Sass/Less.
- Copia de archivos y directorios.

Genymotion: Es un emulador de Android el cual ayuda a probar aplicaciones en múltiples dispositivos sin tenerlos físicamente, prestando casi las mismas características como sensores y cámaras, ofreciendo un entorno virtual de Android. En comparación con la herramienta oficial de Google para este propósito presenta mejoras en cuanto a velocidad y facilidad de instalación y uso.

iOS Simulator: Es un simulador de dispositivos para IOS el cual permite rápidamente correr y probar los prototipos de las aplicaciones móviles durante el proceso de desarrollo. Corre en Mac simulando iPhone, iPad, Apple Watch, o Apple TV. Dentro de sus principales funcionalidades se pueden destacar:

- Encontrar la mayoría de problemas de diseño y funcionales.
- Probar la aplicación usando herramientas de desarrollo.

Requerimientos funcionales

Los requerimientos de la aplicación móvil son los siguientes:

- **Ingreso a la aplicación:** El usuario debe ingresar a la aplicación con su código y contraseña para acceder al resto de funciones.
- **Mostrar perfil:** La aplicación proporcionará una vista con información personal y académica del usuario validado en el sistema. Se debe brindar la opción para refrescar la información de la base de datos.

- **Consultar horario:** La aplicación debe mostrar una vista con el horario de clases del usuario y permitir filtrar y sincronizar con la base de datos la información presentada.
- **Consultar notas:** La aplicación debe mostrar una vista con las notas del usuario agrupadas por asignatura y permitir filtrar y refrescar la información con la base de datos.
- **Consultar historial:** La aplicación debe mostrar una vista con el historial de notas del usuario agrupadas por semestre y asignatura. El sistema debe permitir el filtro de la lista por cualquier campo (semestre, nombre asignatura, etc) y refrescar la información con la base de datos.
- **Noticias:** La aplicación debe proporcionar un feed de noticias de interés sobre la universidad.
- **Calculadora de notas:** La aplicación debe suministrar una vista para calcular las notas y promedios sobre las notas actuales del usuario.
- **Ubicaciones del campus:** La aplicación debe ofrecer una opción para ver las ubicaciones de interés de la universidad sobre un mapa. El sistema debe ser capaz de ubicar y mostrar la posición actual del usuario mediante GPS.
- **Alarma de clase:** La aplicación debe proporcionar una opción para activar y desactivar una alarma como recordatorio para cada una de las asignaturas en el horario del usuario. El usuario podrá parametrizar el tiempo previo (en minutos) para el recordatorio. Estas alarmas deben ser almacenadas persistentemente en el dispositivo.
- **Contactos de interés:** La aplicación debe mostrar una lista de contactos de interés de la universidad con información de contacto disponible (Email, teléfono, fax).

Requerimientos no funcionales

Los requisitos no funcionales para la aplicación, es decir, los que no especifican el comportamiento del sistema, son:

- **Rendimiento:** La aplicación debe desempeñar su función de una manera fluida. Se debe buscar la experiencia de uso más agradable para el usuario.
- **Interfaz:** Clara y concisa. No debe dar lugar a la confusión del usuario y debe seguir los estándares de diseño de interfaces para móviles (iOS y Android).
- **Disponibilidad:** La app debe funcionar sin conexión a Internet, ya que algunos estudiantes tienen conexión a internet limitada.
- **Accesibilidad:** La app debe ser legible y tiene que seguir los patrones de accesibilidad de Google y iOS.

- **Seguridad:** Se debe garantizar la seguridad en el acceso a la información. La información personal solo puede ser accesible para el usuario autorizado para ello.
- **Usabilidad:** Cualquier estudiante debe ser capaz de utilizar la aplicación y acceder a toda la funcionalidad sin ningún tipo de restricción.
- **Estabilidad:** La app debe ser capaz de manejar los errores ocurridos durante la ejecución de la misma.
- **Mantenimiento:** La aplicación debe ser mantenida y actualizada, dando posibilidad a mejorar el rendimiento y la usabilidad en cualquier momento.
- **Optimización:** El consumo de batería y de datos debe ser optimo, y nunca dejar procesos huérfanos que consuman memoria y batería. El tiempo de ejecución debe ser mínimo, para mejorar los tiempos de respuesta y la experiencia de usuario.

Alcance del sistema

Prototipo de una aplicación móvil para Instituciones de Educación Superior **IES**, con el caso de uso de la Universidad del Magdalena y su departamento de admisiones **AYRE**. La aplicación móvil se centra en brindar a los estudiantes un plataforma para la fácil visualización de sus datos académicos como notas, horario de clases e historial. Además de sus datos personales almacenados por el campus y un conjunto de herramientas las cuales complementan y hacen de la aplicación una excelente utilidad para el estudiante.

Diseño del sistema

Arquitectura del sistema

Arquitectura física

La arquitectura predominante Cliente-Servidor ha dado paso a un nuevo modelo en el que hay un servidor web de API's que es consumido por varios servicios o aplicaciones. Este modelo es el propuesto para el prototipo de AyRE móvil, por lo tanto obtendrá la información académica codificada en JSON a través de un API REST que es el encargado de la persistencia de los datos; por otro lado, el cliente (App) manejará algunos datos localmente como la sesión y preferencias del usuario de las herramientas académicas.

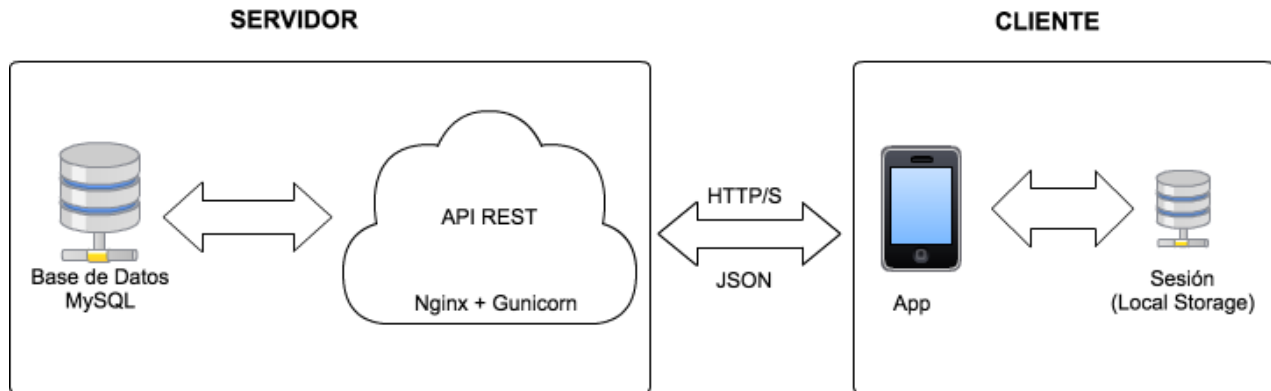


Figura 1 Arquitectura física

Arquitectura lógica

La arquitectura lógica de la aplicación móvil esta basada en el patrón MVVM (Model-View-ViewModel).

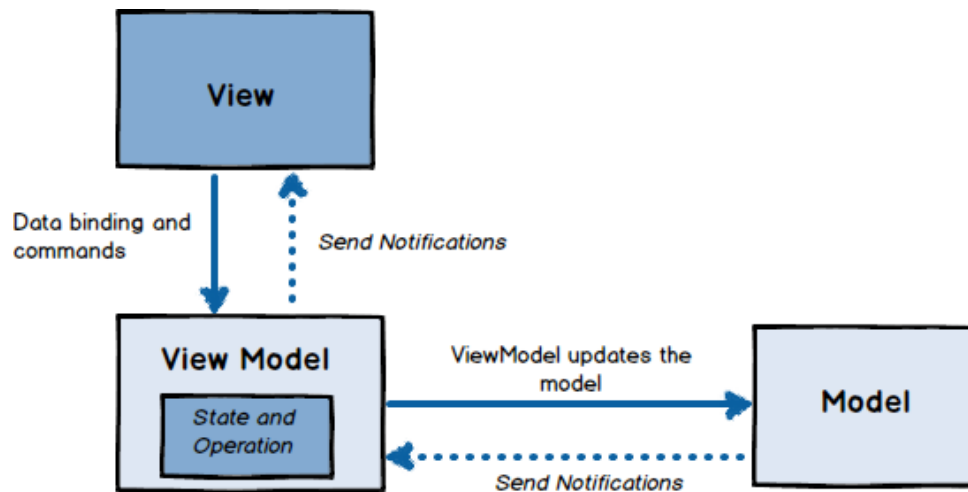


Figura 2 Arquitectura lógica

MVVC facilita la separación de desarrollo de la interfaz gráfica de usuario del desarrollo de la lógica de negocios (modelo de datos). Este modelo es similar al extendido patrón MVC (Model-View-Controller), la diferencia principal radica en el ViewModel (sustituye al Controller en MVC) que cuenta con una sincronización entre la vista y el modelo. A continuación se describen cada una de las capas de esta arquitectura:

Model: Es la capa de datos, la información académica y de autenticación es obtenida del API mediante llamadas AJAX a los servicios REST con lo que se alimentarán los modelos.

View: Como en el patrón MVC, la vista es la interfaz de usuario.

ViewModel: Es una abstracción de las propiedades publicas y comandos que expone la vista. En lugar de un controlador se tiene un binder, que media la comunicación entre la vista y el modelo, sincronizándose.

Esquema de Seguridad

El principal reto de seguridad en AyRE móvil es la autenticación y autorización de usuarios para controlar el acceso a la información, debido que el cliente móvil se alimenta de un API REST se tienen en cuenta los siguientes factores:

- El API acepta peticiones de cualquier dominio de origen (*). Se debe configurar CORS en el API para permitir peticiones desde cualquier terminal móvil. CORS (Cross-origin resource sharing) es un mecanismo que da controles de acceso a dominios cruzados para servidores web, lo que habilita la transferencia segura de datos mediante HTTP en dominios cruzados.
- Las API RESTful no tienen información de estado, por lo tanto no es posible almacenar la sesión en el servidor (cookies, BD, archivos).

En base a lo anterior, se optó por implementar un sistema de autenticación por token el cual se describe a continuación:

Funcionamiento:

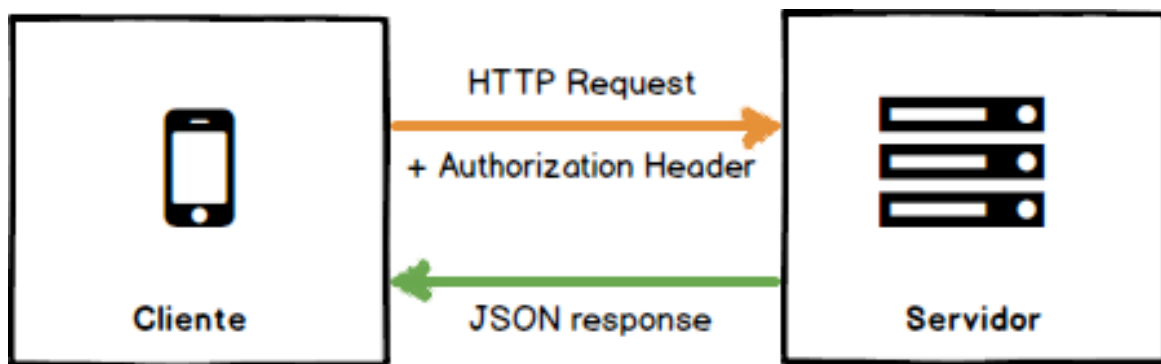


Figura 3 Funcionamiento token

1. El cliente debe autenticarse con sus credenciales (código y contraseña) y posteriormente el API devuelve un token de acceso.
2. El cliente debe enviar el token en el encabezado de autorización en las peticiones HTTP subsiguientes.
3. El API verifica si el token es válido y en caso de éxito devuelve la información solicitada, en caso de falla devuelve el código y mensaje del error.

Como se genera el Token:

Cuando un usuario se autentica el API genera una cadena de caracteres codificada con un algoritmo HMAC SHA-256.

Consideraciones:

- Para este prototipo se expone el API bajo el protocolo HTTP, pero para su uso en producción es necesario utilizar el protocolo HTTPS para cifrar la comunicación entre el cliente y el servidor y de esta manera evitar que información sensible pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión.
- Al no utilizar cookies para almacenar la información del usuario, podemos evitar ataques CSRF (Cross-Site request Forgery) que manipulen la sesión que se envía al servidor.
- Los tokens se configuran para que expiren después de un tiempo, lo que añade una capa adicional de seguridad.

Modelo de casos de uso

Descripción general de los actores

Usuario: Es un estudiante universitario activo, inactivo, o egresado

Diagramas del modelo de casos de uso

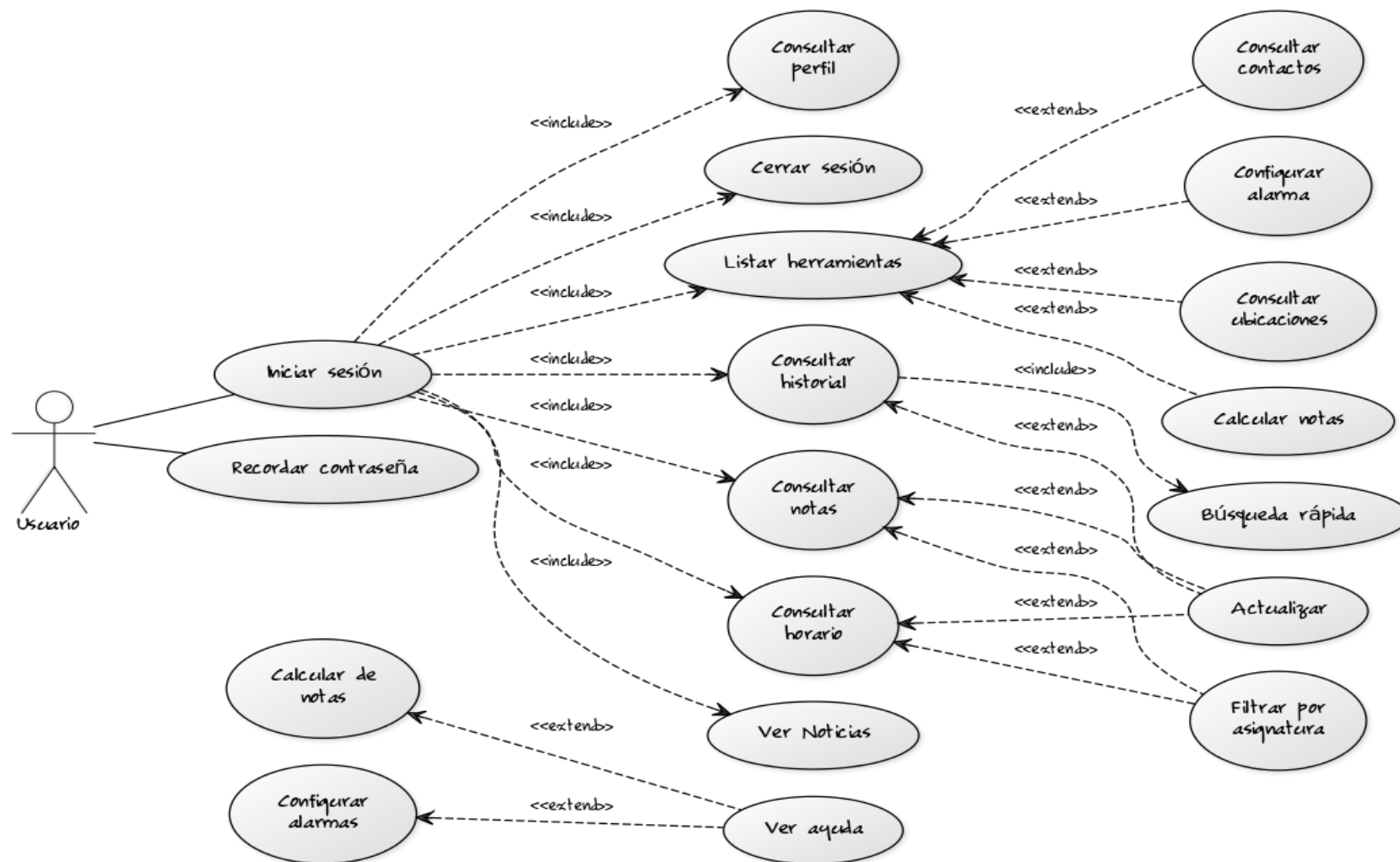


Figura 4 Diagrama casos de uso

Nombre	Iniciar sesión
Descripción	Validar el código y contraseña para permitir su acceso a la aplicación
Actores	Usuario
Precondiciones	El estudiante debe estar registrado en la base de datos con un código y contraseña validos.
Flujo Normal	<ol style="list-style-type: none"> 1. El app despliega un formulario para capturar el código estudiantil y la contraseña. 2. El actor digita código y contraseña y ejecuta la opción ingresar. 3. El sistema valido que el código y la contraseña corresponda a la existente en la base de datos. 4. El app despliega el perfil del usuario que se validó.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El código y contraseña proporcionados por el usuario no corresponden a los registrados en la base de datos. 4. Se muestra un mensaje de error.
Poscondiciones	El usuario se ha validado correctamente, permitiendo el acceso a las demás opciones del app.

Tabla 1 Caso de uso: Iniciar sesión

Nombre	Recordar contraseña
Descripción	Proporciona un método para recuperar la contraseña.
Actores	Usuario
Precondiciones	El estudiante debe estar registrado en la base de datos.
Flujo Normal	<ol style="list-style-type: none"> 1. El app despliega el formulario de inicio de sesión junto con la opción de “Recordar contraseña”. 2. El usuario ejecuta la opción “Recordar contraseña”. 3. El app abre el navegador predeterminado del dispositivo con la url de la pagina de la institución de recuperación de contraseña.
Flujo Alternativo	Ninguno

Poscondiciones	El usuario ha recordado su contraseña correctamente y puede volver a intentar iniciar sesión en el app.
-----------------------	---

Tabla 2 Caso de uso: Recordar contraseña

Nombre	Cerrar sesión
Descripción	Permite al usuario cerrar sesión, quedando sin acceso para visualizar algún modulo del app.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario abre el menú principal. 2. El usuario ejecuta la opción “Cerrar sesión” 3. El sistema procesa los datos. 4. El app retorna a la vista de inicio de sesión.
Flujo Alternativo	Ninguno
Poscondiciones	El usuario es redirigido al caso de uso de Iniciar sesión sin poder acceder a ningún modulo del app.

Tabla 3 Caso de uso: Cerrar sesión

Nombre	Consultar perfil
Descripción	Muestra la información personal y académica del usuario, estado, anomalías y paz y salvos.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. App retorna la vista de perfil después de que el usuario inicia sesión.
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de menú 2. El usuario ejecuta la opción perfil 3. El App retorna la vista de perfil
Poscondiciones	Ninguna

Tabla 4 Caso de uso: Consultar perfil

Nombre	Ver noticias
Descripción	Muestra noticias de interés sobre la universidad.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario abre el menú principal. 2. El usuario ejecuta la opción “Noticias”

	<ol style="list-style-type: none"> 3. El app carga una lista con noticias con una breve descripción. 4. El usuario hace “tap” en alguna noticia. 5. El app abre el navegador con la url de la noticia completa.
Flujo Alternativo	Ninguno
Poscondiciones	Ninguno

Tabla 5 Caso de uso: Ver noticias

Nombre	Consultar horario
Descripción	Muestra la información del horario en un formato de asignaturas por días.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	Ninguno
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de menú 2. El usuario ejecuta la opción Horario El App lo lleva a la vista de Horario
Poscondiciones	Ninguno

Tabla 6 Caso de uso: Consultar horario

Nombre	Consultar notas
Descripción	Muestra la información de las notas por asignaturas.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	Ninguno
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de menú 2. El usuario ejecuta la opción Notas El App lo lleva a la vista de Notas
Poscondiciones	Ninguno

Tabla 7 Caso de uso: Consultar notas

Nombre	Consultar historial
Descripción	Muestra la información del historial académico del usuario por periodos académicos.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	Ninguno

Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de menú 2. El usuario ejecuta la opción Historial El App lo lleva a la vista de Historial
Poscondiciones	Ninguno

Tabla 8 Caso de uso: Consultar historial

Nombre	Actualizar
Descripción	Actualiza los datos con el servidor
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se dirige a una de las siguientes vistas: Perfil, Horario, Notas o Historial. 2. El usuario presiona el botón de actualizar en la zona de encabezado. 3. El sistema sincroniza los datos con el servidor. <ol style="list-style-type: none"> 1. Se actualiza la fecha de la ultima actualización.
Flujo Alternativo	<ol style="list-style-type: none"> 3. El sistema arroja un mensaje de error debido a que el usuario no cuenta con conexión a internet.
Poscondiciones	El app actualiza la información localmente y estará disponible la próxima vez que el usuario consulte alguna vista y no cuenta con conexión a internet.

Tabla 9 Caso de uso: Actualizar

Nombre	Filtrar por asignatura
Descripción	Filtra la información que se visualiza a solo la asignatura seleccionada.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se dirige a una de las siguientes vistas: Horario, Notas, Calculadora. 2. Presiona sobre la barra del filtro o el icono. 3. Se muestra una lista de las asignaturas disponibles para filtrar. 4. Se selecciona una asignatura o todas Regresa a la vista principal y aplica el filtro.
Flujo Alternativo	Ninguno
Poscondiciones	Ninguno

Tabla 10 Caso de uso: Filtrar por asignatura

Nombre	Búsqueda rápida
Descripción	Filtra por la frase insertada en la lista que se muestra.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario se dirige a la vista de historial. 2. Ingresa su patrón de búsqueda en el campo. <p>La aplicación realiza el filtro sobre la información presentada.</p>
Flujo Alternativo	Si se borra completamente el campo de filtro se muestra toda la información normalmente.
Poscondiciones	Ninguno

Tabla 11 Caso de uso: Búsqueda rápida

Nombre	Listar herramientas
Descripción	Se muestran todas las herramientas disponibles con opción para iniciarlas o mostrar ayuda al respecto
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	Ninguno
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de menú 2. El usuario ejecuta la opción Herramientas. <p>El App lo lleva a la vista de Herramientas.</p>
Poscondiciones	El usuario selecciona algunas de las herramientas y es redirigido a ésta.

Tabla 12 Caso de uso: Listar herramientas

Nombre	Calcular notas
Descripción	El usuario puede modificar las notas que no hayan sido digitadas aun para observar el cambio en los promedios.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la herramienta calculadora de notas desde la vista de herramientas.

	<ol style="list-style-type: none"> 2. El usuario modifica las notas en la lista de asignaturas. 3. El sistema actualiza los valores de promedios. 4. El usuario presiona el botón de resetear las notas. <p>El sistema actualiza los valores de promedios.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario presiona el botón de generador de notas. 2. El sistema actualiza los valores de promedios 3. El usuario presiona el botón de resetear las notas. <p>El sistema actualiza los valores de promedios</p>
Poscondiciones	Ninguno.

Tabla 13 Caso de uso: Calcular notas

Nombre	Consultar ubicaciones
Descripción	Muestra en un mapa de Google Maps marcadores con información de las principales ubicaciones del campus.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la herramienta calculadora de notas desde la vista de herramientas. 2. El usuario navega por el mapa y los marcadores. 3. El usuario toca uno de los marcadores. <p>Se muestra una imagen del sitio con el nombre del mismo.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario selecciona el icono de ubicación. 2. El sistema le muestra su ubicación en un icono dentro del mapa. 3. El usuario selecciona en el icono de búsqueda. 4. El sistema le muestra una lista filtrable de los lugares. 5. El usuario hace un toque en alguno de los sitios. <p>Se muestra una imagen del sitio con el nombre del mismo.</p>
Poscondiciones	Ninguno.

Tabla 14 Caso de uso: Consultar ubicaciones

Nombre	Configurar alarma
Descripción	Configura alarmas de asignaturas en forma de notificaciones locales.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la herramienta alarma de asignaturas desde la vista de herramientas. 2. El usuario selecciona el botón de agregar alarma. 3. El sistema le muestra un modal para configurar la nueva alarma. 4. El usuario selecciona la asignatura, el estado y el tiempo de anticipación para la alarma. <p>El Usuario selecciona el icono de guardar alarma.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 5. El Usuario cancela el formulario de nueva alarma. 6. El usuario selecciona la alarma. 7. Se muestra un modal para confirmar el borrado de la alarma. <p>El usuario da un toque sobre el estado de la alarma para activarlo o desactivarlo.</p>
Poscondiciones	El sistema mostrará la notificación en el teléfono en la fecha y hora programada.

Tabla 15 Caso de uso: Configurar alarma

Nombre	Consultar contactos
Descripción	Muestra los datos por departamentos y facultades del campus universitario registrados en el sistema.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario selecciona la herramienta contactos desde la vista de herramientas. <p>El usuario navega por los diferentes contactos.</p>
Flujo Alternativo	<ol style="list-style-type: none"> 1. El usuario selecciona el icono de búsqueda.

	<ol style="list-style-type: none"> Se muestra un listado filtrado de las facultades y departamentos registrados en el sistema. El usuario selecciona uno de las facultades o departamentos. <p>El sistema muestra la facultad resaltada.</p>
Poscondiciones	Ninguno.

Nombre	Ver ayuda
Descripción	Muestra información de guía para las utilidades.
Actores	Usuario
Precondiciones	El usuario debe estar previamente validado en el sistema (Iniciar sesión).
Flujo Normal	<ol style="list-style-type: none"> El usuario selecciona la opción de ayuda para alguna de las herramientas. <p>Se visualiza la ayuda para esta herramienta.</p>
Flujo Alternativo	Ninguno.
Poscondiciones	Ninguno.

Tabla 16 Caso de uso: Ver ayuda

Modelo de datos

El acceso a los datos del sistema esta dado por un API REST que proporciona endpoints (URLS finales) para la autenticación de usuarios y obtención de la información académica necesaria para el app.

A continuación, se describe los endpoints del API:

Url raíz: <http://ayreprototype.tk/>

URL	/auth/login/
Método	POST
Campos	<ul style="list-style-type: none"> - username: charfield - password: charfield
Descripción	Obtiene el token de usuario a través del codigo y contraseña.
Respuesta	<p>200 OK:</p> <pre>{ "token": "****" }</pre> <p>404 NOT FOUND:</p> <pre>{ "message": "User doesn't exist" }</pre> <p>401 UNAUTHORIZED:</p> <pre>{ "message": "Invalid login info" }</pre>

Tabla 17 API login

URL	/personal/
Método	GET
Campos	<i>Authorization: Token ***</i>
Descripción	Obtiene la información personal del usuario validado en el sistema. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=***
Respuesta	200 OK: <pre>{ "personal": { "CODEST": "-", "NOMBRES": "-", "NUMERO_DOCUMENTO": "-", "SEXO": "-", "EDAD": "-", "ESTRATO": "-", "CELULAR": "-", "TELEFONO": "-", "FECHA_NACIMIENTO": "-", "EMAIL": "-", "EMAIL_INST": "-", "CIUDADORIGEN": "-", "CIUDADRESIDENCIA": "-", "COLEGIO": "-", "DECIMO": "-", "PENSION_10": "-", "UNDECIMO": "-", "PENSION_11": "-", "CODPRG": "-", "PROGRAMA": "-", "PROM_ACUM": "-", "ESTADO": "-", "SIT_ACAD": "-", "ACU_LIQ": "-", "READMISIONES": "-" }, "academicos": { "MATS_NORMALES": "-", "CRED_NORMALES": "-", "MATS_HOMOLOGADOS": "-", "CRED_HOMOLOGADOS": "-", "MATS_VACACIONALES": "-", "CRED_VACACIONALES": "-", "MATS_VALIDACIONES": "-" } }</pre>

	<pre> "CRED_VALICACIONES": "-", "MATS_RECONOCIDOS": "-", "CRED_RECONOCIDOS": "-", "MATS_DISTANCIA": "-", "CRED_DISTANCIA": "-", "MATS_PERDIDOS": "-", "CRED_PERDIDOS": "-", }, "requisitos_grado": { "REQUISITO_PREPRACTICA": "-", "REQUISITO_PRACTICA": "-", "REQUISITO_GRADO": "-", "SUFICIENCIA_INGLES": "-", "JUDICAT_MONOGRAFIA": "-", "ASIGNATURAS_OPTATIVAS": "-", "PROYECTO_ACADEMICO": "-" }, "paz_y_salvos": { "ADMISIONES": "-", "BIBLIOTECA": "-", "CARTERA": "-", "CLINICAS": "-", "LABORATORIOS": "-", "ANOMALIAS": "-" } } } 401 UNAUTHORIZED: {"message": "Invalid Token - Authorization Required"} </pre>
--	---

Tabla 18 API personal

URL	/schedule/
Método	GET
Campos	<i>Authorization: Token ***</i>
Descripción	Obtiene el horario del usuario validado. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=***
Respuesta	200 OK: <pre> [{ "ASIGNATURA": "-", "TIPO_ASIGNATURA": "-", "TIPO_REGISTRO": "-", "CODMAT": "-", </pre>

	<pre> "DOCENTE": "-", "GRUPO": "6" "DOMINGO": "11-12
Lugar 1 ; 15-18
lugar 2", "JUEVES": "-", "MARTES": "-", "VIERNES": "-", "LUNES": "-", "SABADO": "-", "MIERCOLES": "-", }, ...] 401 UNAUTHORIZED: {"message": "Invalid Token - Authorization Required"} </pre>
--	--

Tabla 19 API horario

URL	/scores/
Método	GET
Campos	<i>Authorization: Token ***</i>
Descripción	Obtiene las notas del usuario validado en el sistema. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=***
Respuesta	<p>200 OK:</p> <pre> [{ "ASIGNATURA": "-", "NOTA1": "-", "NOTA3": "-", "NOTA2": "-", "FINAL": "-", "HAB": "-", "DEF": "-", "CODIGO": "-", }, ...] 401 UNAUTHORIZED: {"message": "Invalid Token - Authorization Required"} </pre>

Tabla 20 API notas

URL	/history/
Método	GET

Campos	<i>Authorization: Token ***</i>
Descripción	Obtiene el historial de notas del usuario validado en el sistema. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=***
Respuesta	200 OK: <pre>[{ "periodo": "-", "prom_acumulado": "-", "prom_semestral": "-", "asignaturas": [{ "ASIGNATURA": "-", "NOTA1": "-", "NOTA2": "-", "NOTA3": "-", "NOTA_FIN": "-", "NOTA_HAB": "-", "NOTA_DEF": "-", "TREG": "NOR", "COD_MATERIA": "-", "CRED": "-", }, ...] }, ...]</pre> 401 UNAUTHORIZED: {"message": "Invalid Token - Authorization Required"}

Tabla 21 API historial

URL	/photo/
Método	GET
Campos	<i>Authorization: Token ***</i>
Descripción	Obtiene el avatar del usuario validado en el sistema. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=***
Respuesta	200 OK: image/jpeg 401 UNAUTHORIZED: Unauthorized

Tabla 22 API Foto

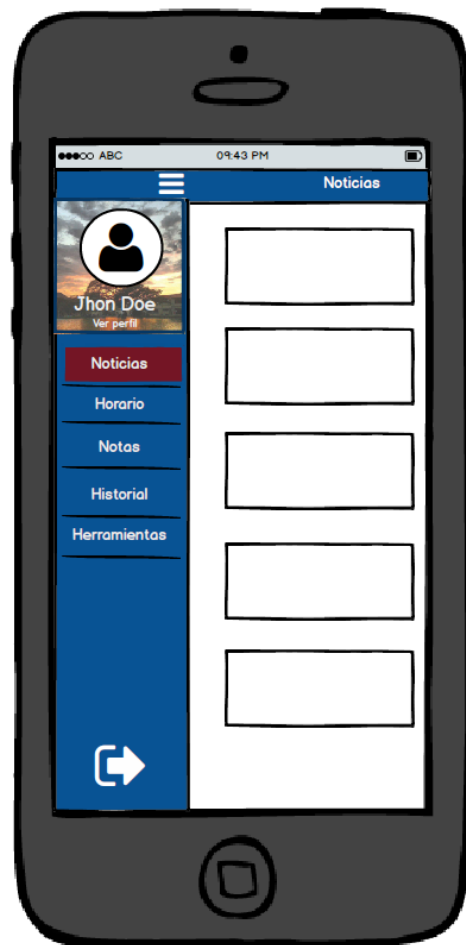
Diseño de interfaz

A continuación se presentan los diseños iniciales (mockups) de la interfaz grafica del app, estos fueron desarrollados con la herramienta Balsamiq Mockups 3.



Login

Figura 5 Login mockup



Menu

Figura 6 Menu mockup



Horario

Figura 8 Horario mockup



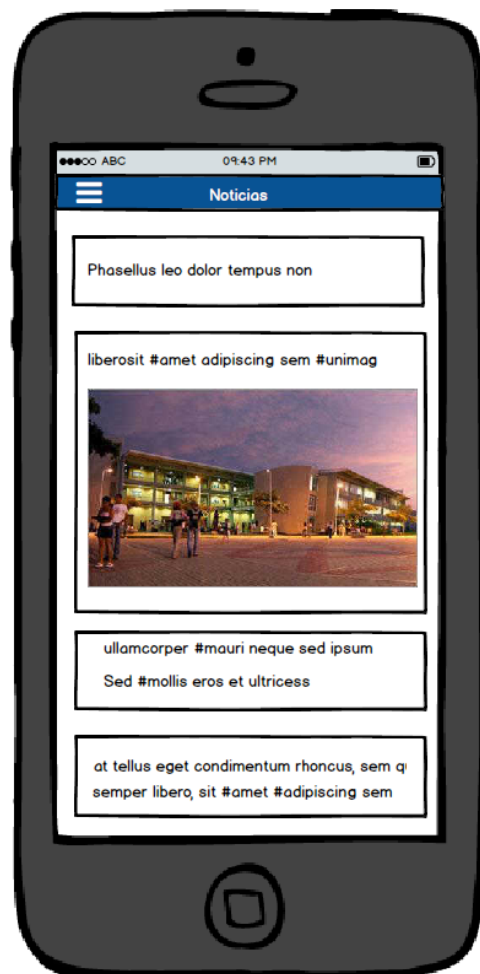
Notas

Figura 7 Notas mockup



Historial (sabana)

Figura 10 Historial mockup



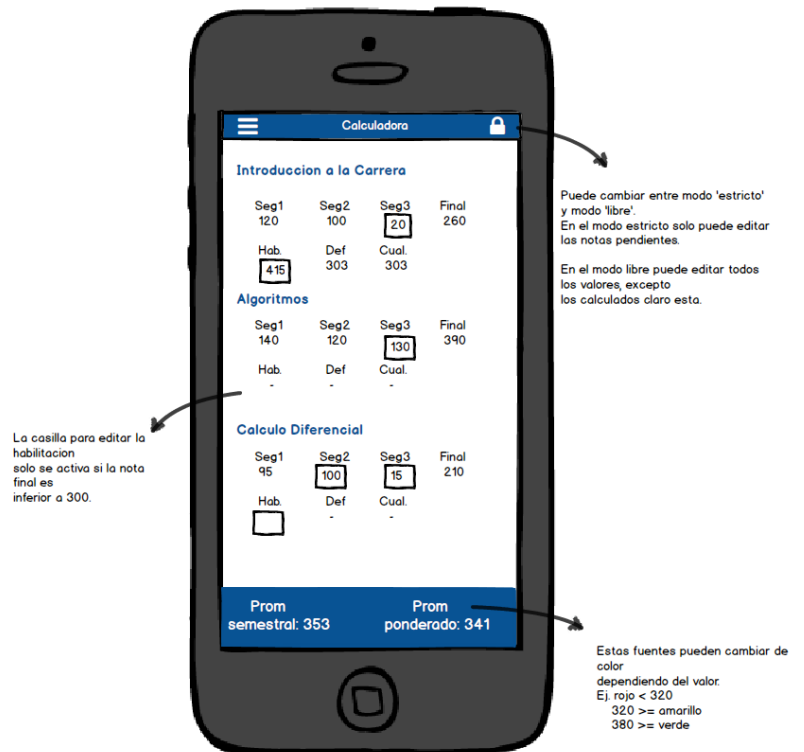
Noticias

Figura 9 Noticias mockup



Menu Herramientas

Figura 11 Herramientas mockup



Calculadora de notas

Figura 12 Calculadora mockup



Ubicaciones del campus

Figura 13 Ubicaciones mockup



Figura 14 Alarma de clase mockup



Contactos de interés

Figura 15 Contactos mockup

Implementación

Parte servidor

El servidor de prueba para el api permite la creación de los usuarios, inicio de sesión, crear o editar información para cada usuario como datos personales, historial, horario, y notas.

Flask:

Se usó la siguiente arquitectura en el desarrollo de este servicio:

```
</ayre/prototype-ayre-api/
├── app/
│   ├── auth/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   └── resources.py
│   ├── base/
│   │   ├── __init__.py
│   │   ├── decorators.py
│   │   └── models.py
│   ├── history/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   └── resources.py
│   ├── personal/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   └── resources.py
│   ├── schedule/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   └── resources.py
│   ├── scores/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   └── resources.py
│   ├── templates/
│   │   └── index.html
├── migrations/
├── uploads/
├── config.py
├── db.sqlite
├── LICENSE
├── manage.py*
└── README.md
```

Figura 16 Estructura de archivos en el servidor

En la carpeta principal app.

auth: para la creación y manejo de los usuario.

base: funciones útiles reutilizables en todos los demás servicios.

history, personal, schedule, scores: modelos con la definición de los datos, y resources para la definición de las peticiones.

templates: raíz del servicio con información de la ayuda para el api.

uploads: directorio para guardar las imágenes de los usuarios.

app/_init_.py: registro e inicializacion de los servicios para el uso de flask.

Para la consulta a la base de datos se uso el ORM para flask **SQLAlchemy** que ofrece que variedad de manejo para base de datos y relaciones.

Parte cliente

Estructura de los archivos

Se organizaron los códigos fuentes por características en lugar del modelo tradicional de carpetas de modelos, controladores, vistas, servicios, etc. La estructura por características facilita la modularización de la aplicaciones organizándolas en funciones específicas haciéndola mas sencilla para escalar y mantener.

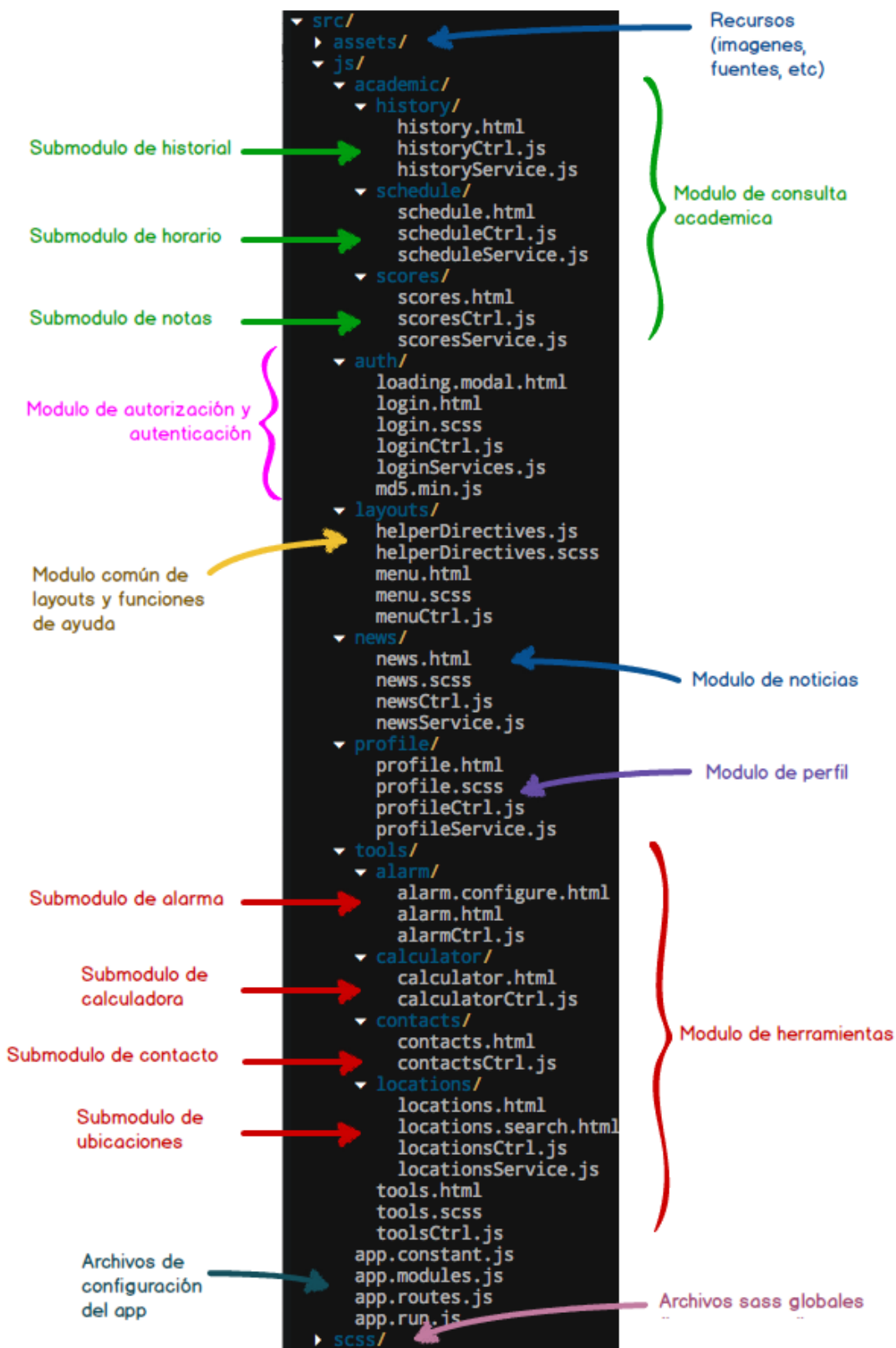


Figura 17 Estructura de archivos del app

Vistas finales



Figura 21 Splash vista final



Figura 18 Login vista final

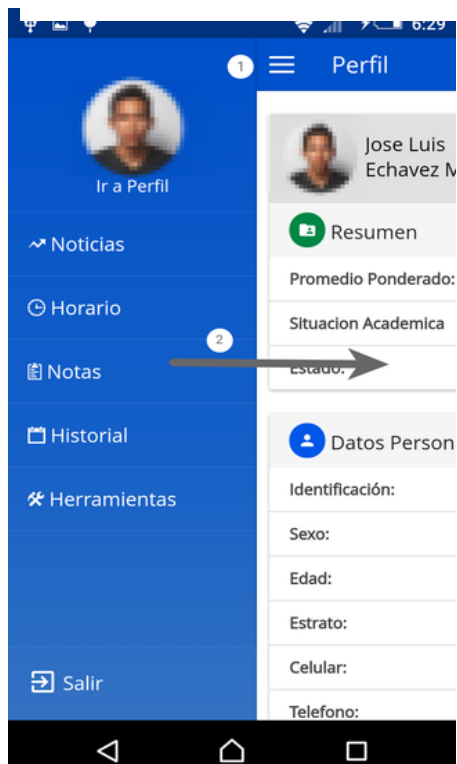


Figura 19 Menu vista final

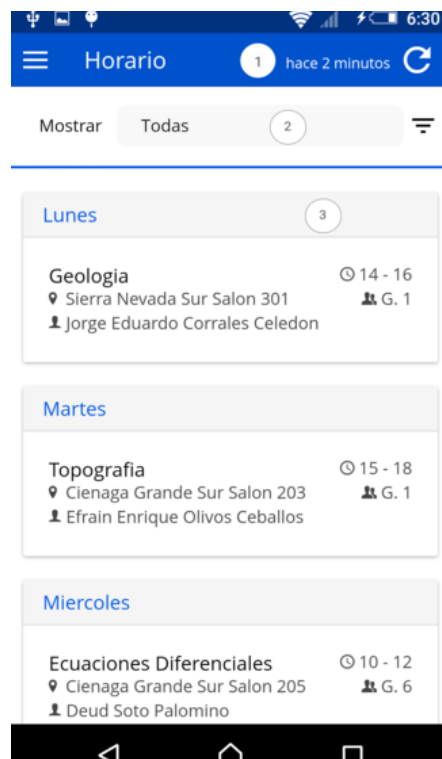


Figura 20 Horario vista final

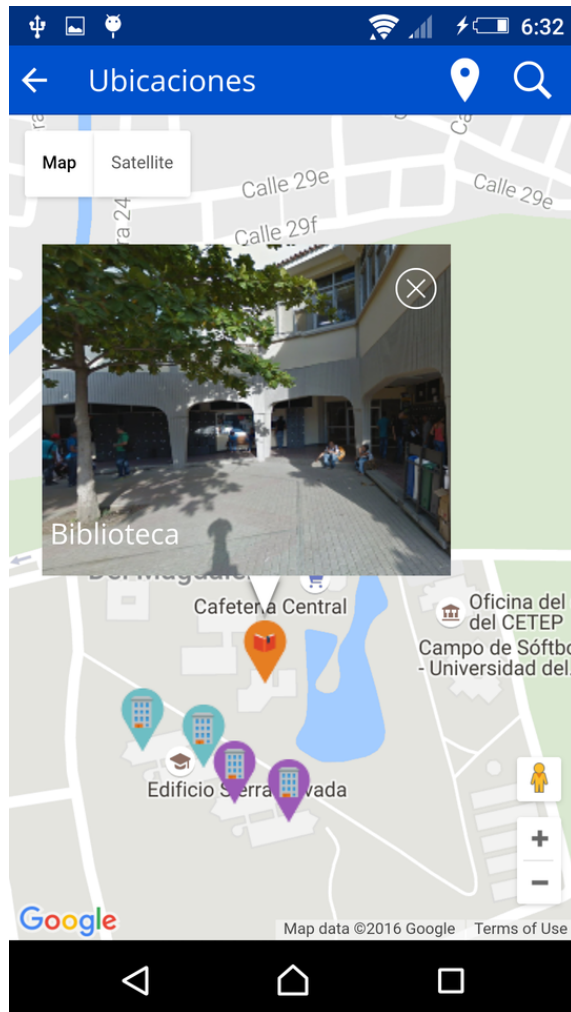


Figura 22 Ubicaciones vista final



Figura 23 Noticias vista final

Pruebas

Prueba de servicios web

Para las pruebas del API REST se utilizó el popular servicio en línea de pruebas automatizadas vREST de la empresa Optimizory Technologies Pvt. Ltd.

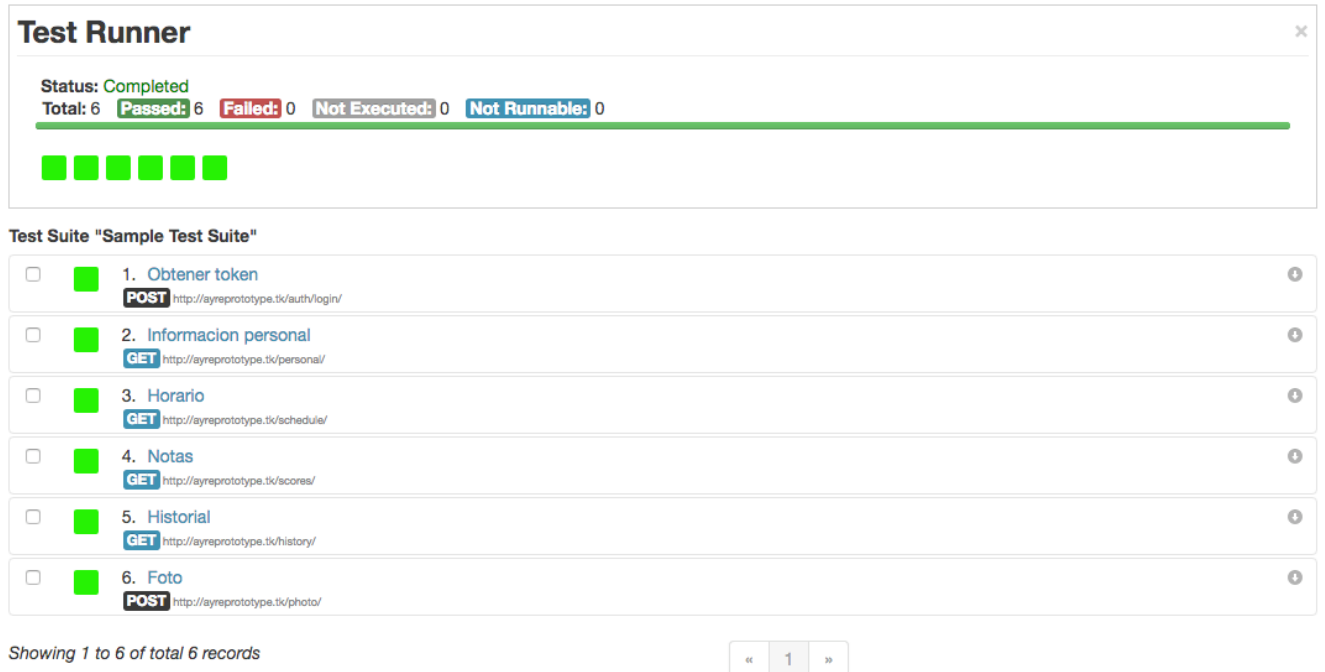


Figura 24 Resultados pruebas API

Como se observa en la figura, todos los endpoints del API pasaron correctamente las pruebas, devolviendo un código de estado HTTP de 200.

Las características del servidor donde está alojado el API en el momento de las pruebas son las siguientes:

Proveedor: Digital Ocean

Tipo: Cloud

Memoria: 512MB

Procesador: 1 Core

Disco: 20GB SSD

Transferencia: 1TB

Pruebas en dispositivos móviles

Se diseñaron las pruebas antes de su implementación, como lo indica la metodología Mobile-D. A continuación se presenta los entornos utilizados y los resultados de las pruebas:

Entorno

Se tomó variedad de equipos de alta, media y baja gama.

Android

Resolución	Dispositivo	S.O	GPU	RAM	CPU	Resultado
1080x1920	Sony Xperia Z2a	Android v4.4.2 (KitKat)	Quad-core 2.3 GHz Krait 400	3 GB	Quad-core 2.3 GHz Krait 400	Pasó
480x800	Samsung Galaxy J1	Android v4.4.4 (KitKat)	Mali-400	512 MB	Dual-core 1.2 GHz Cortex-A7	Pasó
480x854	Huawei Ascend G510	Android, v4.1 (Jelly Bean)	Adreno 203	512 MB	Dual-core 1.2 GHz Cortex-A5	Pasó
480x800	iPro Wave 4.0	Android v4.2.2 (Jelly Bean)	-	512 MB	Dual-core 1.0 Ghz	Pasó
720x1280	Motorola Moto G X1032	Android OS, v4.3 (Jelly Bean)	Adreno 305	1 GB	Quad-core 1.2 GHz Cortex-A7	Pasó
600x1024	Samsung Galaxy Tab 3	Android v4.1.2 (Jelly Bean)	Power VR SGX540	1 GB	Dual-core 1.2 GHz Cortex-A9	Pasó

Tabla 23 Entorno de pruebas android

iOS

Dispositivo	O.S	Resolución	Resultado
iPhone 6	iOS 8.0.2	Retina HD 1080x1920	Pasó
iPhone 6	iOS 9.3.0	Retina HD 1080x1920	Pasó
iPhone 5	iOS 8.0.2	Retina 640x1136	Pasó
iPhone 4s	iOS 8.0.2	640x960	Pasó
iPhone 4s	iOS 9.0.0	640x960	Pasó
iPad 2	iOS 7.1.2	768 x 1024	Pasó

Tabla 24 Entorno de pruebas iOS

Resultado de las pruebas

Lista de funcionalidades sometidas a prueba

Funcionalidad	Resultado
Comprobar conexión a internet en el dispositivo	Pasó
Comprobar aplicaciones estándar en el dispositivo	Pasó
Instalación del app	Pasó
Comprobar que el icono del app se muestre adecuadamente en todos los menús	Pasó
Nota de nombre de la aplicación bajo el icono en todos los menús	Pasó
Comprobar que se pueda salir de la aplicación correctamente	Pasó
Interfaz de usuario	
Todos los elementos gráficos, textos y animaciones tienen alta resolución.	Pasó
Comprobar que el scroll y swipe funciona en el app	Pasó
Las animaciones funcionan apropiadamente en todos los dispositivos	Falló
El menú principal se muestra correctamente	Pasó
Proceso de carga	
Comprobar que la barra de carga se muestra correctamente y funcione como se espera	Pasó
Mostrar mensajes de error en caso de problemas con la conectividad	Pasó
Inicio de sesión	
Comprobar la vista de login	Pasó
Comprobar la funcionalidad de login	Pasó
Comprobar la funcionalidad de recordar contraseña	Pasó
Información académica	
La vista de perfil se presenta correctamente	Pasó
Las tablas de horario, notas e historial se muestran apropiadamente	Pasó
Los filtros y búsquedas funcionan como se espera	Pasó

Herramientas académicas	
La calculadora de notas funciona como se espera	Pasó
Comprobar que el mapa de ubicaciones cargue correctamente	Pasó
Comprobar que se muestren mensaje de error en caso de problemas con GPS o conexión a internet en el mapa de ubicaciones	Falló
La alarma de clase carga los datos correctos y funciona como se espera	Pasó
Los contactos de interés se muestran correctamente.	Pasó
Las instrucciones de uso de cada herramienta se muestran apropiadamente	Pasó
Eventos en segundo plano	
Comprobar si los datos se almacenaron en dispositivo después de abrir el app por segunda vez sin conexión a internet.	Pasó
Eliminar la aplicación	
Comprobar la eliminación de la aplicación en ejecución	Pasó
Comprobar la aplicación después de removerla del dispositivo e instalarla nuevamente	Pasó

Tabla 25 Resultado de pruebas funcionales

Lista de bugs

Clave	Resumen	Prioridad	Entorno	Descripción
1	Algunas animaciones se ven forzadas debido a falta de recursos (GPU)	Media	Huawei Ascend G510	<p>Pasos para reproducir:</p> <ol style="list-style-type: none"> 1. Correr el app 2. Iniciar sesión 3. Abrir el menú principal <p>Resultado actual: La animación se ve lenta.</p> <p>Resultado esperado: La aplicación debe verse fluida. Si el dispositivo es de baja gama debe desactivarse esta animación.</p>
2	Cuando se entra a la herramienta de ubicaciones sin conexión a internet no se muestra un mensaje error.	Mayor	Todos	<p>Pasos para reproducir:</p> <ol style="list-style-type: none"> 1. Comprobar que no se cuenta con conexión a internet. 2. Correr el app 3. Iniciar sesión 4. Abrir el menú principal

				4. Seleccionar herramientas 6. Seleccionar herramienta de ubicaciones Resultado actual: La vista no carga nada (se queda en blanco). Resultado esperado: Se debe mostrar un mensaje para comprobar la conexión a internet.
--	--	--	--	---

Tabla 26 Lista de bugs

Despliegue

Servidor

La aplicación en el servidor esta desarrollado bajo Python con el Micro-Framework Flask, en Ubuntu 16.04. Se usara Gunicorn Application Server para desplegar la aplicación y nginx para actuar como proxy inverso.

Correr el servidor

1. Instalando las dependencias:

```
sudo apt-get install python-pip python-dev nginx
sudo pip install virtualenv
```

2. Instalando los requerimientos del proyecto.

Ubicarse en el directorio del proyecto.

```
cd ~/prototype-ayre-api
virtualenv env
source env/bin/activate
pip install -r requirements.txt
pip install gunicorn flask
```

3. configurando wsgi

```
nano ~/prototype-ayre-api/ayre.py
```

Contenido del wsgi.py para iniciar la aplicación.

```
from prototype-ayre-api import application
```

```
if __name__ == "__main__":  
    application.run()
```

4. Creando un script de arranque automático

```
sudo vim /etc/init/ayre.conf
```

```
description "Gunicorn application server running ayre"
```

```
start on runlevel [2345]
```

```
stop on runlevel [!2345]
```

```
respawn
```

```
setuid user
```

```
setgid www-data
```

```
env PATH=/home/user/ayre/ayre/env/bin
```

```
chdir /home/user/prototype-ayre-api
```

```
exec gunicorn --workers 3 --bind unix:ayre.sock -m 007 wsgi
```

5. Iniciando el servicio.

```
sudo start ayre
```

6. Configurando Nginc para que reciba las solicitudes.

```
sudo nano /etc/nginx/sites-available/ayre
```

```
server {
```

```
    listen 80;
```

```
    server_name server_domain_or_IP;
```

```
    location / {
```

```
        include proxy_params;
```

```
        proxy_pass http://unix:/home/user/prototype-ayre-api/ayre.sock;
```

```
    }
```

```
}
```

haciendo la configuración publica.

```
sudo ln -s /etc/nginx/sites-available/ prototype-ayre-api /etc/nginx/sites-enabled
```

Reiniciando el servidor nginx para hacer visibles los cambios.

```
sudo nginx -t && sudo service nginx restart
```


App

Optimización para despliegue

Para la optimización de la aplicación cliente se dispone de Gulp.js la cual es un paquete de Node.js que junto a algunos plugins actúa como herramienta para automatización de tareas en JavaScript, contamos con las siguientes tareas:

- **Sass:** Para esta tarea se usó el plugin **gulp-sass** que compila el código Sass a CSS y **gulp-minify-css**, que comprime y minifica el archivo CSS para aligerar su peso y carga.
- **JS:** La función de esta tarea es juntar los archivos JavaScript de la aplicación específicamente los de Angular por tanto se usaron los siguientes plugins **gulp-angular-filesort**, el cual toma los archivos y los ordena según la prioridad de inyección para la correcta ejecución de módulos y servicios, **gulp-concat** para juntar todos los archivos a uno solo resultante, **gulp-ng-annotate** el cual prepara el código de Angular para la minificación y finalmente **gulp-uglify** que minifica y comprime todo el código disminuyendo su peso y tiempo de carga.
- **Vendor:** En esta función se juntan y minifican en un solo archivo resultante las librerías externas usadas en la aplicación tomadas de **vendor.json** usando **gulp-concat** y **gulp-uglify**.
- **Templates:** Los archivos HTML tienen un tratamiento diferente ya que son cargados como módulos de AngularJS precargándolos como **\$templateCache** con lo que no se necesita llamarlos cada vez que son requeridos. Primero los archivos HTML de la aplicación son minificados con la librería **gulp-minify-html**, luego de esto son pasados a Angular con la librería **gulp-ng-html2js**, quedando como resultado archivos JS que son concatenados y minificados para así generar un solo archivo comprimido.
- **Fonts:** Para esta tarea las fuentes son simplemente copiadas de su origen como librerías externas y puesta en la carpeta donde la aplicación carga las fuentes.
- **Images:** Las imágenes son simplemente copiadas de su origen como librerías externas y puesta en la carpeta donde la aplicación carga las imágenes.

Según la estructura usada la carpeta de trabajo y llamada **src/** dentro del proyecto contiene todos los archivos crudos los cuales serán procesados con Gulp.js y puestos en el directorio original que usará Ionic Framework para construir la aplicación: **www/**.

Compilación de las Aplicaciones.

Las aplicaciones son compiladas con ayuda de Apache Cordova y el SDK de cada plataforma de destino. Para el caso de Android, usamos el Android SDK y algunas herramientas incluidas en este para la firma de la aplicación y posterior publicación en la tienda. Y para IOS, se requiere un equipo MacBook o iMac con las herramientas necesarias para el despliegue IOS simulator y xcode 7 para firmar y subir la aplicación a la tienda.

La configuración general de la app se edita en el archivo **config.xml**, como versión de la aplicación, nombre del paquete, permisos, y configuraciones de los diferentes plugins usados.

- **Cordova-plugin-device:** Obtiene información nativa del dispositivo.
- **Cordova-plugin-whitelist:** Permite la consulta a servicios externos a la aplicación.
- **Cordova-plugin-statusbar:** Cambia el color de la barra de estado en android y IOS.
- **Ionic-plugin-keyboard:** Permite el uso del teclado nativo en las aplicaciones ionic.
- **Cordova-plugin-geolocation:** Incluye el uso del GPS nativo de la aplicación.
- **Cordova-plugin-splashscreen:** Permite el uso del splash screen nativo en la app.
- **Cordova-plugin-inappbrowser:** Hace uso un navegador integrado para ver recursos externos o enviarlos al navegador del dispositivo.
-

Las imagen para el logo y SplashScreen son guardadas en la carpeta **resources/** y generadas para las diferentes resoluciones y versiones de las plataformas IOS y Android con una herramienta de ionic a través del comando: **ionic resources**.

La construcción final de los ejecutables para las dos versiones se hace con el siguiente comando: **ionic build ios** y **ionic build android**, respectivamente para cada plataforma las cuales generan un archivo **IPA** y un archivo **APK**.