Versión 1.1

Fecha: 17/12/2024

Logotipo, nombre de la empresa

Descripción generada automáticamente

Manual Técnico

Equipo 205 - SICUE UCO

**Alejandro Gómez Amaro**

**Jorge Santos Jiménez**

**Raúl Castillo Díaz**

Índice

[1. Introducción pág.3](#_Toc184724334)

[2. Planificación y metodología de la aplicación: pág.3](#_Toc184724335)

[Sprint 1. (7 al 14 de Octubre) pág.3](#_Toc184724336)

[Sprint 2. (14 al 21 de Octubre) pág.4](#_Toc184724338)

[Sprint 3. (21 al 28 de Octubre) pág.4](#_Toc184724340)

[Sprint 4. (28 de Octubre al 4 de Noviembre) pág.4](#_Toc184724342)

[Sprint 5. (4 al 18 de Noviembre) pág.4](#_Toc184724344)

[Sprint 6. (25 de Noviembre al 2 de Diciembre) pág.5](#_Toc184724346)

[Sprint 7. (2 al 16 de Diciembre) pág.5](#_Toc184724348)

[Sprint 8. (16 al 20 de Diciembre) pág.5](#_Toc184724350)

[3. Especificación de Requisitos pág.6](#_Toc184724352)

[3.1. Requisitos Iniciales pág.6](#_Toc184724353)

[3.2. Estandarización IEEE 830 y métodos débiles de análisis pág.7](#_Toc184724354)

[3.2.1. Anotación de los requisitos pág.7](#_Toc184724355)

[3.2.2. Checklist de análisis pág.8](#_Toc184724356)

[3.2.3. Lista de errores pág.9](#_Toc184724357)

[3.2.4. Matriz de Interacción: pág.10](#_Toc184724358)

[3.3. Requisitos ajustados: pág.10](#_Toc184724359)

[3.3.1. Requisitos funcionales: pág.10](#_Toc184724360)

[3.3.2. Requisitos de información: pág.11](#_Toc184724361)

[3.3.3. Requisitos no funcionales: pág.12](#_Toc184724362)

[4. Análisis de Requisitos pág.13](#_Toc184724366)

[4.1. Historias de Usuario: pág.13](#_Toc184724367)

[4.2. Casos de Uso pág.16](#_Toc184724368)

[4.2.1. Lenguaje de Modelado Unificado (UML) pág.16](#_Toc184724369)

[4.2.2. Diagrama de Casos de Uso pág.20](#_Toc184724370)

[5. Diseño del Sistema pág.20](#_Toc184724372)

[5.1. Diagrama de Clases UML pág.20](#_Toc184724373)

[5.2. Especificación de Clases pág.21](#_Toc184724374)

[5.3. Diagramas de Secuencias pág.23](#_Toc184724375)

[5.3.1. Registro de la solicitud de participación pág.23](#_Toc184724376)

[5.3.2. Anulación de la solicitud pág.23](#_Toc184724377)

[5.3.3. Consulta del estado de la solicitud pág.24](#_Toc184724378)

[6. Validación del Sistema pág.24](#_Toc184724379)

[6.1. Matriz de Trazabilidad de Requisitos vs Casos de Uso pág.24](#_Toc184724380)

[6.2. Matriz de Trazabilidad de Casos de Uso vs Clases pág.24](#_Toc184724381)

[7. Implementación del Sistema pág.25](#_Toc184724382)

[7.1. Prioridad de creación de Historias de Usuario pág.25](#_Toc184724383)

[7.2. Decisiones y Estrategias de Implementación pág.25](#_Toc184724384)

[8. Pruebas Unitarias pág.26](#_Toc184724385)

[8.1 Diseño de Pruebas pág.26](#_Toc184724386)

[8.2 Informe de Errores pág.30](#_Toc184724387)

[9. Referencias pág.31](#_Toc184724388)

# Introducción

Esta aplicación está destinada a la gestión de la información de los alumnos que desean solicitar la movilidad SICUE en la Universidad de Córdoba. Actualmente, dicho proceso se administra enteramente de manera manual, por lo que hacer las búsquedas tiene un coste elevado que deriva en complicaciones logísticas. Sin embargo, el desarrollo de esta aplicación software supone la herramienta precisa para acabar con todos estos obstáculos y ofrecer a los alumnos y gestores una vía de entendimiento clara, eficaz y sencilla.

El objetivo principal de este mecanismo es obtener copias de respaldo, mecanizar los procesos informáticos y clarificar los posibles planes de convalidación tanto para alumnos como para el profesorado. Todas estas ideas tienen como propósito tener toda la información centralizada y accesible para la Organización de Relaciones Internacionales (ORI), profesores y alumnos interesados por dicho plan de movilidad.

La funcionalidad de la aplicación SICUE UCO se basa en la administración de datos de los estudiantes, profesores y gestión de los planes de convalidación de cada titulación en función de la universidad. La estancia del alumno puede ser de un cuatrimestre o un curso completo, según los planes de convalidación acordados con las universidades que tengan este convenio. Dichos planes deben de ser precisos debido a la relación de créditos y asignación de asignaturas, cada plan de convalidación es específico a una titulación concreta.

Por otro lado, cada vez que se quiera acceder a la aplicación se necesitan las credenciales otorgadas por la Universidad de Córdoba (UCO) a cada alumno.

En cuanto al profesorado también goza de esta movilidad, para poder continuar con su docencia y poder adquirir otras competencias y ganar experiencia en otros ambientes.

# Planificación y metodología de la aplicación:

El desarrollo de la aplicación se basado en una metodología Scrum, es decir, dividimos el tiempo en Sprints que podemos resumir de forma que encontramos:

### Sprint 1. (7 al 14 de Octubre)

### Donde el dueño del producto fue Alejandro Gómez Amaro.

En este sprint inicial, el equipo se enfocó en el análisis y clasificación de requisitos del sistema:

* Clasificar los requisitos en funcionales, de información y no funcionales.
* Verificar que los requisitos cumplen el estándar IEEE 830 y los métodos de análisis débiles.
* Clasificar los requisitos según su origen (inferidos o solicitados por el cliente). El objetivo principal fue establecer una base sólida para los requisitos del sistema, asegurando que fueran claros y viables.

### Sprint 2. (14 al 21 de Octubre)

### Donde el dueño del producto fue Jorge Santos Jiménez.

Durante este sprint, se trabajó en:

* Crear historias de usuario que reflejaran las necesidades y expectativas del cliente.
* Añadir estructura a cada historia de usuario.
* Verificar que todas las historias cumplan los requisitos establecidos en el Sprint 1. El propósito fue convertir los requisitos en historias prácticas y comprensibles, útiles para el desarrollo.

### Sprint 3. (21 al 28 de Octubre)

### Donde el dueño del producto fue Raúl Castillo Díaz.

El equipo pasó a los casos de uso, abordando tareas más técnicas:

* Crear un diagrama de casos de uso en Visual Paradigm.
* Adaptar los casos de uso al formato UML.
* Verificar que los casos de uso cumplan con los requisitos especificados anteriormente. Esta fase facilitó la transición hacia un diseño técnico más detallado del sistema.

### Sprint 4. (28 de Octubre al 4 de Noviembre)

### Donde el dueño del producto fue Alejandro Gómez Amaro.

En este sprint, se enfocaron en el diseño UML:

* Crear el diagrama de clases UML para representar las entidades y sus relaciones.
* Refinar y especificar los requisitos para garantizar su precisión.
* Verificar que todo estuviera correctamente documentado y alineado con las fases previas. El objetivo fue desarrollar un modelo conceptual del sistema que guiara la implementación.

### Sprint 5. (4 al 18 de Noviembre)

### Donde el dueño del producto fue Jorge Santos Jiménez.

El equipo trabajó en el diseño de interacciones:

* Crear una matriz de trazabilidad entre requisitos, casos de uso y diagramas de clases.
* Desarrollar diagramas de secuencias que incluyeran acciones como crear, anular y consultar solicitudes.
* Verificar que la entrega cumpla con los estándares de calidad establecidos. Este sprint cerró el ciclo con un enfoque en las interacciones detalladas del sistema.

### Sprint 6. (25 de Noviembre al 2 de Diciembre)

### Donde el dueño del producto fue Raúl Castillo Díaz.

Durante este sprint, el equipo comenzó a trabajar en la implementación y documentación técnica:

* Crear un repositorio en GitHub para centralizar y gestionar el código fuente.
* Utilizar C++ como lenguaje de programación y Eclipse como IDE.
* Iniciar la implementación del sistema en el Manual Técnico.
* Comenzar a crear el Manual de Usuario.
* Desarrollar la función de "Gestionar Plan de Convalidación". Este sprint marcó el inicio de la implementación técnica y la creación de los manuales necesarios para el sistema.

### Sprint 7. (2 al 16 de Diciembre)

### Donde el dueño del producto fue Alejandro Gómez Amaro.

* Crear las funciones "Crear Solicitud" y "Consultar Solicitud".
* Finalizar el Manual de Usuario.
* Crear las Pruebas Unitarias y documentarlas en el Manual Técnico. Este sprint permitió avanzar en el desarrollo y asegurar la calidad del sistema a través de pruebas unitarias.

### Sprint 8. (16 al 20 de Diciembre)

### Donde el dueño del producto fue Jorge Santos Jiménez.

En este sprint, se llevaron a cabo actividades de cierre para garantizar la calidad y finalización de los entregables:

* Depurar el código para asegurar que las funciones implementadas cumplen correctamente con su propósito.
* Verificar y finalizar el Manual Técnico.
* Verificar y finalizar el Manual de Usuario. El objetivo fue completar la documentación y depuración para entregar un sistema funcional y bien documentado.
* Grabación de la explicación de la utilidad de la aplicación finalizada.

Además de todo esto en la nuestra página de [Trello](https://trello.com/invite/b/6703eecfedf497f2dd39cd71/ATTIa6dc4503c98879f86cfdca46c46bdbb722EF4277/practica-1) podemos encontrar cuatro tableros, las Practicas 1, 2 y 4 donde encontramos los sprints divididos por entregas además de los apuntes obtenidos en las reuniones y las Historias de Usuario y un cuarto tablero Documentación donde están todos los links del logo de la aplicación, los archivos del grupo y otros documentos que nos han resultado útiles en la elaboración de este trabajo.

# Especificación de Requisitos

## Requisitos Iniciales

Requisitos funcionales:

1. Solicitud de inscripción al programa SICUE de alumnos y profesores.
2. Consulta de inscripciones **de alumnos y profesores**.
3. Anulación de inscripciones **de alumnos y profesores**.
4. **Filtrado y clasificación por**: Universidad, Centro, Titulación.
5. Actualizaciones de la solicitud vía email.
6. Gestión de planes de convalidación (cuatrimestre o curso completo) asignaturas y prácticas.

Requisitos de información:

1. **Datos almacenados** de los alumnos y profesores.
2. Datos almacenados de todas las Universidades y posibles convalidaciones de asignaturas y prácticas.
3. Historial de solicitudes y anulaciones de solicitudes.

Requisitos no funcionales:

De producto:

1. Usabilidad: El sistema debe ser intuitivo para los usuarios.
2. **Rendimiento**: El sistema debe soportar múltiples usuarios simultáneos sin afectar su rendimiento.

De organización:

1. **Clasificación**: Los datos deben ser almacenados y clasificados en la ORI de manera eficiente.

Externos:

1. **Accesibilidad**: La aplicación debe ejecutarse dentro de la web de la UCO.
2. Seguridad: Solo los usuarios autenticados pueden acceder a la aplicación y se debe garantizar la confidencialidad de los datos.
3. Mantenimiento: La aplicación debe ser fácilmente modificable y mantenible para actualizaciones futuras.

## Estandarización IEEE 830 y métodos débiles de análisis

### Anotación de los requisitos

* + RF1: Solicitud de inscripción
  + **RF2:** Consulta de inscripciones
  + **RF3:** Anulación de inscripciones
  + RF4: Filtrado y clasificación
  + **RF5:** Actualizaciones de la solicitud
  + **RF6:** Gestión de planes de convalidación
  + **RI1: Datos** de los alumnos y profesores
  + **RI2: Datos** de universidades
  + RI3: Historial de solicitudes
  + RNF1: Usabilidad
  + RNF2: Rendimiento
  + RNF3: Clasificación y almacenamiento en la ORI
  + RNF4: Accesibilidad en la web de la UCO
  + RNF5: Seguridad
  + RNF6: Mantenimiento (Se elimina en los requisitos finales)

### Checklist de análisis

|  |  |
| --- | --- |
| Atributo de calidad a considerar | Pregunta |
| Corrección | - ¿El requisito refleja fielmente las necesidades?  - ¿Cada requisito está correctamente especificado para cumplir con los objetivos del sistema? |
| No ambigüedad | - ¿Cada requisito tiene una única interpretación clara?   |  | | --- | | - ¿Los términos utilizados son claros y comprensibles para todos los involucrados? | |
| Completitud | - ¿Están cubiertas todas las funciones y necesidades?  - ¿Faltan detalles en algún requisito que puedan ser necesarios para la implementación? |
| Consistencia | - ¿Los requisitos son consistentes entre sí, sin contradicciones?  - ¿La relación entre requisitos es coherente y no da lugar a ambigüedades? |
| Concisión | - ¿El requisito es lo suficientemente breve y directo para ser claro?  - ¿Hay detalles innecesarios o redundantes en la descripción del requisito? |
| Realizabilidad | - ¿Es técnicamente posible implementar el requisito con la tecnología actual?   |  | | --- | | - ¿Se han considerado las limitaciones técnicas que pueden afectar la implementación? | |
| Verificabilidad | - ¿Es posible probar el requisito para confirmar que se cumple?  - ¿Puede realizarse un caso de prueba específico para validar el requisito? |
| Modificabilidad | - ¿El requisito está estructurado para que sea fácil realizar cambios si es necesario?  - ¿Se encuentran definidas las condiciones bajo las cuales el requisito podría ser modificado? |
| Trazabilidad | |  | | --- | | - ¿Es posible rastrear el origen del requisito a un objetivo del sistema o una solicitud del cliente? |   - ¿Es posible vincular el requisito a su implementación específica dentro del sistema? |
| Independencia del diseño | - ¿El requisito evita hacer referencia a detalles de diseño específicos? |
| Clasificación | - ¿El requisito se encuentra clasificado adecuadamente como de producto, organizacional o externo? |
| Estabilidad | - ¿El requisito es estable o podría cambiar en el futuro debido a factores externos o de negocio? |
| Relevancia | - ¿El requisito es relevante y necesario para el funcionamiento del sistema? |

### Lista de errores

|  |  |  |
| --- | --- | --- |
| No de requisito | Defectos detectados | Acciones recomendadas |
| RF2 | No se detalla cómo los usuarios podrán acceder a su historial completo de inscripciones. | Incluir una funcionalidad que permita a los usuarios ver su historial completo de inscripciones previas. |
| RF3 | No se especifican los procedimientos para confirmar o verificar la anulación de una inscripción. | Establecer un proceso de confirmación para la anulación, como una notificación al usuario por correo electrónico. |
| RF5 | No se especifica el tipo de actualizaciones (confirmación, anulación, cambios) ni la frecuencia de los correos. | Aclarar los tipos de notificación por correo (por ejemplo, confirmación de inscripción, cambios en la solicitud). |
| RF6 | No se establecen criterios de elegibilidad o proceso de aprobación para asignaturas y prácticas. | Especificar los criterios de elegibilidad para convalidación y el proceso para la aprobación de planes de convalidación. |
| RI3 | No se aclara cuánto tiempo se almacenará el historial ni si los usuarios pueden acceder a él. | Definir el tiempo de retención de los datos históricos y aclarar si los usuarios tienen acceso a su historial completo. |
| RNF2 | No se define un número máximo de usuarios concurrentes para realizar pruebas de carga. | Establecer un límite de usuarios simultáneos y realizar pruebas de carga para confirmar el rendimiento del sistema. |
| RNF4 | No se especifica compatibilidad con navegadores o dispositivos móviles. | Aclarar los navegadores compatibles y considerar el soporte para dispositivos móviles para mejorar la accesibilidad. |
| RNF5 | No se especifican controles de seguridad ni protocolos específicos de control de acceso. | Incluir controles de seguridad regulares y definir protocolos específicos para el control de acceso y la autenticación. |
| RNF6 | No se especifica un plan de mantenimiento o la documentación necesaria para futuras actualizaciones. | Crear un plan de mantenimiento que incluya procedimientos de actualización y documentación técnica para los desarrolladores. |

### Matriz de Interacción:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF1 | RF2 | RF3 | RF4 | RF5 | RF6 | RI1 | RI2 | RI3 | RNF1 | RNF2 | RNF3 | RNF4 | RNF5 |
| RF1 |  | S | S |  | S | S | S |  |  | S |  | S |  | S |
| RF2 |  |  | S |  | S |  |  |  | S | S |  |  | S |  |
| RF3 |  |  |  |  | S |  |  |  | S | S |  |  | S |  |
| RF4 |  |  |  |  |  |  |  |  |  |  |  | S |  |  |
| RF5 |  |  |  |  |  | S |  |  |  |  |  |  | S |  |
| RF6 |  |  |  |  |  |  |  |  |  |  |  |  |  | S |
| RI1 |  |  |  |  |  |  |  | S |  |  |  | S |  | S |
| RI2 |  |  |  |  |  |  |  |  |  |  |  | S |  | S |
| RI3 |  |  |  |  |  |  |  |  |  | S |  |  |  | S |
| RNF1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RNF2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RNF3 |  |  |  |  |  |  |  |  |  |  |  |  |  | S |
| RNF4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RNF5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Donde S significa solapamiento y C, conflicto (no contiene)

## Requisitos ajustados:

**Tras realizar la e**standarización IEEE 830 y aplicar métodos débiles de análisis concluimos con los siguientes requisitos:

### Requisitos funcionales:

1. Solicitud de inscripción al programa SICUE de alumnos y profesores

* La aplicación deberá permitir a los alumnos y profesores de la UCO solicitar su inscripción en el programa SICUE, completando campos obligatorios como nombre, DNI, titulación, y nivel académico.
* Se deberán implementar validaciones para asegurar que todos los datos requeridos sean correctos y completos antes de enviar la solicitud.
* Los datos proporcionados se validarán en tiempo real y se mostrará un mensaje de confirmación al usuario una vez finalizada la solicitud.

1. Consulta de inscripciones de alumnos y profesores

* La aplicación deberá permitir a los usuarios consultar de sus inscripciones en el programa SICUE, incluyendo estado actual, fecha de solicitud y detalles específicos del plan seleccionado.

1. Anulación de inscripciones de alumnos y profesores

* La aplicación deberá permitir a los usuarios anular su inscripción en el programa SICUE, confirmando la anulación mediante una notificación de confirmación.

1. Filtrado y clasificación

* Los datos de los estudiantes y profesores estarán clasificados en la aplicación por Universidad, Centro, y Titulación, facilitando la consulta y la gestión de información.
* Los administradores y miembros del ORI podrán aplicar filtros de clasificación para acceder a la información relevante de manera organizada.

1. Actualizaciones de la solicitud vía email

* La aplicación enviará notificaciones automáticas por correo electrónico si hay alguna actualización en la inscripción (Solicitud, anulación, aprobación o denegado de la solicitud).
* El sistema permitirá a los usuarios elegir qué correo desea recibir las notificaciones.

1. Gestión de planes de convalidación (cuatrimestre o curso completo)

* La aplicación permitirá a los estudiantes seleccionar entre un plan de convalidación de cuatrimestre o curso completo, gestionando asignaturas y prácticas según convenios con universidades asociadas.
* Se implementarán criterios específicos de elegibilidad y un proceso de aprobación para la convalidación de asignaturas y prácticas, en colaboración con la ORI.

### Requisitos de información:

1. Datos almacenados de los alumnos y profesores

* La aplicación almacenará los datos personales y académicos de alumnos y profesores, incluyendo nombre, apellidos, DNI, titulación, y nivel de idiomas.
* Los datos estarán protegidos mediante cifrado y control de acceso, cumpliendo con las políticas de protección de datos, todo ello gestionado por ORI y UCO externamente.

1. Datos almacenados de universidades y convalidaciones

* La aplicación almacenará los datos de todas las universidades asociadas y sus posibles convalidaciones de asignaturas y prácticas.
* Se establecerá un proceso para la actualización periódica de estos datos para asegurar que estén siempre actualizados.

1. Historial de solicitudes y anulaciones

* La aplicación guardará un historial completo de todas las solicitudes y anulaciones realizadas por los usuarios.
* Los datos históricos se conservarán durante un período de tiempo especificado y serán accesibles a los usuarios para su consulta.

### Requisitos no funcionales:

### De producto:

1. Usabilidad:

* La aplicación deberá ser intuitiva y fácil de usar, asegurando que los estudiantes y profesores puedan completar sus inscripciones y consultas de manera eficiente.
* Se definirán métricas de usabilidad (por ejemplo, tiempo promedio para completar la inscripción) y se realizarán pruebas con usuarios reales para verificar que la interfaz es accesible y comprensible para todos.

2. Rendimiento:

* La aplicación deberá ser capaz de gestionar múltiples inscripciones y consultas de forma simultánea sin afectar su rendimiento.
* Se establecerá un número máximo de usuarios simultáneos, y se realizarán pruebas de carga para verificar que el sistema cumple con los requisitos de rendimiento.

### De organización:

3. Clasificación y almacenamiento en la ORI:

* Los datos gestionados por la aplicación deberán almacenarse en la base de datos de la Oficina de Relaciones Internacionales (ORI) de la UCO, permitiendo su clasificación y filtrado por Universidad, Centro y Titulación.
* Se implementará un mecanismo de respaldo y se especificará la frecuencia de actualización de los datos en la base de la ORI.

### Externos:

1. Accesibilidad en la web de la UCO:

* La aplicación deberá ejecutarse de manera óptima dentro de la plataforma web de la UCO, siendo accesible desde cualquier navegador web compatible con los sistemas de la universidad.
* Se especificarán los navegadores y versiones compatibles y se definirán requisitos mínimos de sistema para el acceso óptimo.

1. Seguridad:

* El acceso a la aplicación estará limitado a usuarios autenticados mediante el sistema de autenticación de la UCO, garantizando la protección de los datos personales y académicos mediante cifrado y políticas de seguridad establecidas por la universidad.
* Se establecerán controles de seguridad regulares y se detallarán los mecanismos de autenticación y control de acceso específico.

# Análisis de Requisitos

## 4.1. Historias de Usuario:

|  |  |
| --- | --- |
| H1 | Como alumno o profesor de la UCO, quiero poder solicitar mi inscripción en el programa, para poder participar en el intercambio académico. |
| Descripción: Los usuarios de la UCO (alumnos y profesores) podrán inscribirse en el programa SICUE, completando los campos obligatorios y recibiendo una confirmación.  Responsable: Alejandro Gómez Amaro.  Criterios de validación:   * El usuario debe completar todos los campos obligatorios (nombre, DNI, titulación, nivel académico) para enviar la solicitud. * Los datos se validarán en tiempo real, y si faltan datos o son incorrectos, se mostrará un mensaje de error. * El sistema deberá mostrar un mensaje de confirmación al usuario al completar la solicitud.   Prioridad: Alta.  Estimación en días: 5  Sprint: 1 | |

|  |  |
| --- | --- |
| H2 | Como alumno o profesor de la UCO, quiero consultar el estado de mi inscripción en el programa SICUE, para verificar mi participación y el plan seleccionado. |
| Descripción: Los usuarios podrán ver el estado de su inscripción en el programa SICUE, incluyendo la fecha de solicitud y detalles específicos del plan.  Responsable: Jorge Santos Jiménez.  Criterios de validación:   * El usuario debe estar autenticado para poder consultar sus inscripciones. * La consulta debe mostrar el estado actual de la inscripción, la fecha de solicitud y detalles del plan seleccionado. * Si no existen inscripciones, el sistema deberá informar que no hay inscripciones activas.   Prioridad: Media.  Estimación en días: 3  Sprint: 1 | |

|  |  |
| --- | --- |
| H3 | Como usuario del programa, quiero poder anular mi inscripción, para retirar mi participación del programa cuando sea necesario. |
| Descripción: La aplicación permitirá a los usuarios anular su inscripción en el programa SICUE y recibir una confirmación de la anulación.  Responsable: Raúl Castillo Díaz.  Criterios de validación:   * El sistema debe enviar una notificación de confirmación de la anulación por correo electrónico y en la interfaz. * La inscripción debe marcarse como anulada y quedar registrada en el historial de inscripciones. * Solo las inscripciones activas podrán ser anuladas.   Prioridad: Media.  Estimación en días: 3  Sprint: 1 | |

|  |  |
| --- | --- |
| H4 | Como administrador quiero poder filtrar y clasificar los datos de estudiantes y profesores, para gestionar la información de manera organizada y eficiente. |
| Descripción: Los datos de los usuarios estarán clasificados en la aplicación por Universidad, Centro y Titulación, y los administradores podrán aplicar filtros para gestionar la información.  Responsable: Alejandro Gómez Amaro.  Criterios de validación:   * Los filtros por Universidad, Centro y Titulación deben estar disponibles y funcionar correctamente. * Solo los administradores y miembros de la ORI tendrán acceso a las opciones de filtrado. * La clasificación debe actualizarse en tiempo real para reflejar cambios recientes.   Prioridad: Media.  Estimación en días: 4  Sprint: 1 | |

|  |  |
| --- | --- |
| H5 | Como usuario de la UCO, quiero recibir actualizaciones de mi solicitud por email, para estar informado sobre el estado de mi inscripción. |
| Descripción: La aplicación enviará notificaciones automáticas por correo electrónico si hay alguna actualización en la inscripción.  Responsable: Jorge Santos Jiménez.  Criterios de validación:   * El usuario debe poder elegir a que correo desea recibir las notificaciones. * Las notificaciones deben enviarse para las actualizaciones de solicitud, anulación, aprobación y denegación. * Si el correo del usuario es inválido, el sistema deberá informar del error y solicitar una corrección.   Prioridad: Media.  Estimación en días: 3  Sprint: 1 | |

|  |  |
| --- | --- |
| H6 | Como estudiante participante, quiero seleccionar un plan de convalidación (cuatrimestre o curso completo), para gestionar la convalidación de asignaturas y prácticas según el convenio con la universidad asociada. |
| Descripción: Los estudiantes podrán elegir entre un plan de convalidación de cuatrimestre o curso completo. Se gestionarán las asignaturas y prácticas según el convenio con universidades asociadas.  Responsable: Raúl Castillo Díaz.  Criterios de validación:   * Solo los estudiantes elegibles, podrán seleccionar el plan de convalidación. * La opción seleccionada debe reflejarse en el estado de la inscripción y mostrarse en la consulta de inscripciones. * Si el estudiante no cumple con los requisitos de elegibilidad, el sistema mostrará un mensaje indicando el motivo.   Prioridad: Alta.  Estimación en días: 5  Sprint: 2 | |

|  |  |
| --- | --- |
| H7 | Como administrador de la ORI, quiero acceder a los datos almacenados de los alumnos y profesores, para verificar su información de forma segura. |
| Descripción: La aplicación almacenará los datos personales y académicos de alumnos y profesores incluyendo planes de convalidación con protección mediante cifrado y control de acceso.  Responsable: Alejandro Gómez Amaro.  Criterios de validación:   * Solo usuarios autorizados podrán acceder a los datos almacenados de alumnos y profesores. * Los datos estarán cifrados y protegidos según las políticas de la UCO. * Los accesos deberán registrarse para auditorías de seguridad.   Prioridad: Alta.  Estimación en días: 4  Sprint: 2 | |

|  |  |
| --- | --- |
| H8 | Como administrador, quiero acceder al historial de solicitudes y anulaciones, para revisar la información de participaciones y anulaciones en el programa SICUE. |
| Descripción: La aplicación guardará un historial de todas las solicitudes y anulaciones realizadas por los usuarios, accesible para los administradores y los usuarios con acceso autorizado.  Responsable: Jorge Santos Jiménez.  Criterios de validación:   * El historial de solicitudes y anulaciones debe estar disponible solo para usuarios autorizados. * El historial deberá conservarse durante el tiempo especificado en la política de retención de datos. * Cada solicitud y anulación debe incluir detalles como fecha, estado y tipo de solicitud.   Prioridad: Media  Estimación en días: 3  Sprint: 2 | |

## 

## 4.2. Casos de Uso

### 4.2.1. Lenguaje de Modelado Unificado (UML)

|  |  |
| --- | --- |
| CU1 | Registro de solicitud de participación en SICUE |
| Descripción: Los alumnos y profesores de la UCO solicitan su inscripción en el programa SICUE.  Objetivo: Permitir a los usuarios registrarse en el programa SICUE.  Condiciones:   * El usuario debe ser un alumno o profesor de la UCO. * El usuario debe completar todos los campos obligatorios (nombre, DNI, titulación, nivel académico).   Actores principales: Alumnos y profesores.  Escenario principal:   * 1. El usuario accede a la opción de inscripción en la aplicación.   2. El sistema muestra el formulario de inscripción.   3. El usuario completa los campos obligatorios.   4. El sistema valida los datos en tiempo real.   5. El usuario envía la solicitud.   6. El sistema confirma el registro exitoso y envía un mensaje de confirmación.   Escenarios alternativos:  1a. Si el usuario no tiene permisos para inscribirse, el sistema deniega el acceso y  muestra un mensaje de error.  3a. Si faltan campos obligatorios, el sistema muestra un mensaje de error y solicita al  usuario completar los datos. | |

|  |  |
| --- | --- |
| CU2 | Consulta de estado de solicitud SICUE |
| Descripción: Los usuarios consultan el estado de su solicitud en el programa SICUE, incluyendo detalles del plan seleccionado.  Objetivo: Permitir a los usuarios consultar el estado y detalles de su solicitud en el programa SICUE.  Condiciones:   * El usuario debe estar autenticado. * Debe existir una solicitud activa asociada al usuario.   Actores principales: Alumnos y profesores.  Escenario principal:   1. El usuario selecciona la opción de consulta de estado en la aplicación. 2. El sistema muestra el historial de solicitudes del usuario. 3. El usuario selecciona la solicitud que desea consultar. 4. El sistema muestra el estado, fecha y detalles del plan de la solicitud seleccionada.   Escenarios alternativos:  2a. Si el usuario no tiene una solicitud activa, el sistema muestra un mensaje  informando que no hay solicitudes activas. | |

|  |  |
| --- | --- |
| CU3 | Anulación de solicitud |
| Descripción: Los usuarios anulan su inscripción en el programa SICUE y reciben una confirmación de anulación.  Objetivo: Permitir a los usuarios anular su participación en el programa SICUE.  Condiciones:   * El usuario debe tener una solicitud activa. * La solicitud debe estar en estado pendiente o en proceso.   Actores principales: Alumnos y profesor.  Escenario principal:   * 1. El usuario accede a la opción de anulación en su solicitud.   2. El sistema muestra los detalles de la solicitud y pide confirmación de anulación.   3. El usuario confirma la anulación.   4. El sistema marca la solicitud como anulada y envía una notificación al usuario.   Escenarios alternativos:  2a. Si la solicitud no puede ser anulada (ya completada), el sistema muestra un  mensaje indicando que no es posible anular la solicitud. | |

|  |  |
| --- | --- |
| CU4 | Gestión de convalidación de asignaturas |
| Descripción: Los estudiantes gestionan sus planes de convalidación de asignaturas y prácticas según los convenios con universidades asociadas.  Objetivo: Permitir a los estudiantes gestionar la convalidación de sus asignaturas y prácticas.  Condiciones:   * El estudiante debe tener una solicitud activa en SICUE. * Deben existir convenios de convalidación con universidades asociadas.   Actor principal: Estudiante.  Escenario principal:   * 1. El estudiante selecciona la opción de gestión de convalidación.   2. El sistema muestra los convenios y asignaturas/prácticas elegibles.   3. El estudiante selecciona las asignaturas/prácticas para convalidación.   4. El sistema valida las elecciones según los convenios.   5. El estudiante confirma la selección y el sistema registra la solicitud de convalidación.   Escenarios alternativos:  4a. Si la asignatura o práctica no cumple los criterios, el sistema muestra un mensaje  de error. | |

|  |  |
| --- | --- |
| CU5 | Gestión de profesorado inscrito |
| Descripción: Los administradores y miembros de la ORI gestionan el profesorado inscrito en el programa SICUE.  Objetivo: Permitir a los administradores y miembros de la ORI ver y gestionar el profesorado inscrito en SICUE.  Condiciones:   * El usuario debe ser administrador o miembro de la ORI. * El sistema debe tener registros de profesorado inscrito.   Actor principal: Administrador.  Escenario principal:   * 1. El administrador accede a la sección de gestión de profesorado.   2. El sistema muestra la lista de profesores inscritos en SICUE.   3. El administrador selecciona un profesor para ver o editar detalles.   4. El sistema permite modificar datos o actualizar el estado del profesor en el programa.   Escenarios alternativos:  2a. Si no existen profesores registrados, el sistema muestra un mensaje indicando  que no hay inscripciones. | |

|  |  |
| --- | --- |
| CU6 | Filtrado de solicitudes por universidad y titulación |
| Descripción: Los administradores aplican filtros de universidad y titulación para gestionar las solicitudes de SICUE de forma organizada.  Objetivo: Facilitar la consulta de solicitudes específicas mediante filtros de universidad y titulación.  Condiciones:   * El usuario debe ser administrador o miembro de la ORI. * Deben existir solicitudes en el sistema.   Actores principales: Administrador.  Escenario principal:   * 1. El administrador accede a la sección de solicitudes.   2. El sistema muestra los filtros de universidad y titulación.   3. El administrador aplica los filtros deseados.   4. El sistema muestra las solicitudes que cumplen los criterios de filtrado.   Escenarios alternativos:  4a. Si no existen solicitudes que cumplan los filtros aplicados, el sistema muestra un  mensaje indicando que no hay resultados. | |

|  |  |
| --- | --- |
| CU7 | Gestión de planes de convalidación |
| Descripción: Los administradores gestionan los planes de convalidación de asignaturas y prácticas de los estudiantes.  Objetivo: Permitir a los administradores gestionar y aprobar convalidaciones de asignaturas y prácticas.  Condiciones:   * El usuario debe ser administrador o miembro de la ORI. * Deben existir solicitudes de convalidación en el sistema.   Actores principales: Administrador.  Escenario principal:   1. El sistema muestra las solicitudes de convalidación de los estudiantes. 2. El administrador revisa y selecciona una solicitud para aprobación o rechazo. 3. El sistema registra la decisión y actualiza el estado de la solicitud.   Escenarios alternativos:  1a. Si no existen solicitudes, el sistema muestra un mensaje indicando que no hay  solicitudes para gestionar. | |

|  |  |
| --- | --- |
| CU8 | Visualización de actualizaciones de solicitudes |
| Descripción: Los administradores visualizan el historial de actualizaciones de solicitudes de los alumnos y profesores en el sistema.  Objetivo: Permitir la revisión y auditoría de las solicitudes que se realizan, aprueban, deniegan y cancelan.  Condiciones:   * El usuario debe ser administrador o miembro de la ORI. * Deben existir registros de movimientos en el sistema.   Actor principal: Administrador.  Escenario principal:   1. El administrador accede a la sección de historial de movimientos. 2. El sistema muestra los filtros para buscar por fecha, usuario o tipo de acción. 3. El administrador aplica los filtros deseados y visualiza el historial correspondiente.   Escenarios alternativos:  2a. Si no existen movimientos que cumplan los filtros, el sistema muestra un mensaje  indicando que no hay resultados. | |

### 4.2.2. Diagrama Descripción generada automáticamenteDiagrama de Casos de Uso

# Lenguaje

# Diseño del Sistema

## Diagrama Descripción generada automáticamente 5.1. Diagrama de Clases UML

## 5.2. Especificación de Clases

|  |  |  |
| --- | --- | --- |
| Clase: Solicitud | | |
| Descripción: Conjunto de solicitudes completes; solicitante, destino, asignaturas… | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -Usuario | Class | Datos del solicitante |
| -Plan de Convalidación | Class | Prácticas y asignaturas solicitadas |
| -fecha | String | Fecha de la solicitud |
| -estado | String | Estado de la solicitud |
| Operaciones | | |
| Nombre | Tipo | Descripción |
| +crearSolicitud() | Class Solicitud | Crea una solicitud |
| +verSolicitud | Void | Muestra una solicitud |
| +eliminarSolicitud() | Class Solicitud | Elimina una solicitud |

|  |  |  |
| --- | --- | --- |
| Clase: Usuarios | | |
| Descripción: Conjunto de los usuarios | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -nombre | String | Nombre del administrador |
| -dni | String | DNI del administrador |
| -email | String | Correo del administrador |

|  |  |  |
| --- | --- | --- |
| Clase: Usuario, tipo: Administrador | | |
| Descripción: Conjunto de los usuarios tipo administradores | | |
| Operaciones | | |
| Nombre | Tipo | Descripción |
| +verHistorial() | Void | Muestra las actualizaciones de solicitudes |
| +aceptarSolicitud() | Void | Acepta una solicitud |
| +denegarSolicitud() | Void | Denega una solicitud |
| +crearPlanConvalidación | Class Plan de Convalidación | Crear Plan de Convalidación |
| +verSolicitudes | Void | Muestra solicitudes |

|  |  |  |
| --- | --- | --- |
| Clase: Usuario, tipo: Profesores | | |
| Descripción: Conjunto de los usuarios tipo profesores | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -departamento | String | Departamento del que forma parte |

|  |  |  |
| --- | --- | --- |
| Clase: Estudiantes | | |
| Descripción: Conjunto de los usuarios tipo estudiantes | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -carrera | String | Carrera que cursa |

|  |  |  |
| --- | --- | --- |
| Clase: Universidad | | |
| Descripción: Conjunto de las universidades | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -id | String | Identificador de la universidad |
| -facultad | String | Facultad de la universidad |
| -direccion | String | Localización de la universidad |

|  |  |  |
| --- | --- | --- |
| Clase: Plan de Convalidacion | | |
| Descripción: Conjunto de asignaturas, prácticas y duración de la movilidad | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -duracion | String | Duración de la movilidad |
| -Universidades | Class | Universidad a donde se va a realizar la movilidad |

|  |  |  |
| --- | --- | --- |
| Clase: Plan de Convalidacion, tipo: Plan Profesores | | |
| Descripción: Tipo de planes de convalidación de los profesores | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -departamento | String | Departamento donde va a trabajar |

|  |  |  |
| --- | --- | --- |
| Clase: Plan de Convalidacion, tipo: Plan Estudiantes | | |
| Descripción: Tipo de planes de convalidación de los estudiantes | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -Asignaturas | Class | Asignaturas a cursar |

|  |  |  |
| --- | --- | --- |
| Clase: Asignaturas | | |
| Descripción: Conjunto de las asignaturas | | |
| Atributos | | |
| Nombre | Tipo | Descripción |
| -nombre | String | Nombre de la asignatura |
| -codigo | Int | Código de la asignatura |
| -creditos | Int | Créditos de la asignatura |

## 5.3. Diagramas de Secuencias

### Imagen que contiene Diagrama Descripción generada automáticamente5.3.1. Registro de la solicitud de participación

### Imagen que contiene Gráfico Descripción generada automáticamente5.3.2. Anulación de la solicitud

### Imagen que contiene Diagrama Descripción generada automáticamente5.3.3. Consulta del estado de la solicitud

# Validación del Sistema

## Matriz de Trazabilidad de Requisitos vs Casos de Uso

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | CU1 | CU2 | CU3 | CU4 | CU5 | CU6 | CU7 | CU8 |
| RF1 |  |  |  |  |  |  |  |  |
| RF2 |  |  |  |  |  |  |  |  |
| RF3 |  |  |  |  |  |  |  |  |
| RF4 |  |  |  |  |  |  |  |  |
| RF5 |  |  |  |  |  |  |  |  |
| RF6 |  |  |  |  |  |  |  |  |
| RI1 |  |  |  |  |  |  |  |  |
| RI2 |  |  |  |  |  |  |  |  |
| RI3 |  |  |  |  |  |  |  |  |

Donde los recuadros verdes son los que relacionan dicho Requisito con Caso de Uso (fila/columna)

## Matriz de Trazabilidad de Casos de Uso vs Clases

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Profesores | Alumnos | Universidad | Administradores | Solicitud | Plan de Convalidación | Asignaturas | Practicas |
| CU1 |  |  |  |  |  |  |  |  |
| CU2 |  |  |  |  |  |  |  |  |
| CU3 |  |  |  |  |  |  |  |  |
| CU4 |  |  |  |  |  |  |  |  |
| CU5 |  |  |  |  |  |  |  |  |
| CU6 |  |  |  |  |  |  |  |  |
| CU7 |  |  |  |  |  |  |  |  |
| CU8 |  |  |  |  |  |  |  |  |

Donde los recuadros verdes son en los que el Caso de Uso involucra la Clase (fila/columna)

# Implementación del Sistema

## 7.1. Prioridad de creación de Historias de Usuario

El uso de la metodología **Scrum ha** permitido una planificación, priorización y entrega incremental de funcionalidades de forma que las funcionalidades base se crearon con prioridad y posteriormente se fueron incorporando el resto como hemos podido ver en el apartado de historias de usuario. Hacemos un pequeño resumen clasificado por orden de prioridad y sprint como han sido organizadas:

En el primer sprint, se desarrollaron las funcionalidades esenciales del sistema:

* **H1:** Solicitar inscripción en el programa. Prioridad alta. Función esencial para los usuarios, ya que permite la participación en el intercambio académico.
* **H3:** Anular inscripción. Prioridad media. Es importante para garantizar información veraz y permitir correcciones en las inscripciones.
* **H2:** Consultar estado de la inscripción. Prioridad media. Aunque no afecta directamente la funcionalidad de la aplicación, es clave para ofrecer transparencia a los usuarios.
* **H5:** Recibir actualizaciones por email. Prioridad media. Fundamental para mantener a los usuarios informados automáticamente sobre el estado de su inscripción.
* **H4:** Filtrar y clasificar datos. Prioridad media. Facilita la gestión de datos por parte de los administradores mediante filtros y clasificaciones en tiempo real.

En el segundo sprint, se trabajaron funcionalidades complementarias con menor urgencia inicial:

* **H6:** Seleccionar un plan de convalidación. Prioridad alta. Clave para los estudiantes, ya que les permite gestionar su plan académico, aunque depende de la base funcional creada previamente.
* **H7:** Acceso seguro a datos de alumnos y profesores. Prioridad alta. Importante para los administradores, pero no esencial para una versión inicial del sistema.
* **H8:** Acceder al historial de solicitudes y anulaciones. Prioridad media. Facilita el trabajo administrativo, aunque es menos relevante en las primeras etapas del desarrollo.

## 7.2. Decisiones y Estrategias de Implementación

* **Se ha usado un diseño modular con C++ para garantizar mantenibilidad, escalabilidad y modelado del sistema preciso.**
* **Se desarrolló una interfaz intuitiva para facilitar el uso a administradores y usuarios.**
* **Los datos son organizados y protegidos por terceros tanto planes de convalidación (ORI) como los datos de los alumnos (UCO) de forma que solo hay que almacenar las inscripciones reduciendo el costo y simplificando el almacenamiento.**
* **El uso de Eclipse IDE facilita el desarrollo modular y pruebas unitarias del sistema.**
* **Usar los repositorios en GitHub permite la colaboración eficiente y seguimiento de cambios del proyecto por parte de todos los miembros del grupo.**
* **La metodología Scrum asegura entregas continuas, priorización de tareas y adaptación a las necesidades del cliente debido a sus continuas reuniones y trabajo colaborativo.**

**Aquí podemos encontrar el repositorio de** [**GitHub**](https://github.com/alegomezamaro/Equipo-205)

# Pruebas Unitarias

## 8.1 Diseño de Pruebas

|  |  |
| --- | --- |
| PU1 | Solicitud inscripción programa profesor SICUE |
| Fecha: 03/12/2024  Historia de Usuario: Como profesor de la UCO, quiero poder solicitar mi inscripción en el programa, para participar en el intercambio académico.  Responsable de la prueba: Alejandro Gómez Amaro  Descripción: Verificar que los usuarios pueden inscribirse en el programa completando todos los campos obligatorios y recibiendo un mensaje de confirmación al finalizar.  Prerrequisitos:   * El usuario debe estar autenticado en el sistema * Los planes de convalidación deben estar cargados.   Entradas:   * Nombre: “Ángel Correa Gaitán” * DNI: “12345678A” * Correo: i22cogaa@uco.es * Carrera: “Ingeniería Informática” * Departamento: “Departamento Ingeniería de Software”   Acciones a realizar:   1. Acceder con credenciales válidas al sistema 2. Navegar a la sección de inscripción 3. Completar los campos obligatorios 4. Enviar solicitud   Salida esperada:   1. Validación de datos en tiempo real 2. Si falta algún dato, mostrar mensaje de error 3. Al completar solicitud, mostrar confirmación   Salida obtenida:   * Solicitud registrada exitosamente.   Observaciones: Si hubiese algún error la salida seria distinta:   * Código del plan no válido. * Ya existe una solicitud registrada con este DNI. | |

|  |  |
| --- | --- |
| PU2 | Consulta de inscripción en el programa SICUE |
| Fecha: 05/12/2024  Historia de Usuario: Como alumno o profesor de la UCO, quiero consultar el estado de mi inscripción en el programa SICUE, para verificar mi participación y el plan seleccionado.  Responsable de la prueba: Jorge Santos Jiménez  Descripción: Verificar que los usuarios pueden consultar el estado de su inscripción, incluyendo la fecha de solicitud y los detalles del plan seleccionado.  Prerrequisitos:   1. El usuario debe estar autenticado 2. El sistema debe contener al menos una inscripción activa asociada al usuario   Entradas: Credenciales del usuario  Acciones a realizar:   1. Acceder al sistema con credenciales válidas 2. Navegar a “Estado de inscripción” 3. Calcular información de inscripción activa   Salida esperada: Si existe una inscripción activa, mostrar estado actual, fecha de solicitud y detalles del plan. En caso contrario, mostrar un mensaje indicando que no hay.  Salida obtenida:   * Se ha mostrado la solicitud, el orden de prioridad de planes de convalidación, la fecha de la solicitud y el estado de esta, “pendiente”.   Observaciones: El estado de la solicitud puede tener 3 posibles respuestas: Pendiente, aceptado o rechazado. | |

|  |  |
| --- | --- |
| PU3 | Creación de plan de convalidación profesores |
| Fecha: 08/12/2024  Historia de Usuario: Como administrador de la ORI, quiero acceder a los datos almacenados de los alumnos y profesores, para verificar su información de forma segura.  Responsable de la prueba: Alejandro Gómez Amaro  Descripción: Los administradores pueden crear nuevos planes de convalidación con los datos facilitados por la ORI.  Prerrequisitos:   * El usuario debe ser un administrador o miembro de la ORI. * Los datos de deben estar verificados por la ORI.   Entradas: Información de la universidad, carrera y departamento.  Acciones a realizar:   * Iniciar sesión como administrador o miembro de la ORI. * Acceder a la sección de "Crear plan de convalidación". * Introducir información de la universidad, carrera y departamento.   Salida esperada: El plan de convalidación se debe crear correctamente.  Salida obtenida: El plan de convalidación se ha creado sin problema.  Observaciones: Hay que tener en cuenta que los datos estén correctamente adjuntados debido a que serán los planes de convalidación que se adjuntaran a los usuarios . | |

|  |  |
| --- | --- |
| PU4 | Creación de plan de convalidación alumnos |
| Fecha: 09/12/2024  Historia de Usuario: Como administrador de la ORI, quiero acceder a los datos almacenados de los alumnos y profesores, para verificar su información de forma segura.  Responsable de la prueba: Alejandro Gómez Amaro  Descripción: Los administradores pueden crear nuevos planes de convalidación con los datos facilitados por la ORI.  Prerrequisitos:   * El usuario debe ser un administrador o miembro de la ORI. * Los datos de deben estar verificados por la ORI.   Entradas: Información de la universidad, carrera y asignaturas.  Acciones a realizar:   * Iniciar sesión como administrador o miembro de la ORI. * Acceder a la sección de "Crear plan de convalidación". * Introducir información de la universidad, carrera y asignaturas (Hemos introducido en el valor número de créditos un string).   Salida esperada: Error debido a que no puede haber un string como número de créditos.  Salida obtenida: La aplicación ha entrado en bucle y generado infinitas asignaturas.  Observaciones: Hay que tenerlo en cuenta para eliminar ese bug. | |

|  |  |
| --- | --- |
| PU5 | Envío de notificaciones por email sobre actualizaciones de solicitud |
| Fecha: 10/12/2024  Historia de Usuario: Como usuario de la UCO, quiero recibir actualizaciones de mi solicitud por email, para estar informado sobre el estado de mi inscripción.  Responsable de la prueba: Jorge Santos Jiménez  Descripción: Verificar que el sistema envíe notificaciones automáticas por correo electrónico cuando haya una actualización en la inscripción (aprobación, anulación, etc.).  Prerrequisitos:   * El usuario debe estar autenticado y tener un correo válido registrado. * El sistema debe tener configurado el envío de notificaciones.   Entradas: Credenciales y actualización de solicitud  Acciones a realizar:   * Iniciar sesión con las credenciales válidas. * Ingresar a la sección de notificaciones de usuario. * Cambiar la configuración de notificación de correo (si es necesario). * Realizar una acción que actualice el estado de la inscripción (por ejemplo, aprobación de solicitud).   Salida esperada: El sistema envía un email con la notificación de dicha actualización, si el correo es inválido, muestra mensaje de error solicitando corrección.  Salida obtenida: El aviso se ha enviado correctamente tanto al crear solicitud, como al eliminarla y cuando han actualizado su estado (Aceptado / denegado)  Observaciones: Las actualizaciones vía correo electrónico no se pueden desactivar. | |

|  |  |
| --- | --- |
| PU6 | Selección de Plan de Convalidación |
| Fecha: 13/12/2024  Historia de Usuario: Como estudiante participante, quiero seleccionar un plan de convalidación (cuatrimestre o curso completo), para gestionar la convalidación de asignaturas y prácticas según el convenio con la universidad asociada.  Responsable de la prueba: Raúl Castillo Díaz  Descripción: Verificar que el estudiante puede seleccionar correctamente entre un plan de cuatrimestre o curso completo, y que la opción seleccionada se refleja en la inscripción.  Prerrequisitos:   * El estudiante debe estar autenticado y ser elegible para seleccionar un plan de convalidación. * Los planes de convalidación deben estar disponibles en el sistema.   Entradas: Credenciales y selección del plan de convalidación  Acciones a realizar:   * Iniciar sesión con las credenciales válidas. * Acceder a la sección "Crear Solicitud". * Tras la solicitud se accede a la selección “Prioridad Plan de Convalidación”. * Elegir los planes deseados (En orden de prioridad). * Confirmar la selección y acabar con la inscripción.   Salida esperada:   * El sistema registra la selección del plan y lo refleja en el estado de la inscripción. * Si el estudiante no es elegible, el sistema muestra un mensaje indicando el motivo.   Salida obtenida:   * Una vez finalizada la selección de planes de convalidación se imprime un mensaje de “Solicitud guardada correctamente”   Observaciones: La cantidad de planes de convalidación elegidos no tiene límite, pero hay que tener en cuenta que ira en orden de prioridad. | |

## 8.2 Informe de Errores

|  |  |  |  |
| --- | --- | --- | --- |
| ID Prueba | Acción | Error | Solución |
| PU1 | Completar inscripción en el programa SICUE | Código del plan no válido o duplicado | Validar que el código del plan sea único con DNI antes de completar la inscripción |
| Faltan datos obligatorios en el formulario | Asegurar que todos los campos obligatorios están completados antes de permitir el envío |
| PU2 | Consultar estado de inscripción | No hay inscripciones activas asociadas al usuario | Verificar que el usuario tenga al menos una inscripción activa en el sistema |
| PU3 | Creación de plan de convalidación profesores | No hay planes de convalidación creados | Una vez creado el sistema deben introducirse un mínimo de planes para que los profesores puedan escoger |
| PU4 | |  | | --- | |  |  |  | | --- | | Creación de plan de convalidación alumnos | | Si se introduce erróneamente un valor incorrecto en un campo int rompe el fichero de almacenaje | Verificar que el valor se introduce correctamente de forma automática por el sistema |
| PU5 | Enviar notificaciones por correo electrónico | Correo electrónico del usuario no válido | Implementar una validación previa del correo electrónico y solicitar corrección si es inválido |
| PU6 | |  | | --- | |  |  |  | | --- | | Seleccionar plan de convalidación | | Usuario no elegible para seleccionar el plan de convalidación | Notificar al usuario sobre los criterios de elegibilidad antes de permitir la selección |

# Referencias

Para hacer el trabajo hemos usado ayuda de los siguientes archivos facilitados en Moodle:

* Practica 1.1 – Organización y apuntes
* Practica 1.1 – Diapositivas
* Guía de Trello
* Vídeo proyectos Scrum con Trello
* Practica 1.2 – Organización y apuntes
* Practica 1.2 – Diapositivas
* Especificación de Requisitos IEEE 830
* Métodos débiles de análisis
* Scrum historias de usuario
* Buenos prácticas casos de uso
* Ejemplo caso de uso
* Practica 2 – Organización y apuntes
* Practica 2 – Diapositivas
* Git documentación oficial
* GitHub documentación oficial
* GitHub Cheatsheet (HTML)
* Practica 4 – Organización y apuntes
* Practica 4 – Diapositivas
* Plantilla para pruebas
* Instrucciones documentación de practicas

Además, hemos usado tres webs para elaborar el trabajo:

* Visual Paradigm, para representar el diagrama de Casos de Uso, diagrama de Clases UML y diagramas de Secuencia.
* Chat GPT-4o, que nos ha ayudado a generar resúmenes y a elaborar Historias de Usuario y Casos de Uso según las directrices que le hemos facilitado, además nos ha facilitado el programa creando los menús y convirtiendo el diagrama de clases en clases.
* Trello, donde hemos desarrollado 4 tableros:
  + Practica 1, con los sprints correspondientes, las reuniones y las Historias de Usuario.
  + Practica 2, que incluye los sprints y las reuniones.
  + Practica 4, donde vemos los sprints y las reuniones.
  + Documentación, donde tenemos acceso directo a todos los links de Moodle, diagramas y carpeta compartida para que todos los miembros tengamos acceso a ellos.