



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 4**

**Aplicación para facilitar el acceso a la información no confidencial generada por el Sistema de Gestión para el Ingreso a la Educación Superior**

**Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas**

**Autor: Leandro Alejo Aguila**

**Tutores: Ing. Neysa Baldoquín Alonso**

**Ing. Ernesto Gutierrez Ramos**

**Cotutores: Ing. Susana Becerra Rodriguez**

**Ing. Adrián Arencibia Herrera**

**La Habana, 28 de junio de 2018**

**Curso: 2017-2018**

## *Declaración de Autoría*

---

Declaro que soy el único autor del trabajo de diploma titulado Aplicación para facilitar el acceso a la información no confidencial generada por el Sistema de Gestión para el Ingreso a la Educación Superior y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmo la presente declaración jurada de autoría a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2018.

\_\_\_\_\_  
Firma del autor

Leandro Alejo Aguila

\_\_\_\_\_  
Firma de la tutora

Neysa Baldoquín Alonso

\_\_\_\_\_  
Firma del tutor

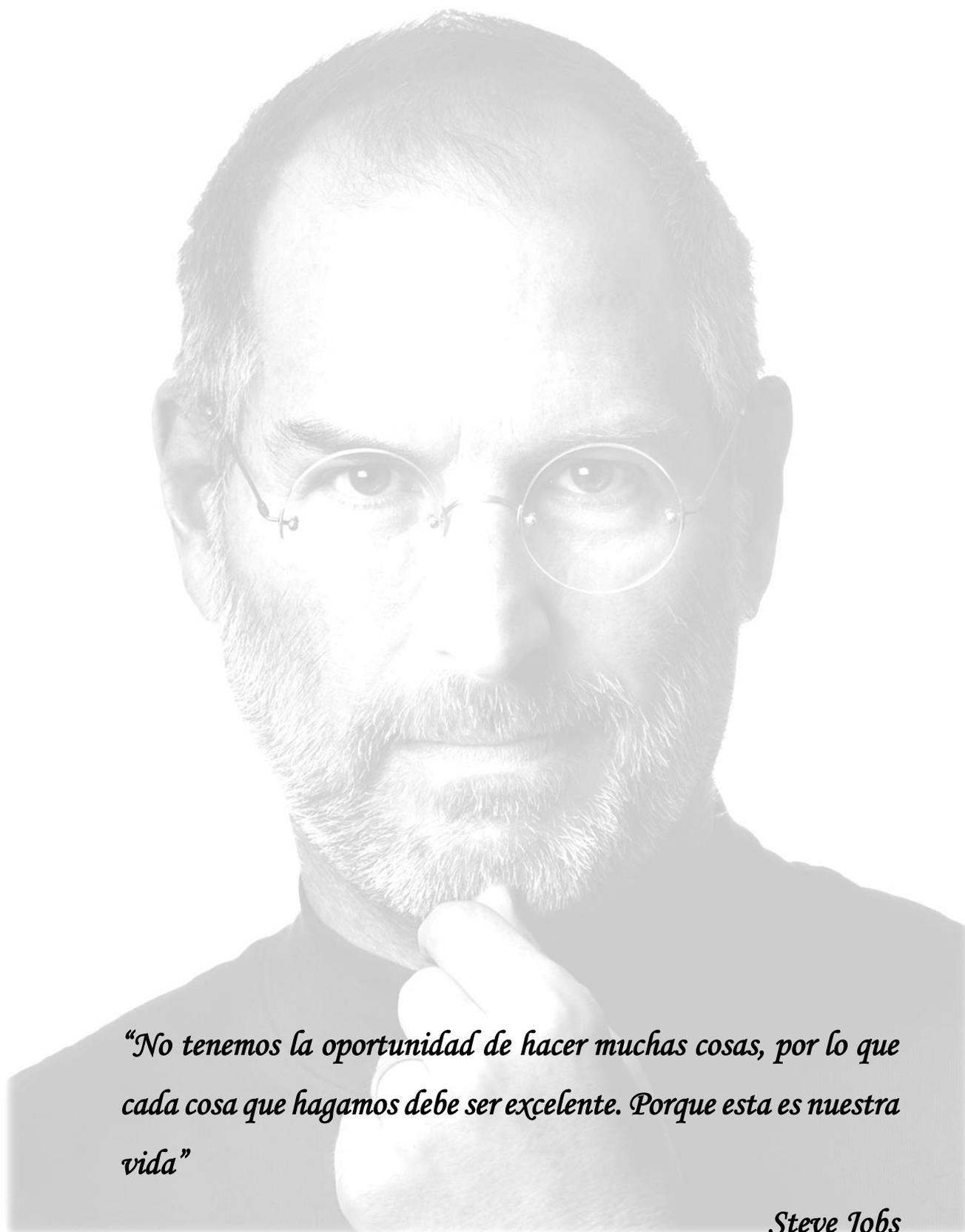
Ernesto Gutierrez Ramos

\_\_\_\_\_  
Firma de la co-tutora

Susana Becerra Rodriguez

\_\_\_\_\_  
Firma del co-tutor

Adrián Arencibia Herrera



*“No tenemos la oportunidad de hacer muchas cosas, por lo que cada cosa que hacemos debe ser excelente. Porque esta es nuestra vida”*

*Steve Jobs*

*A mis padres, porque a ustedes les debo mi vida y son el motor impulsor de todo lo que hago.*

*A mis hermanos, porque espero que en un futuro no muy lejano pueda estar presente en sus logros.*

*A mi novia, porque me amaste en todo el camino y deseo que me ames el resto del mismo.*

*A mi familia y la de mi novia porque siempre me ayudaron y desearon lo mejor.*

## *Agradecimientos*

---

*A mis padres Yusimi y Leonel, por su dedicación y sacrificio constante, sin ustedes esto no hubiera sido posible.*

*A mi novia Diana, por su amor y apoyo incondicional durante todo el camino recorrido a pesar de la distancia.*

*A mi familia, por su apoyo en el transcurso de mi vida, en especial a mis tías Marlen y Mildrey quienes siempre estuvieron pendiente de mí y de mi carrera.*

*A mi otra familia (la de Diana), por permitirme formar parte de ella, en especial a mis suegros Margarita y Manuel por acogerme como un verdadero hijo.*

*A mis compañeros de la universidad por los buenos ratos, las enseñanzas y esos recuerdos que solo se quedan en la memoria, en especial a Sergio con quien compartí y aprendí mucho.*

*A mis tutores por su guía y ayuda, por el tiempo dedicado y sus consejos.*

*A los profesores que de una forma u otra han intervenido en mi formación profesional y personal.*

*A todas las personas que de una forma u otra contribuyeron en la realización de esta tesis y en mi formación en general.*

**Resumen**

En el Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES) se trabaja en la incorporación de una capa de servicios web que permita el acceso a la información no confidencial asociada al proceso de ingreso. Esta información se puede consultar por otras vías menos asequibles como ir personalmente a los lugares donde se realiza el proceso, en ocasiones distantes geográficamente, o por vía telefónica. La manera actual de acceder a la información en ocasiones no garantiza simultaneidad y claridad con el proceso de ingreso. Por esta razón la presente investigación tiene como objetivo desarrollar una aplicación móvil que permita mejorar la accesibilidad y portabilidad de la información no confidencial generada por el SIGIES. Para guiar el desarrollo de la aplicación se utilizó la metodología AUP variación UCI enmarcada en el escenario 4 de Historias de Usuario, generando los artefactos que define cada una de sus fases. Se empleó Java v7.0 como lenguaje de programación; Visual Paradigm v8.0 y Android Studio v3.0 como herramienta para modelar diagramas y entorno de desarrollo respectivamente. Como resultado, los usuarios podrán contar con un APK que mediante el consumo de servicios web sea capaz de proveer información no confidencial generada por el SIGIES, además, podrán sincronizar los últimos cambios realizados permitiendo mejorar la accesibilidad y portabilidad de la información. Se validó el funcionamiento de la aplicación a través de pruebas de software detectando y corrigiendo las no conformidades existentes.

**Palabras claves:** APK, servicios web, sincronización de datos

## Índice

|  |    |
|--|----|
| Índice de figuras.....                                       | 3  |
| Índice de tablas .....                                       | 4  |
| Introducción .....   | 6  |
| Capítulo 1: Fundamentación teórica de la investigación ..... | 10 |
| 1.1 Introducción.....  | 10 |
| 1.2 Estudio de soluciones similares.....                     | 10 |
| 1.2.1 Stack Overflow .....                                   | 10 |
| 1.2.2 LabCoat .....  | 10 |
| 1.2.3 Moodle Mobile (MM).....                                | 11 |
| 1.2.4 RepoDroid.....   | 11 |
| 1.3.1 Dispositivos móviles .....                             | 12 |
| 1.3.2 Android .....  | 13 |
| 1.3.3 APK.....   | 14 |
| 1.3.4 SIGIES.....  | 15 |
| 1.3.5 Servicio web .....                                     | 15 |
| 1.3.6 Sincronización de datos.....                           | 16 |
| 1.4 Tecnologías.....   | 17 |
| 1.4.1 Kit de desarrollo de <i>software</i> .....             | 17 |
| 1.4.2 Entorno de desarrollo integrado .....                  | 17 |
| 1.4.3 Lenguaje de desarrollo Java .....                      | 18 |
| 1.4.4 Herramienta CASE .....                                 | 19 |
| 1.4.5 Balsamiq.....  | 20 |
| 1.5 Metodología de desarrollo .....                          | 20 |
| Conclusiones parciales .....                                 | 24 |
| Capítulo 2: Propuesta de solución .....                      | 25 |
| 2.1 Introducción.....  | 25 |
| 2.2 Descripción de la propuesta de solución.....             | 25 |
| 2.3 Modelo de dominio.....                                   | 26 |
| 2.4 Requisitos de <i>software</i> .....                      | 27 |

|  |    |
|--|----|
| 2.4.1 Requisitos funcionales .....             | 27 |
| 2.4.2 Requisitos no funcionales.....           | 28 |
| 2.5 Historias de Usuario .....                 | 29 |
| 2.6 Descripción del patrón arquitectónico..... | 31 |
| 2.7 Patrones de diseño .....                   | 32 |
| 2.7.1 Patrones GRASP .....                     | 33 |
| 2.7.2 Patrones GOF .....                       | 33 |
| 2.8 Diagrama de clases del diseño.....         | 34 |
| 2.9 Diagrama de despliegue .....               | 34 |
| 2.9 Conclusiones del capítulo .....            | 35 |
| Capítulo 3: Implementación y pruebas.....      | 36 |
| 3.1 Introducción.....                          | 36 |
| 3.2 Diagrama de componente.....                | 36 |
| 3.3 Estándares de codificación.....            | 37 |
| 3.4. Casos de Prueba .....                     | 38 |
| 3.5 Pruebas de software.....                   | 39 |
| 3.5.2 Pruebas de aceptación .....              | 40 |
| 3.5 Conclusiones parciales .....               | 41 |
| Conclusiones generales .....                   | 42 |
| Recomendaciones .....                          | 43 |
| Bibliografía .....                             | 44 |
| Anexos.....                                    | 47 |
| Anexo 1: Historias de usuario.....             | 47 |
| Anexo 2: Casos de prueba.....                  | 65 |

**Índice de figuras**

Fig. 1 Modelo de dominio .....27

Fig. 2 Patrón arquitectónico MVC para Android .....32

Fig. 3 Diagrama de clases.....34

Fig. 4 Diagrama de despliegue.....35

Fig. 5 Diagrama de componente .....37

Fig. 6 No conformidades.....41

## Índice de tablas

|  |    |
|--|----|
| Tabla 1 Principales características de las aplicaciones analizadas .....                   | 12 |
| Tabla 2 Fases de AUP y AUP-UCI .....   | 22 |
| Tabla 3 HU Mostrar datos del estudiante .....  | 30 |
| Tabla 4 Modelo de caso de prueba correspondiente al RF11 .....                             | 38 |
| Tabla 5 HU Autenticar usuario.....   | 47 |
| Tabla 6 HU Sincronizar datos con la capa de servicios web .....                            | 48 |
| Tabla 7 HU Listar provincias .....   | 49 |
| Tabla 8 HU Listar plan de plazas provincial .....  | 50 |
| Tabla 9 HU Filtrar plan de plazas provincial.....  | 51 |
| Tabla 10 HU Mostrar datos de una carrera del plan de plazas provincial.....                | 52 |
| Tabla 11 HU Mostrar solicitud del estudiante .....   | 54 |
| Tabla 12 HU Mostrar notas por cada asignatura examinada .....                              | 55 |
| Tabla 13 HU Mostrar exámenes exonerados.....   | 56 |
| Tabla 14 HU Mostrar datos de la vía de ingreso del estudiante .....                        | 57 |
| Tabla 15 HU Mostrar asistencia .....   | 59 |
| Tabla 16 HU Mostrar resultado de la asignación.....  | 60 |
| Tabla 17 HU Mostrar estado del servicio militar .....                                      | 61 |
| Tabla 18 Mostrar etapa actual del proceso de ingreso.....                                  | 63 |
| Tabla 19 HU Mostrar actas de exámenes .....  | 64 |
| Tabla 20 Caso de prueba 1: Autenticar usuario .....  | 65 |
| Tabla 21 Caso de prueba 2: Sincronizar datos con la capa de servicios web.....             | 66 |
| Tabla 22 Caso de prueba 3: Listar provincias.....  | 67 |
| Tabla 23 Caso de prueba 4: Listar plan de plazas provincial .....                          | 68 |
| Tabla 24 Caso de prueba 5: Filtrar plan de plazas provincial.....                          | 68 |
| Tabla 25 Caso de prueba 6: Mostrar datos de una carrera del plan de plazas provincial..... | 69 |
| Tabla 26 Caso de prueba 7: Mostrar solicitud del estudiante.....                           | 69 |
| Tabla 27 Caso de prueba 8: Mostrar notas por cada asignatura examinada .....               | 70 |
| Tabla 28 Caso de prueba 9: Mostrar si está o no exonerado de algún examen.....             | 70 |
| Tabla 29 Caso de prueba 10: Mostrar datos de la vía de ingreso del estudiante .....        | 71 |
| Tabla 30 Caso de prueba 11: Mostrar datos del estudiante .....                             | 71 |

|   |    |
|---|----|
| Tabla 31 Caso de prueba 12: Mostrar asistencia.....                   | 72 |
| Tabla 32 Caso de prueba 13: Mostrar resultado de la asignación .....  | 72 |
| Tabla 33 Caso de prueba 14: Mostrar estado del servicio militar ..... | 73 |
| Tabla 34 Caso de prueba 15: Mostrar etapa actual del proceso .....    | 73 |
| Tabla 35 Caso de prueba 16: Mostrar actas de examen .....             | 74 |

### Introducción

Con el surgimiento de Internet y el desarrollo vertiginoso de las Tecnologías de la Información y las Comunicaciones (TIC) se ha abierto una gama de posibilidades para la sociedad. El acceso a la información no se ve limitado a barreras geográficas y es posible encontrar en Internet una gran diversidad de información. Según Francis Pisani *"Ninguna tecnología ha adentrado tan rápidamente en nuestra sociedad como la tecnología de la información y las comunicaciones"* (1).

En los últimos años, el uso de dispositivos móviles ha influido decisivamente sobre la manera en que se gestiona la información. El impacto de esta revolución en las principales formas de comunicación, el compartir información y cómo se resuelven las diversas problemáticas, hace más contundente el uso de estos ya que el usuario está conectado con servicios digitales y esta tendencia va en aumento. Estos dispositivos tienen acceso a la información de una forma más rápida y eficiente, además de tener la disponibilidad de llevarlos a cualquier lugar y conectarlos en cualquier red (2).

Los dispositivos móviles pueden ser de pequeño tamaño y jugar un papel primordial atendiendo a sus características distintivas. Entre las principales características están (3):

- Capacidades especiales de procesamiento
- Conexión permanente o intermitente a una red
- Memoria limitada
- Diseños específicos para una función principal y versatilidad para el desarrollo de otras funciones
- Tanto su posesión como su operación se asocia al uso individual de una persona, la cual puede configurarlos a su gusto

La movilidad que ofrecen estos dispositivos y su diseño ajustado a facilidades para transportarlos, permiten que puedan ser sincronizados con los servicios que proveen sistemas externos como actualización de aplicaciones y de datos. Esta modalidad de consumir servicios, se ha expandido y popularizado en gran medida gracias a los dispositivos inteligentes o *smartphones* (3).

Los servicios web en Internet son comprensibles y de fácil acceso; permiten a los desarrolladores de aplicaciones acceder a una gran cantidad de servicios de terceros<sup>1</sup> e integrarlos de forma relativamente sencilla, incrementando significativamente la utilidad y capacidad de sus aplicaciones. Aumentan la flexibilidad de procesos de diversa índole al integrarse con otras aplicaciones además de, aportar independencia entre la aplicación que usa el servicio web y el propio servicio ya que pueden ser llamados

---

<sup>1</sup> Servicios suministrados y operados por proveedores ajenos al negocio, tales como Twitter y YouTube.

por cualquier aplicación, usando cualquier lenguaje de programación y bajo cualquier Sistema Operativo (en lo adelante SO). Independientemente de sus propiedades o de las plataformas sobre las que se instalen, los cambios a lo largo del tiempo en uno de los *endpoints*<sup>2</sup> no deben afectar al otro; todo esto es gracias a su característica principal: interoperabilidad (4).

En el contexto mundial existen entidades, compañías e instituciones que brindan servicios web a los usuarios con el objetivo de facilitarles el trabajo en la red a través de los dispositivos móviles. Se pueden mencionar los servicios brindados por Facebook, Google, los sitios web para servicios de compra y venta a nivel internacional tales como Amazon y EBay. Todas estas compañías tienen sus aplicaciones nativas para dispositivos móviles, facilitando el acceso a la información.

En este contexto, Cuba desarrolla el programa de informatización de la sociedad, sustentado en la creación de una infraestructura tecnológica y la generación de servicios y contenidos digitales. Dicho programa constituye la principal forma de incentivar el acceso de los ciudadanos a estos servicios y contenidos, así como al empleo de las Tecnologías de la Información y la Comunicaciones (5).

Teniendo en cuenta lo anterior, el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas (UCI) desarrolla el Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES). En dicho sistema se trabaja en la incorporación de una capa de servicios web para permitir el acceso a la información no confidencial asociada al proceso de ingreso. Esta información actualmente se puede consultar por otras vías menos asequibles como: ir personalmente a los lugares donde se hace el proceso, en ocasiones distantes geográficamente, o por vía telefónica. La manera actual de acceder a la información en ocasiones no garantiza simultaneidad y claridad. Además, a pesar del auge de las TIC en Cuba, los interesados en el proceso no cuentan con facilidades que, a través del empleo de estas tecnologías, les permitan consultar y analizar informaciones asociadas al proceso de ingreso.

Teniendo en cuenta dicha situación, se presenta el siguiente **problema de la investigación**: ¿Cómo mejorar la accesibilidad y portabilidad de la información no confidencial generada por el SIGIES?

Se define como **objeto de estudio**: aplicaciones que permiten la sincronización y consumo de datos a través de servicios web; enmarcado en el **campo de acción**: aplicaciones móviles que permiten la sincronización y el consumo de datos a través de servicios web.

---

<sup>2</sup> Cada uno de los elementos que se sitúan en ambos extremos de la red que sirve como canal de comunicación.

Para dar solución al problema científico de la presente investigación se plantea el siguiente **objetivo general**: desarrollar una aplicación móvil que permita mejorar la accesibilidad y portabilidad de la información no confidencial generada por el SIGIES.

Los **objetivos específicos** para asegurar el cumplimiento del objetivo trazado son:

- Elaborar un marco teórico de la investigación a través de un estudio del estado del arte de aplicaciones para dispositivos móviles que acceden a servicios web, para lograr una mejor comprensión del tema.
- Realizar del levantamiento de requisitos funcionales y no funcionales.
- Analizar e implementar una aplicación, que brinde solución al problema planteado.
- Realizar pruebas para validar el cumplimiento de los requerimientos de la solución desarrollada.

Para el correcto desarrollo de la investigación, se emplearon los siguientes **métodos científicos**:

### **Métodos teóricos:**

- **Análisis histórico-lógico:** Consiste en investigar y estudiar los aspectos generales del objeto de estudio, caracterizando su funcionamiento y desarrollo. Se utilizó para el estudio de soluciones similares e investigaciones que abordan el tema de aplicaciones para dispositivos móviles a nivel nacional e internacional. Las cuales servirán como punto de partida para el desarrollo de la solución en su conjunto.
- **Método analítico-sintético:** Está basado en separar los elementos que intervienen en la realización de un fenómeno determinado para estudiarlos de forma individual y luego de manera integral. Se empleó para analizar desde diferentes aristas los conceptos asociados a las aplicaciones para dispositivos móviles y sintetizar la información recopilada, permitiendo describir las características generales y las relaciones esenciales entre estas.
- **Método de la modelación:** Consiste en crear una representación o un modelo con el objetivo de investigar la realidad y descubrir nuevas relaciones y cualidades del fenómeno que se estudia. Se utilizó para crear abstracciones y modelar diversos diagramas en cada uno de los flujos de trabajo según la metodología seleccionada.

### **Métodos empíricos:**

- **Observación científica:** Consiste en la percepción directa del objeto de investigación permitiendo conocer la realidad a través de datos o características específicas. Se empleó para observar el

funcionamiento de algunas aplicaciones para dispositivos móviles existentes, y obtener un registro visual de las características comunes en estas y que pueden formar parte de la solución.

### **Estructura capitular**

Capítulo I: Fundamentación teórica de la investigación.

En este capítulo se realiza un estudio del estado del arte de la presente investigación. Se exponen los principales conceptos relacionados con el objeto de estudio y el campo de acción. Se abordan aplicaciones similares, metodología de desarrollo de *software* utilizada, herramientas, lenguajes y tecnologías a tener en cuenta para el proceso de desarrollo de aplicaciones Android.

Capítulo II: Propuesta de solución.

Este capítulo contiene una descripción de la propuesta de solución, donde se abordan características particulares de la misma, incluye el diagrama de modelo del dominio y los de clases del diseño generados a partir del problema planteado. Comprende también, los requisitos funcionales y no funcionales que posteriormente deberá cumplir aplicación propuesta.

Capítulo III: Implementación y pruebas

En el capítulo tres se procede a la implementación de la aplicación, garantizando la solución al problema de la presente investigación. Se realiza el diagrama de componentes, se define el estándar de codificación a emplear y se ejecutan pruebas de software con el fin de garantizar una mayor calidad del producto.

## Capítulo 1: Fundamentación teórica de la investigación

### 1.1 Introducción

En el presente capítulo se realiza un análisis de aplicaciones similares y se destacan las principales características de los elementos asociados a la temática a investigar. Se precisan un conjunto de fundamentos y conceptos que constituyen el marco teórico relacionado con el objeto de estudio definido en la investigación. Además, se exponen la metodología de desarrollo de *software*, las tecnologías y herramientas que sirven de apoyo en la implementación de la propuesta de solución.

### 1.2 Estudio de soluciones similares

Se realiza un estudio del estado del arte analizando aplicaciones que consumen servicios web existentes tanto a nivel internacional como nacional y se profundiza en las principales características que poseen y que pudieran utilizarse o no en el desarrollo de la solución.

#### 1.2.1 Stack Overflow

- Aplicación donde los programadores profesionales y entusiastas hacen sus preguntas más difíciles y obtienen respuestas de sus compañeros.
- Las preguntas se pueden leer, buscar, compartir, responder, comentar, marcar, votar y filtrar por más nuevas y votos.
- El usuario recibe una notificación automática cuando alguien responde o comenta su pregunta subida.
- Se pueden activar y desactivar notificaciones mediante la configuración.
- Se pueden elaborar preguntas cuando no se está conectado.
- Las herramientas de reducción facilitan la adición de imágenes y bloques de código.
- Se puede regresar a las preguntas con una vista del historial de navegación que enumera las preguntas que se han revisado recientemente (6).

#### 1.2.2 LabCoat

- Aplicación en la que se pueden administrar proyectos, crear problemas e incluso aceptar solicitudes de fusión, todo desde un dispositivo móvil.
- Es de código abierto permitiendo contribuir con la aplicación y así ayudar a la comunidad de LabCoat.
- Brinda algunas funcionalidades como (7):
  - Ver, editar, comentar y cerrar proyectos
  - Ver confirmaciones de fusión
  - Ver, editar y comentar fusiones

# Capítulo 1: Fundamentación Teórica de la Investigación

---

- Ver, editar y comentar solicitudes de fusiones
- Buscar y ver archivos
- Administrar grupos y miembros del proyecto

## 1.2.3 Moodle Mobile (MM)

- Aplicación oficial de Moodle que utiliza tecnologías web comunes, entre ellas REST (Transferencia de Estado Representacional) como protocolo para obtener y enviar información a la plataforma *Moodle*.
- El diseño de la aplicación se creó empleando HTML 5 y CSS 3, la interacción con el teléfono y el empaquetado se realizó con *PhoneGap*<sup>3</sup>.
- Se encuentra disponible para dispositivos con los sistemas operativos *Android* e *iOS*.
- Es un reemplazo de la anterior Moodle App.
- Entre sus funcionalidades se encuentran (8):
  - Subir fotos y videos a la *web* (seleccionados de la galería o tomados instantáneamente de la cámara del móvil)
  - Grabar y subir un archivo de audio
  - Ver la lista de participantes en los cursos donde el usuario está inscrito
  - Ver la información de estos participantes
  - Mandar un mensaje privado a algún participante
  - Agregar una nota privada acerca de algún participante
  - Permite el trabajo *offline* con sincronización automática
  - Agregar los participantes a la lista de contactos del teléfono
  - Acceso a la versión web de la plataforma
  - Acceso a los cursos de la plataforma

## 1.2.4 RepoDroid

- Aplicación que ayuda a descargar e instalar las apps publicadas en el sitio <http://androiduclv.edu.cu> de forma rápida y sencilla navegando por las diferentes categorías de *software* presentes en el mercado.
- Solo se necesita seleccionar del catálogo la app que se desea instalar en el dispositivo y Android guiará el proceso de instalación.
- Desarrollada para que pueda ser utilizada en dispositivos con versiones de Android 2.2 en adelante (9).

---

<sup>3</sup> *Framework* para el desarrollo de aplicaciones móviles

# Capítulo 1: Fundamentación Teórica de la Investigación

A continuación, se muestra una tabla donde se evidencian las principales características de las aplicaciones analizadas:

**Tabla 1 Principales características de las aplicaciones analizadas**

| Característica\APK   | Stack Overflow | LabCoat | ModleMobile | ReprDroid |
|--|----------------|---------|-------------|-----------|
| Notificación   | x              | x       | x           | -         |
| Filtro   | x              | x       | x           | -         |
| Protocolo para obtener información de la capa de servicios web | REST           | REST    | REST        | REST      |
| Librería utilizada para realizar las peticiones web            | OkHttp         | OkHttp  | OkHttp      | OkHttp    |
| Sistema gestor de BD   | SQLite         | SQLite  | SQLite      | SQLite    |

Después de haber realizado un análisis de cada una de las aplicaciones similares que consumen servicios web, se procede a seleccionar las características que formarán parte de la propuesta de solución. Para la selección se tuvo en cuenta que las funcionalidades deben cumplir con los requisitos propuestos por el cliente. Las características son:

- Posibilidad de buscar y filtrar la información.
- REST como protocolo para obtener información de la capa de servicios web ya que los servicios consumidos están implementados usando dicho protocolo.
- SQLite como sistema gestor de bases de datos ya que no es un proceso independiente con el que el software se comunica pues pasa a ser parte integral del mismo. El software utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en el *host*.

## 1.3 Conceptos asociados a la investigación

### 1.3.1 Dispositivos móviles

Un dispositivo móvil se puede definir como un dispositivo de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, que ha sido diseñado específicamente para una función, pero que puede llevar a cabo otras funciones más generales. Existe gran variedad de dispositivos móviles, desde los reproductores de audio portátiles hasta los navegadores GPS, pasando por los teléfonos móviles, los PDA (Asistente Digital Personal, del inglés

# Capítulo 1: Fundamentación Teórica de la Investigación

---

*Personal Digital Assistant*) o las tabletas; los dispositivos móviles más utilizados y conocidos en la actualidad son los teléfonos móviles (10).

## Tipos de dispositivos móviles

- Dispositivo Móvil de Datos Limitados (*Limited Data Mobile Device*): Son teléfonos móviles clásicos que se caracterizan por tener una pantalla pequeña de tipo texto. Ofrecen servicios de datos generalmente limitados a SMS (*Short Message Service*) y acceso WAP (protocolo de aplicaciones inalámbricas del inglés *Wireless Application Protocol*).
- Dispositivo Móvil de Datos Básicos (*Basic Data Mobile Device*): Se caracterizan por tener una pantalla de mediano tamaño, menú o navegación basada en iconos, y ofrecer acceso a correos, lista de direcciones y SMS. En algunos casos estos dispositivos poseen un navegador web básico. Un ejemplo son los teléfonos inteligentes o *smartphones*.
- Dispositivo Móvil de Datos Mejorados (*Enhanced Data Mobile Device*): Se caracterizan por tener pantallas de medianas a grandes (por encima de los 240 x 120 píxeles), navegación de tipo *stylus* (accesorio de computadoras, generalmente una PDA o un *Smartphone* con forma alargada y estrecha similar a un bolígrafo moderno que permite ayudar a navegar o alcanzar una mayor precisión al señalar en pantallas táctiles). Además, pueden ofrecer las mismas características que el "Dispositivo Móvil de Datos Básicos", más las aplicaciones nativas<sup>4</sup> y corporativas<sup>5</sup> usuales. Este tipo de dispositivos incluyen Sistemas Operativos como Windows Mobile, IOS y Android.

### 1.3.2 Android

Android es un SO con una plataforma abierta para dispositivos móviles adquirido por Google y la *Open Handset Alliance* (alianza comercial de 84 compañías que se dedican a desarrollar estándares abiertos para dispositivos móviles). Su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones (11).

Android bajo la definición de Google se considera un *software stack* o una pila de *software*, ya que está conformada por (11):

- El SO, donde todas las funciones se desarrollan.

---

<sup>4</sup> Desarrolladas específicamente para un SO por ejemplo Microsoft Office Mobile (Word, Excel, Power Point).

<sup>5</sup> Permiten ejecutar y optimizar distintos aspectos como los sistemas de ventas, finanzas, operaciones bancarias, compras, fabricación e inventarios.

# Capítulo 1: Fundamentación Teórica de la Investigación

---

- El *middleware* o lógica de intercambio de información entre aplicaciones que permite la conexión entre redes.
- Las aplicaciones o Interfaces de Programación de Aplicaciones (en lo adelante APIs) que constituyen todos los programas que el dispositivo puede ejecutar.

Este SO se torna realmente atractivo por diversas características, entre ellas se encuentran:

- Es multitareas permitiendo mantener distintas aplicaciones corriendo al mismo tiempo.
- Plataforma totalmente libre basado en Linux que permite desarrollar aplicaciones y/o modificar las ya existentes con lenguaje de Java.
- Compatible con una gran variedad de *hardware* en el mercado (tabletas y dispositivos celulares de marcas como: Motorola, Samsung, ZTE, Huawei, LG) permitiendo al usuario elegir el dispositivo que mejor se ajusta a sus necesidades.
- Posee una plataforma de distribución digital de aplicaciones móviles (APK) llamada Google Play Store (anteriormente Android Market) donde se tiene acceso a muchas aplicaciones que pueden ser utilizadas.
- Permite realizar actualizaciones del SO en línea, siempre y cuando el dispositivo cumpla los requerimientos necesarios.
- Puede operar soluciones tecnológicas referentes al uso de redes sociales, mensajería instantánea, correo electrónico, modificación y lectura de procesadores de palabras, hojas de cálculo, presentaciones, lectura de formatos pdf.
- Se puede conseguir mucha información a través de documentos web o libros.
- Cuenta con el gran apoyo y la capacidad tecnológica proporcionada por “Google”.

Se escoge Android como SO a emplear porque es un software de código abierto, lo cual es útil para los desarrolladores, quienes pueden experimentar y probar e introducir sus particularidades. Está presente en gran variedad de dispositivos móviles lo que fomenta el uso y el desarrollo de aplicaciones para dicho SO. Además, domina el mercado de dispositivos móviles en Cuba y cuenta con una amplia comunidad de desarrolladores que fomenta activamente la retroalimentación.

### 1.3.3 APK

Un APK o un archivo con extensión .apk contiene una aplicación para el SO Android. Es un acrónimo de su nombre en inglés *Android Application Package*. Es muy parecido a los archivos ejecutables que se tiene en Windows (.exe) desde los que se instalan softwares. El formato APK es básicamente un formato de compresión parecido al ZIP (*Zone Improvement Plan*) y que dentro contiene todos los archivos necesarios

## *Capítulo 1: Fundamentación Teórica de la Investigación*

---

para que funcione una aplicación en Android. Un archivo APK contiene, en la mayoría de los casos, el instalador para aplicaciones o juegos. El SO se encarga de instalar el archivo APK y utilizar cada uno de los archivos que se necesitan para hacer funcionar la aplicación (12).

### **1.3.4 SIGIES**

De acuerdo a la documentación existente se define el Sistema de Gestión para el Ingreso a la Educación Superior (SIGIES) como:

Un producto de software que gestiona los subprocesos de **Organización, Exámenes, Asignación y Otorgamiento** comprendidos en el proceso de Ingreso a la Educación Superior. El sistema tiene como objetivo general informatizar el proceso de Ingreso a la Educación Superior utilizando tecnologías libres, con un despliegue nacional en cada Comisión de Ingreso Provincial (CIP) del país incluyendo el municipio especial Isla de la Juventud. Permite gestionar toda la información correspondiente al proceso de organización y realización de los exámenes. De igual forma se encarga de tramitar todo lo relacionado con los procesos de asignación y otorgamiento de plazas. Además, brinda la posibilidad de generar reportes a cerca de cada uno de los procesos mencionados anteriormente, así como de las estadísticas del proceso de ingreso en general.

### **1.3.5 Servicio web**

Un servicio web es un componente al que se puede acceder mediante protocolos Web estándar, utilizando un formato para el intercambio de datos. Normalmente se refiere con servicio web a una colección de procedimientos (métodos) a los que se pueden llamar desde cualquier lugar de Internet. Siendo este mecanismo de invocación totalmente independiente de la plataforma que se utilice y del lenguaje de programación en el que se haya implementado internamente el servicio por lo que se pueden utilizar para integrar aplicaciones escritas en diferentes lenguajes y que se ejecutan en plataformas diferentes (4).

El W3C (*World Wide Web Consortium*) define un servicio web como un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web y que intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios los solicitan llamando a estos procedimientos a través de la Web. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario (13).

# Capítulo 1: Fundamentación Teórica de la Investigación

---

Después de un estudio realizado sobre los conceptos presentados se decide definir que un servicio web no es más que un conjunto de protocolos y estándares que se utilizan para el intercambio de información entre distintas aplicaciones desarrolladas en diferentes lenguajes y que se ejecuten diferentes plataformas.

## Características de un Servicio web (4):

- Accesible a través de la Web por lo que se deben utilizar protocolos de transporte de datos, estándares como HTTP y codificar los mensajes en un lenguaje estándar que pueda conocer cualquier cliente que quiera utilizar el servicio.
- Contiene una descripción de sí mismo para saber cuál es su función e interfaz, de manera que pueda ser utilizado de forma automática por cualquier aplicación, sin la intervención del usuario.
- Localizable mediante algún mecanismo que permita encontrar un Servicio Web que realice una determinada función; posibilitando que una aplicación localice el servicio que necesite de forma automática.

### 1.3.6 Sincronización de datos

Según el colectivo de *Microsoft Corporation* la sincronización de los datos se refiere al proceso de propagación de los cambios en los datos entre el publicador y los suscriptores (consumidores de los datos) de manera que ambos contengan la misma versión de archivos en los dispositivos. La sincronización puede producirse (14):

- De forma continua, lo que es típico de la replicación transaccional.
- A petición, lo que es típico de la replicación de mezcla.
- Según una programación, lo que es típico de la replicación de instantáneas.

Los autores del Diccionario de Informática y Tecnología se refieren a la sincronización como: “el proceso del establecimiento de consistencia entre datos en fuentes remotas y de la continua armonización de los datos en el tiempo” (15).

Después de un estudio realizado sobre los conceptos presentados se decide trabajar con el de los autores Diccionario de Informática y Tecnología porque se adapta a las características de la investigación, así como el que aporta mayor entendimiento sobre el concepto en sí.

## 1.4 Tecnologías

### 1.4.1 Kit de desarrollo de *software*

Un SDK (*Software Development Kit*), o kit de desarrollo de *software*, es un conjunto de herramientas que ayudan a la programación de aplicaciones para un entorno tecnológico particular. Es decir, las aplicaciones desarrolladas sobre el SDK estarán destinadas a algún SO, plataforma, *hardware*, consola de videojuegos o paquete de *software* en especial. Son muchos los recursos que puede contener un SDK, a continuación, se mencionan algunos de ellos (16):

- Una API puede verse como una abstracción del funcionamiento interno del entorno sobre el que se va a desarrollar. Se trata de un conjunto de funciones, rutinas, estructuras de datos, clases y variables que permiten manipular el mecanismo de la plataforma sin conocerlo internamente.
- Un entorno de desarrollo integrado (IDE) es un editor que ayuda a escribir fácilmente el código fuente del programa. Generalmente, también brinda una interfaz amigable para dos aplicaciones fundamentales:
  - *Debugger* (depurador): Permite probar el programa en cada paso de su ejecución.
  - *Compiler* (compilador): Traduce el código fuente a lenguaje de máquina, obteniendo así un programa ejecutable.
- Código de ejemplo y otra documentación, como punto de partida para empezar a desarrollar aplicaciones.
- Un emulador del entorno. Por ejemplo, si se desarrolla una aplicación para móviles desde una computadora de escritorio, permite saber cómo la vería el usuario final.

### 1.4.2 Entorno de desarrollo integrado

Un Ambiente de Desarrollo Interactivo o Entorno de Desarrollo Integrado, del inglés *Integrated Development Environment* (IDE) es una aplicación de *software*, que proporciona servicios integrales para facilitarle al programador de computadora el desarrollo de *software*. Normalmente, un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador, además, la mayoría tienen auto-completado inteligente de código (17).

Algunos IDEs contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales como SharpDevelop y Lazarus. El límite entre un entorno de desarrollo integrado y otras partes del entorno de desarrollo de *software* más amplio no está bien definido. Muchas veces, a los efectos de

# Capítulo 1: Fundamentación Teórica de la Investigación

---

simplificar la construcción de la interfaz gráfica del usuario (por sus siglas en inglés GUI) se integra un sistema controlador de versión y varias herramientas. Muchos IDE modernos también cuentan con un navegador de clases, un buscador de objetos y un diagrama de jerarquía de clases, para su uso con el desarrollo de *software* orientado a objetos (17).

Los entornos de desarrollo integrados están diseñados para maximizar la productividad del programador proporcionando componentes muy unidos con interfaces de usuario similares. Los IDE presentan un único programa en el que se lleva a cabo todo el desarrollo. Generalmente, este programa suele ofrecer muchas características para la creación, modificación, compilación, implementación y depuración de *software* (17). A continuación, se menciona el IDE a utilizar para el desarrollo de la propuesta de solución:

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Además del potente editor de códigos y las herramientas para desarrolladores, Android Studio 3.1 ofrece funciones que aumentan la productividad durante la compilación de aplicaciones para para Android, como las que se mencionan a continuación (18):

- Un sistema de compilación basado en Gradle flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que se pueden realizar desarrollos para todos los dispositivos Android.
- Instant Run para aplicar cambios mientras la aplicación se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudar a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y *frameworks* de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad y compatibilidad de versión.
- Compatibilidad con C++ y NDK.

### 1.4.3 Lenguaje de desarrollo Java

Java es un lenguaje de programación concurrente y orientado a objetos, diseñado para ser ejecutado en cualquier dispositivo sin la necesidad de ser recompilado ya que las aplicaciones son ejecutadas por una plataforma que es la encargada de interpretar la aplicación y ejecutarla. Su filosofía es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como "*write once, run anywhere*" o WORA) (19).

**Ventajas de Java** (19):

# Capítulo 1: Fundamentación Teórica de la Investigación

---

- Orientado a objetos: La programación orientada a objetos resulta muy conveniente para la mayoría de las aplicaciones, y es esencial para los videojuegos. Entre las ventajas más evidentes que ofrece se encuentra un gran control sobre el código y una mejor organización, dado que basta con escribir una vez los métodos y las propiedades de un objeto, independientemente de la cantidad de veces que se utilicen.
- Flexible: Lenguaje especialmente preparado para la reutilización del código; permite a los usuarios tomar un programa que se haya desarrollado tiempo atrás y actualizarlo con mucha facilidad, ya sea para agregar funciones o adaptarlo a un nuevo entorno.
- Funciona en cualquier plataforma: A diferencia de los programas que requieren de versiones específicas para cada SO, las aplicaciones desarrolladas en Java funcionan en cualquier entorno, dado que no es el sistema quien las ejecuta, sino la máquina virtual (conocida como *Java Virtual Machine* o *JVM*), permitiendo a los programadores desentenderse de la compatibilidad con el hardware.
- Su uso no acarrea inversiones económicas: Programar en Java es gratis pues no es necesario adquirir licencia, solo descargar el kit de desarrollo (*Java Development Kit* o *JDK*).
- Fuente abierta: Ofrece el código de casi todas sus librerías nativas para que se puedan conocer y estudiar en profundidad, o bien ampliar su funcionalidad, beneficiándose uno mismo y a los demás.
- Es un lenguaje expandible: Cada programador tiene la libertad de revisar y mejorar el código nativo de Java, y su trabajo puede convertirse en la solución a los problemas de muchas personas en todo el mundo.

## 1.4.4 Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering*) son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el costo de las mismas en términos de tiempo y dinero. Algunos objetivos a los que están dirigidas son (20):

- Permitir la aplicación práctica de metodologías estructuradas, que al ser realizadas con una herramienta consiguen agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes *software*.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

# Capítulo 1: Fundamentación Teórica de la Investigación

---

**Visual Paradigm:** Es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de vida de desarrollo de un software (21). La versión de la herramienta usada es 8.0 y algunas de sus ventajas son:

- Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- Integración con otras aplicaciones. Permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad.
- Trabajo multiusuario. Permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.
- Reutilización. Facilita la reutilización, ya que disponemos de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos.
- Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta.

## 1.4.5 Balsamiq

Balsamiq es una herramienta de diseño de interfaz de usuario para crear prototipos (a veces llamados maquetas). Se puede usar para generar bocetos digitales de una idea o concepto para una aplicación o sitio web, además de, facilitar la discusión y la comprensión antes de escribir cualquier código. Los prototipos completados se pueden usar para realizar pruebas a los usuarios, aclarar su visión, obtener retroalimentación de las partes interesadas u obtener aprobación para comenzar el desarrollo. Es fácil de usar, cuenta con una amplia gama de objetos prediseñados como menús, listas, botones, contenedores e iconos (22). Debido a su facilidad de uso y sus múltiples opciones gráficas se utilizó esta herramienta en su versión 3.5.15 para el diseño de las interfaces de usuario.

## 1.5 Metodología de desarrollo

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo

# Capítulo 1: Fundamentación Teórica de la Investigación

---

ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. (23).

En la actualidad la rapidez y el dinamismo en la industria del *software* han hecho replantear los cimientos sobre los que se sustenta el desarrollo de *software* tradicional. El mercado está marcando la tendencia en la ingeniería del *software* teniendo como características principales atender a las necesidades de rapidez, flexibilidad y variantes externas que hacen del entorno una ventaja más competitiva al aumentar la productividad y satisfacer las necesidades del cliente en el menor tiempo posible para proporcionar mayor valor al negocio. Ante esta situación, el grado de adaptación de las metodologías tradicionales a los entornos de trabajo no son del todo eficientes y no cubren las necesidades del mercado actual. Existen una gran cantidad de metodologías para el desarrollo de *software*, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías ágiles (23).

Las metodologías tradicionales se basan en las buenas prácticas dentro de la ingeniería del *software*, siguiendo un marco de disciplina estricto y un riguroso proceso de aplicación. Las metodologías ágiles, en cambio, representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible y con cambios constantes, omitiendo documentación rigurosa y métodos formales (23).

Las metodologías ágiles basan su fundamento en la adaptabilidad de los procesos de desarrollo, un modelo de desarrollo ágil generalmente es un proceso incremental, cooperativo, sencillo y adaptativo, poniendo de relevancia que la capacidad de respuesta de un cambio es más importante que el seguimiento estricto de un plan (23).

La metodología seleccionada para el desarrollo se basa en la metodología utilizada para los proyectos productivos de la UCI para lograr una estandarización en los productos con propósito de la certificación de la norma CMMI (Integración de Modelos de Madurez de Capacidad del inglés Capability Maturity Model Integration). La metodología que se propone es el Proceso Unificado Ágil del inglés *Agile Unified Process* de Scott Ambler la cual es una versión simplificada del Proceso Racional Unificado (RUP). Este describe de una manera simple y fácil como entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP (24).

El AUP aplica técnicas ágiles incluyendo (25):

- Desarrollo Dirigido por Pruebas del inglés *Test Driven Development* (TDD)
- Modelado Ágil
- Gestión de Cambios Ágil
- Refactorización de Base de Datos para mejorar la productividad

# Capítulo 1: Fundamentación Teórica de la Investigación

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva.

- Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

## Variación de AUP para la UCI

Al no existir una metodología de *software* universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de *software* tiene entre sus objetivos aumentar la calidad del *software* que se produce, de ahí la importancia de aplicar buenas prácticas, para ello se contó con el Modelo CMMI-DEV v1.3. que constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora, dichas prácticas se centran en el desarrollo de productos y servicios de calidad (24).

## Descripción de las fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que se llama Ejecución y se agrega la fase de Cierre. Para una mayor comprensión se muestra la siguiente tabla (24).

Tabla 2 Fases de AUP y AUP-UCI

| Fases AUP | Fases Variación AUP-UCI | Objetivos de las fases (Variación AUP-UCI)   |
|-----------|-------------------------|--|
| Inicio    | Inicio                  | Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de |

# Capítulo 1: Fundamentación Teórica de la Investigación

|              |           |   |
|--------------|-----------|---|
|              |           | tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.  |
| Elaboración  | Ejecución | En esta fase se ejecutan las actividades requeridas para desarrollar el <i>software</i> , incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. |
| Construcción |           |   |
| Transición   |           |   |
|              | Cierre    | En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.   |

## Escenarios para la disciplina Requisitos

A partir de que el Modelado de negocio propone tres variantes a utilizar en los proyectos (Casos de Uso del Negocio (CUN), Descripción de Procesos del Negocio (DPN) o Modelo Conceptual (MC)) y existen tres formas de encapsular los requisitos (CUS, HU, DRP), surgen cuatro escenarios para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, quedando de la siguiente forma (24):

Escenario No 1:

Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS

$$\text{CUN} + \text{MC} = \text{CUS}$$

Escenario No 2:

Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS

$$\text{MC} = \text{CUS}$$

Escenario No 3:

Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

$$\text{DPN} + \text{MC} = \text{DRP}$$

Escenario No 4:

Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

$$\text{HU}$$

# *Capítulo 1: Fundamentación Teórica de la Investigación*

---

Durante la investigación se utilizará el escenario número cuatro porque aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado se obtiene un negocio muy bien definido. El cliente estará siempre acompañando al desarrollador para convenir los detalles de los requisitos de manera que se puedan implementar, probar y validar. Además, es recomendada en proyectos no muy extensos, ya que una Historia de Usuario no debe poseer demasiada información.

## **Conclusiones parciales**

Una vez concluido el desarrollo del capítulo se arriba a las siguientes conclusiones:

- Se realizó un análisis del estado del arte sobre las aplicaciones móviles que consumen servicios web para determinar algunas características que debe contener la aplicación, así como las herramientas necesarias para la confección de la misma.
- Se determinaron características que debe contener la aplicación tales como: buscar y filtrar la información, REST como protocolo para obtener información de la capa de servicios web y SQLite como sistema gestor de bases de datos.
- Se decide utilizar como base para la implementación de la solución el SO Android v4.0 y se hará uso del IDE Android Studio v3.1, utilizando como lenguaje de programación Java. Como herramienta para el modelado Visual Paradigm v8.0 empleando el lenguaje de modelado UML y para guiar el proceso de desarrollo y generar los artefactos ingenieriles se seleccionó la metodología AUP variante UCI, en su escenario cuatro.

### Capítulo 2: Propuesta de solución

#### 2.1 Introducción

En el presente capítulo se muestra el modelo de dominio para representar la estructura de la solución, así como la relación entre las diferentes clases con el objetivo de vincular los conceptos asociados a la investigación. Se realiza el levantamiento y especificación de requisitos, tanto funcionales como no funcionales necesarios a cumplir para la elaboración de la solución. Se formula la propuesta de solución, así como la arquitectura y los patrones empleados en el desarrollo de la aplicación.

#### 2.2 Descripción de la propuesta de solución

La propuesta de solución consiste en la implementación de una aplicación para dispositivos móviles con SO Android (APK) que sea capaz de proveer información no confidencial generada por el SIGIES y de interés para las personas implicadas en el proceso de ingreso. Se define como información no confidencial los datos que no se consideran sensible para el proceso de ingreso.

Estos pueden ser consultados por los usuarios que tengan un dispositivo móvil con la aplicación instalada y estén conectados a una red inalámbrica. Teniendo en cuenta la privacidad del usuario la información es clasificada en dos categorías:

- Pública: Es la información que puede ser accedida por el usuario sin necesidad de autenticarse. Esta puede ser:
  - Plan de plazas provincial.
- Personal: Es la información que puede ser accedida por el usuario autenticándose mediante el escaneo del código QR de un CI perteneciente a un aspirante a la Educación Superior existente en la base de datos del proceso de ingreso. Esta información puede ser:
  - Datos del estudiante.
  - Datos de la vía de ingreso del estudiante.
  - Plazas solicitadas por el estudiante.
  - Exámenes exonerados del estudiante.
  - Actas de exámenes referente al estudiante.
  - Nota de los exámenes del estudiante.
  - Asistencia del estudiante a los exámenes.
  - Resultado de la asignación del estudiante.
  - Estado del servicio militar del estudiante.
  - Etapa actual del proceso de ingreso.

Para la correcta visualización y actualización de los datos en la aplicación, se establece una comunicación con la capa de servicios web de SIGIES. Esta comunicación se realiza mediante el consumo de servicios web utilizando el protocolo REST, permitiendo mejorar la accesibilidad de la información. Con la utilización de SQLite como sistema gestor de base de datos se garantiza el acceso y análisis de la información de forma *offline*, mejorando su portabilidad.

### 2.3 Modelo de dominio

Un diagrama de modelo de dominio no es más que una representación visual de clases conceptuales u objetos del mundo real en un dominio de interés que brinda conceptos significativos para el dominio del problema con el objetivo de conocer y comprender el negocio al cual el sistema va a servir (25).

A continuación, se mencionan los conceptos que se emplearon en el modelo de dominio:

**Plaza:** Es una carrera disponible en el plan de plazas de una provincia por la que el estudiante puede optar.

**Solicitud:** En una boleta con las opciones de carreras establecidas solicitadas por un estudiante.

**Plan de plazas:** Es un listado de carreras que el estudiante puede solicitar.

**Carrera:** Es un conjunto de cursos académicos que una persona debe cumplir para obtener un título profesional.

**Estudiante:** Es una persona que aspira al ingreso de la educación superior.

**Vía de ingreso:** Es el comportamiento que puede tener un estudiante para el otorgamiento de la carrera.

**Provincia:** Es una división administrativa territorial a la que pertenecen los estudiantes y donde se estudian las carreras.

**Examen:** Es una evaluación que se le realiza al estudiante para medir los conocimientos y habilidades en una determinada asignatura con el objetivo de determinar su idoneidad para cursar una carrera universitaria.

**Nota:** Es un valor numérico entre 0 y 100 asignado a un examen.

**Calificación:** Es una nota que alcanza el examen una vez revisado.

**Reclamación:** Es una nota que recibe el examen luego de una segunda revisión.

**Mostrado de examen:** Es una nota con la que finaliza la acción de mostrar el examen al estudiante.

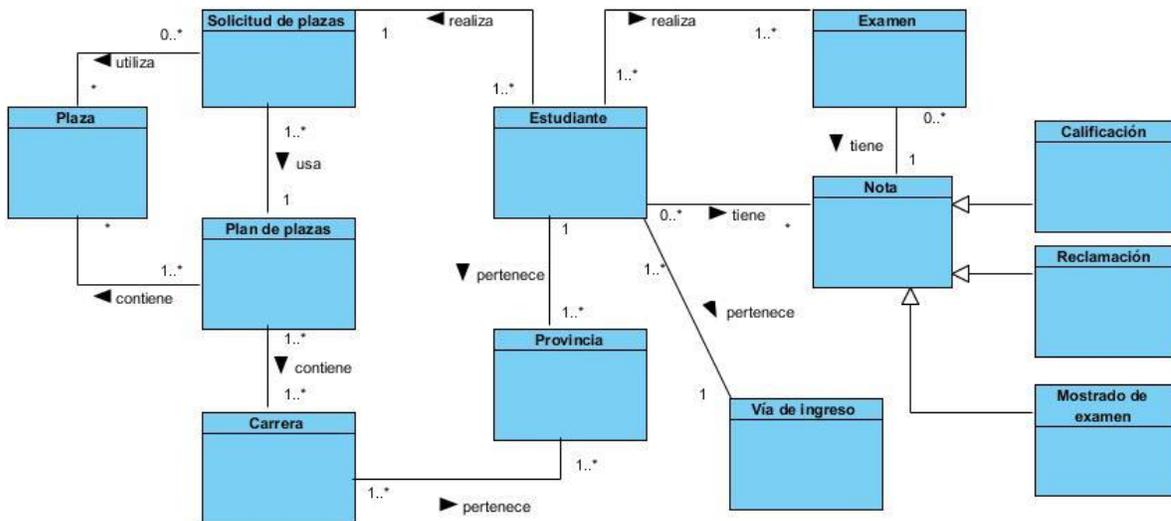


Fig. 1 Modelo de dominio

### 2.4 Requisitos de software

Los requisitos de *software* cumplen un papel primordial en el proceso de producción de *software*, ya que se enfocan en la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; minimizando problemas relacionados con la mala gestión de los requerimientos (26).

Se define como requisito a la descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso (26).

Los requisitos de *software* pueden dividirse en dos categorías: requisitos funcionales y requisitos no funcionales.

#### 2.4.1 Requisitos funcionales

Los requisitos funcionales definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas (26).

**RF1.**Autenticar usuario: La aplicación permite al usuario escanear el código QR (Respuesta Rápida del inglés *Quick Response*) de su carné de identidad y proveerle acceso a la información solicitada.

**RF2.**Sincronizar datos con la capa de servicios web: La aplicación permite sincronizar los datos que existen en la aplicación y en la capa de servicios web.

**RF3.**Listar provincias: La aplicación permite listar el nombre de las quince provincias y el municipio especial existentes en Cuba.

**RF4.**Listar plan de plazas provincial: La aplicación permite listar el plan de plazas de la provincia seleccionada con anterioridad. El listado muestra datos tales como nombre, vía de ingreso, modalidad y sexo.

**RF5.**Filtrar plan de plazas provincial: La aplicación permite filtrar el plan de plazas de la provincia seleccionada teniendo en cuenta algunos criterios de búsqueda tales como nombre, vía de ingreso, modalidad y sexo.

**RF6.**Mostrar datos de una carrera del plan de plazas provincial: La aplicación permite mostrar los datos de la carrera seleccionada en el plan de plazas provincial tales como convocatoria, procesamiento, vía de ingreso, sexo, modalidad, centro de educación superior, plazas disponibles, plazas disponibles iniciales, demanda, ingresos requeridos y provincia.

**RF7.**Mostrar solicitud del estudiante: La aplicación muestra la solicitud de plazas hecha por el estudiante. Muestra datos tales como nombre y centro de educación superior.

**RF8.**Mostrar notas por cada asignatura examinada: La aplicación muestra las notas de las asignaturas examinadas por el estudiante. Muestra datos tales como asignatura y nota.

**RF9.**Mostrar si está o no exonerado de algún examen: La aplicación muestra si el estudiante está exonerado de algún examen.

**RF10.**Mostrar datos de la vía de ingreso del estudiante: La aplicación muestra los datos de la vía de ingreso del estudiante tales como nombre, nombre corto, exámenes, modalidad, tipo de escalafón, índice académico, tipo de asignación, aporta personal al SMA y provincia de asignación.

**RF11.**Mostrar datos del estudiante: La aplicación muestra los datos del estudiante tales como nombre, nombre corto, exámenes, modalidad, tipo de escalafón, índice académico, tipo de asignación, aporta al SMA y provincia de asignación.

**RF12.**Mostrar asistencia: La aplicación muestra la asistencia a los exámenes del estudiante. Muestra datos tales como asignatura y asistencia.

**RF13.**Mostrar resultado de la asignación: La aplicación muestra el resultado de la asignación.

**RF14.**Mostrar estado del servicio militar: La aplicación muestra el estado del servicio militar.

**RF15.**Mostrar etapa en la que se encuentra el proceso: La aplicación muestra la etapa actual del proceso.

**RF16.**Mostrar actas de exámenes: La aplicación muestra las actas de exámenes. Muestra datos tales como signatura, número de acta y preuniversitario.

### 2.4.2 Requisitos no funcionales

Los requisitos no funcionales se asocian con características que de una u otra forma puedan limitar el sistema, no describen información a guardar, ni funciones a realizar, sino características de funcionamiento (26).

### **Usabilidad:**

**RNF1.** La aplicación debe ajustarse al entorno donde será usada y utilizar el castellano como lenguaje base, además podrá ser utilizada por cualquier usuario que tenga conocimientos básicos en el uso de aplicaciones para dispositivos móviles. Debe notificar al usuario ante la presencia de un error y siempre que sea posible las acciones similares deben ejecutarse de similar forma.

### **Portabilidad:**

**RNF2.** Se necesitará un dispositivo con Android 4.0 o superior. Se recomienda 512MB de RAM o superior.

### **Restricciones de diseño e implementación:**

**RNF3.** El lenguaje de desarrollo a utilizar será Java, el IDE será Android Studio y como motor de bases de datos SQLite.

### **Apariencia o interfaz externa:**

**RNF4.** El diseño de las interfaces será sencillo, con pocas imágenes y colores y la información aparecerá correctamente organizada de forma tal que el usuario, pueda encontrar lo que busca rápidamente.

### **Confiabilidad:**

**RNF5.** La veracidad de la información obtenida debe corresponder sin ninguna alteración con los elementos obtenidos como respuesta a la sincronización ejecutada.

### **Accesibilidad:**

**RNF6:** El menú debe ser de fácil acceso, las opciones importantes del menú no estarán ocultas, la cantidad de información y el número de objetos mostrados estarán limitados, se utilizarán iconos conocidos y conceptos de navegación sencillos además de utilizar encabezados breves y descriptivos para estructurar la información.

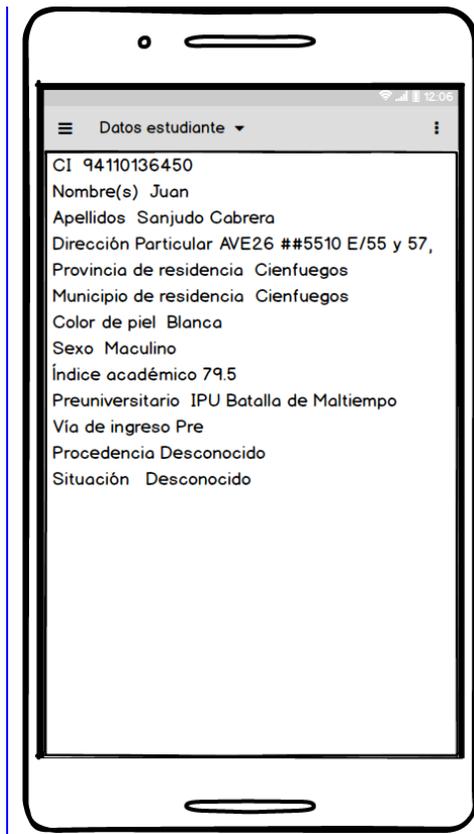
## **2.5 Historias de Usuario**

Las historias de usuario son requerimientos que expresan el problema que el sistema o producto software debe resolver. Las historias de usuario son un enfoque de requerimientos ágil que se focaliza en establecer conversaciones acerca de las necesidades de los clientes. Son descripciones cortas y simples de las funcionalidades del sistema, narradas desde la perspectiva de la persona que desea dicha funcionalidad, usualmente un usuario. Poseen las siguientes características: una descripción escrita que será utilizada para planificar y posteriormente disgregar los detalles con el dueño del producto, las conversaciones propiamente dichas con el dueño del producto y las pruebas que han de determinar si las historias están finalizadas o no (27).

A continuación, se muestra la HU correspondiente al **RF11 Mostrar datos del estudiante**. Las demás historias de usuario pueden ser consultadas en los [anexos](#).

Tabla 3 HU Mostrar datos del estudiante

|  |   |
|--|---|
|  |   |
| <b>Número:</b> 11  | <b>Nombre del requisito:</b> Mostrar datos del estudiante |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era                           |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días                             |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días                                |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p>Permitir consultar los datos de la vía de ingreso del estudiante tales como:</p> <ul style="list-style-type: none"> <li>▪ CI</li> <li>▪ Nombre(s)</li> <li>▪ Apellidos</li> <li>▪ Dirección Particular</li> <li>▪ Provincia de residencia</li> <li>▪ Municipio de residencia</li> <li>▪ Color de piel</li> <li>▪ Sexo</li> <li>▪ Índice académico</li> <li>▪ Preuniversitario</li> <li>▪ Vía de ingreso</li> <li>▪ Procedencia</li> <li>▪ Situación</li> </ul> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> <p>Para ver los datos del estudiante hay que:</p> <ul style="list-style-type: none"> <li>-Estar autenticado en el sistema</li> <li>-Deben existir los datos del estudiante en el sistema.</li> </ul> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> <p>Cuando el usuario selecciona la acción mostrar datos del estudiante se muestra al usuario un listado con los datos del estudiante.</p> |   |
| <b>Observaciones:</b>  |   |
| <b>Prototipo de interfaz:</b>  |   |



### 2.6 Descripción del patrón arquitectónico

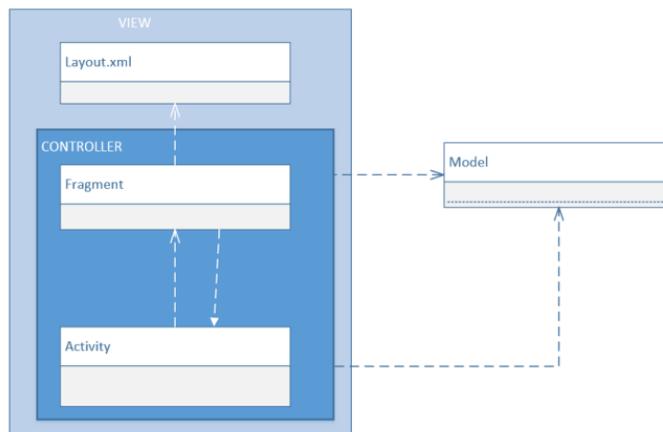
En el desarrollo de la solución propuesta se utilizó el patrón arquitectónico Modelo Vista Controlador (en lo adelante MVC). Su principal objetivo consiste en separar los datos de una aplicación (Modelo), la interfaz de usuario (Vista) y la lógica de negocios (Controlador) en tres componentes distintos que al final conforman la aplicación. Propicia la facilidad de mantenimiento, reutilización del código y separación de conceptos (28).

**Modelo:** Es la capa donde se trabaja con los datos, por tanto, contiene mecanismos para acceder a la información y también para actualizar su estado. Los datos se tienen habitualmente en una base de datos de manera que los modelos tendrán todas las funciones que accederán a las tablas y harán los correspondientes *selects*, *updates*, *inserts* (28).

**Vista:** La vista, contiene el código de la aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que permite renderizar los estados de la aplicación. En las vistas nada más se tiene los códigos que permiten mostrar la salida. En la vista generalmente se trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requieren los datos a los modelos y ellas se genera la salida, tal como requiera la aplicación (28).

**Controlador:** Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información. Es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo (28).

En Android el patrón arquitectónico MVC posee una particularidad en cuanto a la vista y el controlador, definiéndose estos dos como un par Controlador-Vista. En este caso particular el controlador notifica al modelo y el modelo notifica a todos los dependientes (pares Controlador-Vista) asociados a él. Para este par la vista contiene una instancia de su controladora y la controladora además posee una instancia de la vista. Las vistas (layouts) son archivos .xml que contienen los componentes que conforman la Interfaz de Usuario (UI), las clases controladoras (.java) contienen las instancias de cada uno de estos componentes responsabilizándose por el control de los mismos, mientras que el modelo contiene una estructura de datos de objetos complejos o simplemente datos primarios (29).



**Fig. 2 Patrón arquitectónico MVC para Android**

### 2.7 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En lugar de reinventar soluciones se pueden utilizar patrones que representan soluciones ya probadas y documentadas a problemas de desarrollo de software que están sujetos a contextos similares tales como acceso a datos, creación de objetos y operaciones entre sistemas. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (30).

### 2.7.1 Patrones GRASP

Los patrones GRASP (Patrones de Software de Asignación de Responsabilidades Generales del inglés *General Responsibility Assignment Software Patterns*) describen los principios fundamentales para asignar responsabilidades a los objetos (31).

En el desarrollo de la aplicación se utilizarán algunos patrones GRASP estos son:

- **Experto:** La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Este patrón se puede observar en la clase `Scan.java` que es la encargada de controlar en la aplicación la tarea de autenticación.
- **Controlador:** Representa al sistema completo y es el manejador artificial de los eventos. Consiste en la asignación de responsabilidades para controlar el flujo de los eventos que genera el sistema a clases específicas. Su uso se evidencia en la clase `Principal.java` pues en esta se manejan la mayoría de los eventos realizados por la aplicación.

### 2.7.2 Patrones GOF

Los patrones GOF (Pandilla de los Cuatro del inglés *Gang of Four*) fueron creados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Ellos recopilaron y documentaron 23 patrones de diseño aplicados por expertos diseñadores de software orientado a objetos. Estos patrones se clasifican en 3 categorías basadas en su propósito (32):

- **Creacionales:** Inicialización y configuración de objetos, ayudan a encapsular y abstraer dichos objetos.
- **Estructurales:** Separan la interfaz de la implementación, se ocupan de cómo las clases y objetos se agrupan para formar estructuras más grandes.
- **De comportamiento:** Describen la comunicación entre objetos o clases.

En el desarrollo de la aplicación se utilizaron algunos patrones GOF, estos son:

- **Modelo Vista Controlador:** Este patrón pertenece a la categoría creacional y plantea la separación del problema en tres capas, el modelo representa la realidad; el controlador conoce los métodos y atributos del modelo, recibe y realiza lo que el usuario quiere hacer; y la vista muestra un aspecto del modelo y es utilizada por el controlador para interactuar con el usuario. Este patrón se evidencia en toda la aplicación.



de servicios web que en este caso será la encargada de brindar la información solicitada. La comunicación entre el dispositivo móvil donde deberá estar instalada la aplicación y la capa de servicios web se realizará a través del protocolo HTTP (Protocolo de Transferencia de Hipertexto, del inglés Hypertext Transfer Protocol). La conexión entre ambos dispositivos se realizará mediante conexiones inalámbricas (WIFI) a través de un punto de acceso.

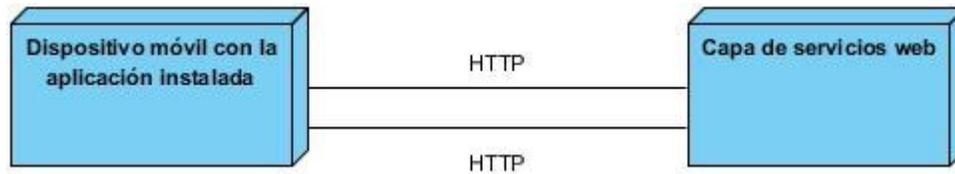


Fig. 4 Diagrama de despliegue

### 2.9 Conclusiones del capítulo

Luego de haber concluido el presente capítulo se arribó a las siguientes conclusiones:

- La realización del modelo de dominio permitió representar y comprender los conceptos más significativos que intervienen en el desarrollo de la aplicación.
- La identificación de los requisitos funcionales y no funcionales y la elaboración de las correspondientes historias de usuarios propiciaron las funcionalidades que brindará la aplicación y la guía para la implementación de las mismas.
- Se seleccionó el patrón arquitectónico y los patrones de diseño con el objetivo de facilitar las tareas de implementación.
- Se realizó el diagrama de clases del diseño con el propósito de que la aplicación cumpla con las funcionalidades identificadas.

### **Capítulo 3: Implementación y pruebas**

#### **3.1 Introducción**

En este capítulo se documentan los principales aspectos relacionados con la implementación de los elementos identificados durante la realización del diseño. Se muestra la organización y las dependencias lógicas de los componentes mediante el diagrama de componente. Además, se exponen los estándares de codificación a utilizar con el propósito de lograr una organización en el código, así como los resultados de las pruebas y las validaciones realizadas a la aplicación, comprobando que las funcionalidades desarrolladas cumplan con los requisitos establecidos.

#### **3.2 Diagrama de componente**

Un diagrama de componentes muestra los elementos del diseño de un sistema de software. Permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces. Además, muestra la relación entre componentes de software ya sean componentes de código fuente, binarios o ejecutables, así como sus dependencias, comunicación y ubicación (35).

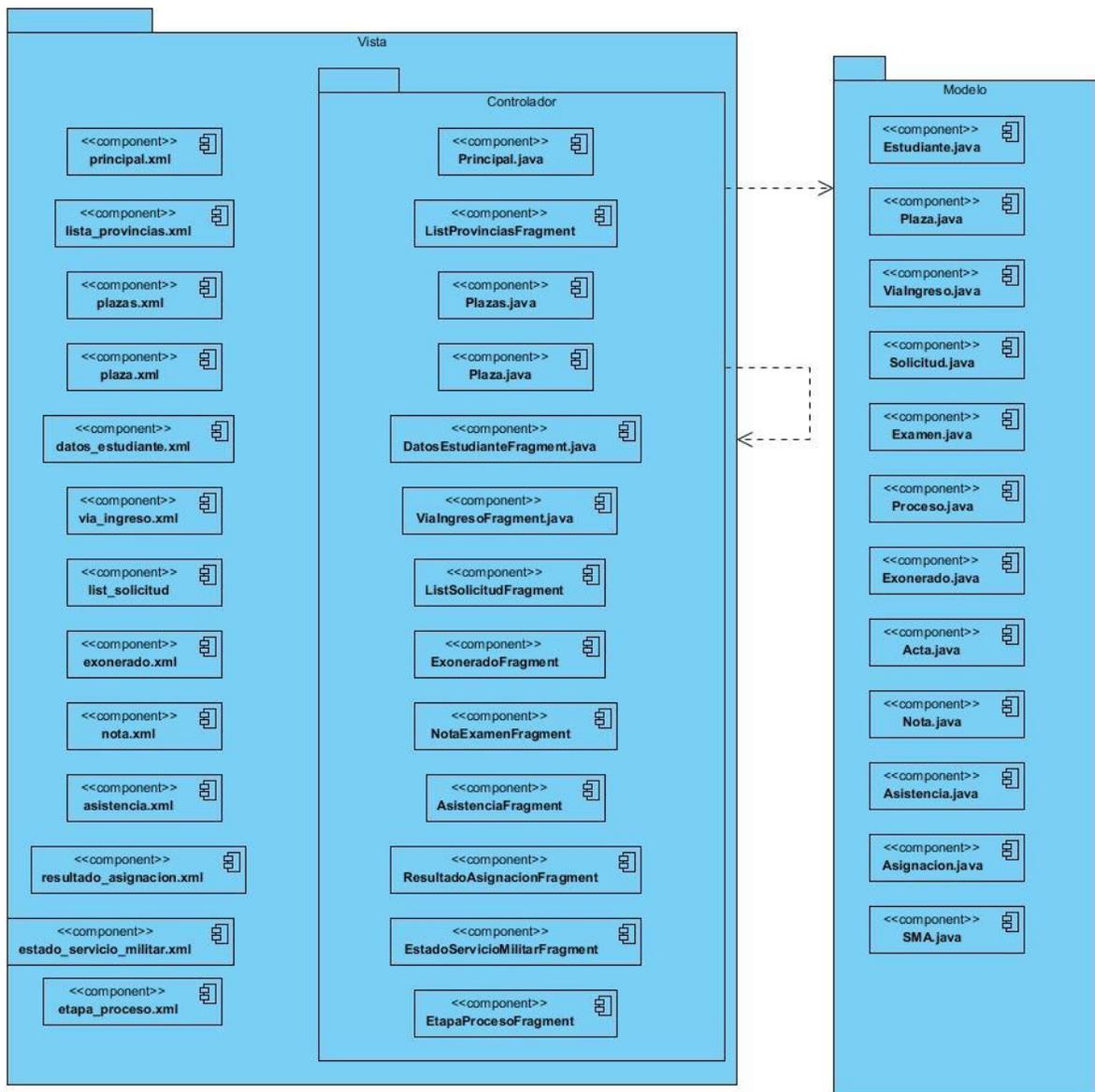


Fig. 5 Diagrama de componente

### 3.3 Estándares de codificación

Los estándares de código, son parte de las llamadas buenas prácticas o mejores prácticas, estas son un conjunto no formal de reglas, que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad del código notablemente. Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto permite que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea más fácil de mantener (36).

Se entiende como estándar de código a un conjunto de convenciones establecidas de antemano (denominaciones, formatos) para la escritura de código. Estos estándares varían dependiendo del lenguaje

## Capítulo 3: Implementación y pruebas

de programación elegido y además varían en cobertura, algunos son más extensos que otros. Pero hay algunos puntos que todo estándar debería cubrir (36):

- Comentarios: Ayudan a entender y revisar más fácilmente el código
- Convenio de denominaciones: Proporciona una guía clara de cómo se deben elegir los nombres de clases, funciones y variables
- Legibilidad: Provee facilidad de lectura y buena estructuración visual que acompaña la lógica del problema

En el presente trabajo se tomó como referencia el estándar de codificación basado en convenciones de código para el lenguaje de programación Java, el cual establece una serie de parámetros para el trabajo de codificación.

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas (Ej. Principal, DatosEstudiante). Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo). Los nombres de las interfaces y los fragments siguen la misma regla que las clases (37).

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo forma en mayúscula (Ej. onViewCreated() ) (37).

Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula, deben ser cortos, pero con significado excepto para variables índices temporales. Las palabras restantes que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres guion bajo "\_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje (37).

### 3.4. Casos de Prueba

Los casos de prueba contienen un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y postcondiciones de ejecución, desarrollados para un objetivo particular de condición de prueba, tal como para ejercer una ruta de un programa en particular o para verificar el cumplimiento de un requisito específico (38).

A continuación, se muestra el Diseño de caso de prueba correspondiente al **RF11**, los demás se encuentran en los [anexos](#).

**Tabla 4 Modelo de caso de prueba correspondiente al RF11**

|  |
|--|
| <b>Descripción general:</b> permitir consultar los datos de un estudiante. |
|--|

| <b>Condiciones de ejecución:</b> para consultar los datos de un estudiante hay que estar autenticado y debe existir en la base de datos al menos un estudiante. |   |   |                                |
|---|---|---|--------------------------------|
| <b>Escenario</b>  | <b>Descripción</b>  | <b>Respuesta del sistema</b>  | <b>Flujo central</b>           |
| <b>EC 1.1</b><br>Consultar datos de un estudiante   | El usuario selecciona la opción<br>Mostrar datos del estudiante | El sistema muestra los datos de un estudiante, tales como: <ul style="list-style-type: none"><li>▪ CI</li><li>▪ Nombre</li><li>▪ Dirección particular</li><li>▪ Provincia de residencia</li><li>▪ Municipio de residencia</li><li>▪ Color de piel</li><li>▪ Sexo</li><li>▪ Índice académico</li><li>▪ Vía de ingreso</li><li>▪ Preuniversitario</li><li>▪ Procedencia</li><li>▪ Situación</li></ul> | Principal/Menú/DatosEstudiante |

### 3.5 Pruebas de software

Las pruebas de software son una técnica dinámica de validación y verificación (V&V), las cuales incluyen un conjunto de tareas que garantizan que el software implemente correctamente funciones específicas y aseguran que se construye cumpliendo con los requerimientos del cliente (38).

#### 3.5.1 Pruebas de rendimiento

Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación, los cuales generan bajo rendimiento. Este tipo de pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones de negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento y requerimientos no funcionales (39).

## Capítulo 3: Implementación y pruebas

Para la realización de estas pruebas se utilizó *Android Device Monitor* la cual es una herramienta que proporciona una interfaz gráfica de usuario para depurar y analizar aplicaciones Android. Dicha herramienta está incluida en *Android Studio*.

| Características de dispositivo |                     |         | Rendimiento |              |
|--------------------------------|---------------------|---------|-------------|--------------|
| Dispositivo móvil              | RAM del dispositivo | Android | RAM         | Red          |
| LG Tribute                     | 1536 MB             | 6.0.1   | 2837 KB     | 4 106, 25 KB |
| Winstar                        | 512 MB              | 4.2.2   | 2748 KB     | 3 978, 15 KB |
| Samsung Galaxy Core Prime      | 1024 MB             | 4.4.4   | 2957 KB     | 4 003, 75 KB |
| Samsung Galaxy S8              | 4096 MB             | 7.0     | 2808 KB     | 4250 KB      |

Después de realizar las pruebas de rendimiento se concluye que la aplicación tuvo un comportamiento satisfactorio en diferentes dispositivos durante una hora de prueba.

### 3.5.2 Pruebas de aceptación

Son pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (40).

#### Resultados de las pruebas

Con el objetivo de comprobar el cumplimiento de los requisitos propuestos se hace uso de las pruebas de caja negra teniendo en cuenta las pruebas de aceptación detectando y corrigiendo algunas no conformidades. Para ello se efectuaron tres iteraciones, en las cuales se detectaron 66 no conformidades; de estas 6 se clasificaron de complejidad alta, 42 de complejidad media y 18 de complejidad baja.

Las no conformidades fueron clasificadas según su complejidad por lo que se definen de la siguiente forma, arrojando los resultados que se exponen:

- Alta son errores de funcionalidad
- Media son errores de interfaz
- Baja son errores de ortografía

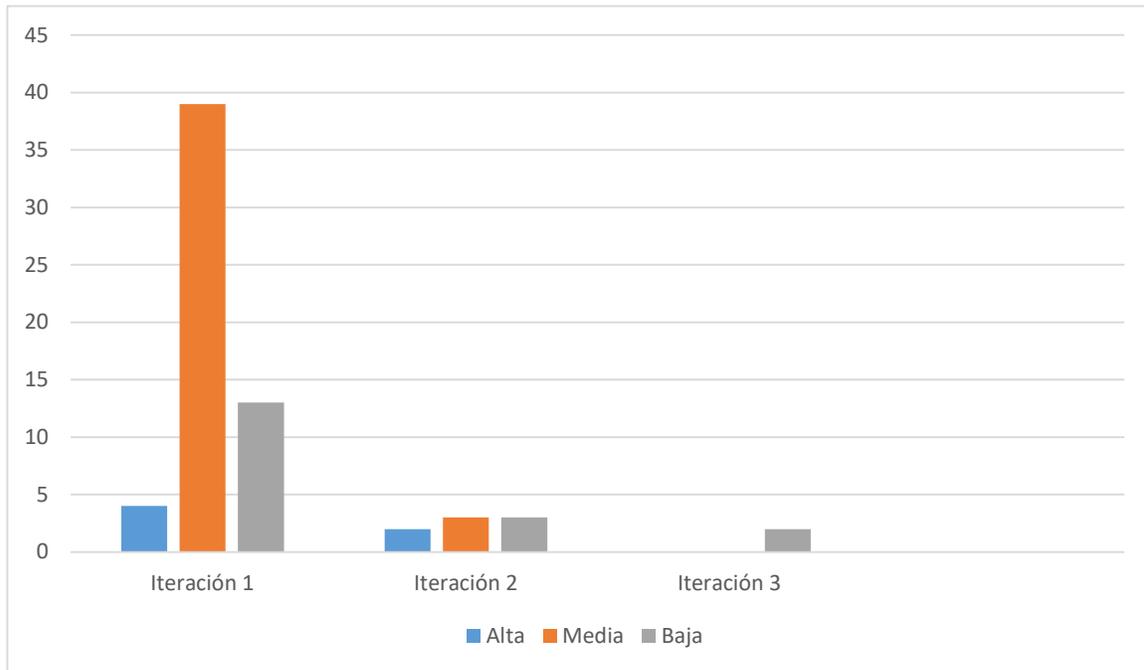


Fig. 6 No conformidades

Las no conformidades encontradas durante la tercera iteración fueron solucionadas durante la regresión. Dado que eran problemas de significación baja muy puntuales no fue requerido documentar una cuarta iteración.

### 3.5 Conclusiones parciales

Al concluir este capítulo se termina la realización de la aplicación, obteniendo los siguientes resultados:

- La elaboración del diagrama de componentes representó la vista de implementación del sistema y se obtuvo al finalizar la fase de implementación el producto final con las funcionalidades requeridas.
- El estándar de codificación seleccionado para el lenguaje de programación Java permitió organizar el código de la solución.
- La elaboración de los diseños de casos de pruebas permitió identificar los errores del sistema.
- Las pruebas de software aplicadas permitieron validar el correcto funcionamiento de la propuesta de solución.

### Conclusiones generales

Con la realización del presente trabajo se arribaron a las siguientes conclusiones:

- El desarrollo de la aplicación permite la consulta de la información no confidencial generada por el SIGIES.
- La utilización de REST como protocolo para intercambiar datos con la capa de servicios web y SQLite como sistema gestor de base de datos mejora la accesibilidad de los datos generados durante el proceso de ingreso y permite su consulta y análisis *offline* respectivamente.
- La realización de las pruebas de aceptación mediante la técnica de caja negra, permitieron detectar 56 no conformidades en la primera iteración, las cuales se fueron corrigiendo en las demás iteraciones, manifestando que la aplicación cumple con los requisitos de aceptación del cliente.

### **Recomendaciones**

Teniendo en cuenta los resultados obtenidos en el presente trabajo, se proponen las siguientes recomendaciones:

- Agregar un sistema de notificaciones.
- Agregar una funcionalidad que permita personalizar la aplicación.
- Agregar un apartado de preguntas frecuentes respondiendo las principales interrogantes del proceso de ingreso a la Educación Superior.

## Bibliografía

1. **Pisani, Francis.** La necesidad de una nueva cultura digital. [En línea] 22 de febrero de 2009. <http://archivo.eluniversal.com.mx/articulos/52629.html>.
2. **Pacheco, Yuceli Mejía.** *METODOLOGÍA DE SEGURIDAD PARA EL MANEJO DE DISPOSITIVOS MÓVILES Y LA VINCULACIÓN CON.* México : s.n., 2016.
3. **Soriano, Anaid Gevara.** *Dispositivos Móviles.* Ciudad de México : Universidad Nacional Autónoma de México, 2015.
4. **Universidad de Alicante.** Introducción a los Servicios Web. [En línea] 26 de 6 de 2014. [Citado el: 6 de 12 de 2017.] <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>.
5. **Morales, Yudy Castro.** Granma. *Informatización de la sociedad, plataforma para impulsar el desarrollo.* [En línea] 1 de 6 de 2018. <http://www.granma.cu/cuba/2018-06-01/informatizacion-de-la-sociedad-plataforma-para-impulsar-el-desarrollo-01-06-2018-23-06-12>.
6. **Stack Overflow.** [En línea] <https://es.stackoverflow.com/>.
7. **GitLab API.** [En línea] <https://docs.gitlab.com/ee/api/>.
8. **Moodle Mobile.** [En línea] [https://docs.moodle.org/all/es/Moodle\\_Mobile](https://docs.moodle.org/all/es/Moodle_Mobile).
9. **Humanos.** Humanos. [En línea] 12 de 6 de 2015. <http://humanos.uci.cu/2015/06/12/comparte-tu-software-repo-droid/>.
10. **Baz, Alonso Arturo, Artime, Irene Ferrerira y Rodriguez, María Alvarez.** *Dispositivos Móviles.* Oviedo : Universidad de Oviedo, 2014.
11. **Kristel Malave Polanco, José Luis Beauperthuy Taibo.** "ANDROID" EL SISTEMA OPERATIVO DE GOOGLE. Venezuela : Negotium, 2011. ISSN 1856-1810.
12. **Ortega, Luis.** ¿Qué es un archivo APK y para qué sirve? [En línea] 11 de 2017. [Citado el: 11 de 1 de 2018.] <https://www.androidpit.es/android-para-principiantes-apk>.
13. **W3C Consortium.** El W3C de la A a la Z. [En línea] 2017. [Citado el: 6 de 12 de 2017.] <https://www.w3c.es/Divulgacion/a-z/>.
14. **Microsoft.** Sincronizar Datos. [En línea] Microsoft, 7 de 3 de 2017. [Citado el: 5 de 12 de 2017.] <https://docs.microsoft.com/es-es/sql/relational-databases/replication/synchronize-data>.
15. **Alegsa, Leandro.** Diccionario de Informática y Tecnología. [En línea] Alegsa, 5 de 7 de 2016. [Citado el: 6 de 12 de 2017.] [http://www.alegsa.com.ar/Dic/sincronizar\\_datos.php](http://www.alegsa.com.ar/Dic/sincronizar_datos.php).

16. **4r Soluciones.** ¿Qué es un kit de desarrollo de software (SDK)? [En línea] 28 de 2 de 2013. [Citado el: 11 de 12 de 2017.] <http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/>.
17. **Falcón, Yuniel Suárez.** De programación. [En línea] 1 de 2 de 2016. [Citado el: 7 de 12 de 2017.] <http://deprogramacion.cubava.cu/2016/02/01/que-es-un-ide/>.
18. **Android Studio.** Android Studio. [En línea] <https://developer.android.com/studio/intro/index.html>.
19. **Ramírez, Iván.** Softonic. *¿Qué es Java?* [En línea] 10 de 4 de 2013. [Citado el: 7 de 12 de 2017.] <https://www.softonic.com/articulos/que-es-java>.
20. **Vyeira, Victor.** mindmeister. *Herramientas CASE.* [En línea] 9 de 7 de 2016. [Citado el: 8 de 12 de 2017.] <https://www.mindmeister.com/es/752475322/herramientas-case>.
21. **Visual Paradigm.** Visual Paradigm. [En línea] 2014. [Citado el: 12 de 12 de 2017.] <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/1s1y2/PracticaVP.pdf>.
22. **Balsamiq.** Balsamiq. [En línea] <https://docs.balsamiq.com/>.
23. **Maida, Esteban Grabiél y Pacienza, Julián.** *Metodologías de desarrollo de software.* Argentina : Universidad Católica Argentina, 2015.
24. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2014.
25. **Ramírez, Randy.** Modelo del Dominio. [En línea] 2 de 3 de 2015. [Citado el: 8 de 1 de 2018.] <https://prezi.com/ofzugydds2x9/modelo-del-dominio-modelo-de-negocio/?webgl=0>.
26. **Chaves, Michael Arias.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 7 de 2007. [Citado el: 15 de 12 de 2017.] <http://www.redalyc.org/html/666/66612870011>. SSN 1409-4746.
27. **Izaurrealde, María Paula.** *Caracterización de Especificación de Requerimientos de Entornos Ágiles: Historias de Usuario.* 2013.
28. **Alvarez, Miguel Angel.** Qué es MVC. [En línea] 2 de 1 de 2014. <https://desarrolloweb.com/articulos/que-es-mvc.html>.
29. **Acosta La o, Lennon y Lugo Marquez, Luis Manuel .** *Módulo para la creación de juegos del tipo observación en el software ATcnea.* 2017.
30. **Tedeschi, Nicolás.** ¿Qué es un Patrón de Diseño? [En línea] Microsoft, 2015. <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
31. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Mexico : s.n., 1999. 970-17-0261-1.

32. **Angel**. Patrones GOF. [En línea] 5 de 2011. <http://geektheplanet.net/5462/patrones-gof.xhtml>.
33. **Fonseca, Miguel Enrique Verdecia y Ortiz, Osvaldo Manuel González**. *Visor de reportes para móviles con sistema*. La Habana : s.n., 2013.
34. **Cabrera, Jorge Luis Arteaga y Tafur, Roviro Enrique Acuña**. *Desarrollo de una Aplicación Móvil y una Guía de Turismo para la Visualización y Descripción de los Sitios Turísticos del Centro de la Ciudad de Cartagena utilizando Realidad Aumentada*. Cartagena : s.n., 2014.
35. **Microsoft**. Diagramas de componentes de UML. [En línea] 2015. <https://msdn.microsoft.com/es-es/library/dd409390.aspx>.
36. **Merkury**. Estandares de Codificación. [En línea] 12 de 1 de 2017. <https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/>.
37. **UCI, Programa de Mejora Universidad de las Ciencias Informáticas, Basado en Convenciones de Código para el lenguaje de programación JAVA™ por Scott Hommel Sun Microsystems Inc. Traducción por Alberto Molpeceres. Estándar de Codificación para JAVA**. 2016.
38. **Pmoinformática**. Pruebas de aceptación de software según el ISTQB. [En línea] 8 de 2016. <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>.
39. **Quality**. Quality. *Innovando en Calidad y Servicio*. [En línea] 2016. [vyvquality.com](http://vyvquality.com).
40. **Pmoinformática**. [En línea] 8 de 8 de 2016. <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>.

## Anexos

## Anexo 1: Historias de usuario

Tabla 5 HU Autenticar usuario

|  |   |
|--|---|
|  |   |
| <b>Número:</b> 1   | <b>Nombre del requisito:</b> Autenticar Usuario |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era                 |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 2días                   |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 1 días                      |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir que el usuario se autentique para consultar la información que desee.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>El usuario se autenticará una vez que acceda a la aplicación.<br/>El sistema mostrará una vista para escanear el CI</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b><br/>La aplicación debe permitir que el usuario escanee su CI.<br/>Una vez escaneado correctamente el CI queda autenticado en la aplicación.<br/>Si el CI escaneado no es válido se mostrará nuevamente la posibilidad de realizar la acción.<br/>La aplicación debe validar, haciendo uso de los servicios del sistema, que el ci sea válido.</p> |   |
| <b>Observaciones:</b>  |   |
| <b>Prototipo de interfaz:</b>  |   |

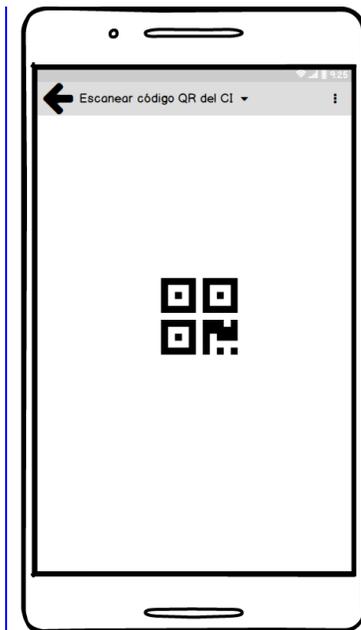


Tabla 6 HU Sincronizar datos con la capa de servicios web

|  |   |
|--|---|
|  |   |
| <b>Número:</b> 2   | <b>Nombre del requisito:</b> Sincronizar datos con la capa de servicios web |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era   |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 2días   |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 1 días  |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir que el usuario sincronice los datos de la aplicación con los de la capa de servicios web.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>El usuario puede o no estar autenticado dependiendo de los datos que desea sincronizar.<br/>El sistema brindará la posibilidad de sincronizar los datos deslizando hacia abajo en la pantalla del dispositivo.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b><br/>La aplicación debe permitir que el usuario sincronice los datos.<br/>Una vez sincronizados correctamente los datos el usuario podrá visualizarlos.</p> |   |

|   |
|---|
| Si no se pueden sincronizar correctamente los datos se mostrará un mensaje informando al usuario que ha ocurrido un error en la conexión. |
| <b>Observaciones:</b>   |
| <b>Prototipo de interfaz:</b>   |

**Tabla 7 HU Listar provincias**

|   |  |
|---|--|
|   |  |
| <b>Número:</b> 3  | <b>Nombre del requisito:</b> Listar provincias |
| <b>Programador:</b> Leandro Alejo Aguila  | <b>Iteración Asignada:</b> 1era                |
| <b>Prioridad:</b> Media   | <b>Tiempo Estimado:</b> 1día                   |
| <b>Riesgo en Desarrollo:</b> N/A  | <b>Tiempo Real:</b> 1día                       |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir listar las provincias teniendo en cuenta sus nombres.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>Para listar provincias debe existir al menos una provincia en el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b><br/>Cuando el usuario selecciona la acción listar plan de plazas. Se muestra al usuario un listado con los nombres de las provincias para que seleccione la provincia en cuestión de la desea ver el plan de plazas.</p> |  |
| <b>Observaciones:</b>   |  |
| <b>Prototipo de interfaz:</b>   |  |



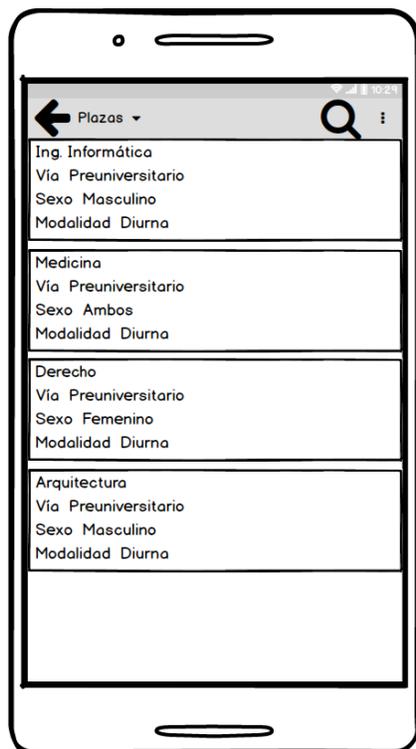
Tabla 8 HU Listar plan de plazas provincial

|   |   |
|---|---|
|   |   |
| <b>Número:</b> 4  | <b>Nombre del requisito:</b> Listar plan de plazas provincial |
| <b>Programador:</b> Leandro Alejo Aguila  | <b>Iteración Asignada:</b> 1era                               |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 3días                                 |
| <b>Riesgo en Desarrollo:</b> N/A  | <b>Tiempo Real:</b> 2 días                                    |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir consultar las carreras pertenecientes al plan de plazas de la provincia en cuestión.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>Para listar el plan de plazas de una provincia debe existir al menos un plan de plazas en el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> |   |

Cuando el usuario selecciona la acción listar plan de plazas de una provincia. Se muestra al usuario un listado con las plazas disponibles del plan de plazas seleccionado.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 9 HU Filtrar plan de plazas provincial**

|  |  |
|--|--|
|  |  |
| <b>Número:</b> 5   | <b>Nombre del requisito:</b> Filtrar plan de plazas provincial |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era                                |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días                                  |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días                                     |
| <b>Descripción:</b>  |  |
| <b>1- Objetivo:</b>  |  |
| Permitir filtrar el plan de plazas de una provincia teniendo en cuenta el nombre, vía de ingreso, sexo y |  |

modalidad de las plazas.

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para filtrar plan de plazas debe existir el plan plazas perteneciente a la provincia seleccionada en el sistema.

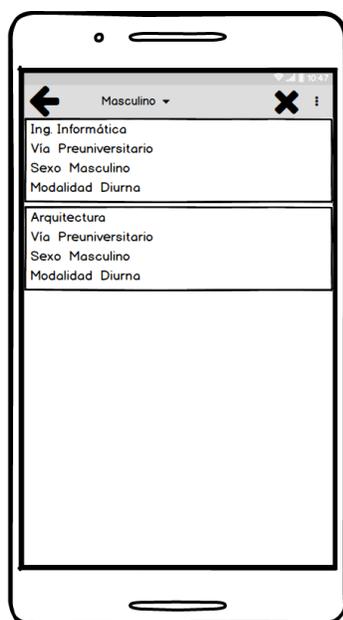
**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la opción buscar e inserta el criterio de búsqueda el sistema muestra un listado de los elementos plazas que cumplen con el criterio establecido.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 10 HU Mostrar datos de una carrera del plan de plazas provincial**

|  |  |  |  |
|--|--|--|--|
| <b>Número:</b> 6                         |  | <b>Nombre del requisito:</b> Mostrar datos de una carrera del plan de plazas provincial. |  |
| <b>Programador:</b> Leandro Alejo Aguila |  | <b>Iteración Asignada:</b> 1era  |  |
| <b>Prioridad:</b> Alta                   |  | <b>Tiempo Estimado:</b> 3días  |  |
| <b>Riesgo en Desarrollo:</b> N/A         |  | <b>Tiempo Real:</b> 2 días   |  |
| <b>Descripción:</b>                      |  |  |  |

**1- Objetivo:**

Permitir consultar los datos de una carrera del plan de plazas provincial tales como:

- Convocatoria
- Procesamiento
- Vía de ingreso
- Sexo
- Modalidad
- Centro de educación superior
- Plazas disponibles
- Plazas disponibles iniciales
- Demanda
- Ingresos requeridos
- Provincia

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para ver datos de una carrera del plan de plazas provincial debe existir al menos una carrera del plan de plazas provincial seleccionado en el sistema.

**3- Comportamientos válidos y no válidos (flujo central y alternos):****4- Flujo de la acción a realizar:**

Inicialmente se muestra al usuario un listado con las carreras del plan de plazas que han sido incluidas en el sistema. Una vez seleccionada una carrera, podrá ver sus datos.

**Observaciones:****Prototipo de interfaz:**

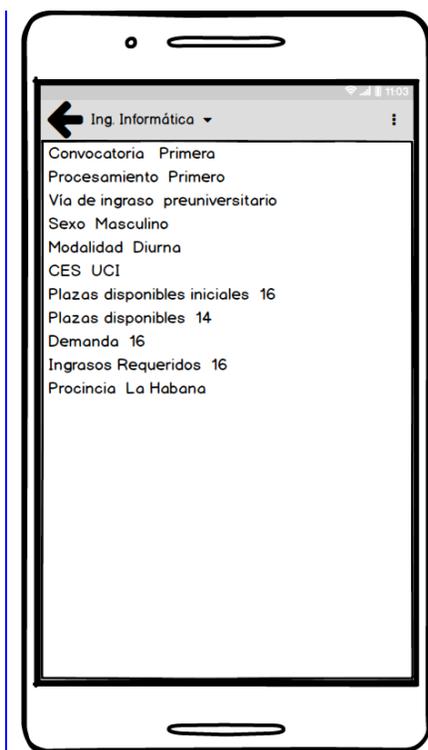


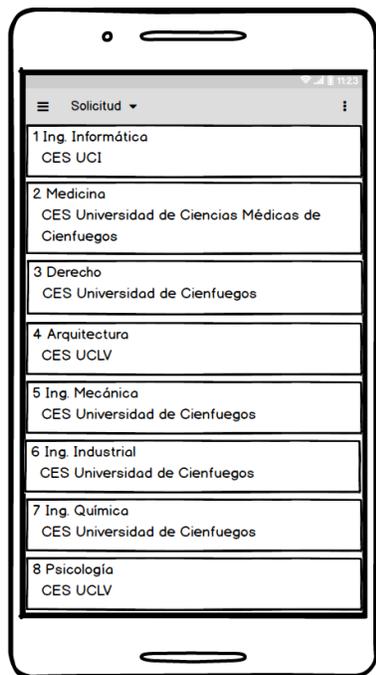
Tabla 11 HU Mostrar solicitud del estudiante

|  |   |
|--|---|
|  |   |
| <b>Número:</b> 7   | <b>Nombre del requisito:</b> Mostrar solicitud del estudiante |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era                               |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días                                 |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días                                    |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir consultar la solicitud de plazas hecha por el estudiante.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>Para ver la solicitud del estudiante hay que:<br/>-Estar autenticado en el sistema<br/>-Debe existir la solicitud del estudiante en el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b></p> |   |

Cuando el usuario selecciona la acción mostrar solicitud se muestra al usuario un listado con las solicitudes

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 12 HU Mostrar notas por cada asignatura examinada**

|  |  |
|--|--|
|  |  |
| <b>Número:</b> 8   | <b>Nombre del requisito:</b> Mostrar notas por cada asignatura examinada |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era  |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días  |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días   |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir consultar las notas por cada asignatura examinada.</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>Para verlas notas por cada asignatura examinada hay que:</p> <ul style="list-style-type: none"> <li>-Estar autenticado en el sistema</li> <li>-Debe existir al menos la nota de una asignatura en el sistema.</li> </ul> |  |

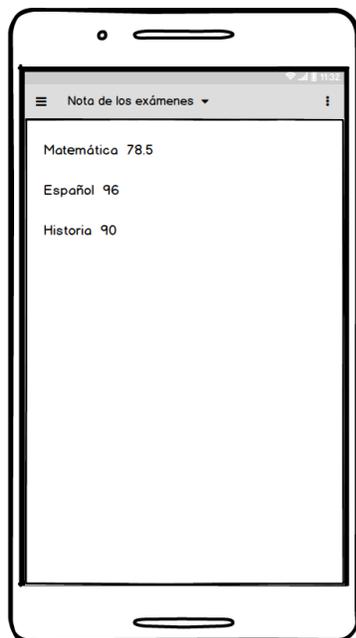
**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar notas por cada asignatura examinada se muestra al usuario un listado con las notas de las asignaturas correspondientes.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 13 HU Mostrar exámenes exonerados**

|  |  |  |  |
|--|--|--|--|
| <b>Número:</b> 9   |  | <b>Nombre del requisito:</b> Mostrar exámenes exonerados |  |
| <b>Programador:</b> Leandro Alejo Aguila                             |  | <b>Iteración Asignada:</b> 1era                          |  |
| <b>Prioridad:</b> Alta   |  | <b>Tiempo Estimado:</b> 3días                            |  |
| <b>Riesgo en Desarrollo:</b> N/A                                     |  | <b>Tiempo Real:</b> 2 días                               |  |
| <b>Descripción:</b>  |  |  |  |
| <b>1- Objetivo:</b>  |  |  |  |
| Permitir consultar los exámenes exonerados del estudiante.           |  |  |  |
| <b>2- Acciones para lograr el objetivo (precondiciones y datos):</b> |  |  |  |

Para ver los exámenes exonerados del estudiante hay que:

-Estar autenticado en el sistema

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar exámenes exonerados se muestra al usuario un listado con los exámenes exonerados.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 14 HU Mostrar datos de la vía de ingreso del estudiante**

|  |  |  |  |
|--|--|--|--|
| <b>Número:</b> 10  |  | <b>Nombre del requisito:</b> Mostrar datos de la vía de ingreso del estudiante |  |
| <b>Programador:</b> Leandro Alejo Aguila                                     |  | <b>Iteración Asignada:</b> 1era  |  |
| <b>Prioridad:</b> Alta   |  | <b>Tiempo Estimado:</b> 3días  |  |
| <b>Riesgo en Desarrollo:</b> N/A   |  | <b>Tiempo Real:</b> 2 días   |  |
| <b>Descripción:</b>  |  |  |  |
| <b>1- Objetivo:</b>  |  |  |  |
| Permitir consultar los datos de la vía de ingreso del estudiante tales como: |  |  |  |
| <ul style="list-style-type: none"> <li>▪ Nombre</li> </ul>                   |  |  |  |

- Nombre corto
- Exámenes
- Modalidad
- Tipo de escalafón
- Índice académico
- Tipo de asignación
- Aporta personal al SMA
- Provincia de asignación

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para ver los datos de la vía de ingreso del estudiante hay que:

-Estar autenticado en el sistema

-Deben existir los datos de la vía de ingreso del estudiante en el sistema.

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar datos de la vía de ingreso se muestra al usuario un listado con los datos de la vía de ingreso.

**Observaciones:**

**Prototipo de interfaz:**

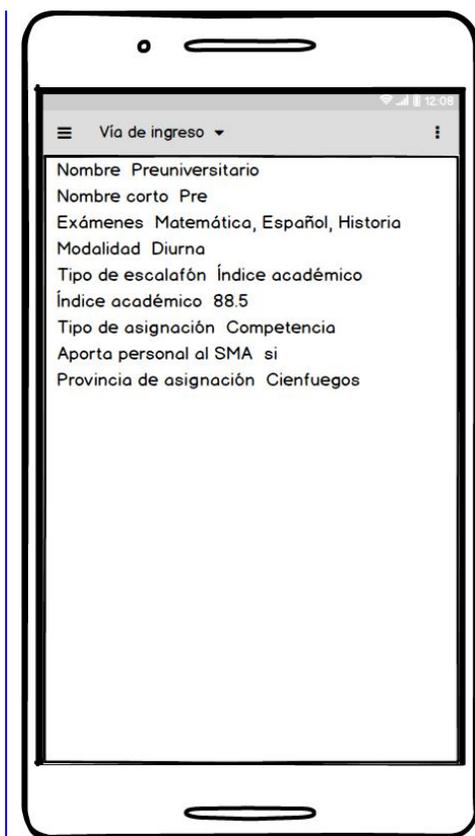


Tabla 15 HU Mostrar asistencia

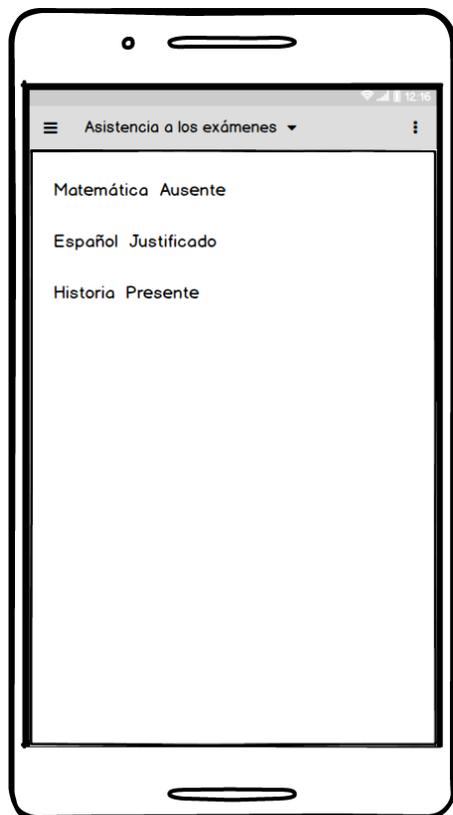
|  |   |
|--|---|
|  |   |
| <b>Número:</b> 12  | <b>Nombre del requisito:</b> Mostrar asistencia |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era                 |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días                   |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días                      |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>                 Permitir consultar la asistencia a los exámenes de ingreso:</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>                 Para ver los datos del estudiante hay que:<br/>                 -Estar autenticado en el sistema<br/>                 -Deben existir la asistencia a los exámenes en el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> |   |

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar asistencia se muestra al usuario un listado con la asistencia a los exámenes.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 16 HU Mostrar resultado de la asignación**

|  |   |
|--|---|
|  |   |
| <b>Número:</b> 14                        | <b>Nombre del requisito:</b> Mostrar resultado de la asignación |
| <b>Programador:</b> Leandro Alejo Aguila | <b>Iteración Asignada:</b> 1era                                 |
| <b>Prioridad:</b> Alta                   | <b>Tiempo Estimado:</b> 1día                                    |
| <b>Riesgo en Desarrollo:</b> N/A         | <b>Tiempo Real:</b> 1día  |
| <b>Descripción:</b>                      |   |
| <b>1- Objetivo:</b>                      |   |

Permitir consultar el resultado de la asignación.

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para ver el resultado de la asignación hay que:

- Estar autenticado en el sistema
- Deben existir el resultado de la asignación en el sistema.

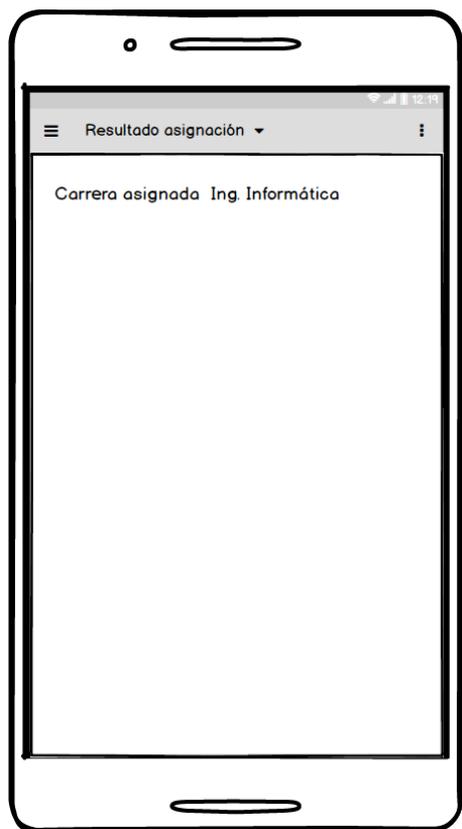
**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar resultado de la asignación se muestra al usuario el resultado de la asignación.

**Observaciones:**

**Prototipo de interfaz:**



**Tabla 17 HU Mostrar estado del servicio militar**

|            |  |
|------------|--|
|            |  |
| Número: 14 | Nombre del requisito: Mostrar estado de servicio militar |

|  |                                 |
|--|---------------------------------|
| <b>Programador:</b> Leandro Alejo Aguila | <b>Iteración Asignada:</b> 1era |
| <b>Prioridad:</b> Alta                   | <b>Tiempo Estimado:</b> 1día    |
| <b>Riesgo en Desarrollo:</b> N/A         | <b>Tiempo Real:</b> 1día        |

**Descripción:**

**1- Objetivo:**

Permitir consultar el estado de servicio militar.

**2- Acciones para lograr el objetivo (precondiciones y datos):**

Para ver el estado de servicio militar hay que:

- Estar autenticado en el sistema
- Deben existir el estado de servicio militar en el sistema.

**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar estado de servicio militar se muestra al usuario el estado de servicio militar.

**Observaciones:**

**Prototipo de interfaz:**

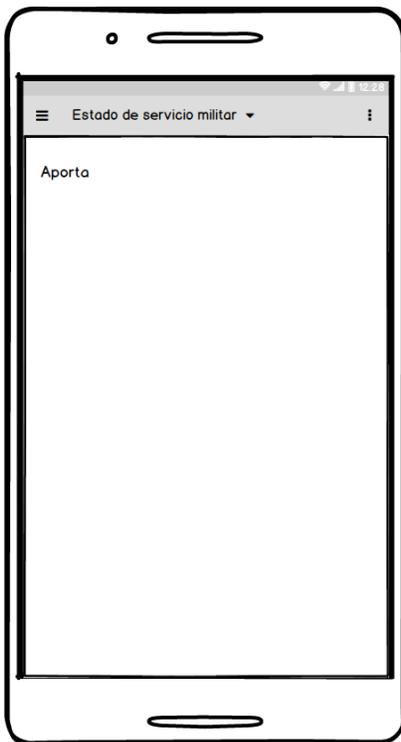


Tabla 18 Mostrar etapa actual del proceso de ingreso

|  |  |
|--|--|
|  |  |
| <b>Número:</b> 15  | <b>Nombre del requisito:</b> Mostrar etapa actual del proceso de ingreso |
| <b>Programador:</b> Leandro Alejo Aguila   | <b>Iteración Asignada:</b> 1era  |
| <b>Prioridad:</b> Alta   | <b>Tiempo Estimado:</b> 3días  |
| <b>Riesgo en Desarrollo:</b> N/A   | <b>Tiempo Real:</b> 2 días   |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b><br/>Permitir consultar el estado la etapa actual del proceso</p> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b><br/>Para ver la etapa actual del procesohay que:<br/>-Estar autenticado en el sistema<br/>-Deben existir la etapa actual del procesoen el sistema.</p> <p><b>3- Comportamientos válidos y no válidos (flujo central y alternos):</b></p> <p><b>4- Flujo de la acción a realizar:</b><br/>Cuando el usuario selecciona la acción mostrar etapa actual del proceso se muestra al usuario la etapa actual del proceso.</p> |  |
| <b>Observaciones:</b>  |  |
| <b>Prototipo de interfaz:</b>  |  |

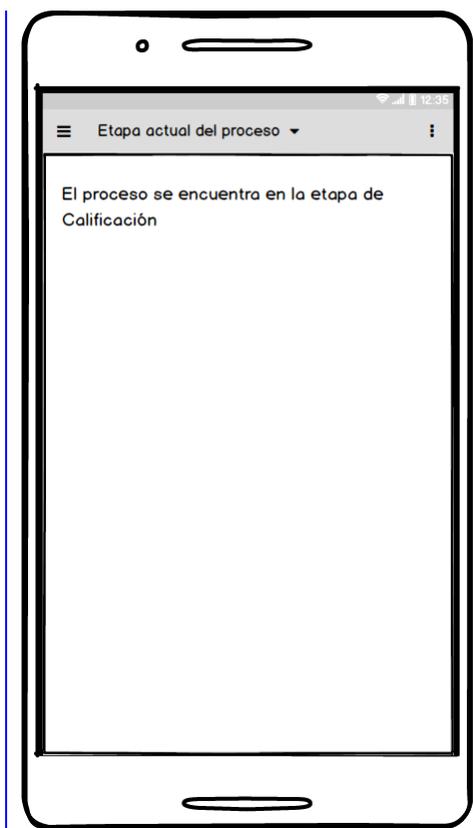


Tabla 19 HU Mostrar actas de exámenes

|   |  |
|---|--|
|   |  |
| <b>Número:</b> 16   | <b>Nombre del requisito:</b> Mostrar actas de exámenes |
| <b>Programador:</b> Leandro Alejo Aguila  | <b>Iteración Asignada:</b> 1era                        |
| <b>Prioridad:</b> Alta  | <b>Tiempo Estimado:</b> 3días                          |
| <b>Riesgo en Desarrollo:</b> N/A  | <b>Tiempo Real:</b> 2 días                             |
| <p><b>Descripción:</b></p> <p><b>1- Objetivo:</b></p> <p>Permitir consultar los datos de las actas de los exámenes tales como:</p> <ul style="list-style-type: none"> <li>▪ Asignatura</li> <li>▪ Numero de acta</li> <li>▪ Preuniversitario</li> </ul> <p><b>2- Acciones para lograr el objetivo (precondiciones y datos):</b></p> |  |

Para ver los datos de las actas de exámenes hay que:

- Estar autenticado en el sistema
- Deben existir los datos los datos de las actas de exámenes en el sistema.

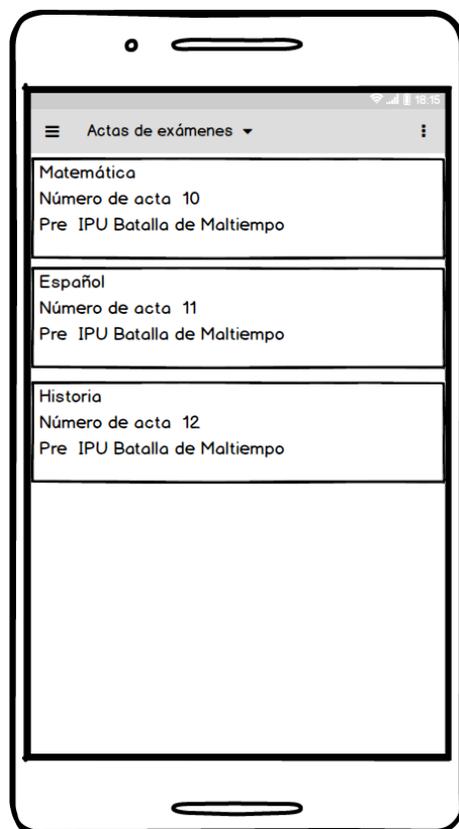
**3- Comportamientos válidos y no válidos (flujo central y alternos):**

**4- Flujo de la acción a realizar:**

Cuando el usuario selecciona la acción mostrar datos de las actas de los exámenes se muestra al usuario un listado con los datos de las actas de los exámenes.

**Observaciones:**

**Prototipo de interfaz:**



**Anexo 2: Casos de prueba**

**Tabla 20 Caso de prueba 1: Autenticar usuario**

|  |
|--|
| <b>Descripción general:</b> permitir autenticarse en la aplicación |
| <b>Condiciones de ejecución:</b>                                   |

| <ul style="list-style-type: none"> <li>▪ El usuario debe contar con un dispositivo móvil con el APK instalado</li> <li>▪ El CI debe ser válido</li> <li>▪ Debe estar conectado a una red inalámbrica.</li> </ul> |   |  |  |
|--|---|--|--|
| Escenario  | Descripción   | Respuesta del sistema  | Flujo central  |
| <b>EC 1.1</b><br>Autenticarse en la aplicación   | El usuario escanea un CI haciendo uso del APK                           | La aplicación valida el CI y le da acceso a la información requerida   | Página Principal/Menú/Información personal X/Escanear CI a través del código QR                                  |
| <b>EC 1.2</b> Datos incorrectos  | El usuario escanea un CI no válido                                      | La aplicación valida el CI y muestra el mensaje de error "El código escaneado no contiene los datos necesarios para acceder a la información solicitada." y brinda la opción Aceptar | Página Principal/Menú/Información personal X/Escanear CI a través del código QR/Datos incorrectos                |
| <b>EC 1.3</b><br>Opción Aceptar  | El usuario selecciona la opción Aceptar presente en el mensaje de error | Regresa al <b>EC 1.1</b>   | Página Principal/Menú/Información personal X/Escanear CI a través del código QR/Datos incorrectos/Opción Aceptar |
| <b>EC 1.4</b><br>Opción Cambiar usuario  | El usuario selecciona la opción Cambiar usuario                         | Regresa al <b>EC 1.1</b>   | Página Principal/Icono tres puntos verticales/Cambiar usuario  |

**Tabla 21 Caso de prueba 2: Sincronizar datos con la capa de servicios web**

|   |
|---|
| <b>Descripción general:</b> permitir sincronizar los datos con la capa de servicios web.  |
| <b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>▪ Contar con un dispositivo móvil con el APK instalado</li> </ul> |

| <ul style="list-style-type: none"> <li>▪ Estar conectado a una red inalámbrica</li> <li>▪ Los datos a sincronizar deben estar disponibles.</li> </ul> |   |   |                              |
|---|---|---|------------------------------|
| Escenario   | Descripción   | Respuesta del sistema                             | Flujo central                |
| <b>EC 1.1</b><br>Sincronizar los datos con la capa de servicios web   | El usuario desliza hacia abajo en la pantalla del dispositivo | El sistema muestra los datos sincronizados        | Vista X/Deslizar hacia abajo |
| <b>EC 1.1 Error</b><br>al sincronizar los datos con la capa de servicios web  | El usuario desliza hacia abajo en la pantalla del dispositivo | El sistema muestra el mensaje "Error de conexión" |                              |

**Tabla 22 Caso de prueba 3: Listar provincias**

| <b>Descripción general:</b> permitir listar las provincias                            |  |   |  |
|---|--|---|--|
| <b>Condiciones de ejecución:</b> Contar con un dispositivo móvil con el APK instalado |  |   |  |
| Escenario   | Descripción                                      | Respuesta del sistema   | Flujo central                                    |
| <b>EC 1.1</b><br>Listar provincias  | El usuario selecciona la opción plan de plazas   | El sistema muestra un listado de provincias y el municipio especial | Página Principal/Menú/Plan de plazas             |
| <b>EC 1.2</b><br>Listar plan de plazas por provincia                                  | El usuario selecciona el nombre de una provincia | FORTES_SIGIES_APK_DCP_Listar plan de plazas provincial.ods          | Página Principal/Menú/Plan de Plazas/Provincia X |

**Tabla 23 Caso de prueba 4: Listar plan de plazas provincial**

| <b>Descripción general:</b> permitir listar las el plan de plazas provincial                          |   |   |  |
|---|---|---|--|
| <b>Condiciones de ejecución:</b> El plan de plazas de la provincia seleccionada debe estar disponible |   |   |  |
| Escenario   | Descripción   | Respuesta del sistema   | Flujo central  |
| <b>EC 1.1</b><br>Listar el plan de plazas provincial  | El usuario selecciona una provincia                             | El sistema muestra un listado con las plazas disponibles de la provincia seleccionada | Página principal/<br>Menú/Plan de Plazas/Provincias                                |
| <b>EC 1.2</b><br>Mostrar datos de una carrera del plan de plazas provincial                           | El usuario selecciona una carrera del plan de plazas provincial | FORTES_SIGIES_APK_DCP_Mostrar datos de una carrera del plan de plazas provincial.ods  | Página principal/<br>Menú/Plan de Plazas/Provincia X/Plazas Provinciales/Carrera X |

**Tabla 24 Caso de prueba 5: Filtrar plan de plazas provincial**

| <b>Descripción general:</b> permitir filtrar las el plan de plazas provincial |  |   |   |
|---|--|---|---|
| <b>Condiciones de ejecución:</b>  |  |   |   |
| Escenario   | Descripción  | Respuesta del sistema   | Flujo central   |
| <b>EC 1.1</b><br>Filtrar el plan de plazas provincial                         | El usuario selecciona la opción buscar e introduce un criterio de búsqueda | El sistema muestra un listado con las plazas disponibles que corresponden con el criterio de búsqueda introducido | Página principal/ Menú/Plan de Plazas/Provincias/Plazas |
| <b>EC 1.2</b><br>Eliminar filtro  | El usuario selecciona el   | El sistema interrumpe el filtrado. Regresa al <b>EC 1.1</b>   | Página principal/ Menú/Plan de Plazas/Provincia         |

|  |                       |  |   |
|--|-----------------------|--|---|
|  | icono cancelar<br>(X) |  | X/Plazas Provinciales/<br>Icono lupa/Icono Cancelar |
|--|-----------------------|--|---|

**Tabla 25 Caso de prueba 6: Mostrar datos de una carrera del plan de plazas provincial**

| <b>Descripción general:</b> permitir mostrar datos de una carrera del plan de plazas provincial |   |  |   |
|---|---|--|---|
| <b>Condiciones de ejecución:</b>  |   |  |   |
| Escenario   | Descripción   | Respuesta del sistema  | Flujo central   |
| <b>EC 1.1</b><br>Mostrar datos de una carrera del plan de plazas provincial                     | El usuario selecciona una carrera del plan de plazas provincial | El sistema muestra un listado con los datos de la plaza seleccionada tales como: <ul style="list-style-type: none"> <li>▪ Convocatoria</li> <li>▪ Procesamiento</li> <li>▪ Vía de ingreso</li> <li>▪ Sexo</li> <li>▪ Modalidad</li> <li>▪ Centro de educación superior</li> <li>▪ Plazas disponibles</li> <li>▪ Plazas disponibles iniciales</li> <li>▪ Demanda</li> <li>▪ Ingresos requeridos</li> <li>▪ Provincia</li> </ul> | Página principal/ Menú/Plan de Plazas/Provincia<br>X/Plazas provinciales/ Plaza X |

**Tabla 26 Caso de prueba 7: Mostrar solicitud del estudiante**

| <b>Descripción general:</b> permitir mostrar la solicitud del estudiante  |             |                       |               |
|---|-------------|-----------------------|---------------|
| <b>Condiciones de ejecución:</b>  |             |                       |               |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Debe existir la solicitud en la base de datos</li> </ul> |             |                       |               |
| Escenario   | Descripción | Respuesta del sistema | Flujo central |

|   |  |  |                        |
|---|--|--|------------------------|
| <b>EC 1.1</b><br>Mostrar solicitud del estudiante | El usuario selecciona la opción plazas solicitadas | El sistema muestra un listado con las plazas solicitadas | Página principal/ Menú |
|---|--|--|------------------------|

**Tabla 27 Caso de prueba 8: Mostrar notas por cada asignatura examinada**

|   |  |  |   |
|---|--|--|---|
| <b>Descripción general:</b> permitir mostrar notas por cada asignatura examinada  |  |  |   |
| <b>Condiciones de ejecución:</b>  |  |  |   |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir las notas en la base de datos</li> </ul> |  |  |   |
| <b>Escenario</b>  | <b>Descripción</b>                             | <b>Respuesta del sistema</b>                               | <b>Flujo central</b>                      |
| <b>EC 1.1</b><br>Mostrar notas por cada asignatura examinada  | El usuario selecciona la opción nota de examen | El sistema muestra las notas de las asignaturas examinadas | Página principal/ Menú/Plazas solicitadas |

**Tabla 28 Caso de prueba 9: Mostrar si está o no exonerado de algún examen**

|  |   |   |                                  |
|--|---|---|----------------------------------|
| <b>Descripción general:</b> permitir ver si el estudiante está o no exonerado de algún examen  |   |   |                                  |
| <b>Condiciones de ejecución:</b>   |   |   |                                  |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir los datos correspondientes en la base de datos</li> </ul> |   |   |                                  |
| <b>Escenario</b>   | <b>Descripción</b>                        | <b>Respuesta del sistema</b>                  | <b>Flujo central</b>             |
| <b>EC 1.1</b><br>Mostrar si el estudiante está o no exonerado de algún examen  | El usuario selecciona la opción exonerado | El sistema muestra las asignaturas exoneradas | Página principal/ Menú/Exonerado |

**Tabla 29 Caso de prueba 10: Mostrar datos de la vía de ingreso del estudiante**

| <b>Descripción general:</b> permitir mostrar datos de la vía de ingreso del estudiante  |  |   |  |
|---|--|---|--|
| <b>Condiciones de ejecución:</b>  |  |   |  |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir los datos de la vía de ingreso del estudiante en la base de datos</li> </ul> |  |   |  |
| <b>Escenario</b>  | <b>Descripción</b>                             | <b>Respuesta del sistema</b>  | <b>Flujo central</b>                                 |
| <b>EC 1.1</b><br>Mostrar datos de la vía de ingreso del estudiante  | El usuario selecciona la opción vía de ingreso | El sistema muestra los datos de la vía de ingreso tales como: <ul style="list-style-type: none"> <li>▪ Nombre</li> <li>▪ Nombre corto</li> <li>▪ Exámenes</li> <li>▪ Modalidad</li> <li>▪ Tipo de escalafón</li> <li>▪ Índice académico</li> <li>▪ Tipo de asignación</li> <li>▪ Aporta personal al SMA</li> <li>▪ Provincia de asignación</li> </ul> | Página principal/<br>Menú/Datos de la vía de ingreso |

**Tabla 30 Caso de prueba 11: Mostrar datos del estudiante**

| <b>Descripción general:</b> permitir mostrar datos del estudiante  |                          |   |  |
|--|--------------------------|---|--|
| <b>Condiciones de ejecución:</b>   |                          |   |  |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir los datos del estudiante en la base de datos</li> </ul> |                          |   |  |
| <b>Escenario</b>   | <b>Descripción</b>       | <b>Respuesta del sistema</b>                                  | <b>Flujo central</b>                           |
| <b>EC 1.1</b><br>Mostrar   | El usuario selecciona la | El sistema muestra los datos de la vía de ingreso tales como: | Página principal/<br>Menú/Datos del estudiante |

|                      |                             |   |  |
|----------------------|-----------------------------|---|--|
| datos del estudiante | opción datos del estudiante | <ul style="list-style-type: none"> <li>▪ CI</li> <li>▪ Nombre(s)</li> <li>▪ Apellidos</li> <li>▪ Dirección Particular</li> <li>▪ Provincia de residencia</li> <li>▪ Municipio de residencia</li> <li>▪ Color de piel</li> <li>▪ Sexo</li> <li>▪ Índice académico</li> <li>▪ Preuniversitario</li> <li>▪ Vía de ingreso</li> <li>▪ Procedencia</li> <li>▪ Situación</li> </ul> |  |
|----------------------|-----------------------------|---|--|

**Tabla 31 Caso de prueba 12: Mostrar asistencia**

|  |  |   |                                      |
|--|--|---|--------------------------------------|
| <b>Descripción general:</b> permitir mostrar la asistencia del estudiante a os exámenes  |  |   |                                      |
| <b>Condiciones de ejecución:</b>   |  |   |                                      |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir los datos correspondientes en la base de datos</li> </ul> |  |   |                                      |
| <b>Escenario</b>   | <b>Descripción</b>                         | <b>Respuesta del sistema</b>  | <b>Flujo central</b>                 |
| EC 1.1<br>Mostrar asistencia a los exámenes  | El usuario selecciona la opción asistencia | El sistema muestra la asistencia del estudiante a los exámenes de ingreso | Página principal/<br>Menú/Asistencia |

**Tabla 32 Caso de prueba 13: Mostrar resultado de la asignación**

|  |  |  |  |
|--|--|--|--|
| <b>Descripción general:</b> permitir mostrar resultado de la asignación  |  |  |  |
| <b>Condiciones de ejecución:</b>   |  |  |  |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Debe existir el resultado de la asignación en la base de datos</li> </ul> |  |  |  |

| Escenario   | Descripción  | Respuesta del sistema                            | Flujo central  |
|---|--|--|--|
| <b>EC 1.1</b><br>Mostrar resultado de la asignación | El usuario selecciona la opción resultado de la asignación | El sistema muestra el resultado de la asignación | Página principal/<br>Menú/Resultado de la asignación |

**Tabla 33 Caso de prueba 14: Mostrar estado del servicio militar**

| <b>Descripción general:</b> permitir mostrar el estado del servicio militar   |   |   |   |
|---|---|---|---|
| <b>Condiciones de ejecución:</b>  |   |   |   |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Debe existir el estado del servicio militar en la base de datos</li> </ul> |   |   |   |
| Escenario   | Descripción   | Respuesta del sistema                             | Flujo central   |
| <b>EC 1.1</b><br>Mostrar estado del servicio militar  | El usuario selecciona la opción estado del servicio militar | El sistema muestra el estado del servicio militar | Página principal/<br>Menú/Estado del servicio militar |

**Tabla 34 Caso de prueba 15: Mostrar etapa actual del proceso**

| <b>Descripción general:</b> permitir mostrar etapa actual del proceso de ingreso  |  |   |  |
|---|--|---|--|
| <b>Condiciones de ejecución:</b>  |  |   |  |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Debe existir el estado del servicio militar en la base de datos</li> </ul> |  |   |  |
| Escenario   | Descripción  | Respuesta del sistema                       | Flujo central                                      |
| <b>EC 1.1</b><br>Mostrar la etapa actual del proceso de ingreso   | El usuario selecciona la opción etapa actual del proceso | El sistema muestra etapa actual del proceso | Página principal/<br>Menú/Etapa actual del proceso |

**Tabla 35 Caso de prueba 16: Mostrar actas de examen**

| <b>Descripción general:</b> permitir mostrar actas de examen  |   |   |   |
|---|---|---|---|
| <b>Condiciones de ejecución:</b>  |   |   |   |
| <ul style="list-style-type: none"> <li>▪ El usuario debe estar autenticado en la aplicación</li> <li>▪ Deben existir las actas de examen en la base de datos</li> </ul> |   |   |   |
| <b>Escenario</b>  | <b>Descripción</b>                              | <b>Respuesta del sistema</b>  | <b>Flujo central</b>                        |
| <b>EC 1.1</b><br>Mostrar actas de examen  | El usuario selecciona la opción actas de examen | El sistema muestra las actas de examen. Se mostrarán datos tales como: <ul style="list-style-type: none"> <li>▪ Asignatura</li> <li>▪ Numero de acta</li> <li>▪ Preuniversitario</li> </ul> | Página principal/<br>Menú/Actas de exámenes |