



Universidad de las Ciencias Informáticas

Facultad 1

# **Desarrollo de una aplicación móvil para el Directorio de Personas en la Universidad de las Ciencias Informáticas**

Tesis para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor:** Armando Lourenço da Silva e Costa

**Tutores:** Lic. Antonio Rey Roque

Ing. Alejandro Arias Naranjo

La Habana, 2020

## Declaración de autoría

Declaro por este medio que yo **Armando Lourenço da Silva e Costa**, con carné de identidad **95061400406** soy el autor principal del trabajo titulado “**Desarrollo de una aplicación móvil para el Directorio de Personas en la Universidad de las Ciencias Informáticas**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ de \_\_\_\_\_

Autor:

---

Armando Lourenço da Silva e Costa

Tutores:

---

Lic. Antonio Rey Roque

---

Ing. Alejandro Arias Naranjo

## **Agradecimientos**

Agradecer en primera instancia a Nzambi por darme la vida. Agradecer a mis padres Jonas y Francisca por la educación y la preocupación que han tenido por mis estudios. Agradecer a mis amadas hermanas Cândida y Lukenia por la fuerza, por el cuidado que siempre me brindaran. Agradecer a mis tios Mateus Costa por la dirección y la inspiración profesional, en su silencio me felicitaba y me amonestaba, a mi tío Paito por enseñar la base de la matemática. Agradecer a mis profesores de las enseñanzas precedentes y a mi profesor del curso de Inglés “Teacher Puff”. Agradecer a mis compañeros de “batalla” que de forma significativa me ayudarán a desarrollarme como persona. Agradecer a mis tutores por la paciencia y la orientación que me han dado especialmente al profesor Tony, a mi padre cubano Joel que siempre estuvo allí para alertarme de mis objetivos. Agradecer a mis amistades.

Agradecer a la Universidad de las Ciencias Informáticas por hacer de mi estadia em Cuba, la mejor posible.

## **Dedicatoria**

Dedico a mis padres y hermanas.

## Epígrafe

“Sí se puede”

## Resumen

Actualmente se hace más uso de las aplicaciones móviles que de navegadores de internet, por eso las descargas de aplicaciones móviles ha aumentado de forma exponencial en las tiendas de aplicaciones. La Universidad de las Ciencias Informáticas (UCI) cuenta hoy con un directorio que funciona como un catálogo para facilitar la búsqueda de información de los ciudadanos de la UCI, aprovechando las fortalezas tecnológicas que dispone la UCI, se ha creado una aplicación móvil para dicho directorio. El desarrollo de la solución está guiado por la metodología de desarrollo Scrum. Para su elaboración se usan herramientas y tecnologías como: Visual Paradigm, Android Studio, Java. Se aplicaron pruebas funcionales y de aceptación, evidenciándose que existe una correspondencia entre el objetivo y los resultados obtenidos, las pruebas permiten garantizar que los requerimientos de la aplicación sean cumplidos. En el presente trabajo se hace una presentación de todo el análisis e especificaciones del problema propuesto.

**Palabras claves:** Aplicación Móvil, Android, Sistemas de Búsquedas, Directorio.

## **Abstract**

Actually people use more mobile applications than web navigator, reason why the downloads of apps has been increased in the appstores. Google recognize the web sites that are developed following rules named by mobile first. La Universidad de las Ciencias Informáticas (UCI) has today a directory that Works as a catalogue to facilitate the searching of information from UCI citizens, taking advantage of technologic strength available in UCI, was built a mobile app for directory. The development of the project is guide by Scrum methodology. For the elaboration is used tools and technology like: Visual Paradigm, Android Studio, Java. Was applied functional and acceptance testing, making evident that exist a conformity between the objective and the obtained results, the testing allows to certify that the requirements of the application are executed. In this work it is showing all the analysis and specification of the proposed problem.

**Keywords:** Mobile application, Android, Searching System, Directory.

## Lista de tablas

Tabla 1- Las principales tecnologías para desarrollo de aplicaciones nativas .....	21
Tabla 2 – Ventajas y desventajas de las categorías de desarrollo .....	22
Tabla 3 – Product BackLog con los requisitos funcionales.....	33
Tabla 4 – Product BackLog con los requisitos no funcionales .....	34
Tabla 5 – Historia de Usuario “Buscar Persona” .....	35
Tabla 6 – Historia de Usuario “Agregar Favorito” .....	<b>¡Error! Marcador no definido.</b>

## Lista de figuras

Figura 1: Modelo de dominio.....	31
Figura 2: Mapa de Navegación de la aplicación.....	38
Figura 3: Funcionamiento del patrón modelo-vista-controlador.....	39
Figura 4: Modelo físico de datos .....	41
Figura 5- Diagrama de despliegue del sistema .....	43
Figura 6- Grafo resultante “Técnica Camino Básico”.....	45

# Índice

Resumen .....	vi
Lista De Tablas.....	8
Lista De Figuras.....	8
Introducción .....	12
Capítulo I – Fundamentación Teórica .....	16
1. Introducción .....	16
1.2. Sistema De Búsqueda De Información.....	16
1.1.1. Directorio .....	17
1.3. Estudio De Homólogos.....	17
1.4. Aplicaciones Móviles .....	20
1.4.1. Aplicaciones Nativas.....	20
1.4.2. Web Apps.....	21
1.4.3. Entorno De Desarrollo Multiplataforma .....	21
1.5. Servicios Web .....	22
1.6. Ciclo De Desarrollo De Software .....	23
1.7. Metodología De Desarrollo De Software .....	23
1.7.1. Selección De La Metodología A Utilizar .....	24
1.8. Herramientas Y Tecnologías A Utilizar .....	26
1.8.1. Sistema Operativo Android .....	26
1.8.2. Android Studio V3.6.....	27

1.8.3.	Lenguajes De Programación.....	27
1.8.4.	Lenguaje Unificado De Modelado Uml.....	28
1.8.5.	Sqlite .....	28
1.8.6.	Visual Paradigm.....	28
1.8.7.	Pencil.....	28
	Capitulo Ii - Propuesta De Solución.....	30
2.1.	Introducción.....	30
2.2.	Modelo De Dominio.....	30
2.3.	Propuesta De Solución.....	32
2.4.	Producto Backlog .....	32
2.5.	Historia De Usuario .....	35
2.6.	Mapa De Navegación.....	37
2.7.	Estilo Arquitectónico Cliente-Servidor.....	38
2.8.	Patrón De Arquitectura .....	38
2.9.	Patrones De Diseño .....	40
2.9.1.	Patrones De Asignación De Responsabilidad (Grasp).....	40
2.10.	Modelo Datos (Bd).....	41
2.11.	Conclusiones Parciales.....	42
	Capitulo Iii - Implementación Y Pruebas .....	43
3.1.	Introducción.....	43
3.2.	Diagrama De Despliegue .....	43
3.3.	Pruebas De Software .....	44
3.3.1.	Métodos De Pruebas .....	44

3.3.2.	Pruebas De Aceptación .....	44
3.3.3.	Pruebas Unitaria .....	44
3.3.4.	Pruebas Funcionales .....	46
3.3.4.1.	Diseño De Casos De Pruebas.....	46
3.3.5.	Descripción De Los Estándares De Codificación Utilizados .....	49
3.4.	Conclusiones Parciales .....	50
	Conclusiones Generales.....	51
	Recomendaciones .....	52
	Referencias Bibliográficas.....	53
	Glosario De Términos .....	58
	Anexos.....	59

## Introducción

En la actualidad la telefonía celular se ha convertido en una de las tecnologías más usadas a diario por millones de personas alrededor de mundo. Casi treinta años de evolución constante en las comunicaciones han permitido que los dispositivos móviles, según FUNIBER se conviertan en una parte fundamental para el desarrollo cotidiano de cualquier actividad emprendida por el individuo, convirtiéndolo en el dispositivo electrónico más usado en el mundo.

Las personas cada vez más están utilizando teléfono inteligente, ya que gracias a las tareas que son capaces de realizar y a la cantidad de funciones que incorpora, puede mejorar el estilo de vida de los usuarios. Seguramente los teléfonos celulares han pasado a ser uno de los elementos imprescindibles para la vida social de cualquier persona. Según Deloitte los celulares, se usan un promedio 47 veces al día y los cubanos no están muy lejos de estas estadísticas, según informa CubaDebate, ETECSA cerró el 2017 con 4,22 millones de líneas móviles activas.

Para el desarrollo de la aplicación propuesta se han observado las pautas del Programa Rector de la Informatización de la sociedad cubana, estas establecen una guía base para el desarrollo de las TICs, estableciendo pautas y prioridades con el objetivo de garantizar: La soberanía tecnológica, La informatización en función de desarrollar y modernizar coherentemente todas las esferas de la sociedad y apoyo a las prioridades del país. La utilización y el acceso masivo de los ciudadanos a los servicios soportados sobre las TICs. En las instituciones se hace necesario el constante uso de la comunicación e información para la coordinación de actividades entre los individuos, la toma de decisiones y la delegación de funciones. El uso de los sistemas de búsqueda en las organizaciones, permiten la fácil localización y comunicación del personal que las conforman (Cruz y Carrasco, 2013).

A medida que la información es más creciente e ilimitada, resulta necesario para las instituciones desarrollar mecanismos que permitan localizarla, manipularla y tratarla. Para ello el uso de las TICs han permitido el desarrollo de sistemas automatizados, con búsquedas especializadas que permiten encontrar datos específicos de un tema determinado a gran velocidad. Búsquedas que se pueden iniciar por medio de palabras clave que identifican de una forma u otra el tema de interés de las personas. Dentro de las vías que se pueden utilizar para realizar este proceso se encuentran los directorios (Justiz y Peñalver,2016).

Los directorios son guías que permiten localizar personas, organismos y entidades públicas o privadas; se puede definir como una lista de personas o de organizaciones ordenadas alfabética o sistemáticamente, proporcionando direcciones, cargos, funciones y datos similares, de tal manera que permite al usuario localizar, identificar y obtener informaciones acerca de las personas u organismos existentes en una industria, mercado, grupo, sector de actividad o disciplina determinada (Arvai, Halácsy Y Adam, 2016).

La Universidad de las Ciencias Informáticas es un campus con un alto grado de informatización de referencia en el contexto latinoamericano. En la UCI cuando se necesita obtener información de estudiantes o trabajadores, uno de los sistemas que más se utiliza es el Directorio de Personas, que permite consultar la información pública de una persona en la universidad. Según una encuesta realizada a una muestra de 155 personas de la universidad y una entrevista al Director de Redes y Servicios Telemáticos, se obtuvo como resultado que uno de los servicios más solicitados en la institución es el Directorio UCI (Justiz y Peñalver, 2016). Sin embargo, en ocasiones no satisface las necesidades del cliente, pues carece de diferentes funciones que son de interés para el usuario. Por ejemplo: no puede ser accedido sin un navegador web, no cuenta con diseño adaptativo tornando la visibilidad del contenido restringida afectando así su usabilidad, no permite al usuario crear sus propias listas de personas. El directorio publicado bajo la dirección <https://directorio.uci.cu> está en funcionamiento desde el año 2011, fue desarrollado en el CMS Drupal en su versión 3, evidenciando un gran atraso en cuanto a tecnología de desarrollo esto hace que sea prácticamente imposible de actualizar a versiones mas recientes del CMS y provocando un alto grado de complejidad el proceso de mantenimiento.

La UCI cuenta con varios espacios donde existe la posibilidad de conectarse a una WI-FI lo cual permite a los usuarios establecer una conexión con la red de la universidad y consumir los servicios que esta brinda. Además existen personas que por sus funciones de trabajo cuenta con dispositivos móviles con un punto de acceso lo que le permite desde cualquier ubicación tener acceso a la red de la universidad, por lo que desarrollar una aplicación para dispositivos móviles que permita la obtención de información de las personas permitirá mayor usabilidad del Directorio de Personas de la UCI.

Para dar solución al problema anteriormente descrito se plantea el siguiente **problema de investigación**:  
¿Cómo mejorar las vías de acceso al servicio de búsqueda y obtención de información referente a personas en la Universidad de las Ciencias Informáticas?

Para dar solución a este problema se tomará como **objeto de estudio** de la presente investigación: Las vías de acceso al servicio de búsqueda y obtención de información referente a personas. Derivándose como **campo de acción**: Las vías de acceso al servicio de búsqueda y obtención de información referente a personas en la Universidad de las Ciencias Informáticas. El **objetivo general** de esta investigación es: : Desarrollar una aplicación móvil que permita mejorar las vías de acceso al servicio de búsqueda y obtención de información referente a personas en la Universidad de las Ciencias Informáticas. Para darle respuesta al objetivo general propuesto se definen los siguientes **objetivos específicos**:

1. Analizar los conceptos relacionados con las aplicaciones para Android y aplicaciones sobre directorio.
2. Realizar el análisis y diseño de la aplicación móvil para el directorio de personas (UCI).
3. Probar la propuesta de solución desarrollada para comprobar el correcto funcionamiento de la misma.
4. Implementar la propuesta de solución desarrollada.

Para responder al desarrollo de la investigación se emplean los métodos científicos: **teóricos y empíricos**. Dentro de los **métodos empíricos** se aplicará la **entrevista** (ver Anexo 2) para obtener información referente sobre el directorio universitario que hoy existe para navegadores web; así como para precisar el problema a resolver, los problemas existentes y los servicios que brinda actualmente el directorio. Las entrevistas han permitido fácilmente al investigador tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica el investigador se acerca al problema de una forma natural interactuando con el cliente. Se aplica la **revisión bibliográfica**, que se pone de manifiesto en las revisiones a la bibliografía existente y en el momento de consultar la información en sitios de interés nacional e internacional, de manera similar se utilizó la **observación** método práctico como resultado de ser algo inherente al hombre.

De los **métodos teóricos** se usará el método **analítico-sintético** mediante el mismo se realiza el estudio y análisis de los elementos relacionados con los sistemas de búsqueda y los directorios, posibilitando definir las principales características de las herramientas, lenguajes y metodología a utilizar. El **modelado** se empleó para la construcciones de diagramas, mapas, el diseño de interfaz gráfica para una mejor representación o comprensión de la solución propuesta. El **método histórico-lógico** se utilizó para el estudio crítico de los trabajos anteriores acerca de otras aplicaciones o soluciones similares.

Para una mejor comprensión de la investigación, se definió una estructura capitular organizada en tres capítulos. Los capítulos que lo conforman son los siguientes:

**Capítulo 1. Fundamentación teórica:** Este capítulo expone los principales conceptos que serán tratados en el transcurso de la investigación. Se realiza la descripción y análisis de los sistemas existentes y una caracterización de las herramientas, tecnologías y la metodología de desarrollo a utilizar para la construcción de la propuesta de solución.

**Capítulo 2. Análisis y diseño de la solución:** En este capítulo se realiza un análisis de la propuesta de solución. Se muestra el modelo conceptual y se define mediante las técnicas de obtención de requisitos, los requisitos funcionales y no funcionales que debe cumplir el sistema. Se describe la arquitectura y patrones de diseño que se utilizan en la investigación.

**Capítulo 3. Implementación y validación de la solución:** Se aplican las técnicas para validar los requisitos definidos. Se realizan pruebas a la solución a través de la estrategia de pruebas seleccionada y se valoran los resultados de la investigación. Se elabora el diagrama de despliegue. El documento, incluye las conclusiones generales y las bibliografías consultadas que sirven de soporte a la investigación.

# CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA

## 1. Introducción

En este capítulo se abordan un conjunto de conceptos como fundamentación de la propuesta del trabajo investigativo para un mejor dominio del problema. Se destacan características de los elementos asociados al tema a investigar, a través del estudio de sistemas homólogos se analizan aplicaciones tanto en el ámbito nacional como el internacional. Se exponen la metodología y las principales herramientas que sirven de apoyo para la realización de la solución al problema planteado.

### 1.2. Sistema de búsqueda de información

Los sistemas de búsqueda de información son sistemas automáticos de recuperación de información que almacenan datos sobre páginas web en una base de datos. Son un conjunto de elementos interrelacionados destinados a la captura, depuración, almacenamiento, recuperación, actualización y tratamiento de dato (Justiz y Peñalver, 2016); por lo que constituyen una importante herramienta para recuperar y buscar información de forma dirigida a través de palabras claves que el interesado indica. Existen herramientas de búsqueda que ayudan a encontrar lo que se desea.

Dentro de las herramientas que se utilizan para la búsqueda de información, se encuentran (Negrín y Martínez, 2016):

- **Motores de búsquedas** consisten en bases de datos confeccionadas por robots, son sistemas de búsquedas por palabras clave, permiten el acceso al Texto completo de páginas seleccionadas. Ejemplos: Google, Bing, Ask, Yahoo! Search.
- **Meta buscadores** realizan de forma rápida y superficial sus búsquedas en varios motores a la vez. Ejemplos: Clusty, Ixquick, MetaCrawler, Dogpile.
- **Directorios** son sistemas de búsqueda por temas o categorías, la información es seleccionada y procesada manualmente.

### 1.2.1. Directorio

Según la Biblioteca Nacional de España, directorio es guía o lista de direcciones. Se pueden presentar en distintos soportes: papel, *offline* o en línea. El acceso en línea se está convirtiendo en la forma de presentación más habitual, ya que a la precisión, rapidez en la consulta y ahorro de espacio, se une la posibilidad de actualización frecuente o inmediata de sus contenidos. La American Library Association define como lista de personas u organizaciones ordenada sistemáticamente, por regla general alfabéticamente o por clases, figurando la dirección, filiación, etc. para las personas y la dirección, cargos, funciones y datos similares para las organizaciones.

Los directorios son fuentes de información que no que cambian con frecuencia cuya consulta permite obtener información puntual y precisa de forma inmediata sobre áreas o sectores de actividad muy diversos: administrativa, comercial, científica, industrial, empresarial, cultural, etc. La presente investigación se enfoca en el estudio de directorio de personas, la función de los directorios de personas es facilitar datos de identificación y localización de personas.

### 1.3. Estudio de homólogos

En una búsqueda exhaustiva de aplicaciones similares y una revisión bibliográfica para el análisis documental de trabajos que abordan temas relacionados a la presente tesis, con el objetivo de obtener una aproximación de las funcionalidades que debe tener la solución. A nivel internacional se han desarrollado diferentes aplicaciones que gestionan de manera automática la forma de obtención de información. Seguidamente se presentarán las aplicaciones estudiadas.

**Directorio de la Universidad de la República de Uruguay**, brinda la información de los diferentes servicios universitarios y oficinas de la Universidad de Uruguay, sin necesidad de hacer búsqueda, es decir, ofrecer informaciones principales como: dirección, biblioteca, institutos, decanato, extensión (Directorio de la República, 2015).

**La Cámara de Comercio Americana del Perú (AmCham Perú)** le provee información completa y detallada de todos los asociados y sus representantes además de los datos de instituciones, embajadas y otras Cámaras de Comercio, lo que le permite conocer quién es quién en el mundo de los negocios. Así mismo contiene datos básicos sobre el clima de negocios en el Perú y las regulaciones que afectan la

inversión. Para acceder al Directorio de asociados, debe comprobar su membresía a AmCham enviando un correo a: [directorio@amcham.org.pe](mailto:directorio@amcham.org.pe). Si ya cuenta con los datos necesarios de ingreso, puede acceder directamente por autenticación (Directorio Cámara de Comercio, 2016).

**Directorio Nacional de Personas Jurídicas** constituyen una valiosa herramienta de información referencial, para quienes tengan interés en conocer si una determinada persona jurídica está inscrita o no, así como si determinada denominación o razón social ya ha sido adoptada por otra persona jurídica inscrita. El Directorio tiene una actualización mensual, para su consulta libre por los interesados (Sunarp, 2016).

A nivel nacional Cuba no se encuentra excluida en cuanto a la utilización y desarrollo de los directorios, pues actualmente existen varias empresas que lo utilizan para el manejo y manipulación de la información. Un ejemplo de ellas, es la de ETECSA que cuenta con el directorio telefónico de todo el país, dicho directorio constituye una útil herramienta para la sociedad cubana y un soporte informativo de referencia, o sea, pone a disposición de personas jurídicas y naturales cubanas autorizadas a ejercer alguna actividad comercial, una sección donde estos pueden darse a conocer incluyendo sus datos de contacto, la información del **Directorio Telefónico Oficial de Cuba** es actualizada cada dos años (Bhait, 2011).

**Artex**<sup>1</sup> es una aplicación móvil que contiene un directorio (guía) que divide los lugares en listados: de tiendas, centros culturales, así como de las Academias de Arte y turismo en Cuba.

**AlaMesa**<sup>2</sup> es un proyecto de promoción de la gastronomía cubana y gestión de los servicios asociados, permite encontrar cafeterías, bares, restaurantes, entre otros lugares sin necesidad de conectarse a Internet. Es un directorio de restaurantes y entidades gastronómicas, con el que es posible interactuar a través de un buscador y una serie de enlaces propuestos bajo diversas categorías. Dicho directorio contiene información precisa sobre cada una de las entidades registradas en la base de datos, lo cual ofrece al usuario final los elementos básicos necesarios para decidir a donde ir a comer. Paralelo a la disponibilidad libre de costo de la información completa en el directorio. AlaMesa ofrece a los restaurantes interesados un catálogo de servicios compensados que permiten fortalecer su actividad promocional.

---

<sup>1</sup> Se puede descargar en Apklis <http://apklis.cu/>

<sup>2</sup> idem

Existen otras organizaciones como las universidades y centros de estudios que utilizan sistemas informáticos similares para consultar datos específicos de toda su comunidad. La Universidad de las Ciencias Informáticas, como parte de la comunicación institucional cuenta con un portal de cara a Internet. El directorio del Portal de la Universidad solo muestra la información de los principales directivos y algunas de las personas involucradas en el proceso de investigación. Este agrupa las búsquedas por tres criterios: personas, universidades y teléfonos, y permite filtrar por el nombre, cargo y formación.

A continuación, se describen brevemente sus funcionalidades. **El Directorio de Personas:** muestra los datos generales de los principales directivos de la Universidad y las personas que participan activamente en el proceso de investigación. Permite buscar por categoría docente (*Instructor recién graduado, Instructor, Profesor Asistente, Profesor Auxiliar y Profesor Titular*) y por categoría científica (*Máster y Doctor*). **El Directorio de universidades cubanas:** muestra un listado de universidades y centros universitarios que existen en el país. Este permite buscar por provincias. **El Directorio Telefónico:** muestra los teléfonos de las áreas de la Universidad y de sus principales directivos. Permite buscar por el nombre del área (Justiz y Peñalver, 2016).

Por otra parte la UCI internamente, cuenta con un Directorio UCI, incluye el Directorio de personas mediante el cual se hace posible el proceso de búsqueda de información de todas las personas del campus a través de criterios, tales como: Nombre, Apellidos Solapín, No. Expediente y Usuario. Además de apoyarse en diversos filtros de búsquedas como: Categoría, Cargo, Área, Provincia, Municipio. El Directorio UCI incluye también el Directorio Telefónico y el Directorio de Aplicaciones UCI (Justiz y Peñalver, 2016).

Después de analizar cada una de las soluciones, se identificaron características comunes, la presencia de un buscador que permite introducir texto, no es necesario la autenticación, las formas de búsqueda y la posibilidad de aplicar filtros o la combinación de estos. Se verificó que las soluciones no cumplen todas las expectativas requeridas, por ejemplo, el Directorio UCI no permite al usuario crear sus propias listas de personas, no está implementado con un diseño adaptativo y dado a que las características actuales de su implementación no permiten una actualización tecnológica, se desarrollará una aplicación para dispositivos móviles, se tendrá en cuenta las funcionalidades analizadas para conformar los requisitos de la propuesta de solución. Aunque las aplicaciones estudiadas fueron desarrolladas para contextos diferentes, aportaron conocimientos acerca de las características, implementación y principios de funcionamiento, que sirve como base para guiar el desarrollo de la solución propuesta, se ha tenido en cuenta las características propuestas

por la consultora Gartner, dicha consultora propone que la aplicación móvil debe ser fácil de usar, libre, trabajar online o/y offline y que pueda trabajar en la mayor cantidad de dispositivos.

## **1.4. Aplicaciones móviles**

Las aplicaciones móviles son programas desarrollados para que funcionen en dispositivos móviles (Cáceres et al, 2013). La tecnología celular posee dos características principales que la hacen diferente a otras formas: movilidad y amplio alcance. Los dispositivos móviles generalmente se dividen en dos categorías, teléfonos y *tabletas*. Para el desarrollo de aplicaciones móviles se pueden usar marcos de trabajos nativos estos están diseñados para sistemas operativos propios de estos dispositivos, es decir, utilizan un entorno de desarrollo integrado, un lenguaje y herramientas propias de cada plataforma por lo general no son compatibles entre estas. Por otro lado están los marcos de trabajo multiplataforma, que operan en independencia de los sistemas operativos. Existen varios sistemas operativos móviles, entre ellos los más usados son: Androide creada por Google en 2008 y los iOS creado por Apple en 2007. Microsoft también tiene su sistema operativo móvil, el Windows Phone siendo este el tercer sistema más usado (IDC, 2020).

Para maximizar el uso de un software, este debe tener como características poder ejecutarse y ser compatible con la mayor cantidad de sistemas operativos y versiones de estos. Una solución consiste en el desarrollo nativo de la aplicación para cada una de las plataformas existentes utilizando el entorno de desarrollo integrado (IDE por sus siglas en inglés), el lenguaje y las herramientas propias de cada plataforma.

### **1.4.1. Aplicaciones nativas**

Las aplicaciones nativas son las propias de cada plataforma. Deben ser desarrolladas en total compatibilidad con la plataforma a la que va implementada. Los desarrollos que pretenden soportar plataformas diferentes suelen necesitar un esfuerzo extra (Vique, 2013). Las aplicaciones nativas ofrecen la posibilidad de acceder a todas las capacidades del dispositivo, su rendimiento es alto, sin embargo, no es posible reutilizar el código fuente para implementarse en plataformas diferentes, el esfuerzo y tiempo se multiplica y se elevan los costos de desarrollo, actualización y distribución de nuevas versiones (Montan, 2017), (Delía et al, 2014).

En la Tabla 1 se muestran las principales tecnologías para el desarrollo de aplicaciones nativas en correspondencia con los sistemas operativos mas usados en dispositivos móviles.

Tabla 1- Principales tecnologías para desarrollo de aplicaciones nativas

Plataforma	Lenguaje	Ambiente de Desarrollo
Android	Java/Kotlin	Android Studio/Eclipse
iOS	Objective-C	XCode
Windows Phone	C#	Visual Studio

### 1.4.2. Web Apps

Estas aplicaciones, están diseñadas para ejecutarse desde un navegador web, se desarrollan con tecnologías afines (HTML, CSS y JavaScript) y cuentan con una serie de características favorables: no necesitan adecuarse a ningún sistema operativo, son independientes de la plataforma y su puesta en marcha es rápida y sencilla (Delía et al, 2014). Estas aplicaciones presentan algunas desventajas en comparación con las aplicaciones nativas. El acceso a los recursos del dispositivo es limitado, la experiencia del usuario es afectada por el hecho de que la interfaz no es nativa, la aplicación no es instalada en el dispositivo, los tiempos de respuesta decaen siendo afectados por la interacción cliente-servidor y generalmente solo funcionan con el dispositivo conectado a Internet.

### 1.4.3. Entorno de desarrollo multiplataforma

El desarrollo multiplataforma procura optimizar la relación costo/beneficio compartiendo la misma codificación entre las versiones para las distintas plataformas. Entre otras ventajas sobresalen: menor tiempo y costo de desarrollo; prestaciones cercanas a las nativas con acceso al hardware del dispositivo (Delía et al, 2014).

En la Tabla 2 se muestra una caracterización de las ventajas y desventajas que existen entre los tipos de aplicaciones anteriormente definidas.

Tabla 2 – Ventajas y desventajas de las categorías de desarrollo

<b>Criterio</b>	<b>Nativa</b>	<b>Web Apps</b>	<b>Híbrida</b>
Multiplataforma	No	Sí	Sí
Costo de desarrollo	Alto	Bajo	Bajo/Medio
Desempeño	Alto	Bajo	Intermediario
Aceso al dispositivo	Total	Limitado	Total
Utilización offline	Sí	No	Sí
Tienda de aplicaciones	Sí	No	Sí
Experiencia del usuario	Alta	Baja	Media/Alta
Facilidad de mantenimiento	Baja	Alta	Alta

## 1.5. Servicios Web

Con los avances del Internet y de los protocolos de comunicación basados en *Extensible Markup Language* (XML), han surgido los servicios web con el objetivo de integrar sistemas, un servicio web es un grupo de protocolos usado para intercambiar datos entre aplicaciones (Rosario, 2012). La idea de publicar o consumir servicios web en dispositivos móviles nace del importante uso que estos están teniendo en los últimos tiempos y al incremento permanente de las prestaciones de dichos dispositivos (segundo, Hernan y Joaquin, 2015). Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio accediendo a estos mediante la red.

Los servicios web son accedidos a través de un servidor web y de un modo independiente de la plataforma, utilizando protocolos y estándares (HTTP, SOAP, WSDL, UDDI). Para el desarrollo de la solución se selecciona como protocolo de servicios *Simple Object Access Protocol* (SOAP). Es un protocolo de

comunicación HTTP basado en XML usado para invocar aplicaciones o intercambio de mensajes, en un ambiente independiente de la plataforma y lenguaje de programación (Papazoglou, 2008).

## 1.6. Ciclo de desarrollo de software

El ciclo de vida de desarrollo móvil no es en gran medida diferente que el *Software Development Lifecycle* (SDLC) para web o aplicaciones de escritorio. Con frecuencia el proceso tiene 5 partes fundamentales (Davidortinau, 2019):

1. Concepción– Todas aplicaciones comienzan con una idea.
2. Diseño – La fase de diseño consiste en definir la experiencia del usuario de la aplicación, como es el área general, transformar esa experiencia en un diseño de interfaz de usuario adecuado, generalmente con la ayuda de un diseñador gráfico.
3. Desarrollo – En esta fase es donde se construye la aplicación.
4. Pruebas – En esta fase se realizan las pruebas funcionales de la aplicación y los errores detectados son corregidos.
5. Implantación – En esta fase es donde la aplicación se pone a disposición de los usuarios finales.

## 1.7. Metodología de desarrollo de software

En los años 70, debido a la llamada crisis del Software, muchos proyectos enfrentaban dificultades para ser entregados debido al gran crecimiento de la demanda del producto. Algunos de estos problemas eran por el hecho de no existir una metodología definida para la creación de software (Sommerville, 2011).

La metodología de desarrollo de software es lo que permite asegurar el mejor resultado posible de un proyecto. Existen las **metodologías ágiles y las tradicionales**, las **metodologías tradicionales** se caracterizan por el gran nivel de detalle aún en el inicio del proyecto, por su total rigidez y muy dirigidos a la documentación que se genera tras cada una de las actividades desarrolladas, mientras que las **metodologías ágiles** se diferencian por la entrega progresiva de los productos a los clientes, o sea son incrementales e iterativas, garantizando así una mayor flexibilidad y adaptabilidad para trabajar en un ambiente de cambios constantes (Sommerville, 2011).

En 2001 un grupo de 17 desarrolladores elabora “The Agile Manifesto”, algunos elementos del manifiesto que resumen en buena medida la filosofía detrás de esta metodología, son: Los individuos y sus interacciones son más importantes que los procesos y las herramientas. Poner el software en funcionamiento es más importante que realizar una documentación exhaustiva. La colaboración con el cliente es más importante que la negociación del contrato. Responder al cambio es más importante que seguir un plan (Velez, Zapata y Eno, 2018).

Según un estudio, conducido por PwC, indica que los métodos ágiles son 28% más exitosos que los tradicionales. Los proyectos de software en general presentan muchos problemas como: cancelación de proyectos; sobrepasan el presupuesto previsto; no terminan en el plazo. Lo que demuestra déficit en la gestión de los proyectos de software. La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas en las actividades del proyecto con el objetivo de alcanzar los requisitos del mismo (Battistella, Atangarlin y Von Wangenheim, 2016).

### 1.7.1. Selección de la metodología a utilizar

Existe hoy una variedad de metodologías o métodos ágiles para el desarrollo de software, para el presente trabajo se escogió **Scrum**, un marco de trabajo o *framework* creado por Jeff Sutherland y Ken Schwaber en los años 1990 para ayudar organizaciones que trabajan en proyectos complejos, que tienen la necesidad de obtener resultados rápidos, con requisitos que son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Battistella, Atangarlin y Von Wangenheim, 2016).; Scrum tiene como finalidad la entrega de valor en períodos cortos de tiempo, basada en tres pilares: la transparencia, inspección y adaptación (Abellán, 2020). En este marco de trabajo el cliente tiene un papel central para el desarrollo de las actividades, interactuando fuertemente con el equipo de desarrollo.

En Scrum existen 3 roles: **Product Owner** establece los requerimientos del proyecto e indica las prioridades de la lista de tareas, es quien posee los conocimientos del negocio y de las necesidades del cliente, es el responsable en reunir las informaciones y conocimientos necesarios para el equipo de desarrollo entender lo que será necesario para desarrollar el producto. **Scrum Master** guía la elaboración del proyecto y **Equipo de desarrollo**, asegura que la metodología está siendo debidamente aplicada. Scrum se basa en la teoría de control de procesos empíricos o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce (Painka, Marcelo y Marchi, Késsia, 2013).

Los artefactos que genera la metodología Scrum son:

- ❖ **Product Backlog** - son las lista de requisitos del proyecto (*user stories*).
- ❖ **Sprint Backlog** – se refiere a la iteración o ciclo de actividades del equipo de proyecto, o sea, es la descomposición del *product backlog* (Selección de requerimientos y/o escenas) en tareas para lograr un incremento del producto. Cada sprint puede durar como máximo 30 días.
- ❖ **Increment** - las reuniones diarias (permiten al equipo saber el estado del proyecto, posibles desviaciones, estado de la calidad, posibles riesgos) versión del software, potencialmente desplegable en el ambiente de producción, obtenida como resultado de la ejecución de un sprint.

La metodología Scrum propone las siguientes tres fases (Mariño y Alfonso, 2014):

1) **Fase de Planeamiento** – es subdividida en: a) Planeación – se define el equipo del proyecto, herramientas, el sistema de desarrollo y se crea el *product backlog* con la lista de requerimientos conocidos hasta ese momento, se definen prioridades para los requerimientos y se estima el esfuerzo necesario para llevar a cabo la implementación de los mismos; y b) Diseño Arquitectónico – se define la arquitectura del producto que permita implementar los requerimientos definidos.

2) **Fase de Desarrollo** - es la parte ágil, donde el sistema se desarrolla en *sprints*. Cada *sprint* incluye las fases tradicionales del desarrollo de software – relevamiento de requerimientos, análisis, diseño, implementación y entrega.

3) **Fase de Finalización** – incluye integración, pruebas y documentación. Indica la implementación de todos los requerimientos, quedando el *product backlog* vacío y el sistema listo para entrar en producción.

De acuerdo con los datos de la Scrum Alliance, 62% de todos los proyectos desarrollados con la metodología Scrum son exitosos; Scrum es sencillo de entender y es usado mundialmente, para (Schawaber y Sutherland, 2017):

1. Pesquisar e identificar mercados factibles, tecnologías y funcionalidades de productos;
2. Desallorar productos y mejoras.
3. Lanzar productos y mejoras frecuentes, pudiendo llegar a varias veces por día.

4. Desarrollar y mantener ambientes en la *Cloud* y otros ambientes operacionales para uso de productos.

5. Mantener y renovar productos.

Una metodología tiene que ser usada de forma simple y efectiva, sin hacer las tareas de diseño excesivamente complejas y scrum es bastante útil en ese sentido.

## 1.8. Herramientas y tecnologías a utilizar

### 1.8.1. Sistema operativo android

Androide es un sistema operativo (Jordan y Greyling, 2011) de código abierto basado en Linux (Burnette, 2010), creado para celulares y otros dispositivos. A continuación se exponen algunas de las principales ventajas:

- ✓ **Software Libre.** Android está liberado con licencia Apache y es software libre, lo que lo convierte en un sistema operativo totalmente libre para que cualquier desarrollador no solo pueda modificar su código, sino también mejorarlo. Puede publicar las nuevas mejoras y el nuevo código, ayudar a mejorar el sistema para futuras versiones sin depender de fabricantes o distribuidores. Al ser código abierto garantiza que, en caso de haber un error de programación, sea detectado y reparado con rapidez, al no existir ninguna traba legal para tracear el código fuente, ni depender de alguien para pedir autorización.
- ✓ **Comunidad de desarrollo.** Android cuenta con la comunidad más grande a nivel mundial de desarrolladores, con muchos de eventos, concursos, competencias y reuniones, así como múltiples vías de comunicación como foros y *chats* oficiales para fomentar la participación y la colaboración para encontrar mejoras e ideas para futuras versiones. Por otro lado, las modificaciones o mejoras no dependerán de un limitado equipo de desarrolladores de una empresa, sino que contarán con el apoyo, respaldo y participación de todos los desarrolladores del mundo.
- ✓ **Costos.** Precisamente por el hecho de que Androide puede ser instalado en teléfonos de cualquier fabricante o incluso en otros dispositivos sin costos adicionales de licencias.

## 1.8.2. Android Studio v3.6

Android Studio es un Entorno de Desarrollo Integrado (IDE), oficial para la plataforma Android, publicado gratuitamente por Google (Klausen, 2018) para los desarrolladores Android. Es basado en IntelliJ IDEA (Fortes y Azevedo, 2018), uno de los IDE para desarrollo en Java. Android Studio soporta los siguientes lenguajes de programación: Kotlin, el cual fue creado inicialmente en 2012 por la compañía rusa JetBrains, que busca sustituir a Java como Lenguaje estándar para la creación de aplicaciones en el sistema operativo Android. El pasado año del 2017 Google anunció el soporte oficial para Kotlin (Fortes y Azevedo, 2018).

## 1.8.3. Lenguajes de programación

Para el desarrollo de aplicaciones Android la programación se puede realizar utilizando alguno de los siguientes lenguajes: Java, Kotlin y el uso de XML, la parte lógica de la aplicación se desarrolla con Java o Kotlin y la parte visual, con XML. El XML es un lenguaje que nos permite incorporar componentes visuales a nuestra aplicación tales como botones o cajas de texto (Ávila, 2018).

Kotlin basándose en algunos pilares clave como (Hernández, 2019) :

- Concisión frente a la verbosidad presente en un lenguaje con años como es Java. Con Kotlin se puede ahorrar la cantidad de “verbosidad”. Por verbosidad se puede definir a la cantidad de código escrito necesario para realizar una acción concreta.
- Interoperabilidad, sin duda la más importante, la facilidad de transformación de proyectos Java a Kotlin y su posible convivencia en una misma aplicación. No tienen una sintaxis compatible, pero Kotlin sí está diseñado para interoperar con Java, es decir es posible mezclar código creado en Java y Kotlin en un mismo archivo y compilar sin problemas.
- De fácil mantenimiento y escalable.

Java es un lenguaje de programación orientado a objeto. Es un lenguaje fácil de aprender (Oracle, 2019). Las comunidades de desarrolladores Java y Kotlin son muy activa y está creando continuamente soluciones para Androide.

## **1.8.4. Lenguaje Unificado de Modelado UML**

El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio (Larman, 2015). El objetivo de UML es proporcionar a desarrolladores de software, arquitectos de sistemas e ingenieros de software de herramientas para el análisis, diseño e implementación de sistemas basados en software.

### **1.8.4.1. Visual Paradigm**

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación) fácil de usar que propicia un conjunto de ayudas para el desarrollo de programas informáticos, en el presente proyecto se ha usado Visual Paradigm UML Enterprise Edition 8.0 para la construcción de los diagramas.

## **1.8.5. Pencil**

Pencil es una herramienta de código abierto disponible para todas plataformas para la creación de prototipo de interfaz de usuario de forma rápida y sencilla. En el presente proyecto se usó la versión 3.0.4.

## **1.8.6. SQLite**

Un sistema de bases de datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia base de datos como una especie de archivo electrónico, es decir, es un depósito o contenedor de una colección de archivos de datos computarizados. Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo (Date, 2001):

- ✓ Insertar datos dentro de los archivos existentes
- ✓ Modificar datos en archivos existentes
- ✓ Eliminar datos de los archivos existentes

En cuanto a bases de datos, Android provee la opción Sqlite, que permite crear y administrar una base de datos local. Sqlite es muy usado con aplicaciones móviles. La mayoría de los sistemas de gestión de base de datos (SGBD's) ofrecen un lenguaje declarativo de alto nivel. La más conocida es SQL, del inglés Structured Query Language. Este lenguaje permite hacer la definición de los datos, la inserción, actualización y consulta de los datos (Cavalho, Azevedo y Abreu, 2007).

## 1.9. Conclusiones parciales

El análisis realizado en el capítulo ha permitido:

- El estudio de los temas fundamentales sobre la problemática, para mejor dominio de la investigación.
- El análisis de las principales características y aplicaciones de los sistemas de búsqueda en el ámbito internacional y nacional ha demostrado las limitantes de los sistemas, como por ejemplo algunas de las soluciones estudiadas no presentan buscadores, es decir, la información que muestran no es dinámica, lo que imposibilita al usuario realizar búsquedas más específicas de lo que desea.
- Además, se efectuó la caracterización de la metodología, lenguajes, tecnologías y herramientas a emplear en el desarrollo de la solución. La metodología a utilizar para guiar el desarrollo de la solución es Scrum por ser una metodología simple y efectiva, no hace la tarea del diseño excesivamente compleja. Se utiliza Visual Paradigm para UML como herramienta para el modelado del sistema. Como IDE a utilizar Android Studio.

## **CAPITULO II - PROPUESTA DE SOLUCIÓN**

### **2.1. Introducción**

En el presente capítulo se hace un análisis que consiste en conseguir una comprensión más precisa de los requisitos, refinarlos, estructurarlos y proporcionar una visión general de la solución propuesta. También, siguiendo la metodología Scrum, se desarrollarán los artefactos pertinentes. Se realiza un estudio de la arquitectura y patrones de diseño.

### **2.2. Modelo de dominio**

El modelo de dominio o modelo conceptual es una abstracción del sistema, o sea, representa el concepto principal de la investigación y las ideas asociadas a este (Larman, 2015).

La modelación del negocio tiene los siguientes propósitos:

- Entender la estructura y la dinámica de la organización.
- Entender los problemas actuales e identificar mejoras potenciales.
- Asegurarse de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización.
- Derivar los requerimientos del sistema.

Para una mejor comprensión de la aplicación a desarrollar, en la Figura 1 se muestra el modelo de dominio de la presente investigación.

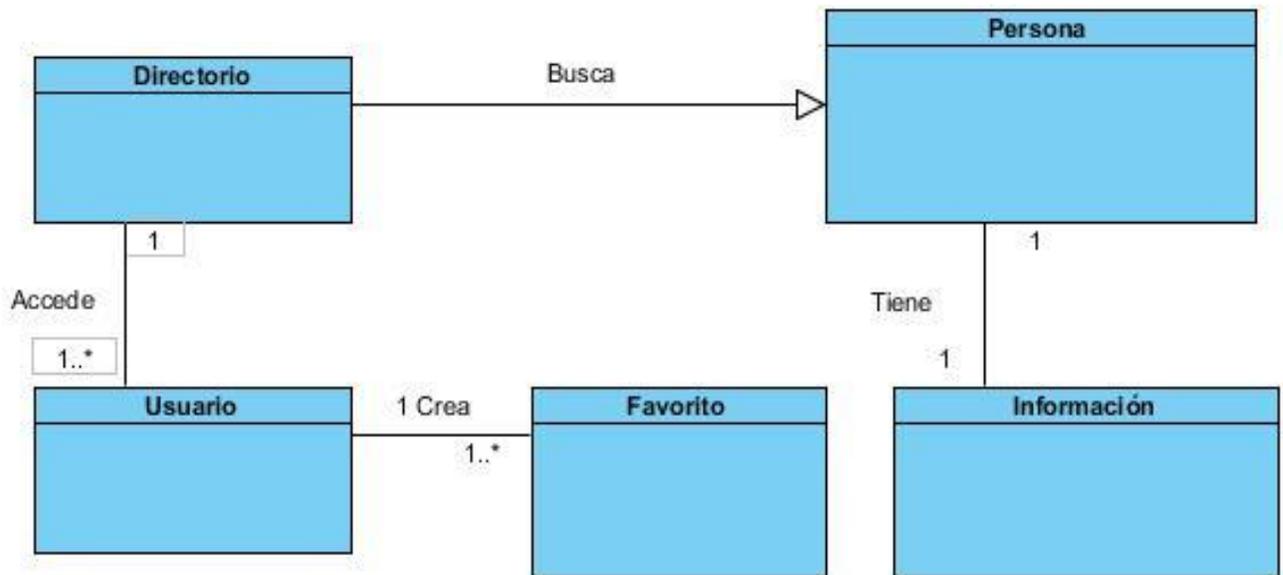


Figura 1: Modelo de dominio

Para un mayor entendimiento del modelo de dominio y con el objetivo de lograr un lenguaje común para la comprensión del contexto del sistema, se muestran a continuación los conceptos más importantes utilizados en el modelo:

**Directorio:** servicio que brinda la posibilidad a los usuarios la búsqueda de información referentes a personas de la universidad.

**Información:** representa la fuente de datos que va a proporcionar la información que es mostrada a través del directorio de personas.

**Usuario:** representa a todo el personal de la universidad que accede al Directorio UCI para consultar información.

**Persona:** representa la información de las personas consultadas.

**Favorito:** representa la funcionalidad que permite al usuario crear su lista de contactos.

### **2.3. Propuesta de solución**

La propuesta de solución es una idea o visión del software a implementar, esto permite que el cliente y el equipo de trabajo puedan tener una idea de como quedará el producto final y la utilidad del mismo. El sistema propuesto se corresponde con una aplicación para dispositivos móviles con sistema operativo Androide. Este permitirá a los usuarios realizar búsquedas rápidas atendiendo a los diferentes criterios de pesquisas especificados, para identificar, localizar o contactar personas en la universidad sin la necesidad de un navegador web. Destacándose la característica de dar la posibilidad a los usuarios de crear sus propias listas de contactos UCI. La solución será desarrollada para ser instalada de forma nativa en los dispositivos móviles que cuenten con Androide. Al contar con un diseño adaptativo la visualización de la información será óptima en los diferentes dispositivos, aprovechando al máximo la región útil de la pantalla.

### **2.4. Producto BackLog**

Producto BackLog es una lista que reúne las funcionalidades deseadas por el cliente y elementos indispensables para garantizar el éxito de un proyecto. Su función es documentar todos los requisitos que van surgiendo desde el inicio y durante el desarrollo del proyecto. Es importante destacar que es una lista dinámica que está en constante evolución. Los requisitos se clasifican en dos grupos: requisitos funcionales y requisitos no funcionales. La captura de requisitos, se obtuvo de fuentes diversas: documentos, aplicaciones existentes, y fundamentalmente a través de entrevistas con el cliente, en este caso, la Dirección del Centro de Desarrollo de Tecnologías para la Formación.

A continuación, se especifican los requisitos funcionales de la aplicación:

Tabla 3 – Producto BackLog con los requisitos funcionales

Producto BackLog				
Nº	Nombre	Descripción	Estimación	Prioridad (Alta/baja/media)
RF1	Leer códigos de Barra	Se requiere la lectura de códigos de Barra asociados a los solapines	3 semanas	Media
RF2	Buscar persona	Se requiere buscar las informaciones básicas de una persona.	2 semanas	Alta
RF3	Mostrar datos de la persona buscada	Se requiere presentar los datos organizados de de una persona buscada.	1semana	Média
RF4	Filtrar por provincia	Se requiere hacer una búsqueda por provincia	1semana	Media
RF5	Filtrar por facultad	Permite el usuario hacer una búsqueda por facultad	1 semana	Media
RF6	Filtrar por año	Se requiere hacer una búsqueda por año	1semana	Media
RF7	Filtrar por curso	Se requiere hacer una búsqueda por curso	1semana	Media
RF8	Agregar Favoritos	Se requiere guardar los usuarios favoritos .	1 semana	Media
RF9	Visualizar Favoritos	Se requiere visualizar los usuarios favoritos .	1semana	Media
RF10	Eliminar Favoritos	Se requiere borrar un determinado usuarios de los favoritos .	1semana	Media
RF11	Visualizar Mapa de la UCI	Se requiere que la aplicación permita saber la ubicación de áreas de la universidad	2 semanas	Baja
RF12	Ampliar la imagen	Ampliar la imagen de una persona cuando se le hace click	1semana	Baja

Los requisitos no funcionales imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener (Justiz y Peñalver, 2016). En la Tabla 4 se muestran los requisitos no funcionales definidos para la propuesta de solución.

Tabla 4 – Producto BackLog con los requisitos no funcionales

Product BackLog				
Nº	Nombre	Descripción	Estimación	Prioridad (Alta/baja/media)
RNF1	Portabilidad	Se requiere que la aplicación funcione en Smartphones con S.O Android 7.0 o superior		Media
RNF2	Usabilidad	La terminología del menú debe ser constante en todo el sitio. No pueden existir páginas internas donde existan elementos diferentes del menú o simplemente no aparezcan.  La aplicación será utilizado por todo el personal que tenga acceso a la red de la UCI.		Media
RNF3	Soporte	El sistema debe ser escalable, para en un futuro incorporarle nuevas funcionalidades en caso de ser necesarias.		Media
RNF4	Interfaz de la Aplicación	Se requiere una interfaz de usuario sencilla, fácil de usar pero elegante a la vez.	4 semanas	Alta
RNF5	Seguridad	Tendrá acceso a la aplicación (datos) solo el personal que tenga acceso a la red de la UCI.		Alta

## 2.5. Historia de Usuario

Con el objetivo de presentar los requisitos funcionales de una manera más consistente y detallada, se realizaron las historias de usuario (HU) lo que permite hacer una descripción de los requisitos funcionales de la aplicación. A continuación se muestran ejemplos de HU generadas en la investigación.

Tabla 5 – Historia de Usuario “Buscar Persona”

Historia de Usuario	
<b>Código:</b> HU12	<b>Nombre Historia de Usuario:</b> Buscar Persona
<b>Modificación de Historia de Usuario Número:</b> <i>ninguna</i>	
<b>Referencia:</b> RF2	
<b>Programador:</b> Armando Costa	<b>Iteración Asignada:</b> <i>Primera</i>
<b>Prioridad:</b> <i>Alta</i>	<b>Tiempo estimado:</b> <i>2 semanas</i>
<b>Riesgo en Desarrollo:</b> <i>Medio</i> (Falta experiencia con el desarrollo de herramientas y tecnologías)	
<p><b>Descripción:</b> muestra la interfaz con buscador que permite a los usuarios Buscar una determinada persona. Para la búsqueda de una persona el usuario indica el criterio o filtro de búsqueda:</p> <ul style="list-style-type: none"> <li>▲ Se debe especificar un criterio de búsqueda, (obligatorio): campo de texto</li> </ul> <p>El botón:</p> <ul style="list-style-type: none"> <li>▲ Buscar: para buscar la persona.</li> </ul> <p>Una vez introducido el criterio de búsqueda el usuario selecciona el botón y se muestra una tela con informaciones de las personas que coincidan.</p> <p>En el área de iconos flotantes se encuentran las opciones Favoritos.</p>	
<p><b>Observaciones:</b></p> <ol style="list-style-type: none"> <li>1. En caso que la persona no existe, se muestra un mensaje “La persona no existe o verificar la descripción de búsqueda”.</li> <li>2. El usuario deberá llenar al menos uno de los campos mostrados para obtener los resultados de la búsqueda.</li> </ol>	

3. En caso de que introduzca datos incorrectos o que no se encuentra en el sistema se muestra un mensaje de error “ No hay resultados de la búsqueda”.
4. En caso de escribir un solo carácter en el campo obligatorio se mostrará el mensaje “Entre al menos 2 caracteres”.
5. Si el usuario introduce uno de los criterios especificados correctamente se mostrarán todos los datos relacionados al criterio especificado.

**Rol:** Usuario

**Prototipo de interfaz:**



The screenshot shows a mobile application interface titled "Directorio" (Directory). At the top, there is a search bar labeled "Buscar:" with a magnifying glass icon. Below the search bar are several input fields for search criteria: "Nombre:", "Apellidos:", "Usuario:", "Solapin:", "Apartamento:", "Teléfono:", "Categorías:", "Cargos:", "Áreas:", "Provincias:", "Municipios:", and "Género:". Each field has a corresponding dropdown arrow on the right. A blue circular button with a white envelope icon is located at the bottom right of the form. The interface is displayed on a mobile device screen with a status bar at the top showing the time 17:03 and various icons.

**IU. Buscar Persona.**

Tabla 6 – Historia de Usuario “Agregar Favorito”

Historia de Usuario	
<b>Código:</b> HU13	<b>Nombre Historia de Usuario:</b> Agregar Favorito
<b>Modificación de Historia de Usuario Número:</b> <i>ninguna</i>	
<b>Referencia:</b> RF2	
<b>Programador:</b> Armando Costa	<b>Iteración Asignada:</b> <i>Primera</i>
<b>Prioridad:</b> Media	<b>Puntos Estimados:</b> <i>1 semanas</i>
<b>Riesgo en Desarrollo:</b> <i>Medio</i>	
<p><b>Descripción:</b> La historia de usuario permite guardar la información de una determinada persona. El usuario selecciona el ícono representado por una estrella</p> <p>El botón:</p> <ul style="list-style-type: none"> <li>▲ Favorito: para guardar la información de una determinada persona.</li> </ul> <p>Una vez seleccionado el botón favorito se actualiza el listado y se muestra un mensaje de información “El elemento ha sido agregado a la lista satisfactoriamente”.</p> <p>En el área de iconos flotantes se encuentran las opciones Listar Favoritos</p>	
<b>Rol:</b> Usuario	

## 2.6. Mapa de Navegación

Los mapas de navegación se utilizan para ofrecer una visión global de todo el sistema y mostrar todos los flujos y enlaces que existen entre las diferentes pantallas. Una vez descrita la solución propuesta se muestran como quedaron distribuidas las funcionalidades de la propuesta de solución.

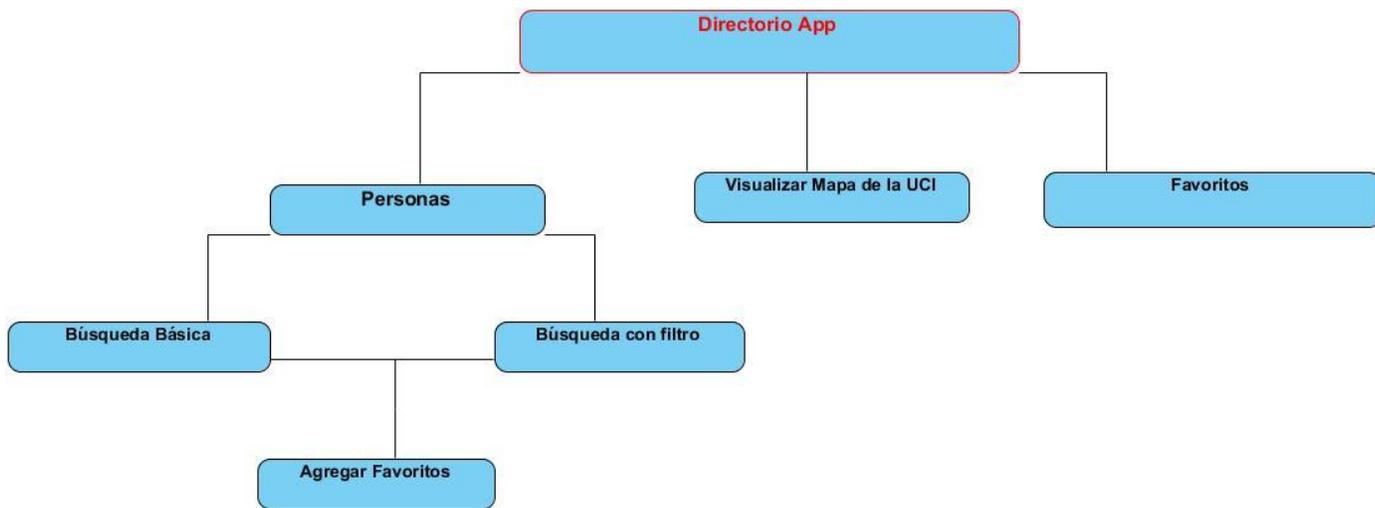


Figura 2: Mapa de Navegación de la aplicación

## 2.7. Estilo arquitectónico Cliente-Servidor

El estilo arquitectónico permite definir los componentes del sistema, su relación e interacción (Bahit, 211). En el desarrollo de la investigación se usó el estilo arquitectónico Cliente-Servidor, que es un modelo de aplicación distribuido en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes (Justiz, 2016). Un cliente realiza peticiones a otro programa conocido como servidor que le da respuesta. Ambas partes deben estar conectadas entre sí mediante una red. Básicamente el servidor disponibiliza un conjunto de servicios y el cliente hace uso de estos servicios. Dicho estilo arquitectónico facilita la integración entre sistemas diferentes independientemente de las tecnologías en que estén desarrollados.

## 2.8. Patrón de arquitectura

Un patrón arquitectónico define la estructura básica de un Sistema (Bahit, 2011). Modelo-vista-controlador (MVC) es uno de los tantos patrones de arquitectura de software, separa los datos de una aplicación (Modelo), la interfaz de usuario (Vista), y la lógica de control en tres componentes distintos (Controlador) que se relacionan entre sí. Para la presente investigación se empleó el MVC, ayuda al programador a colocar cada elemento de código en su lugar y a construir aplicaciones altamente modulares, extensibles, portables,

mantenibles y fácilmente escalables. MVC divide las aplicaciones en tres niveles de abstracción (Bahit, 2011):

- **Modelo:** representa la lógica de negocio. Es el encargado de acceder de forma directa a los datos actuando como intermediario con la base de datos. Un ejemplo dentro de la propuesta de solución es la clase ConsumoInfo.
- **Vista:** es la encargada de mostrar la información al usuario de forma gráfica y humanamente legible. En Android, las interfaces se construyen en XML y un ejemplo dentro de la propuesta de solución es la vista busqueda.xml.
- **Controlador:** es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario (entradas de mouse o teclado) solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma humanamente legible. Un ejemplo dentro de la propuesta de solución es la clase DirectorioActivity.



Figura 3: Funcionamiento del patrón modelo-vista-controlador

## 2.9. Patrones de diseño

Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces. Los patrones de diseño se pueden utilizar en cualquier lenguaje de programación orientado a objetos (Martínez, 2018). El patrón de diseño proporciona una solución probada a un problema común, documentada en un formato coherente. De la definición expuesta se puede deducir que los patrones de diseño proponen una forma de recopilación del conocimiento y reutilización de la experiencia de los desarrolladores, para solucionar problemas que ocurren de forma frecuente en el desarrollo de software.

### 2.9.1. Patrones de asignación de responsabilidad (GRASP)

Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades (Bahit, 2011). Los patrones GRASP pueden ser:

- Experto en Información- Asigna una responsabilidad al experto en información -la clase que tiene la información necesaria para realizar la responsabilidad. Se utilizó en clases como VerPersona.
- Creador- Asigna a una clase B la responsabilidad de crear una instancia de clase A. Se utilizó este patrón en clases como DirectorioActivity para la creación de los objetos.
- Bajo Acoplamiento- Una clase con bajo acoplamiento no depende de muchas clases. Se utilizó este patrón en la clase ConsumoInfo.

### 2.9.2. Patrones de la pandilla de los cuatro (Gang-of-Four)

Los patrones GOF son útiles durante el diseño de objetos (Bahit, 2011). Los patrones GoF pueden ser de tres tipos: de creación, comportamiento y estructurales. En la solución propuesta se utilizó el patrón de creación **Singleton** (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase, forzando a que solo se pueda crear una única instancia. En la implementación realizada se utiliza en la clase DirectorioActivity.

## 2.10. Modelo datos (BD)

Un modelo de datos es una colección de conceptos y reglas que se emplean para describir la estructura de una base de datos que incluye entidades, atributos y relaciones entre estos (Justiz, 2016). A continuación, se presenta el modelo de datos que contiene las entidades que serán utilizadas por las funcionalidades a desarrollar y las relaciones entre ellas, las mismas representan las tablas en la base de datos.

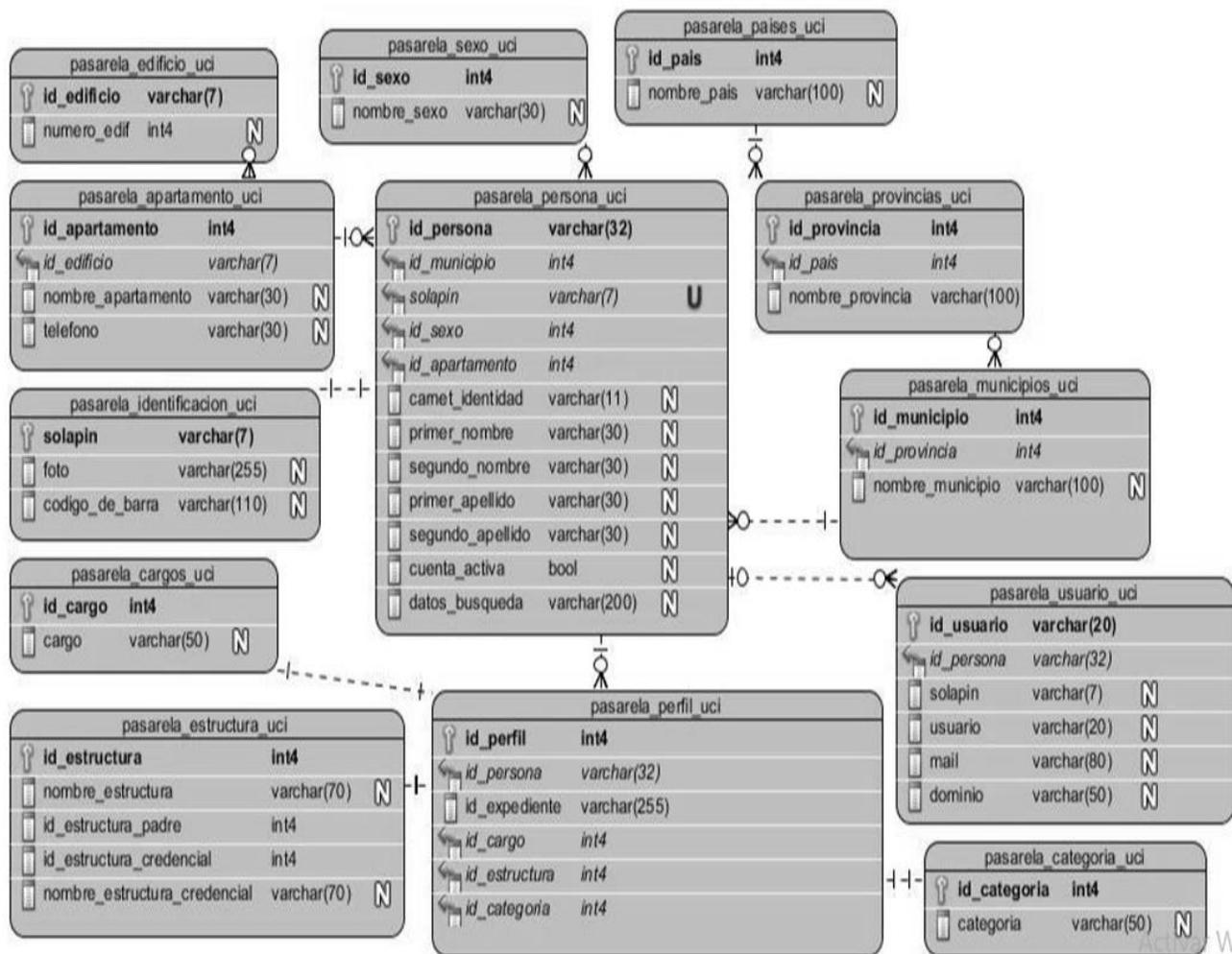


Figura 4: Modelo físico de datos (Fuente: Directorio v2.0 de la Universidad de las Ciencias Informáticas)

## 2.11. Conclusiones parciales

En el presente capítulo se realizó el análisis y diseño de la aplicación, , arrojando los siguientes resultados:

- La descripción del modelo de dominio permitió representar cómo se realiza el proceso de búsqueda de personas en el directorio de la UCI, representados a través de las clases más importantes dentro del contexto del sistema, lo que permite desarrollar con mayor facilidad la solución propuesta.
- El uso de técnicas para la obtención de requisitos proporcionó comprender, identificar y describir los requerimientos funcionales y no funcionales con los que cumple la solución. Se identificaron 12 requisitos funcionales y 5 no funcionales.
- La utilización de patrones de diseño y arquitectura permitieron diseñar una solución robusta, flexible y escalable. Se utilizó el patrón MVC. A partir del análisis, el diseño realizado y los artefactos generados quedan sentadas las bases para la implementación, pruebas y validación de la solución propuesta.

# CAPITULO III - IMPLEMENTACIÓN Y PRUEBAS

## 4.1. Introducción

En este capítulo, se presentarán los diagramas de despliegue, se documentarán las pruebas realizadas al *software* desarrollado, que son un elemento vital para garantizar la calidad del software y representan una revisión final de las especificaciones, del diseño y la codificación.

## 4.2. Diagrama de despliegue

El diagrama de despliegue se utiliza para mostrar la estructura física del sistema, incluyendo las relaciones entre el *hardware* y el *software* que se despliega. A continuación, se muestra en la figura que representa el diagrama de despliegue del sistema.

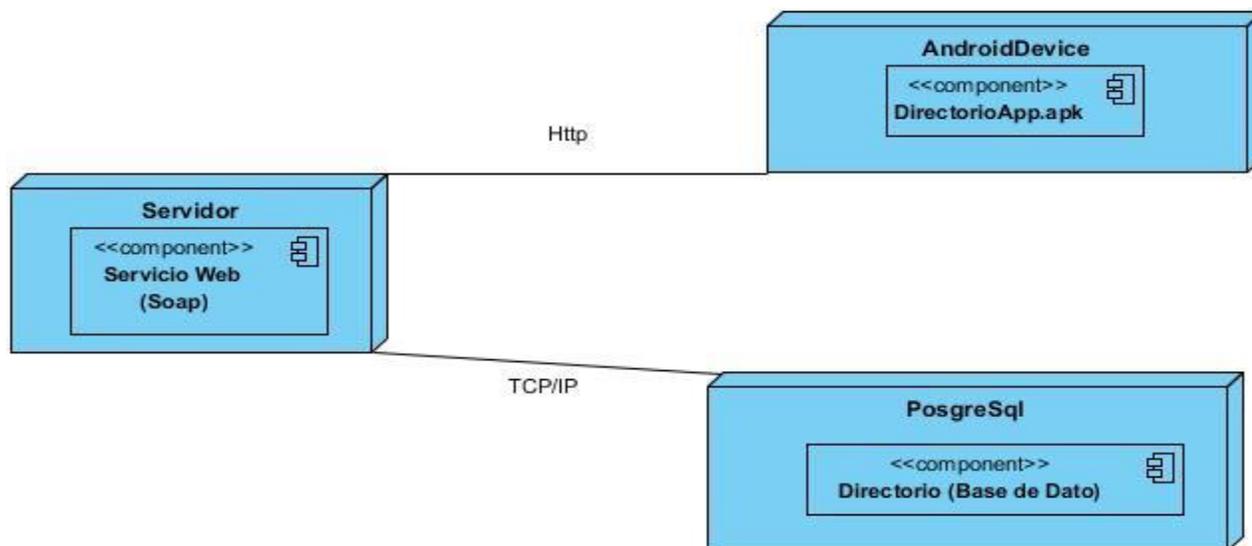


Figura 5- Diagrama de despliegue del sistema

**HTTP:** protocolo de transferencia de hipertexto, es un estándar de red que sigue el esquema petición respuesta entre un cliente y un servidor.

**TCP/IP:** familia de protocolos de Internet. es un conjunto de estándares de red en la que se basa la red global y que permite la transmisión de datos entre redes de computadoras sin importar el tamaño (TCP, 2008).

### 4.3. Pruebas de software

Las pruebas de software son seguramente la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas. Las pruebas de software se definen como una actividad en el cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificada, los resultados se observan y registran y se realiza una evaluación de algún aspecto (Fernández, 2016). Las pruebas de software son partes del ciclo de vida del software, permiten verificar la calidad de un producto buscando defectos o errores a través de pruebas, por otro lado permiten verificar que el software cumple con los requisitos de software definidos para un determinado proyecto.

#### 4.3.1. Métodos de pruebas

Los enfoques de los diseños de casos de pruebas pueden ser de **Caja Negra (funcional)** se centra en las funciones, entradas y salidas. **Caja Blanca (estructural)** este enfoque se centra en la estructura interna del programa.

#### 4.3.2. Pruebas de aceptación

Son las únicas pruebas que son realizadas por los usuarios expertos. Consiste en comprobar si el producto está listo para ser implantado para el uso operativo en el entorno del usuario. **Pruebas alfa** las realiza el usuario en presencia de personal de desarrollo del proyecto haciendo uso de una máquina preparada para las pruebas. Las pruebas de aceptación son dirigidas por el cliente, el usuario comprueba que el sistema hace lo especificado en el contrato. El desarrollo de la prueba fue satisfactorio, quedando como constancia el acta de aceptación del cliente accesible en los Anexos.

#### 4.3.3. Pruebas unitaria

Se aplican a un componente del software (función, una clase,). Estas pruebas las ejecuta el desarrollador, cada vez que va probando fragmentos de código para ver si todo funciona como se desea. Las pruebas unitarias se centran detectar errores en el código. A continuación se muestra un método donde se aplica la técnica de camino básico.

```

public boolean onOptionsItemSelected(MenuItem item) {
    1: int id = item.getItemId();
    2: if (id == R.id.action_settings) {
    3: return true;
    } 4: else if (id == R.id.login) {
        5: Intent i = new Intent(DirectorioActivity.this, LoginMain.class);
        6: i.putExtra("b", "DirectorioActivity");
        7: startActivity(i);
        8: finish();
    }
    9: return super.onOptionsItemSelected(item);
}

```

**Complejidad Ciclomática:**

Para el cálculo de la complejidad ciclomática existen tres fórmulas diferentes, se utilizan a continuación.

$$V(G) = \# \text{ de regiones} = 3$$

$$V(G) = (A - N) + 2 = 10 - 9 + 2 = 3$$

$$V(G) = P + 1 = 3$$

**A:** Cantidad de Aristas **N:** Cantidad de Nodos

**P:** Número de nodos predicado en el grafo

**Caminos lógicos:**

Camino1: 1-9

Camino2: 1-2-3-4-8-9

Camino3: 1-2-3-5-6-7-8-9

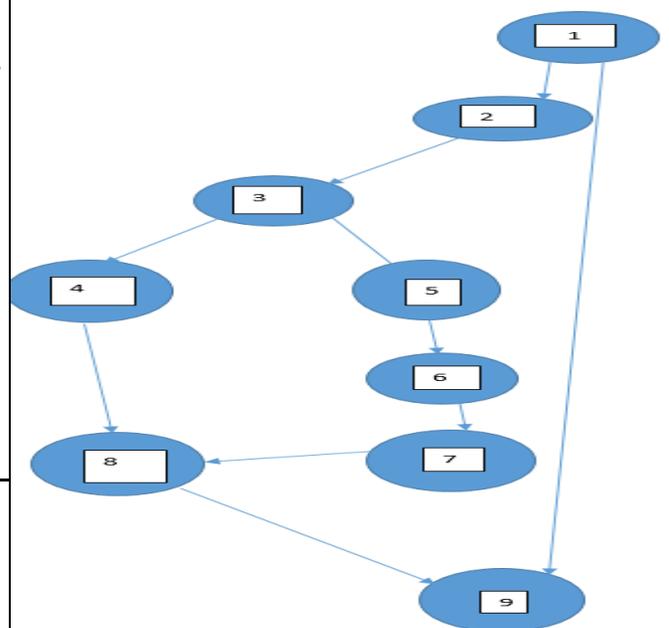


Figura 6- Grafo resultante "Técnica Camino Básico".

Una vez calculada la complejidad ciclomática se define como límite superior 3, lo que indica que hay que realizarle al código tres pruebas, para garantizar que este se ejecute completamente al menos una vez.

### 4.3.4. Pruebas funcionales

Este tipo de prueba se realiza sobre el sistema funcionando, comprobando que cumpla con las especificaciones funcionales de un componente o sistema (lo que vamos a testear, el software o una parte de él).. Para estas pruebas, se utilizan las especificaciones de casos de prueba, los probadores se enfocan en el funcionamiento de la interfaz del sistema a partir del estudio de sus entradas y salidas (Tapia 2014), (Pressman, 2013). Se utiliza el método de caja negra para evaluar. En resumen se realizan las pruebas funcionales para comprobar si el software cumple las funciones esperadas (Ver anexo 1).

#### 4.3.4.1. Diseño de casos de pruebas

Los casos de prueba son un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Cada caso de prueba debe definir el resultado de salida esperado que se comparará con el realmente obtenido. Al generar casos de prueba, se deben incluir tanto datos de entrada válidos y esperados como no válidos e inesperados.

**Técnica particiones o clases de equivalencia** identifican los valores de los datos de entrada. Se definen dos tipos de clases de equivalencia:

- **Clases Válidas:** Entradas válidas al programa.
- **Clases no Válidas:** Valores de entrada erróneos.

Estas clases se pueden representar en una tabla. A continuación, se presenta el diseño del caso de prueba perteneciente al requisito *buscar persona*.

#### **Descripción general**

Permitir a los usuarios buscar a una determinada persona.

#### **Condiciones de ejecución**

Para buscar una persona en la aplicación hay que:

- Estar conectado a la red de la UCI
- Especificar el criterio de búsqueda

## SC Buscar persona

En el diseño de caso de prueba **V** indica Válido, **I** Inválido y **NA** es irrelevante (no es necesario proporcionar un valor del dato). Las variables: Buscador, Nombre en pizarra, Teléfono, Área; significan los valores de entrada de datos para los casos de pruebas en cada escenario; la respuesta del sistema indica su comportamiento y el flujo central, los pasos que se deben ejecutar para completar cada escenario. Para ejecutar la prueba el usuario podía estar o no autenticado en el sistema.

A continuación, se describe un ejemplo de un diseño de caso de prueba que se le realizó a la solución.

Escenario	Descripción	Buscador	Respuesta del sistema	Flujo central
EC 1.1 Buscar datos de forma correcta.	Mediante el escenario se busca una persona correctamente.	V	El sistema realiza la búsqueda y muestra la información solicitada con los datos: Nombre, Teléfono, Apartamento...	1. El usuario debe estar conectado a la red de la UCI. 2. Una vez dentro, selecciona la opción "Personas" y se muestra una interfaz con un buscador que permite buscar por: nombre, usuario, Además de los filtros de búsqueda 3. El usuario introduce los criterios que desea y presiona el botón Buscar. 4. El Sistema muestra los datos en un listado con los datos: Teléfono, Nombre...
		alourenco		
		V		
		eldade		
EC 1.2 Buscar datos incorrectos.	Mediante el escenario se busca una persona agregando datos incorrectos.	I	El sistema muestra un mensaje: "No hay resultados de búsqueda."	1. El usuario debe estar conectado a la red de la UCI. 2. Una vez dentro, selecciona la opción "Personas" y se muestra una interfaz con un buscador que permite buscar por: nombre, usuario,
		"@"		

				Además de los filtros de búsqueda 3. El usuario introduce los criterios de búsqueda incorrectamente y presiona el botón Buscar. 4. El sistema muestra un mensaje: "No hay resultados de búsqueda."
EC 1.3 Buscar datos sin introducir criterios de búsquedas.	Mediante el escenario de información sin introducir datos.	NA	El sistema muestra un mensaje: "Llena por lo menos uno de los criterios de búsqueda."	1. El usuario debe estar conectado a la red de la UCI. 2. Una vez dentro, selecciona la opción "Personas" y se muestra una interfaz con un buscador que permite buscar por: nombre, usuario, Además de los filtros de búsqueda 3. El usuario no introduce criterios de búsqueda y presiona el botón Buscar. 4. El sistema muestra un mensaje: "Llena por lo menos uno de los criterios de búsqueda."

Nombre de la variable	Tipo de datos	Descripción
Buscador	Campo de texto	Permite buscar una persona por los criterios: nombre, usuario

Las pruebas se realizaron en tres iteraciones y se detectaron errores de interfaz, ambigüedad y validación. En la primera iteración se identificaron 10 no conformidades para un total de 7 requisitos revisados, en la segunda 3 no conformidades de 5 requerimientos y en la última no se encontraron errores, cumpliendo con los requisitos funcionales expuestos y se demostró la calidad de la solución implementada.

### 4.3.5. Descripción de los estándares de codificación utilizados

Las convenciones de código son importantes para los programadores por un gran número de razones<sup>3</sup>:

- ✓ El 80% de coste del código de un programa va a su mantenimiento.
- ✓ Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo.
- ✓ Si distribuyes tu código fuente como un producto, necesitas asegurarte de que esta bien hecho y presentado como cualquier otro producto.

Para que funcionen las convenciones, cada persona que escribe software debe seguir la convención. Por lo tanto la propuesta de solución siguió las convenciones:

- **Nombre de fichero:** define la forma de escribir las extensiones y nombre de los ficheros, ej: Tipo de fichero.class.
- **Organización de ficheros** los ficheros de más de 2000 líneas son incómodos y deben ser evitados. La primera línea de los ficheros fuentes Java es la sentencia package. Ej: package com.jg.directorio.
- **Comentarios:** Los comentarios se deben usar para dar descripciones de código y facilitar información adicional que no es legible en el código mismo. Los comentarios deben contener sólo información que es relevante para la lectura y entendimiento del programa. Los comentarios en programas Java se escriben `//`, y `/*...*/`.
- **Declaraciones:** las **variables** deben comenzar por minúscula. Se recomienda una declaración por línea, no poner diferentes tipos en la misma línea.  
Ej: `int nivel;`  
`int tam;`
- **Inicialización:** Intentar inicializar las variables locales donde se declaran. La única razón para no inicializar variable donde se declara es si el valor inicial depende de algunos cálculos que deben ocurrir.

---

<sup>3</sup> Convenciones de Código Java por Scott Hommel. Sun Microsystems Inc.

## 4.4. Conclusiones parciales

En este capítulo se llevó a cabo las actividades de implementación y prueba atendiendo a la metodología definida en el primer capítulo. Las mismas arrojaron los siguientes resultados:

- Se obtuvo el diagrama de despliegue, con el cual se logró mostrar la estructura física del sistema de la aplicación.
- La realización de las pruebas permitió confirmar la correcta implementación de los requisitos definidos y facilitó identificar un conjunto de no conformidades, que fueron resueltas, lo que permitió determinar y asegurar la calidad de la solución para que sea usada por la comunidad de la Universidad de las Ciencias Informáticas.
- La definición de los estándares de codificación a tener en cuenta para la implementación de la solución, permitieron garantizar que el código posea calidad, menos errores y que pueda ser reutilizado por otros desarrolladores

## Conclusiones generales

Con la investigación realizada, el diseño y la implementación de una solución móvil para la búsqueda de información de Personas en la UCI se obtuvieron resultados que permiten arribar a las siguientes conclusiones:

- ❖ La revisión bibliográfica, el estudio de soluciones similares y el análisis de antecedentes encontrados en la literatura, permitieron determinar las principales funcionalidades que debe contener la aplicación móvil para el Directorio de Personas.
- ❖ Se desarrolló una aplicación Android que cumple con los requerimientos del cliente apoyada en las características de los sistemas similares estudiados. Además, se garantiza que la solución desarrollada contribuye con la política de soberanía tecnológica por la que aboga Cuba.
- ❖ Las pruebas realizadas al software permitieron determinar satisfactoriamente la calidad de la solución propuesta, ya que fue sometida a pruebas realizadas al sistema mediante la técnica de caja negra usando partición por equivalencia. Lo que permitió validar del correcto funcionamiento del sistema propuesto se obtuvo una aplicación que cumple con los requisitos y que satisface las necesidades del cliente comprobada por la prueba de aceptación.
- ❖ La aplicación creada no substituye la existente, es más otra forma de acceder el directorio, lo que proporciona mayor usabilidad al Directorio a la hora de consultar o localizar una persona en el campus. La aplicación fue creada para teléfono con sistema operativo Android, sistema dominante para teléfonos.

## Recomendaciones

Para el desarrollo de futuras versiones del Sistema Directorio v2.0 de la Universidad de las Ciencias Informáticas se recomienda:

- ❖ Continuar enriqueciendo el software con la implementación de funcionalidades como los filtros de búsquedas, activar la función favoritos.

## Referencias Bibliográficas

1. JUSTIZ Munder y PEÑALVER Luis. Directorio v2.0 de la Universidad de las Ciencias Informáticas. 2016.
2. TAPIA, Areli. *Modelo de Implementacion*. En: Fundamentos de Ingeniería de Software. [En Línea]. 2014. Disponible en: <https://studylib.es/doc/701559/descripci%C3%B3n-breve>
3. ARVAI, Peter; HALÁCSY, Péter y ADAM Somlai-Fischer. Prezi Inc. *Manual de búsqueda documental y práctica bibliográfica*. [En línea] 2016. [Citado el: 26 de Maio de 2020.] Disponible en: <https://prezi.com/vbb3rziyx1o/introduccion/>.
4. AVILA, Juan Félix. V Jornadas Iberoamericanas de Innovación Educativa en el Ámbito de las TIC y las V Jornadas Iberoamericanas de Innovación Educativa en el Ámbito de las TIC y las TAC Las Palmas de TAC Las Palmas de Gran Canaria, 15 y 16 de noviembre de 2018.
5. BAHIT, Eugenia. *POO y MVC en PHP: El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC*. 2011
6. BATTISTELLA Paulo Eduardo, STANGARLIN DE CAMARGO André y VON WANGENHEIM, Christiane Gresse, SCRUM-Scape: Jogo educacional de Role-Playing Game (RPG) para ensinar SCRUM. 2016, Anais do XXVII Simpósio Brasileiro de Informática na Educação. En V Congresso Brasileiro de Informática na Educação. Brazil.
7. BURNETTE, E., Hello, Android Introducing Google's Mobile Development Platform. 3ª ed. 2010
- CÁCERES, R., ANDRES, Roy, ALEXIS Genoff y PATRICIA Paola. Apps móviles como herramientas de apoyo al aprendizaje matemático informal en Educación Superior. 2013. p.2.
8. CARVALHO, Vidal, AZEVEDO, Ana y ABREU, António. Microsoft Access 2007. 1ª Edición, Portugal: Centro Atlántico. 2007. p. 111
9. CRUZ, Yanet y CARRAZCO, Mirelys. Sistema de directorios para la Universidad de las Ciencias Informáticas. Universidad de las Ciencias Informáticas. 2013.

10. DATE, C. J. Date. Introducción a los Sistemas de Bases de datos. 7ed. México: Pearson Education. 2001, p.24.
11. DELÍA, L.N., GALDÁMEZ, N., THOMAS, P. J., CORBALÁN, L. C., y PESADO, P. M Análisis experimental de desarrollo de aplicaciones móviles multiplataforma. 2014.
12. DIRECTORIO - CÁMARA DE COMERCIO AMERICANA DEL PERÚ. [En línea]. Perú. 2016. [Citado el: 03 de Junio de 2020.] Disponible en: <https://amcham.org.pe/sala-de-prensa/publicaciones/directorio>
13. DIRECTORIO DE LA UNIVERSIDAD DE LA REPÚBLICA. [En línea], Uruguay. 2015 - [Citado el: 03 de Junio de 2020.] Disponible en: <http://udelar.edu.uy/directorio>
14. FERNÁNDEZ, Luis. *Un Sondeo sobre la práctica actual de pruebas de software en Español*. Reicis Revista Española de Innovación. Calidad e Ingeniería del Software. pp.43-54. vol 1, nº002, Madrid, España. 2016.
15. FORTES, X. P., AZEVEDO, de Sá Neves. Desenvolvimento de aplicativo android para vagas de empregos utilizando a metodologia ágil scrum, in Revista Científica da FASETE. 2018. p. 243.
16. HERNÁNDEZ, Carlos. Aplicación móvil con Android para el seguimiento de toma de medicamentos usando Kotlin. [En línea]. 2019. Disponible en: <https://hdl.handle.net/20.500.12371/4714>
17. JORDAN, L., GREYLING, Pieter, Pratical Android Projects. 2011.
18. KLAUSEN, P., Java 16: Mobile Phones and Android Software Development. 1ª ed. 2018. p. 9.
19. LARMAN, Craig. UML y Patrones: Una introducción al análisis y diseño drientado a objetos y al proceso unificado. 2ªEdición. 2015.
20. MARIÑO, Sonia y ALFONZO, Pedro. Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación, in Scientia Et Technica. 2014. p. 413-418.
21. MARTÍNEZ, Francisco Javier. Guía de Construcción de Software en Java con Patrones de Diseño. Universidad de Oviedo. 2018.
22. MONTAN, J.S. Avaliação de Plataformas Híbridas para Desenvolvimento de Aplicações para Dispositivos Móveis. 2017. v.2, p.116-125.

23. NEGRÍN, Laritza y MARTÍNEZ, Esteban. *Herramientas para la búsqueda y recuperación de información en Internet*. Grupo de Vigilancia Tecnológica Centro de Información Científico-Técnica Universidad de las Ciencias Informáticas. 2016.
24. Oracle. (2019). Java Development. Disponible en <http://www.oracle.com/technetwork/java/javase/10-0-1-relnotes-4308875.htm>
25. PAPAOGLOU, Mike. SIMPLE Object Access Protocol (SOAP). [En línea]. 2008. Disponible en: [https://www.cs.colorado.edu/~kena/classes/7818/f08/lectures/lecture\\_3\\_soap.pdf](https://www.cs.colorado.edu/~kena/classes/7818/f08/lectures/lecture_3_soap.pdf)
26. PRESSMAN, Roger S. *Ingeniería de Software, un enfoque práctico*. Séptima edición. McGrawHill. México. 2013. p. 585
27. ROSÁRIO, Jean Carlos. WEB SERVICES (SOAP X REST). 2012 EXTENSIBLE Markup Language (XML). Disponible en: <http://www.w3.org/XML/>.
28. SCHWABER, Ken y SUTHERLAND, Jeff. *O guia definitivo para o Scrum: As regras do Jogo*. 2017.
29. SEGUNDO M. A. , HERNAN, José A. A. y JOAQUIN. *Diseño e Implementación de un Servicio Web (Web Services) Para la Búsqueda de Hoteles en la Zona Costanera del Departamento de Córdoba*. 2015
30. SOMMERVILLE, Ian. *Software Engineering*. 9ª Edición. Nueva York. 2011. Editorial: Pearson Education. p. 773
31. SUNARP. [En línea], 2015. [Citado el: 03 de Junio de 2020.] . Disponible en: <https://www.sunarp.gob.pe/bus-personas-juridicas.asp>
32. TCP, Protocolo. CCM. [En línea] 2008. Disponible en: <http://es.ccm.net/contents/281-protocolo-tcp>.
33. VÉLEZ, S., ZAPATA, J. A. y HENAO, A, *Gestión de Proyectos: Origen, instituciones, metodologías, estándares y certificaciones*, in *Entre Ciencia e Ingeniería*. 2018. p. 68-76.
34. VIQUE, R.R. *Métodos para el desarrollo de Aplicaciones Móviles*. 2013.
35. Os dispositivos eletrônicos mais populares entre o público - FuniBlogs - FUNIBER. [En línea] 2015. [Citado el: 22 de noviembre de 2019.] Disponible en: <https://blogs.funiber.org/pt/empresas-e-rh/2015/10/02/funiber-os-dispositivos-eletronicos-mais-populares-entre-o-publico>

36. Deloitte US | Audit, Consulting, Advisory, and Tax Services. [Citado el: 22 de noviembre de 2019.] Disponible en: <https://www2.deloitte.com/us/en.html>
37. Cuba-Llega-A-Cinco-Millones-De-Lineas-Moviles-Activas. [Citado el: 22 de noviembre de 2019.] Disponible en: <http://www.cubadebate.cu/noticias/2018/04/11/cuba-llega-a-cinco-millones-de-lineas-moviles-activas/>
38. DE ESPAÑA, B. N.. Directorios : definición, características y contenido. Biblioteca Nacional de España. [En línea] 2011. [Citado el: 11 de septiembre de 2019.] Disponible en: [http://www.bne.es/es/Micrositios/Guias/ObrasReferencia/AnuariosDirectorios/Directorios/?fbclid=IwAR0XU\\_zLF\\_Vmm8POTj1tE2HI9t5LBoXkGRKcUn4RmtXnNDfviG714kTTyvM](http://www.bne.es/es/Micrositios/Guias/ObrasReferencia/AnuariosDirectorios/Directorios/?fbclid=IwAR0XU_zLF_Vmm8POTj1tE2HI9t5LBoXkGRKcUn4RmtXnNDfviG714kTTyvM)
39. R. J. C. / J., M. Las 19 mejores aplicaciones del año. EL PAÍS. [En línea] 2013. Disponible en: [https://elpais.com/tecnologia/2013/12/16/actualidad/1387205308\\_779223.html](https://elpais.com/tecnologia/2013/12/16/actualidad/1387205308_779223.html)
40. Gartner. Disponible en: <http://www.gartner.com/newsroom/id/2592315>
41. IDC - Smartphone Market Share - OS. [Citado el: 10 de noviembre de 2019.] Disponible en: <https://www.idc.com/promo/smartphone-market-share/os>.
42. davidortinau. Mobile software development lifecycle - Xamarin. [Citado el: 06 de diciembre de 2019.] Disponible en: <https://docs.microsoft.com/pt-pt/xamarin/cross-platform/get-started/introduction-to-mobile-sdlc>
43. History: The Agile Manifesto. [Citado el: 09 de septiembre de 2019.] Disponible en: <http://agilemanifesto.org/history.html>
44. PricewaterhouseCoopers. Agile Project Delivery Confidence. [Citado el: 16 de Junio de 2020.] Disponible en: <https://www.pwc.com/gx/en/services/audit-assurance/risk-assurance/agile-project-delivery-confidence.html>
45. What is Scrum?. [Citado el: 16 de Junio de 2020.] Disponible en: <https://www.scrum.org/resources/what-is-scrum>
46. ABELLÁN, E. Metodología Scrum: qué es y cómo funciona. WeAreMarketing.com. . [En línea] 2020. [Citado el: 16 de junio de 2020.] Disponible en: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>

47. PAINKA, Marcelo. Marchi, Késsia. *Utilização das Metodologias Ágeis Xp E Scrum para o Desenvolvimento Rápido de Aplicações*, Brazil, 2013.
48. Scrum Alliance. The 2015 State of Scrum Report. Disponible en: <https://www.scrumalliance.org/ScrumRedesignDEVSite/media/scrumalliancemedi/files%252520and%252520pdfs/state%252520of%252520scrum/scrum-alliance-state-of-scrum-2015.pdf>
49. Home - Pencil Project. [Citado el: 20 de febrero de 2020.]Disponible en: <https://pencil.evolus.vn>

## **Glosario de términos**

Aquí se recogen y definen los conceptos más relevantes y críticos para la investigación.

IOs: Sistema operativo para dispositivos móviles de Iphone.

SDK: Siglas de software development kit. Herramientas necesarias para poder realizar el desarrollo de aplicaciones en una plataforma.

UI (User Interface – Interfaz de Usuario): responsable de la interacción del usuario y software.

API: Application Programming Interface

SQLite: Biblioteca en lenguaje C que implementa una base de datos SQL

FRAMEWORK de aplicaciones es marco de trabajo que permite realizar aplicaciones de manera más sencilla, ordenada y mantenible.

WSDL ( Web Services Description Language): descripción de servicios.

UDDI (Universal Discovery Description and Integration): hallazgo de servicios.

CMS (Content Management System).

## **Anexos**

### **1. Lista de Chequeo de Usabilidad de la aplicación móvil para el Directorio UCI**

El objetivo general de la lista de chequeo es evaluar las especificaciones de la aplicación desarrollada.

**<Directorio Apk>**

**<V1.0>**

Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.

El objetivo general de la lista de chequeo es evaluar y verificar las especificaciones de la propuesta de solución.

Forma de Uso:

Peso: Define si el indicador a evaluar es crítico o no.

Evaluación (Eval): Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal (cuando la respuesta al indicador sea “No”) y 0 en caso que elemento revisado no presente errores (cuando la respuesta al indicador sea “Sí”).

<b>Peso</b>	<b>Indicadores a Evaluar</b>	<b>Eval</b>	<b>Criterio de Éxito</b>
Crítico	1. ¿Cada pantalla empieza con un título que describe su contenido?	0	Éxito: si
Crítico	2. ¿La terminología del menú es constante en todo el sitio?	0	Éxito: si
Crítico	3. ¿Los iconos que aparecen se identifican claramente con lo que representan?	0	Éxito: si
Crítico	4. ¿El lenguaje es simple, con un tono adecuado?	0	Éxito: si
Crítico	5. ¿La información que se presenta en la aplicación es fácil de entender y memorizar?	0	Éxito: si
	6. ¿La información está organizada con categorías lógicas, fácilmente memorizables para el usuario?	0	Éxito: si
	7. ¿El orden de los menús es por frecuencia de utilización?	0	Éxito: si
Crítico	8. ¿El lenguaje está más cercano al utilizado por el usuario que al informático o técnico?	0	Éxito: si

	9. ¿El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: utilizar “Buscar” en vez de “OK”...)?	0	Éxito: si
	10. ¿Los íconos son visualmente y conceptualmente distintos pero mantienen una armonía?	0	Éxito: si
	11. ¿Existe suficiente espacio entre los elementos de acción (links, botones, etc) para prevenir que el usuario haga click en el elemento incorrecto?	0	Éxito: si
	12. ¿Las cajas de texto en los formularios tienen el tamaño adecuado para el dato que se debe introducir?	0	Éxito: si
	13. ¿La aplicación permite completar tareas introduciendo solamente los datos obligatorios?	0	Éxito: si
	14. ¿El motor de búsqueda maneja correctamente (No arroja ningún resultado) las búsquedas	0	Éxito: si

	vacías (cuando no se introduce nada)?		
	15. ¿La página de resultados de una búsqueda no muestra resultados duplicados (ni duplicados reales ni duplicados muy parecidos)?	0	Éxito: si
	16. ¿La caja de búsqueda es suficientemente grande para manejar la longitud de las consultas más comunes?	0	Éxito: si
Crítico	17. ¿Se mantiene una navegación consistente y coherente en todas las pantallas?	0	Éxito: si
Crítico	18. ¿Se dan indicaciones para completar campos problemáticos?	0	Éxito: si
Crítico	19. ¿Puede un usuario sin experiencia utilizar la aplicación?	0	Éxito: si
Crítico	20. ¿Se mantiene una tipografía coherente en toda la aplicación?	0	Éxito: si
Crítico	21. ¿Los tipos y tamaños de letra son legibles y distinguibles?	0	Éxito: si

Crítico	22. ¿El menú de navegación aparece en un lugar destacado?	0	Éxito: si
Crítico	23. ¿El uso de los colores es moderado?	0	Éxito: si
	24. ¿Hay suficiente contraste entre el fondo el texto?	0	Éxito: si
Crítico	25. ¿El mensaje de error es entendible e informa cuales son las acciones correctoras?	0	Éxito: si
Crítico	26. ¿El mensaje de error permite volver a la situación anterior?	1	Éxito: si
Crítico	27. ¿La ayuda ofrecida está ligada a la tarea o sección en curso?	0	Éxito: si
Crítico	28. ¿El sitio está diseñado para necesitar el mínimo de ayuda y de instrucciones?	1	Éxito: si
Crítico	29. ¿Ha identificado errores ortográficos?	1	Éxito: no

**Evaluación:** \_\_\_\_\_ **Bien** \_\_\_\_\_

Nombre y Apellido del Evaluador: \_\_\_\_\_ Armando e Costa \_\_\_\_\_

## 2. Entrevista

Para recopilar la información se utilizó fundamentalmente: la entrevista que es la mayor fuente de información del analista, permitió determinar las necesidades del sistema, la adquisición por parte del equipo de desarrollo de la información necesaria para desarrollar un producto de calidad.

1. ¿Cuál es el objetivo del directorio actual?
2. ¿En que tecnologías se ha desarrollado el directorio.uci.cu?
3. ¿Cuáles son las limitaciones que presenta el directorio actual?
4. ¿Cuáles son las características fundamentales que se espera que debe realiza la nueva aplicación?
5. ¿De dónde se puede obtener los datos generales de los ciudadanos de la uci?
6. ¿Qué datos manejan el servicio web?
7. ¿Quién puede acceder al directorio?

### **Entrevista realizada al:**

Director del Centro de Desarrollo de Tecnologías para la Formación: Alexander Mompíe