



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA
CARRERA DE INGENIERÍA DE SISTEMAS

**DISEÑO Y DESARROLLO DE UNA APLICACIÓN PARA EL REGISTRO Y
MONITOREO DE ESTUDIANTES QUE PARTICIPAN EN EL ASOCIACIONISMO
SALESIANO UNIVERSITARIO (ASU) Y TUTORÍAS ENTRE PARES AL ESTILO
SALESIANO (TEPES) PARA LA GESTIÓN, CONTROL Y ENTREGA DE
CERTIFICADOS**

Trabajo de titulación previo a la obtención del
título de Ingeniero de Sistemas

AUTOR: CHRISTIAN STANLEY NOVILLO PESANTEZ
TUTOR: ING. SANTIAGO TEODORO CASTRO ARIAS, MSC.

Cuenca - Ecuador

2024

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Christian Stanley Novillo Pesantez con documento de identificación N° 0103700498 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 20 de diciembre del 2024

Atentamente,



Christian Stanley Novillo Pesantez

0103700498

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Christian Stanley Novillo Pesantez con documento de identificación N° 0103700498, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: “Diseño y desarrollo de una aplicación para el registro y monitoreo de estudiantes que participan en el Asociacionismo Salesiano Universitario (ASU) y Tutorías entre Pares al Estilo Salesiano (TEPES) para la gestión, control y entrega de certificados”, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 20 de diciembre del 2024

Atentamente,



Christian Stanley Novillo Pesantez

0103700498

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Santiago Teodoro Castro Arias con documento de identificación N° 0104775473, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO Y DESARROLLO DE UNA APLICACIÓN PARA EL REGISTRO Y MONITOREO DE ESTUDIANTES QUE PARTICIPAN EN EL ASOCIACIONISMO SALESIANO UNIVERSITARIO (ASU) Y TUTORÍAS ENTRE PARES AL ESTILO SALESIANO (TEPES) PARA LA GESTIÓN, CONTROL Y ENTREGA DE CERTIFICADOS, realizado por Christian Stanley Novillo Pesantez con documento de identificación N° 0103700498, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 20 de diciembre del 2024

Atentamente,



Ing. Santiago Teodoro Castro Arias, MsC.

0104775473

DEDICATORIA

A mi familia, mi inspiración y mi mayor apoyo. A Johanna, mi compañera en cada paso de este camino, por compartir conmigo sueños, sacrificios y alegrías. A Nicolás y Joaquín, por su paciencia y amor incondicional, que me impulsaron a seguir adelante. A Julio y Regina, mis primeros maestros, por sus sabios consejos y ejemplo de vida, que me guiaron en este proceso de crecimiento personal y profesional. Esta dedicatoria es un pequeño homenaje a cada uno de ustedes, por ser parte fundamental de este logro.

RESUMEN

Con el objetivo de optimizar la gestión de los Grupos de Asociacionismo Salesiano Universitario (ASU) y las Tutorías entre Pares al Estilo Salesiano (TEPES), se ha desarrollado una solución tecnológica que automatiza los procesos asociados a estas actividades en la Universidad Politécnica Salesiana. El sistema está compuesto por dos módulos: ASU y TEPES.

El módulo de ASU permite la creación y gestión de grupos y eventos de ASU en el backend (Formularios para la administración del software), con perfiles de animadores, coordinadores y participantes. Además, incluye opciones de configuración para gestionar las áreas de acción establecidas, campus, sedes y períodos académicos de cada grupo. En el frontend, los estudiantes pueden inscribirse en los grupos y eventos a través de formularios de registro, lo que facilita la administración y el seguimiento del proceso. El sistema mantiene un historial detallado de los ciclos en los que los estudiantes han participado y los eventos a los que han asistido.

Por su parte, el módulo de TEPES también cuenta con un backend para la gestión de tutores y estudiantes, y para configurar los períodos académicos, sedes, campus y materias correspondientes a cada carrera. En el frontend, los estudiantes pueden solicitar tutorías mediante formularios, y los tutores pueden asignarse y registrar la duración. El historial de tutorías se almacena para su posterior consulta.

Ambos módulos integran la comunicación mediante correos electrónicos y mensajes de WhatsApp en cada etapa del proceso, proporcionando información necesaria del proceso a estudiantes, tutores, animadores, coordinadores y participantes.

El desarrollo del sistema ha priorizado la escalabilidad y usabilidad. La metodología ágil SCRUM, junto con el framework de Odoo, ha sido clave en la creación de esta plataforma, que reduce la carga administrativa y mejora la experiencia de usuarios, fortaleciendo la gestión de los Grupos ASU y TEPES.

Palabras clave: ASU, TEPES, Odoo, SCRUM, Backend, Frontend, Plataforma

ABSTRACT

With the aim of optimizing the management of the Salesian University Association Groups (ASU) and Peer Tutoring in the Salesian Style (TEPES), a technological solution has been developed to automate the processes associated with these activities at the Universidad Politécnica Salesiana. The system consists of two modules: ASU and TEPES.

The ASU module allows for the creation and management of ASU groups and events in the backend, with profiles for facilitators, coordinators, and participants. Additionally, it includes configuration options to manage each group's areas of action, campus, locations, and academic periods. In the frontend, students can register for groups and events through registration forms, streamlining the administration and tracking of the process. The system maintains a detailed record of the cycles in which students participated and the events they attended.

The TEPES module also features a backend for managing tutors and students, as well as configuring academic periods, locations, campuses, and courses corresponding to each program. In the frontend, students can request tutoring sessions through forms, and tutors can assign themselves and log the duration of each session. The tutoring history is stored for future reference.

Both modules integrate communication through emails and WhatsApp messages at each stage of the process, providing necessary information to students, tutors, facilitators, coordinators, and participants.

The system's development prioritized scalability and usability. The agile SCRUM methodology, along with the Odoo framework, has been key in creating this platform, which reduces administrative workload and enhances the user experience, strengthening the management of ASU and TEPES Groups.

Keywords: ASU, TEPES, Odoo, SCRUM, Backend, Frontend, Platform

Introducción	9
Problema.....	10
Justificación.....	10
Objetivos	11
General.....	11
Específicos.....	11
Revisión de la literatura o fundamentos teoricos	12
Introducción a Odoo	12
Arquitectura de Odoo	12

Postgresql como base de datos para Odoo.....	14
Arquitectura de Postgresql.....	15
Python como lenguaje de desarrollo para Odoo	18
Arquitectura de Odoo con python	19
Ubuntu servidor como entorno para Docker Odoo	19
Arquitectura de Docker en Ubuntu servidor.....	20
Introducción a la gestión de contenedores.....	20
Arquitectura de Portainer.....	21
Control de versiones Git.....	21
Arquitectura de Git	22
Marco metodológico	23
Propuesta de solución	23
Diseño arquitectónico	23
Módulos del sistema:	23
Metodologia	23
Pasos para implementar Odoo con scrum.....	25
Descripción de actividades de los objetivos del proyecto	26
1. Módulo ASU (G. Asociacionismo Salesiano Universitario)	28
2. Módulo TEPES (Tutorías entre Pares al Estilo Salesiano).....	29
3. Comunicación y Notificaciones	30
4. Gestión Administrativa	30
5. Mantenimiento y mejoras	31
Resultados	31
Sprint 1: Configuración inicial y base del sistema	31
Sprint 2: Gestión de Grupos y Eventos ASU	33
Sprint 3: Gestión de tutorías en TEPES.....	36
Sprint 4: Notificaciones y comunicaciones	38
Sprint 5: Pruebas y optimización.....	40
Sprint 6: Finalización y documentación	40
Cronograma.....	41
Presupuesto.....	43
Conclusiones.	44
Recomendaciones.....	45

Introducción

La gestión eficiente de las actividades extracurriculares desempeña un papel importante en la Universidad Politécnica Salesiana, por lo que han creado Grupos de Asociacionismo Salesiano Universitario (ASU) para impulsar y promover espacios de expresión juvenil universitario en diferentes ámbitos de participación al estilo salesiano y un plan de acompañamiento tutorial entre Pares al Estilo Salesiano (TEPES), para el desarrollo integral de los estudiantes.

Este proyecto presenta una solución tecnológica diseñada específicamente para optimizar la administración de los Grupos ASU y el programa TEPES, facilitando la organización de eventos, la asignación de tutores y la inscripción de estudiantes, todo ello mediante una plataforma digital integrada. La automatización de estos procesos no solo busca reducir la carga administrativa, sino también mejorar la experiencia de todos los involucrados, brindando una herramienta eficiente que permita gestionar de manera centralizada y transparente estas actividades extracurriculares.

Problema

Los Grupos de Asociacionismo Salesiano Universitario (ASU) y el Grupo de Tutorías entre Pares al estilo Salesiano (TEPES) de la Universidad Politécnica Salesiana, en la actualidad no cuentan con sistema para la gestión eficiente de información sobre estudiantes que asisten a sus programas. La falta de una herramienta que automatice el control y monitoreo genera inconvenientes dentro del proceso de Gestión de los Grupos.

Ente los principales inconvenientes tenemos:

- Falta de integridad de datos, la información sobre los estudiantes, como nombres, correos electrónicos e identificación, no se encuentra centralizada y segura, lo que aumenta el riesgo de errores, inconsistencias y pérdida de datos.
- Dificultad para el seguimiento de actividades, la ausencia de un sistema automatizado impide un seguimiento eficaz de la participación de los estudiantes, limitando la capacidad para reconocer y premiar el compromiso estudiantil.
- Archivos en Excel, el uso de hojas de cálculo para gestionar la información de los estudiantes resulta poco eficiente y escalable, dificultando el manejo de grandes volúmenes de datos y aumentando la probabilidad de errores humanos.
- Limitaciones para la consulta y análisis de datos históricos sobre los estudiantes, lo que dificulta la toma de decisiones informadas y la identificación de tendencias relevantes.
- Dificultad para generar informes precisos y actualizados sobre la población estudiantil que participan en ASU y TEPES.

Justificación

Los Grupos ASU actúan sobre varias áreas de acción entre las que tenemos Académico, Comunicacional, Cultural, Deportivo y Socio Político, en la actualidad se encuentran en funcionamiento en la Sede Cuenca con 33 Grupos, Guayaquil con 30 Grupos y Quito con 44 Grupos con un gran número de estudiantes inscritos.

En 2014, la SENESCYT implementó la "política de cuotas" para permitir el acceso a la universidad a grupos históricamente excluidos. La UPS implementó la "tutoría entre pares al estilo salesiano" (TEPES) y la "tutoría docente" para prevenir la deserción estudiantil. Esta iniciativa se encuentra presente en Cuenca, Quito y Guayaquil. En el periodo académico 2023-2023 se desarrollan tutorías a 434 estudiantes y en el periodo 2023-2024 a 219 estudiantes.

Con el avance de la tecnológica estos Grupos enfrentan desafíos en la gestión de sus estudiantes, la implementación de un software permitirá mejorar la integridad de datos almacenados evitando información dispersa en diferentes formatos y sistemas, facilidad para escalar los procesos a medida que aumenta la población estudiantil, menor probabilidad de errores humanos y retrasos en la gestión de la información, con información precisa y oportuna para identificar estudiantes y brindar apoyo oportuno.

Al implementar un sistema único y centralizado para gestionar inscripciones, automatiza procesos manuales y repetitivos, reduciendo tiempos y costos, facilita el acceso a datos históricos y la generación de informes permitiendo la consulta y análisis de datos históricos sobre la población estudiantil en el cual se puede identificar tendencias y patrones relevantes para la toma de decisiones informadas, generar informes precisos y actualizados por Sedes y Grupos tanto ASU como TEPES sobre inscripciones y la participación estudiantil.

Objetivos

General

Desarrollar un sistema para el registro y monitoreo de estudiantes que participan en el Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES) para la gestión, control y entrega de certificados.

Específicos

- **OE1.** Estudiar y analizar los procesos del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).

- **OE2.** Definir e implementar los requerimientos funcionales para el sistema de gestión del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).
- **OE3.** Desarrollar una plataforma web que permita la gestión y monitoreo del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).
- **OE4.** Desarrollar un módulo para la consulta y registro de estudiantes del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).
- **OE5.** Desarrollar un módulo para el control y manejo de Grupos ASU y TEPES, creación, asignación de tutores, periodos académicos, entrega de certificados y comunicación mediante email.

Revisión de la literatura o fundamentos teoricos

Introducción a Odoo

Odoo es una plataforma de gestión empresarial de código abierto que ofrece un conjunto completo de aplicaciones para cubrir diversas áreas funcionales, como ventas, contabilidad, recursos humanos, inventario y más. Una de sus principales ventajas es su carácter modular, lo que permite a las organizaciones seleccionar y utilizar solo las aplicaciones que necesiten, adaptando la plataforma a sus requerimientos específicos. Además, al ser escalable, Odoo es adecuado tanto para pequeñas empresas como para grandes instituciones, como universidades, que buscan gestionar una variedad de procesos dentro de un solo sistema integrado.

Arquitectura de Odoo

La arquitectura de Odoo sigue el modelo cliente-servidor. El servidor es responsable de manejar y procesar toda la lógica de negocio, así como de almacenar los datos en una base de datos PostgreSQL. Odoo está desarrollado principalmente en Python, un lenguaje de programación flexible y fácil de mantener. Por su parte, el cliente (la interfaz web) está diseñado para facilitar la interacción de los usuarios con el sistema a través de navegadores web. Esta separación de responsabilidades permite que el sistema sea eficiente, escalable y fácil de actualizar. La interfaz web se basa en tecnologías como HTML, CSS y JavaScript para garantizar una experiencia de usuario dinámica e interactiva.

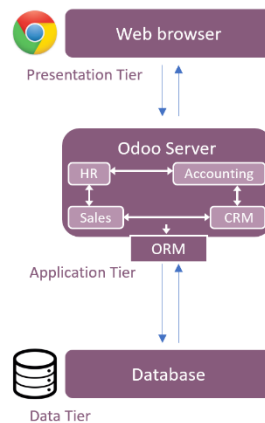


Fig.1 Arquitectura Odoo

Desarrollo de aplicaciones con Odoo

El desarrollo de aplicaciones en Odoo se realiza utilizando su framework, el cual está basado en Python. Este framework permite crear módulos personalizados que se integran a la plataforma principal de manera fluida. Las aplicaciones pueden extender o modificar la funcionalidad de Odoo mediante la reutilización de componentes existentes, lo que reduce el tiempo de desarrollo. Además, el uso de XML para definir vistas y formularios permite una personalización avanzada de la interfaz de usuario. Los desarrolladores tienen acceso a una amplia gama de herramientas proporcionadas por el framework, como ORM (Object-Relational Mapping), controladores para la interfaz web, y mecanismos de seguridad y acceso.

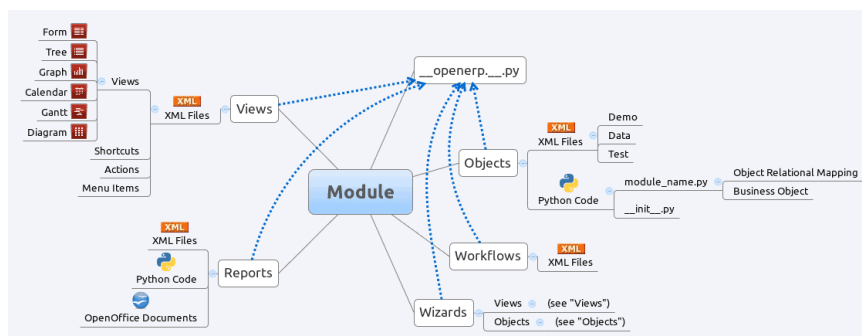


Fig.2 Arquitectura Módulos Odoo

Metodologías para el desarrollo de aplicaciones con Odoo

Para llevar a cabo un desarrollo eficiente en Odoo, se suelen adoptar metodologías ágiles como Scrum. Estas metodologías permiten una gestión efectiva del proyecto mediante la entrega incremental de funcionalidades y la capacidad de adaptarse a cambios en los requerimientos.

En proyectos de gran envergadura, como el desarrollo de sistemas para universidades o empresas con múltiples procesos interconectados, Scrum facilita la organización del trabajo en sprints cortos, permitiendo iteraciones rápidas y revisiones constantes. Realizando estos pasos podemos validar que el producto final cumpla con las expectativas del cliente y sea de alta calidad.

Buenas prácticas para el desarrollo de aplicaciones con Odoo

El desarrollo de aplicaciones con Odoo implica seguir una serie de buenas prácticas para garantizar que el código sea eficiente, fácil de mantener y escalable. Entre las principales recomendaciones se encuentran:

- Uso adecuado de control de versiones con la implementación herramientas como Git para gestionar cambios en el código, permitiendo un trabajo colaborativo y organizado, además de un historial completo de las modificaciones.
- Código limpio y bien documentado, siguiendo convenciones estándar, y acompañarlo de comentarios y documentación técnica que facilite su comprensión por otros desarrolladores.
- Pruebas unitarias y de integración para asegurar que cada componente funcione correctamente, tanto de forma aislada (pruebas unitarias) como en conjunto con otros módulos (pruebas de integración).
- Revisión de código por pares para asegurar la calidad del código antes de integrarlo a la rama principal. Esto reduce errores y fomenta el intercambio de conocimiento entre el equipo de desarrollo.

Estas prácticas no solo optimizan el ciclo de desarrollo, sino que también contribuyen a la estabilidad y robustez del sistema, reduciendo el riesgo de errores en producción y mejorando la experiencia de los usuarios.

Postgresql como base de datos para Odoo

PostgreSQL es una base de datos relacional de código abierto reconocida por su robustez, escalabilidad y fiabilidad. Elegida la base de datos predeterminada para Odoo, PostgreSQL proporciona el soporte necesario para manejar las exigencias de las aplicaciones empresariales

modernas, permitiendo una integración fluida con la plataforma Odoo. A continuación, se analizarán en detalle las características clave de PostgreSQL, su arquitectura, procesos de implementación, backup y otras funcionalidades avanzadas que lo posicionan como una solución ideal de bases de datos en el entorno de Odoo.

Arquitectura de Postgresql

PostgreSQL sigue una arquitectura cliente-servidor que se compone de varios componentes esenciales que interactúan entre sí para gestionar las operaciones de almacenamiento, recuperación y manipulación de datos. Los principales componentes son:

- Postmaster proceso principal de PostgreSQL, encargado de iniciar otros procesos y gestionar conexiones con los clientes.
- Procesos de Backend que manejan las conexiones de clientes, procesan las consultas y gestionan la ejecución de transacciones.
- Buffer de Compartición que almacena los datos recuperados de disco en la memoria, optimizando el rendimiento del sistema.
- Sistema de Archivos y Memoria que utiliza técnicas avanzadas de almacenamiento en disco y gestión de memoria para garantizar un rendimiento óptimo.
- WAL (Write-Ahead Logging) que registra las transacciones antes de aplicarlas a la base de datos, lo que garantiza la recuperación ante fallos.

Esta arquitectura permite que PostgreSQL maneje múltiples usuarios y transacciones simultáneamente, con una eficiente gestión de los recursos del sistema y un control de concurrencia eficaz mediante el Control de Concurrencia Multiversión (MVCC).

Procesos de implementación de Postgresql en Odoo

La implementación de PostgreSQL como base de datos para Odoo sigue un proceso detallado que asegura que ambas tecnologías se integren de manera efectiva y puedan escalar de acuerdo con las necesidades de la organización. Los pasos básicos para la implementación incluyen:

- PostgreSQL puede instalarse en múltiples plataformas (Linux, Windows, macOS), dependiendo de los requerimientos de la organización. Odoo se comunica con PostgreSQL mediante una conexión directa, permitiendo la creación de instancias de bases de datos específicas para cada instancia de Odoo.

- Configuración Inicial que implica ajustes en los parámetros clave como el tamaño del buffer, los límites de conexiones y la configuración del archivo postgresql.conf. Es importante optimizar la configuración de memoria y de disco para asegurar que PostgreSQL pueda manejar grandes volúmenes de datos y usuarios concurrentes.
- Seguridad y Acceso que proporciona mecanismos avanzados de seguridad como la autenticación basada en roles y el uso de SSL para cifrar las conexiones. La implementación de controles de acceso a nivel de usuario y de base de datos es esencial en un entorno como Odoo, donde se manejan datos sensibles de las empresas.
- Replicación y Escalabilidad que soporta varios tipos de replicación, como la replicación en caliente (Hot Standby) y la replicación lógica, que permiten distribuir la carga entre varias instancias de la base de datos y garantizar la disponibilidad del sistema en caso de fallos.

Mecanismos de backup en PostgreSQL

El sistema de backup en PostgreSQL es fundamental para asegurar la integridad y disponibilidad de los datos. Existen diferentes opciones de backup que permiten recuperar la base de datos en caso de fallos:

- Backup en Caliente (Hot Backup) que utiliza la replicación en tiempo real para crear copias de seguridad sin interrumpir las operaciones de la base de datos. Este método es particularmente útil para grandes implementaciones de Odoo donde el tiempo de inactividad no es aceptable.
- pg_dump herramienta incorporada en PostgreSQL que permite realizar copias de seguridad a nivel de base de datos o tablas individuales. Este método es útil para copias de seguridad completas o parciales de Odoo y para migraciones entre diferentes versiones de PostgreSQL.
- WAL Archiving que permite recuperar la base de datos hasta un punto específico en el tiempo utilizando los registros de escritura adelantada (WAL). Este método asegura que, en caso de un fallo grave, se puedan restaurar los datos hasta el último registro antes del fallo.

- Backup Incremental lo que reduce significativamente el tiempo de copia de seguridad y el espacio de almacenamiento necesario al guardar solo los cambios realizados desde el último backup completo.

Opciones avanzadas en postgresql

PostgreSQL ha evolucionado significativamente en los últimos años, añadiendo varias características avanzadas que son particularmente útiles en el contexto de Odoo:

- Soporte para JSON y JSONB que permite el almacenamiento y manipulación de datos JSON, lo que es útil en Odoo para manejar datos semiestructurados. La versión JSONB optimiza las búsquedas y actualizaciones de estos datos, mejorando el rendimiento en consultas complejas.
- Particionamiento de Tablas funcionalidad permite dividir tablas grandes en subtablas más pequeñas basadas en criterios específicos, lo que mejora el rendimiento en consultas y reduce la fragmentación de los datos. Es particularmente útil en Odoo cuando se manejan grandes cantidades de datos históricos.
- Índices Avanzados, diferentes tipos de índices, como índices GIN, GiST y BRIN, que optimizan las búsquedas en conjuntos de datos grandes y específicos. Esto es beneficioso en Odoo para mejorar el rendimiento en aplicaciones que requieren consultas rápidas y eficientes.
- Extensiones, permite añadir funcionalidades mediante extensiones. Algunas de las más populares son PostGIS, que añade capacidades geoespaciales a la base de datos, y pg_stat_statements, que permite monitorizar el rendimiento de las consultas. Estas extensiones ofrecen la posibilidad de adaptar PostgreSQL a necesidades específicas de cada implementación de Odoo.
- Replicación y Alta Disponibilidad, ofrece replicación síncrona y asíncrona, lo que garantiza la disponibilidad de la base de datos ante fallos. Además, con la ayuda de herramientas de terceros como pgpool-II y Patroni, es posible configurar clústeres de bases de datos para asegurar una alta disponibilidad y balanceo de carga en entornos Odoo de gran escala.

PostgreSQL se ha consolidado como una base de datos avanzada y confiable, ideal para la integración con Odoo debido a su soporte para transacciones ACID, su escalabilidad y

rendimiento, su capacidad de replicación y backup, y su comunidad activa que contribuye al constante mejoramiento del sistema. Las características avanzadas como el soporte para JSON, la replicación de bases de datos, y las opciones de particionamiento de tablas y extensiones, hacen de PostgreSQL una solución altamente adaptable a las necesidades complejas de las organizaciones que implementan Odoo. Estas capacidades permiten a PostgreSQL no solo manejar grandes volúmenes de datos y usuarios concurrentes, sino también asegurar la continuidad operativa y la integridad de los datos, lo que lo convierte en la base sólida para el éxito de proyectos empresariales basados en Odoo.

Python como lenguaje de desarrollo para Odoo

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que ha ganado popularidad debido a su sintaxis clara, simplicidad y enfoque en la legibilidad del código. Desarrollado por Guido van Rossum y lanzado en 1991, Python se ha convertido en uno de los lenguajes más utilizados en el desarrollo de aplicaciones web, análisis de datos, inteligencia artificial, automatización de sistemas y más.

En el contexto de Odoo, Python es fundamental ya que el framework en el que está construido Odoo está basado en este lenguaje. Gracias a su flexibilidad y facilidad de uso, Python permite que los desarrolladores implementen soluciones empresariales robustas y escalables con una curva de aprendizaje moderada.

Python y Odoo

Odoo es un ERP modular y de código abierto que facilita la gestión empresarial de diferentes áreas, como ventas, inventarios, contabilidad, recursos humanos, y más. Python es el núcleo del desarrollo de Odoo, lo que significa que las aplicaciones y módulos que componen el sistema están escritos principalmente en Python, permitiendo a los desarrolladores:

- Personalizar módulos existentes.
- Desarrollar nuevos módulos adaptados a necesidades específicas.
- Integrar Odoo con otros sistemas y servicios.

Python es responsable de manejar la lógica empresarial en Odoo, que se conecta con su motor de base de datos, el ORM (Object Relational Mapper), para interactuar con PostgreSQL, la base de datos subyacente.

Arquitectura de Odoo con python

Odoo sigue una arquitectura de tres capas (cliente-servidor):

- Cliente: Es la interfaz web o cliente móvil que los usuarios interactúan.
- Servidor: Esta capa está escrita en Python y es donde reside la lógica empresarial. A través de Python, Odoo procesa las reglas de negocio, ejecuta funciones del sistema y gestiona las interacciones entre los módulos.
- Base de Datos (PostgreSQL): Python en Odoo utiliza un ORM que facilita la interacción con PostgreSQL sin que los desarrolladores deban escribir consultas SQL complejas.

Ubuntu servidor como entorno para Docker Odoo

Docker es una plataforma de código abierto que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores portátiles. Los contenedores Docker proporcionan un entorno aislado que incluye todo lo necesario para ejecutar una aplicación, lo que asegura que funcione de manera consistente en cualquier entorno, ya sea en desarrollo, pruebas o producción. Esta tecnología facilita la implementación y gestión de aplicaciones al ofrecer ventajas como portabilidad, escalabilidad y eficiencia en el uso de recursos.

Ubuntu Server es una de las distribuciones de Linux más utilizadas en servidores y entornos de producción debido a su estabilidad, seguridad y facilidad de administración. Es una opción excelente para ejecutar Docker, dado que proporciona un entorno robusto y compatible con la mayoría de las tecnologías actuales. Ubuntu Server facilita la instalación y gestión de Docker, permitiendo a los administradores desplegar aplicaciones en contenedores con facilidad.

La implementación de Odoo en contenedores Docker ofrece varias ventajas, incluyendo:

- Portabilidad: Docker asegura que la aplicación Odoo funcione de la misma manera en diferentes entornos, desde el desarrollo hasta la producción.
- Escalabilidad: Los contenedores Docker pueden ser fácilmente escalados para manejar mayores volúmenes de tráfico o cargas de trabajo.
- Aislamiento: Cada contenedor opera en un entorno separado, lo que minimiza los conflictos entre aplicaciones y facilita la gestión de dependencias.

Arquitectura de Docker en Ubuntu servidor

La arquitectura de Docker en Ubuntu Server sigue un modelo basado en contenedores que incluye:

- Docker Engine es el componente principal que ejecuta y gestiona los contenedores. En Ubuntu Server, Docker Engine se instala como un servicio que maneja la ejecución y orquestación de contenedores.
- Contenedores Docker son entornos aislados que ejecutan instancias de Odoo. Cada contenedor incluye la aplicación Odoo y todas sus dependencias necesarias, como el servidor web y las librerías de Python.
- Imágenes docker son plantillas inmutables usadas para crear contenedores. Las imágenes de odoo se pueden descargar desde repositorios públicos como docker hub o se pueden construir a partir de un dockerfile personalizado.
- Docker compose es una herramienta que permite definir y ejecutar aplicaciones multi-contenedor. Con docker compose, se puede configurar servicios como odoo y postgresql en un solo archivo YAML y gestionar su implementación de manera simplificada.

Introducción a la gestión de contenedores

Con el crecimiento de las aplicaciones basadas en contenedores, la necesidad de herramientas para gestionar estas tecnologías ha crecido exponencialmente. Docker, Kubernetes y otras plataformas de contenedorización permiten que los desarrolladores y administradores de sistemas creen, implementen y gestionen aplicaciones en entornos aislados. Sin embargo, la gestión manual de estos contenedores puede volverse compleja a medida que aumentan en número y diversidad. Es en este contexto donde surgen herramientas como **Portainer**, que ofrecen una solución visual e intuitiva para la gestión de contenedores.

Portainer es una plataforma de gestión de contenedores de código abierto que proporciona una interfaz gráfica (GUI) para la administración de entornos Docker y Kubernetes. Su objetivo principal es simplificar la interacción con los contenedores, permitiendo a los usuarios gestionar sus entornos sin necesidad de recurrir a comandos en la línea de terminal.

Portainer permite a los usuarios controlar múltiples aspectos de sus contenedores, como el despliegue, la monitorización, la administración de redes y volúmenes, la gestión de imágenes, y la orquestación de aplicaciones. Además, ofrece una capa adicional de seguridad y control a través de políticas de acceso y permisos de usuario.

Arquitectura de Portainer

La arquitectura de Portainer está diseñada para ser ligera y fácil de implementar en entornos tanto de desarrollo como de producción. Se basa en dos componentes principales:

- Portainer Server es el corazón de la plataforma, encargado de gestionar los recursos de Docker o Kubernetes. Se ejecuta como un contenedor y actúa como un punto central para recibir las solicitudes de los usuarios y distribuir las a los entornos subyacentes de Docker o Kubernetes.
- Portainer Agent es el componente que facilita la gestión de múltiples nodos Docker. Se comunica con el servidor de Portainer, lo que permite escalar la administración a través de varios hosts de Docker y Kubernetes.

Control de versiones Git

Git es un sistema de control de versiones distribuido que permite a los desarrolladores gestionar y realizar un seguimiento de los cambios en el código fuente de un proyecto. Es una herramienta esencial en el desarrollo de software, ya que permite colaborar de manera eficiente, mantener el historial de cambios, y gestionar diferentes versiones de un proyecto. Git es ampliamente utilizado en el desarrollo de proyectos modulares y escalables como Odoo, facilitando la colaboración entre equipos y la integración continua de funcionalidades.

Importancia de Git en el desarrollo de Odoo

Odoo es un sistema ERP modular y de código abierto, lo que significa que los desarrolladores pueden crear y personalizar módulos según las necesidades específicas de cada organización. En este contexto, Git se convierte en una herramienta crucial para la implementación y desarrollo de Odoo, ya que:

- Colaboración Eficiente entre múltiples desarrolladores al permitir que trabajen en diferentes partes del código de forma simultánea sin interferencias. Cada desarrollador puede trabajar en su propia rama sin afectar el código principal.
- Control de Versiones permite mantener un registro detallado de los cambios en el código de los módulos de Odoo, facilitando la identificación de errores y la restauración a versiones anteriores si es necesario.
- Desarrollo Modular, Odoo se basa en módulos que pueden ser desarrollados y mantenidos de manera independiente. Con Git, cada módulo puede tener su propio repositorio o gestionarse dentro de un repositorio monolítico, permitiendo un desarrollo ordenado y controlado.
- Integración Continua (CI) se integra fácilmente con herramientas de integración continua (CI), lo que permite la automatización de pruebas y despliegues en el entorno de Odoo. Esto asegura que los cambios en el código sean probados automáticamente antes de ser desplegados en producción.

Arquitectura de Git

Git utiliza una arquitectura distribuida, lo que significa que cada desarrollador tiene una copia completa del repositorio en su máquina local. Esto facilita el trabajo sin conexión y mejora la seguridad y disponibilidad del código. Los componentes principales de Git son:

- Repositorio Local en donde cada desarrollador tiene una copia del proyecto con todo su historial de versiones. Esto permite realizar cambios y pruebas de manera independiente.
- Repositorio Remoto alojado en plataformas como GitHub, GitLab o Bitbucket, el repositorio remoto es el lugar donde se almacenan y comparten los cambios para que otros miembros del equipo puedan acceder a ellos.
- Branching y Merging permite la creación de ramas (branches) que separan el trabajo en curso del código estable. Esto es útil en Odoo, ya que permite desarrollar nuevas funcionalidades o corregir errores sin interrumpir el funcionamiento del sistema principal.
- Commits cada vez que se guarda un cambio en el código, se realiza un "commit" que registra esos cambios. Esto proporciona un historial detallado de qué cambios se realizaron, cuándo y por quién, lo que es útil para la auditoría y depuración en Odoo.

Marco metodológico

Propuesta de solución

Diseño arquitectónico

Se realizará una implementación mediante Docker para el Servidor de Odoo y el Servidor de PostgreSQL, basados en los repositorios oficiales de Docker Hub.

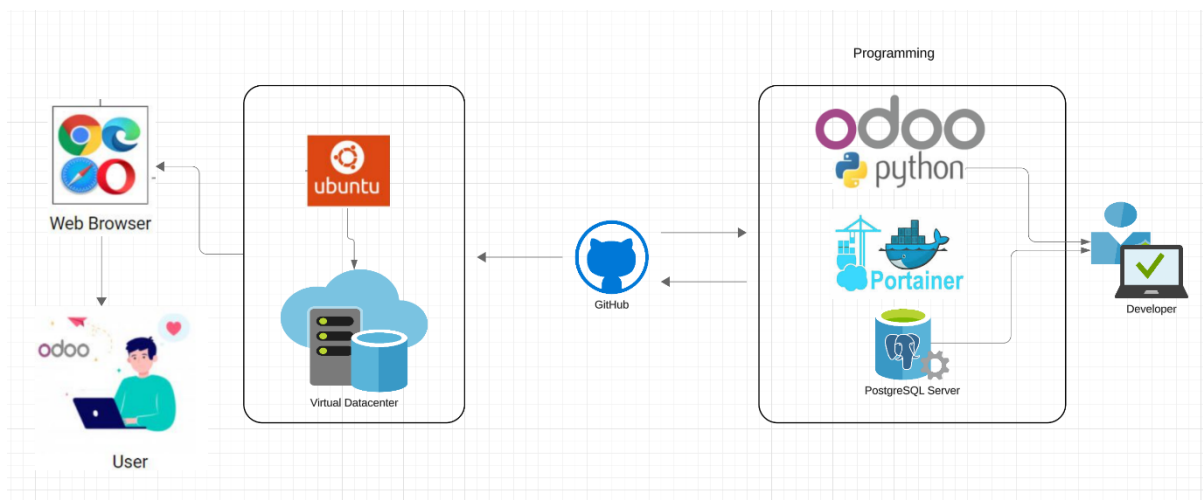


Fig.3 Arquitectura Odoo ASU y TEPES

Módulos del sistema:

De acuerdo con el problema de estudio se procederá a desarrollar un sistema web para la Gestión de Grupos ASU y TEPES, que permitirá realizar una gestión integral con los siguientes módulos:

- Estudiantes (Registro de estudiantes, intereses)
- Tutores o responsables (Habilitación de estudiantes, asistencia o aprobación)
- Coordinadores de Proyectos (Crear grupos ASU y TEPES, Asignar tutores)
- Administración (Crear usuarios del sistema, Parametrización del sistema.)

Metodologia

Beneficios de usar scrum para implementar odoo

Los beneficios de usar Scrum para implementar Odoo incluyen:

- Mayor flexibilidad y capacidad de adaptación que permite a los equipos adaptar el proyecto a medida que avanza, en función de los comentarios de los usuarios y las necesidades cambiantes del negocio.
- Mejor comunicación y colaboración que fomenta la comunicación y la colaboración entre todos los miembros del equipo, lo que conduce a una mejor comprensión del proyecto y mejores resultados.
- Entrega más rápida de valor que divide el proyecto en sprints cortos, lo que permite al equipo entregar valor al cliente con mayor frecuencia.
- Mayor satisfacción del cliente mediante la participación en el proceso de desarrollo de Scrum conduce a una mayor satisfacción del cliente con el producto final.

Roles en scrum para la implementación de Odoo

Los roles principales en Scrum para la implementación de Odoo son:

- **Scrum Master:** El Scrum Master es responsable de facilitar el proceso de Scrum y eliminar los impedimentos para el equipo.
- **Product Owner:** El Product Owner es responsable de definir los requisitos del producto y priorizar las funcionalidades.
- **Equipo de desarrollo:** El equipo de desarrollo es responsable de diseñar, desarrollar y probar el software Odoo.

Artefactos de scrum para la implementación de Odoo

Los artefactos principales de Scrum para la implementación de Odoo son:

- **Product Backlog:** El Product Backlog es una lista priorizada de todas las funcionalidades que se deben implementar en Odoo.
- **Sprint Backlog:** El Sprint Backlog es una lista de las funcionalidades que se implementarán en un sprint determinado.
- **Sprint Burndown Chart:** El Sprint Burndown Chart es un gráfico que muestra el progreso del equipo hacia la finalización del Sprint Backlog.

Eventos de scrum para la implementación de Odoo

Los eventos principales de Scrum para la implementación de Odoo son:

- **Sprint Planning:** El Sprint Planning es una reunión en la que el equipo de desarrollo y el Product Owner planifican el próximo sprint.
- **Sprint Review:** El Sprint Review es una reunión en la que el equipo de desarrollo muestra el trabajo completado en el sprint al Product Owner y a las partes interesadas.
- **Sprint Retrospective:** El Sprint Retrospective es una reunión en la que el equipo de desarrollo reflexiona sobre el sprint anterior, identifica las áreas de mejora y planifica las mejoras para el próximo sprint.

Pasos para implementar Odoo con scrum

Los pasos para implementar Odoo con Scrum son:

1. Definir los objetivos del proyecto de manera clara y concisa antes de comenzar a implementarlo.
2. Crear un Product Backlog para incluir todas las funcionalidades que se deben implementar en Odoo. El Product Owner debe priorizar las funcionalidades en función del valor que aportarán al negocio.
3. Seleccionar un equipo de desarrollo que debe tener las habilidades y la experiencia necesarias para implementar Odoo con éxito.
4. Elegir un Scrum Master que debe ser un facilitador eficaz y tener un buen conocimiento de la metodología Scrum.
5. Planificar el primer sprint con el equipo de desarrollo y el Product Owner. Esto implica seleccionar las funcionalidades del Product Backlog que se implementarán en el sprint y estimar el tiempo que llevará completarlas.
6. Implementar las funcionalidades del Sprint Backlog con el equipo de desarrollo. El Scrum Master debe facilitar el proceso y eliminar los impedimentos para el equipo.
7. Revisar el sprint donde el equipo de desarrollo muestra el trabajo completado en el sprint al Product Owner y a las partes interesadas. El Product Owner proporciona comentarios y prioriza las funcionalidades para el siguiente sprint.

8. Retrospectiva del sprint, el equipo de desarrollo reflexiona sobre el sprint anterior, identifica las áreas de mejora y planifica las mejoras para el próximo sprint.

9. Repetir los pasos 6 a 8 hasta que se implementen todas las funcionalidades del Product Backlog.

Descripción de actividades de los objetivos del proyecto

- OE1. Estudiar y analizar los procesos del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).

No.	Actividad
1	Análisis de los procesos del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES)
2	Entrevistas con los miembros del Grupo ASU y Tutorías Entre Pares al Estilo Salesiano (TEPES) para obtener información del estado actual de los procesos.
3	Proceso de validación de información.
4	Realizar una documentación de la información relevante para la realización del sistema.

Tabla 1. Actividades OE1

- OE2. Definir e implementar los requerimientos funcionales para el sistema de gestión del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).

No.	Actividad
1	Toma de requerimientos funcionales mediante entrevistas con los miembros de los Grupos ASU y TEPES.
2	Definir la información o parámetros necesarios para los módulos a desarrollar de acuerdo con los requerimientos funcionales
3	Validación y aprobación de los módulos a desarrollar.

Tabla 2. Actividades OE2

- OE3. Diseñar y desarrollar una plataforma web que permita la gestión y monitoreo del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).

No.	Actividad
-----	-----------

1	Análisis requerimientos funcionales mediante entrevistas con los miembros de los Grupos ASU y TEPES.
2	Diseño para el desarrollo de los módulos aprobados, para posteriormente asignar una fecha y el responsable.
3	Validación y aprobación de los diseños.

Tabla 3. Actividades OE3

- OE4. Desarrollar un módulo para la consulta y registro de estudiantes del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).

No.	Actividad
1	Revisión del requerimiento funcional de registro de estudiantes para los Grupos ASU y TEPES
2	Desarrollo del módulo de registro de estudiantes
3	Pruebas de funcionalidad.
4	Realizar la documentación y manuales necesarios del módulo.

Tabla 4. Actividades OE4

- OE5. Desarrollar un módulo para el control y manejo de Grupos ASU y TEPES, creación, asignación de tutores, periodos académicos, entrega de certificados y comunicación mediante email.

No.	Actividad
1	Revisión del requerimiento funcional del módulo para gestión de grupos ASU y TEPES
2	Desarrollo del módulo para gestión de grupos ASU y TEPES
3	Pruebas de funcionalidad.
4	Realizar la documentación y manuales necesarios del módulo.

Tabla 5. Actividades OE5

Roles del Proyecto

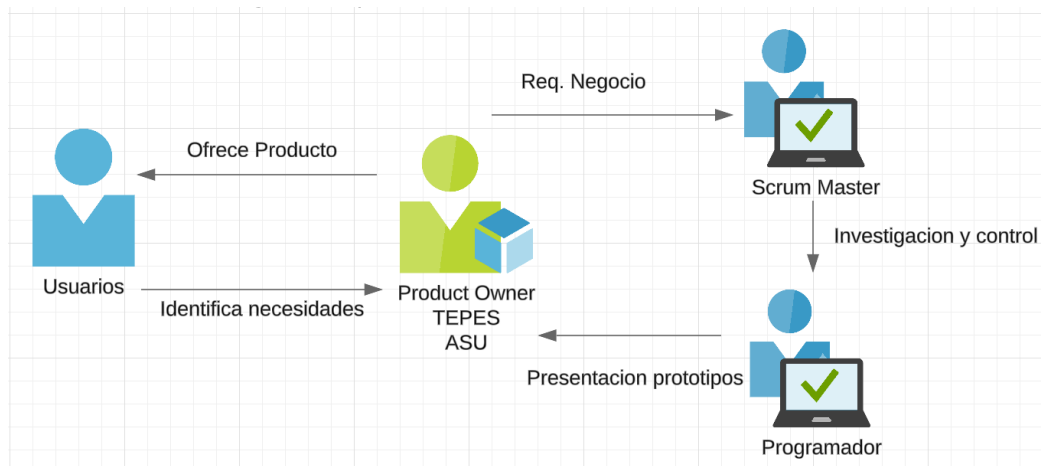


Fig.4 Roles del Proyecto

Product backlog del proyecto ASU y TEPES

Cada ítem del backlog lo detallamos en forma de historias de usuario.

1. Módulo ASU (G. Asociacionismo Salesiano Universitario)

- **Historia de Usuario: Crear Grupos ASU**
 - Descripción: Como administrador, quiero poder crear nuevos grupos de ASU con información como nombre, área de acción, campus y período académico para que los estudiantes puedan inscribirse.
 - Prioridad: Alta
- **Historia de Usuario: Gestión de Animadores, Coordinadores y Participantes**
 - Descripción: Como administrador, quiero gestionar los animadores (docentes), coordinadores y participantes de cada grupo de ASU para garantizar la correcta asignación de roles.
 - Prioridad: Alta
- **Historia de Usuario: Registro de Estudiantes en Grupos ASU**
 - Descripción: Como estudiante, quiero poder registrarme en los grupos de ASU a través de un formulario en el frontend para poder participar en las actividades.
 - Prioridad: Alta
- **Historia de Usuario: Historial de Participación en Grupos ASU**

- Descripción: Como administrador, quiero mantener un historial de los ciclos en los que el estudiante participó en los grupos de ASU para un mejor seguimiento.
- Prioridad: Media
- **Historia de Usuario: Gestión de Eventos ASU**
 - Descripción: Como administrador, quiero crear y gestionar eventos relacionados con los grupos de ASU para facilitar la planificación y participación de los estudiantes.
 - Prioridad: Alta
- **Historia de Usuario: Registro de Estudiantes en Eventos ASU**
 - Descripción: Como estudiante, quiero poder registrarme en los eventos ASU a través de un formulario en el frontend para poder participar en las actividades.
 - Prioridad: Alta

2. Módulo TEPES (Tutorías entre Pares al Estilo Salesiano)

- **Historia de Usuario: Gestión de Tutores y Estudiantes**
 - Descripción: Como administrador, quiero gestionar la asignación de tutores y estudiantes para facilitar el proceso de tutorías.
 - Prioridad: Alta
- **Historia de Usuario: Solicitud de Tutorías por Estudiantes**
 - Descripción: Como estudiante, quiero poder solicitar tutorías a través de un formulario en el frontend para obtener apoyo académico.
 - Prioridad: Alta
- **Historia de Usuario: Asignación de Tutorías por Tutores**
 - Descripción: Como tutor, quiero poder asignarme tutorías para organizar mi carga de trabajo y apoyar a los estudiantes.
 - Prioridad: Alta
- **Historia de Usuario: Registro de Tutorías Realizadas**

- Descripción: Como tutor, quiero registrar las tutorías realizadas y los tiempos en que se realizaron para llevar un control de mis sesiones.
- Prioridad: Media
- **Historia de Usuario: Historial de Tutorías**
 - Descripción: Como administrador, quiero llevar un historial detallado de todas las tutorías realizadas para un mejor seguimiento del rendimiento académico de los estudiantes.
 - Prioridad: Media

3. Comunicación y Notificaciones

- **Historia de Usuario: Notificaciones por Correo Electrónico**
 - Descripción: Como administrador, quiero enviar notificaciones por correo electrónico a estudiantes, tutores, animadores y coordinadores para mantenerlos informados sobre eventos y tutorías.
 - Prioridad: Media
- **Historia de Usuario: Notificaciones por WhatsApp**
 - Descripción: Como administrador, quiero enviar notificaciones automáticas por WhatsApp en cada etapa del proceso de registro, asignación y realización de tutorías y eventos.
 - Prioridad: Media

4. Gestión Administrativa

- **Historia de Usuario: Configuración de Áreas de Acción y Campus**
 - Descripción: Como administrador, quiero configurar las áreas de acción, campus, sede y períodos académicos para cada grupo de ASU y para las tutorías TEPES.
 - Prioridad: Media
- **Historia de Usuario: Exportación de Datos de Participación**

- Descripción: Como administrador, quiero poder exportar informes de participación en grupos ASU y tutorías TEPES en formato Excel o PDF para facilitar la administración y evaluación.
- Prioridad: Baja

5. Mantenimiento y mejoras

- **Historia de Usuario: Pruebas Automáticas de Funcionalidades**
 - Descripción: Como desarrollador, quiero implementar pruebas automáticas de las funcionalidades clave del sistema para asegurar la calidad del código.
 - Prioridad: Alta
- **Historia de Usuario: Escalabilidad y Optimización de la Base de Datos**
 - Descripción: Como administrador del sistema, quiero optimizar la base de datos de Odoo (PostgreSQL) para asegurar que el sistema pueda manejar un gran número de usuarios y transacciones sin perder rendimiento.
 - Prioridad: Alta

Resultados

Sprint 1: Configuración inicial y base del sistema

Objetivo: Establecer la infraestructura básica del proyecto ASU y TEPES, implementando las bases de datos y configuraciones principales en Odoo.

- **Configuración de Odoo y PostgreSQL**
 - Instalar y configurar Docker Odoo en el servidor con PostgreSQL como base de datos en el servidor Ubuntu en la infraestructura de CEDIA otorgado por la Universidad Politécnica Salesiana, el departamento de Infraestructura habilito la VPN de conexión segura al servidor para realizar la instalación:
 - Docker y Docker Compose
 - Portainer Gestor de Contenedores
 - Contenedor de Odoo, Postgres y pgAdmin

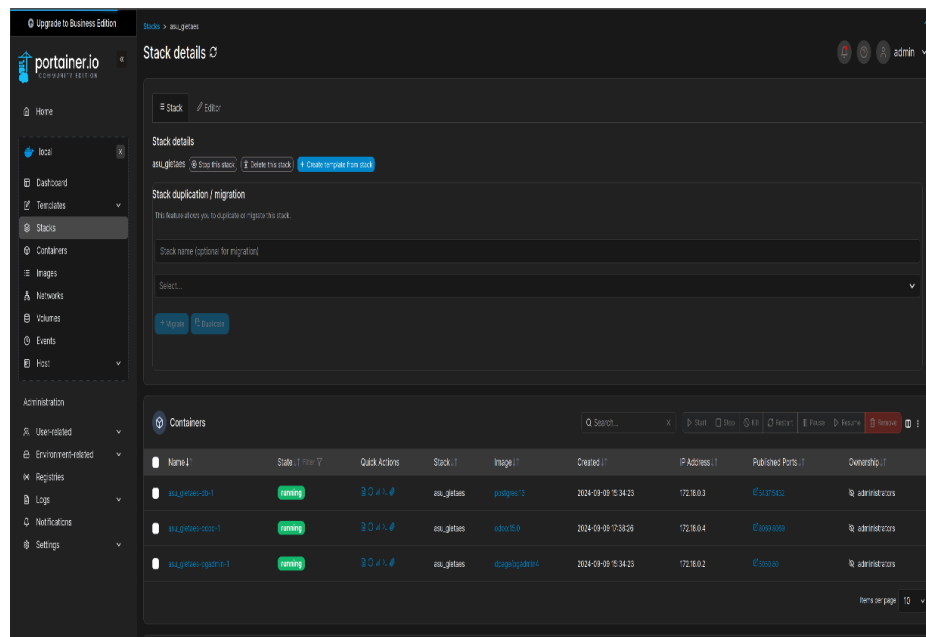


Fig.5

Gestor de Contenedores Portainer

- Configuración de repositorios y control de versiones (Git).

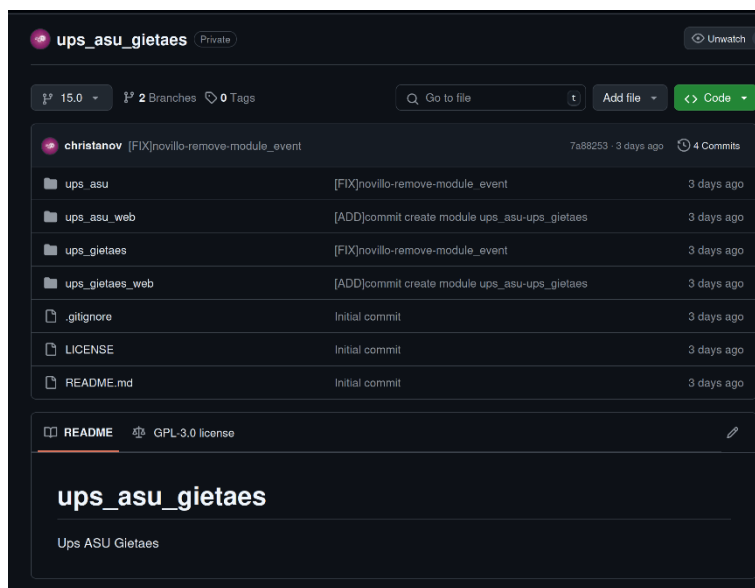


Fig.6 Control de versiones GitHub

- **Creación de la Estructura de Módulos ASU y TEPES**
 - Crear la estructura base de los módulos ASU y TEPES en Odoo.
 - Definir los modelos de datos iniciales para Grupos, Eventos, Tutores y Estudiantes.

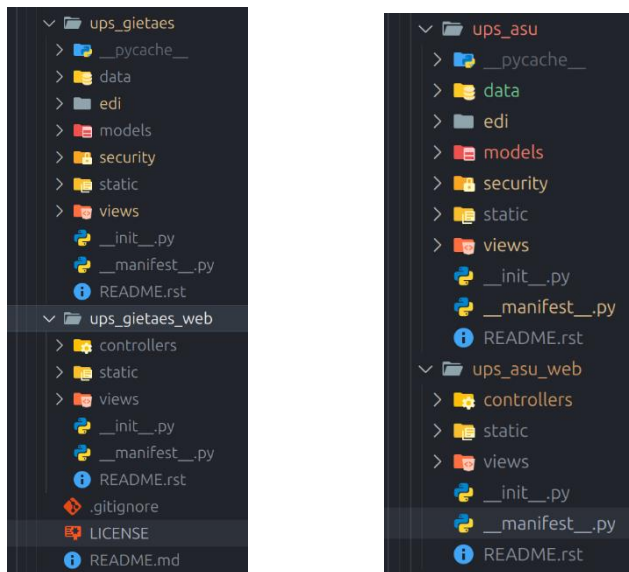


Fig.7 Módulos ASU y TEPES

- **Configuración de Áreas de Acción, Campus y Sedes**
 - Permitir al administrador configurar áreas de acción, campus y sedes.

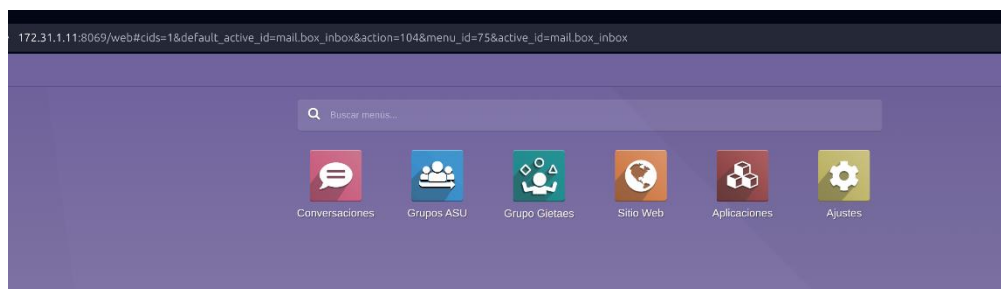


Fig.8 Escritorio del Sistema Odoo

Entregables:

- Estructura básica del sistema lista. (ANEXO1: Manual Proceso de Instalación de Docker Odoo)
- Configuración inicial de los módulos ASU y TEPES. (ANEXO2: Manual de Usuario)

Sprint 2: Gestión de Grupos y Eventos ASU

Objetivo: Implementar la funcionalidad de gestión de grupos y eventos para el módulo ASU.

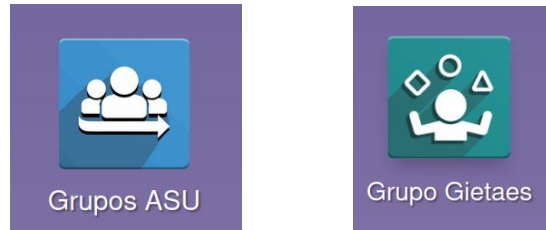


Fig.9 Iconos de Módulos ASU y TEPES

- **Crear Grupos ASU**

- Permitir al administrador crear grupos de ASU con información como nombre, área de acción, campus, sede y período académico.

Fig.10 Formulario creación de Grupo

- **Gestión de Animadores, Coordinadores y Participantes**

- Implementar la gestión de perfiles de animadores (docentes), coordinadores y participantes.

Grupos ASU				
Tutores				
Cédula	Nombre	Correo UPS	Correo Personal	Celular
☐ 1234567891	Juan Perez	juan@up.edu.ec	juanp@gmail.com	099800832
☐ 1234567892	Ana Garcia	ana@up.edu.ec	anag@gmail.com	099800833
☐ 1234567893	Luis Martinez	luis@up.edu.ec	luism@gmail.com	099800834
☐ 1234567894	Marta Lopez	marta@up.edu.ec	martal@gmail.com	099800835
☐ 1234567895	Jose Gonzalez	jose@up.edu.ec	joseg@gmail.com	099800836
☐ 1234567896	Carmen Perez	carmen@up.edu.ec	carmenp@gmail.com	099800837
☐ 1234567897	Miguel Torres	miguel@up.edu.ec	miguelt@gmail.com	099800838
☐ 1234567898	Laura Romero	laura@up.edu.ec	laurar@gmail.com	099800839

Grupos ASU				
Estudiantes				
Cédula	Nombre	Correo UPS	Correo Personal	Celular
☐ 1095294567	Nicola Novillo	nov@up.edu.ec	nov@gmail.com	099404156
☐ 1103709198	Wilmer Novillo	wn@up.edu.ec	wn@gmail.com	099404156
☐ 1254021330	Garcia Pedro	crovilla@est.up.edu.ec		
☐ 1124021331	Martinez Manuel	crovilla@est.up.edu.ec	prueba@gmail.com	099999999
☐ 5023434899	Fernandez Maria	crovilla@est.up.edu.ec	prueba2@gmail.com	099800832
☐ 519041630301	Nicola prueba	prueba3@up.edu.ec	prueba3@gmail.com	099800832
☐ 303498343	Gonzalez Domingo	crovilla@est.up.edu.ec		

Fig.11 Formularios Coordinadores, Animadores y Participantes

- **Registro de Estudiantes en Grupos ASU**
 - Desarrollar un formulario en el frontend para que los estudiantes se inscriban en los grupos de ASU.

SITIO WEB		Grupos ASU		Eventos ASU		CIETAES		Contactos	
Formulario Inscripción Grupo ASU									
Búsqueda Estudiante									
Cédula Consultar									

SITIO WEB		Grupos ASU		Eventos ASU		CIETAES		Contactos	
Formulario Inscripción Grupo ASU									
Información Personal									
Cédula 5023434899					Correo Electrónico UPS crovilla@est.up.edu.ec				
Apellidos y Nombres Fernandez Maria					Grupo ASU Club Automotriz				
Correo Electrónico Personal prueba24@gmail.com					Tipo Sangre O RH+				
Teléfono Celular 099800832									
Información Complementaria									
Nombre Contacto Emergencia Wilmer Novillo					Teléfono Celular Contacto 099404156				
¿El interesado tiene alergias que requieran atención? Si					Describe las alergias polvo				
¿El interesado tiene alguna condición médica que requieran atención? Si					Describe la condición médica taquicardia				
¿Tiene un seguro privado para el grupo familiar? Si					Escriba el Nombre del Seguro ISS				
¿El interesado tiene algún tipo de trastorno específico de aprendizaje? Si					Describe el trastorno				
¿El interesado tiene alguna discapacidad? Si					Describe la discapacidad				
¿Tiene carne de discapacidad? Si									
Guardar									

Fig.12 Formulario Inscripcion Grupo ASU

Fig.13 Formulario Registro Grupo ASU

- **Gestión de Eventos ASU**
 - Permitir la creación y gestión de eventos para los grupos de ASU.

Fig.14 Formulario Registro Evento ASU

Entregables:

- Módulo ASU funcional con creación y gestión de grupos.
- Registro de estudiantes en grupos.

Sprint 3: Gestión de tutorías en TEPES

Objetivo: Implementar las funcionalidades de gestión de tutorías en el módulo TEPES.

- **Gestión de Tutores y Estudiantes**
 - Permitir al administrador asignar tutores y estudiantes en el sistema.

Grupo Galeas

Inscripciones

Cuentas

Comentarios

Configuración

Administrador

Usuarios

Inicio

Estudiantes

+ Nuevo

Filtros

Agregar un filtro

Favoritos

1-17-17

Código	Nombre	Carrera UPS	Carrera Personal	Ocupado	Sede UPS	ID Carrera	Número Carrera	Nivel	ID Materia	Número Materia	Tutor
1001607709	José Luis Alvaréz	jpalvarez@ups.edu.ec	jpalvarez@gmail.com	0006630406		181	INGENIERIA CIVIL	0	0		
2004890543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400		181	INGENIERIA CIVIL	0	29.937	MECANICA NEWTONIANA	
2004380543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400		181	INGENIERIA CIVIL	0	35.572	TOPOGRAFIA II	
2004890543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400		181	INGENIERIA CIVIL	0	35.472	TOPOGRAFIA I	
2004890543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400	MATRIZ OLINCA	181	PRESUPUESTO CIVIL	0	35.472	TOPOGRAFIA I	
2004380543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400	MATRIZ OLINCA	181	INGENIERIA CIVIL	0	35.572	TOPOGRAFIA II	
2004890543	González Domercq	ggonzalez@ups.edu.ec	ggonzalez@gmail.com	0006630400	MATRIZ OLINCA	181	INGENIERIA CIVIL	0	35.472	TOPOGRAFIA I	

Fig.15 Formulario Registro Tutores y Estudiantes

- **Solicitud de Tutorías por Estudiantes**
 - Desarrollar un formulario en el frontend para que los estudiantes soliciten tutorías.

SITIO WEB

Inicio

Personalizar Web

Profilarme

UNIVERSIDAD SALESIANA

30

Inicio

Grupos ASU

Eventos ASU

GIETAES

Contáctenos

Administrador

Contáctenos

Formulario Inscripción Tutorías GIETAES

Información Personal

Cédula

3031988343

Apellidos y Nombres

González Doménica prueba

Correo Electrónico UPS

movillo@est.upi.edu.ec

Correo Electrónico Personal

prueba@gmail.com

Teléfono Celular

0995-101010

Sede

MATRIZ CUENCA

Carrera

INGENIERÍA CIVIL

Nivel

3

Materia

MECÁNICA NEWTONIANA

Formulario Enviado Exitosamente

Fig.16 Formulario Solicitud e Inscripción de Tutorías

- **Asignación de Tutorías por Tutores**
 - Implementar la funcionalidad para que los tutores se asignen tutorías disponibles.

Fig.17 Formulario Asignación de Tutores

- **Registro de Tutorías Realizadas**
 - Desarrollar la funcionalidad para que los tutores registren las tutorías realizadas, incluyendo la duración.

Nro. Registro	Nombre Estudiante	Sede	Carrera	Nivel	Materia	Tutor	Fecha Tutoría	Duración	Comentarios	Aprobar
TR-00003	González Doménica prueba	MATRIZ CUENCA	INGENIERÍA CIVIL	3	MECÁNICA NEWTONIANA	González Doménica prueba	dd / mm / aaaa			☐
TR-00018	González Doménica	MATRIZ CUENCA	INGENIERÍA CIVIL	3	EOLACIONES DIFERENCIALES	González Doménica	dd / mm / aaaa			☐

Fig.18 Formulario Confirmación de tutoría.

Entregables:

- Módulo TEPES funcional con solicitud y asignación de tutorías.
- Gestión de tutores y estudiantes.

Sprint 4: Notificaciones y comunicaciones

Objetivo: Implementar las notificaciones automáticas por correo electrónico y WhatsApp para mantener informados a los estudiantes, tutores, animadores y coordinadores.

- **Notificaciones por Correo Electrónico**

- Configurar el envío de correos electrónicos en cada etapa del proceso, tanto en ASU como en TEPES.

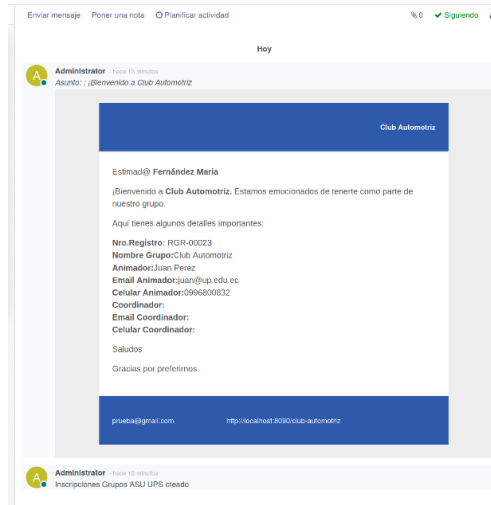


Fig.19 Formulario Notificación mail

- **Notificaciones por WhatsApp**

- Desarrollar la funcionalidad de envío de mensajes de WhatsApp para notificaciones en cada paso de registro, asignación y realización de eventos y tutorías.

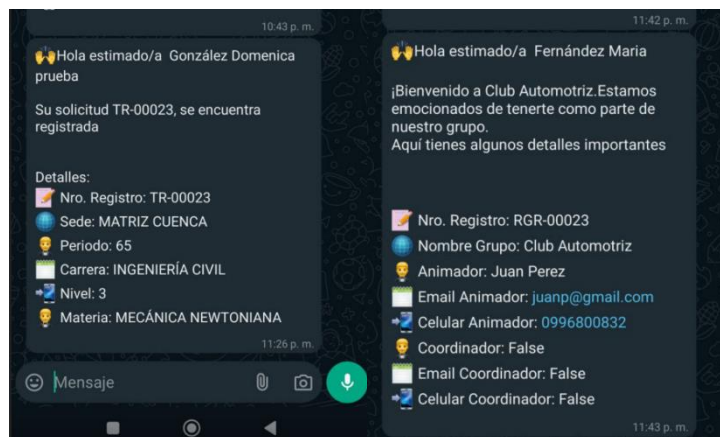


Fig.20 Formulario Notificación Whatsapp

Entregables:

- Sistema de notificaciones automatizado para correo y WhatsApp.

Sprint 5: Pruebas y optimización

Objetivo: Realizar pruebas de calidad, optimizar el sistema y asegurar que la base de datos y las funcionalidades puedan escalar según las necesidades.

- **Pruebas Automáticas de Funcionalidades**
 - Implementar pruebas automáticas para las funcionalidades críticas del sistema (gestión de grupos, eventos, tutorías, etc.).
- **Optimización de la Base de Datos PostgreSQL**
 - Realizar ajustes y optimización en la base de datos para mejorar el rendimiento con grandes volúmenes de datos.

Entregables:

- Sistema probado y optimizado para la escalabilidad.
- Optimización de la base de datos.

Sprint 6: Finalización y documentación

Objetivo: Documentar el sistema, preparar los manuales de usuario y realizar ajustes finales antes del despliegue.

- **Documentación Técnica**
 - Crear la documentación técnica del sistema, incluyendo el diseño de la arquitectura y las guías de instalación y mantenimiento.
- **Manual de Usuario**
 - Desarrollar el manual de usuario para los administradores, tutores y estudiantes.
- **Despliegue Final**
 - Desplegar el sistema en producción y realizar las últimas pruebas de aceptación por parte del usuario.

Entregables:

- Documentación técnica ANEXO1 y manual de usuario ANEXO2.
- Sistema desplegado y funcionando en producción.

Cronograma

Objetivo	Actividad	Fecha Inicio	Fecha Fin	Días	Horas
OE1. Estudiar y analizar los procesos del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).	1.Análisis de los procesos del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES)	15/05/24	20/05/24	5	20
	2.Entrevistas con los miembros del Grupo ASU y Tutorías Entre Pares al Estilo Salesiano (TEPES) para obtener información del estado actual de los procesos.	21/05/24	25/05/24	5	20
	3.Proceso de validación de información.	27/05/24	30/05/24	4	16
	4.Realizar una documentación de la información relevante para la realización del sistema.	31/05/24	05/06/24	5	20
OE2. Definir e implementar los requerimientos funcionales para el sistema de gestión del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).	1.Toma de requerimientos funcionales mediante entrevistas con los miembros de los Grupos ASU y TEPES.	06/06/24	12/06/24	6	24
	2.Diseño para el desarrollo de los módulos aprobados, para posteriormente asignar una fecha y el responsable.	13/06/24	19/06/24	8	32
	3.Validación y aprobación de los módulos a desarrollar.	20/06/24	25/06/24	5	20
OE3. Diseñar y desarrollar una plataforma web que permita la gestión y monitoreo del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).	1.Análisis requerimientos funcionales mediante entrevistas con los miembros de los Grupos ASU y TEPES.	26/06/24	02/07/24	6	24
	2.Diseño para el desarrollo de los módulos aprobados, para posteriormente asignar una fecha y el responsable.	03/07/24	09/07/24	6	24
	3.Validación y aprobación de los diseños.	10/07/24	15/07/24	5	20

OE4. Desarrollar un módulo para la consulta y registro de estudiantes del Asociacionismo Salesiano Universitario (ASU) y Tutorías Entre Pares al Estilo Salesiano (TEPES).	1. Revisión del requerimiento funcional de registro de estudiantes para los Grupos ASU y TEPES	16/07/24	20/07/24	5	20
	2.Desarrollo del módulo de registro de estudiantes	22/07/24	29/07/24	7	28
	3.Pruebas de funcionalidad.	30/07/24	05/08/24	6	24
	4.Realizar la documentación y manuales necesarios del módulo.	06/08/24	09/08/24	4	16
OE5. Desarrollar un módulo para el control y manejo de Grupos ASU y TEPES, creación, asignación de tutores, periodos académicos, entrega de certificados y comunicación mediante email.	1.Revisión del requerimiento funcional del módulo para gestión de grupos ASU y TEPES	12/08/24	16/08/24	5	20
	2.Desarrollo del módulo para gestión de grupos ASU y TEPES	17/08/24	24/08/24	7	28
	3.Pruebas de funcionalidad.	26/08/24	31/08/24	6	24
	4.Realizar la documentación y manuales necesarios del módulo.	02/09/24	06/09/24	5	20

Total, de Horas por estudiante: 400.

Fecha de Inicio: 15/05/2024

Fecha de Finalización: 06/09/2024

Presupuesto

DENOMINACIÓN	CANT.	COSTO UNITARIO	COSTO TOTAL
	Unidades	Dólares	dólares
1. Bienes			
Papel Bond A-4	3	5,00	15,00
Copias	100	0,05	5,00
2. Tecnológico			
Computadora portátil	2	900,00	1800,00
Celular inteligente	1	500,00	500,00
2. Servicios			
Servidor	3	40	120,00
Servicio de transporte	1	50,00	50,00
Servicios de Internet	3	30,00	90,00
Material Bibliográfico	2	40,00	80,00
4. Personal			
Estudiante investigador	2	1800,00	3600,00
Asesoría especializada	1	800,00	800,00
3. Otros			
Imprevistos	1	100,00	100,00
TOTAL			7160,00

Conclusiones.

El desarrollo del sistema ASU y TEPES, integrado en la plataforma Odoo, ha permitido automatizar de manera eficiente los procesos de gestión de los Grupos de Asociacionismo Salesiano Universitario (ASU) y las Tutorías entre Pares al Estilo Salesiano (TEPES) en la Universidad Politécnica Salesiana. A través de este proyecto, se han logrado los siguientes resultados clave:

Automatización de Procesos Administrativos mediante la reducción significativa la carga administrativa al automatizar tareas que previamente eran manuales, como la creación de grupos, la asignación de tutores, la inscripción de estudiantes y la generación de reportes.

Mejora de la Experiencia del Usuario mediante la implementación de un frontend intuitivo ha facilitado el registro de los estudiantes en grupos y eventos de ASU, así como en las tutorías de TEPES. Al contar con formularios claros y accesibles, se ha mejorado la experiencia de los estudiantes, que ahora pueden realizar sus solicitudes y seguimientos de manera más eficiente.

La Escalabilidad y Flexibilidad del Sistema Odoo ha demostrado ser una solución adecuada debido a su capacidad de adaptación y escalabilidad. El uso de una arquitectura modular ha permitido la personalización de los módulos ASU y TEPES para ajustarse a las necesidades específicas de la institución, con la posibilidad de expandir o modificar funcionalidades en el futuro.

Transparencia y Seguimiento de historiales detallados de participación en grupos ASU y tutorías TEPES proporciona un registro claro y confiable de la actividad de los estudiantes. Esta información es valiosa tanto para los tutores como para los administradores al momento de evaluar el compromiso y el progreso de los estudiantes.

La Integración de Comunicaciones Eficientes mediante notificaciones automáticas de correos electrónicos y WhatsApp en cada etapa del proceso ha mejora la comunicación entre los estudiantes, tutores, animadores y coordinadores. Esto asegura que todas las partes involucradas estén informadas de los eventos y tutorías en tiempo real.

El uso de Metodologías Ágiles como Scrum permitió un desarrollo iterativo y colaborativo, lo que facilitó la adaptación a los cambios y la priorización de las necesidades críticas a lo largo del proyecto. Este enfoque permitió la entrega de un producto funcional desde etapas tempranas del desarrollo, con mejoras continuas.

En conclusión, el proyecto ASU y TEPES proporciona una solución tecnológica robusta que optimiza la gestión. El uso de Odoo como plataforma ERP ha permitido la creación de una herramienta eficiente, escalable y adaptable que mejora tanto la administración como la experiencia de los usuarios involucrados, sentando las bases para futuros desarrollos.

Recomendaciones.

Integración de un Módulo de Reportes Avanzados que permita generar estadísticas más detalladas y personalizadas sobre la participación de los estudiantes en los Grupos ASU y las Tutorías TEPES. Esto podría incluir indicadores clave como asistencia, rendimiento, impacto de las tutorías en el desempeño académico, y participación en eventos. Estos informes facilitarían la toma de decisiones para los administradores y permitirían un análisis profundo de los beneficios de estos programas.

Implementación de Inteligencia Artificial para la Asignación de Tutores en el módulo TEPES. Un sistema de recomendación basado en IA podría analizar las preferencias de los estudiantes, la disponibilidad de los tutores y el desempeño en tutorías anteriores para hacer asignaciones automáticas, optimizando así la carga de trabajo de los tutores y mejorando la experiencia del estudiante.

Sistema de Evaluación y Retroalimentación para Tutores y Estudiantes en donde se pueda calificar y dejar comentarios sobre los tutores, y viceversa. Este sistema de retroalimentación permitiría mejorar la calidad de las tutorías y la interacción entre estudiantes y tutores, fomentando una cultura de mejora continua.

Automatización de Certificaciones y Reconocimientos con un módulo que automatice la emisión de certificados para los estudiantes que hayan cumplido con un número mínimo de participaciones en grupos y eventos ASU o que hayan asistido a un determinado número de tutorías TEPES. Esto no solo motivaría a los estudiantes, sino que también facilitaría la acreditación y el reconocimiento formal de su participación.

Integración de Herramientas de Videoconferencia dado que las tutorías o eventos ASU pueden llevarse a cabo de forma remota, sería beneficioso integrar plataformas de videoconferencia como Odoo Meet, Zoom, o Google Meet. Esto permitiría la realización de tutorías en línea directamente desde la plataforma, ofreciendo mayor flexibilidad para estudiantes y tutores.

Referencias Bibliográficas.

- Cárdenas-Tapia, J., Pesántez-Avilés, F., Torres-Toukourmidis, Á., Herrera Cando, D., Narváez-Encalada, L. G., & Álvarez-Castro, R. (2023). Grupos Culturales de Asociacionismo Salesiano Universitario. Historia, eventos y reflexiones.
- CÁRDENAS, J. (2023). INFORME DE RENDICIÓN DE CUENTAS. Obtenido de <https://www.ups.edu.ec/documents/20121/262141/2023+Informe+del+Rector.pdf>
- Jara, V. O. (2018). El plan de acompañamiento tutorial para la tutoría entre pares al estilo salesiano: propuesta para su generalización en la UPSEcuador. (U. L. HABANA-CUBA, Editor) Obtenido de <https://www.ups.edu.ec/documents/20121/259210/PAT+-+TEPeS+UPS.pdf>

- UPS. (7 de mayo de 2024). Grupos ASU. Obtenido de <https://www.ups.edu.ec/grupos-asu>
- Geschwinde, E., & Schönig, H. J. (2002). PostgreSQL developer's handbook. Sams Publishing.
- Oleksiuk, V. P., Verbovetskyi, D. V., & Hrytsai, I. A. (2024). Design and development of a game application for learning Python. In CEUR Workshop Proceedings (pp. 111-124).
- Ajata, M. A. A. (2023). DESARROLLO ÁGIL DE SOFTWARE CON SCRUM. In MEMORIAS DEL CONGRESO NACIONAL DE INFORMÁTICA (No. 1, pp. 51-55).
- Reis, D. (2018). Odoo 12 Development Essentials: Fast-track your Odoo development skills to build powerful business applications. Packt Publishing Ltd.
- The PostgreSQL Global Development Group (2024) Documentación de PostgreSQL: <https://www.postgresql.org/>
- Odoo Inc.(2024) Documentación oficial de Odoo: <https://www.odoo.com/>