



Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Ingeniería
Carrera Ingeniería Informática

TÍTULO DEL TRABAJO DE DIPLOMA

**Aplicación móvil para la Gestión de Solicitud de Descarga en
la Universidad de Cienfuegos.**

**Autor:
Reinol de León Viamonte.**

**Tutor:
Juan Carlos Oropesa Hernández.**

**Consultante:
Alfredo Rafael Espinosa Palenque.**

**Cienfuegos, Cuba
Curso 2021**

Declaración de autoría

Declaro que soy el único autor (Declaramos que somos los únicos autores) de este trabajo y autorizo (autorizamos) al <Nombre del Departamento> de (de la, del) <Nombre de la Entidad> y al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmo (firmamos) la presente a los ____ días del mes de ____ del ____.

(Si procede)

Nombre completo del primer autor

Nombre completo del segundo autor

(Si procede)

Nombre completo del primer tutor

Nombre completo del segundo tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Tutor

Firma Tutor

Firma ICT

Firma Vicedecano

Opinión del usuario

El Trabajo de Diploma, titulado <Título del Trabajo de Diploma>, fue realizado en nuestra entidad <Nombre de la Entidad que utilizará el sistema>. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado nos satisface:

- Totalmente
- Parcialmente en un ____ %

Los resultados de este Trabajo de Diploma le reportan a nuestra entidad los beneficios siguientes (cuantificar):

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a <valor> MN y/o <valor> CUC. (Este valor debe ser REAL, no indica lo que se reportará, sino lo que reporta a la entidad. Puede desglosarse por conceptos, tales como: cuánto cuesta un software análogo en el mercado internacional, valor de los materiales que se ahorran por la existencia del software, valor anual del (de los) salario(s) equivalente al tiempo que se ahorra por la existencia del software).

Y para que así conste, se firma la presente a los __ días del mes de ____ del año ____.

Nombre del representante de la entidad

Cargo

Firma

Cuño

Opinión del tutor

Título: <Título del trabajo de diploma>

Autor(es): <Nombres y apellidos del autor o los autores>

El(Los) tutor(es) del presente Trabajo de Diploma considera(mos) que durante su ejecución el(los) estudiante(s) mostró(arón) las cualidades que a continuación se detallan.

<El tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes: Independencia, Originalidad, Creatividad, Laboriosidad y Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios)>.

Por todo lo anteriormente expresado considero que el estudiante está (no) apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <2 – Desaprobado, 3 – Aprobado, 4 – Bien, 5 – Excelente>.

<Si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

Y para que así conste, se firma la presente a los ___ días del mes de _____ del año ____.

(Si procede)

Nombre completo del primer tutor
<Grado científico, Categoría docente
y/o investigativa>

Nombre completo del segundo tutor
<Grado científico, Categoría docente
y/o investigativa>

Fecha: _____

Agradecimientos

Agradecer a todas las personas que me apoyaron en todo este bonito proceso, a mi papa mi mama mi abuela y a todos mis compañeros de la beca y de aula.

Dedicatoria

A mi familia.

Resumen

El presente trabajo se desarrolló en la Universidad de Cienfuegos. El desarrollo de la aplicación móvil para el gestionamiento del proceso de solicitudes de descarga surge como necesidad debido a que, aunque la institución cuenta con el Sistema Automático de Descarga(SAD) para realizar este proceso, su accesibilidad se torna limitada al uso de una computadora conectada a la red o también se puede acceder desde un teléfono inteligente a través de un navegador, pero las vistas del SAD no siempre se acoplan a las pantallas de los dispositivos móviles. La aplicación móvil garantiza una mejor accesibilidad al sistema ya que el usuario podrá disponer de ella desde su propio dispositivo móvil, además de facilitar el acceso al sistema también lo agiliza debido a que la aplicación cuenta con una interfaz similar a la del sitio esto trae como consecuencia que no sea necesario cargar las vistas del SAD. La aplicación se desarrolló para dispositivos móviles con sistema operativo Android ya que este es más popular entre los usuarios, esto garantiza su despliegue en la Universidad de Cienfuegos y la accesibilidad para un mayor número de usuarios. Para ello la metodología de desarrollo de software seguida fue Rational Unified Process y el lenguaje de modelado Unified Modeling Language. Como herramienta principal se estableció: Visual Paradigm para UML 8.0 Enterprise Edition para la modelación los requerimientos. Como principal resultado se obtiene una aplicación móvil Android capaz de gestionar el proceso de solicitud de descargas.

Palabras clave: Aplicación móvil, accesibilidad, dispositivos móviles, usuarios , sistema operativo Android.

Summary

The present work was developed at the University of Cienfuegos. The development of the mobile application to manage the download request process arises as a necessity because, although the institution has the Automatic Download System (SAD) to carry out this process, its accessibility is limited to the use of a computer. connected to the network or it can also be accessed from a smartphone through a browser, but the SAD views do not always match the screens of mobile devices. The mobile application guarantees better accessibility to the system since the user can have it from their own mobile device, In addition to facilitating access to the system, it also streamlines it because the application has an interface similar to that of the site, this means that it is not necessary to load the SAD views. The application was developed for mobile devices with Android operating system since it is more popular among users, this guarantees its deployment at the University of Cienfuegos and accessibility for a greater number of users. For this, the software development methodology followed was Rational Unified Process and the Unified Modeling Language modeling language. The main tool was: Visual Paradigm for UML 8.0 Enterprise Edition for modeling the requirements. The main result is an Android mobile application capable of managing the download request process.

Keywords: mobile application, accessibility, mobile devices, users, Android operating system.

Índice

Summary	I
Introducción.....	2
1 – Fundamentos teóricos.....	7
1.1 – Introducción	7
1.2 – Descripción del dominio del problema.....	7
1.3 – Descripción del objeto de estudio	8
1.3.1 - Objetivos estratégicos de la organización	8
1.3.2 - Flujo actual de los procesos y análisis crítico de la ejecución de estos	9
1.4 – Descripción de los sistemas existentes	10
1.5 – Tendencias, metodologías y/o tecnologías actuales	12
Metodologías de desarrollo de software	12
1.5.1 - Proceso Unificado de Desarrollo de Software (RUP)	14
1.6 - Tecnologías y lenguajes	14
1.6.1- Arquitectura del sistema	14
1.6.2- Lenguajes	15
1.7- Herramientas utilizadas	16
1.8– Conclusiones.....	18
2 – Análisis y diseño de la propuesta de solución	19
2.1 – Introducción	19
2.2.1 – Definición de las entidades y los conceptos principales	19
2.2.2 – Reglas del negocio a considerar.....	21
2.2.3 – Representación del modelo del dominio.....	22
2.3 – Requisitos.....	22
2.3.1 – Descripción del sistema propuesto	22
2.3.2 – Concepción general del sistema.....	22
2.3.3 – Requerimientos funcionales.....	22
2.3.4 – Requerimientos no funcionales	23
2.4 – Modelo de casos de uso del sistema	24
2.4.1– Actores del sistema	25
2.4.2 – Diagramas de casos de uso del sistema.....	25
2.4.3 – Descripción de los casos de uso del sistema.....	25
2.5 – Construcción de la solución propuesta	29
2.5.1 – Diagrama de clases del diseño	29
2.5.2 – Diagrama de implementación	30
2.5.3 – Principios de diseño	31
2.5.4 – Estándares en la interfaz de la aplicación	31
2.5.5 – Tratamiento de errores	31
2.5.6 – Concepción General de la ayuda.....	32
2.6 – Conclusiones.....	32
3 – Estudio de factibilidad y validación de la solución	33
3.1 – Introducción	33
3.2 – Estudio de factibilidad	33

3.2.1 – Planificación por casos de usos.....	33
3.2.2 – Determinación de los costos	34
3.2.3– Beneficios tangibles e intangibles.....	41
3.2.4 – Análisis de costos y beneficios	42
3.3 – Validación de la solución.....	42
3.3.1- “Autenticar”.....	43
3.3.2 - “crear solicitud de descarga”	43
3.3.3- “Listar descargas”	44
3.3.4- “Aprobar descarga”	45
3.3.5“Denegar descarga”	45
3.3.6“Cambiar prioridad a la solicitud de descarga”	46
3.4 – Conclusiones	46
Conclusiones	47
Recomendaciones	48
Referencias bibliográficas	49
Glosario de términos	52
Anexos.....	53
Anexo 1 – prototipo autenticarece.....	53
Anexo 2 – prototipo crear solicitud de descarga.....	53
Anexo 3 – prototipo buscar solicitud de descarga. }	53
Anexo 4 – prototipo listar solicitud de descarga.....	54
Anexo 5 – prototipo ordenar solicitud de descarga.	54
Anexo 6 – prototipo modificar solicitud de descarga.....	54
Anexo 7 – prototipo eliminar solicitud de descarga.....	54

Índice de tablas

Tabla 1. Definición de actores del sistema a automatizar	25
Tabla 2. Descripción del caso de uso de sistema < Autenticar >	26
Tabla 3. Descripción del caso de uso de sistema < Crear solicitud de descarga >	26
Tabla 4. Descripción del caso de uso de sistema < Buscar solicitud de descarga >	27
Tabla 5. Descripción del caso de uso de sistema < Listar solicitudes de descarga	27
Tabla 6. Descripción del caso de uso de sistema < Ordenar solicitudes de descarga >	28
Tabla 7. Descripción del caso de uso de sistema < Modificar solicitud de descarga >	28
Tabla 8. Descripción del caso de uso de sistema < Eliminar solicitud de descarga >	29
Tabla 9. Clasificación de los actores atendiendo el factor de peso	34
Tabla 10. Factor de peso de los casos de uso del sistema.	35
Tabla 11. Cálculo del factor de peso de los casos de uso.	36
Tabla 12. Significado, peso y valor de cada factor de complejidad técnica.	37
Tabla 13. Significado, peso y valor de cada factor ambiente	39
Tabla 14. Cálculo el esfuerzo para cada actividad.....	40

Índice de figuras

Figura 1. Diagramas de clases del modelo de objetos del dominio	22
Figura 2. Diagrama del caso de uso de la aplicación móvil.....	25
Figura 3. Diagramas de clases.....	29
Figura 4. Diagramas de actividades en android	30
Figura 5. Diagrama de implementación	30

Introducción

La descarga segura de contenidos es un tema importante en el mundo móvil porque ha generado bastante dinero y porque se considera como una de las mayores fuentes de ingresos futuros. Sobre este tema han ido apareciendo tecnologías propietarias que finalmente han desembocado en un estándar común, tanto de los procedimientos de descarga como de los métodos de protección de los contenidos.[1]

Un gestor de descargas es una aplicación que se encarga de gestionar, optimizar y acelerar la descarga de archivos. No estamos hablando de las descargas que hacemos desde Google Play o tareas de actualización que se realizan en segundo plano, sino de todos esos archivos PDF, fotos o videos que bajamos desde internet a nuestro terminal Android.[2]

Con un buen gestor podemos manejar perfectamente varias descargas simultáneas sin problema, retomar descargas fallidas, evitar cortes, mejorar la velocidad de descarga, definir un número máximo de threads por descarga, y en definitiva, potenciar la descarga de archivos en nuestro dispositivo.[2]

El Departamento de Redes de la Universidad de Cienfuegos brinda varios servicios a sus usuarios, entre ellos se encuentran el de Correo Electrónico, el de Internet y el de Acceso Telefónico. También cuenta con una página web oficial donde se exponen noticias actualizadas de índole local, nacional e internacional. Dicho departamento, cuenta con un nodo para cada una de las áreas organizativas de la Universidad, destinado a atender a los usuarios de esa área. En cada nodo hay un Administrador de Red que es el responsable de atender a los usuarios.[3]

La Universidad de Cienfuegos (Ucf) cuenta con un Sistema Automático de Descarga (SAD), en el cual el usuario de la Ucf puede acceder mediante un portátil conectado a la red wifi o cableada de la Universidad o desde cualquier PC de uno de los laboratorios informáticos de la misma. Los usuarios pueden realizar la solicitud de descarga de un archivo, el administrador de red recibe la solicitud y la acepta o la deniega, en caso de

ser aceptada, se le informa al usuario cuando se haya completado la descarga correspondiente a su solicitud.

Para poder acceder al SAD se debe dirigir hasta un laboratorio, usar una PC portátil o se pudiera ingresar al sitio mediante el navegador de un teléfono móvil previamente conectado a una red de la Ucf, cabe destacar que este proceso puede tardar un poco ya que se tendrá que descargar las vistas del SAD y estas no siempre se adaptan correctamente a las pantallas de los teléfonos móviles, esto trae como consecuencia que el usuario no pueda disfrutar de una experiencia placentera en el sitio, por tanto esto conlleva a que los usuarios no usen los servicios que brinda el Sistema Automático de Descargas con mucha frecuencia.

Debido a lo anterior planteado surge la necesidad de hacer más viable el acceso y el uso del SAD para los usuarios de la Universidad de Cienfuegos. Esta investigación toma como objetivo principal dar solución a los problemas antes expuestos y obtener un producto que satisfaga a los usuarios de la Ucf.

El desarrollo de una aplicación móvil con sistema Operativo Android garantizará un mejor acceso al sitio ya que el usuario podrá contar con una aplicación en su dispositivo móvil con todo lo necesario para realizar el procedimiento de gestión de solicitud de descarga. Esto agilizará y facilitará todo el proceso tanto para el usuario como para el administrador ya que tendrán acceso al sitio desde cualquier red inalámbrica de la Ucf de manera rápida.

Después de realizar una investigación y entrevistarse con usuarios de la Ucf se eligió realizar la aplicación para dispositivo Android ya que la gran mayoría de usuarios utilizan teléfonos con sistema operativo Android.

Problema

¿Cómo facilitar y agilizar el proceso de Solicitud de Descargas en el Departamento de Redes de la Universidad de Cienfuegos?

Idea a defender

El desarrollo de una aplicación móvil Android facilitará y agilizará el acceso al Sistema Automático de Descarga, además proporcionará una mejor experiencia en la red agilizando el proceso de gestión de solicitudes de descarga.

Objeto de estudio

El proceso de solicitudes de descargas en el Sistema Automático de Descargas en la Universidad de Cienfuegos.

Campo de acción

El proceso de solicitud y procesamiento de descargas en el Departamento de Redes de la Universidad de Cienfuegos.

Objetivo general

Desarrollo de una aplicación Android que facilite el acceso al Sistema Automático de Descarga de la Universidad de Cienfuegos, y agilice el proceso de gestión de las solicitudes de descargas.

Objetivos específicos

- Analizar el proceso de solicitud de descargas en la Universidad de Cienfuegos.
- Diseñar una aplicación móvil Android que mejore el seguimiento y procesamiento del proceso de solicitud de las descargas en la Universidad de Cienfuegos.
- Implementar una aplicación Android.
- Validar la aplicación.
- Desplegar la aplicación en la Universidad de Cienfuegos.

Tareas a desarrollar

- Entrevista a los administradores de red para recopilar información sobre el proceso de solicitud de descargas.
 - Entrevista con los usuarios para saber que sistema operativo móvil es más común en la Universidad de Cienfuegos.
 - Búsqueda de aplicaciones móviles actuales vinculados al objeto de estudio.
 - Estudio y selección de tecnologías, lenguajes, herramientas y metodologías actuales.
-
- Planificación de las etapas de desarrollo de la aplicación.
 - Desarrollo de la aplicación según las etapas planificadas.
 - Confección de la documentación de la aplicación.
 - Determinación de la factibilidad la aplicación propuesta.
 - Aplicación de los casos de pruebas funcionales.

El aporte práctico:

- Contar con una aplicación móvil para facilitar a los usuarios de la Ucf el acceso al SAD.
- Los usuarios podrán acceder al sistema se descargas desde cualquier red de la Ucf sin necesidad de una computadora portátil.
- El administrador no deberá estar pendiente a su computadora para gestionar las solicitudes de descargas.

El aporte social:

- Se contribuye a la mejora del proceso de gestión de solicitudes de descarga.
- Brindarle accesibilidad a un mayor número de usuarios al SAD.

Capítulo 1 Se abordan los aspectos teóricos para fundamentar la investigación, se analiza el objeto de estudio, los sistemas existentes vinculados a la investigación, así como las tendencias, tecnologías y metodologías actuales a emplear.

Capítulo 2 Se presenta la descripción actual del dominio según la metodología RUP, se identifican sus procesos, se enuncian las reglas a tener en cuenta, se definen los actores y trabajadores del mismo, son descritos los casos de uso y se muestran los diagramas de casos de uso para una mejor comprensión. Se definen los requisitos funcionales y no funcionales, el modelo del sistema, los actores y casos de uso del sistema y se obtiene los diagramas de casos de uso del sistema.

Capítulo 3 En este capítulo se muestra la factibilidad del proyecto, mediante la variante de puntos de casos de usos. Se presentan las pruebas funcionales realizadas para la validación del sistema.

1 – Fundamentos teóricos

1.1 – Introducción

En el presente capítulo se abordan los principales conceptos asociados al dominio del problema, se realiza un análisis crítico del flujo actual de los procesos que son objeto de automatización y se expone un análisis de algunos de los sistemas informáticos existentes para el control y seguimiento de las solicitudes de descargas. Se describen las principales metodologías, tecnologías, lenguajes y herramientas actuales seleccionadas para realizar este trabajo.

1.2 – Descripción del dominio del problema

Para lograr una correcta comprensión del problema resulta necesario conocer algunos conceptos relacionados con el objeto de estudio:

API: La interfaz de programación de aplicaciones es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas de programación. Una de las principales funciones de las API es poder facilitarles el trabajo a los desarrolladores y ahorrarles tiempo y dinero. [4]

Archivo: Conjunto organizado de unidades de información (bits) almacenados en un dispositivo. Se les denomina de esa manera como metáfora a partir de los archivos tradicionales de oficina, escritos en papel, ya que vendrían a ser su equivalente digital. Pueden tener diferentes funciones como simplemente contener información de manera ordenada, como los archivos de texto, y permitir el acceso a ella por parte de programas determinados, hasta archivos ejecutables que desencadenan cierta secuencia de acciones (y de otros archivos) que tienen como resultado una acción concreta. [5]

Complemento: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión. Permiten

que los desarrolladores externos colaboren con la aplicación principal extendiendo sus funciones, reducir el tamaño de la aplicación y separar el código fuente de la aplicación a causa de la incompatibilidad de las licencias de software.[6]

URL (Uniform Resource Locator): Secuencia estándar de caracteres que identifica y permite localizar y recuperar una información determinada en la Internet. Eso que escribimos en la casilla del navegador o explorador para visitar una página web determinada, comúnmente referido como “dirección”, es justamente su URL. [7]

Servidor Web. Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.[8]

Android. Es un Sistema Operativo además de una plataforma de Software basada en el núcleo de Linux. Diseñada en un principio para dispositivos móviles. Android permite controlar dispositivos por medio de bibliotecas desarrolladas o adaptadas por Google mediante el lenguaje de programación Java.[9]

Aplicación Móvil Las aplicaciones móviles son programas diseñados para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades.[10]

1.3 – Descripción del objeto de estudio

1.3.1 - Objetivos estratégicos de la organización

MISIÓN

La Universidad de Cienfuegos “Carlos Rafael Rodríguez”, dedicada a la formación integral continua de profesionales competentes y comprometidos con la Patria

Socialista contribuye mediante el conocimiento y la innovación a la dinamización del desarrollo socioeconómico del territorio de Cienfuegos y de la sociedad cubana.[3], [11]

VISIÓN

Somos una Universidad de Excelencia que:

- Promueve una cultura general integral acorde con los valores de la sociedad cubana.
- Ofrece profesionales líderes comprometidos con la transformación de la sociedad cubana.
- Exhibe una alta visibilidad de sus resultados científicos y de innovación.
- Impacta en el desarrollo económico y social del territorio y la sociedad, satisfaciendo las necesidades de superación profesional y la implementación de resultados de investigación y la innovación.[3] [11]

1.3.2 - Flujo actual de los procesos y análisis crítico de la ejecución de estos

Cualquier usuario de la red de la Universidad de Cienfuegos puede solicitar una descarga en el nodo destinado para su área. El Administrador de Red de dicho nodo es el encargado de gestionar las solicitudes de usuarios interesados en descargar cualquier archivo, video de YouTube, o directorio de Internet y verificar que sean para uso docente o investigativo. Dichas solicitudes se hacen personalmente, también pueden solicitarla mediante el Sistema Automático de Descargas de la universidad de Cienfuegos SAD o se pueden enviar mediante correo electrónico con asunto: "Solicitud de Descarga", aunque pudiera contener cualquier variante que el administrador entienda que se trate de una solicitud de descarga y en el cuerpo del correo contenga la URL de la descarga.

Este proceso puede llegar a ser engorroso debido a que con el continuo avance de las tecnologías los usuarios requieren realizar esta operación en el menor tiempo posible,

de una manera que no conlleve dificultad a la hora de realizar el proceso de solicitud de descarga.

1.4 – Descripción de los sistemas existentes

- **Download Manager for Android** tiene búsquedas integradas en Google, Youtube, Twitter y Yahoo!. Permite la descarga de archivos de más de 2 GB; permite reanudar de forma fiable las descargas interrumpidas; incorpora descargas en segundo plano; y su reproductor multimedia es compatible con archivos de audio (AAC, MP3 y M4A) y video (M4V, MP4 y MOV).
- **Loader Droid** para Android es capaz de descargar cualquier tipo de archivo, sin ninguna restricción. Descarga los archivos mediante la conexión que se le especifique (solo WiFi o WiFi e Internet móvil) y si la conexión se interrumpe, todas las descargas se suspenden inmediatamente, para que no preocupe por el consumo de datos. Además, soporta la descarga en múltiples corrientes, lo que acelera la descarga al triple, es estable, rápida y muy fácil de usar.
- **Turbo Download Manager** es un acelerador de descargas que hace uso de múltiples conexiones HTTP para descargar los archivos al teléfono. La aceleración de descargas se puede habilitar mediante el aumento del "número máximo de conexiones por descarga" y se puede optimizar aún más su velocidad de descarga mediante la modificación del "tamaño del buffer" en "Configuración". Con esta aplicación tendrás descargas hasta 5 veces más rápidas de archivos de tamaño ilimitado y controles de descarga: En cola, Pausa, Reanudar y Recuperar. Sin embargo, Turbo Download Manager sólo realiza descargas de archivos desde servidores que utilizan enlaces directos.
- **Download Manager Accelerator** supera la mayor parte de las limitaciones de los gestores de descarga por defecto. Lo mejor de contar con una aplicación como esta es que permite descargar archivos de cualquier tipo o tamaño de manera eficiente a nuestro teléfono o tableta, sin tener que estar al pendiente de cada una de ellas. Algunas de sus características más importantes son: capacidad de

reanudar descargas interrumpidas de forma fiable sin pérdida de datos, programación de descargas, barra de progreso de las descargas, interfaz de usuario sencilla y limpia, entre muchas otras más.

- **Download Manager Free** no sólo permite descargar vídeos, música o fotografías, sino también software, documentos (Word, Excel, PowerPoint, PDF), archivos comprimidos (ZIP, RAR) entre otros más. Además, esta aplicación tiene un acelerador inteligente de descargas que, mediante la segmentación dinámica de archivos puede acelerar la velocidad de tus descargas.
- **Fast Download Manager** es un excelente gestor de descargas para Smartphone y tablets Android que te permitirá descargar archivos de forma rápida y sencilla. Al igual que los anteriores permite pausar, cancelar y reiniciar cualquier descarga, ya sea que este pausada o rota. Soporta la descarga de archivos de más de 2 GB, ya sean imágenes, videos, música, software, documentos de Office (Word, PowerPoint, Excel), PDF, RAR, ZIP, etc. Para acelerar las descargas, el usuario tiene a posibilidad de configurar el número máximo de descargas paralelas y optimizar el tamaño de búfer

No se conoce de la existencia de alguna aplicación móvil a nivel nacional con funcionalidades similares a la aplicación propuesta en este proyecto.

Dichas aplicaciones poseen características que pudieran ser de valor para el actual trabajo. La mayoría de estas aplicaciones móviles no son idóneos para dicha función porque no garantiza el control del objetivo de las descargas q debe ser únicamente con fines educativos e investigativos. Además, el flujo de trabajo deseado en el actual trabajo es diferente debido a que las descargas se realizarán en el servidor del SAD no en el propio dispositivo móvil, el objetivo propuesto es poder gestionar el proceso de solicitud de descargas desde un teléfono inteligente Android no realizar la descarga en el dispositivo. De las aplicaciones antes expuestas se toma la idea de poder reintentar la solicitud de descarga en caso de que esta se encuentre en estado fallido.

1.5 – Tendencias, metodologías y/o tecnologías actuales

Metodologías de desarrollo de software

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología tienen la necesidad de venir documentadas para que los programadores que estarán dentro de la planeación del proyecto comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir[12]

El desarrollo de aplicaciones móviles tiene las mismas complicaciones que el desarrollo de una página web. Los desarrollos de software sufren también problemas. Una de las cualidades que tiene es que la duración es más corta y hay mucha competencia en el sector. De esta manera, obliga a que esté constantemente innovando e implementando cambios en la plataforma de desarrollo.[13]

Metodologías ágiles

Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que poseen características como las siguientes: [13], [14]

- Especialmente preparados para cambios durante el proyecto.
- Proceso menos controlado, con pocos principios.
- No existe contrato tradicional o al menos es bastante flexible.
- El cliente es parte del equipo de desarrollo.
- Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio.
- Pocos roles.
- Menos énfasis en la arquitectura del software.

Las metodologías ágiles son extremadamente prácticas ya que generalmente obvian gran parte de la documentación y están más preparadas para utilizarse en proyectos cuyos requisitos son imprecisos y cambian constantemente durante todo el proceso. Ejemplos de algunas metodologías ligeras podrían ser eXtreme Programming (XP), SCRUM y Crystal Methodologies.[14], [15]

Metodologías tradicionales

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y con requisitos muy cambiantes, pero son especialmente recomendadas en proyectos con las siguientes características:[15], [16]

- Cierta resistencia a los cambios.
- Proceso mucho más controlado, con numerosas políticas o normas.
- Existe un contrato prefijado con el cliente.
- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Grupos grandes y posiblemente distribuidos.
- Muchos roles.
- La arquitectura del software es esencial y se expresa mediante modelos.

En las metodologías tradicionales todo está mucho más controlado y se genera muchísima documentación antes de proceder a implementar el proyecto, con mucho mayor peso del análisis y el diseño sobre el proyecto. Entre las principales metodologías de este tipo están Rational Unified Process (RUP), ICONIX y Métrica 3.

Ambos tipos de metodologías tienen sus ventajas y desventajas, pero el uso de las mismas está vinculado al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.). Después de analizar las ventajas de cada tipo y las particularidades del producto que se desarrolla, se concluye que es recomendable el

uso de una metodología tradicional y por las características que se detallan a continuación fue seleccionado el Proceso Unificado de Desarrollo de Software (RUP).

1.5.1 - Proceso Unificado de Desarrollo de Software (RUP)

RUP unifica a todo el equipo de desarrollo de software y optimiza la productividad de cada miembro del equipo, brindándoles la experiencia de los líderes de la industria y las lecciones aprendidas de miles de proyectos.[17], [18]

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

1.6 - Tecnologías y lenguajes

Después de realizar un estudio de las principales tecnologías y lenguajes utilizados en la actualidad se exponen las características de las que se pretenden utilizar.

1.6.1- Arquitectura del sistema

Arquitectura cliente-servidor

TCP es un protocolo orientado a conexión donde las aplicaciones utilizan un modelo cliente-servidor en las comunicaciones. Un servidor es una aplicación que ofrece un servicio a usuarios de internet; un cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas. [19]

Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte. El servidor es un programa que recibe una solicitud, realiza el servicio requerido y

devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo. [20]

1.6.2- Lenguajes

UML

El Lenguaje de Modelado Unificado ayuda a especificar, visualizar y documentar modelos de sistemas de software, incluyendo su estructura y diseño (también se puede utilizar UML para modelar otros sistemas que no sean de software). Utilizando cualquiera de las numerosas herramientas basadas en UML, se pueden analizar los requisitos de su aplicación futura y diseñar una solución que los satisfaga. [21]

UML es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño. Los principales beneficios de UML son:[22]

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas utilizando conceptos orientados a objetos.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

Kotlin

Kotlin es un lenguaje de programación estático, gratuito, de código abierto y de propósito general. Se trata de un sistema inicialmente diseñado para JVM (Java Virtual Machine) y Android, que combina características de programación funcional y programación orientada a objetos.[23]

Interoperabilidad con código Java, aunque la sintaxis del lenguaje no es compatible con Java, Kotlin está diseñado para interoperar completamente con él. Esto significa que con una base de código existente escrita en Java puede interactuar correctamente con Kotlin y viceversa. Además, Kotlin abordó una serie de problemas de Java (por ejemplo, no hay tipos sin procesar, las referencias nulas están controladas por el sistema de tipos y no tiene excepciones marcadas), lo que sin duda hizo que el lenguaje fuera muy útil para los desarrolladores. Si lo comparamos con Java, también podemos ver que Kotlin es mucho más seguro. Por ejemplo, los nulos en Java, por ejemplo, son muy propensos a errores, mientras que en Kotlin, de forma predeterminada, los nulos ni siquiera existen.[23], [24]

1.7- Herramientas utilizadas

Visual Paradigm para UML

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.[25]

Visual Paradigm también ofrece:

- Navegación intuitiva entre la escritura del código y su visualización.
- Potente generador de informes en formato PDF/HTML.
- Documentación automática Ad-hoc.
- Ambiente visualmente superior de modelado.
- Sofisticado diagramador automáticamente de layout.

Esta poderosa herramienta ofrece a los desarrolladores de software la plataforma de desarrollo de vanguardia para construir aplicaciones de calidad en el menor tiempo posible.[26]

PowerShell

PowerShell, llamado Windows PowerShell originariamente, es una interfaz de línea de comandos o CLI (Command-Line Interface) que tiene la posibilidad de ejecutar Scripts (unión de comandos) y que facilita la configuración, administración y automatización de tareas multiplataforma, además dispone de un lenguaje de scripting. Powershell tiene una salida basada en objetos por lo que acepta y devuelve objetos de .NET ya que está basado en .NET CLR (Common Language Runtime) esto es ideal para automatización de procesos.[27]

Fue utilizado para realizar pruebas locales.

Android Studio

Es un IDE, una interfaz de desarrollo. En realidad, es una especie de escritorio de trabajo para un desarrollador. Allí se encuentra nuestro proyecto, las carpetas del mismo, los archivos que hay en él, y todo lo necesario para acabar creando la aplicación. Lo mejor de Android Studio es que ha sido creado por Google y fue presentado hace poco tiempo, por lo que no hablamos de una herramienta antigua y nada depurada, sino de un programa muy moderno que encima ha sido creado por los mismos que han creado el sistema operativo. [28]

Entre otras cosas, cuenta con algunas herramientas que nos facilitarán mucho el desarrollo de las aplicaciones, como por ejemplo el poder pre visualizar las aplicaciones en diferentes Smartphone y tablets para saber cómo está quedando el código que nosotros estamos editando, y cómo se ve en los diferentes tipos de pantalla que existen. [28]

API REST

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la

interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).[29]

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta). Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima.[29], [30]

En otras palabras, las API le permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla. [29], [30]

1.8– Conclusiones

Se definieron los principales conceptos asociados al dominio del problema para lograr un mayor entendimiento del mismo. Se realizó un análisis crítico del flujo actual de los procesos donde se detectó que existen deficiencias en el proceso de solicitud de descargas.

También se realizó una búsqueda de algunos sistemas informáticos existentes relacionados con la gestión de la información referente al proceso de descargas.

Además, se analizaron las metodologías, tecnologías, lenguajes y herramientas necesarias para el desarrollo del sistema propuesto, concluyendo utilizar RUP como metodología de desarrollo del software y UML como lenguaje de modelado. Se detallaron los lenguajes y herramientas utilizadas para el desarrollo del trabajo.

2 – Análisis y diseño de la propuesta de solución

2.1 – Introducción

En el presente capítulo se describen las características y conceptos fundamentales del modelo de dominio, se definen los actores que interactúan con el mismo, los requisitos funcionales y no funcionales, se realiza el modelo de casos de uso del sistema y la descripción de los mismos, se define la arquitectura del sistema.

2.2 – Modelo del dominio

Un modelo del dominio captura los tipos de objetos más importantes en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema.

2.2.1 – Definición de las entidades y los conceptos principales

La **red informática de la Universidad de Cienfuegos** posee un nodo central y un centro de administración en cada área, físicamente distribuida en estrella. Ya dentro de cada área la red es distribuida físicamente en forma jerárquica como un árbol. [31]

Se concibió en forma de redes físicamente separadas, que todo tráfico entre ellas debe pasar a través del nodo central.

Servicios de la red Ucf

1. Servicio de cuenta de usuario personal e intransferible con seguridad adecuada, que se usa de forma integrada para todos los servicios.
2. Servicio de correo, calendario y tareas.

3. Servicio de educación a distancia tanto en postgrado como en pregrado soportado por el proyecto Moodle de open source.
4. Servicio de internet y salida de correo internacional controlado por el proyecto de software libre SICC, administrándose cuotas personales y de áreas y con posibilidad de auditorías y control personal de acceso a su cuenta.
5. Servicio de video conferencias con la aplicación de software libre Open Meeting.
6. Servicio de mensajería instantánea con el software libre Jabber.
7. Servicio de noticias con el software cubano FREED
8. Servicio de nube como repositorio personal con software libre.
9. Servicio de actualización de hora para los diferentes sistemas que los necesiten.
10. Servicio de acceso inalámbrico con más de 30 puntos de acceso
11. Servicio de actualización de software tanto de software libre como de Windows y antivirus
12. Repositorio de Libros
13. Servicio de red social Tikiti con proyecto de software libre.
14. Servicio de publicación web
15. Servicio de publicación de blog.
16. Servicio de encuestas personalizadas con software libre.
17. Wikipedia actualizada en la red interna
18. Servicio de Telefonía IP con Open source Asterix/
19. Servicio para desarrollo de proyectos y aplicaciones.
20. Servicio de nombres de dominio y configuración automática
21. Servicio de inventario de Medios Básicos e infraestructura.
22. Repositorio de Música.
23. Descargas de aplicaciones.

Usuario, en informática (**user**), un usuario es un individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema, además se utiliza para clasificar a diferentes privilegios, permisos a los que tiene acceso un usuario o grupo de usuario, para interactuar o ejecutar con el ordenador o con los programas instalados en este. En sentido general, un usuario es el conjunto de privilegios, permisos, recursos o

dispositivos, a los cuales se tiene acceso. Es decir, un usuario puede ser tanto una persona como una máquina, un programa.[31]

Los **administradores de red** son básicamente el equivalente de red de los administradores de sistemas: mantienen el Hardware y Software de la red. Esto incluye el despliegue, mantenimiento y monitoreo del engranaje de la red: Switches, Routers, Cortafuegos. Las actividades de administración de una red por lo general incluyen la asignación de direcciones, asignación de protocolos de ruteo y configuración de tablas de Ruteo así como, configuración de Autenticación y autorización de los servicios.

Frecuentemente se incluyen algunas otras actividades como el mantenimiento de las instalaciones de red tales como los controladores y ajustes de las Computadoras e Impresoras. A veces también se incluye el mantenimiento de algunos tipos de servidores como VPN, Sistemas detectores de intrusos.

2.2.2 – Reglas del negocio a considerar

- El usuario debe de estar conectado a una red inalámbrica o física de la universidad de Cienfuegos.
- Las solicitudes de descargas serán aprobadas o denegadas por el administrador.
- Una vez finalizada la descarga el administrador de red deberá informar al usuario que solicitó la descarga.
- Las descargas deben ser para uso docente e investigativo.

2.2.3 – Representación del modelo del dominio

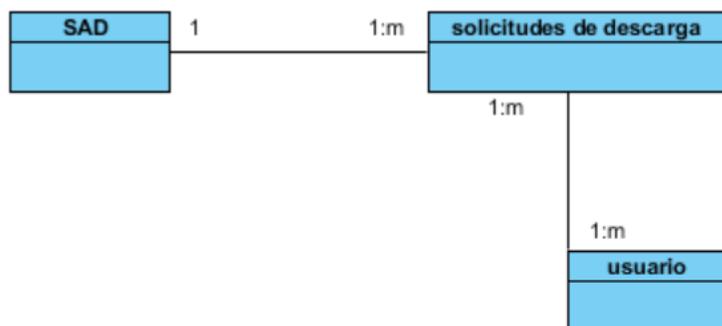


Figura 1. Diagramas de clases del modelo de objetos del dominio

2.3 – Requisitos

2.3.1 – Descripción del sistema propuesto

2.3.2 – Concepción general del sistema

La aplicación móvil permitirá al usuario autenticarse en el Sistema Automático de Descargas de la Universidad de Cienfuegos, y partiendo de esta información, le brindará acceso a determinadas vistas de acuerdo a su rol. Posibilitará crear solicitudes de descargas además de poder ordenarlas atendiendo a diferentes especificaciones ya sea por su nombre, prioridad, fecha de creación o su estado, además podrá eliminar la solicitud de descarga siempre y cuando el administrador no haya modificado su estado. El administrador es el que define el estado de prioridad de la solicitud ya sea normal o alto, además de ser el encargado de aceptar o denegar la solicitud de descarga.

2.3.3 – Requerimientos funcionales

- R1. Autenticar.
- R2. Cerrar sesión.
- R3. Crear solicitud de descarga.

- R4. Buscar solicitud de descarga.
- R5. Listar solicitudes de descargas propias.
- R6. Ordenar solicitudes de descargas.
- R7. Cambiar prioridad de la solicitud de descarga.
- R8. Aprobar solicitud de descarga.
- R9. Denegar solicitud de descarga.
- R10. Eliminar solicitud de descarga.

2.3.4 – Requerimientos no funcionales

- **Apariencia o interfaz externa.**

El sistema deberá tener una apariencia amigable, mostrará el color violeta partiendo del diseño original de la web en su barra de menú y será discreta a su vez, su interfaz deberá ser muy legible y debe lucir profesional.

- **Usabilidad.**

La aplicación móvil debe ser fácil de usar. Este producto permitirá una mejor accesibilidad al sistema automático de descargas de la Universidad de Cienfuegos.

- **Rendimiento.**

Deberá poseer velocidad de procesamiento, adquirida al formular los procedimientos con simplicidad para prevenir esperas prolongadas ganando así eficiencia, efectividad y aprovechamiento de recursos.

- **Soporte.**

Se le deben realizar diversas pruebas para comprobar la correcta funcionalidad de la aplicación. No debería ser complejo modificarla luego de terminado en caso de que se diera la necesidad de hacerlo. Su instalación deberá ser simple para evitar cualquier complicación por parte del usuario.

- **Portabilidad.**

La aplicación podrá ser instalada en cualquier teléfono móvil con sistema operativo Android.

- **Seguridad.**

Se debe garantizar un control estricto sobre la seguridad de la información teniendo en cuenta el establecimiento de niveles de acceso. No se deben permitir accesos sin autorización al sistema. Cada usuario solo tendrá acceso a sus descargas. El usuario solo puede acceder a la aplicación usando sus usuario y contraseña asignados por el departamento de red de la Universidad de Cienfuegos.

- **Confiabilidad.**

Se deberán de tomar las precauciones necesarias para prevenir fallos de cualquier tipo, ya sean por parte de la aplicación o del usuario, previniéndolos mediante validaciones y definiciones de reglas a seguir.

- **Interfaz interna.**

Deberá poseer componentes que faciliten el manejo de datos para el usuario y/o proporcionen confiabilidad al sistema.

- **Ayuda y documentación en línea.**

Poseerá un manual de usuario para explicar debidamente el uso del sistema, pero no tendrá documentación en línea.

- **Software.**

La aplicación necesita del lado de los usuarios un teléfono móvil con sistema operativo Android 5.0 Lollipop o superior.

- **Hardware.**

Se necesitará disponer de un teléfono celular con al menos, 512 MB de RAM y 20 MB HDD disponible del lado del cliente.

2.4 – Modelo de casos de uso del sistema

2.4.1– Actores del sistema

Nombre del actor	Descripción
Usuario	Puede autenticarse, crear una solicitud de descarga, ordenar sus solicitudes de descargas, eliminar la solicitud.
Administrador	Realiza las mismas funcionalidades de un usuario normal, además de aceptar o denegar las solicitudes de descargas y cambiar su prioridad.

Tabla 1. Definición de actores del sistema a automatizar

2.4.2 – Diagramas de casos de uso del sistema

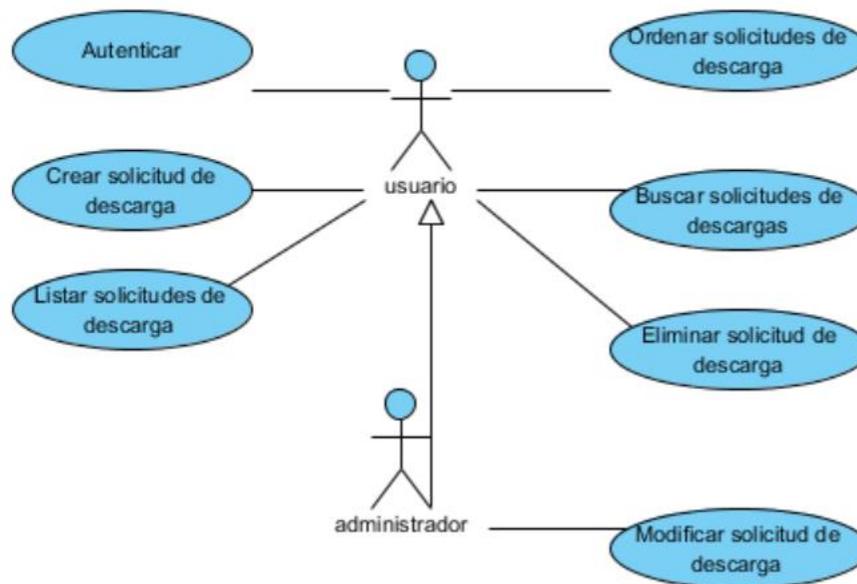


Figura 2. Diagrama del caso de uso de la aplicación móvil.

2.4.3 – Descripción de los casos de uso del sistema

Descripción textual de los casos de uso del sistema.

Caso de uso	Autenticar.
-------------	-------------

Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita al usuario iniciar sesión en la aplicación.
Resumen	El caso de uso comienza cuando el usuario introduce su nombre y su contraseña en la aplicación, tiene la opción de recordar contraseña. El caso de uso finaliza cuando el usuario se autentica en la aplicación .
Referencias	Requisitos funcionales asociados R1
Precondiciones	Debe de estar correctamente conectado al servidor de la aplicación y poseer un usuario y una contraseña registrada en el sistema.
Post-condiciones	Luego de autenticarse se muestra una vista correspondiente con el rol del usuario en el sistema.
Requisitos Especiales	
Prototipo	Anexo 1

Tabla 2. Descripción del caso de uso de sistema < Autenticar >

Caso de uso	Crear solicitud de descarga.
Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita crear una solicitud de descarga.
Resumen	El caso de uso comienza cuando el usuario añade una nueva solicitud de descarga la cual debe contar con un nombre, una breve descripción de su contenido y el URL del archivo q desea descargar. El caso de uso termina cuando el usuario confirma la creación de la solicitud de descarga.
Referencias	Requisitos funcionales asociados R3
Precondiciones	El usuario debe de estar autenticado en la aplicación.
Post-condiciones	La solicitud es enviada al administrador, posteriormente este acepta o deniega la solicitud.
Requisitos Especiales	No deben de existir campos vacíos en la petición de la solicitud.
Prototipo	Anexo 2

Tabla 3. Descripción del caso de uso de sistema < Crear solicitud de descarga >

Caso de uso	Buscar solicitud de descarga
Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita al usuario buscar una solicitud de descarga

	especifica
Resumen	El caso de uso inicia cuando el usuario introduce en la barra de búsqueda el nombre de una solicitud de descarga específica, la aplicación procede a realizar la búsqueda y devuelve como resultado la solicitud de descarga requerida. El caso de uso finaliza cuando la aplicación devuelve el resultado de la búsqueda.
Referencias	Requisitos funcionales asociados R4
Precondiciones	La solicitud de descarga que desea buscar debe existir .
Post-condiciones	La aplicación devuelve el resultado de la búsqueda solicitada.
Requisitos Especiales	No devuelve nada si no existe el nombre de la solicitud de descarga.
Prototipo	Anexo 3

Tabla 4. Descripción del caso de uso de sistema < Buscar solicitud de descarga >

Caso de uso	Listar solicitudes de descarga
Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita listar las solicitudes de descarga.
Resumen	El caso de uso comienza cuando el usuario ordena a la aplicación listar sus solicitudes de descargas donde puede visualizarlas. El caso de uso finaliza cuando la aplicación muestra en pantalla una lista con las solicitudes de descargas propias.
Referencias	Requisitos funcionales asociados R5 R6
Precondiciones	Deben existir solicitudes de descargas.
Post-condiciones	Las solicitudes de descargas deben de aparecer listadas .
Requisitos Especiales	En caso de que el usuario tenga el rol de administrador podrá listar todas las solicitudes de descargas existentes .
Prototipo	Anexo 4

Tabla 5. Descripción del caso de uso de sistema <Listar solicitudes de descarga

>

Caso de uso	Ordenar solicitudes de descarga.
Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita ordenar las solicitudes de descargas.
Resumen	El caso de uso comienza cuando el usuario decreta a la aplicación ordenar las solicitudes de descarga, las solicitudes pueden ser ordenadas atendiendo a diferentes campos ya sea su fecha de creación, nombre, estado y prioridad. El caso de uso termina cuando se realiza el ordenamiento de las solicitudes.
Referencias	Requisitos funcionales asociados R7

Precondiciones	Es necesario q existan solicitudes de descargas.
Post-condiciones	Se muestran las solicitudes ordenadas de acuerdo con el campo elegido por el usuario.
Requisitos Especiales	
Prototipo	Anexo 5

Tabla 6. Descripción del caso de uso de sistema < Ordenar solicitudes de descarga >

Caso de uso	Modificar solicitud de descarga.
Actores	Administrador (inicia el caso de uso)
Propósito	Posibilita al administrador realizar modificaciones en las solicitudes de descarga
Resumen	El caso de uso inicia cuando el administrador recibe una solicitud de descarga, posteriormente puede cambiar la prioridad de la solicitud y aceptarla o denegarla según considere conveniente. El caso de uso finaliza cuando el administrador termina una de estas funcionalidades.
Referencias	Requisitos funcionales asociados R8 R9 R10
Precondiciones	La solicitud de descarga debe de estar en estado solicitada o pendiente para poder realizar modificaciones .
Post-condiciones	Se efectúa el cambio realizado por el administrador .
Requisitos Especiales	La solicitud que fue denegada no se le podrán realizar modificaciones.
Prototipo	Anexo 6

Tabla 7. Descripción del caso de uso de sistema < Modificar solicitud de descarga >

Caso de uso	Eliminar solicitud de descarga.
Actores	Usuario (inicia el caso de uso)
Propósito	Posibilita eliminar la solicitud de descarga.
Resumen	

El caso de uso inicia cuando el usuario elimina una solicitud de descarga previamente creada, actualizando el listado de sus solicitudes y finalizando el caso de uso.	
Referencias	Requisitos funcionales asociados R11
Precondiciones	La solicitud q va a ser eliminada no puede haber sido denegada por el administrador.
Post-condiciones	Se elimina la solicitud de descarga del registro del sistema.
Requisitos Especiales	
Prototipo	Anexo 7

Tabla 8. Descripción del caso de uso de sistema < Eliminar solicitud de descarga

>

2.5 – Construcción de la solución propuesta

2.5.1 – Diagrama de clases del diseño

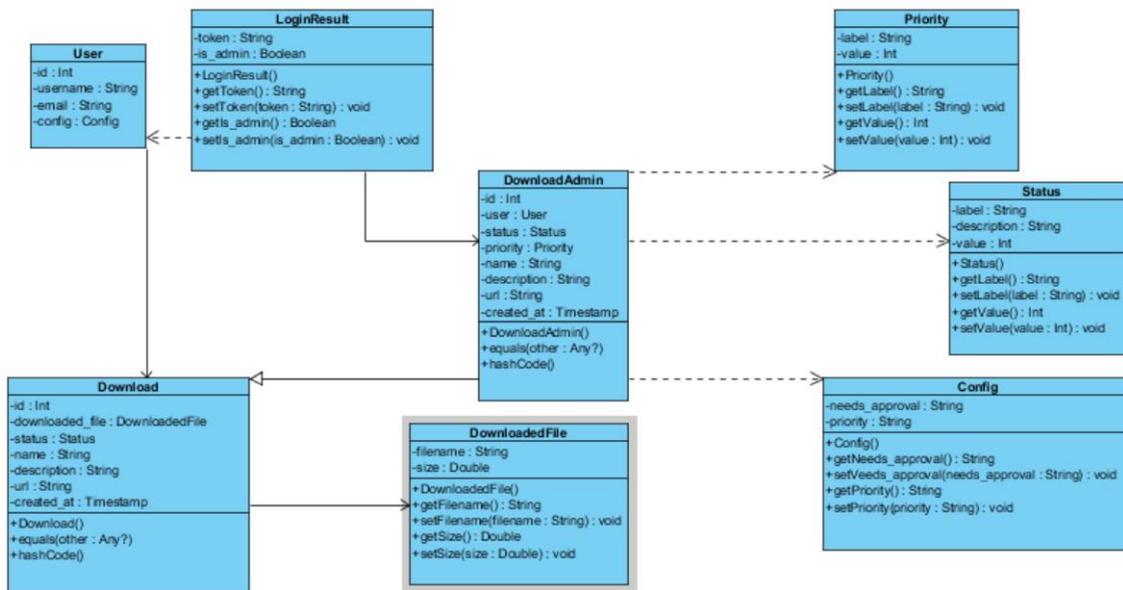


Figura 3. Diagramas de clases

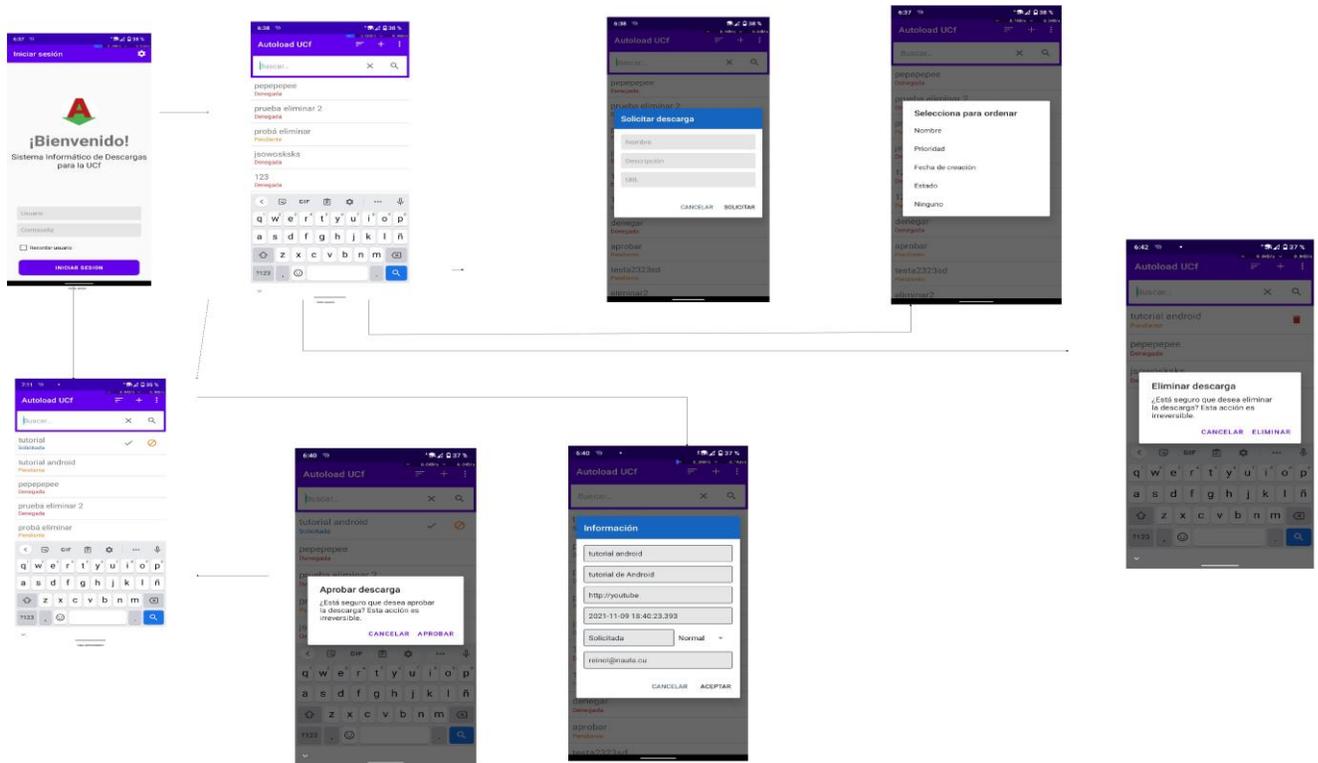


Figura 4. Diagramas de actividades en android

2.5.2 – Diagrama de implementación

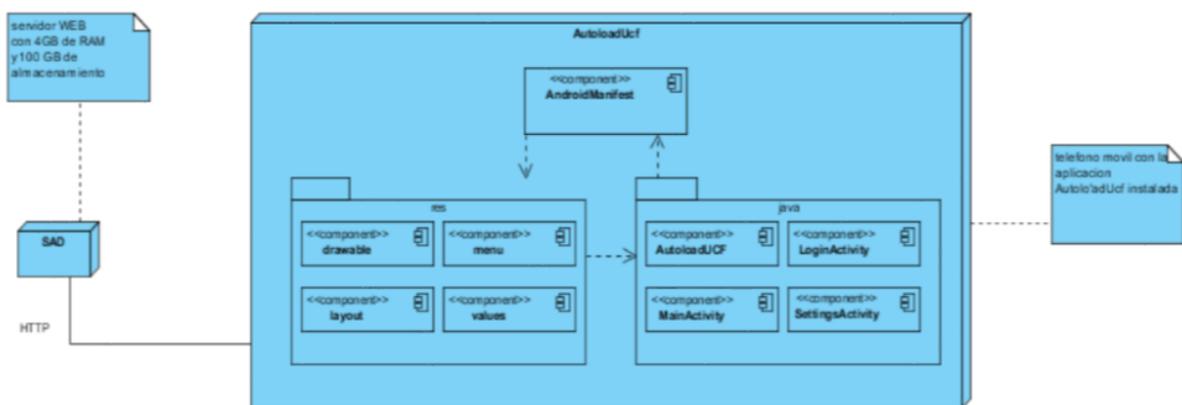


Figura 5. Diagrama de implementación

2.5.3 – Principios de diseño

El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

Para diseñar una buena experiencia móvil se deben desencadenar reacciones positivas y satisfacer los principios básicos de la usabilidad es importante mostrar el contenido y las funcionalidades de mayor importancia desde el primer momento. El contenido secundario debe estar disponible en menús desplegados, de terminologías simples para no confundir a los usuarios. La navegación debe de ser intuitiva, se le ha de hacer el camino muy sencillo al usuario para completar cualquier tarea. Los elementos de la interfaz deben de ser claramente visibles.

2.5.4 – Estándares en la interfaz de la aplicación

La interfaz de entrada/salida diseñada para la aplicación se concibió íntegramente aprovechando las potencialidades que ofrece el Kotlin y las interacciones se basan en selecciones de tipo menú y en acciones físicas sobre elementos de código visual, como son los botones y menús. Predomina el color violeta para la barra de navegación, color blanco para los menús y para las fuentes se utiliza el negro, todo contrastado con un color de fondo blanco. La carga visual se distribuye de manera cómoda evitando acumulaciones engorrosas y cumpliendo con la regla de distribución de la atención. Como parte del diseño de la aplicación móvil se integran una barra de navegación con menús incluidos en la parte de arriba de la vista de la aplicación que permite al usuario el acceso a todas las funcionalidades en todo momento. La entrada de información por parte de los usuarios se realiza a través de los componentes del formulario, minimizando así el ingreso de datos que resulta engorroso para el usuario.

2.5.5 – Tratamiento de errores

La aplicación está diseñada de forma tal, que las posibilidades de introducir información errónea por parte del usuario sean mínimas, aunque en muchas ocasiones el usuario escribe datos y en otras selecciona elementos de la pantalla, se mantienen validaciones

en cada campo, y en caso de errores se comunica el error cometido a través de mensajes de error. Estos se muestran en un lenguaje de fácil comprensión para los usuarios.

2.5.6 – Concepción General de la ayuda

La ayuda constituye una parte imprescindible en todo sistema. La aplicación móvil está diseñada para contar con la ayuda de un manual para la fácil comprensión de las funcionalidades que brinda. En el documento se encuentran explicadas todas las opciones del sistema con el fin de evitar errores por parte del usuario. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario. Además, cada botón cuenta con un texto de ayuda que explica la función que realiza.

2.6 – Conclusiones

1. La descripción del proceso de gestión de solicitudes de descargas como estudio principal, permite la identificación de los actores y trabajadores del sistema, cuyos roles ayudaron a la comprensión de las acciones y la identificación de los objetos con los que interactúan, sirviendo a la elaboración de los modelos de casos de uso del sistema, a su vez permiten una mejor comprensión para el modelado del mismo.
2. El sistema propuesto cuenta con la definición de 10 requisitos funcionales agrupados en 7 casos de uso para facilitar la gestión del negocio. La identificación y descripción de los actores y casos de uso del sistema detalla los roles y acciones que son llevadas a cabo en el funcionamiento del sistema.

3 – Estudio de factibilidad y validación de la solución

3.1 – Introducción

Este capítulo trata de diferentes aspectos relacionados al estudio de la factibilidad del producto. Se estiman el esfuerzo humano y el tiempo de desarrollo que se requieren para la elaboración del sistema, así como los costos y los beneficios tangibles e intangibles que reporta la utilización del mismo. Se realiza el análisis entre los costos y los beneficios para concluir si es factible o no desarrollarlo, para ello se utiliza el método de estimación mediante el análisis de Puntos de Casos de Uso.

3.2 – Estudio de factibilidad

3.2.1 – Planificación por casos de usos

Para estimar el esfuerzo en proyectos basados en casos de uso existen varios métodos entre los que se encuentran: el Análisis de Puntos de Función, COCOMO II y una variante más reciente, Análisis de Puntos de Casos de Uso. Esta técnica última posibilita obtener la duración del desarrollo de un proyecto, independientemente del lenguaje de programación, las metodologías, plataformas y tecnologías utilizadas, sin embargo, tiene presente otros aspectos importantes que influyen en el proyecto.

La estimación por medio del análisis de Puntos de Casos de Uso es un método propuesto por Gustav Karner, reutilizado y refinado luego por otros muchos autores. Es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "peso" a un número de factores, para contabilizar el tiempo y costo estimados para la terminación del proyecto

3.2.2 – Determinación de los costos

Cálculos de los puntos de casos de uso sin ajustar

$$UUCP = UAW + UUCW$$

Donde:

UUCP: puntos de casos de uso sin ajustar.

UAW: factor de peso de los actores sin ajustar.

UUCW: factor de peso de los casos de uso sin ajustar.

Factor de peso de los actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la complejidad de los actores del sistema.

Para obtener este valor se le asigna un valor a cada tipo de actor como se muestra en la tabla siguiente.

Tipo	Descripción	Peso
Simple.	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación. (API).	1
Medio.	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo.	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 9 Factor de peso de los actores de los actores del sistema

Actor	Tipo de actor	Factor de peso
Usuario	Complejo	3
Administrador	Complejo	3

Tabla 9. Clasificación de los actores atendiendo el factor de peso

Como se describe en la tabla anterior existen en el sistema dos actores de tipo complejo, debido a que interactúan con el sistema mediante una interfaz gráfica.

Multiplicando la cantidad de actores de cada tipo por el peso correspondiente se obtiene:

$$UAW = \Sigma (\text{Actor} * \text{Factor de peso}) = 6.$$

Factor de peso de los casos de uso

Este valor se calcula mediante un análisis de la cantidad de casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.[32] Los criterios se muestran en la siguiente tabla.

Tipo	Descripción	Peso
Simple.	El Caso de Uso contiene de 1 a 3 transacciones.	5
Medio.	El Caso de Uso contiene de 4 a 7 transacciones.	10
Complejo.	El Caso de Uso contiene 8 o más transacciones.	15

Tabla 10. Factor de peso de los casos de uso del sistema.

Caso de uso	Transacciones	Clasificación
Autenticar	3	Simple
Crear solicitud de descarga	3	Simple
Buscar solicitud de descarga	3	Simple
Listar solicitud de descarga	3	Simple
Ordenar solicitudes de descarga	3	Simple
Modificar solicitud de descarga	3	Simple
Eliminar solicitud de descarga	3	Simple

Tabla 11. Cálculo del factor de peso de los casos de uso.

Como puede verse en la tabla de clasificación anterior el sistema está conformado por 7 casos de uso simples.

De ahí que el factor de peso de los casos de uso sin ajustar puede calcularse como:

$$\mathbf{UUCW} = (7 * 5) = 35.$$

Por tanto:

$$\mathbf{UUCP} = UAW + UUCW$$

$$UUCP = 6 + 35 = 41.$$

Cálculo de los puntos de casos de uso ajustados

Una vez que se tienen los puntos de casos de uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$\mathbf{UCP} = UUCP * TCF * EF.$$

UCP = Puntos de casos de uso ajustados.

UUCP = Puntos de casos de uso sin ajustar.

TCF = Factor de complejidad técnica.

EF = Factor de ambiente.

Factor de complejidad técnica

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de estos factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado, el peso de cada uno, el valor asignado y el total.

Factor	Descripción	Peso	Valor	Comentario	Peso_i*Valor_i
T1	Sistema distribuido.	3	3	El sistema es distribuido.	9
T2	Objetivos de performance o tiempo de respuesta.	2	2	Velocidad limitada por las entradas provistas por el usuario.	4
T3	Eficiencia del	2	1	Escasas restricciones de	2

	usuario final.			eficiencia.	
T4	Procesamiento interno complejo.	2	3	Hay cálculos complejos en varias ocasiones.	6
T5	Código reutilizable.	1	3	El código es reutilizable casi en su totalidad.	3
T6	Facilidad de instalación.	3	3	Es fácil de instalar.	9
T7	Facilidad de uso.	3	4	Es relativamente fácil de usar.	12
T8	Portabilidad.	4	5	El sistema es portable.	20
T9	Facilidad de cambio	2	3	Se requiere un costo moderado de mantenimiento.	6
T10	Concurrencia.	1	0	No hay concurrencia.	0
T11	Objetivos especiales de seguridad.	2	3	Posee un control estricto de seguridad.	6
T12	Acceso directo a terceras partes.	1	0	No hay accesos directos a terceras partes.	0
T13	Entrenamiento a los usuarios.	1	1	Sistema fácil de usar.	1
Total					$\Sigma(\text{Peso} \cdot \text{Valor}) = 78$

Tabla 12. Significado, peso y valor de cada factor de complejidad técnica.

El factor de complejidad técnica se calcula mediante la siguiente ecuación.

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso} * \text{Valor}) = 0.6 + 0.01 * 78 = 1.38.$$

Factor ambiente

Las habilidades y el entrenamiento de los involucrados en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del factor de ambiente. El cálculo del mismo es similar al cálculo del factor de

complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5. En la siguiente tabla se muestra el significado y el peso de cada uno de estos factores.

Factor	Descripción	Peso	Valor	Comentario	Peso_i * Valor_i
E1	Familiaridad con el modelo de proyecto utilizado.	2	3	Existe familiarización con el modelo.	6
E2	Experiencia en la aplicación.	2	3	Existe experiencia debido a que se han utilizado las mismas herramientas en momentos anteriores.	6
E3	Experiencia en orientación a objetos.	2	2	Existe experiencia en programación orientado a objetos.	4
E4	Capacidad del analista líder.	2	3	El analista líder está capacitado.	6
E5	Motivación.	3	4	Existe motivación.	12
E6	Estabilidad de los requerimientos.	3	2	Requerimientos estables casi en su totalidad.	6

E7	Personal part-time.	-1	0	El equipo es full-time.	0
E8	Dificultad del lenguaje de programación.	-1	3	Se usará lenguaje PHP del cual se tiene un nivel medio.	-3
Total					$\sum (P_i * V_i) = 37$

Tabla 13. Significado, peso y valor de cada factor ambiente

El factor de ambiente se calcula mediante la siguiente ecuación.

$$EF = 1.4 - 0.03 * \sum (\text{Peso} * \text{Valor}) = 1.4 - 0.03 * 37 = 0.29.$$

Por tanto:

$$UUCP = 41.$$

$$TCF = 1.38.$$

$$EF = 0.29.$$

$$UCP = UUCP * TCF * EF = 41 * 1.38 * 0.29 = 16,4082.$$

De los puntos de casos de uso a la estimación del esfuerzo

$$E = UCP * CF.$$

E: Esfuerzo estimado en horas hombres.

CF: Factor de conversión.

- Se contabilizan cuántos factores de los que afectan al factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
- Se contabilizan cuántos factores de los que afectan al factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas hombre.

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este proyecto si se analizan los valores tabulados anteriormente, es posible percatarse de que el total es 2, por lo que:

CF = 20 horas-hombre/punto de caso de uso.

E = UCP * CF.

E = 16,4082 * 20 horas/hombre.

E = 328,164 horas/hombre.

E ≈ 328 horas/hombre.

Actividad	Esfuerzo (%)	Valor (horas/hombre)
Análisis	20	164
Diseño	20	164
Implementación	40	328
Pruebas	10	82
Sobrecarga (otras actividades)	10	82
Total	100	820

Tabla 14. Cálculo el esfuerzo para cada actividad.

Cálculo de tiempo y costo

Luego de haber obtenido el esfuerzo total necesario para el desarrollo del sistema se puede calcular a partir del valor obtenido el tiempo necesario y a partir de este el costo asociado por concepto de salario, así como los beneficios tangibles e intangibles que se obtienen con el desarrollo del sistema.

Cálculo de tiempo

Trabajando 25 días al mes y 8 horas diarias como promedio, se tiene que:

Duración (días) = Total de horas/hombre entre 8 horas diarias = $820 / 8 = 102.5$.

Duración (meses) = Total de días / 25 días al mes = $103 / 25 = 4,12 \approx 4$.

Por tanto:

Se concluye que el proyecto requiere de aproximadamente 4 meses.

Cálculo de costo

Tomando como salario promedio mensual \$3500.00 MN.

Costo CUP = 4 meses * \$3500.00.

Costo CUP = \$14000.00.

Por tanto:

Se concluye que el proyecto tiene un costo asociado de \$14000.00 CUP

3.2.3– Beneficios tangibles e intangibles

Los beneficios tangibles son las ventajas económicas cuantificables que obtiene la organización a través del uso del sistema informático y se pueden estimar en pesos, recursos y tiempo ahorrado. Como beneficio tangible de este proyecto se encuentra:

- El desarrollo de la aplicación tiene un costo de 14000 MN, lo que contribuye a la reducción de gastos de la empresa por concepto de no pagar por un software.
- Se ahorra electricidad ya q no será necesario utilizar los pc de los laboratorios para el procesamiento de las solicitudes de descarga.

Por su parte, los beneficios intangibles, son aquellos beneficios difíciles de cuantificar que obtiene una organización a través del uso de un sistema de información, pero por ello no dejan de ser menos importantes. Dentro de los beneficios intangibles del trabajo se destacan:

- La aplicación expande el uso del SAD debido a que los usuarios tienen mayor acceso a un dispositivo móvil que a un PC.
- Disminución de tiempo a la hora de realizar el flujo de trabajo referente al proceso de gestión de solicitud de descarga.
- Garantiza la seguridad y acceso a la información.

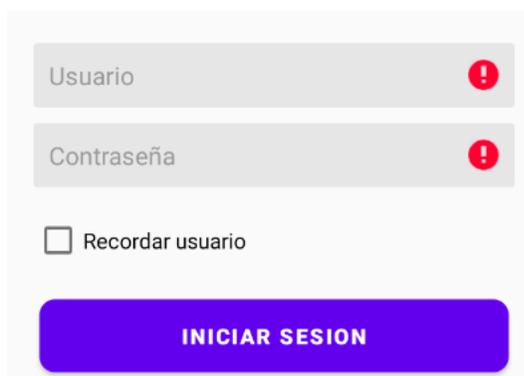
3.2.4 – Análisis de costos y beneficios

La aplicación resultante de esta investigación, no implica ningún costo para la Universidad de Cienfuegos, aunque el desarrollo de todo producto informático va asociado a un costo y su justificación económica está dada por los beneficios tangibles e intangibles que se producen. El desarrollo de la aplicación no solicitó inversión de ningún tipo, ni en medios técnicos, ni en requerimientos de software para su ejecución, pero su desarrollo está valorado en 14000 MN, por lo que la aplicación ha reportado un ahorro económico al centro, no solo por sus prestaciones, sino también por su desarrollo.

3.3 – Validación de la solución

Las pruebas son un aspecto crucial en el control de calidad del desarrollo de software y, dentro de estas, las pruebas funcionales, en las cuales se verifica de forma dinámica el comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba. Las pruebas funcionales son las que se aplican al producto final, permitiendo detectar los puntos del producto que no cumplen sus especificaciones, es decir, que no funcionan correctamente. Como propuesta de validación para la solución de esta investigación se han diseñado casos de prueba para validar los requerimientos funcionales del sistema informático.

3.3.1- “Autenticar”



The screenshot shows a login form with two input fields: 'Usuario' and 'Contraseña'. Both fields have a red exclamation mark icon to their right, indicating a validation error. Below the fields is a checkbox labeled 'Recordar usuario' which is unchecked. At the bottom of the form is a blue button with the text 'INICIAR SESION' in white capital letters.

Figura 6.< Validacion autenticar>

Acción: Iniciar sesión.

Descripción: Permite al usuario iniciar sesión en la aplicación y acceder a sus solicitudes de descargas.

Validaciones:

- Usuario: No puede estar vacío y debe existir en el sistema del directorio activo.
- Contraseña: No puede estar vacío y debe coincidir con la contraseña del usuario en el servidor.

3.3.2 - “crear solicitud de descarga”



The screenshot shows a form titled 'Solicitar descarga' in a blue header. Below the header are three input fields: 'Nombre', 'Descripción', and 'URL'. Each field has a red exclamation mark icon to its right, indicating a validation error. At the bottom of the form are two buttons: 'CANCELAR' and 'SOLICITAR'.

Figura 7. Validacion <crear solicitud de descarga>

Acción: Solicitar descarga.

Descripción: El usuario deja campos vacíos, se muestra una señal de error.

Validaciones:

- Nombre de la descarga: No puede estar vacío.
- Descripción: No puede estar vacío.
- Url: No puede estar vacío.

3.3.3- “Listar descargas”

Denied Download Denegada		
Pending Download Fallida		
Requested Download Solicitada		
apache-cassandra-4.0.0-bin.tar.gz Descargada		
My awesome download Archivada		
My awesome download Archivada		
My awesome download Pendiente		

Figura 8. Validacion <listar descargas>

Acción: Listar descargas

Descripción: Permite a los usuarios ver sus descargas, así como el estado de cada una.

Validaciones: El usuario debe estar autenticado.

3.3.4- “Aprobar descarga”

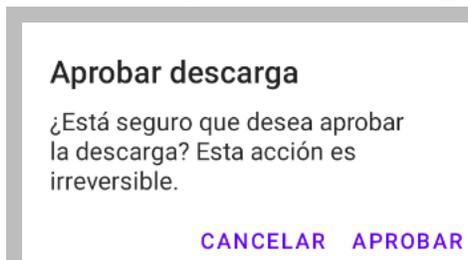


Figura 9. Validacion <aprobar descarga>

Acción: Aprobar descarga.

Descripción: Permite al administrador aprobar una descarga solicitada por cualquier usuario.

Validaciones:

- La descarga tiene que tener estado SOLICITADA.
- El usuario debe contar con privilegios de administración.

3.3.5“Denegar descarga”

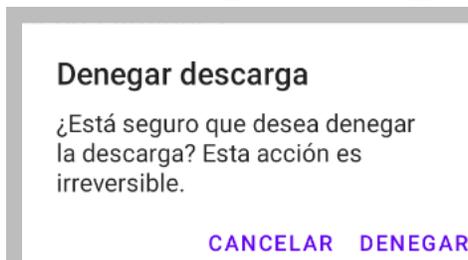


Figura 10. Validacion <denegar descarga>

Acción: Denegar descarga.

Descripción: Permite al administrador denegar una descarga solicitada por cualquier usuario.

Validaciones:

- La descarga tiene que tener estado SOLICITADA.
- El usuario debe contar con privilegios de administración.

3.3.6 “Cambiar prioridad a la solicitud de descarga”



The image shows a validation dialog box with several input fields and a dropdown menu. The fields contain the following text: 'tutorial android', 'tutorial de Android', 'http://youtube', '2021-11-09 18:40:23.393', 'Solicitada', and 'reinol@nauta.cu'. A dropdown menu is open over the 'Solicitada' field, showing 'Normal' and 'Alta' as options. At the bottom of the dialog, there are two buttons: 'CANCELAR' and 'ACEPTAR'.

Figura 11. Validacion <Cambiar prioridad a la solicitud de descarga >

Acción: Cambiar prioridad a descarga.

Descripción: Permite al administrador cambiar la prioridad de descarga a cualquier descarga solicitada.

Validaciones:

- Las prioridades posibles son Normal y Alta.
- El usuario debe contar con privilegios de administración.

3.4 – Conclusiones

1. El estudio de factibilidad mediante el análisis de Puntos de Casos de Uso demuestra que el tiempo para desarrollar el sistema informático propuesto es de aproximadamente 4 meses con un costo que asciende a 14000 MN.
2. El diseño de la validación para la entrada de los datos en el sistema mediante una muestra de casos de prueba permitió tener una idea de los errores que pudieran ocurrir durante el funcionamiento del software

Conclusiones

1. Según los objetivos trazados en la presente investigación, se arriba a la conclusión de que se realizó un análisis del proceso de solicitud de descargas en la Universidad de Cienfuegos, detectándose como deficiencias la necesidad de tener que acceder a una computadora ya sea en el laboratorio o portátil para poder realizar una solicitud de descarga lo que puede resultar engorroso para la mayoría de los usuarios.
2. La aplicación móvil propuesta garantiza facilidad y rapidez a la hora de realizar el procedimiento de gestión de solicitudes de descarga. Al ser desarrollada para sistema operativo Android se cumple con el objetivo de desplegarla en la universidad de Cienfuegos ya q después de un estudio previamente realizado la gran mayoría de usuarios de la Ucf cuentan con dispositivos móviles Android.
3. Para guiar el proceso de desarrollo del sistema informático propuesto se usó la metodología RUP, identificándose 11 requerimientos funcionales, de los que se obtuvieron 7 casos de uso y 2 actores del sistema.
4. El estudio de factibilidad demuestra que el tiempo para desarrollar la aplicación propuesta es de aproximadamente 4 meses con un costo que asciende a 14000 MN. El diseño de casos de prueba para validación del sistema permitió tener una idea de los errores que pudieran ocurrir durante el funcionamiento del software.

Recomendaciones

- Añadir una funcionalidad para que la aplicación pueda enviar una notificación al usuario cuando su descarga haya sido completada sin necesidad de tener la aplicación abierta.

Referencias bibliográficas

- [1] J. Ranchal, «Gestores de descargas o cómo gestionar las transferencias de contenido en Internet», *MuyComputer*, mar. 30, 2020. <https://www.muycomputer.com/2020/03/30/gestores-de-descargas-gratis/> (accedido nov. 09, 2021).
- [2] «Gestor de descargas: qué es, qué ventajas tienen y gestores recomendados». <https://www.xataka.com/basics/gestor-descargas-que-que-ventajas-tienen-gestores-recomendados> (accedido nov. 09, 2021).
- [3] «Servicios - Universidad de Cienfuegos». https://www.ucf.edu.cu/?page_id=820 (accedido nov. 09, 2021).
- [4] S. Plaza Estévez, N. Ramírez Lamela, y C. Acosta Morales, «API de servicios web orientados a accesibilidad», 2016.
- [5] «Archivo en Informática - Concepto, características y formato». <https://concepto.de/archivo-informatico/> (accedido nov. 09, 2021).
- [6] «COMPLEMENTO INFORMATICA: definición de COMPLEMENTO INFORMATICA y sinónimos de COMPLEMENTO INFORMATICA (español)». <http://diccionario.sensagent.com/COMPLEMENTO%20INFORMATICA/es-es/> (accedido nov. 09, 2021).
- [7] «¿Qué es una URL? - Aprende sobre desarrollo web | MDN». https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_URL (accedido nov. 09, 2021).
- [8] S. W. S. usado para est y A. L. C. E. D. P. Y. L. D. Programación, «Servidor Web - EcuRed». https://www.ecured.cu/Servidor_Web (accedido nov. 09, 2021).
- [9] roid T. producida por S. con el SO, roid D. G. W. oficial, y roid W. O. general L. oficial 2005 N. L. L. A. 2 0 y G. 2 I. Multilingüe, «Android - EcuRed». <https://www.ecured.cu/Android> (accedido nov. 09, 2021).
- [10] «Definición y cómo funcionan las aplicaciones móviles». <https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/> (accedido nov. 09, 2021).

- [11] «mision», *Universidad de Cienfuegos*. https://www.ucf.edu.cu/?page_id=407 (accedido nov. 09, 2021).
- [12] «Metodologías del Desarrollo de Software», *OK HOSTING | Hospedaje Web, Dominios, Desarrollo de Software, Marketing Online, SEO*, oct. 19, 2016. <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/> (accedido nov. 09, 2021).
- [13] «Metodologías para el desarrollo de aplicaciones móviles», *Syntonize*, ago. 30, 2021. <https://www.syntonize.com/metodologias-desarrollo-de-aplicaciones-moviles/> (accedido nov. 09, 2021).
- [14] S. D. Amaro Calderón y J. C. Valverde Rebaza, «Metodologías ágiles», *Perú Trujillo*, 2007.
- [15] R. G. Figueroa, C. J. Solís, y A. A. Cabrera, «Metodologías tradicionales vs. metodologías ágiles», *Univ. Téc. Part. Loja Esc. Cienc. Comput.*, vol. 9, pp. 1-10, 2008.
- [16] S. M. Velásquez, J. D. V. Montoya, M. E. G. Adasme, E. J. R. Zapata, A. A. Pino, y S. L. Marín, «Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software», *Rev. CINTEX*, vol. 24, n.º 2, pp. 13-23, 2019.
- [17] I. Jacobson, G. Booch, y J. Rumbaugh, «El Proceso Unificado de Desarrollo de Software (RUP)», *Editor. Addison Wesley Esp.*, 2000.
- [18] F. A. Manchego Peña, «Proceso Unificado de desarrollo de software RUP», 2019.
- [19] E. Marini, «El modelo cliente/servidor», *Recuperado El*, vol. 5, 2012.
- [20] O. Lizama, G. Kindley, y J. Ignacio, «Redes de computadores Arquitectura Cliente-Servidor», *Univ. Tec. Federico St. Maria*, pp. 1-8, 2016.
- [21] C. Larman, L. M. H. Rodríguez, y H. C. Anaya, *UML y Patrones: Introducción al análisis y diseño orientado a objetos*, vol. 2. Prentice Hall, 1999.
- [22] P. Stevens, R. Pooley, M. F. Alarcón, Ó. S. Martínez, y F. P. Sorrozal, *Utilización de UML en Ingeniería del Software con Objetos y Componentes*, vol. 14. Addison Wesley, 2002.

- [23] «¿Qué es Kotlin? Android y el desarrollo de aplicaciones | Tokio», *Tokio School*, feb. 06, 2021. <https://www.tokioschool.com/noticias/que-es-kotlin/> (accedido nov. 09, 2021).
- [24] «En comparación con Kotlin, las ventajas y desventajas de Java - programador clic». <https://programmerclick.com/article/63811888092/> (accedido nov. 09, 2021).
- [25] «Visual Paradigm - EcuRed». https://www.ecured.cu/Visual_Paradigm (accedido nov. 09, 2021).
- [26] P. López, «Herramienta CASE Visual Paradigm», p. 25.
- [27] «¿Qué es PowerShell y para qué sirve? [Noviembre 2021]», *GEEKNETIC*. <https://www.geeknetic.es/PowerShell/que-es-y-para-que-sirve> (accedido nov. 09, 2021).
- [28] «Android Studio - EcuRed». https://www.ecured.cu/Android_Studio (accedido nov. 09, 2021).
- [29] A. Arsaute, F. A. Zorzán, M. Daniele, A. González, y M. Frutos, «Generación automática de API REST a partir de API Java, basada en transformación de Modelos (MDD)», 2018.
- [30] «API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos». <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/> (accedido nov. 09, 2021).
- [31] «Usuario (Informática) - EcuRed». [https://www.ecured.cu/Usuario_\(Inform%C3%A1tica\)](https://www.ecured.cu/Usuario_(Inform%C3%A1tica)) (accedido nov. 11, 2021).
- [32] L. O. F. Monografias.com, «Estimación de software basada en puntos de casos de uso. Ejemplo práctico - Monografias.com». <https://www.monografias.com/trabajos87/estimacion-software-basada-puntos/estimacion-software-basada-puntos.shtml> (accedido jun. 20, 2020).

Glosario de términos

Api REST: Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones

CASE: Ingeniería de Software Asistida por Computación.

CU (Caso de Uso): Secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

IDE (Entorno de Desarrollo Integrado): Aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

TCP/IP: El conjunto TCP/IP tiene un grado muy elevado de fiabilidad, es adecuado para redes grandes y medianas, así como en redes empresariales. Se utiliza a nivel mundial para conectarse a Internet y a los servidores web.

XML (Lenguaje de Marcas Extensible): Utilizado para almacenar datos en forma legible. Permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

Anexos

Anexo 1 – protótipo autenticarece.



Prototipo de formulario de inicio de sesión. Incluye campos de texto para 'Usuario' y 'Contraseña', un checkbox para 'Recordar usuario' y un botón de 'INICIAR SESION'.

Anexo 2 – protótipo crear solicitud de descarga.



Prototipo de formulario para solicitar una descarga. Incluye un botón de 'Solicitar descarga' y campos de texto para 'Nombre', 'Descripción' y 'URL'. En la parte inferior hay botones para 'CANCELAR' y 'SOLICITAR'.

Anexo 3 – protótipo buscar solicitud de descarga. }



Prototipo de campo de búsqueda. Incluye un campo de texto con el placeholder 'Buscar...', un botón de 'X' para cerrar y un ícono de lupa para buscar.

Anexo 4 – protótipo listar solicitud de descarga.

pepepepee
Denegada
prueba eliminar 2
Denegada
probá eliminar
Pendiente

Anexo 5 – protótipo ordenar solicitud de descarga.

Selecciona para ordenar

Nombre

Prioridad

Fecha de creación

Estado

Ninguno

Anexo 6 – protótipo modificar solicitud de descarga.

tutorial	✓	⊘
Solicitada		

Anexo 7 – protótipo eliminar solicitud de descarga.

tutorial android

Pendiente

