Contenido

[1 Paquete de trabajo: Análisis 5](#_Toc145505601)

[1.1 Especificación de requisitos y modelado 5](#_Toc145505602)

[2 Diseño 7](#_Toc145505603)

[2.1 Arquitectura de software 7](#_Toc145505604)

[2.2 Diseño de Base de datos 9](#_Toc145505605)

[2.3 Modelado del diseño con UML 11](#_Toc145505606)

[3 Implementación 13](#_Toc145505607)

[3.1 Configuración de servidores 13](#_Toc145505608)

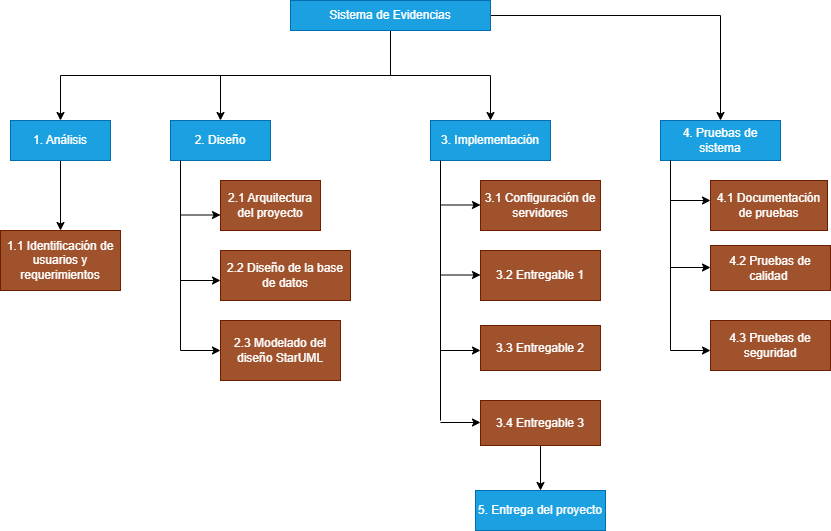
[3.2 Entregable 1 14](#_Toc145505609)

[3.3 Entregable 2 15](#_Toc145505610)

**EDT: Estructura de desglose de trabajo del proyecto**

Una estructura de desglose de trabajo (EDT), también conocida por WBS (*Work Breakdown Structure*) es la descomposición de un proyecto en paquetes de trabajo (conjuntos de tareas) que organiza y define el alcance total del proyecto en términos de costo y tiempo (Carta Gantt). La EDT subdivide el trabajo del proyecto en porciones de trabajo más pequeñas y fáciles de manejar, donde cada nivel descendente representa una definición cada vez más detallada del trabajo del proyecto. En la siguiente figura, las cajas de color amarillo representan paquetes de trabajo porque cada una de ellas representa un conjunto de tareas, por ejemplo, el paquete “1.1 Especificación de requisitos” representa el siguiente conjunto de tareas:

* Comprender el acta de constitución del proyecto
* Preparar una entrevista
* Realizar entrevista
* Analizar entrevista
* Especificar los requisitos identificados en las entrevistas
* Validar los requisitos



# Paquete de trabajo: Análisis

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Especificación de requisitos y modelado | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| El documento de especificación de requisitos se compone de varios puntos clave que sirven como introducción al proyecto en curso:   1. Identificación de requisitos del sistema: Se han identificado seis categorías de requisitos esenciales para el funcionamiento del software, que incluyen requisitos del administrador, del usuario, del revisor, del director de carrera, el comité y del DAC. 2. Análisis de requisitos: Este paso implica la comprensión de los requisitos del sistema, así como la detección de posibles problemas o conflictos entre ellos. 3. Especificación de requisitos: Cada categoría de requisitos se desglosa en una lista detallada de requisitos específicos. Por ejemplo, los requisitos del usuario incluyen funcionalidades como la capacidad de ingresar y visualizar información. 4. Validación de requisitos. La validación de los requisitos se lleva a cabo mediante la revisión y aprobación por parte del cliente y otros interesados en el proyecto. 5. Gestión de requisitos: La gestión de requisitos abarca la administración de cambios en los requisitos a lo largo de la duración del proyecto e incluye la identificación, el análisis, la especificación, la validación y el control de los mismos. | | **Suposiciones**  1. Se asume que los requisitos descritos en este documento son estables una vez que sean aprobados por el cliente.  2. Cualquier petición de cambios de requisitos debe seguir un conducto regular de aprobación.  3. Se asume que los usuarios estarán dispuestos a colaborar activamente en la carga y gestión de evidencias en el sistema.  4. Disponibilidad horaria de todos los Stakeholders.  **Restricciones**  1. Tiempo = 2 semanas para finalizar la especificación de requisitos.  2. El sistema debe ser desarrollado con API. El back-end que se implementará será con el framework NodeJS y en front-end, el framework que se implementará será Angular.  3. El plazo para completar el proyecto es a fines de diciembre.  4. Regulaciones y políticas de la Universidad, en estos casos la UCM puede poner restricciones en la recopilación, almacenamiento y uso de datos de evidencia. |
| **Criterios de aceptación** | | |
| Aprobación del cliente de la especificación de requisitos y el acta de constitución | | |
| **Actividades** | | |
| 1 | Entrevista 1: Levantamiento de requisitos | |
| 2 | Especificación de requerimientos (versión 1) | |
| 3 | Entrevista 2: Validación y levantamiento | |
| 4 | Validación completa de requisitos | |
| 5 | Entrega versión final de requisitos aprobado | |
| 6 | Modelado de caso de uso | |
| Administración | | |
| Responsable: Matías Morales | | |
| Aprobador: Marco Toranzo | | |
| Duración: 2 semanas | | |
| Costo: $ 300.000 | | |

# Diseño

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Arquitectura de software | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| El SDE es una plataforma informática desarrollada para recopilar, almacenar y gestionar evidencias dentro de la universidad. Esta plataforma estará implementada en un sitio web que requerirá un inicio de sesión para que los usuarios autorizados puedan interactuar con las evidencias, publicar y modificar según el cargo del usuario. Para lograr esto, se ha diseñado la arquitectura utilizando un modelo UML y se procederá a desarrollar el back-end en la plataforma NodeJs, mientras que en el front-end se implementará en Angular.  Partes de la arquitectura:  1.Se realizará de acuerdo a lo solicitado por el entrevistado.  2.Se seguirá el diseño del modelo UML para la creación de ventanas, menú, y acciones dentro del SDE.  3. Todas las facultades tendrán acceso a usar el SDE, para mejorar el proceso de evidencias a nivel UCM.  4. El acceso al SDE solicitará el Rut y contraseña.  5. Se garantiza que el diseño, colores y logotipo respetarán los principios de la universidad  6. Será compatible con usuarios que naveguen desde el PC  Además, se esforzará por ofrecer una experiencia de usuario sencilla, proporcionando tutoriales para asegurar una correcta utilización y adaptación por parte de los usuarios. | | **Suposiciones**   1. Estructura de plataforma aprobada y son estables una vez aprobados por el cliente 2. Utilización de NodeJS para back-end y Angular para front-end. 3. Diseño y colores aprobados y son estables una vez aprobados por el cliente. 4. Se asume que los usuarios no generarán una perspectiva negativa por la plataforma SDE. 5. Se está abierto a modificaciones siguiente el conducto regular siempre y cuando este dentro del plazo.   **Restricciones**  1. El tiempo para el desarrollo es igual a cinco semanas.  2.Se debe utilizar sólo NodeJS y Angular para el desarrollo.  3. El presupuesto no debe exceder los $170.000 mil pesos.  4. No se harán modificaciones solicitadas por el cliente fuera del plazo, ya que se debe seguir el conducto regular.  5. Los colores y diseños deben ser únicos, siguiendo la línea de la UCM. |
| **Criterios de aceptación** | | |
| Aprobación del arquitecto de software e ingenieros de software | | |
| **Actividades** | | |
| 1 | Definiciones de requisitos del SDE. | |
| 2 | Interpretar y analizar los casos del sistema de evidencia según las solicitudes por el entrevistado. | |
| 3 | Identificación de los componentes principales, relaciones y como se ejecutan entre sí. | |
| 4 | Usar las tecnologías adecuadas para la creación del front-end y back-end para el funcionamiento del SDE. | |
| 5 | Usar las estructuras de datos ingresadas en la base de datos. | |
| 6 | Diseñar login de acceso para usuarios. | |
| 7 | Diseñar registro de almacén de evidencias en base de datos. | |
| 8 | Cumplir con los protocolos de diseño, logo y colores de la Universidad Católica Del Maule. | |
| Administración | | |
| Responsable: Matías Morales | | |
| Aprobador: Marco Toranzo | | |
| Duración: 5 semanas. Esto será realizado en forma conjunto con las actividades de los otros paquetes de diseño. | | |
| Costo: $ 170.000 | | |
| Herramientas: Angular, NodeJS, PostgreSQL. | | |

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Diseño de Base de datos | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Para el diseño de la base de datos se iniciará con la definición de los requisitos donde se identificarán las necesidades y expectativas que tendrá el cliente con respecto al sistema de información. A continuación, se hará un diseño conceptual donde se definirá un esquema que relacione los conceptos que serán usados en el sistema. Para luego seguir con un diseño lógico que transformara el modelo conceptual en un modelo lógico. A partir de esto se realizará un diseño físico donde se decidirá como se implementará el modelo lógico obtenido anteriormente para el sistema de la base de datos. A continuación, se hará una implementación, es decir que aquí se creara la base datos utilizando el gestor de base de datos que como equipo se seleccionara. Luego de haber creado dicha base de datos se realizarán sus respectivas pruebas para verificar el funcionamiento de la base de datos. A continuación de las pruebas y corroborando que todo esté funcionando correctamente, se desplegara la base de datos para que quede en su completo funcionamiento. Cuando el sistema esté funcionando se hará de vez en cuando un mantenimiento, donde se podrán realizar ajustes y actualizaciones acorde a lo que sea necesario. | | **Suposiciones**  1. Especificación de requisitos aprobada y son estables.  2. Los usuarios estarán dispuestos a la carga y gestión de datos de base de datos.  **Restricciones**   1. El back-end será implementado con el framework NodeJS 2. El front-end será implementado con el framework Angular 3. El gestor de base de datos será PostgreSQL 4. El plazo de diseño será de 1 semana 5. La Universidad Católica del Maule puede agregar restricciones en la recopilación y uso de la información almacenada en la base de datos. 6. La privacidad y seguridad de los datos se deberá adjuntar a los estándares de seguridad |
| **Criterios de aceptación** | | |
| Diseño de la Base de Datos satisface toda la especificación de requisitos entregadas por el cliente | | |
| La Base de Datos funcione correctamente | | |
| **Actividades** | | |
| 1 | Comprensión de la especificación de requisitos | |
| 2 | Identificación de las tablas | |
| 3 | Comprensión de la relación que existirá entre las tablas | |
| 4 | Diseño y validación del diseño de base de datos acorde con la especificación de requisitos | |
| 5 | Implementación de la Base de Datos en el gestor seleccionado (PostgreSQL) | |
| 6 | Comprobación de errores y realización de pruebas a la Base de Datos | |
| 7 | Despliegue y mantenimiento de la Base de Datos para su correcto funcionamiento | |
| Administración | | |
| Responsable: Matías Morales | | |
| Aprobador: Marco Toranzo | | |
| Duración: 1 semana. Esto será realizado en forma conjunto con las actividades de los otros paquetes de diseño. | | |
| Herramientas: PostgreSQL, Ordenadores con Windows 10 | | |
| Costo: $ 170.000 | | |

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Modelado del diseño con UML | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Para la creación del modelo UML, se utilizarán 2 diagramas pertinentes:  1.Diagrama de casos de Uso: Se utilizará para definir interacciones entre entidades (Usuarios, administradores, etc.) y el sistema.  2.Diagrama de Clases: Permitirá representar la estructura de clases, atributos y relaciones, especialmente para categorías como “Informes” y “Materiales Educativos”.  Estos diagramas son esenciales para especificar la implementación de métodos y lograr una representación completa y clara del diseño del software. | | **Suposiciones**  1. Se ha obtenido la aprobación de la especificación de requisitos.  2. Se ha optado por utilizar un enfoque orientado a objetos como paradigma de diseño.  3. Se cuenta con un equipo o recursos disponibles para llevar a cabo la tarea de diseño de UML.  4. Acorde a la entrevista realizada, se obtendran datos para realizar el modelo del diseño con UML.  **Restricción**  1.Existe un plazo limitado de una semana para completar esta fase de diseño.  2. Los recursos disponibles, como tiempo y personal, son limitados y deben ser utilizados eficazmente. |
| **Criterios de aceptación** | | |
| Los diagramas han sido validados y aprobados por el Cliente y los Stakeholders | | |
| **Actividades** | | |
| 1 | Entender los requisitos especificados | |
| 2 | Reconocer las categorías | |
| 3 | Crear diagrama de casos de uso | |
| 4 | Definir clases y atributos | |
| 5 | Crear diagrama de clases | |
| 6 | Validar y refinar el modelo | |
| 8 | Documentar el modelo | |
| Administración | | |
| Responsable: José Sepúlveda | | |
| Aprobador: Marco Toranzo | | |
| Duración: 1 semana. Se llevará a cabo en una semana y se coordinará con las tareas de los demás conjuntos de diseño. | | |
| Costo: $ 70.000 | | |

# Implementación

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Configuración de servidores | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Configuración de servidores para la creación del software. Para esto se va a adquirir un servidor y se usará el entorno NodeJS. | | **Suposiciones**  1. Framework, gestor de base de datos, versión del sistema operativo y hardware seleccionados.  2. Recurso disponible para la compra de un servidor privado.  **Restricciones**  1. Uso de Framework NodeJS y Angular  2. Gestor de la base de datos PostgreSQL  3. Versión del sistema operativo Windows 10  4. Uso de computadoras |
| **Criterios de aceptación** | | |
| Debe ser aprobado por el equipo de desarrollo | | |
| **Actividades** | | |
| 1 | Selección del servidor | |
| 2 | Adquisición del servidor | |
| 3 | Actualización de las dependencias del sistema operativo | |
| 4 | Instalación de los lenguajes de programación | |
| 5 | Instalación de gestor de base de datos | |
| 6 | Seguridad del entorno de trabajo | |
| 7 | Instalación y configuración de un entorno integrado | |
| 8 | Crear repositorio privado en GitHub | |
| 9 | Creación de proyecto en Trello (Scrum) e invitar a los integrantes del proyecto | |
| Administración | | |
| Responsable: José Sepúlveda | | |
| Aprobador: Marco Toranzo | | |
| Duración: 2 semana | | |
| Costo: $ 35.000 | | |

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Entregable 1 | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Este paquete incluye el cumplimiento de los requisitos del cliente, los cuales serán, la creación de los CRUD de las tablas, priorizando las tablas que tengas menos relaciones entre sí, es decir, que dichas tablas no posean claves foráneas. No obstante, al terminar los CRUD de las tablas simples de igual manera se continuará con las demás tablas completando la mitad de los requerimientos solicitados por el cliente. | | **Suposiciones**  1. Lenguaje, framework, SGBD, versión del sistema operativo y hardware funcionando correctamente.  **Restricciones**  1. Cuatro semanas para finalizar el entregable.  2. Para el desarrollo del front-end se utilizará el framework Angular.  3. Para el desarrollo del back-end se utilizará el framework NodeJS. |
| **Criterios de aceptación** | | |
| Aprobado por propietario del producto, y los equipos de desarrollo y seguridad | | |
| **Actividades** | | |
| 1 | Selección de proveedor de VPS (servidor privado virtual) | |
| 2 | Compra del servidor | |
| 3 | Actualización de las dependencias del sistema operativo | |
| 4 | Instalación de los lenguajes de programación | |
| 5 | Instalación de servidor de base de datos y creación de usuario con privilegios | |
| 6 | Seguridad del entorno de trabajo | |
| 7 | Instalación y configuración de IDE | |
| 8 | Crear repositorio privado en GitHub | |
| 9 | Creación de proyecto en Trello | |
| 10 | Codificación y pruebas unitarias | |
| 11 | Revisión del entregable por el equipo de desarrollo del producto | |
| 12 | Incorporar ajustes dados por el equipo de desarrollo del producto | |
| 13 | Presentación el entregable al cliente y registrar sus observaciones | |
| 14 | Aplicación de control de versión e incorporación a la rama maestra de GitHub (con la aprobación de QA) | |
| Administración | | |
| Responsable: Matías Morales | | |
| Aprobador: Marco Toranzo | | |
| Duración: 4 semanas. | | |
| Costo: $ 400.000 | | |

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Entregable 2 | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Este paquete considera la implementación del núcleo del sistema, es decir, todo lo que mantiene en pie y funcionamiento en la página. Se entregarán los procesos principales del sistema, los cuales son:  -Gestión de usuarios.  -Almacenamiento de evidencias.  -Reporte y análisis.  Se crearan los CRUD de las tablas faltantes ya mencionadas en el entregable 1, finalizando con el requisito más importante y riguroso, la cual consiste en la parte principal del problema que es la tabla de Evidencias. | | **Suposiciones**  1. Lenguaje, framework, SGBD, versión del sistema operativo y hardware funcionando correctamente  **Restricciones**  1. Cuatro semanas para finalizar el entregable.  2. Para el desarrollo del front-end se utilizará el framework Angular.  3. Para el desarrollo del back-end se utilizará el framework NodeJS. |
| **Criterios de aceptación** | | |
| Aprobado por propietario del producto, y los equipos de desarrollo y seguridad | | |
| **Actividades** | | |
| 1 | Pruebas de integración y funcionales del sistema | |
| 2 | Revisión del entregable por el propietario del producto | |
| 3 | Incorporar ajustes dados por el propietario del producto | |
| 4 | Presentación el entregable al cliente y registrar sus observaciones | |
| 5 | Aprobación del producto | |
| 6 | Soporte y mantenimiento | |
| 7 | Aplicación de control de versión e incorporación a la rama maestra de github (con la aprobación de QA) | |
| Administración | | |
| Responsable: Matías Morales | | |
| Aprobador: Marco Toranzo | | |
| Duración: 4 semanas. | | |
| Costo: $ 500.000 | | |

|  |  |  |
| --- | --- | --- |
| **Código y nombre del paquete de trabajo**  **Código de cuenta** | | |
| Entregable 3 | | |
| **Descripción del trabajo** | | **Suposiciones y restricciones** |
| Este paquete considera la finalización de la creación de la página web, corrigiendo errores notificados por el cliente en el entregable 2 tanto con back-end y front-end y obteniendo un resultado del SDE.  1.Se entregarán manuales de uso  2.Vídeo tutorial para el correcto uso del SDE por los usuarios. | | **Suposiciones**  1. Funcionamiento correcto del back-end y front-end  2. Entregable 2 está funcionando correctamente.  3. Funcionamiento correcto del servicio digital.  4.Perfecto funcionamiento de hardware y software para la correcta ejecución del SDE.  5. El director de escuela como el comité podrán revisar las evidencias subidas por el responsable o profesor.  **Restricciones**  1. Dos semanas para finalizar el entregable 3  2.Los costos no deben superar los $50.000  3.Una semana de holgura por casos de emergencias. |
| **Criterios de aceptación** | | |
| Aprobado por el cliente del producto, y los equipos de desarrollo y seguridad | | |
| **Actividades** | | |
| 1 | Las maquinas donde se va a ejecutar el SDE deben tener un correcto funcionamiento, por lo que se procede a revisar hardware y software. | |
| 2 | Realización de pruebas para casar fallas en el SDE. | |
| 3 | Solucionar fallas obtenidas en caso de encontrarse. | |
| 4 | Realización de manuales de uso para un correcto uso por parte de los usuarios. | |
| 5 | Incorporar video tutorial en el SDE para los usuarios. | |
| 6 | Reuniones diarias para organizar presentación del SDE al cliente. | |
| 7 | Mostrar la versión final del SDE al cliente. | |
| Administración | | |
| Responsable: Jonathan Molina | | |
| Aprobador: Marco Toranzo | | |
| Duración: 3 semanas. | | |
| Costo: $ 50.000 | | |