



FACULTAD DE INFORMATICA Y CIENCIAS APLICADAS

ESCUELA DE INFORMÁTICA

Asignatura:

Programación II

Docente:

Alfredo Omar

Ciclo:

02-2024.

Sección:

03

Rodríguez Torres

Periodo 4

Tema:

Documentación

Fecha de entrega:

17/11/2024

Proyecto AcademiEnroll

Fase IV

Integrantes

Durán Cabezas Diego Vladimir	25-1148-2023
Durán Cabezas Erick Manuel	25-1149-2023
Hernández Arévalo Osaki Vladimir	25-2627-2023
Hernández Molina Diego Enrique	25-2862-2023
López García José Adonay	25-1647-2023
Martínez Salmerón Gerardo David	25-2769-2023
Mejía Martinez Jonathan Alexander	25-3190-2023

Índice

Introducción	4
Capitulo I.....	5
1.1 Antecedentes	5
1.2 Importancia del proyecto	5
1.2.1 Planteamiento del Problema	7
1.2.2 Definición del aplicativo.....	9
1.2.3 Justificación del Aplicativo.....	10
1.3 Objetivos.....	11
1.3.1 General.....	11
1.3.2 Específicos	11
1.4 Alcances.....	11
1.5 Limitaciones.....	12
1.6 Factibilidad del proyecto.....	12
1.6.1 Factibilidad Operativa:.....	13
1.6.2 Factibilidad Económica:	14
1.7 Cronograma de actividades.....	16
Capitulo II.....	25
2.1 Técnicas y herramientas de investigación.....	25
2.3 Análisis de Resultados	29
Capitulo III.....	32
3.1 Diseño del Sistema.....	32
3.2 Mapa del sistema o diagrama grafico modular	42

3.3	Descripción de los distintos módulos utilizados para la construcción del sistema	43
3.4	Descripción de las Distintas Funciones Utilizadas:	48
3.5	Base de datos.....	51
3.6	Seguridad Del Sistema.....	55
	Referencias.....	59
	Glosario.....	61
	Anexos	66

Introducción

Las instituciones de educación superior deben adoptar tecnologías modernas para mejorar la eficiencia operativa y la gestión académica, ya que los sistemas de inscripción tradicionales presentan problemas como inestabilidad y errores frecuentes, lo que afecta la experiencia de estudiantes y personal y puede perjudicar la reputación institucional.

El proyecto "AcademiEnroll" propone una solución innovadora para modernizar el proceso de inscripción, utilizando tecnologías avanzadas como ASP.NET Core y Razor Pages. Este sistema busca gestionar un mayor volumen de solicitudes de manera eficiente, reducir errores humanos, mejorar la experiencia del usuario y optimizar recursos internos.

El documento detalla el desarrollo de un proyecto que analiza los problemas del sistema de inscripción actual, resaltando la obsolescencia de las tecnologías utilizadas y su incapacidad para atender la creciente demanda de estudiantes. Esto justifica la necesidad de un nuevo sistema basado en tecnologías robustas, escalables y seguras, para mejorar la gestión de cupos, la validación automática de prerrequisitos y la protección de datos académicos.

Se establecen objetivos generales y específicos, como crear un sistema de inscripción en línea más rápido, preciso y seguro, que incluya la gestión de cupos, la generación de reportes automáticos y la validación de requisitos previos. El objetivo final es transformar la experiencia de inscripción en un proceso más eficiente y satisfactorio para estudiantes y personal administrativo. La metodología del proyecto detalla las técnicas y herramientas utilizadas para desarrollar el sistema, incluyendo revisiones documentales, pruebas de concepto, prototipos y una investigación técnica para asegurar que el sistema sea escalable y seguro. También se analizan las herramientas empleadas, como Visual Studio, Razor Pages, C#, SQL Server y otras tecnologías, que han facilitado la creación de un sistema académico moderno y eficiente.

El análisis de resultados muestra avances significativos, como la automatización del proceso de inscripción y la integración efectiva de la base de datos, lo que ha mejorado la precisión y velocidad del sistema. Sin embargo, el documento también indica áreas que necesitan más trabajo, como la implementación completa de reportes automáticos para administradores y la optimización del rendimiento del sistema ante alta demanda.

Capítulo I

Marco Referencial

1.1 Antecedentes

En el marco de la educación superior, la digitalización de procesos académicos se ha convertido en una prioridad para instituciones que buscan mejorar la experiencia de estudiantes, docentes y administradores. Los sistemas de gestión académica han surgido como herramientas esenciales para centralizar funciones como la inscripción de materias, el seguimiento de calificaciones y la asignación de docentes a cursos, facilitando la operación eficiente de estas instituciones.

Un ejemplo destacado es **Moodle**, una plataforma de aprendizaje de código abierto ampliamente utilizada por universidades e institutos educativos. Moodle permite a los docentes crear aulas virtuales, gestionar recursos y evaluar a los estudiantes a través de actividades interactivas. Su modelo de software libre lo convierte en una herramienta accesible y adaptable a las necesidades específicas de cada institución, lo que ha contribuido a su implementación en universidades como la UNAM y el Instituto Politécnico Nacional. La flexibilidad de Moodle radica en su capacidad para soportar diferentes modelos educativos y proporcionar soluciones escalables, desde cursos individuales hasta programas completos de educación en línea

1.2 Importancia del proyecto

El proyecto "AcademiEnroll" es de vital importancia debido a varios factores que afectan directamente tanto a la operatividad interna de la institución como a la experiencia de los estudiantes. En el contexto actual, las instituciones de educación superior enfrentan una creciente demanda tecnológica, especialmente en áreas administrativas como la inscripción de estudiantes, la gestión de horarios y el registro académico. Este proyecto tiene el objetivo principal de abordar los retos tecnológicos que han surgido en la institución debido al aumento del número de matriculados y la obsolescencia del sistema actual.

El sistema de inscripción existente, desarrollado hace varios años en ASP clásico, ha quedado desfasado frente a las exigencias modernas. A medida que la academia creció en cantidad de alumnos, los servidores comenzaron a sufrir constantes caídas, afectando la disponibilidad del servicio y generando frustración tanto en estudiantes como en personal administrativo. Un sistema que no responde a las necesidades actuales representa no solo un

obstáculo operativo, sino también una amenaza para la reputación institucional y su capacidad de atraer nuevos estudiantes. Por lo tanto, el desarrollo del sistema "AcademiEnroll", utilizando la tecnología ASP.NET Core, es crucial para proporcionar una solución escalable, eficiente y adaptada a las demandas tecnológicas del presente y futuro.

La importancia de este proyecto radica en varios puntos clave:

Escalabilidad y modernización tecnológica: Al migrar a un framework más avanzado como ASP.NET Core, la institución podrá gestionar un mayor volumen de peticiones simultáneas sin experimentar los problemas de caída del servidor que ocurren con el sistema antiguo. Esto no solo mejorará la capacidad de respuesta del sistema en picos de demanda (como los periodos de inscripción), sino que también garantizará que el sistema pueda seguir creciendo junto con la institución.

Mejora en la experiencia del usuario: Uno de los objetivos centrales del proyecto es mejorar la interfaz de usuario (UI) y la experiencia general de navegación. El proceso de inscripción es una de las interacciones más críticas entre los estudiantes y la institución, y debe ser lo más fluido y rápido posible. Con una UI más intuitiva, los estudiantes podrán completar sus inscripciones en menor tiempo, reduciendo errores y mejorando la percepción de la institución como una entidad eficiente y moderna.

Automatización de procesos administrativos: Actualmente, los procesos de inscripción y gestión de estudiantes requieren una intervención significativa por parte del personal administrativo. La automatización de estos procesos permitirá que el sistema gestione automáticamente la selección de cursos, validación de prerrequisitos, asignación de cupos y generación de horarios, lo cual reducirá los tiempos de espera y disminuirá el riesgo de errores humanos. Esto liberará recursos dentro de la institución que podrán ser dirigidos a otras tareas más estratégicas.

Mejora en la precisión y seguridad de la información: Con un sistema moderno que centraliza la gestión de datos de estudiantes, se podrá garantizar una mayor precisión en los registros y transacciones. Además, el nuevo sistema implementará niveles de seguridad avanzados, lo que asegurará que cada usuario tenga acceso únicamente a la información y operaciones pertinentes a su rol (administrador, docente o estudiante). Esto no solo protege la

privacidad de los datos, sino que también contribuye a un entorno académico más organizado y seguro.

Mantenimiento y mejora continua: Uno de los grandes problemas con los sistemas heredados es la dificultad para realizar actualizaciones y mantenimiento. Con "AcademiEnroll", se establecerá un plan de mantenimiento regular, lo que permitirá realizar actualizaciones de manera eficiente y mantener el sistema en óptimas condiciones. Esto asegurará la longevidad y adaptabilidad del sistema a medida que las necesidades de la institución cambien.

En resumen, la importancia de este proyecto va más allá de solucionar problemas técnicos inmediatos; se trata de garantizar que la institución pueda seguir creciendo de manera eficiente y moderna, mejorando la experiencia de los estudiantes y optimizando la administración interna. La implementación de este sistema no solo resolverá problemas actuales, sino que sentará las bases para una gestión académica sólida y escalable a largo plazo.

1.2.1 Planteamiento del Problema

El planteamiento del problema de este proyecto se centra en las limitaciones y dificultades que enfrenta la institución "Academi Course" con su actual sistema de inscripción de estudiantes. Estas dificultades han sido cada vez más evidentes a medida que la cantidad de alumnos ha crecido, especialmente en los períodos de mayor actividad como el inicio de semestre y las semanas de inscripción. El sistema actual fue desarrollado hace varios años utilizando ASP clásico, una tecnología que ha quedado obsoleta frente a las necesidades modernas de gestión académica. Este marco tecnológico ha alcanzado su límite en términos de escalabilidad, capacidad y mantenimiento, lo que ha resultado en caídas frecuentes de los servidores, errores en la inscripción y una pobre experiencia para los usuarios.

El problema principal es que el sistema actual no puede manejar la creciente demanda de los estudiantes. Durante los períodos de inscripción, la cantidad de solicitudes simultáneas colapsa los servidores, lo que provoca retrasos significativos en el proceso y en muchos casos, errores de registro. Este tipo de fallas genera una gran frustración tanto en los estudiantes que no pueden completar su inscripción de manera eficiente, como en el personal administrativo que se ve obligado a lidiar con problemas técnicos y quejas constantes. Además, la falta de capacidad del sistema para manejar estos picos de demanda pone en riesgo la reputación de la institución,

ya que los estudiantes y sus familias perciben la ineficiencia del proceso como un reflejo de la calidad general de la academia.

Otro aspecto problemático es que el sistema no permite una adecuada gestión de los cursos y horarios. La asignación de cupos en las clases es manual y a menudo se generan conflictos de horarios que deben ser resueltos por el personal, lo que añade una capa adicional de complejidad y riesgo de errores. Además, los procesos de validación de prerrequisitos para inscribirse en cursos específicos no están bien automatizados, lo que permite que estudiantes se inscriban en cursos sin cumplir con las condiciones necesarias, lo cual afecta tanto el rendimiento académico como la administración del flujo de estudiantes.

Finalmente, la gestión de la información de los estudiantes es otro punto crítico. El sistema actual no ofrece suficiente flexibilidad ni seguridad en la administración de los datos de los estudiantes, lo que dificulta el acceso a información precisa y actualizada para los docentes, administradores y los mismos estudiantes. Esto representa un riesgo tanto desde el punto de vista de la eficiencia operativa como de la seguridad de los datos.

El proyecto "AcademiEnroll" busca solucionar este problema mediante el desarrollo de un sistema informático moderno y robusto que centralice y automatice todos los procesos relacionados con la inscripción y gestión de estudiantes. Este nuevo sistema permitirá:

Manejo eficiente del flujo de estudiantes: Al automatizar la inscripción, la gestión de cupos, la validación de requisitos y la asignación de horarios, se evitarán los cuellos de botella actuales y se asegurará que cada estudiante pueda realizar sus trámites de manera rápida y eficiente.

Optimización de los recursos tecnológicos: La implementación de ASP.NET Core y otros avances tecnológicos permitirá una mayor escalabilidad, lo que significa que el sistema podrá manejar un número mucho mayor de solicitudes simultáneas sin experimentar caídas ni interrupciones.

Automatización de validaciones: Uno de los mayores retos del sistema actual es la validación de los prerrequisitos para los cursos. Con el nuevo sistema, esta validación será automática, lo que reducirá el error humano y asegurará que los estudiantes solo se inscriban en cursos para los que estén realmente preparados.

Centralización y seguridad de la información: La base de datos centralizada permitirá una gestión más precisa de la información de los estudiantes, además de implementar un sistema de permisos por roles, asegurando que cada usuario tenga acceso solo a la información y funciones que le correspondan según su rol (estudiante, docente o administrador).

1.2.2 Definición del aplicativo

"AcademiEnroll" es un sistema informático diseñado específicamente para centralizar, automatizar y optimizar todos los procesos relacionados con la inscripción y la gestión académica de los estudiantes en la institución "Academi Course". Se trata de un aplicativo moderno, desarrollado utilizando tecnologías avanzadas como ASP.NET Core, que permitirá gestionar de manera eficiente el creciente volumen de solicitudes de inscripción, la asignación de cupos, la validación de prerrequisitos, y la administración de horarios de manera automatizada.

El sistema "AcademiEnroll" será capaz de manejar un gran número de solicitudes simultáneas sin experimentar caídas ni interrupciones, gracias a su capacidad de escalabilidad. Esto garantizará una experiencia fluida tanto para los estudiantes como para el personal administrativo, especialmente durante los períodos críticos de inscripción. Además, el sistema contará con una base de datos centralizada que permitirá un acceso rápido y seguro a la información de los estudiantes, evitando problemas como los errores de registro y la duplicación de datos.

Entre las funcionalidades clave del aplicativo, se incluyen:

- **Automatización de la inscripción de estudiantes:** Los estudiantes podrán inscribirse de manera rápida y eficiente, sin la intervención manual de los administradores.
- **Gestión automatizada de cupos y horarios:** La asignación de cupos y la planificación de horarios se realizará de manera automática, minimizando el riesgo de conflictos y errores.
- **Validación automática de prerrequisitos:** Los estudiantes solo podrán inscribirse en cursos para los que cumplan con los requisitos previos, lo que asegurará un flujo académico adecuado y una mejor calidad educativa.

- **Seguridad de la información:** El sistema contará con un robusto esquema de seguridad, que protegerá los datos sensibles de los estudiantes y garantizará que cada usuario tenga acceso únicamente a la información que le corresponda, según su rol (estudiante, docente o administrador).

En definitiva, "AcademiEnroll" es una solución integral que abordará todos los puntos débiles del sistema actual, proporcionando una plataforma tecnológica avanzada que mejorará la eficiencia operativa y la experiencia de los usuarios.

1.2.3 Justificación del Aplicativo

La justificación para el desarrollo e implementación de "AcademiEnroll" se basa en la necesidad crítica de resolver las múltiples limitaciones del sistema actual de inscripción de "Academi Course". La tecnología con la que fue desarrollado el sistema existente ha quedado desfasada, lo que no solo afecta la capacidad operativa de la institución, sino que también repercute directamente en la satisfacción de los estudiantes y en la reputación de la academia.

Uno de los principales motivos que justifican este nuevo sistema es la incapacidad del sistema actual para manejar grandes volúmenes de solicitudes de inscripción. Durante los períodos de mayor demanda, los servidores colapsan y se generan errores que afectan la experiencia del estudiante, causando frustración y retrasos en los procesos administrativos. "AcademiEnroll" solucionará este problema mediante la implementación de una plataforma moderna y escalable que podrá soportar un número mucho mayor de solicitudes simultáneas, eliminando las interrupciones del servicio.

Otro aspecto importante es la automatización de tareas que actualmente se realizan de manera manual. La gestión de cupos, la asignación de horarios y la validación de prerequisites son procesos que, si bien son esenciales, representan una gran carga administrativa cuando no están automatizados. Al automatizar estos procesos, el nuevo sistema reducirá los errores humanos y optimizará los tiempos de inscripción, permitiendo que los estudiantes se inscriban en los cursos adecuados de manera más rápida y eficiente.

La seguridad de la información es otro factor clave en la justificación del proyecto. El sistema actual presenta debilidades en la gestión de datos sensibles de los estudiantes, lo que representa un riesgo significativo desde el punto de vista de la protección de la información.

"AcademiEnroll" implementará un esquema de seguridad robusto y centralizado, que no solo garantizará la protección de los datos, sino que también asegurará que los diferentes usuarios del sistema (estudiantes, docentes, administradores) tengan acceso únicamente a la información relevante para su rol, mejorando así la confidencialidad y la eficiencia operativa.

En resumen, la implementación de "AcademiEnroll" no solo es necesaria para resolver los problemas actuales del sistema de inscripción, sino que también permitirá que "Academi Course" se mantenga competitiva y eficiente en el largo plazo. El nuevo sistema mejorará la experiencia del usuario, reducirá la carga administrativa y garantizará la seguridad de los datos, asegurando que la institución esté preparada para enfrentar los desafíos del futuro.

1.3 Objetivos

1.3.1 General

Desarrollar un sistema web que automatice y gestione el proceso de inscripción en una institución de educación superior, permitiendo a los estudiantes realizar inscripciones en línea de manera más rápida y precisa.

1.3.2 Específicos

1. Implementar un sistema que gestione la inscripción de los estudiantes en las diferentes carreras y materias, asegurando la validación de requisitos previos y cupos disponibles.
2. Integrar una base de datos que almacene la información de estudiantes, materias, y carreras, permitiendo el acceso y la modificación de datos de manera segura y eficiente y precisa
3. Facilitar a los administradores de la institución la capacidad de generar reportes automáticos sobre inscripciones y gestionar las modificaciones necesarias en el sistema.

1.4 Alcances

- El sistema permitirá la inscripción en línea de estudiantes en las carreras que están disponibles en la institución.
- Administrará los cupos por asignatura, verificando que se cumplan los prerrequisitos y verificando la cantidad de estudiantes que seleccionen dicha materia con su dicho horario.

- Generará reportes de inscripción para los administradores de la institución.

1.5 Limitaciones

- Falta de cooperación de los interesados: a veces, los profesores, estudiantes y el personal administrativo no proporcionan la información necesaria a tiempo. Esto puede llevar al retraso del desarrollo, y la calidad podría verse comprometida.
- Cambios en los requisitos académicos: las políticas académicas y las necesidades del sistema pueden cambiar constantemente. Esto puede causar confusión y aumentar los costos del proyecto.
- Limitaciones tecnológicas: los departamentos de tecnología podemos no tener la cantidad adecuada de personas y la infraestructura para desarrollar el proyecto de la manera más óptima.
- Disponibilidad de recursos: falta de recursos técnicos como programadores e infraestructura de apoyo.
- Problemas de comunicación: la falta de comunicación entre los miembros del equipo y los interesados puede traer problemas.
- Restricciones de tiempo: los periodos de implementación pueden ser cortos y este puede ser un desafío.
- Riesgos de seguridad: los sistemas deben ser seguros y para proteger los datos académicos, algunas funciones pueden verse limitadas o requieren más esfuerzo.

Depender de terceros: el uso de bases de datos de alumnos y de sistemas de gestión académica a veces puede no ser fiable.

1.6 Factibilidad del proyecto

1. Estimación de Costos

a) Costos de Desarrollo

- **Herramientas de desarrollo y software:**
 - **Visual Studio:** La versión Community, no tiene ningún costo adicional.
 - **SQL Server:** SQL Server Express está disponible sin costo de licencias.
 - **Bootstrap, HTML, CSS:** Son tecnologías de código abierto y no tienen costos directos.
- **Costo de Personal:**
 - **Desarrollador(es):** Como los programadores son estudiantes, no hay gastos directos.
 - **Administrador de Bases de Datos:** Los estudiantes también trabajarán en la gestión y optimización de una base de datos.
 - **Tester/QA:** La validación de la calidad y funcionalidad del sistema será parte del proyecto académico.

b) Costos de Infraestructura

- **Servidores y Alojamiento:**
 - Si se alojará en un servidor propio o un servicio en la nube (Azure, AWS), el costo dependerá de los recursos necesarios para manejar el tráfico esperado y la base de datos.
 - **Estimación:** Un servidor básico puede costar entre \$20 y \$200 al mes, dependiendo de la capacidad, o más si se requieren servicios avanzados.
- **Mantenimiento:**
 - Actualizaciones, resolución de errores, soporte y mejoras continuas después del lanzamiento.
 - **Estimación de costos anuales:** 10%-20% del costo total de desarrollo.

1.6.1 Factibilidad Operativa:

El proyecto propone la siguiente factibilidad operativa con beneficios como

- **Reducción de Costos Administrativos:** El proceso de inscripción y gestión de notas se ha digitalizado y automatizado, lo que reduce significativamente la carga de trabajo del personal administrativo. Esto podría significar que no es necesario contratar o reasignar personal para tareas más estratégicas.
- **Estimación de Ahorro:** El costo anual de varios empleados administrativos puede reducirse o reasignarse, lo que puede representar ahorros.
- **Reducción de Errores:** La automatización reduce la posibilidad de errores humanos en el registro de notas y la inscripción de materias, lo que evita costos asociados con correcciones y mejora la calidad del servicio.
- **Aumento de la Eficiencia:** Los estudiantes podrán administrar sus inscripciones y ver sus notas en línea, lo que reducirá el tiempo de espera en las oficinas administrativas. Esto puede aumentar la satisfacción del estudiante y la percepción de calidad del servicio.
- **Mejora en la Experiencia del Usuario:** Al proporcionar un sistema moderno y fácil de usar, la institución podría mejorar su reputación y atraer a más estudiantes.
- **Acceso Rápido a la Información:** Los estudiantes y el personal académico tendrán acceso a la información desde cualquier lugar, lo que mejora la organización y la toma de decisiones.

1.6.2 Factibilidad Económica:

1. Costos Estimados del Proyecto

Desarrollo del Sistema

- **Diseño y Programación:**

Desarrollo de funcionalidades como login, validación de roles, CRUD para usuarios, gestión de notas y administración.

Costo estimado: \$15,000 USD (considerando un equipo de desarrollo de 7 personas trabajando durante 6 meses).

- Pruebas y Optimización:
 - ✓ Pruebas de funcionalidad, seguridad y rendimiento.
 - ✓ Costo estimado: \$500 USD.

Infraestructura Tecnológica

- Servidor y Dominio:
 - ✓ Hospedaje web y registro del dominio para la página.
 - ✓ Costo estimado: \$600 USD/año.

Base de Datos:

Implementación de una base de datos relacional (por ejemplo, MySQL).

Costo estimado: Incluido en el servidor o \$100 USD/mes si es un servicio externo.

Capacitación y Soporte

Entrenamiento para administradores y usuarios clave (docentes y administradores).

- ✓ Costo estimado: \$400 USD.

Mantenimiento y soporte técnico anual.

- ✓ Costo estimado: \$600 USD.
 - Total de Costos Iniciales Aproximados: \$16,200 USD.
 - Costo Anual de Mantenimiento: \$1,000 USD.
 - 2 Beneficios Esperados
 - Eficiencia Administrativa:

Automatizar procesos como la inscripción de estudiantes y la gestión de notas reduce el tiempo y los errores manuales, optimizando los recursos humanos.

Accesibilidad y Transparencia:

Los estudiantes y docentes tienen acceso inmediato a información clave, mejorando la comunicación y la satisfacción del usuario.

- Reducción de Costos Operativos:

Al digitalizar los procesos, se reduce la necesidad de recursos físicos, como papel y almacenamiento. Se estima un ahorro anual de \$1,000 USD en recursos administrativos.

- Incremento en la Satisfacción de los Usuarios:

Una herramienta bien diseñada mejora la experiencia de los estudiantes y docentes, posicionando mejor a la institución.

3. Relación Costo-Beneficio

Inversión Inicial: \$10,000 USD.

Beneficios Anuales Estimados: \$15,200 USD (ahorros + eficiencia).

Punto de Equilibrio: 2 años después de la implementación.

Proyección a Largo Plazo:

En un horizonte de 5 años, los beneficios acumulados alcanzarían \$60,000 USD, superando ampliamente los costos iniciales y de mantenimiento ($\$16,200 + \$3,200 = \$19,400$ USD). Factibilidad en Tiempos:

El proyecto está programado para el ciclo 02-2024, lo que sugiere que hay tiempo suficiente para el desarrollo y la implementación de las funcionalidades clave, como la automatización de inscripciones, gestión de horarios y cupos, así como la validación de prerequisites. Sin embargo, la migración de un sistema heredado puede requerir una planificación cuidadosa y fases de prueba antes de su implementación final.

Conclusión temporal: Moderada factibilidad. Si bien el proyecto tiene un tiempo de entrega razonable, la migración de un sistema existente y la implementación de nuevas características pueden extender los plazos si no se gestionan adecuadamente.

1.7 Cronograma de actividades.

El diagrama de Gantt del proyecto "AcademiEnroll" permite visualizar la planificación y el desarrollo de las actividades involucradas en la Fase IV. En este se refleja la organización temporal de las tareas, asignación de responsabilidades y los

plazos para el cumplimiento de cada una de las etapas del proyecto, abarcando desde septiembre de 2024 hasta principios de diciembre de 2024.



1. Introducción (1 septiembre)

Adonay García redacta el contexto general del proyecto y las bases sobre las cuales se desarrollará el trabajo. Esta actividad establece los cimientos del proyecto, cubriendo el propósito y las metas principales.

2. Capítulo I (2 - 22 septiembre)

Gerardo Martínez lidera la creación del primer capítulo, abordando temas cruciales como los antecedentes y la importancia del proyecto. Se enfoca en establecer el marco teórico y la justificación del proyecto.

3. Antecedentes (3 - 4 septiembre)

Jonathan Martínez recopila y organiza información relevante sobre el tema del proyecto, proporcionando una base histórica y contextual para el desarrollo posterior.

4. Importancia del proyecto (4 - 6 septiembre)

Adonay García profundiza en la relevancia y el impacto del proyecto, explorando cómo contribuirá al campo y su aplicabilidad en contextos prácticos.

5. Planteamiento del Problema (5 septiembre)

Osaki Arévalo define el problema central que el proyecto busca resolver, identificando los desafíos clave que deben abordarse durante el desarrollo.

6. Definición del aplicativo (6 septiembre)

Erick Durán describe las características principales del sistema a desarrollar, detallando cómo este solucionará el problema planteado.

7. Justificación del Aplicativo (7 septiembre)

Diego Molina argumenta la necesidad del aplicativo, proporcionando razones claras de su viabilidad y beneficios tanto a nivel técnico como social.

8. Objetivos (8 - 9 septiembre)

Gerardo Martínez establece los objetivos generales y específicos del proyecto, marcando las metas que se deben alcanzar para cumplir con la visión propuesta.

9. General (9 septiembre)

Jonathan Martínez detalla el objetivo general del proyecto, asegurando que todas las actividades futuras estén alineadas con este propósito central.

10. Específicos (10 septiembre)

Adonay García desarrolla los objetivos específicos que guiarán las fases del proyecto, detallando acciones concretas que deben llevarse a cabo.

11. Alcances (11 septiembre)

Erick Durán define los límites del proyecto, especificando lo que está incluido y lo que no se abordará dentro de este marco.

12. Limitaciones (12 septiembre)

Diego Molina identifica las restricciones del proyecto, tales como recursos limitados, restricciones de tiempo o barreras tecnológicas.

13. Factibilidad del proyecto (13 - 14 septiembre)

Gerardo Martínez evalúa la viabilidad del proyecto desde diferentes perspectivas, como operativa, económica y de tiempo.

14. Factibilidad Operativa (13 septiembre)

Jonathan Martínez analiza si el proyecto puede ser implementado con los recursos y capacidades actuales, enfocándose en la operatividad.

15. Factibilidad Económica (14 septiembre)

Adonay García evalúa los costos asociados al proyecto y asegura que el presupuesto sea adecuado para su desarrollo.

16. 1.6.3 Factibilidad en Tiempos (15 septiembre)

Osaki Arévalo revisa si el proyecto se puede completar dentro del tiempo estipulado, considerando todas las fases y actividades previstas.

17. Cronograma de actividades (16 septiembre)

Erick Durán desarrolla un cronograma detallado con fechas de inicio y fin para cada actividad del proyecto, asegurando la correcta planificación del tiempo.

18. Entrega Capítulo I (17 septiembre)

Gerardo Martínez prepara la entrega del primer capítulo, consolidando toda la información recolectada y organizada hasta ese momento.

19. Capítulo II (23 septiembre - 20 octubre)

Gerardo Martínez toma la responsabilidad de desarrollar el segundo capítulo, que se centrará en las técnicas y herramientas de investigación que se utilizarán en el proyecto. Este capítulo incluirá una revisión de las fuentes documentales y el análisis de la investigación técnica necesaria para llevar a cabo el desarrollo del sistema.

20. Técnicas y herramientas de investigación (24 septiembre - 25 septiembre)

Jonathan Martínez lidera el análisis y selección de las técnicas de investigación más apropiadas para el proyecto. Este paso es crucial para obtener la información necesaria de manera eficiente y eficaz.

21. Técnicas de Investigación (26 - 27 septiembre)

Adonay García explora las diferentes técnicas que se utilizarán para recolectar datos e información para el desarrollo del sistema, como encuestas, entrevistas y otros métodos cualitativos o cuantitativos.

22. Revisión de Literatura y Fuentes Documentales (28 - 29 septiembre)

Erick Durán se enfoca en revisar la literatura relevante y las fuentes documentales que aporten una base sólida de conocimientos previos que guiarán el desarrollo del sistema.

23. Pruebas de Concepto y Prototipos (30 septiembre - 1 octubre)

Diego Molina lidera las pruebas de concepto y prototipos, realizando los primeros ensayos con ideas preliminares del sistema para evaluar su viabilidad y efectividad.

24. Investigación Técnica (2 - 3 octubre)

Gerardo Martínez coordina la investigación técnica para abordar los aspectos técnicos más complejos del sistema, como la arquitectura y las herramientas necesarias para su desarrollo.

25. Herramientas de investigación (4 octubre)

Jonathan Martínez selecciona y describe las herramientas que se utilizarán para la recolección y análisis de datos, garantizando que sean apropiadas para los objetivos del proyecto.

26. Herramientas utilizadas para elaborar el sistema (5 - 20 octubre)

Adonay García detalla las herramientas que se utilizarán para la creación del sistema, incluyendo lenguajes de programación, bases de datos y otras herramientas técnicas que permitirán su desarrollo.

27. Lenguaje de programación (5 octubre)

Osaki Arévalo selecciona el lenguaje de programación más adecuado para el sistema, considerando aspectos como la facilidad de implementación y la escalabilidad.

28. Sistema Gestor de Base de Datos (6 octubre)

Erick Durán especifica el sistema gestor de base de datos que se utilizará para almacenar la información y garantizar su accesibilidad durante el funcionamiento del sistema.

29. Otras herramientas utilizadas (7 - 8 octubre)

Diego Molina explora y describe otras herramientas que se emplearán en el desarrollo del sistema, como software para pruebas, control de versiones, o herramientas de diseño.

30. Análisis de Resultados (9 - 20 octubre)

Gerardo Martínez lidera el análisis de los resultados obtenidos durante el desarrollo y las pruebas del sistema, evaluando si se están alcanzando los objetivos previstos.

31. Entrega Capítulo II (20 octubre)

Gerardo Martínez organiza y entrega el Capítulo II, completando la documentación relacionada con la investigación, técnicas, herramientas y resultados preliminares obtenidos hasta ese momento.

32. Capítulo III (21 octubre - 16 noviembre)

Gerardo Martínez coordina la creación del Capítulo III, que se centrará en el diseño e implementación del sistema, incluyendo la estructura modular, la base de datos, y los aspectos de seguridad. Este capítulo es clave para la implementación técnica del proyecto.

33. Diseño del sistema (22 - 24 octubre)

Diego Enrique Hernández Molina desarrolla el diseño del sistema, estableciendo la estructura y apariencia general, asegurando una interfaz intuitiva y funcional.

34. Mapa del sistema o diagrama gráfico modular (25 - 26 octubre)

Diego Vladimir Durán Cabezas crea un mapa o diagrama modular del sistema, visualizando los diferentes componentes y sus interacciones.

35. Descripción de los distintos módulos (27 - 29 octubre)

Osaki Arévalo documenta y describe cada uno de los módulos que componen el sistema, detallando su propósito y funcionamiento en el contexto del proyecto.

36. Descripción de las distintas funciones utilizadas (30 - 31 octubre)

Brandol explica en detalle las funciones empleadas en el sistema, incluyendo la lógica y el rol de cada una en el desempeño global.

37. Base de Datos - Descripción y Diagrama Entidad-Relación (1 - 3 noviembre)

Erick Manuel Durán Cabezas elabora la descripción de la base de datos, incluyendo el Diagrama Entidad-Relación que representa las relaciones entre las diferentes tablas y entidades.

38. Base de Datos - Diccionario de Datos (4 - 5 noviembre)

Gerardo Martínez documenta el Diccionario de Datos, definiendo los atributos y detalles de cada campo en la base de datos, garantizando claridad en la estructura y uso de los datos.

39. Seguridad del sistema - Claves de acceso y Roles de usuario (6 - 7 noviembre)

Adonay García desarrolla las claves de acceso y asigna roles de usuario, implementando niveles de seguridad que limitan el acceso a funciones específicas según los permisos asignados.

40. Seguridad del sistema - Privilegios asignados a los usuarios (8 - 10 noviembre)

Alex define los privilegios para cada usuario, diferenciando los permisos y accesos según roles, y asegurando que cada usuario tenga el acceso adecuado para su rol en el sistema.

41. Entrega Capítulo III (16 noviembre)

Gerardo Martínez se encarga de entregar el tercer capítulo, que incluye toda la documentación del proyecto, incluidos los objetivos, avances, integración y análisis de los resultados finales.

42. Objetivo General (21 octubre)

Osaki Arévalo formaliza el objetivo general del proyecto, alineando todas las actividades y avances con esta meta principal.

43. Avances en los Objetivos Específicos (22 - 23 octubre)

Erick Durán documenta los avances realizados en relación con los objetivos específicos del proyecto, detallando las fases ya completadas y las que quedan por realizar.

44. Gestión de inscripciones con validación de requisitos y cupos (24 - 25 octubre)

Diego Molina trabaja en la implementación del sistema de gestión de inscripciones, asegurando que se validen los requisitos y cupos disponibles en el sistema.

45. Integración de la base de datos (26 - 27 octubre)

Gerardo Martínez coordina la integración de la base de datos con el sistema, asegurando que la información esté correctamente organizada y accesible para los usuarios.

46. Generación de reportes (28 - 29 octubre)

Jonathan Martínez desarrolla las funciones que permitirán generar reportes desde el sistema, proporcionando resúmenes y análisis de la información recabada.

47. Referencias (30 octubre)

Adonay García se encarga de la recopilación y organización de todas las referencias bibliográficas utilizadas a lo largo del proyecto, garantizando que se sigan los estándares apropiados de citación.

48. Glosario (31 octubre)

Gerardo Martínez elabora el glosario, definiendo términos técnicos y específicos que se utilizaron en el proyecto para facilitar su comprensión.

49. Anexos (1 - 2 noviembre)

Gerardo Martínez organiza y entrega los anexos, incluyendo diagramas, gráficos y cualquier otro material adicional relevante para la comprensión completa del proyecto.

50. Entrega Capítulo III (16 noviembre)

Gerardo Martínez se encarga de entregar el tercer capítulo, el cual incluye toda la documentación final del proyecto, incluidos los objetivos, avances, integración y análisis de los resultados finales.

Capítulo II

Metodología y técnicas de investigación

2.1 Técnicas y herramientas de investigación

2.1.1. Técnicas de Investigación

Para el desarrollo de este proyecto, utilizamos la búsqueda documental como técnica de investigación. Este método se basa en la recopilación y análisis de información a partir de fuentes disponibles en internet, como artículos, videos, sitios web especializados y otros recursos digitales. La búsqueda documental es útil para obtener conocimientos previos y contextuales sobre el tema, ya que permite acceder a una gran variedad de fuentes de manera rápida y eficiente.

2.1.2 Herramientas de investigación

herramientas de investigación empleadas en el desarrollo del proyecto "AcademiEnroll", describiendo su utilización y la relevancia que tienen dentro del contexto del proyecto:

1. Estudios de Caso

- **Uso:** Se revisaron estudios de caso sobre la implementación de sistemas de inscripción en instituciones educativas, con un enfoque particular en la migración de sistemas heredados (como ASP clásico) a tecnologías modernas como ASP.NET Core.
- **Relevancia:** Estos estudios proporcionaron una comprensión más profunda de los desafíos comunes y las mejores prácticas en la creación de sistemas escalables, eficientes y seguros, lo que permitió tomar decisiones informadas sobre la arquitectura y la estrategia de implementación del proyecto.

2. Prototipos

- **Uso:** Se diseñaron y probaron prototipos interactivos de la interfaz de usuario (UI) utilizando herramientas como Figma y Adobe XD. Los prototipos permitieron simular el flujo de inscripción, la navegación y la interacción con el sistema.

- Relevancia: Los prototipos facilitaron la validación temprana de la experiencia del usuario (UX), ayudando a identificar posibles áreas de mejora en el diseño y asegurando que el sistema sea intuitivo y fácil de usar para los estudiantes y el personal administrativo.

3. Análisis Comparativo

- Uso: Se realizaron análisis comparativos entre diferentes soluciones tecnológicas disponibles, evaluando frameworks como ASP.NET Core frente a otros sistemas más antiguos, como ASP clásico. También se compararon opciones de bases de datos y estrategias de seguridad para elegir las más adecuadas para el sistema de inscripción.
- Relevancia: El análisis comparativo permitió seleccionar las tecnologías y enfoques más adecuados para garantizar que el sistema "AcademiEnroll" sea escalable, seguro y capaz de manejar el aumento de estudiantes y solicitudes en la institución educativa.

2.2.1 Lenguaje de programación

El desarrollo del sistema "AcademiEnroll" ha sido realizado utilizando C# como lenguaje de programación principal. C# es un lenguaje orientado a objetos creado por Microsoft que se destaca por su robustez, flexibilidad y capacidad para gestionar aplicaciones complejas. Esta elección responde a la necesidad de implementar un sistema de inscripción moderno y escalable, que pueda responder eficazmente a las demandas tecnológicas actuales y futuras de la institución.

C# ofrece una integración nativa con el framework ASP.NET Core, lo cual es crucial para este proyecto. La combinación de C# y ASP.NET Core permite el desarrollo de aplicaciones web dinámicas y seguras, proporcionando herramientas avanzadas para el manejo de solicitudes simultáneas y la administración de datos. Gracias a esta integración, el sistema es capaz de implementar funcionalidades complejas como la validación automática de prerequisites, la asignación de cupos, y la automatización de tareas administrativas, reduciendo significativamente los errores humanos y optimizando los tiempos de inscripción.

El uso de C# también facilita el mantenimiento continuo del sistema. Dado que es un lenguaje ampliamente utilizado en la industria, existen abundantes recursos, bibliotecas y soporte técnico que garantizan que el sistema pueda ser actualizado y mejorado a lo largo del tiempo. Además, su compatibilidad con la programación asincrónica y el manejo de errores contribuye a una experiencia de usuario más fluida y a un rendimiento del sistema que se mantiene estable incluso en periodos de alta demanda, como los procesos de inscripción masiva.

En conclusión, la elección de C# como lenguaje de programación central para el proyecto "AcademiEnroll" no solo responde a necesidades técnicas inmediatas, sino que también proporciona una base sólida para futuras expansiones y mejoras del sistema, garantizando su adaptabilidad y eficiencia en la gestión académica.

2.2.2 Sistema Gestor de base de datos

El sistema "AcademiEnroll" utiliza SQL Server como su sistema gestor de base de datos (SGBD). La elección de esta plataforma responde a la necesidad de manejar grandes volúmenes de datos de manera eficiente, escalable y segura, permitiendo una administración robusta de la información relacionada con estudiantes, materias, inscripciones y validación de prerrequisitos.

SQL Server se ha integrado de manera fluida con las tecnologías principales utilizadas en el desarrollo del sistema, como ASP.NET Core y Entity Framework Core. Esta integración no solo facilita el acceso y manipulación de datos, sino que también permite una rápida adaptación a las necesidades cambiantes de la institución académica. Entity Framework Core habilita el mapeo objeto-relacional, permitiendo que los desarrolladores trabajen directamente con objetos de dominio sin necesidad de escribir consultas SQL de manera manual, lo que agiliza y optimiza el proceso de desarrollo.

Uno de los beneficios clave de SQL Server es su capacidad para manejar transacciones de manera confiable, lo que asegura que las operaciones críticas, como las inscripciones y la asignación de cupos, se realicen de manera atómica y consistente, minimizando el riesgo de errores y problemas de integridad de los datos. Esto es especialmente crucial durante los picos de demanda, como los periodos de inscripción masiva, donde el sistema debe procesar múltiples solicitudes simultáneamente sin comprometer la estabilidad.

La estructura actual de la base de datos incluye una serie de tablas fundamentales, tales como:

- **Estudiantes:**

Almacena la información personal, académica y de contacto de cada estudiante inscrito, así como su historial de inscripciones y estados dentro de la institución.

- **Materias:**

Contiene los detalles de las materias ofrecidas por la institución, incluyendo prerequisites, profesores asignados, horarios y límites de cupos por clase.

- **Inscripciones:**

Registra las solicitudes de inscripción realizadas por los estudiantes, junto con el estado de cada inscripción (pendiente, confirmada, o rechazada) y el seguimiento de los cupos disponibles.

- **Prerrequisitos:**

Define las relaciones entre las materias y los requisitos necesarios para que un estudiante pueda inscribirse en ellas, garantizando el flujo adecuado a través de los distintos niveles académicos.

- **Roles y Usuarios:**

Implementa un control de acceso detallado para asegurar que los distintos tipos de usuarios (administradores, docentes, estudiantes) solo puedan acceder a la información y realizar las acciones pertinentes según su nivel de permisos.

La base de datos ha sido diseñada con una fuerte orientación a la escalabilidad, de manera que pueda manejar la creciente cantidad de estudiantes y materias sin perder rendimiento ni comprometer la experiencia de usuario. Se han implementado índices sobre las columnas más consultadas y claves primarias para optimizar las consultas frecuentes, reduciendo los tiempos de respuesta en operaciones como la búsqueda de estudiantes o materias, y el cálculo de disponibilidad de cupos en tiempo real.

El sistema actual también se beneficia de las capacidades de respaldo y recuperación de SQL Server, que permiten una rápida respuesta en caso de cualquier eventualidad técnica, asegurando que los datos académicos permanezcan seguros y accesibles en todo momento.

En resumen, SQL Server ha demostrado ser una elección robusta para la gestión de la base de datos de "AcademiEnroll", proporcionando un entorno confiable, seguro y eficiente que garantiza la integridad de los datos académicos y ofrece una base sólida sobre la cual continuar desarrollando y mejorando el sistema de inscripción automatizado.

2.2.3 Otras herramientas utilizadas

Hasta el momento de esta documentación no se ha hecho uso de herramientas externas de las que brinda ASP.NET Core al crear un proyecto vacío que son las siguientes:

- **Razor Pages:** Piensa en esto como una manera elegante de mezclar HTML con código o de servidor, todo en uno.
- **MVC:** Este patrón de diseño divide tu aplicación en tres partes: la lógica (modelos), lo que ves (vistas) y cómo interactúan (controladores).
- **Entity Framework Core:** Esto te permite trabajar con bases de datos usando objetos de .NET, así no tienes que escribir tanto SQL.
- **Blazor:** Una herramienta genial que te permite crear aplicaciones web interactivas usando C# en lugar de JavaScript.
- **Identity:** Un sistema para manejar la autenticación y autorización, o sea, quién puede entrar y qué puede hacer dentro de tu app.
- **Middleware:** Piezas de software que se ejecutan entre las peticiones y respuestas HTTP, haciendo cosas como autenticación, almacenamiento en caché, etc.
- **Dependency Injection:** Este patrón de diseño te ayuda a gestionar dependencias, mejorando la modularidad y facilitando las pruebas de tu app.

2.3 Análisis de Resultados

Breve teoría o marco conceptual:

La búsqueda documental se fundamenta en el enfoque de recolección de información secundaria, donde el investigador examina documentos y recursos existentes en lugar de generar

datos primarios. Según el marco conceptual de la investigación documental, esta técnica permite construir una base teórica sólida y adquirir conocimientos variados, lo cual facilita la comprensión del fenómeno estudiado. En este caso, la búsqueda en Google y el uso de videos en línea nos permitieron explorar distintas perspectivas y prácticas relacionadas con sistemas de inscripción y registro en educación superior.

Presentación y análisis de resultados:

A través de la técnica de búsqueda documental en Internet, recopilamos información sobre prácticas y sistemas utilizados en el registro y gestión de inscripciones en centros educativos. Las siguientes son las principales conclusiones y análisis de los resultados obtenidos:

1. Sistemas de registro y gestión en educación:

Al investigar en fuentes como artículos, videos y sitios web especializados, identificamos características comunes en los sistemas de inscripción, como la facilidad de uso, la seguridad de los datos y la capacidad de generar reportes de rendimiento académico. Estos elementos se tomaron en cuenta al conceptualizar el sistema "Academi Course".

2. Análisis de las necesidades del usuario:

La información recopilada destacó la importancia de una interfaz intuitiva que permita a los usuarios (administradores y estudiantes) acceder a sus datos de manera rápida y segura. Los videos y artículos enfatizaron que un sistema exitoso debe reducir el tiempo de procesamiento y facilitar la navegación para usuarios con distintos niveles de habilidades tecnológicas.

3. Comparación con otros sistemas:

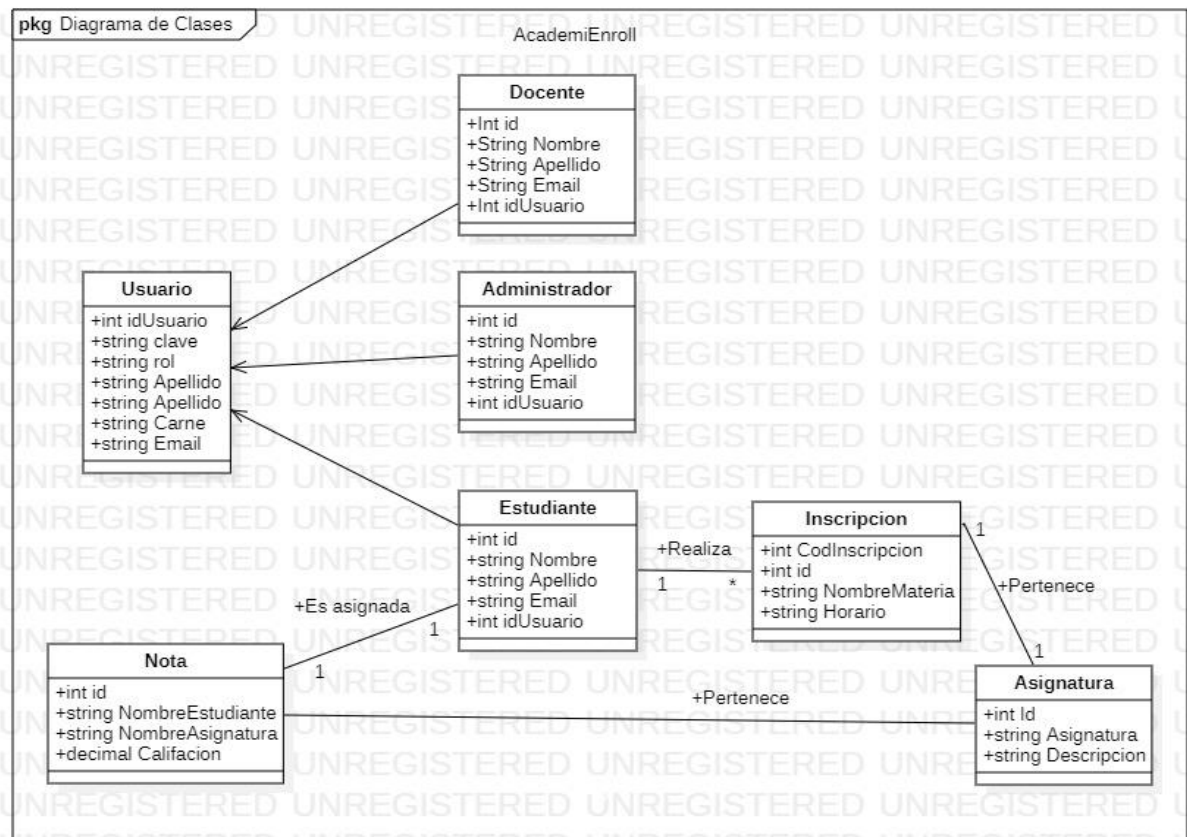
La revisión de otros sistemas similares, a través de un análisis comparativo en los recursos disponibles en línea, mostró que los sistemas más eficientes integran módulos de inscripción, seguimiento de calificaciones y comunicación entre usuarios. Este análisis nos ayudó a definir los módulos esenciales para "Academi Course", priorizando la funcionalidad y la adaptabilidad a distintas plataformas.

Capítulo III

Diseño del Sistema

3.1 Diseño del Sistema

Diagrama de Clases:



Clases Principales

1. Usuario

- Clase general con atributos comunes para todos los usuarios del sistema: idUsuario, clave, rol, Apellido, Carne, Email.
- Docente, Administrador y Estudiante son de esta clase, estos tipos de usuario heredan a la clase, en otras palabras, todos estos tienen las mismas características básicas de la clase padre.

2. Docente

- Utilizado para representar al docente de la institución.
- Tiene un id, Nombre, Apellido, y Email, así como un idUsuario, que indica la relación con el usuario.
- Se utiliza heredando de la clase Usuario ya que lo es.

3. Administrador

- Esta clase representa a los administradores del sistema, encargados de la gestión de inscripciones y generación de reportes.
- Como en el caso del Docente, también hereda los atributos de la clase Usuario.

4. Estudiante

- Representa a los estudiantes que pueden inscribirse en diferentes asignaturas.
- Contiene atributos como id, Nombre, Apellido, y Email, y un idUsuario que permite identificar la relación con la clase Usuario.
- Se relaciona con la clase Inscripcion a través de una relación "Realiza", indicando que un estudiante puede realizar múltiples inscripciones.

5. Inscripcion

- Esta clase representa el acto de inscripción en una asignatura específica.
- Contiene atributos como CodInscripcion, id, NombreMateria, y Horario.
- La relación entre Estudiante e Inscripcion es de "1 a muchos" (un estudiante puede realizar varias inscripciones).

6. Asignatura

- Esta clase define las asignaturas que están disponibles para inscribirse, con atributos como Id, Asignatura y Descripcion.

- Existe una relación entre Inscripcion y Asignatura llamada "Pertenece", indicando que una inscripción está asociada a una asignatura específica.

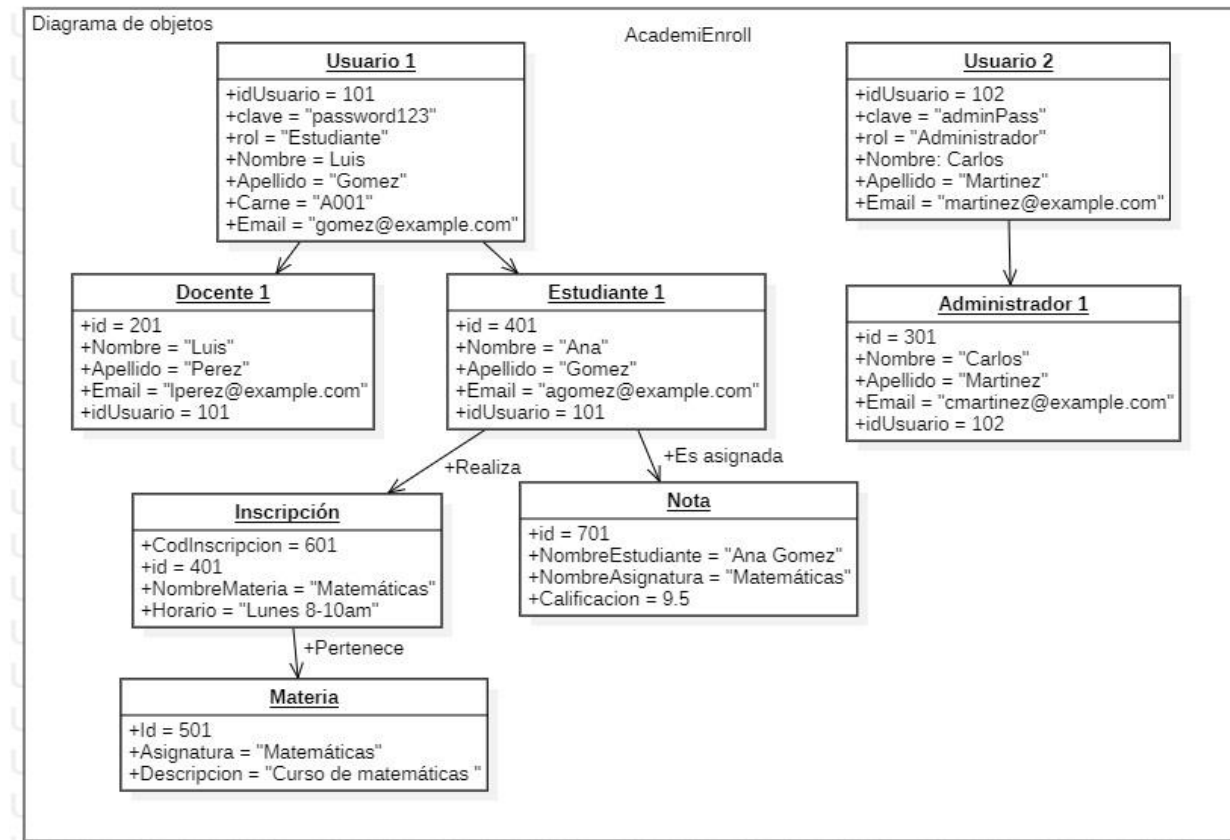
7. Nota

- Representa las calificaciones obtenidas por los estudiantes en las asignaturas.
- Tiene atributos como id, NombreEstudiante, NombreAsignatura, y Calificacion.
- La relación entre Estudiante y Nota indica que a un estudiante se le asigna una nota en una asignatura específica.

Relaciones Principales

- **Herencia:** Las clases Docente, Administrador y Estudiante heredan de la clase Usuario, indicando que todos ellos comparten características comunes pero poseen roles distintos en el sistema.
- **Asociación:**
 - Entre Estudiante e Inscripcion, existe una relación de "1 a muchos", ya que un estudiante puede realizar múltiples inscripciones.
 - Entre Inscripcion y Asignatura, también existe una relación de "1 a 1" llamada "Pertenece", en la cual cada inscripción está asociada a una asignatura específica.
- **Agregación/Composición:**
 - La relación entre Estudiante y Nota muestra que un estudiante puede tener varias notas asignadas para diferentes asignaturas.

Diagrama de objetos:



- En el diagrama de objetos, las clases Estudiante, Inscripción y Asignatura (llamada Materia en el diagrama) permiten a los estudiantes gestionar el proceso de inscripción. La relación entre Estudiante e Inscripción refleja el proceso de inscripción en línea que los estudiantes realizaran.
- El objeto Estudiante 1 representa un ejemplo de un estudiante inscrito en una materia específica, demostrando cómo los estudiantes pueden interactuar con el sistema para seleccionar materias.
- En el diagrama, la clase Inscripción se relaciona con Estudiante y Asignatura (Materia en el diagrama). Esto permite gestionar la inscripción en diferentes materias.
- El diagrama refleja cómo los datos de los estudiantes (Estudiante), materias (Materia) y detalles de inscripción (Inscripción) estarán estructurados y

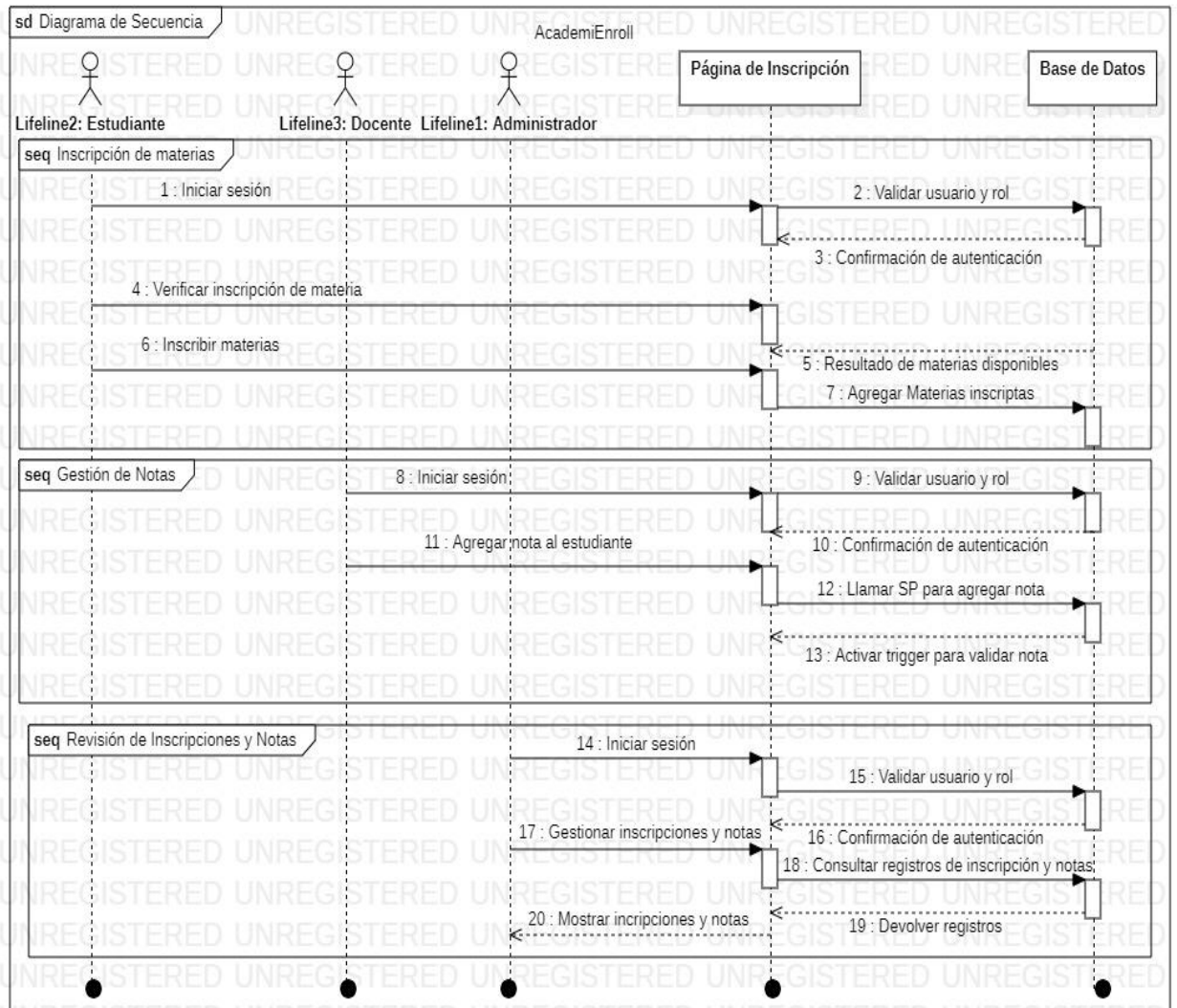
almacenados. Al integrarse con una base de datos, estas clases permitirán el almacenamiento y acceso eficiente a esta información.

- La seguridad y eficiencia en el acceso a los datos estarán en parte aseguradas por la relación estructurada entre las clases, lo que facilita la gestión y recuperación de información en función de las necesidades del sistema
- La clase Administrador permite a los administradores acceder y gestionar la información del sistema. Además, los administradores podrán generar reportes basados en la información contenida en Estudiante, Inscripción y Materia.
- La relación entre Administrador y Usuario muestra que los administradores tienen acceso autorizado al sistema, lo cual es esencial para ejecutar modificaciones y manejar reportes.

Resumen

El diagrama de objetos ilustra cómo los diferentes componentes interactúan entre sí para cumplir con los objetivos propuestos. Las clases y las relaciones reflejan las funcionalidades necesarias para automatizar y gestionar el proceso de inscripción en línea, así como para permitir el acceso seguro a la información y la generación de reportes por parte de los administradores. Esto asegura que el sistema no solo sea rápido y preciso, sino también seguro y adaptable a las necesidades de gestión académica de la institución.

Diagrama de Secuencia:



1. Inscripción de materias

Esta secuencia describe el flujo para que un **Estudiante** se inscriba en materias:

- **Paso 1:** El estudiante inicia sesión en el sistema.
- **Paso 2:** La **Página de Inscripción** valida el usuario y rol del estudiante con la **Base de Datos**.

- **Paso 3:** La **Página de Inscripción** confirma la autenticación del estudiante.
- **Paso 4:** El estudiante solicita verificar las materias en las que puede inscribirse.
- **Paso 5:** La página consulta a la **Base de Datos** y obtiene el resultado de las materias disponibles.
- **Paso 6:** El estudiante elige las materias en las que desea inscribirse.
- **Paso 7:** La **Página de Inscripción** agrega las materias seleccionadas a las inscripciones del estudiante en la **Base de Datos**.

2. Gestión de notas

Esta secuencia muestra el proceso para que un Docente gestione las notas de los estudiantes:

- **Paso 8:** El docente inicia sesión en el sistema.
- **Paso 9:** La **Página de Inscripción** valida al usuario y su rol en la **Base de Datos**.
- **Paso 10:** La página confirma la autenticación del docente.
- **Paso 11:** El docente solicita agregar una nota para un estudiante.
- **Paso 12:** La página llama a un procedimiento almacenado (SP) en la **Base de Datos** para agregar la nota.
- **Paso 13:** Se activa un disparador (trigger) en la base de datos para validar la nota ingresada.

3. Revisión de inscripciones y notas

Esta secuencia permite al **Administrador** revisar las inscripciones y notas en el sistema:

- **Paso 14:** El administrador inicia sesión en el sistema.
- **Paso 15:** La **Página de Inscripción** valida al usuario y su rol en la **Base de Datos**.
- **Paso 16:** La página confirma la autenticación del administrador.

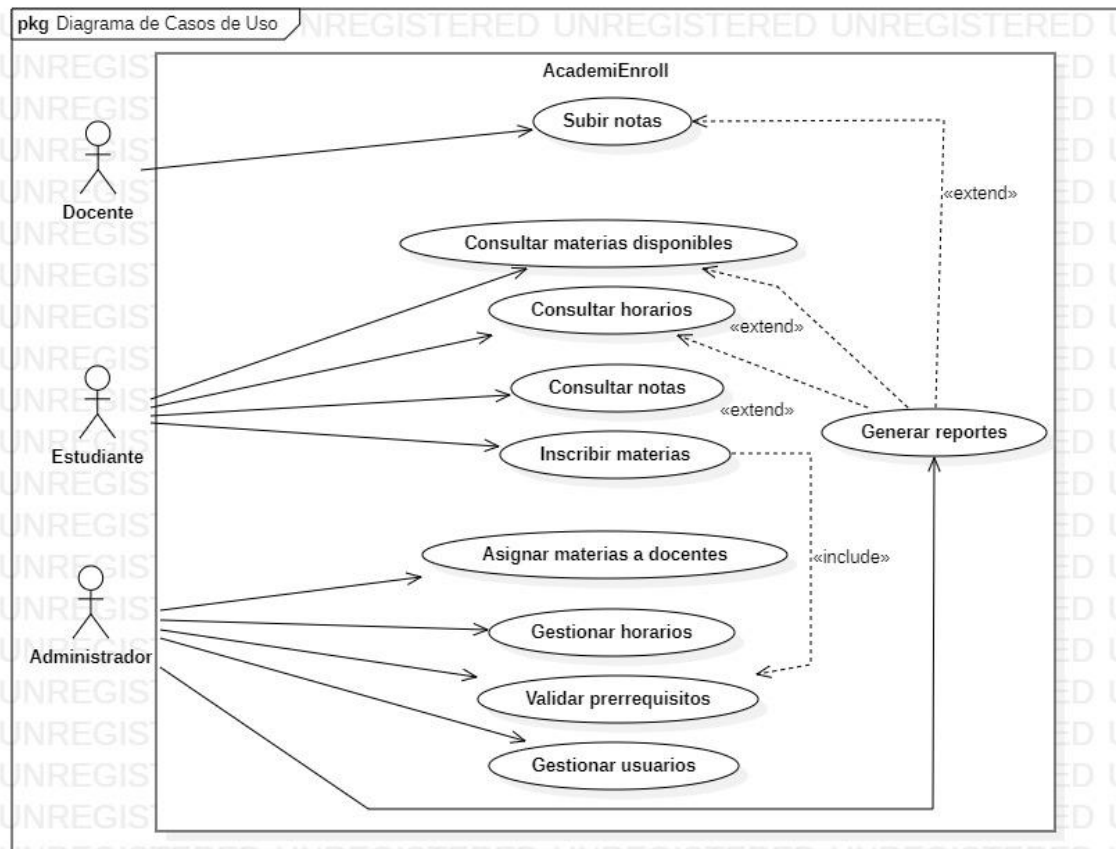
- **Paso 17:** El administrador solicita gestionar los registros de inscripciones y notas.
- **Paso 18:** La página consulta a la **Base de Datos** para obtener los registros correspondientes.
- **Paso 19:** La base de datos devuelve los registros solicitados.
- **Paso 20:** La página muestra los registros de inscripciones y notas al administrador.

Resumen

Este diagrama de secuencia muestra las interacciones clave en el sistema:

- **Inscripción de materias** por parte de estudiantes.
- **Gestión de notas** por parte de los docentes.
- **Revisión de inscripciones y notas** por parte de los administradores.

Cada proceso sigue una estructura de autenticación inicial, seguido de acciones específicas según el rol del usuario y, finalmente, la actualización o consulta en la base de datos. Este diseño sigue los requisitos del sistema de proporcionar acceso seguro y específico según el rol del usuario, mientras se asegura una interacción eficiente con la base de datos.



Actores y Casos de Uso

1. Docente

- **Subir notas:** El docente puede registrar las notas de los estudiantes.
- **Consultar materias disponibles:** Los docentes pueden ver las materias que están disponibles en el sistema.
- **Generar reportes** (extiende de Subir notas y Consultar materias disponibles): Permite a los docentes generar reportes relacionados con las materias y notas de sus estudiantes.

2. Estudiante

- **Consultar materias disponibles:** Los estudiantes pueden ver las materias que están disponibles para inscripción.

- **Consultar horarios:** Los estudiantes pueden ver los horarios de las materias. Este caso de uso se extiende de Consultar materias disponibles, ya que primero deben conocer las materias para consultar los horarios.
- **Consultar notas:** Los estudiantes pueden ver sus calificaciones en las materias inscritas. Este caso de uso también extiende Generar reportes, permitiendo obtener detalles de sus notas.
- **Inscribir materias:** Permite a los estudiantes inscribirse en las materias seleccionadas.

3. Administrador

- **Asignar materias a docentes:** El administrador puede asignar materias a los docentes según el área de especialización o necesidades académicas.
- **Gestionar horarios:** Los administradores pueden organizar y ajustar los horarios de las materias.
- **Validar prerrequisitos:** Los administradores pueden verificar si un estudiante cumple con los requisitos previos para inscribirse en una materia específica.
- **Gestionar usuarios:** Permite al administrador gestionar los usuarios del sistema, incluyendo la creación, modificación y eliminación de cuentas de estudiantes, docentes y otros administradores.
- **Generar reportes** (incluido en varios casos): Permite generar reportes sobre diferentes aspectos, como inscripciones, notas y asignación de docentes. Este caso de uso es accesible para el administrador y está relacionado con otras funcionalidades, lo que le permite un análisis integral del sistema.

Relación entre Casos de Uso

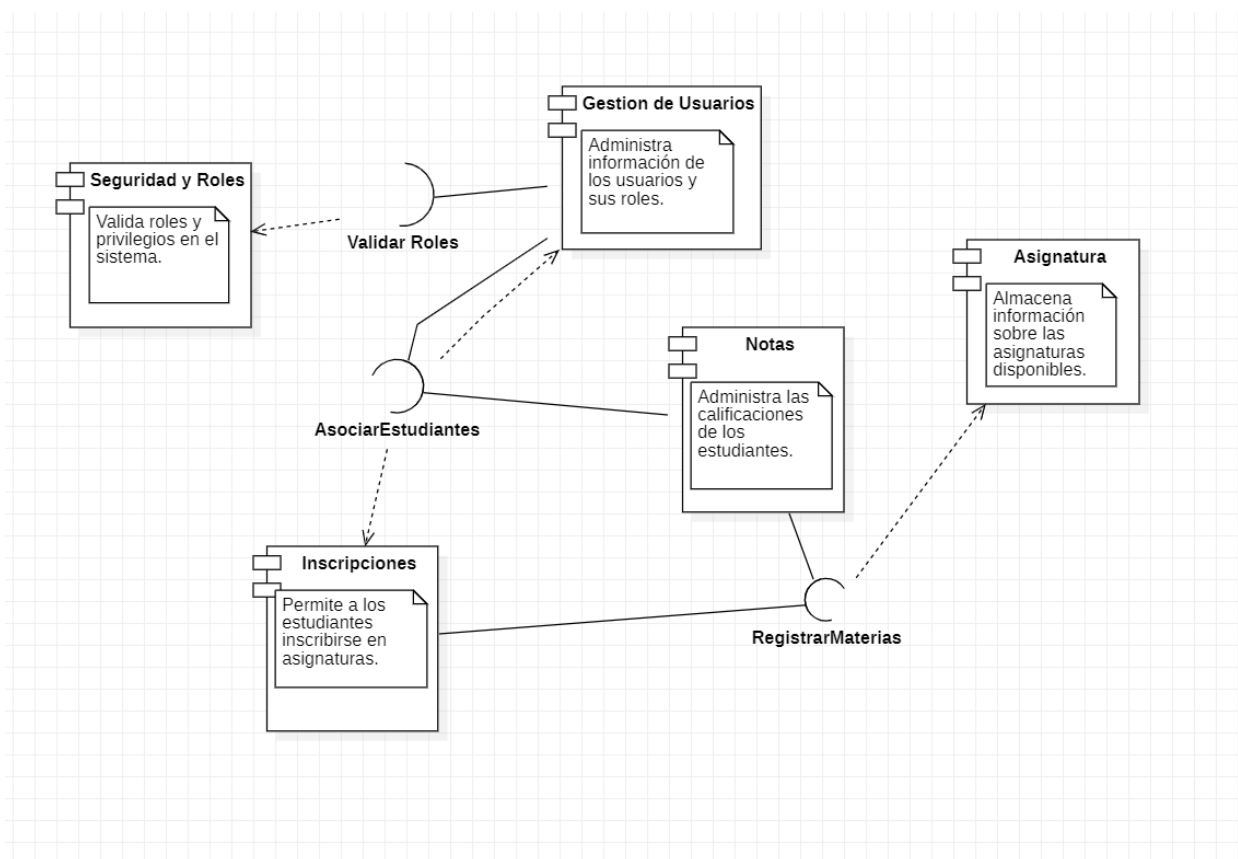
- **«extend»:** Indica una relación opcional donde un caso de uso complementa a otro, como en **Consultar horarios** y **Consultar notas**, que extienden **Consultar materias disponibles** para que el usuario pueda ver los horarios y notas de las materias en las que está inscrito.

- «**include**»: La relación «include» aparece en **Generar reportes** con otros casos, indicando que la generación de reportes es una función común que se incluye en varios procesos administrativos, permitiendo al sistema crear informes detallados.

Resumen

El presente diagrama de casos de uso define las funcionalidades clave del sistema y la manera en que cada rol interactúa con ellas. Estudiantes pueden ver y gestionar sus inscripciones y notas; docentes pueden asignar notas; mientras que administradores tienen un control más amplio, gestionando tanto el acceso de los usuarios como las inscripciones y horarios de las materias. La estructura del diagrama asegura que las responsabilidades estén bien definidas y distribuidas, alineándose con los objetivos del sistema de proporcionar una plataforma eficiente, segura y fácil de usar para la gestión académica.

3.2 Mapa del sistema o diagrama grafico modular



3.3 Descripción de los distintos módulos utilizados para la construcción del sistema

En esta sección se describen los módulos los módulos clave de aplicativo web, para ellos se detallarán los formularios, clases, atributos y métodos principales. Con el propósito de brindar una descripción del funcionamiento y justificar as decisiones tomadas desde una perspectiva funcional.

1. Clases principales:

- **Administrador:** Clase que contiene el modelo de datos de la entidad que gestiona el sistema, se utiliza para gestionar todas las demás clases del sistema.
- **Docente:** Clase que contiene el modelo de docentes, su principal uso es para gestionar docentes que estén dentro del sistema.
- **Estudiante:** Esta clase contiene los atributos de la entidad estudiantes permitiendo pasar los datos entre la capa de modelo y la de procesos para realizar operaciones como inicio de sesión, consulta de datos e inscripción de materias.
- **Inscripción:** Este modelo encapsula los campos necesarios para realizar una inscripción de materias por parte del alumno mediante el uso de su portal educativo en línea.
- **Notas:** Este modelo contiene los atributos necesarios para ingresar, gestionar y editar datos usado en las vistas de alumnos, docentes y administradore para gestionar la nota global de los estudiantes.
- **Usuarios:** Este encapsula los componentes necesarios para poder reconocer a un usuario, usado principalmente para validación y autenticación de los usuarios y poder diferenciar entre los diferentes roles disponibles (Administrador, Docente, Alumno).

2. Estructura de Clases y Modelos

Clases Implementadas:

- **Usuario:**

Atributos:

- **idUsuario:** Identificador único del usuario.
- **Correo:** Correo electrónico del usuario.
- **Clave:** Contraseña cifrada.
- **Rol:** Define si el usuario es estudiante, docente o administrador.

Métodos Principales:

- **ValidarLogin(correo, clave):** Verifica si las credenciales ingresadas coinciden con un usuario en la base de datos.
- **AsignarRol(rol):** Asigna el rol correspondiente al usuario después de la validación.

- **Estudiante:**

Atributos:

- **idEstudiante:** Identificador único.
- **Nombre:** Nombre del estudiante.
- **MateriasInscritas:** Lista de materias asociadas al estudiante.
- **Notas:** Calificaciones obtenidas en cada materia.

Métodos Principales:

- **InscribirMateria(materia):** Agrega una materia a la lista de materias inscritas.
- **ConsultarNotas():** Devuelve las calificaciones de todas las materias inscritas.

- **Docente**

Atributos:

- **idDocente:** Identificador único.
- **Nombre:** Nombre del docente.
- **MateriasAsignadas:** Lista de materias que el docente imparte.

Métodos Principales:

- **RegistrarNota(estudiante, materia, nota):** Asocia una calificación a un estudiante en una materia específica.
- **VerEstudiantes(materia):** Lista los estudiantes inscritos en una materia impartida por el docente.
- **Administrador**

Atributos:

- **idAdministrador:** Identificador único.
- **Nombre:** Nombre del administrador.

Métodos Principales

- **RegistrarUsuario(tipo, datos):** Registra un nuevo usuario (estudiante, docente o administrador).
- **AsignarMateria(docente, materia):** Asocia un docente con una materia específica.
- **Materia**

Atributos:

- **idMateria:** Identificador único.
- **Nombre:** Nombre de la materia.
- **Creditos:** Número de créditos académicos.

Métodos Principales:

- **ObtenerDetalles():** Devuelve información completa de la materia.

3. Formularios y Vistas

- **Formularios y Vistas Desarrollados**

Formulario de Login:

- **Función:** Permitir a los usuarios acceder al sistema.
- **Interacción con Clases:** Valida las credenciales en la clase Usuario y asigna el rol para redirigir al panel correspondiente.

Vista de Registro de Usuarios

- **Función:** Permite al administrador registrar estudiantes y docentes.
- **Interacción con Clases:** Llama al método RegistrarUsuario de la clase Administrador.

Vista de Inscripción de Materias

- **Función:** Facilita a los estudiantes la selección de materias.
- **Interacción con Clases:** Utiliza el método InscribirMateria de la clase Estudiante.

Formulario de Registro de Notas

- **Función:** Permite a los docentes asignar calificaciones a los estudiantes.
- **Interacción con Clases:** Utiliza el método RegistrarNota de la clase Docente.

Vista de Consulta de Calificaciones

- **Función:** Los estudiantes pueden ver sus calificaciones.
- **Interacción con Clases:** Llama al método ConsultarNotas de la clase Estudiante.

4. Estrategia de Diseño

- **Enfoque del Proyecto:**

El proyecto sigue el modelo MVC (Modelo-Vista-Controlador) para separar responsabilidades y facilitar el mantenimiento:

- **Modelo:** Define las clases principales (Usuario, Estudiante, Docente, Administrador, Materia) y maneja la lógica de negocio.
- **Vista:** Incluye formularios y pantallas para interactuar con el usuario, como login, registro e inscripción.
- **Controlador:** Actúa como intermediario entre las vistas y los modelos, gestionando las interacciones del usuario y actualizando los datos según sea necesario.

- **Módulos Funcionales:**

Las funcionalidades se agrupan según los roles:

- **Módulo de Estudiante:** Inscripción de materias y consulta de calificaciones.
- **Módulo de Docente:** Gestión de estudiantes y registro de notas.
- **Módulo Administrativo:** Gestión de usuarios y asignación de materias.

5. Justificación del Diseño

- **Clases:**

1. La clase Usuario centraliza la gestión de credenciales y roles, lo que permite un manejo uniforme de los accesos.
2. Las clases Estudiante, Docente y Administrador encapsulan las acciones específicas de cada rol, simplificando la organización del sistema.

- **Formularios:**

1. El formulario de inscripción de materias brinda a los estudiantes una herramienta intuitiva para interactuar con el sistema, lo que mejora su experiencia.
2. El formulario de calificaciones garantiza un flujo de trabajo eficiente para los docentes, alineándose con las necesidades educativas.

- **Arquitectura MVC:**

3. Facilita la escalabilidad del proyecto, permitiendo añadir nuevas funcionalidades sin afectar la estructura existente.
4. Promueve la claridad en el código y facilita el mantenimiento a largo plazo.

3.4 Descripción de las Distintas Funciones Utilizadas:

El sistema propuesto integra funcionalidades clave que garantizan el correcto funcionamiento y la interacción eficiente entre los distintos usuarios: estudiantes, docentes y administradores. A continuación, se describen las funciones principales, su propósito y su desarrollo dentro del sistema:

1. Login con Validación de Roles

- **Propósito:**

Permitir el acceso al sistema diferenciando entre estudiantes, docentes y administradores. Esto asegura que cada usuario acceda únicamente a las funciones correspondientes a su rol, manteniendo la seguridad y la organización del sistema.

- **Desarrollo:**

- ✓ El sistema utiliza un formulario de inicio de sesión que solicita credenciales (correo y contraseña).
- ✓ Al enviarse las credenciales, se valida en la base de datos si el usuario existe y a qué rol pertenece.

Según el rol, el usuario es redirigido a un panel de control específico:

- **Estudiante:** Acceso a inscripción de materias y consulta de calificaciones.
- **Docente:** Gestión de notas y visualización de estudiantes inscritos.
- **Administrador:** Gestión de usuarios (estudiantes y docentes) y asignación de materias.
- **Esencialidad:**

La validación por roles es fundamental para personalizar la experiencia del usuario y garantizar que no haya accesos indebidos a funcionalidades restringidas.

2. CRUD de Estudiante

- **Propósito:**

Administrar la información de los estudiantes, incluyendo su inscripción y asignación de materias.

- **Desarrollo:**
 - ✓ **Leer:** Se implementará el listado de sus materias inscritas y calificaciones.
 - ✓ **Actualizar:** Modificar datos personales o inscripciones.
- **Esencialidad:**

El manejo correcto de los datos de los estudiantes es crucial para que el sistema funcione como una herramienta administrativa confiable.

3. CRUD de Docente

- **Propósito:**

Permitir la gestión de los datos de los docentes, incluidos los cursos que imparten y los estudiantes inscritos.

- **Desarrollo:**
 - ✓ **Crear:** El docente puede cargar las notas de los estudiantes inscritos en su materia

- ✓ **Leer:** Visualización del listado de los estudiantes y las materias asignadas.

- **Esencialidad:**

El registro y seguimiento de los docentes es indispensable para asignar materias y gestionar correctamente las notas de los estudiantes.

4. CRUD de Administrador

- **Propósito:**

Gestionar los datos de todos los usuarios del sistema y realizar asignaciones clave, como vincular docentes con materias.

- **Desarrollo:**

- ✓ **Crear:** Registro de administradores adicionales o asignación de nuevos usuarios.
- ✓ **Leer:** Visualización de todos los datos del sistema, incluyendo estudiantes, docentes y materias.
- ✓ **Actualizar:** Realizar ajustes en los registros y asignaciones.
- ✓ **Eliminar:** Borrar usuarios que ya no pertenezcan al sistema.

- **Esencialidad:**

La función del administrador es central para mantener la estructura operativa del sistema y garantizar la correcta asignación de roles y materias.

5. Carga de Notas

- **Propósito:**

Facilitar a los docentes la inserción y modificación de calificaciones de los estudiantes inscritos en sus materias.

- **Desarrollo:**

- ✓ Los docentes acceden a un listado de estudiantes inscritos en cada materia.

- ✓ Se presenta un formulario para ingresar o actualizar las calificaciones.
- ✓ Las notas se almacenan en la base de datos y se reflejan en el perfil del estudiante para su consulta.

- **Esencialidad**

Es un módulo clave para mantener un registro preciso del desempeño académico, lo que constituye una parte central de la experiencia del estudiante en el sistema.

3.5 Base de datos

3.5.1 Descripción

1. Estructura General:

Nuestra base de datos AcademiEnroll está diseñada para gestionar un sistema de inscripción de educación superior. Incluye tablas para usuarios, estudiantes, docentes, asignaturas, inscripciones y notas, con relaciones bien definidas para mantener la integridad referencial.

2. Tablas Principales y Descripción:

- **Usuario:**

Almacena información sobre los usuarios, incluyendo su clave, rol (Estudiante, Docente, Administrador), nombre, apellido, carné y correo electrónico.

- **Estudiantes:**

Registra datos de los estudiantes. Cada estudiante está relacionado con un usuario específico.

- **Docentes:**

Similar a la tabla Estudiantes, pero para almacenar datos de los docentes.

- **Administrador:**

Tabla dedicada a los administradores, vinculada también a la tabla de usuarios.

- **Inscripciones:**

Registra las asignaturas en las que un estudiante está inscrito, con información como el nombre de la materia y el horario.

- **Asignatura:**

Contiene las asignaturas disponibles, con una breve descripción.

- **Notas:**

Almacena las calificaciones de los estudiantes por asignatura.

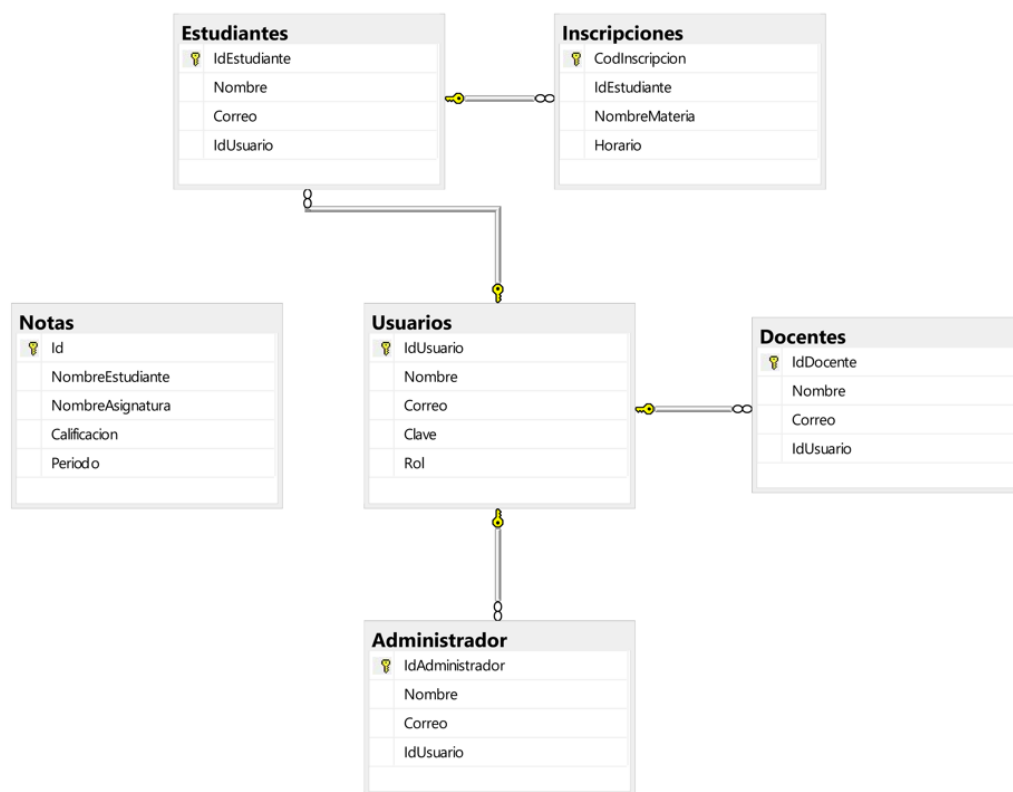
3. Relaciones Clave:

- **Usuario:** es la tabla base para identificar roles y está relacionada con Estudiantes, Docentes, y Administrador.
- **Estudiantes e Incripciones:** están relacionados por el campo IdEstudiante.
- **Incripciones y Asignatura:** están conectadas mediante el nombre de la materia (NombreMateria).
- **Notas:** utiliza los nombres de estudiantes y asignaturas para registrar calificaciones.

4. Validaciones y Procedimientos:

- **Triggers:** aseguran que las asignaturas se registren automáticamente tras una inscripción y que las calificaciones sean válidas (entre 0 y 10).
- **Procedimientos almacenados:** permiten agregar o actualizar notas con validaciones adicionales.

3.5.2 Diagrama Entidad-Relación



3.5.3 Diccionario de Datos

Usuarios				
Campo	Tipo de Dato	Longitud	Restricciones	Descripción
IdUsuario	INT	N/A	Primary Key, Identity	Identificador único y automático de cada usuario en el sistema.
Nombre	NVARCHAR	100	Not Null	Nombre completo del usuario.
Correo	NVARCHAR	100	Not Null, Único	Correo electrónico del usuario, usado como

				identificación única para el acceso al sistema.
Clave	NVARCHAR	100	Not Null	Contraseña de acceso del usuario al sistema.
Rol	NVARCHAR	50	Not Null	Rol que identifica el tipo de usuario (por ejemplo: Administrador, Estudiante, Docente).
IdEstudiante	INT	N/A	Primary Key, Identity	Almacena IdEstudiante

Estudiantes				
Campo	Tipo de Dato	Longitud	Restricciones	Descripción
IdEstudiante	INT	N/A	Primary Key, Identity	Identificador único y automático de cada estudiante en el sistema.
Nombre	NVARCHAR	100	Not Null	Nombre completo del estudiante.
Correo	NVARCHAR	100	Not Null, Único	Correo electrónico del estudiante, usado como contacto y verificación.
IdUsuario	INT	N/A	Foreign Key References Usuarios(IdUsuario)	Referencia al usuario en la tabla Usuarios, identificando al estudiante con su cuenta de usuario.

Docentes				
Campo	Tipo de Dato	Longitud	Restricciones	Descripción
IdDocente	INT	N/A	Primary Key, Identity	Identificador único y automático de cada docente en el sistema.
Nombre	NVARCHAR	100	Not Null	Nombre completo del docente.
Correo	NVARCHAR	100	Not Null, Único	Correo electrónico del docente, usado para contacto y verificación de identidad.
IdUsuario	INT	N/A	Foreign Key References Usuarios(IdUsuario)	Referencia al usuario en la tabla Usuarios, identificando al docente con su cuenta de usuario.

3.6 Seguridad Del Sistema

3.6.1 Claves de Acceso:

Gestión de Claves de Acceso: En el sistema, cada usuario accede a su cuenta mediante un correo electrónico y una contraseña. Este método garantiza una autenticación única basada en correo electrónico para los roles de Estudiante, Docente y Administrador.

1. Protocolos de Seguridad:

- **Autenticación de Usuarios:** La autenticación se realiza mediante cookies para gestionar la sesión de cada usuario, asegurando que solo los usuarios registrados puedan acceder a su perfil y realizar las funciones correspondientes a su rol. Esto refuerza la seguridad y restringe el acceso a funciones no autorizadas.

- **Control de Acceso Basado en Roles (RBAC):** Mediante el control de acceso basado en roles, el sistema garantiza que cada usuario acceda solo a las funciones necesarias para su rol, evitando que los estudiantes o docentes realicen funciones de administración o de gestión de cuentas.

2. Protección del Sistema y Sesiones de Usuario:

- **Expiración de Sesión:** Para proteger la sesión de los usuarios, se ha configurado una expiración de sesión que cerrará la sesión y redirigirá al login tras 5 minutos de inactividad, protegiendo así las cuentas en caso de acceso en dispositivos compartidos o públicos.
- **Cookies de Seguridad:** La sesión se maneja mediante cookies, las cuales administran la autenticación en el sistema. Como mejora futura, se recomienda incluir atributos de seguridad adicionales (como HttpOnly y Secure) en las cookies para reforzar la protección contra posibles ataques de secuestro de sesión.

3.6.2 Roles de Usuario:

Definición de Roles y Permisos:

1. **Estudiante:** Los estudiantes pueden inscribir materias, ver sus notas y modificar sus horarios de materias. Esta asignación restringe su acceso exclusivamente a funciones académicas, impidiendo cualquier acción administrativa.
2. **Docente:** Los docentes tienen acceso para visualizar la información de los estudiantes y calificar sus materias. Esto les permite realizar evaluaciones sin acceso a funciones administrativas o de inscripción de usuarios.
3. **Administrador:** El administrador cuenta con permisos de acceso completo para la creación de cuentas de usuarios y la inscripción de estudiantes. Este rol es responsable de configurar las cuentas de los estudiantes y docentes y de asignar correos institucionales según las políticas de la institución.

3.6.3 Privilegios

La seguridad del sistema está garantizada mediante una estructura de privilegios claramente definida, la cual asigna permisos específicos a cada rol de usuario. Esta separación asegura que cada grupo de usuarios pueda realizar únicamente las acciones necesarias para cumplir con sus funciones, minimizando riesgos de accesos indebidos o manipulaciones no autorizadas.

1. Estudiantes

- **Privilegios Asignados:**
 - ✓ Visualizar su perfil y datos personales.
 - ✓ Inscribirán materias disponibles según su plan de estudios.
 - ✓ Consultar las calificaciones registradas por los docentes.
- **Restricciones:**
 - No pueden modificar datos relacionados con calificaciones ni asignaciones de materias.
 - No tienen acceso a los datos de otros estudiantes, docentes ni a funciones administrativas.
- **Seguridad Implementada:**
 - ✓ El acceso a las funciones está vinculado al ID único del estudiante, garantizando que solo puedan ver su propia información.

2. Docentes

- **Privilegios Asignados:**
 - ✓ Consultar el listado de estudiantes inscritos en las materias que tienen asignadas.
 - ✓ Registrar, visualizar las calificaciones de sus estudiantes.
 - ✓ Acceso a su perfil y a la lista de materias impartidas.
- **Restricciones:**
 - No pueden inscribir estudiantes ni modificar datos personales o académicos ajenos a los estudiantes de sus materias.
 - No tienen acceso a la gestión de usuarios ni a funciones administrativas.

- **Seguridad Implementada:**

- ✓ Cada docente está vinculado únicamente a las materias asignadas, restringiendo su acceso a otros grupos o cursos.

3. Administradores

- **Privilegios Asignados:**

- ✓ Gestión completa de usuarios, incluyendo el registro, actualización y eliminación de estudiantes, docentes y otros administradores.
- ✓ Asignación de materias a docentes.
- ✓ Visualización de todos los datos del sistema para supervisión y auditoría.

- **Restricciones:**

- No tienen acceso directo a modificar calificaciones de estudiantes ni a inscribir materias como estudiante o docente.

- **Seguridad Implementada:**

- ✓ El acceso administrativo está protegido con autenticación de nivel superior, lo que asegura que solo usuarios autorizados puedan operar como administradores.
- ✓ Medidas de Seguridad Adicionales

Autenticación Basada en Roles:

- ✓ Cada usuario debe iniciar sesión con credenciales verificadas, y el sistema asigna automáticamente los privilegios según su rol.
- ✓ Control de Acceso Basado en Sesiones:
- ✓ El sistema genera sesiones individuales para cada usuario, invalidándolas automáticamente después de un período de inactividad

Referencias

- lLibrary. (s.f.). *lLibrary*. Recuperado el 10 de Noviembre de 2024, de Plataforma Moodle:
<https://llibrary.co/article/plataforma-moodle-antecedentes-de-la-investigaci%C3%B3n.zkwx65mz>
- Dougiamas, M. (s.f.). *moodle*. Recuperado el 10 de Noviembre de 2024, de La historia de Moodle: <https://moodle.com/es/acerca-de/la-historia-de-moodle/>
- Estudiante, C. (23 de Febrero de 2022). *YouTube*. Recuperado el 29 de Agosto de 2024, de Crear LOGIN y REGISTRO de usuarios en ASP.NET MVC 5 y SQL SERVER:
<https://www.youtube.com/watch?app=desktop&v=RTc-AKtNlxk>
- Freeman, A. (2017). *Pro ASP.NET Core MVC 2* (Séptima ed.). Apress.
- Google. (s.f.). *Google*. Recuperado el 25 de Agosto de 2024, de como hacer una página de inscripción de alumnos en asp.net core c#:
https://www.google.com/search?q=como+hacer+una+p%C3%A1gina+de+inscripci%C3%B3n+de+alumnos+en+asp.net+core+c%23&oq=como+hacer+una+p%C3%A1gina+de+inscripci%C3%B3n+de+alumnos+en+asp.net+core+c%23&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTI0NTg4ajBqN6gCFLACAQ&client=
- Microsoft. (s.f.). *Microsoft Learn*. Recuperado el 04 de 10 de 2024, de Documentación de ASP.NET: <https://learn.microsoft.com/es-es/aspnet/core/?view=aspnetcore-8.0>
- moodle. (18 de Junio de 2014). *moodle*. Recuperado el 08 de 11 de 2024, de Antecedentes:
<https://docs.moodle.org/all/es/Antecedentes>
- moodle. (s.f.). *moodle*. Recuperado el 29 de Agosto de 2024, de Con Moodle LMS nunca dejarás de aprender: <https://moodle.com/es/customers/educacion-mas-alta/>
- Pearson. (s.f.). *Pearson*. Obtenido de Ejemplos de plataformas E-learning para la educación superior: <https://blog.pearsonlatam.com/educacion-del-futuro/ejemplos-de-plataformas-e-learning-para-la-educacion-superior>
- Robert C, M. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design* (Primera ed.). Pearson.

Software Project Management: A Real-World Guide to Success (Primera ed.). (2003). Pearson.

Tecnologers. (20 de Octubre de 2023). *YouTube*. Recuperado el 19 de Agosto de 2024, de Cómo crear un login y registro de usuarios con .NET Core y SQL Server:

<https://www.youtube.com/watch?v=WyD-StDno9I>

UNAM. (s.f.). Recuperado el 10 de Noviembre de 2024, de Una breve historia de Moodle:

<https://www.revista.unam.mx/vol.17/num8/art60/>

UNESCO. (28 de Mayo de 2019). *unideusto*. Recuperado el 30 de Agosto de 2024, de EDUCACIÓN SUPERIOR:

https://siteal.iiep.unesco.org/sites/default/files/sit_informe_pdfs/siteal_educacion_superior_20190525.pdf

Glosario

A

ASP.NET Core:

Un framework de desarrollo web creado por Microsoft que permite construir aplicaciones web modernas, escalables y seguras, tanto para la nube como para servidores locales.

Automatización:

Proceso en el cual tareas que normalmente requieren intervención humana son ejecutadas por un sistema de manera automática. Ejemplos incluyen la inscripción de estudiantes y la validación de prerequisites.

B

Base de datos:

Conjunto organizado de información almacenada y gestionada electrónicamente, donde se guarda toda la información de los estudiantes, cursos, horarios, etc.

C

Carga administrativa:

Trabajo que debe realizar el personal de la institución para gestionar procesos como la inscripción de estudiantes, validación de requisitos, entre otros.

Cupos:

Número limitado de plazas disponibles para que los estudiantes se inscriban en un curso específico. El sistema automatiza el control de estas plazas.

E

Escalabilidad:

La capacidad de un sistema o aplicación para manejar el incremento en la carga de trabajo (como más usuarios o transacciones) sin perder rendimiento ni sufrir fallos.

Errores humanos:

Fallos o equivocaciones cometidos por personas durante la realización de tareas manuales, como errores en la inscripción o en la gestión de datos.

Errores de registro:

Problemas que ocurren durante el proceso de inscripción, como inscripciones duplicadas o estudiantes registrados en clases incorrectas.

F

Factibilidad económica:

Evaluación de si los costos de desarrollar e implementar el proyecto están justificados por los beneficios obtenidos, como el ahorro en tiempo y recursos.

Factibilidad operativa:

Posibilidad de que el sistema propuesto resuelva los problemas actuales y sea integrado de manera efectiva en el flujo de trabajo de la institución.

Factibilidad técnica:

Capacidad de implementar un proyecto en función de las herramientas tecnológicas disponibles y la experiencia del equipo en dichas tecnologías.

Framework:

Es una plataforma o conjunto de herramientas de software diseñadas para facilitar el desarrollo de aplicaciones. ASP.NET Core es un ejemplo de framework.

G

Gestión académica:

Conjunto de procesos que una institución utiliza para organizar la información y administración de cursos, horarios, inscripciones, etc.

I

Interfaz de Usuario (UI):

Es el diseño y disposición visual con la que los usuarios interactúan en una aplicación. En este proyecto, se busca una interfaz más amigable y fácil de usar.

Interrupciones del servicio:

Períodos en los que el sistema no está disponible para los usuarios debido a fallos técnicos o sobrecargas.

P

Plan de mantenimiento:

Conjunto de acciones planificadas para garantizar que un sistema continúe funcionando correctamente a lo largo del tiempo, incluyendo actualizaciones y correcciones de errores.

Prerrequisitos:

Condiciones o materias que un estudiante debe haber aprobado antes de inscribirse en un curso específico.

Proceso de inscripción:

Conjunto de pasos que un estudiante debe seguir para registrarse en cursos o asignaturas de una institución educativa.

R

Razor Pages:

Una herramienta dentro de ASP.NET Core que facilita la creación de interfaces de usuario al combinar la lógica y el diseño en una estructura más sencilla y organizada.

S

Seguridad de la información:

Protección de los datos contra el acceso no autorizado, su uso indebido o su divulgación. En el sistema propuesto, se prioriza la protección de los datos de los estudiantes.

Servidor:

Un sistema informático que proporciona datos o servicios a otros ordenadores a través de una red. En este caso, los servidores manejan las solicitudes de inscripción de los estudiantes.

Sistema heredado:

Sistema de software antiguo que sigue en uso pero está desactualizado tecnológicamente. En este caso, el sistema de inscripción actual de la institución.

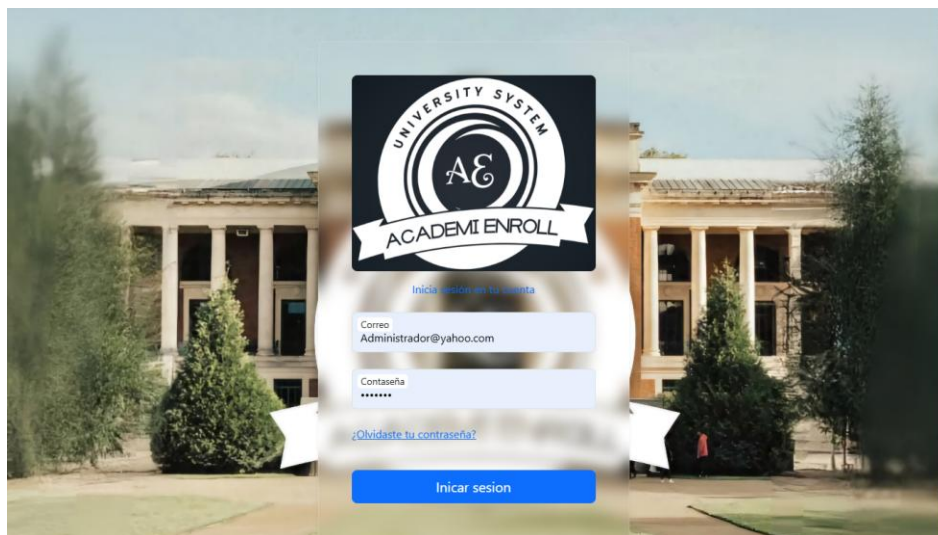
V

Validación de prerrequisitos:

Proceso que verifica si un estudiante cumple con los requisitos previos para inscribirse en una materia o curso. Esto evita que se registren en clases sin la preparación adecuada.

Anexos

1. Capturas Avance Proyecto 75%



a)

Al iniciar el aplicativo web Academi Enroll en su cuarta fase funcional, nos encontraremos con el login el cual está equipado con verificación de roles (Administrador, Docente y Administrador) completamente funcional. Los usuarios se validarán mediante un correo y contraseña registradas

La BD de Academi Enroll está equipada con un administrador por defecto para su testeo, el cual ingresa con el correo: admin@academienroll.com y contraseña: **admin123**



Bienvenido Administrador

b)



Al iniciar el sesión, nos encontraremos con una página vacía, la cual en la entrega final recibirá una UI, de momento es utilizada para validar el redireccionamiento de roles del login, por lo cual se ha colocado un mensaje de bienvenida junto con el rol que la cuenta ingresada posee en el sistema.

También, temporalmente la única forma de navegar por el sistema es mediante los enlaces del navbar, el cual es dinámico y cambia sus opciones según el rol de la cuenta que ha iniciado sesión.



c)

Listado Docentes: Al hacer clic en “Docentes” en el navbar, accesible únicamente como administrador, se redirigirá al CRUD de docentes, diseñado para dar mantenimiento a los docentes registrados.



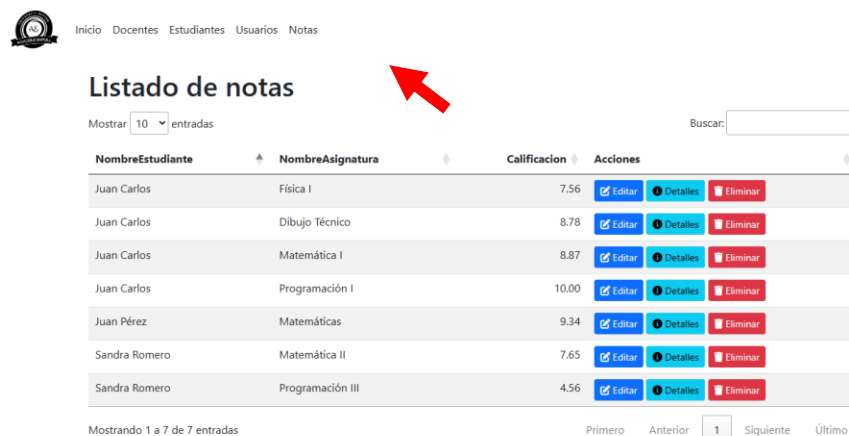
d)

Listado de Estudiantes: Al presionar “Estudiantes” en el navbar se redirigirá al administrador al CRUD de Estudiantes, útil por si un estudiante se da de baja de la universidad Academi Course o desea actualizar sus datos.



e)

Usuarios Registrados: Accesible desde la opción “Usuarios” del navbar, es una utilidad exclusiva del administrador, la cual le permite tener una vista global y acceso al CRUD de todos los usuarios registrados en el sistema, además de que desde “Usuarios” se puede agregar a un nuevo administrador. Diseñado para agilizar el mantenimiento de usuarios globales.



f)

Listado Notas: Diseñado principalmente para “Docentes”, el administrador tiene acceso a esta función pero a diferencia del docentes, el administrador no es capaz de cargar notas sino que en su lugar posee acceso a editar y eliminar notas.

Pensado para que sea el administrador el responsable de llevar a cabo la corrección de notas.



g)

© 2024 - AcademiEnroll - [Privacy](#)

Perfil: Una versión beta de perfil se encuentra disponible a partir de esta entrega, siendo una versión muy temprana que será refinada en la versión final.



h)

© 2024 - AcademiEnroll - [Privacy](#)

Vista Docente: Al iniciar sesión como docente, de momento solo se tiene acceso a Notas en el navbar.



Bienvenido Docente

i)

Vista Docente: Al iniciar sesión como docente, de momento solo se tiene acceso a Notas en el navbar.

[+ Cargar Nota](#)

Listado de notas

Mostrar 10 entradas

Buscar:

NombreEstudiante	NombreAsignatura	Calificacion	Acciones
Juan Carlos	Física I	7.56	Detalles
Juan Carlos	Dibujo Técnico	8.78	Detalles
Juan Carlos	Matemática I	8.87	Detalles
Juan Carlos	Programación I	10.00	Detalles
Juan Pérez	Matemáticas	9.34	Detalles
Sandra Romero	Matemática II	7.65	Detalles
Sandra Romero	Programación III	4.56	Detalles

Mostrando 1 a 7 de 7 entradas

Primero Anterior 1 Siguiente Último

j)

Listado Notas: Accesible mediante el navbar, al acceder a Listado de notas como docentes, solo se tiene acceso a Cargar Nota, el DataTable y detalles de los estudiantes.



Bienvenido Estudiante

k)

Vista Estudiante: Al ingresar al sistema como estudiante, en esta fase solo se tiene acceso a Consultar Notas mediante el navbar, para la entrega final se incluirá también la capacidad de inscribir materias.



Listado de notas

Mostrar entradasBuscar:

NombreEstudiante	NombreAsignatura	Calificacion	Acciones
Sandra Romero	Matemática II	7.65	Detalles
Sandra Romero	Programación III	4.56	Detalles

Mostrando 1 a 2 de 2 entradas

[Primero](#)
[Anterior](#)

[Siguiente](#)
[Último](#)

l)

Listado de Notas: Accesible mediante el navbar, para el estudiante, los datos del DataTable serán filtrados para mostrar únicamente las notas del estudiante que ha ingresado.