



Nombre: CARRION JIMENEZ STALIN ALEXIS

Práctica No. 1

Código: DSUP13-GP-01

Título: Requerimientos de Software

Tiempo de duración: 120 minutos

Fundamentos:

El análisis de requisitos de software es una etapa crucial en el ciclo de vida del desarrollo de software, ya que define lo que el sistema debe hacer y las restricciones bajo las cuales debe operar. Según Alarcón (2000), los requisitos de software son las necesidades y condiciones que un nuevo sistema debe cumplir. Estos requisitos pueden ser funcionales, describiendo las funciones específicas que el sistema debe realizar, y no funcionales, especificando criterios que pueden ser usados para juzgar la operación de un sistema, como el rendimiento, la seguridad y la usabilidad

El análisis de requisitos de software es una etapa crucial en el ciclo de vida del desarrollo de software, ya que define lo que el sistema debe hacer y las restricciones bajo las cuales debe operar. Según Alarcón (2000), los requisitos de software son las necesidades y condiciones que un nuevo sistema debe cumplir. Estos requisitos pueden ser funcionales, describiendo las funciones específicas que el sistema debe realizar, y no funcionales, especificando criterios que pueden ser usados para juzgar la operación de un sistema, como el rendimiento, la seguridad y la usabilidad.

El modelado de requisitos es otra técnica clave que ayuda a clarificar y documentar los requisitos. Se pueden utilizar diversos modelos, como diagramas de casos de uso, diagramas de actividades y diagramas de entidad-relación, que ayudan a visualizar cómo interactúan los componentes del sistema. Según Debrauwer, Ferrandez y Evain (2021), el uso de modelos permite una mejor comunicación entre el equipo de



desarrollo y los stakeholders, asegurando que todos tengan una comprensión común del sistema. Además, los modelos facilitan la identificación de requisitos incompletos, inconsistentes o contradictorios.

La validación y verificación de requisitos son pasos esenciales para asegurar que los requisitos definidos sean correctos y factibles. Debrauwer, Ferrandez y Evain (2021) subrayan que la validación se centra en revisar si los requisitos capturados realmente reflejan las necesidades de los stakeholders, mientras que la verificación asegura que los requisitos sean claros, sin ambigüedades y técnicamente realizables. La validación puede incluir revisiones, prototipos y simulaciones, mientras que la verificación puede involucrar la creación de casos de prueba para verificar que los requisitos se cumplen en el sistema final.

Objetivos

- Reconocer los requerimientos de software previo a la diagramación UML de un sistema académico.
- Identificar el tipo de requerimiento que se va a hacer uso el sistema.
(Funcionales, no funcionales).
- Definir un lenguaje de modelado visual fácil de aprender, pero enriquecido en significado.

Materiales y Herramientas

- Internet
- Software (StarUML)
- Computador



Normas de Seguridad

Uso adecuado del equipamiento

Además, deben manipular los dispositivos con cuidado, evitando movimientos bruscos y asegurándose de seguir las instrucciones específicas para el uso de hardware y software

Mantenimiento de la Seguridad Física

Los estudiantes deben mantener el área de trabajo ordenada y libre de obstáculos que puedan causar accidentes

Cumplimiento de las Políticas de la Institución

Respetar los horarios de uso del laboratorio y las normas de conducta establecidas por la institución.

Supervisión

Realizar todas las pruebas de funcionamiento del sistema bajo la supervisión del docente.

Inspección Previa

Inspeccionar visualmente todos los componentes antes de su uso para detectar posibles daños o defectos.



Preparación Previa del Estudiante

Fundamentos de Ingeniería de Software

Es esencial que los estudiantes tengan una comprensión básica de la ingeniería de software, incluyendo sus fases y procesos. Deben conocer el ciclo de vida del desarrollo de software (SDLC), que incluye etapas como la planificación, análisis, diseño, implementación, pruebas y mantenimiento. Este conocimiento proporciona un marco general que les ayuda a comprender dónde encaja el análisis de requisitos en el contexto más amplio del desarrollo de software.

Principios de Gestión de Proyectos

Antes de la clase práctica, los estudiantes deben leer y estudiar sobre el análisis de requisitos está estrechamente relacionado con la gestión de proyectos. Los estudiantes deben estar familiarizados con los principios básicos de gestión de proyectos, como la definición de alcance, la planificación de tareas, la asignación de recursos y el seguimiento del progreso. Herramientas y técnicas como diagramas de Gantt, metodologías ágiles y SCRUM también son útiles para coordinar y gestionar el proceso de recopilación y análisis de requisitos.

Técnicas de Recolección de Información

Es importante que los estudiantes sepan cómo recolectar información de diversas fuentes. Esto incluye habilidades en la realización de entrevistas, encuestas, talleres y grupos focales, así como en la observación directa y el análisis de documentos existentes. Estas técnicas son cruciales para obtener una comprensión clara y completa de los requisitos de los stakeholders.

Modelado y Documentación de Requisitos

Los estudiantes deberán conocer previamente sobre el modelado de requisitos. Esto incluye el uso de diagramas de casos de uso, diagramas de secuencia, diagramas de



actividad y otros modelos que ayudan a visualizar y comunicar los requisitos. Además, deben estar capacitados en la redacción de documentos de requisitos claros y precisos, utilizando plantillas y estándares de la industria, como IEEE 830.

Habilidades de Comunicación

La capacidad de comunicarse de manera efectiva con los stakeholders es fundamental en el análisis de requisitos. Los estudiantes deben desarrollar habilidades tanto verbales como escritas para poder interactuar con usuarios finales, clientes, desarrolladores y otros interesados de manera clara y efectiva. La empatía y la capacidad de escuchar activamente también son esenciales para comprender verdaderamente las necesidades y expectativas de los stakeholders.



Procedimiento

1. Analizar las características y requerimientos que el software debe cumplir. Ver anexo 1.
2. Identificar los requerimientos funcionales y no funcionales ver anexo (Descripción general del sistema), previo a la lectura de los conceptos.
3. Requerimientos funcionales y no funcionales. Ver anexo (Requerimientos del sistema).
4. Documentar los requerimientos funcionales y no funcionales identificados en la lectura del anexo (Descripción general del sistema). Y colocar en la tabla 1.
5. Describir los requerimientos no funcionales y no funcionales.
6. Analizar la mejor metodología de desarrollo de software que para el sistema académico, hable de cada una de sus etapas y escribir en la tabla 3.

Descripción de Resultados Obtenidos

Requerimientos Funcionales y No Funcionales

Tipo de Requerimiento	Requerimiento
Funcional	Autenticación de usuarios mediante intranet (docente o administrador).
Funcional	Visualizar alumnos matriculados por año lectivo y grado.
Funcional	Registrar, modificar y listar notas por trimestre y asignatura.
Funcional	Registrar, modificar y listar faltas justificadas e injustificadas.



Tipo de Requerimiento	Requerimiento
Funcional	Generar reportes parciales y finales de notas y asistencia.
Funcional	Mostrar alumnos aprobados y reprobados por periodo.
Funcional	Registrar nuevos usuarios, materias, docentes, años de básica y periodos lectivos.
Funcional	Eliminar y actualizar datos de usuarios, asignaturas, años de básica y docentes.
Funcional	Matricular alumnos y realizar cambios de paralelo.
Funcional	Anular procesos realizados por los alumnos.
Funcional	Generar proceso de aprobación académica.
No Funcional	Accesibilidad desde cualquier sistema operativo mediante navegador web.
No Funcional	Interfaz amigable, clara y adecuada para cada rol de usuario.
No Funcional	Compatible con hardware institucional actual.
No Funcional	Comunicación segura y estable mediante protocolo HTTP sobre TCP/IP.
No Funcional	Bajo consumo de memoria y almacenamiento.



Tipo de Requerimiento	Requerimiento
No Funcional	Alta disponibilidad y fiabilidad del sistema.
No Funcional	Sistema escalable para nuevos usuarios, períodos y asignaturas.

Requerimientos Funcionales

Los requerimientos funcionales identificados indican que el sistema debe:

- Permitir el acceso diferenciado según el rol (docente o administrador)
- Gestionar calificaciones y asistencias.
- Emitir distintos tipos de reportes académicos.
- Administrar usuarios, asignaturas, periodos y grados escolares.
- Matricular y gestionar cambios en la información de estudiantes.
- Automatizar procesos académicos como aprobación de estudiantes.

Requerimientos No Funcionales

Los no funcionales garantizan que el sistema:

- Sea accesible desde cualquier sistema con navegador web moderno.
- Funcione con el hardware actual de la institución.
- Utilice protocolos estándar y confiables de comunicación.
- Sea fácil de usar por sus distintos perfiles de usuario.
- Tenga buena respuesta y bajo consumo de recursos.
- Permita crecimiento sin rediseños complejos (escalabilidad).



Metodología de desarrollo

Etapa	Descripción
Inicio del Proyecto	Se identifican los objetivos del sistema, los stakeholders, y se arma el Product Backlog (lista priorizada de funcionalidades).
Planificación del Sprint	Se seleccionan los requerimientos más importantes para desarrollarse en un Sprint (iteración de 2 a 4 semanas).
Desarrollo	Se desarrolla el sistema por módulos funcionales, permitiendo avances incrementales en cada sprint.
Daily Meeting	Reuniones diarias para revisar el avance, identificar obstáculos y coordinar tareas.
Revisión del Sprint	Se presenta al cliente o usuarios una versión funcional para recibir retroalimentación.
Retrospectiva	El equipo analiza qué funcionó y qué se puede mejorar en el siguiente sprint.



Entrega final	Tras varios sprints, se entrega el sistema completo y probado.
----------------------	--

Conclusiones

- La correcta identificación de los requerimientos funcionales y no funcionales es esencial para garantizar que el sistema académico cumpla con las expectativas y necesidades reales de los usuarios, permitiendo una gestión eficiente de calificaciones, asistencias y procesos administrativos.
- La arquitectura basada en web proporciona una solución accesible, compatible y de bajo costo para instituciones educativas, ya que permite el uso del sistema desde cualquier dispositivo con conexión a internet, sin necesidad de instalaciones adicionales.
- La metodología ágil SCRUM es ideal para este tipo de proyectos educativos, ya que permite el desarrollo incremental, la constante retroalimentación de usuarios clave (como docentes y administradores) y una mayor capacidad de adaptación a cambios o mejoras durante el proceso de construcción del software.

Evaluación del Aprendizaje

¿Cuáles son las principales diferencias entre requisitos funcionales y no funcionales, y por qué es importante identificar ambos tipos en un proyecto de software?

Los **requisitos funcionales** describen lo que el sistema debe hacer. Es decir, especifican las funciones, procesos y comportamientos que debe tener el software para cumplir con los objetivos del usuario, como registrar notas, generar reportes, o autenticar usuarios.



Por otro lado, los **requisitos no funcionales** se enfocan en cómo debe comportarse el sistema, abarcando aspectos de calidad como el rendimiento, la seguridad, la escalabilidad, la usabilidad o la compatibilidad.

¿Qué técnicas de recolección de información son más efectivas para el análisis de requisitos y en qué contextos se recomienda utilizarlas?

- Las **entrevistas** son útiles cuando se requiere obtener información detallada y específica de los usuarios clave.
- Las **encuestas** son ideales cuando se busca recopilar opiniones de un grupo amplio de personas en poco tiempo.
- La **observación directa** se recomienda cuando se desea entender cómo se realiza actualmente una tarea o proceso en el entorno real.
- El **análisis documental** es útil para revisar sistemas previos o normativa institucional, mientras que los **talleres colaborativos** son recomendables cuando se busca construir requisitos de forma conjunta con varios actores involucrados, promoviendo una visión compartida.

¿Cómo pueden los diagramas de casos de uso y otros modelos visuales ayudar en el proceso de análisis de requisitos?

Los **diagramas de casos de uso** y otros modelos visuales como diagramas de flujo, diagramas entidad-relación o modelos UML, son herramientas valiosas durante el análisis de requisitos porque permiten representar de manera clara y estructurada la interacción entre los usuarios y el sistema. Ayudan a identificar funciones clave, actores involucrados y relaciones entre procesos, facilitando la comprensión tanto para desarrolladores como para los usuarios no técnicos.



¿Cómo pueden los cambios en los requisitos afectar un proyecto de software y qué estrategias pueden implementarse para gestionar estos cambios de manera efectiva?

Los **cambios en los requisitos** pueden tener un impacto significativo en un proyecto de software si no se gestionan adecuadamente. Estos cambios pueden provocar **retrasos en los plazos, aumento de costos, retrabajo** en funcionalidades ya desarrolladas y **confusión en el equipo de trabajo**, especialmente si los cambios no están documentados ni comunicados de forma clara. Además, pueden afectar la calidad del producto final si se introducen de manera apresurada o sin una evaluación previa.

Criterio	Excelente (5)	Bueno (4)	Satisfactorio (3)	Necesita Mejora (2)	Insuficiente (1)
Análisis de Requisitos	Análisis completo y detallado, abarca todas las necesidades del usuario y stakeholders.	Análisis correcto con pequeñas omisiones en las necesidades del usuario.	Análisis funcional, pero con varios aspectos menores omitidos.	Análisis incompleto con errores significativos en las necesidades del usuario.	Análisis incorrecto o muy incompleto, no abarca las necesidades del usuario.
Diseño del Software	Diseño correcto y optimizado, cumple con todos los requisitos especificados.	Diseño correcto con pequeños errores, cumple con la mayoría de los requisitos.	Diseño funcional con varios errores menores, cumple con los requisitos básicos.	Diseño con errores significativos, cumple parcialmente con los requisitos.	Diseño incorrecto o incompleto, no cumple con los requisitos.
Modelado UML	Modelos UML completos y precisos, reflejan claramente la arquitectura del sistema.	Modelos UML correctos con pequeñas imprecisiones.	Modelos UML funcionales con varios errores menores.	Modelos UML con errores significativos, reflejan parcialmente la arquitectura del sistema.	Modelos UML incorrectos o muy incompletos.
Documentación Técnica	Documentación completa y detallada, fácil de entender y seguir.	Documentación completa y detallada, fácil de entender y seguir.	Documentación funcional, pero con varios errores menores.	Documentación incompleta con errores significativos.	Documentación incorrecta o muy incompleta.
Verificación y Validación	Verificación y validación completas y	Verificación y validación correctas con	Verificación y validación funcionales	Verificación y validación con errores	Verificación y validación incorrectas o



	precisas, cumplen con todos los requisitos y estándares.	pequeños problemas.	con varios problemas menores.	significativos, cumplen parcialmente con los requisitos y estándares.	muy incompletas.
--	--	---------------------	-------------------------------	---	------------------

Bibliografía

Alarcón, R. (2000). *Diseño orientado a objetos con UML*. Madrid: Grupo EIDOS.

Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (2005). *El lenguaje unificado de modelado* (Vol. 9). Madrid: Addison wesley.

Debrauwer, L., Ferrandez, S., Evain, Y. (2021). *Design patterns en php. Los 23 patrones de diseño: descripciones y soluciones ilustradas en uml 2 y php* (2th ed.). ENI Ediciones.

Anexos

Anexo 1.

Descripción general del sistema

Análisis General

Esta sección proporciona una visión detallada del sistema, precisando sus funciones, la información que maneja y todos los elementos implicados en su progreso:

Perspectivas del Producto

- **Objetivo de la Aplicación:** La aplicación está diseñada para ofrecer información general sobre la escuela y detalles específicos para cada tipo de usuario registrado.
- **Interfaz del Sistema:** Como es una aplicación web, puede ser utilizada en cualquier sistema operativo.



- **Interfaz de Usuario:** La interfaz es accesible para todos los usuarios; sin embargo, se necesita autenticarse como docente o administrador, el ingreso será mediante una intranet.
- **Interfaz de Hardware:** El hardware que posee la institución es suficiente para correr el sistema.
- **Interfaz de Software:** El sistema se ejecutará en cualquier dispositivo que tenga un navegador web moderno con acceso a internet
- **Interfaz de Comunicación:** Las comunicaciones se llevarán a cabo utilizando el protocolo HTTP sobre conexiones TCP/IP.
- **Restricción de Memoria:** La aplicación necesita un mínimo y espacio de disco

Funciones del Producto

El sistema está diseñado para asegurar un control seguro de los procesos académicos,

A continuación, se describen las funciones del sistema según los roles de los usuarios:

Usuario Docente

- **Alumnos Reporte:** Seleccionar el año lectivo y el año de básica para mostrar la lista de alumnos matriculados en el año seleccionado.
- **Registro de notas:** Seleccionar el trimestre y la asignatura para mostrar la lista de alumnos matriculados y registrar sus notas.
- **Registrar Faltas:** Seleccionar el año lectivo y registrar las faltas, mostrando las faltas justificadas e injustificadas de cada estudiante.
- **Modificar las notas:** Seleccionar el periodo y la asignatura para mostrar la lista de estudiantes matriculados y modificar sus notas.



- Modificar Faltas: Seleccionar el periodo para registrar las faltas, mostrando las faltas justificadas e injustificadas y permitiendo mantener o eliminar las faltas justificadas.
- Listar Notas Reporte parcial: Seleccionar el periodo, y mostrar un reporte con las asignaturas y las notas correspondientes.
- Listar Notas: Mostrar un reporte con las asignaturas y las notas correspondientes.
- Reporte final: Mostrar un reporte con las asignaturas y las notas finales.
- Mostrar estudiantes Matriculados: Seleccionar el tipo de matrícula para mostrar un reporte de los alumnos matriculados.
- Mostrar Alumnos Aprobados: Seleccionar el periodo, para mostrar un reporte de los alumnos aprobados.
- Mostrar Alumnos Reprobados: Mostrar un reporte de los alumnos reprobados.
- Mostrar asistencia final: Mostrar un reporte de asistencia de los alumnos.

Usuario Administrador

El usuario administrador tiene acceso total a todas las funcionalidades del sistema:

- Registrar nuevos usuarios.
- Registrar nuevas materias.
- Registrar nuevos años de básica.
- Crear periodo lectivo: Registrar nuevos años lectivos.
- Registrar nuevos docentes y asignarlos a un año de básica.
- Eliminar usuarios.
- Eliminar asignaturas.
- Eliminar años de básica.



- Eliminar años lectivos.
- Eliminar docentes.
- Actualizar la información de los usuarios.
- Actualizar la información de los años de básica.
- Actualizar la información de las asignaturas.
- Actualizar la información de los años lectivos.
- Actualizar la información de los docentes.
- Matricular alumnos en un año de básica después de ingresar los datos necesarios.
- Realizar cambios de paralelo para los alumnos.
- Anular los procesos realizados por los alumnos.
- Generar el proceso de aprobación de los alumnos.

Características del Usuario

Existen dos tipos de usuarios: no registrados y registrados.

- **Usuarios No Registrados:** Acceden únicamente a la información general de la institución publicada en el sitio web.
- **Usuarios Registrados:** Tienen acceso a la intranet, diferenciándose en dos subtipos:
 - **Usuario Docente:**
 - **Usuario Administrador:**

Anexo 2

Requerimientos del sistema



Requerimientos Funcionales

Los requisitos funcionales describen las funciones y características específicas que debe tener un sistema para cumplir con sus objetivos. Estos requisitos detallan qué debe hacer el sistema, cómo debe comportarse y las interacciones entre los usuarios y el sistema. Incluyen especificaciones sobre procesos, datos, interfaces y reportes necesarios para garantizar el correcto funcionamiento del software según las necesidades del usuario.

Requisitos no Funcionales

Los requisitos no funcionales definen los criterios de calidad y las restricciones del sistema que no están directamente relacionados con comportamientos específicos. Estos incluyen aspectos como la seguridad, el rendimiento, la usabilidad, la fiabilidad y la escalabilidad. Los requisitos no funcionales aseguran que el sistema sea eficiente, seguro y fácil de mantener, proporcionando una experiencia de usuario satisfactoria y cumpliendo con las expectativas de calidad.



Tecnológico Universitario *Vida Nueva*

