

**MANUAL TÉCNICO**

**PROTOTIPO APLICACIÓN MÓVIL DE SERVICIOS DE CONSULTA DEL MÓDULO ESTUDIANTIL 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 7

Requerimientos técnicos 7

Herramientas y tecnologías utilizadas para los servicios web 7

Herramientas y tecnologías utilizadas para la aplicación móvil 8

Requerimientos funcionales 8

Requerimientos no funcionales 8

Alcance del sistema 8

Diseño del sistema 9

Descripción general 9

Arquitectura del sistema 9

Arquitectura física 9

Arquitectura lógica 10

Esquema de Seguridad 10

Modelo de casos de uso 10

Descripción general de los actores 10

Diagramas del modelo de casos de uso 11

Modelo de datos 17

Diseño de interfaz 22

Implementación 27

Entorno utilizado 27

Parte servidor 28

Parte cliente 28

Pruebas 28

Prueba de servicios web 28

Pruebas en dispositivos móviles 29

Despliegue 29

Servidor 29

Correr el servidor 29

App 29

Optimización para despliegue 29

Publicación en tiendas de aplicaciones 29

**Lista de figuras**

**Lista de tablas**

# 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

Aca se habla de los datos mínimos que comparten todas las universidades..porque se escogieron….cuales son sus posibles varientes, bla bla.

## 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

**Python**: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Es un lenguaje de programación multiparadigma y multiplataforma.

**Flask**: Es un framework web minimalista escrito en Python.

**MySQL**: es un sistema de gestión de base de datos relacional considerada la base de datos open source mas popular del mundo.

## Herramientas y tecnologías utilizadas para la aplicación móvil

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

Ionic framework

Gulp

Cordova

Genymotion

iOS Simulator

## 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 sobe 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

Cras dapibus. Pellentesque dapibus hendrerit tortor. Duis arcu tortor, suscipit eget, imperdiet nec, imperdiet iaculis, ipsum. Praesent porttitor, nulla vitae posuere iaculis, arcu nisl dignissim dolor, a pretium mi sem ut ipsum. Nunc nulla.

# Diseño del sistema

## Descripción general

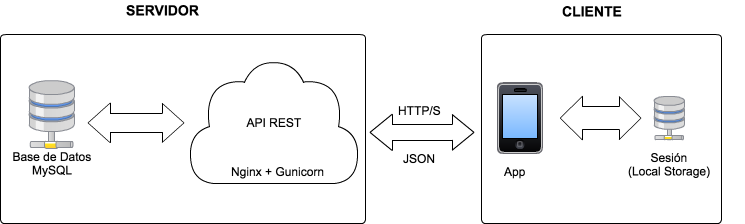
Nullam dictum felis eu pede mollis pretium. Phasellus a est. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Fusce id purus. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Fusce id purus.

Quisque id mi. Donec interdum, metus et hendrerit aliquet, dolor diam sagittis ligula, eget egestas libero turpis vel mi. Vestibulum purus quam, scelerisque ut, mollis sed, nonummy id, metus. Sed mollis, eros et ultrices tempus, mauris ipsum aliquam libero, non adipiscing dolor urna a orci. Nam ipsum risus, rutrum vitae, vestibulum eu, molestie vel, lacus.

## Arquitectura del sistema

### Arquitectura física

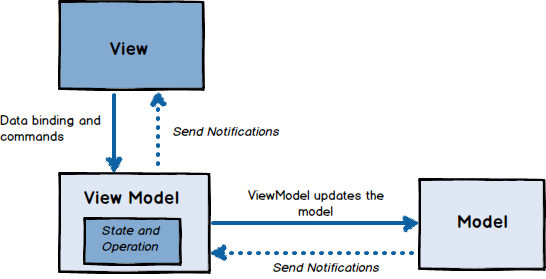
La arquitectura predominante Cliente-Servidor ha dado paso a un nuevo modelo en el que hay un servidor web de API’s que es consumido por varios servicios o aplicaciones. Este modelo es el propuesto para el prototipo de AyRE móvil, por lo tanto obtendrá la información académica codificada en JSON a través 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.



Fuente: los autores

### Arquitectura lógica

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



Fuente: los autores

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.

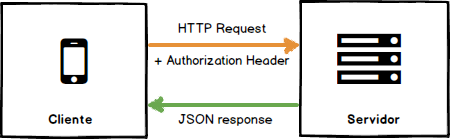
## Esquema de Seguridad

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 el cuenta los siguientes factores:

* El API acepta peticiones de 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**:



Fuente: los autores

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



Fuente: los autores.

|  |  |
| --- | --- |
| **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** | 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** | 1. El código y contraseña proporcionados por el usuario no corresponden a los registrados en la base de datos. 2. Se muestra un mensaje de error. |
| **Poscondiciones** | El usuario se ha validado correctamente, permitiendo el acceso a las demás opciones del app. |

|  |  |
| --- | --- |
| **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** | 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. |

|  |  |
| --- | --- |
| **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** | 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. |

|  |  |
| --- | --- |
| **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** | 1. App retorna la vista de perfil después de que el usuario inicia sesión. |
| **Flujo Alternativo** | 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 |

|  |  |
| --- | --- |
| **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** | 1. El usuario abre el menú principal. 2. El usuario ejecuta la opción “Noticias” 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 |

|  |  |
| --- | --- |
| **Nombre** | Consultar horario |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Consultar notas |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Consultar historial |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **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** | 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. 4. Se actualiza la fecha de la ultima actualización. |
| **Flujo Alternativo** | 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. |

|  |  |
| --- | --- |
| **Nombre** | Filtrar por asignatura |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Búsqueda rápida |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** |  |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Listar herramientas |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Calcular notas |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Consultar ubicaciones |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Configurar alarma |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Consultar contactos |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

|  |  |
| --- | --- |
| **Nombre** | Ver ayuda |
| **Descripción** |  |
| **Actores** | Usuario |
| **Precondiciones** | El usuario debe estar previamente validado en el sistema (Iniciar sesión). |
| **Flujo Normal** |  |
| **Flujo Alternativo** |  |
| **Poscondiciones** |  |

## 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 | - username: charfield  - password: charfield |
| Descripción | Obtiene el token de usuario a través del codigo y contraseña. |
| Respuesta | **200 OK:**  { "token": "\*\*\*" }  **404 NOT FOUND:**  {“message”: “User doesn't exist” }  **401 UNAUTHORIZED:**  {"message": "Invalid login info"} |

|  |  |
| --- | --- |
| URL | /personal/ |
| Método | **GET** |
| Campos | *Authorization: Token \*\*\** |
| 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:**  {  "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": "-",  "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"} |

|  |  |
| --- | --- |
| URL | /schedule/ |
| Método | **GET** |
| Campos | *Authorization: Token \*\*\** |
| 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:**  [  {  "ASIGNATURA": "-",  "TIPO\_ASIGNATURA": "-",  "TIPO\_REGISTRO": "-",  "CODMAT": "-",  "DOCENTE": "-"  "GRUPO": "6"  "DOMINGO": "11-12<br>Lugar 1 ; 15-18<br>lugar 2",  "JUEVES": "-",,  "MARTES": "-",  "VIERNES": "-",  "LUNES": "-",  "SABADO": "-",  "MIERCOLES": "-",  },  ...  ]  **401 UNAUTHORIZED:**  {"message": "Invalid Token - Authorization Required"} |

|  |  |
| --- | --- |
| URL | /scores/ |
| Método | **GET** |
| Campos | *Authorization: Token \*\*\** |
| 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 | **200 OK:**  [  {  "ASIGNATURA": "-",  "NOTA1": "-",  "NOTA3": "-",  "NOTA2": "-",  "FINAL": "-",  "HAB": "-",  "DEF": "-",  "CODIGO": "-",  },  …  ]  **401 UNAUTHORIZED:**  {"message": "Invalid Token - Authorization Required"} |

|  |  |
| --- | --- |
| URL | /history/ |
| Método | **GET** |
| Campos | *Authorization: Token \*\*\** |
| 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:**  [  {  "periodo": "-",  "prom\_acumulado": "-",  "prom\_semestral": "-",  "asignaturas": [  {  "ASIGNATURA": "-",  "NOTA1": "-",  "NOTA2": "-",  "NOTA3": "-",  "NOTA\_FIN": "-",  "NOTA\_HAB": "-",  "NOTA\_DEF": "-",  "TREG": "NOR",  "COD\_MATERIA": "-",  "CRED": "-",  },  …  }  …  ]  **401 UNAUTHORIZED:**  {"message": "Invalid Token - Authorization Required"} |

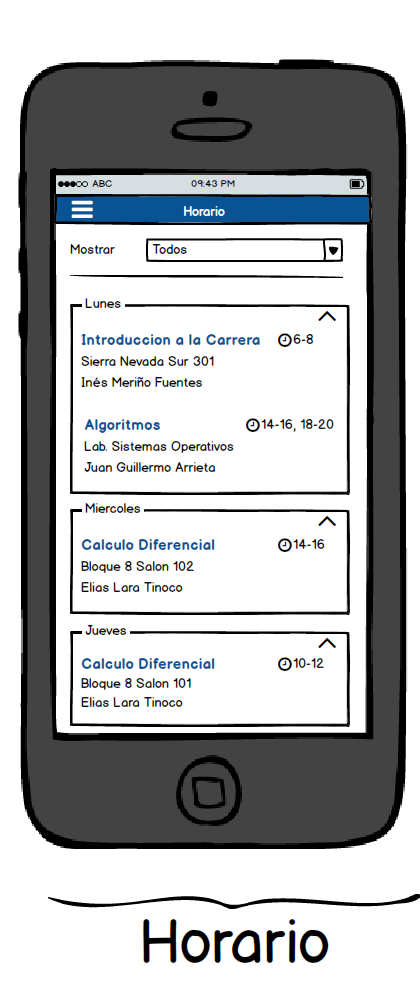
|  |  |
| --- | --- |
| URL | /photo/ |
| Método | **GET** |
| Campos | *Authorization: Token \*\*\** |
| Descripción | Obtiene el avatar del usuario validado en el sistema. Se debe enviar el token en los headers de la petición o como un parámetro en la url: ?token=\*\*\* |
| Respuesta | **200 OK:**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

## 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.



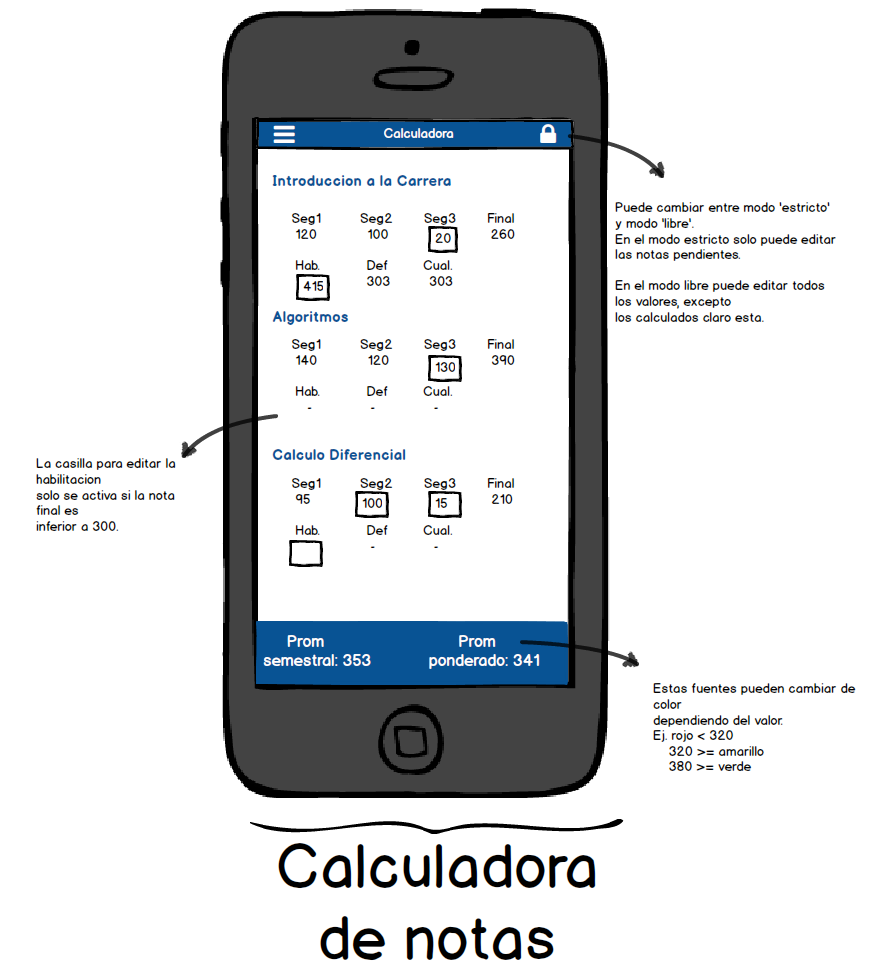
Fuente: los autores Fuente: los autores



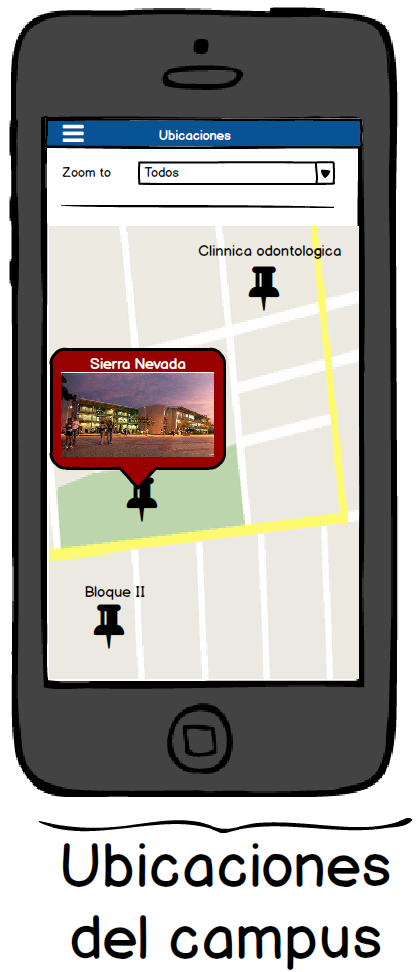
Fuente: los autores



Fuente: los autores Fuente: los autores



Fuente: los autores Fuente: los autores.

 Fuente: Los autores



Fuente: los autores



Fuente: los autores

# Implementación

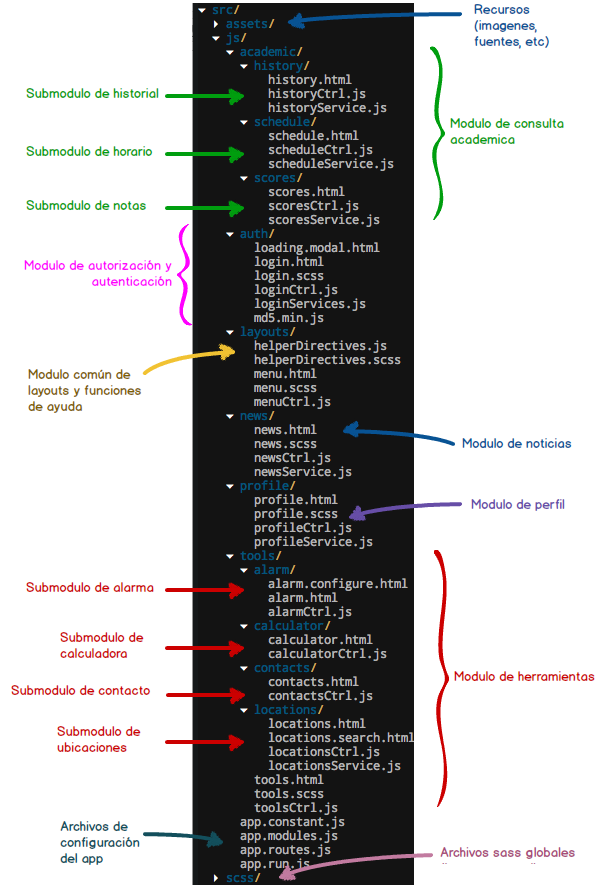
## Parte servidor

Se describe brevemente como se hicieron los servicios, como resuelve las peticiones, si se uso un ORM , etc

## 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 especificas haciéndola mas sencilla para escalar y mantener.



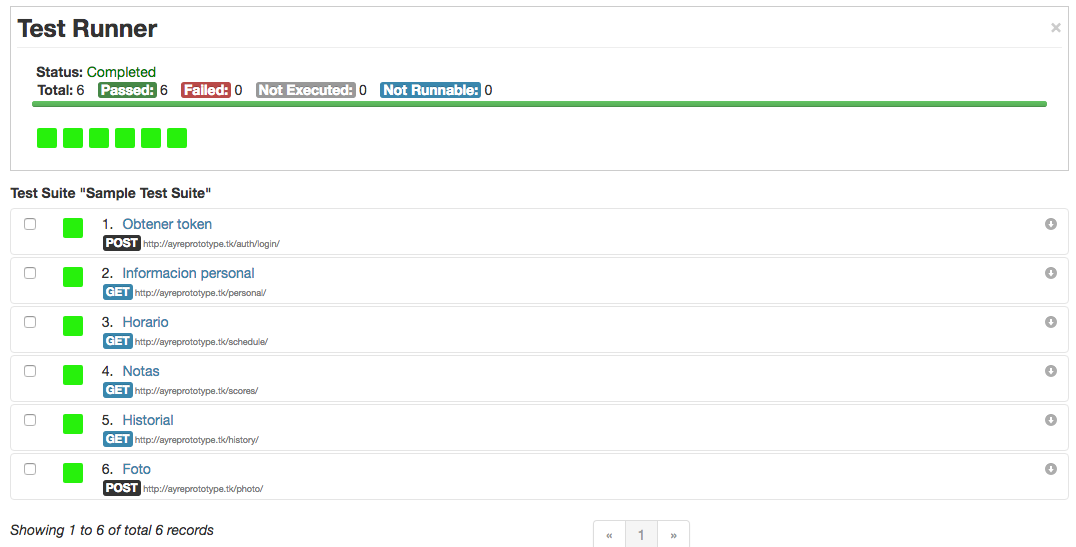
**Vistas finales**

asdasdasdasd

# 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.



Como se observa en la figura X, 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

Aca se pone una tabla de los dispositivos probados en cuales funciona bien, en cuales mas o menos y en cuales no corre.

# Despliegue

## Servidor

Que software utiliza el hosting al momento del despliegue. Si es nginx con wsgi, etc..

### Correr el servidor

Como correr el servidor localmente…

## App

### Optimización para despliegue

Que optimizaciones se hacen con gulp para mejorar el resultado final. Agregar compatibilidad de css, minificacion, optimización de imagenes

### Publicación en tiendas de aplicaciones

#### Play Store

#### App Store