| **DUOC UC - Escuela de informática y telecomunicaciones** |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: Duoc Swap* |
|  |
| **Revisión*: [01]*** |
| **[27/08/2024]** |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.gjdgxs)

[**Ficha del documento 4**](#_heading=h.1fob9te)

[**1. Introducción 5**](#_heading=h.3znysh7)

[1.1.](#_heading=h.2et92p0) Propósito 5

[1.2.](#_heading=h.tyjcwt) Ámbito del Sistema 5

[1.3.](#_heading=h.3dy6vkm) Definiciones, Acrónimos y Abreviaturas 5

[1.4.](#_heading=h.1t3h5sf) Referencias 5

[1.5.](#_heading=h.4d34og8) Visión General del Documento 5

[**2.**](#_heading=h.2s8eyo1) **Descripción General 6**

[2.1.](#_heading=h.17dp8vu) Perspectiva del Producto 6

[2.2.](#_heading=h.3rdcrjn) Funciones del Producto 6

[2.3.](#_heading=h.26in1rg) Características de los Usuarios 6

[2.4.](#_heading=h.lnxbz9) Restricciones 6

[2.5.](#_heading=h.35nkun2) Suposiciones y Dependencias 7

[2.6.](#_heading=h.1ksv4uv) Requisitos Futuros 7

[**3.**](#_heading=h.44sinio) **Requisitos Específicos 8**

[3.1](#_heading=h.2jxsxqh) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.z337ya) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.3j2qqm3) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.1y810tw) *Interfaces de software 8*

[3.2](#_heading=h.4i7ojhp) Requisitos funcionales 9

[3.3](#_heading=h.2xcytpi) Requisitos no funcionales (Organización) y de calidad (Producto) 9

[*3.3.1*](#_heading=h.1ci93xb) *Requisitos de Rendimiento o Eficiencia 9*

[*3.3.2*](#_heading=h.3whwml4) *Requisitos de Seguridad 9*

[*3.3.3*](#_heading=h.2bn6wsx) *Requisitos de Usabilidad 9*

[*3.3.4*](#_heading=h.qsh70q) *Requisitos de Disponibilidad 9*

[*3.3.5*](#_heading=h.3as4poj) *Requisitos de Portabilidad 9*

[*3.3.6*](#_heading=h.1pxezwc) *Requisitos de Mantenibilidad 9*

[*3.3.7*](#_heading=h.49x2ik5) *Requisitos de Funcionalidad 9*

[3.4](#_heading=h.2p2csry) Requisitos No funcionales Organizacionales 10

[**4. Propuesta de Planificación 11**](#_heading=h.147n2zr)

[4.1 Descripción general acerca de la Planificación 11](#_heading=h.3o7alnk)

[*4.1.2 Definición del Equipo de Trabajo 11*](#_heading=h.23ckvvd)

[*4.1.3 Definición de Actividades principales del Proyecto 11*](#_heading=h.ihv636)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 11*](#_heading=h.32hioqz)

[*4.1.6 Carta Gantt 11*](#_heading=h.1hmsyys)

[5. Anexos 12](#_heading=h.41mghml)

[*5.1 Acta de Proyecto 12*](#_heading=h.2grqrue)

[*5.2 Matriz Especificación de Requerimientos 12*](#_heading=h.vx1227)

[*5.3 Prototipado de Software 12*](#_heading=h.3fwokq0)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 12*](#_heading=h.1v1yuxt)

[*5.5 Planilla Carta Gantt 12*](#_heading=h.4f1mdlm)

**Ficha del documento**

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *27/08/2024* |  | Jorge Pavez |  |
|  |  |  |  |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Jorge Pavez* | *Jefe de proyecto* |
| *Matías Carvajal* | *Administrador BD, Calidad y Testing* |
| *Carlos Muñoz* | *Programador, Diseñador* |

**1. Introducción**

**1.1. Propósito**

El propósito de este documento es esclarecer y dar a conocer todos los puntos necesarios para la realización de una aplicación con sistema de Duoc Swap, desde su etapa de inicio, hasta el término del proyecto con toda la información a entendimiento público. Este documento va dirigido a las personas que deseen hacer uso de este, como usuarios y administradores.  
La finalidad de este proyecto consiste en desarrollar un aplicativo móvil que pueda facilitar el intercambio de material educativo en desuso entre participantes de la comunidad educativa DUOC UC, en etapa inicial enfocado en la sede de San Bernardo.  
  
**1.2. Ámbito del Sistema**

El nombre del sistema será Duoc Swap.

El sistema debe permitir crear usuarios con correo DUOC UC, loguearse con correo DUOC, Recupera contraseña, crear publicaciones, buscar publicaciones, buscar por categoría, llevar lista de detalle de publicaciones subidas por el usuario, llevar registro de match de intercambio o regalo, permitir la comunicación entre usuarios por medio de un foro, reactivar publicaciones. A su vez el sistema tendrá un usuario administrador el cual podrá moderar usuarios y publicaciones según corresponda.

Con la implementación de este sistema se busca que los integrantes de la comunidad educativa DUOC UC puedan acceder a un ambiente para realizar donaciones o cambios de artículos educativos en desuso de manera fácil y segura, en la cual puedan interactuar cómodamente con otros usuarios, buscar artículos que le sean relevantes y así potenciar la reutilización de material en desuso con fines de mejorar la obtención de recursos por parte de los alumnos y a su vez reducir los residuos y elementos desechados por otros.

**1.3. Definiciones, Acrónimos y Abreviaturas**

**Ionic:** Framework de desarrollo de aplicaciones móviles y web que permite crear aplicaciones multiplataforma utilizando tecnologías web como HTML, CSS y JavaScript. Se basa en Angular, pero también soporta otros frameworks como React y Vue. Ionic se utiliza principalmente para desarrollar aplicaciones híbridas, es decir, aplicaciones que funcionan tanto en iOS como en Android a partir de un solo código base.

**Angular:** Framework de desarrollo web de código abierto creado y mantenido por Google, diseñado para construir aplicaciones web robustas, dinámicas y de una sola página (Single Page Applications, SPA). Utiliza el lenguaje de programación TypeScript, que es un superconjunto de JavaScript, para estructurar y escribir aplicaciones con características avanzadas.

**Capacitor:** Runtime (entorno de ejecución) para aplicaciones móviles desarrollado por Ionic que permite construir aplicaciones nativas utilizando tecnologías web como HTML, CSS y JavaScript. Al igual que Cordova, Capacitor permite que las aplicaciones web interactúen con las funcionalidades nativas de los dispositivos móviles, como la cámara, el GPS, y el almacenamiento local, pero con una arquitectura más moderna y flexible.

**JavaScript:** Es un lenguaje de programación interpretado, orientado a objetos y con características de programación funcional. Es utilizado principalmente en páginas web para proporcionar interactividad y dinamismo a los usuarios.

**MariaDB:** Sistema de gestión de bases de datos relacional (RDBMS) que es una bifurcación (fork) de MySQL.

**Tags:** Son etiquetas utilizadas en HTML y XML para estructurar y formatear el contenido de una página web.

**Match:** En programación, este término puede referirse a varias cosas diferentes, dependiendo del contexto. Puede significar una comparación de patrones (como en la función match() de Python), o una coincidencia en una base de datos (como en una búsqueda de registros que coincidan con ciertos criterios).

**EDT:** Estructura de Desglose de Trabajo. Es una técnica utilizada en la gestión de proyectos para descomponer el trabajo en tareas más pequeñas y manejables.

**ERS:** Especificación de Requisitos de Software. Es un documento que describe los requisitos funcionales y no funcionales de un sistema de software.

**DAS:** Data Acquisition System. Es un sistema utilizado para recopilar y registrar datos en tiempo real, a menudo en entornos industriales o científicos.

**1.4. Referencias**

Todas las referencias se encuentran en apartado ANEXO

**1.5. Visión General del Documento**

En este documento podremos la descripción general del trabajo que se realizará junto a sus costos y tiempos de entrega. Podremos ver las funcionalidades del producto, características de usuarios, los requisitos específicos, tanto los requisitos funcionales y no funcionales. Al final podremos encontrar los anexos que vendrán adjuntos junto a este documento de ERS

**2. Descripción General**

Se establece como una aplicación móvil estable y con una interfaz de usuario amigable, la cual permita a los estudiantes realizar intercambios de productos educacionales (u otros) de manera segura en la sede de San Bernardo. A su vez busca que con los mismos usuarios de nuestra comunidad educativa pueda visualizar en esta aplicación:

- 1) Establecer intercambios, aprobación y desaprobación de “match” o cruce de productos entre usuarios y poder seleccionar entre las opciones con cual desea hacer el intercambio.

- 2) listas de intercambios, que permita mostrarnos nombre y objeto permutado.

- 3) foro de mensajes para conversaciones entre usuarios, lo cual permitirá una vez concretado el intercambio ponerse de acuerdo entre los usuarios para hacer más fácil el intercambio.

**2.1. Perspectiva del Producto**

Será una aplicación web desarrollada en HTML5, CSS, Javascript,Bootstrap, Django, con conexión a bases de datos Oracle SQL. Se espera una compatibilidad con la mayor cantidad de dispositivos, ya sean de escritorio o móviles ya que se basa en la lectura de navegadores de internet con capacidad de descifrar protocolos HTML, independiente del sistema operativo (Windows, MacOS, Linux, Android, iOS)  
La aplicación se vincula con cada perfil a través de un usuario (correo DUOC UC) y contraseña.

**2.2. Funciones del Producto**

* Creacion de usuario
* Login
* Cambio contraseña
* Búsqueda de productos (por nombre, categoría o tags)
* Publicación de productos
* Lista y administracion de mis publicaciones
* Lista y administración de match entre publicaciones
* Reactivación de publicaciones eliminadas o desactivadas
* Moderación de usuarios, publicaciones y tags que no estén dentro de las normas del aplicativo.

**2.3. Características de los Usuarios**

**Usuario DUOC:**  
- Usuarios que cuenten con correo DUOC UC, ellos pueden publicar, buscar, administrar, republicar, listar publicaciones, aprobar interacciones, confirmar cambios o regalos y a su vez comunicarse una vez aprobado el intercambio/regalo con otros usuarios para lograr concretar la transacción.

**Usuario Admin:**  
Perfil que estará a cargo de moderar la actividad en el aplicativo. Podrá crear/modificar/eliminar usuarios, publicaciones, match y tags.

**2.4. Restricciones**

- Solo se podrá crear usuario con una cuenta de correo electrónico proporcionada por DUOC UC para verificar que sea parte de la comunidad educativa.

- No se podrá publicar artículos peligrosos o que no cumplan con las normas y estándares requeridos por el sistema (que no pongan en riesgo ni ofendan a la comunidad, como armas, artículos tóxicos, artículos ofensivos etc)

-Nuestro proyecto se acoge a la legislación Chilena, la ley 18.045 "Ley de mercado de valores" que aborda el trading y es deber del usuario informarse leyendo nuestro contrato de uso, que se acoge a dicha ley basando nuestras políticas de seguridad de trading.

- Solo se podrá acceder al aplicativo mediante un navegador de internet.

- El límite de tiempo de respuesta con el servidor web no debe ser más de 20 segundos

- Los lenguajes de programación serán HTML5, CSS3, Javascript, Python con el Framework DJango, Java, Bootstrap y PL/SQL con mariadb o sql developer.

**2.5. Suposiciones y Dependencias**

* El sistema depende de conexión a internet para funcionar. No funciona de manera local o escritorio
* Se tendrán que hacer mantenciones a los equipos y servidores al menos 1 vez al mes, idealmente la última semana del mes
* Se necesita que el dispositivo tenga un navegador de internet que pueda procesar protocolos HTML para lograr navegar en el sistema.

**2.6. Requisitos Futuros**

Se espera que el aplicativo pueda mejorar su procesamiento de búsqueda categorizando de mejor manera los productos y a su vez poder llegar a categorizar de manera predictiva las categorías o tags asociados al producto, añadiendo validaciones de políticas de seguridad de trading.

**3. Requisitos Específicos**

**3.1 Requisitos comunes de las interfaces**

**3.1.1 Interfaces de usuario**

* Sistema Amigable con colores corporativos asociados a la paleta de colores de DUOC UC, que cuente con comandos intuitivos, de fácil entendimiento para cada acción que desee realizar el usuario.

**3.1.2 Interfaces de hardware**

Se utilizará un servidor con las siguientes características:  
Servidor local: Windows 11 Home 64-bit., Procesador Intel® Core™ i5-1135G7 Quad-core (4 núcleos)) 2,40 GHz., NVIDIA® GeForce® MX 350 con 2 GB de memoria dedicada., 35,6 cm (14") Full HD (1920 x 1080) 16:9., 8 GB, DDR4 SD RAM., 512 GB SSD.

**3.1.3 Interfaces de software**

* ***Ionic:*** Framework de código abierto utilizado para desarrollar aplicaciones móviles y web híbridas y multiplataforma utilizando tecnologías como HTML, CSS y JavaScript. Permite construir aplicaciones con una sola base de código que funcionan tanto en Android como en iOS.
* ***Angular:*** Framework de desarrollo web de código abierto basado en TypeScript, creado por Google. Se utiliza para construir aplicaciones web de una sola página (SPA) y proporciona una arquitectura modular basada en componentes, lo que facilita la creación de aplicaciones web complejas.
* ***Capacitor:*** Runtime de código abierto creado por Ionic que permite a las aplicaciones web acceder a las API nativas de los dispositivos móviles. Es compatible con iOS, Android y la web, lo que permite ejecutar el mismo código tanto en dispositivos móviles como en navegadores.
* ***MariaDB:*** Sistema de gestión de bases de datos relacional de código abierto, derivado de MySQL. Es conocido por su alta compatibilidad con MySQL y ofrece mejoras en términos de rendimiento, seguridad y nuevas características, siendo ampliamente utilizado en aplicaciones web y empresariales.
* ***JavaScript:*** Lenguaje de programación interpretado utilizado para agregar interactividad a sitios web y aplicaciones web. Es un lenguaje clave para el desarrollo del frontend y backend (mediante Node.js), y es compatible con múltiples frameworks y bibliotecas como Angular, React y Vue.js.

**3.2 Requisitos funcionales**

RF1- Autenticar usuario al iniciar sesión

RF2- Registrarse como usuario

RF3- Restablecer contraseña de usuario

RF4- Agregar imagen a perfil de usuario

RF5- Crear publicacion de producto cambio/regalo

RF6- Ver lista de publicaciones realizadas por el usuario

RF7- Buscar artículos publicados

RF8- Revisar publicaciones de otros usuarios

RF9- Seleccionar tipo de transacción a realizar "cambio/regalo" al revisar una publicaciona activa

RF10- Revisión y confirmación de match entre publicaciones o solicitud de regalo

RF11- Comunicación entre usuarios a partir de un match de publicaciones o solicitud de regalos a través de foro interno (post)

RF12- Reactivación de publicaciones eliminadas

RF13- Administrador de usuarios, publicaciones y tags

RF14- Validaciones de políticas de seguridad de trading.

**3.3 Requisitos no funcionales (Organización) y de calidad (Producto)**

**3.3.1 Requisitos de Rendimiento o Eficiencia**

Se establece que el tiempo de carga del aplicativo web no debe superar los 5 segundos para una respuesta, lo que implica un rendimiento eficiente del sistema para una experiencia de usuario satisfactoria.

**3.3.2 Requisitos de Seguridad**

Se refiere a la protección de la información y acceso no autorizado al sistema (solo usuarios con CORREO DUOC UC), como la restricción de usuarios duplicados y restricciones de publicaciones.

**3.3.3 Requisitos de Usabilidad**

Se refiere a la facilidad de uso y accesibilidad del sistema, como la paleta de colores acorde con el entorno gráfico de DUOC UC, interfaz de usuario fácil e intuitiva, funcionamiento multiplataforma y en navegadores web, mensajes de error y tiempo de respuesta.

**3.3.4 Requisitos de Disponibilidad**

Los usuarios del sistema deben poder acceder al aplicativo en cualquier servicio de ésta en cualquier momento del día contando con internet y el equipo necesario.

**3.3.5 Requisitos de Portabilidad**

El aplicativo debe ser diseñado y desarrollado de manera tal que se ajuste a diferentes tamaños de pantalla, resoluciones y sistemas operativos, permitiendo así su correcta ejecución y navegación en cualquier dispositivo y sistema operativo. Además, es importante considerar el uso de herramientas de desarrollo multiplataforma para asegurar la portabilidad del aplicativo en diferentes plataformas y navegadores web.

**3.3.6 Requisitos de Mantenibilidad**

Se refiere a la facilidad de mantener y actualizar el sistema, así como la validación de los datos del login de cada usuario.

**3.3.7 Requisitos de Funcionalidad**

El sistema no debe permitir la creación de publicaciones de material no educativo con palabras restrictivas claves para un primer control, como "armas, cuchillos, tóxico", etc.

**3.4 Requisitos No funcionales Organizacionales**

1. El sistema debe contar con el logo de DUOC UC para poder mostrar su vinculación con su comunidad.
2. El aplicativo debe contar con la paleta de colores acorde a la utilizada en los sistemas oficiales de DUOC UC, como el color blanco, amarillo, azul marino, plomo y negro como base inicial de la interfaz de usuario.

**4. Propuesta de Planificación**

**4.1 Descripción general acerca de la Planificación**

A continuación se entregará el detalle y de la planificación proyectada para el desarrollo desde sus fase inicial de planificación hasta la fase de implementación y cierre.

**Fase de planificación (días 1 - 8) :**  
En esta fase se definirán los objetivos, requisitos y alcance del proyecto, así como los recursos necesarios para su realización. Se establecerá un plan de trabajo detallado y se asignan roles y responsabilidades al equipo de trabajo.

**Fase de análisis y diseño (días 8-21):**  
En esta fase se recopiló la información necesaria para el diseño del proyecto y se realizó el análisis de requisitos. Se definirá la arquitectura del sistema y se crearán los diseños de interfaz de usuario y base de datos. También se definirán las especificaciones técnicas y funcionales del proyecto.

**Fase de desarrollo (días 21-55):**  
En esta fase se implementará el ambiente de desarrollo y se crearán los scripts de base de datos y consultas. Se llevará a cabo la creación de las funcionalidades y especificaciones del sistema, y se integrarán los diferentes módulos del proyecto. Además, se realizará la implementación de la interfaz de usuario utilizando HTML, CSS y JavaScript.

**Fase de pruebas de QA (días 21-45):**  
En esta fase se realizan pruebas de calidad para asegurar el correcto funcionamiento del sistema. Se comprobará que todas las funcionalidades se ejecuten correctamente y se solucionarán los errores encontrados. También se realizarán pruebas de rendimiento y seguridad del sistema.

**Fase de implementación y cierre (días 45-56):**  
En esta fase se realizará la implementación del sistema en producción y se llevará a cabo la capacitación de los usuarios finales. Se documentará todo el proceso de desarrollo del proyecto y se realizará una revisión final para asegurarse de que se han cumplido todos los objetivos y requisitos del proyecto. Finalmente, se procederá al cierre del proyecto.

**4.1.2 Definición del Equipo de Trabajo**

| NOMBRE RECURSO | SIGLA | | | | | | | ROL | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Jorge Pavez | JP | | | | | | | Jefe de Proyecto | | | | | | | | | | | |
| Carlos Muñoz | APC | | | | | | | Programador | | | | | | | | | | | |
| Matías Carvajal | DBA | | | | | | | Administrador BD | | | | | | | | | | | |
| Matías Carvajal | QA | | | | | | | Calidad Y Testing | | | | | | | | | | | |
| Carlos Muñoz | DG | | | | | | | Diseñador | | | | | | | | | | | |

**4.1.3 Definición de Actividades principales del Proyecto**

Las principales etapas del proyecto son las siguientes:

* Fase de planificación
* Fase de análisis y diseño
* Fase de desarrollo
* Base de datos
* autenticación y seguridad
* Catálogos de productos
* Carrito de canje
* Gestión de órdenes
* Panel de administración
* Módulo de reporte
* Fase de prueba y QA
* Fase de implementación y cierre

**4.1.4 Resumen Costos del Desarrollo del Proyecto**

* Costos por FASE

| COSTO POR FASE | |
| --- | --- |
| Fase de Planificación | $ 622.192 |
| Fase de Análisis y Diseño | $ 1.555.464 |
| Fase de Desarrollo | $ 17.421.160 |
| Fase de QA | $ 2.999.880 |
| Fase de Implementación y Cierre | $ 999.960 |
| **TOTAL HH FASES** | **$ 23.598.656** |

* Costos por Actor o Rol

| NOMBRE | COSTO HH POR ROL | |
| --- | --- | --- |
| Jorge Pavez | Jefe de Proyecto | $ 6.346.304 |
| Carlos Muñoz | Programador | $ 4.199.832 |
| Matías Carvajal | Administrador BD | $ 4.382.016 |
| Matías Carvajal | Calidad Y Testing | $ 4.333.160 |
| Carlos Muñoz | Diseñador | $ 4.337.344 |
| — | **TOTAL HH** | **$ 23.598.656** |