PROBLEMA/NECESIDAD

El proyecto GRCU Manager se presenta como una solución viable para digitalizar y optimizar estos procesos manuales, automatizando las tareas más repetitivas y propensas a error. El sistema permitirá a los estudiantes registrar y organizar requerimientos de manera estructurada, priorizarlos utilizando técnicas como MoSCoW, crear y vincular casos de uso con sus requerimientos correspondientes a través de una matriz de trazabilidad interactiva, mantener un historial de cambios y versiones, incorporar hilos de comentarios para facilitar la colaboración y revisión, adjuntar documentos complementarios, y generar documentación estandarizada de forma automática.

OBJETIVO/PROPÓSITO DEL PROYECTO

Desarrollar una aplicación web, libre y accesible, que permita a estudiantes y docentes de las carreras de Analista de Sistemas (AdeS) y Licenciatura en Sistemas (LeS) gestionar requerimientos y casos de uso de manera integral, incorporando:

* Registro, priorización y trazabilidad bidireccional de requerimientos;
* Historial de cambios y versiones;
* Validación y retroalimentación mediante comentarios y adjuntos;
* Gestión de informes y reducción de tareas repetitivas.

El proyecto será implementado por el equipo 4 BYTES durante el cuatrimestre de la asignatura *Laboratorio de Desarrollo*, utilizando exclusivamente software libre y la infraestructura académica disponible, lo que asegura un costo económico nulo. Su éxito podrá medirse al finalizar la cursada en función de:

* La calidad del prototipo entregado (funcionalidades implementadas y estabilidad).
* La cobertura de los requerimientos planteados.
* El tiempo de desarrollo (ajustado al calendario académico).
* La viabilidad técnica para su despliegue en el entorno universitario.

REQUERIMIENTOS DE USUARIOS IDENTIFICADOS

1. **Registrar requerimientos** de manera ordenada y estructurada.
2. **Organizar y priorizar requerimientos** con la técnica **MoSCoW** (Must, Should, Could, Won’t).
3. **Mantener un historial de cambios y versiones** para reflejar la evolución de los requerimientos.
4. **Definir dependencias entre requerimientos**, para mejorar la trazabilidad y el análisis de impacto.
5. **Agrupar requerimientos en categorías o etiquetas**, para facilitar su organización.
6. **Construir una matriz de trazabilidad bidireccional** entre requerimientos y casos de uso.
7. **Hilo de comentarios por requerimiento**, que permita discutir, aclarar y registrar feedback de distintos usuarios.
8. **Adjuntar documentos y archivos externos** (PDF, Word, imágenes, diagramas) como apoyo a la documentación.
9. **Soporte multiproyecto**, permitiendo gestionar varios proyectos en paralelo.
10. **Definir distintos roles de usuario** por proyecto (ej.: administrador, documentador, verificador/tester, cliente/stakeholder).
11. **Validación por parte del cliente/stakeholder**, con posibilidad de confirmar requerimientos, comentar o asignar importancia.
12. **Aplicación web accesible**, sin necesidad de instalación local.
13. **Uso de software libre**, sin dependencias de licencias privativas.
14. **Generación automática de informes**, reduciendo la repetición de tareas manuales.
15. **Visualización interactiva de la trazabilidad**, con distintos niveles de detalle o “zoom” para recorrer los requerimientos.
16. **Flexibilidad metodológica**, permitiendo usar tanto PSI como enfoques ágiles u otros modelos de trabajo.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Observaciones de la presentación:

1. Disposición de los integrantes. Podemos **mover el espacio de trabajo al centro** para evitar acumularnos en el costado, y también hay que evitar estar tan separados entre sí para que sea más fácil para los alumnos vernos a todos (corrección que le hizo al equipo CoDevIT).

2. Mejorar el **Modelo de Dominio** y fijarse en la bibliografía de Ingeniería de Software.

3. Ver **archivo de sueldos** para realizar una estimación y analizar viabilidad.

4. Colocar **subtítulos** que sean de ayuda en la presentación tanto para nosotros como para los que están escuchando.

5. Explicar el Diagrama del Modelo de Dominio en el documento.

Observaciones del grupo NexTech:

1. Agregaron títulos de **misión y visión**, beneficios y en los avances agregaron la **creación de ramas** y los formatos de los commits, con <tipo> <área>: resumen.

2. Utilizaron los conventional commits que mostraron Hugo y Collareda: feat(), fix(), docs(), refactor(), chore(), test(), style().

En cuanto a "Mejorar la toma de decisiones", le resaltaron a los dos grupos qué decisiones serían esas. Tenerlo en cuenta en caso de que lo mencionemos en alguna presentación.

Nos sugirieron probar el software con datos reales porque eleva el nivel de presentación y también señalaron lo de **sistematizar los tiempos** (excel en nuestro caso) y **revisar el PDF** de las pautas.

Para todos los grupos nos señaló que cuando hablemos lo hagamos con información concreta, en el caso de los chicos de CoDevIT les dijeron que hablen de métricas en particular y se enfoquen en eso.

Y también nos resaltaron a todos nuevamente que nadie solicitó otra entrevista ni comentó dudas al respecto que había quedado pendiente de la vez pasada.