

***Facultad
de
Ciencias***

**HERRAMIENTA PARA LA GESTIÓN DEL
ACCESO DE ESTUDIANTES A LA
UNIVERSIDAD EN EL SOUCAN
(STUDENT UNIVERSITY ACCESS
MANAGEMENT TOOL AT SOUCAN)**

Trabajo de Fin de Grado
para acceder al

GRADO EN INGENIERÍA INFORMÁTICA

Autor: Yago Nava Bolado

Director: Diego García Saiz

Codirector: Javier Muñoz de la Iglesia

Junio – 2024

Resumen

El Servicio de Orientación Universitaria de Cantabria (SOUCAN) enfrenta dificultades al gestionar las adaptaciones extraordinarias para estudiantes de 2º de Bachillerato durante las pruebas de acceso a la universidad, debido a un proceso manual y complejo basado en correos electrónicos y un archivo de Excel.

Para resolver este problema, se propone desarrollar una aplicación web que automatice y simplifique la gestión de estas adaptaciones. El proyecto abarca varias fases: diseño del dominio y de la base de datos, desarrollo del *Back-End* y del *Front-End*, y la integración de todos estos componentes.

El *Back-End* manejará la lógica de negocio y la gestión segura de datos, mientras que el *Front-End* proporcionará una interfaz de usuario intuitiva para los centros educativos, permitiendo el acceso remoto al sistema. El SOUCAN utilizará “Microsoft Access” para administrar localmente la base de datos.

Las tecnologías utilizadas incluyen “SQL Server” para almacenamiento de datos, “Visual Studio” y C# con “.NET Framework” para la *API* (Interfaz de Programación de Aplicaciones), y “Microsoft Access” para la gestión interna. Todo el proyecto se publicará en “GitHub” como software libre, facilitando su accesibilidad y futura colaboración.

Abstract

The University Orientation Service of Cantabria (SOUCAN) faces challenges in managing extraordinary accommodations for students during university entrance exams due to a cumbersome and manual process based on emails and an Excel file.

To address this issue, the development of a web application is proposed to automate and streamline the management of these accommodations. The project involves several phases: designing the domain and database, developing the *Back-End* and *Front-End*, and integrating all these components.

The *Back-End* will handle business logic and secure data management, while the *Front-End* will provide an intuitive user interface for educational centers, enabling remote access to the system. SOUCAN will use Microsoft Access to manage the database locally.

The technologies used include SQL Server for data storage, Visual Studio and C# with .NET Framework for the *API*, and Microsoft Access for internal management. The entire project will be published on GitHub as open-source software, facilitating accessibility and future collaboration.

Índice

1.	Introducción y Objetivos	4
1.1.	Introducción.....	4
1.2.	Objetivos.....	6
2.	Metodología y Calendario.....	9
3.	Software Utilizado.....	13
4.	Requisitos Funcionales y No Funcionales	14
4.1.	Requisitos Funcionales:.....	14
4.2.	Requisitos no funcionales:.....	16
5.	Casos de Uso.....	18
6.	Arquitectura	23
6.1.	Patrón de Diseño	23
6.2.	Escalabilidad y Seguridad	24
6.2.1.	Escalabilidad.....	24
6.2.2.	Seguridad	25
7.	Desarrollo, Diseño e Implementación	26
7.1.	Modelo de Dominio	26
7.2.	Base de datos.....	28
7.2.1.	Modelo Entidad-Relación	28
7.2.2.	Tablas	29
7.2.3.	Procedimientos Almacenados	30
7.3.	Back-End	33
7.4.	Controlador.....	34
7.5.	Front-End	36
8.	Pruebas: Unitarias, Integración y Aceptación.....	43
8.1.	Pruebas Unitarias y Pruebas de Integración	43
8.2.	Pruebas de Aceptación	44
9.	Ejemplo de Uso.....	45
10.	Conclusiones	46
11.	Trabajo Futuro	48

1. Introducción y Objetivos

En este apartado se da lugar a la explicación exhaustiva acerca del desarrollo durante el proyecto, indicando cuál es la necesidad y de qué manera se ha cumplido su resolución. Además, se especifican los objetivos que se buscaban al inicio, junto con cómo se han resuelto.

1.1. Introducción

Gestionar las adaptaciones extraordinarias para estudiantes de 2º de Bachillerato durante la prueba de acceso a la universidad representa un desafío complejo para el Servicio de Orientación Universitaria de Cantabria (SOUCAN). Actualmente, este proceso se realiza a través de comunicaciones por correo electrónico con los centros educativos de Cantabria, que detallan la necesidad de una serie de adaptaciones para su alumnado dentro de un listado existente proporcionado por el SOUCAN, y adjuntan todos los documentos solicitados. Estos documentos se registran en un extenso archivo de Excel, cuya lectura y edición resulta muy complicada debido a su dimensión.

Para abordar eficientemente esta necesidad de gestionar esta gestión de estudiantes con necesidades extraordinarias para la realización de la EBAU (Evaluación de Bachillerato para el Acceso a la Universidad), se propuso el desarrollo integral de una aplicación web, es decir, que su elaboración involucre todos los elementos que se incluyen en la construcción de una aplicación software. Esta herramienta facilitará y automatizará la gestión de tareas tanto al departamento de la Universidad de Cantabria como a los centros educativos de la región en la prestación de este servicio.

El proceso de desarrollo se ha dividido en varias etapas clave, comenzando por el diseño del dominio y los objetos que requiere la aplicación. Esto ha sido crucial ya que implica la identificación y definición de los conceptos esenciales del sistema, así como las relaciones entre ellos. Para realizar este proceso, se ha utilizado la herramienta “Umllet”, la cual proporciona una manera gráfica de representar el diagrama UML a crear.

Posteriormente, se ha procedido con el diseño de la base de datos. Esta fase ha implicado definir la estructura de la base de datos, incluyendo las tablas, relaciones y restricciones necesarias para almacenar de manera eficiente y segura toda la información relevante para la gestión de adaptaciones para los estudiantes, vinculada con el dominio diseñado previamente.

El desarrollo del *Back-End* ha sido fundamental para el correcto funcionamiento del sistema. La lógica de negocio se ha desarrollado a través de una serie de procedimientos almacenados en la base de datos, lo que permitirá un uso seguro y eficiente para el almacenamiento y gestión de datos, permitiendo realizar operaciones de recuperación, inserción, modificación y eliminación de datos. Posteriormente, se ha establecido la conexión de la *API* con la base de datos a través del mapeado de estos procedimientos, implementando los mecanismos necesarios para que la aplicación pueda acceder y manipular la información y relacionarse con los objetos manejados por la interfaz.

En cuanto al desarrollo del *Front-End*, se ha llevado a cabo la creación de la interfaz gráfica de usuario, que será la parte visible y accesible de la aplicación para los usuarios finales. Este proceso se ha realizado mediante el uso de vistas y controladores para obtener la apariencia y funcionalidades deseadas. La interfaz se ha diseñado de manera intuitiva, con el objetivo de facilitar la interacción de los usuarios con el sistema y proporcionar una experiencia satisfactoria. Es importante destacar que esta funcionalidad estará disponible exclusivamente para los centros educativos, permitiendo el uso remoto del servicio. Por otro lado, para el departamento del SOUCAN, se ha utilizado la herramienta "Microsoft Access.". Esto se debe a que permite realizar cambios dentro de la base de datos de manera local y proporciona una funcionalidad de gestión completa, siendo administrada desde las instalaciones de la Universidad de Cantabria.

Para poner en contexto la necesidad de este proyecto, es importante destacar las principales dificultades que se enfrentan actualmente en la gestión de la EBAU y la comunicación entre SOUCAN y centros educativos:

- Comunicación: La comunicación entre SOUCAN y los centros se realiza exclusivamente por correo electrónico, lo que genera malentendidos y falta de claridad.
- Asignaturas EBAU: Los centros consultan la página web de la universidad para conocer las asignaturas disponibles para la EBAU, lo que implica un proceso manual y propenso a errores.
- Adaptaciones para Diagnósticos: La gestión de adaptaciones para estudiantes con necesidades específicas es tediosa y compleja, realizada de forma manual.
- Asignación de Sedes: Las ubicaciones para la EBAU se comunican por correo electrónico, requiriendo que los centros documenten esta información para evitar extravíos.
- Documentación de Estudiantes: La organización de documentos necesarios para adaptaciones es compleja, con riesgo de errores y múltiples comunicaciones para correcciones.
- Reuniones sobre Estudiantes: La documentación de reuniones es difícil de gestionar y localizar debido al alto volumen de estudiantes.
- Persistencia de Datos: La recopilación de datos de convocatorias se almacena de forma independiente, dificultando el análisis estadístico unificado.

Estos puntos serán documentados en detalle posteriormente en el apartado de objetivos, proporcionando una visión clara de las necesidades a tratar y las soluciones que se han aplicado dando una mejor solución.

Para mencionar las tecnologías utilizadas, se ha empleado "SQL Server" para el diseño del almacenamiento de datos, "Visual Studio" junto con el lenguaje C# y el *framework* ".NET Framework" para el desarrollo de la API, y "Microsoft Access" para implementar las funcionalidades que serán utilizadas desde la Universidad de Cantabria.

Todo el proyecto ha sido recogido como Software libre en "GitHub", accesible a través del siguiente enlace: https://github.com/ynb568/AdaptacionesEBAU_SOUCAN

1.2. Objetivos

El objetivo principal por cumplir en este proyecto es facilitar y automatizar el proceso de gestión de adaptaciones necesarias para los estudiantes que desean realizar la EBAU, dentro del departamento del SOUCAN. Todo este procedimiento se puede dividir en varias etapas, las cuales serán realizadas de una manera más sencilla. A continuación, se realiza un contraste entre el proceso actual y el nuevo:

- **Contacto SOUCAN/Centros:**

- **En la actualidad:** La comunicación y las solicitudes se gestionan exclusivamente a través del correo electrónico, lo que puede generar malentendidos y falta de claridad entre las partes involucradas.
- **Con la implementación de la aplicación:** El uso del correo electrónico se limita únicamente al momento en que los centros solicitan un usuario para acceder al servicio. A partir de entonces, la aplicación proporciona una plataforma intuitiva y sencilla que facilita la comunicación y gestión de todas las solicitudes entre ambas partes.

- **Disponibilidad de Asignaturas:**

- **En la actualidad:** Los centros se informan sobre las asignaturas disponibles para que sus alumnos puedan realizar la EBAU consultando la página web de la Universidad. Luego, notifican al SOUCAN sobre las asignaturas para las cuales prevén que los estudiantes se presenten, con el fin de determinar qué aulas pueden requerir adaptaciones.
- **Con la implementación de la aplicación:** Se tiene la capacidad de gestionar las asignaturas que son visibles en el formulario de registro desde el apartado de “Access”. Esto permite mantener actualizado y estructurado el listado de asignaturas disponibles para cada convocatoria, dividido en las distintas fases con las que cuenta la EBAU. Como resultado, los centros pueden realizar el registro de asignaturas de manera más sencilla y automática, sin necesidad de consultar la página web de la Universidad.

- **Disponibilidad de Adaptaciones para diagnósticos:**

- **En la actualidad:** La organización de las adaptaciones disponibles para estudiantes con diagnósticos específicos se gestiona mediante la selección de las adaptaciones pertinentes para cada caso individual en función de la solicitud que se realice. Esto provoca que el proceso resulte tedioso y complejo, por lo que debería estar automatizado.
- **Con la implementación de la aplicación:** Se simplifica y acelera este proceso al estructurarlo, antes de abrir los plazos de registro, se establecen las adaptaciones disponibles para cada posible diagnóstico. Esto estandariza el proceso y permite que todos los estudiantes puedan realizar la solicitud de manera más intuitiva sin dar lugar a confusiones.

- **Asignación de Sedes de EBAU para centros:**

- **En la actualidad:** La universidad comunica a los centros la ubicación donde se llevará a cabo la EBAU mediante correo electrónico. Esto significa que los centros deben documentar esta información para evitar extravíos y la necesidad de solicitar recordatorios de esta.
- **Con la implementación de la aplicación:** Dentro del servicio, el SOUCAN tiene la capacidad de asignar o actualizar la sede de cada centro. Esto garantiza que los centros tengan acceso a esta información de manera rápida y sin riesgo de perderla. De esta manera, se elimina la necesidad de que los centros documenten activamente esta información, ya que pueden confiar en que estará disponible y actualizada en el servicio proporcionado.

- **Organización y Disponibilidad de Documentación sobre Estudiantes:**

- **En la actualidad:** La recopilación de los documentos necesarios para que un estudiante pueda solicitar las adaptaciones necesarias durante el examen se vuelve compleja debido al volumen de archivos almacenados y su distribución en varias carpetas. Esta organización de archivos hace que sea tedioso encontrar documentos específicos de estudiantes en concreto en caso de necesitar revisarlos. Además, existe el riesgo de que los centros adjunten archivos incorrectos, lo que requiere que el SOUCAN se ponga en contacto, solicite el cambio de documento, elimine el incorrecto y finalmente almacene el correcto.
- **Con la implementación de la aplicación:** Se ha incorporado una funcionalidad que, al rellenar el formulario, solicita el documento necesario, permitiendo que sea renombrado de manera estandarizada. Esto facilita la búsqueda del documento cuando sea necesario. Además, el SOUCAN tiene la capacidad de revisar la información de un estudiante registrado y visualizar sus documentos. En caso de ser necesario, puede validarlos o rechazarlos, lo que indica a los centros que deben realizar el cambio de archivo correspondiente a través del servicio.

- **Documentación acerca de Reuniones sobre Estudiantes:**

- **En la actualidad:** En ocasiones, la persona encargada de esta gestión en la universidad debe reunirse tanto con los estudiantes como con los coordinadores de los centros. Es necesario documentar estas reuniones, ya que proporcionan información relevante sobre los estudiantes que dependen de este servicio universitario. Sin embargo, esta forma de organización puede dificultar la localización de los archivos concluyentes debido al gran número de estudiantes involucrados.
- **Con la implementación de la aplicación:** A través del servicio, en la parte gestionada por el SOUCAN, es posible recopilar la información de las reuniones en texto plano. Esto facilita la ubicación de la documentación, ya que está directamente asociada al estudiante

correspondiente dentro de la base de datos. Además, mejora la eficiencia del almacenamiento, ya que, en lugar de guardar documentos completos, la información se almacena en formato de texto.

- **Persistencia de Datos:**

- **En la actualidad:** Toda la recopilación de las distintas convocatorias se almacena de forma independiente, lo que dificulta realizar un análisis estadístico unificado. Esta tarea se vuelve compleja y lenta, ya que requiere revisar múltiples documentos y apuntar los datos por separado para poder realizar la comparativa.
- **Con la implementación de la aplicación:** Esta tarea se simplifica considerablemente al utilizar el servicio, que cuenta con una sección dedicada a realizar operaciones con toda la información recopilada en la base de datos. Además, el registro de cada estudiante está vinculado a la convocatoria activa en ese momento, lo que facilita la comparación de datos. Esta sección está diseñada específicamente para ser implementada, aunque por el momento no se conoce qué tipo de contraste de datos se desea realizar.

2. Metodología y Calendario

En este apartado se detallan el enfoque y las etapas necesarias para el desarrollo del servicio. Se describirán las fases del ciclo de desarrollo, desde la planificación y el diseño hasta la implementación y las pruebas, así como la gestión de tiempos y recursos. Además, se presentará un informe con los hitos logrados en cada plazo, lo que permitirá un seguimiento estructurado del progreso, asegurando la consecución de los objetivos propuestos de manera eficiente y organizada.

En la Imagen 1, se adjunta su debido Diagrama de Gantt, que permite representar de manera gráfica cómo ha sido subdividido en el tiempo el desarrollo de las tareas:

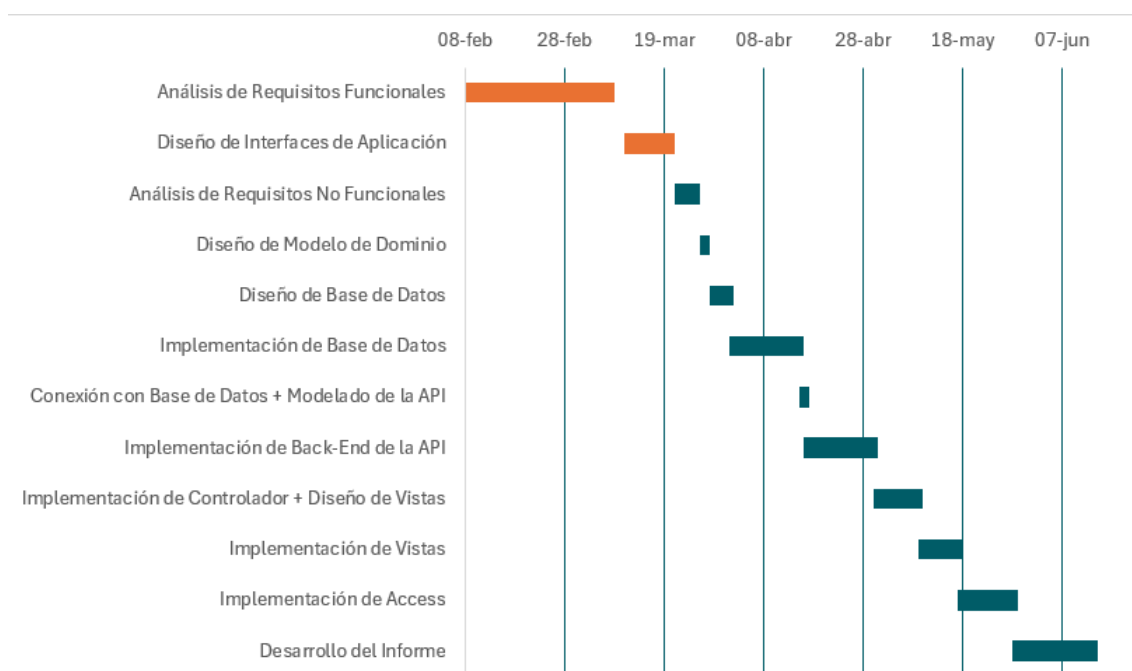


Imagen 1: Diagrama de Gantt - Calendario del Proyecto

Durante el desarrollo de este proyecto, se ha aplicado la metodología *Agile*, un enfoque iterativo y flexible que permite adaptarse rápidamente a los cambios y mejorar continuamente los resultados desarrollados. Esta metodología facilita volver a fases anteriores para revisar y corregir cualquier discrepancia en los objetos desarrollados, asegurando que el proyecto evolucione de manera eficiente y coherente con los requisitos y objetivos.

El diagrama del calendario del proyecto ilustra este enfoque iterativo. Las barras de color naranja representan las fases abordadas durante el período de prácticas, mientras que las barras verdes indican el desarrollo del Trabajo de Fin de Grado (TFG). Este diagrama visualiza cómo las diferentes etapas del proyecto se superponen y se revisan continuamente, siguiendo los principios de Agile.

Etapas 1: Análisis de Requisitos Funcionales

Fechas: 08/02/24 – 08/03/24

El comienzo del desarrollo parte del análisis de requisitos realizado durante el periodo de prácticas en el departamento del SOUCAN, a partir de este trabajo se cuenta con un

documento el cual figura. Este trabajo produjo un documento que recoge todos los requisitos funcionales requeridos para conseguir una funcionalidad completa, los cuales se determinaron a partir de reuniones con las personas encargadas de la gestión del servicio.

Etapas 2: Diseño de Interfaces de Aplicación

Fechas: 11/03/24-20/03/24

Durante esta etapa, se creó un mockup de las interfaces de usuario de la aplicación, así como la interconexión entre ellas. Este mockup visualizó la estructura y disposición de los elementos en cada vista, asegurando una navegación fluida y coherente. Se definieron las transiciones y conexiones entre las diferentes pantallas, proporcionando una base clara y detallada del diseño.

Etapas 3: Análisis de Requisitos No Funcionales

Fechas: 21/03/24 – 25/03/24

A través de varias reuniones con el responsable del departamento del Área de Servicios Web de la Universidad de Cantabria, se identificaron los requisitos no funcionales necesarios para el desarrollo del proyecto. En estas reuniones se definieron los atributos de calidad y las restricciones que debe cumplir el servicio, así como las tecnologías que permitirán abordar estos aspectos.

Fase 4: Diseño de Modelo de Dominio

Fechas: 26/03/24-27/03/24

Tras realizar el análisis de los requisitos, se procede a la fase de modelado del dominio. Esta fase tiene como objetivo representar todos los objetos y relaciones necesarios para implementar la funcionalidad deseada y cumplir con los requisitos establecidos.

Fase 5: Diseño de Base de Datos

Fechas: 28/03/24-01/04/24

Con base en el modelo de dominio previamente desarrollado, se procede a diseñar el almacenamiento de la información en la base de datos. Este modelado relacional se diseña con el fin de preservar la semántica y distribución de los objetos junto con la manera en la que se encuentran relacionados, con el objetivo de almacenar los datos de manera que se pueda acceder y manipular de forma lógica y eficiente.

Fase 6: Implementación de Base de Datos

Fechas: 01/04/24-15/04/24

Después de diseñar las entidades necesarias para el desarrollo, se procede a la conversión del modelo relacional a tablas. Para ello, se utiliza la herramienta "Microsoft SQL Server", que proporciona los recursos necesarios para almacenar todos los objetos manipulados en un servidor. Este proceso se divide en varias subetapas:

1. Creación y paso a tablas: Se utiliza el modelo relacional previo para definir la distribución de las tablas, así como el tipo de datos y las restricciones de sus campos.
2. Generación de objetos para testeo posterior: Tras la creación de las tablas, se generan una serie de objetos testables para que, durante el desarrollo de la *API*, se pueda probar cómo reaccionan al utilizar los distintos procedimientos almacenados.

3. Desarrollo de procedimientos almacenados: Con los objetos de la aplicación ya disponibles, se procede a la creación de los procedimientos necesarios para que el servicio funcione correctamente y realice las operaciones de lectura y escritura definidas en el análisis de requisitos. Además, se verifica el correcto funcionamiento de estos procedimientos utilizando los elementos de prueba previamente generados.

Fase 7: Conexión con Base de Datos + Modelado de la API

Fechas: 15/04/24-16/04/24

En esta etapa, se inicia la implementación de la *API* (Interfaz de Programación de Aplicaciones) utilizando "Visual Studio", que facilita la subdivisión en capas mediante el uso del modelo MVC (Modelo-Vista-Controlador). Dentro de esta estructura, se establece la conexión con el servidor de base de datos para realizar el mapeo de los objetos y los procedimientos almacenados.

Fase 8: Implementación de *Back-End* de la API

Fechas: 16/04/24-30/04/24

Se desarrolla la capa Modelo junto con sus subdivisiones (Capa Entidad, Capa Datos y Capa Negocio). Además, se crea un ejecutable de prueba para comprobar el correcto mapeo de los elementos. A continuación, se detalla la utilidad de las distintas partes:

1. Capa Entidad: En esta capa se define la estructura de los distintos objetos de la *API*, permitiendo representar los objetos de la base de datos y trabajar con ellos a través del servicio.
2. Capa Datos: Su función es mapear todas las operaciones que permiten leer y escribir los elementos de la base de datos. El desarrollo de esta sección es laborioso, ya que es necesario asegurar que todos los elementos almacenados se relacionen adecuadamente con los definidos en la Capa Entidad.
3. Capa Negocio: Se encarga de filtrar los métodos provenientes de la Capa de Datos, verificando que los parámetros sean correctos y permitiendo que se puede utilizar una operación de la Capa de Datos como varias suboperaciones en la Capa de Negocio. Además, en esta capa se crean las operaciones necesarias para complementar la funcionalidad del servicio.

Fase 9: Implementación de Controlador + Diseño de Vistas

Fechas: 30/04/24-09/05/24

En este periodo se completa la Capa Controlador y se estructura la Capa Vista.

1. Capa Controlador: Se encarga de conectar la manipulación de los datos a través de las vistas de la web con las operaciones implementadas en la Capa de Negocio. También establece cómo se relacionan y comunican las distintas vistas de la web.
2. Capa Vista: Únicamente se implementa la estructura y la relación entre las distintas vistas, utilizando el *mock* de interfaces desarrollado en el periodo de prácticas. Posteriormente se diseñará e implementará las mismas.

Fase 10: Implementación de Vistas

Fechas: 09/05/24-17/05/24

Entre estas fechas, una vez finalizado el desarrollo del *Back-End* de la aplicación, se procede a trabajar en el *Front-End*, es decir, a estructurar el apartado visual disponible en la interfaz del usuario. Para esta etapa, se utilizan las vistas generadas previamente junto con el controlador. Esta tarea es bastante compleja, ya que requiere un estudio previo sobre cómo implementar y combinar “HTML” con el *framework* seleccionado (.NET Framework).

Fase 11: Implementación de Access

Fechas: 17/05/24-28/05/24

Tras completar el desarrollo del servicio, se implementa la sección de “Access” para su uso local por los empleados encargados de la gestión del SOUCAN. Antes de esta implementación, se realizan dos reuniones con el encargado del Área de Servicios Web de la universidad para comprender el uso de este entorno de desarrollo. Posteriormente, se implementan todas las funcionalidades solicitadas y se verifica su correcto funcionamiento mediante pruebas de manipulación de datos, asegurando su persistencia en la base de datos.

Fase 12: Desarrollo de Informe

Fechas: 28/05/24-14/06/24

Finalmente, en la última fase del proyecto, se procedió al desarrollo del informe del Trabajo de Fin de Grado (TFG). Esta fase fue crucial para documentar todo el trabajo realizado, desde la concepción inicial del proyecto hasta su implementación y evaluación final.

3. Software Utilizado

En el desarrollo de esta aplicación se han utilizado diversas herramientas y plataformas de software esenciales para alcanzar los objetivos del proyecto. Se detallan las tecnologías seleccionadas, explicando su elección y cómo cada una contribuye a la funcionalidad y calidad de la aplicación final. Esta sección se encuentra subdividida de acuerdo con las distintas partes de la aplicación:

1. Diseño de la aplicación:

- 1.1. **Umllet:** Permite realizar un diseño del modelo de dominio de una manera sencilla e intuitiva, contando con una plantilla con todos los elementos necesarios para esta tarea.

2. Base de datos:

- 2.1. **Microsoft SQL Server:** se utiliza para gestionar la base de datos de la aplicación. Se emplea para la creación de tablas, la inserción de datos de prueba y el desarrollo de procedimientos almacenados. Estas funciones son esenciales para estructurar, organizar y establecer restricciones para los datos, garantizar la integridad de la información y optimizar las operaciones de consulta y manipulación de datos dentro del sistema.

3. API (Interfaz de Programación de Aplicaciones):

- 3.1. **Microsoft Visual Studio:** empleado para la realización de la lógica de la aplicación, junto con el mapeado e interfaces. Se aplica el modelo MVC, el cual permite que se distribuya el desarrollo de la aplicación en capas, permitiendo así una implementación entendible y apta para cambios futuros.
- 3.2. **.NET Framework:** *framework* que se emplea para desarrollar la aplicación, dando una plataforma fiable y versátil para la construcción de aplicaciones seguras y eficientes. En este proyecto, facilita la integración de distintas funcionalidades, la gestión de la lógica de negocio y la creación de una interfaz de usuario.

4. Trabajo local con base de datos:

- 4.1. **Microsoft Access:** se utiliza para gestionar y almacenar datos a nivel local. En este proyecto, “Access” permite realizar tareas de gestión del SOUCAN de manera eficiente, proporcionando una interfaz para la manipulación de datos y garantizando su persistencia en la base de datos.

5. Almacenamiento en Nube:

- 5.1. **GitHub:** para el almacenamiento distribuido de los documentos y archivos resultantes del desarrollo se utiliza esta plataforma, la cual permite establecer el proyecto como una fuente de software libre y aporta la capacidad de realizar un control de versiones durante la implementación.

4. Requisitos Funcionales y No Funcionales

En este apartado se explican los comportamientos y características esenciales que guían el diseño y desarrollo del proyecto. Los requisitos funcionales describen las acciones específicas que el sistema debe realizar, mientras que los requisitos no funcionales se centran en atributos como la seguridad y el rendimiento. Proporciona una visión integral de los criterios que definen el funcionamiento y la calidad del sistema.

4.1. Requisitos Funcionales:

Este apartado no aplica al desarrollo del TFG, ya que los requisitos funcionales necesarios para el correcto desarrollo de la aplicación se recogieron durante el periodo de prácticas previo. Durante estas prácticas, se identificaron todas las necesidades funcionales del departamento del SOUCAN para esta tarea. Por lo tanto, estos detalles proceden del informe de prácticas curriculares y se aportan con el fin de contextualizar el desarrollo:

SOUCAN	Iniciar sesión como usuario de SOUCAN	Se permite acceder a las funcionalidades específicas que se aportan a la gestión por parte del SOUCAN.
	Registrar Centro Educativo	El proceso de registro de usuarios de los centros educativos que buscan acceder a este servicio es llevado a cabo por el SOUCAN. Antes de realizar el registro, el centro educativo debe enviar una solicitud de usuario a [dirección de correo electrónico] para que se le asigne un usuario. Una vez completado este paso, se enviará la información de registro correspondiente, permitiendo al centro educativo acceder con su usuario asignado.
	Establecer Plazos de Registro	El registro de estudiantes para dar paso a la validación de las adaptaciones que necesitan para la EBAU ha de estar limitado en una serie de plazos que serán establecidos/modificados a través de la aplicación.
	Establecer las asignaturas en las que finalmente se matricula el estudiante	Las asignaturas a las que un estudiante finalmente se matricula pueden diferir de las que había planeado inscribir al completar el formulario. Por lo tanto, el SOUCAN se encargará de determinarlas una vez que se haya realizado la matrícula para la prueba.

	Validar Solicitud de Adaptaciones	Es necesario disponer de la capacidad para especificar qué adaptaciones de cada estudiante son validadas por el SOUCAN. Esto posibilitará mostrar a los centros educativos las adaptaciones que estarán disponibles una vez que hayan sido revisadas y aprobadas o rechazadas.
	Validar Documentos	Se debe disponer de la capacidad de poder indicar qué documentos quedan pendientes por adjuntar por parte de los centros e indicar si los aportados son correctos respecto a lo que se solicita.
	Exportar Solicitud Validada	Se cuenta con la capacidad de poder exportar toda la información registrada de cada estudiante cuando sea necesario.
	Listar apuntes de Reuniones acerca de Estudiantes Análisis estadístico de Solicitudes	Se cuenta con un apartado para cada estudiante en el que se registra un listado de toda la información recogida en la reunión/reuniones que se realicen acerca de la solicitud de este.
	Modificar listado de adaptaciones, asignaturas y diagnósticos disponibles en el formulario	Las asignaturas a las que se puede presentar un estudiante en la prueba de la EBAU, las adaptaciones y diagnósticos disponibles pueden fluctuar con el paso de los años por lo que se permite realizar una modificación de lo mostrado para que puedan los centros educativos hacer sus selecciones en el formulario.
	Modificar listado de sedes y centros educativos aplicables a la EBAU de Cantabria	Las sedes y centros educativos disponibles en cada convocatoria de la EBAU varían con el paso de los años. Por lo tanto, es necesario poder modificar su disponibilidad de manera sencilla.
CENTROS EDUCATIVOS	Iniciar sesión como Centro Educativo	Se permite acceder a las funcionalidades específicas que se aportan al uso de la aplicación por parte de los Centros Educativos de Cantabria una vez han solicitado una cuenta al SOUCAN.
	Registrar Estudiante	En el proceso de registro de estudiante se solicitan los datos personales, las asignaturas que tienen previstas matricularse en la EBAU, las distintas adaptaciones que necesitan en función de

		los diagnósticos que poseen y los documentos que acrediten la necesidad de estas.
	Visualizar/Modificar estudiantes registrados	Se cuenta con la capacidad de visualizar a los estudiantes registrados por un centro educativo, permitiendo conocer el estado de sus solicitudes (validadas, rechazadas o pendientes de revisión). En caso necesario, se ofrece la opción de modificar los datos existentes antes de la revisión correspondiente.
	Modificar información de usuario	Los centros educativos tienen la autorización para actualizar determinada información de usuario, como la referente a su orientador, equipo directivo y dirección. No obstante, no se permite modificar datos como el nombre, correo del centro ni la sede, ya que esta información está gestionada por el SOUCAN.
	Cambiar contraseña	Los centros educativos tienen la opción de cambiar su contraseña inicial según prefieran. Pueden optar por modificarla al recibirla inicialmente del SOUCAN o realizar cambios periódicos según lo consideren necesario.

Tabla 1; Análisis de Requisitos Funcionales

4.2. Requisitos no funcionales:

En el desarrollo de aplicaciones, los requisitos no funcionales son esenciales para garantizar que el sistema no solo funcione correctamente, sino que también cumpla con expectativas de rendimiento, seguridad y usabilidad. A continuación, se detallan los principales requisitos no funcionales que se deben considerar para la implementación de esta aplicación:

Eficiencia	La aplicación debe operar de manera eficiente, optimizando el uso de recursos del sistema para asegurar un rendimiento adecuado, tanto en la manipulación de datos en “Microsoft Access” como en las interacciones con la API y “SQL Server”.
Escalabilidad	La estructura debe permitir futuras ampliaciones y mejoras sin impedir su funcionalidad actual. Esto incluye la capacidad de añadir nuevas tablas, procedimientos y funcionalidades. Debe estar diseñada de tal forma que se pueda implementar en un futuro el uso del servicio en remoto para las tareas aplicadas a través de “Access”.

Seguridad	Se deben implementar medidas de seguridad robustas para proteger los datos almacenados en “SQL Server” y accesibles a través de “Microsoft Access”. Esto incluye la autenticación a través de registro de usuarios y la encriptación de datos sensibles, como contraseñas.
Usabilidad	La interfaz de usuario desarrollada en el “.NET Framework” debe ser intuitiva y fácil de usar, facilitando la gestión por parte de los centros educativos. La integración con “Microsoft Access” debe permitir una manipulación de datos fluida y eficiente.
Mantenibilidad	El código y la estructura de la aplicación deben ser fácilmente mantenibles. Esto incluye una documentación clara y comprensible gracias al informe del TFG, así como la adopción de buenas prácticas de programación que faciliten futuras actualizaciones y correcciones gracias a el uso de interfaces.
Compatibilidad	La aplicación debe ser compatible con las versiones de software y hardware utilizadas por el departamento del SOUCAN, garantizando un funcionamiento sin problemas en el entorno operativo existente. Esto quedó concretado junto con el Departamento de Área Web de la universidad, el cual conoce los requisitos necesarios para que se cumpla esto en cada departamento.
Fiabilidad	La aplicación debe ser confiable, asegurando que todas las funcionalidades, desde la gestión de datos en “Microsoft Access” hasta las operaciones de la API a través de los procedimientos almacenados de “SQL Server”, funcionen correctamente y sin errores.

Tabla 2: Análisis de Requisitos No Funcionales

5. Casos de Uso

Para realizar la explicación de los casos de uso en el contexto del desarrollo del proyecto se realiza una estructuración en la que cada caso de uso describe las interacciones entre los usuarios y el sistema, detallando los actores, las condiciones, los flujos de operación y los requisitos específicos, proporcionando una visión clara del funcionamiento del sistema y su relación con los requisitos del proyecto.

1. Inicio Sesión Centro

- 1.1. Descripción: Permite a un centro registrado por el SOUCAN iniciar sesión para tener acceso al servicio.
- 1.2. Actores:
 - 1. Centro Educativo.
 - 2. Sistema de Autenticación.
- 1.3. Precondiciones:
 - 1. El centro ha de estar previamente registrado.
- 1.4. Flujo Principal:
 - 1. El centro accede a la pantalla de inicio de sesión.
 - 2. El centro introduce su correo y contraseña.
 - 3. El sistema verifica la existencia del usuario.
 - 4. El sistema da acceso al centro para el uso del servicio.
- 1.5. Flujos Alternativos:
 - 3a. El sistema no encuentra el centro con dicho correo y contraseña y muestra mensaje de error
 - 3b. El centro introduce datos válidos y vuelve al paso 3.
- 1.6. Postcondiciones:
 - 1. Se debe de conceder acceso al centro al servicio
 - 2. Únicamente se debe de mostrar los datos asociados al centro

2. Cambio de contraseña Centro:

- 2.1. Descripción: Permite a un centro realizar el cambio de contraseña de su cuenta personal dentro del servicio.
- 2.2. Actores:
 - 1. Centro Educativo.
 - 2. Sistema de Autenticación.
- 2.3. Precondiciones:
 - 1. El centro debe de haber iniciado sesión previamente.
- 2.4. Flujo Principal:
 - 1. El centro pulsa el botón “Cambiar Contraseña”.
 - 2. El centro introduce la contraseña previa y la nueva contraseña 2 veces.
 - 3. El sistema verifica la contraseña anterior y cambia a la nueva.
 - 4. El sistema redirige al usuario al inicio de sesión.
- 2.5. Flujos Alternativos:
 - 3a. El sistema detecta que la contraseña previa no es correcta y muestra mensaje de error.
 - 3b. El centro introduce la contraseña correcta y vuelve al paso 3.
- 2.6. Postcondiciones:
 - 1. Se debe de redirigir al centro al inicio de sesión.
 - 2. El centro debe poder acceder al servicio con su nueva contraseña.
 - 3. El centro no debe poder acceder al servicio con la antigua contraseña.

3. Editar información Centro:

- 3.1. Descripción: Permite a un centro realizar el cambio de los datos asociados a su cuenta personal.
- 3.2. Actores:
 1. Centro Educativo.
 2. Sistema de Gestión de Usuarios.
- 3.3. Precondiciones:
 1. El centro debe de haber iniciado sesión previamente.
- 3.4. Flujo Principal:
 1. El centro pulsa el botón “Editar Información Centro”.
 2. El centro modifica los campos que se le permite editar acerca de su cuenta.
 3. El usuario pulsa el botón “Guardar Cambios”.
 4. El sistema verifica la correctitud de las entradas y cambia los datos del centro.
 5. El sistema redirige al centro al menú principal
- 3.5. Flujos Alternativos:
 - 4a. El sistema detecta que los números de teléfono introducidos no tienen el formato correcto y el sistema muestra el mensaje de error con los campos incorrectos.
 - 4b. El centro introduce los números de teléfono correctamente y vuelve al paso 3.
- 3.6. Postcondiciones:
 1. Se debe de redirigir a la página principal.
 2. Los cambios de información acerca del centro deben persistir.

4. Registrar Estudiante Centro:

- 4.1. Descripción: Permite a un centro registrar a estudiantes con sus debidas adaptaciones.
- 4.2. Actores:
 1. Centro Educativo.
 2. Sistema de Gestión de Estudiantes.
- 4.3. Precondiciones:
 1. El centro debe de haber iniciado sesión previamente.
 2. La fecha actual debe de encontrarse dentro del plazo de registro activo.
- 4.4. Flujo Principal:
 1. El centro pulsa el botón “Registrar Estudiante”.
 2. El centro introduce los datos del estudiante, junto con sus asignaturas previstas para la EBAU, las adaptaciones necesarias y los documentos solicitados.
 3. El centro pulsa el botón “Registrar”.
 4. El sistema verifica la correctitud de las entradas y registra al estudiante.
 5. El sistema redirige al centro a el listado de estudiantes registrados.
- 4.5. Flujos Alternativos:
 - 3a. El sistema detecta que los datos introducidos son incorrectos o insuficientes y muestra un mensaje de error con todos los cambios necesarios
 - 3b. El centro introduce los cambios necesarios y vuelve al paso 3.
- 4.6. Postcondiciones:
 1. Se debe redirigir al listado de estudiantes del centro.
 2. Se debe establecer el estado del estudiante como “Pendiente”, lo cual indica que el SOUCAN no ha comprobado la correctitud de los datos todavía.

3. Se debe de permitir posteriormente al centro a acceder al estudiante registrado.
4. El estudiante debe de tener asociado el plazo de registro automáticamente.

5. Revisión datos estudiante:

- 5.1. Descripción: Permite a un centro visualizar el estado de las solicitudes de un estudiante.
- 5.2. Actores;
 1. Centro Educativo
- 5.3. Precondiciones:
 1. El centro debe de haber iniciado sesión previamente.
 2. El estudiante debe de estar registrado previamente (caso de uso: Registrar Estudiante Centro).
- 5.4. Flujo Principal:
 1. El centro pulsa el botón “Ver Estudiantes Registrados”.
 2. El centro pulsa el botón “Ver” de la fila asociada con el estudiante que desea revisar.
 3. El centro comprueba el estado general del estudiante junto con el estado de los documentos y adaptaciones registrados.
- 5.5. Postcondiciones:
 1. En el caso de tener apartados en estado “Rechazado”, el centro debe de modificar la solicitud del estudiante para una posterior revisión.

6. Edición de Estudiante Centro:

- 6.1. Descripción: Permite a un centro editar la solicitud de un estudiante.
- 6.2. Actores:
 1. Centro Educativo.
 2. Sistema de Gestión de Estudiantes
- 6.3. Precondiciones:
 1. El centro debe de haber iniciado sesión previamente.
 2. El estudiante no debe estar ya validado, ya que, en ese caso no se permite realizar la acción.
 3. La fecha actual debe de encontrarse dentro del plazo de registro activo.
- 6.4. Flujo Principal:
 1. El centro pulsa el botón “Ver Estudiantes Registrados”.
 2. El centro pulsa el botón “Editar” de la fila asociada con el estudiante que desea revisar.
 3. El centro modifica los datos de la solicitud del estudiante que se encuentren pendientes o rechazados y que se permitan editar.
 4. El centro pulsa el botón “Guardar cambios”.
 5. El sistema verifica la correctitud de los campos modificados y edita el estudiante.
 6. El sistema redirige al centro a el listado de estudiantes registrados.
- 6.5. Postcondiciones:
 1. Se debe redirigir al listado de estudiantes del centro.
 2. Se debe establecer el estado del estudiante como “Pendiente”, lo cual indica que el SOUCAN no ha comprobado la correctitud de los datos modificados todavía.
 3. El cambio de los datos modificados del estudiante debe persistir.

7. Edición Plazo de Registro:

- 7.1. Descripción: Permite al SOUCAN establecer el periodo de registro activo.
- 7.2. Actores:
 1. SOUCAN.
 2. Sistema de Gestión de Base de Datos (“Microsoft Access”).
- 7.3. Precondiciones:
 1. Debe de existir un plazo de registro activo previo que se ha de desactivar.
- 7.4. Flujo Principal:
 1. El SOUCAN accede a la aplicación de “Microsoft Access”.
 2. El SOUCAN pulsa el botón “Plazos Registro”
 3. El SOUCAN introduce un nuevo plazo de registro
 4. El sistema registra el plazo de registro activo en la base de datos.
- 7.5. Postcondiciones:
 1. El nuevo plazo ha de establecerse como activo automáticamente.
 2. El nuevo plazo ha de estar disponible en el servicio.

8. Modificación de Formulario

- 8.1. Descripción: Permite al SOUCAN realizar cambios de los datos disponibles a seleccionar en el formulario de registro de estudiantes.
- 8.2. Actores:
 1. SOUCAN.
 2. Sistema de Gestión de Base de Datos (“Microsoft Access”).
- 8.3. Precondiciones:
 1. Debe existir la opción de cambiar la lista de adaptaciones disponibles.
 2. Debe existir la opción de cambiar la lista de diagnósticos disponibles.
 3. Debe existir la opción de cambiar la lista de sedes disponibles.
 4. Debe existir la opción de cambiar la lista de asignaturas disponibles.
 5. Debe existir la opción de modificar las asociaciones entre diagnósticos y adaptaciones
- 8.4. Flujo Principal:
 1. El SOUCAN accede a la aplicación de “Microsoft Access”.
 2. El SOUCAN pulsa el botón “Edición Formularios”.
 3. El SOUCAN modifica los datos del formulario que desee.
 4. El sistema guarda los cambios realizados en la base de datos.
- 8.5. Postcondiciones:
 1. El formulario cambiado ha de estar disponibles en el servicio.
 2. Los campos que hayan sido eliminados del formulario no deben desaparecer si no marcarse como inactivos.

9. Acceder y revisar solicitud estudiante:

- 9.1. Descripción: Permite al SOUCAN validar o rechazar solicitudes de estudiantes.
- 9.2. Actores:
 1. SOUCAN.
 2. Sistema de Gestión de Base de Datos (“Microsoft Access”).
- 9.3. Precondiciones:
 1. Los centros han de registrar a los estudiantes junto con sus solicitudes previamente.
- 9.4. Flujo Principal:
 1. El SOUCAN accede a la aplicación “Microsoft Access”.
 2. El SOUCAN pulsa el botón “Información Centros Educativos”.

3. El SOUCAN se mueve entre registros hasta llegar al centro deseado.
 4. El SOUCAN pulsa la ventana Estudiantes Centro.
 5. El SOUCAN se mueve entre registros hasta llegar al estudiante deseado.
 6. El SOUCAN se mueve entre las subpestañas cambiando el campo “validado” de pendiente a validado o rechazado.
 7. El sistema guarda los cambios realizados en la base de datos
- 9.5. Postcondiciones:
1. Los centros han de poder visualizar los cambios de estado de los estudiantes desde el servicio.
 2. Si un estudiante está completamente validado, no se podrá modificar posteriormente.

6. Arquitectura

Esta sección tiene como objetivo proporcionar una visión clara y detallada de la arquitectura de la aplicación, destacando sus componentes fundamentales, sus interacciones y las tecnologías utilizadas. A través de diagramas y descripciones detalladas, se establecerá una base sólida para la implementación del sistema, garantizando que se cumplan los principios de escalabilidad, mantenibilidad y seguridad.

6.1. Patrón de Diseño

La arquitectura de la aplicación sigue el patrón de diseño MVC (Modelo-Vista-Controlador), que divide la aplicación en tres capas principales: Modelo, Controlador y Vista. Esta separación facilita la gestión del código, mejora la mantenibilidad y permite un desarrollo más organizado. En la Imagen 2, referenciada en la bibliografía, se puede comprender de una manera más visual:

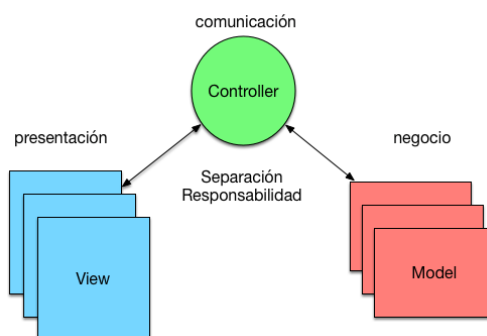


Imagen 2: Representación Modelo MVC

Modelo: Se subdivide en secciones que se encargan de realizar una tarea determinada para permitir la independencia de cada apartado.

- **Capa Entidad:** Define las entidades que representan los datos del dominio. Estas entidades son clases que encapsulan las propiedades y comportamientos de los objetos que se enlazan con las tablas de la base de datos con la *API*.
- **Capa Datos:** Maneja la comunicación con la base de datos. Incluye el acceso a los procedimientos almacenados que encapsulan toda la lógica de negocio permitiendo así aplicarlo al servicio a través del mapeado de los mismos.
- **Capa Negocio:** Contiene la comprobación de lógica de negocio de la aplicación. Esta capa procesa los parámetros introducidos para comprobar su correctitud y dar una respuesta correcta o mostrar un mensaje de error al emplearlo para aportar información al controlador. Además de esto, permite añadir funcionalidades que permiten el correcto funcionamiento de operaciones que no son obtenidas desde la base de datos.

Controlador: Actúa como intermediario entre el Modelo y la Vista. Los controladores reciben las entradas del usuario a través de la Vista, procesan estas entradas empleando los métodos de la Capa Negocio, y finalmente actualizan la Vista con los datos apropiados. Esta capa asegura que la lógica de la aplicación esté separada de la interfaz de usuario.

Vista: Responsable de la presentación de los datos. Las vistas generan la interfaz de usuario y presentan los datos al usuario final. Utiliza “HTML”, “CSS” y “Bootstrap” para asegurar una presentación atractiva y responsiva. Las vistas se actualizan basándose en los datos proporcionados por los controladores.

En conjunto, estas tres capas trabajan de manera cohesiva para proporcionar una aplicación bien estructurada y fácil de mantener. La implementación del patrón MVC con “.NET Framework” asegura una clara separación de responsabilidades y facilita la colaboración en el desarrollo de la aplicación.

6.2. Escalabilidad y Seguridad

La aplicación está diseñada teniendo en cuenta tanto la escalabilidad como la seguridad para garantizar un rendimiento óptimo y la protección de los datos en un entorno en constante evolución:

6.2.1. Escalabilidad

La aplicación ha sido diseñada de manera que permite ser fácilmente ampliable sin necesidad de modificar las implementaciones previas. Esta capacidad de escalabilidad se logra a través de prácticas que aseguran que el sistema pueda crecer y adaptarse a las nuevas necesidades sin comprometer la estabilidad y funcionalidad actual.

1. Arquitectura Modular:

- 1.1. **Modularidad:** La aplicación se encuentra dividida en módulos independientes, cada uno con una responsabilidad específica. Esto permite añadir nuevas funcionalidades simplemente creando nuevos módulos o extendiendo los existentes sin afectar otras partes del sistema.
- 1.2. **Interfaz Bien Definida:** Los módulos se comunican entre sí mediante interfaces bien definidas y documentadas, lo que facilita la integración de nuevos componentes. Esto aplica en concreto a la capa de negocio que conecta el *Back-End* con el *Front-End*.

2. Principio de Abierto/Cerrado:

- 2.1. **Abierto para Extensión, Cerrado para Modificación:** La arquitectura sigue el principio de diseño *SOLID* de estar abierta para la extensión, pero cerrada para la modificación. Esto significa que se pueden añadir nuevas funcionalidades mediante la extensión de clases o la adición de nuevos módulos sin la necesidad de alterar el código existente.

6.2.2. Seguridad

Para garantizar la seguridad de la aplicación, se han integrado varias medidas clave que protegen tanto los datos como el acceso a la información.

1. Autenticación:

- **Sistema de *Login*:** La aplicación requiere que los usuarios se autenticuen mediante un sistema de *login* antes de poder acceder a sus funcionalidades. Este proceso asegura que solo los usuarios autorizados puedan entrar al sistema.

2. Autorización:

- **Gestión de permisos:** Los permisos se gestionan cuidadosamente para asegurar que cada usuario únicamente pueda realizar acciones y acceder a los datos que le pertenecen. Esto evita el acceso no autorizado a información sensible.

3. Prevención de Inyecciones SQL:

- **Consultas Parametrizadas:** Todas las interacciones con la base de datos se realizan utilizando consultas parametrizadas. Esto asegura que los datos ingresados por los usuarios se manejen de manera segura, evitando “inyecciones SQL”.

4. Encriptación de Datos:

- **Datos en Reposo:** Los datos sensibles almacenados en la base de datos, como las contraseñas de los usuarios, se encriptan. Esto asegura que, incluso si la base de datos es comprometida, los datos permanezcan protegidos. De esta manera, no se puede acceder a la información de un usuario sin su contraseña, garantizando la seguridad de la información almacenada.

7. Desarrollo, Diseño e Implementación

En este apartado se ofrece una explicación detallada de los distintos componentes que conforman la estructura de desarrollo del proyecto y cómo estos se interrelacionan. Se describen las funcionalidades de cada componente, su papel dentro del sistema global, y se analiza la forma en que se comunican entre sí para asegurar un funcionamiento cohesivo y eficiente del software. Este análisis proporciona una visión integral de la arquitectura del proyecto y destaca la importancia de cada elemento en el éxito del desarrollo del sistema.

7.1. Modelo de Dominio

Como se muestra en la Imagen 3 (disponible en el directorio “DiagramasModelo” en “Github”, para poder ser más legible), Se ha desarrollado el modelo de dominio como la base a partir de la cual se construye todo el proyecto. Este modelo proporciona una representación estructurada de las entidades y relaciones clave del sistema de gestión de estudiantes, permitiendo un diseño coherente y una implementación efectiva del sistema. A continuación, se realiza una explicación de los objetos generados:

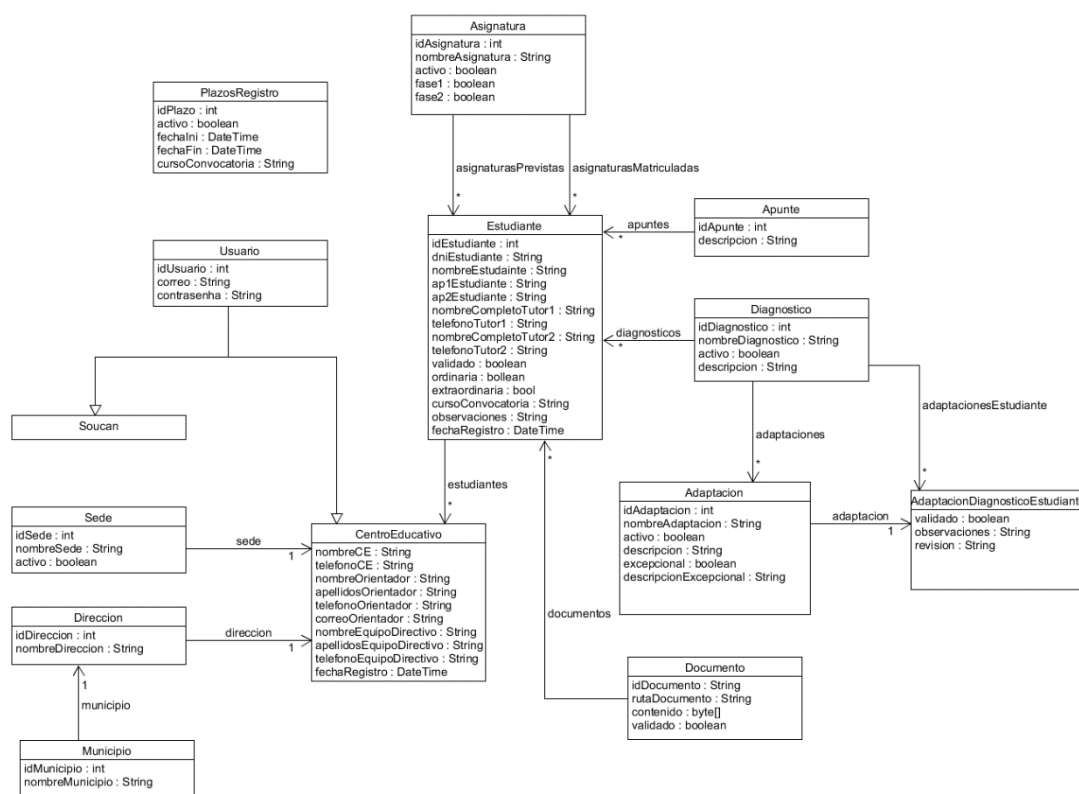


Imagen 3: Modelo de Dominio

- **Usuario:** Esta entidad almacena la información básica sobre los usuarios registrados en el sistema. A partir de ella se define una herencia que diferencia dos tipos principales de usuarios:
 - **Soucan:** Usuario asociado al sistema SOUCAN, creado para implementaciones futuras.
 - **CentroEducativo:** Usuario que representa a un centro educativo junto con la información asociada a dicho centro.

- **Sede:** Entidad que registra las sedes disponibles para la realización de la prueba de EBAU. Cada sede puede ser activada o desactivada según su disponibilidad en los formularios del sistema.
- **Dirección:** Contiene la dirección en texto plano donde se ubica el centro educativo.
- **Municipio:** Entidad que indica un municipio de una dirección. Permite a los centros educativos seleccionar su ubicación garantizando la normalización de los datos.
- **Estudiante:** Representa la información sobre los estudiantes de cada centro educativo a registrar, referenciando al centro educativo correspondiente. Además, se asocia con varias entidades para gestionar la información relacionada con solicitudes al SOUCAN.
- **Documento:** Define los documentos que los estudiantes deben aportar en los formularios para justificar la solicitud de adaptaciones. Almacena la ruta donde se pueden ubicar los documentos adjuntados.
- **Diagnóstico:** Permite definir los diagnósticos disponibles para la solicitud de adaptaciones. Estos diagnósticos pueden ser activados o desactivados para ser mostrados en el formulario.
- **Adaptacion:** Establece qué adaptaciones están disponibles asociadas a cada diagnóstico. Proporciona una descripción detallada de cómo cada adaptación será utilizada durante el examen, incluyendo si es excepcional o no, y permite su activación o desactivación en el formulario.
- **AdaptacionDiagnosticoEstudiante:** Tabla intermedia que asocia adaptaciones específicas con diagnósticos para cada estudiante. Esto permite gestionar qué adaptaciones específicas necesita cada estudiante basado en su diagnóstico.
- **Apunte:** Permite al recoger información en texto plano acerca de los documentos resultantes de reuniones con los orientadores de los centros sobre un estudiante específico.
- **Asignatura:** Describe las asignaturas disponibles en el formulario permitiendo indicar su previsión de presentación y las asignaturas a las que finalmente se matriculan. Puede activar o desactivar asignaturas para ajustarse a los cambios de cada convocatoria.
- **PlazosRegistro:** Describe los plazos activos para el registro de estudiantes. Cada estudiante tendrá un campo que indica el plazo en el que se ha inscrito.

7.2. Base de datos

La base de datos del sistema ha sido desarrollada utilizando “SQL Server”. Esta base de datos se compone de dos secciones principales: las tablas que almacenan los datos y los procedimientos almacenados que gestionan toda la lógica de negocio.

7.2.1. Modelo Entidad-Relación

El modelo relacional se ha diseñado con el fin de dar soporte al modelo de dominio de la aplicación. Este modelo, como se puede comprobar en la Imagen 4 (disponible en el directorio “DiagramasModelo” en “Github”, para poder ser más legible), organiza los datos en tablas interrelacionadas, permitiendo una gestión eficiente y estructurada de la información. Cada tabla representa una entidad del modelo de dominio y sus relaciones proporcionando una base sólida y escalable para el funcionamiento del sistema.

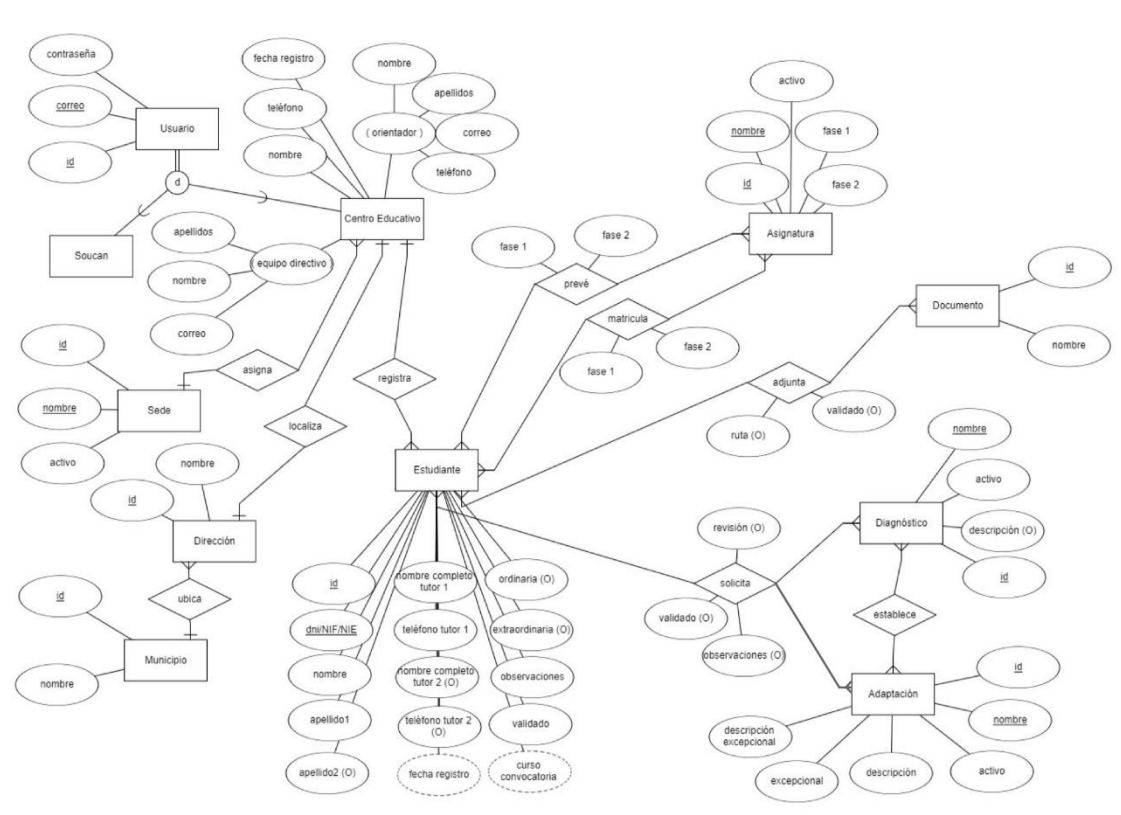


Imagen 4: Modelo Entidad-Relación

7.2.2. Tablas

Las tablas en “SQL Server” se utilizan para almacenar los datos de la aplicación de manera estructurada. Como se puede comprobar en la Imagen 5 (disponible en el directorio “DiagramasModelo” en “Github”, para poder ser más legible), cada tabla está diseñada con columnas que representan los atributos de las entidades que forman parte del modelo entidad-relación y se han implementado de tal manera que permitan recoger el funcionamiento deseado.

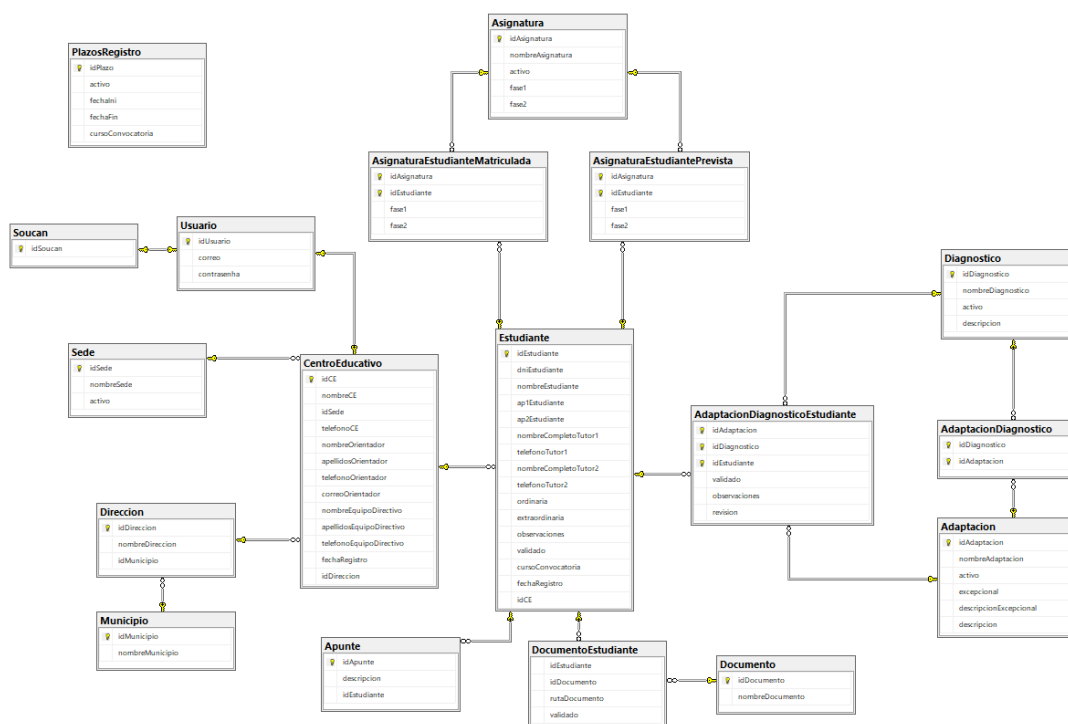


Imagen 5: Modelo Relacional

- **Usuario:** Almacena la información acerca de los usuarios registrados en el sistema. Está diseñada para ser ampliable y compatible con los usuarios del sistema SOUCAN. Actualmente, se extiende a "CentroEducativo", una entidad que almacena toda la información relacionada con los centros educativos registrados en el sistema. Además, se utilizan varias tablas auxiliares para la definición de atributos específicos:
 - **Dirección:** Contiene la dirección en la que se ubica el centro en forma de texto plano. Está diseñado de esta manera por si en algún momento se desea seccionar la misma en distintos apartados (Código Postal, Localidad, Calle, etc.).
 - **Municipio:** Almacena todos los municipios existentes en Cantabria, permitiendo a los centros educativos seleccionar su ubicación de entre los municipios disponibles. Esta tabla garantiza la normalización de los datos de ubicación.
 - **Sede:** Registra las distintas sedes disponibles para la realización de la prueba de EBAU. Incluye la capacidad de activar o desactivar sedes, lo cual permite controlar su visibilidad en los formularios del sistema. Esta funcionalidad es esencial para la gestión dinámica de las sedes.

- **Estudiante:** Recoge la información acerca de los estudiantes registrados a través del servicio por cada centro educativo, haciendo referencia al centro educativo al que pertenece cada estudiante. Además, tiene una asociación con una serie de tablas intermedias que permiten almacenar los datos relacionados con la solicitud realizada al SOUCAN, mencionadas a continuación, junto con una tabla auxiliar:
 - **Apunte:** Permite al SOUCAN recoger información en forma de texto plano acerca de los documentos resultantes de las reuniones que se realizan con los orientadores de los centros sobre el estudiante en concreto.
- **Asignatura:** Describe las asignaturas disponibles en el formulario para que los estudiantes puedan indicar su previsión de presentación, además de las asignaturas a las que se matriculan finalmente. Cuenta con la capacidad de activar o desactivar asignaturas, permitiendo ajustarse a los cambios que puedan ocurrir en cada convocatoria.
- **Documento:** Establece qué documentos deben aportar los estudiantes en los formularios, con el fin de justificar la solicitud de adaptaciones. Permite almacenar la ruta para que el SOUCAN conozca dónde ubicar los documentos adjuntados en el registro.
- **Diagnóstico:** Permite definir qué diagnósticos están disponibles para la solicitud. Además, cuenta con la posibilidad de activarlos o desactivarlos para mostrarlos o no en el formulario a rellenar por los centros.
- **Adaptación:** Aporta la capacidad de establecer qué adaptaciones están disponibles para cada diagnóstico existente mediante una tabla intermedia de asociación. La tabla de adaptaciones permite explicar el razonamiento y la descripción de cómo cada adaptación servirá durante el examen, y especificar si es excepcional o no, junto con su descripción excepcional. Además, cuenta con la posibilidad de activarlos o desactivarlos para mostrarlos o no en el formulario a rellenar por los centros.
- **Plazo Registro:** Tiene la funcionalidad de describir los plazos que se encuentran activos para el registro de estudiantes. Además, permite establecer en cada estudiante un campo que describe el plazo en el que se ha inscrito.

7.2.3. Procedimientos Almacenados

Los procedimientos almacenados en “SQL Server” gestionan la lógica de negocio del sistema, encapsulando las operaciones complejas que mejoran la eficiencia, seguridad y mantenibilidad del código. Todas las operaciones comprueban la correctitud de las salidas para evitar errores más adelante.

- **Lectura:** Esta sección engloba todas las operaciones de lectura que se realizan sobre la base de datos, es decir, las operaciones que recuperan datos de las tablas almacenadas en el sistema:
 - **sp_obtenPlazoRegistroActivo:** recupera el único plazo que se encuentra activo.
 - **sp_obtenMunicipioDireccion:** obtiene los datos del municipio que se encuentra asociado a una dirección.
 - **sp_obtenDireccionCentro:** obtiene los datos de la dirección que se encuentra asociada a un centro educativo.
 - **sp_obtenSedeCentro:** obtiene los datos de la sede que se encuentra asociada a un centro educativo.
 - **sp_obtenAdaptacionDiagnostico:** obtiene la adaptación que se encuentra asociada a un diagnóstico.
 - **sp_listaDiagnosticosEstudiante:** obtiene los diagnósticos que se encuentran asociados a un estudiante.
 - **sp_listaAdaptacionesDiagnosticoEstudiante:** obtiene las adaptaciones que se encuentran asociadas a un diagnóstico que a su vez se encuentran asociadas a un estudiante.
 - **sp_obtenAsignaturasPrevistasEstudiante:** obtiene las asignaturas previstas que se encuentran asociadas a un estudiante.
 - **sp_obtenAsignaturasMatriculadasEstudiante:** obtiene las asignaturas matriculadas que se encuentran asociadas a un estudiante.
 - **sp_listaApuntesEstudiante:** obtiene los apuntes que se encuentran asociados a un estudiante.
 - **sp_obtenCentro:** obtiene los datos de un centro en concreto.
 - **sp_obtenEstudianteCentro:** obtiene los datos de un estudiante asociado a un centro.
 - **sp_obtenNombreCentro:** obtiene el nombre de un centro
 - **sp_listaAdaptacionesDiagnostico:** obtiene las adaptaciones que se encuentran asociadas a un diagnóstico.
 - **sp_obtenDiagnostico:** obtiene los datos de un diagnóstico.
 - **sp_obtenAdaptacion:** Obtiene los datos de una adaptación.
 - **sp_listaEstudiantesCentro:** obtiene los estudiantes que se encuentran asociados a un centro educativo.
 - **sp_listaSedes:** obtiene todas las sedes disponibles.
 - **sp_listaMunicipios:** obtiene todos los municipios disponibles.
 - **sp_listaDiagnosticos:** obtiene todos los diagnósticos disponibles.
 - **sp_listaAdaptaciones:** obtiene todas las adaptaciones disponibles.
 - **sp_listaAsignaturas:** obtiene todas las asignaturas disponibles.
 - **sp_listaDocumentos:** obtiene todos los documentos que se encuentran disponibles.
 - **sp_listaDocumentosEstudiante:** obtiene todos los documentos que se encuentran asociados a un estudiante.
 - **sp_listaDiagnosticosEstudiante:** obtiene todos los diagnósticos que se encuentran asociados a un estudiante.

- **Inserción:** Esta sección engloba parte las operaciones de escritura que se realizan sobre la base de datos, en concreto las operaciones que insertan datos en las tablas existentes:
 - **sp_registraCentroEducativo:** registra un nuevo centro educativo en la base de datos junto con sus datos pasados como parámetros
 - **sp_registraEstudiante:** registra un nuevo estudiante en la base de datos junto con sus datos pasados como parámetros
 - **sp_registraAsignaturaPrevistaEstudiante:** registra una asignatura prevista dentro de las disponibles asociándolo a un estudiante junto con las fases que se pasan como parámetros
 - **sp_registraDocumentoEstudiante:** registra un documento asociándolo a un estudiante junto con la ruta de acceso al mismo pasada como parámetro
 - **sp_registraAdaptacionDiagnosticoEstudiante:** registra una adaptación junto con el diagnóstico al que se encuentra asociada en un estudiante, permitiendo añadir un campo de observaciones pasado como parámetro.
- **Eliminación:** Esta sección engloba todas las operaciones de eliminación que se realizan sobre la base de datos, es decir, las operaciones eliminan filas de las tablas existentes:
 - **sp_eliminaDiagnosticosEstudiante:** Elimina todos los diagnósticos de un estudiante junto con todas las adaptaciones que se encontraban asociadas al mismo.
 - **sp_eliminaAsignaturasPrevistasEstudiante:** Elimina todas las asignaturas previstas que se encontraban asociadas a un estudiante.
 - **sp_eliminaDocumentosEstudiante:** Elimina todos los documentos que se encontraban asociados a un estudiante, incluyendo las rutas especificadas.
- **Modificación:** Esta sección engloba la otra parte de las operaciones de escritura que se realizan sobre la base de datos, es decir, cambian valores de los atributos de las filas ya existentes en las tablas existentes:
 - **sp_modificaDatosCentro:** Modifica los campos que se pasan como parámetros de los datos de un centro.
 - **sp_modificaDatosEstudiante:** Modifica los campos que se pasan como parámetros de los datos de un estudiante.
- **Auxiliares:** Esta sección concreta cuenta con funcionalidades específicas que requiere la implementación del servicio:
 - **sp_cambiaContraseña:** Modifica el campo contraseña de un usuario, comprobando que la contraseña aportada original es correcta.
 - **sp_validarCentroEducativo:** Se encarga de comprobar las credenciales de correo y contraseña de un usuario.

- **No Aplicados:** esta sección se basa en varios procedimientos que fueron implementados previamente para el desarrollo del servicio, pero no han sido requeridos para lograr el alcance especificado. Estos se mantienen debido a que, para implementaciones futuras podrán ser de utilidad, debido a que incluyen parte de las funciones requeridas para ampliar el servicio web a que albergue el uso por parte del SOUCAN:
 - **sp_obtenApunteEstudiante:** obtiene todos los apuntes que se encuentran asociados a un estudiante.
 - **sp_gestionaEstadoDiagnostico:** permite cambiar el atributo “activo” a un diagnóstico.
 - **sp_gestionaEstadoAdaptacion:** permite cambiar el atributo “activo” a una adaptación.
 - **sp_gestionaEstadoAsignatura:** permite cambiar el atributo “activo” a una asignatura.
 - **sp_gestionaEstadoSede:** permite cambiar el atributo “activo” a una sede.
 - **sp_asociaAdaptacionADiagnostico:** permite establecer la relación entre un diagnóstico y una adaptación a través de una tabla intermedia.
 - **sp_anhadeDiagnostico:** registra un nuevo diagnóstico junto con sus atributos.
 - **sp_anhadeAdaptacion:** registra una nueva adaptación junto con sus atributos.
 - **sp_anhadeAsignatura:** registra una nueva asignatura junto con sus atributos.
 - **sp_anhadeApunteEstudiante:** permite registrar un nuevo apunte para un estudiante en concreto.
 - **sp_muestraInfoEstudiante:** recupera la información acerca de un estudiante
 - **sp_registraAsignaturaMatriculadaEstudiante:** permite registrar una asignatura junto con sus fases a un estudiante concreto.
 - **sp_listaAsignaturasPrevistasEstudianteF1:** obtiene todas las asignaturas previstas de la fase 1 asociadas a un estudiante.
 - **sp_listaAsignaturasPrevistasEstudianteF2:** obtiene todas las asignaturas previstas de la fase 2 asociadas a un estudiante.

7.3. Back-End

El *Back-End* de la aplicación ha sido desarrollado en “C#” utilizando el “.NET Framework” en el entorno de desarrollo “Visual Studio”. Este componente se encarga de manejar la lógica de negocio, aplicar las operaciones del servidor y gestionar la comunicación con la base de datos.

Utilizando el “.NET Framework”, el *Back-End* se beneficia de su robustez, escalabilidad y rendimiento eficiente. Este *framework* proporciona un amplio conjunto de herramientas y librerías que simplifican el desarrollo de aplicaciones. La elección de estos permite aprovechar características avanzadas del desarrollo de servicios web.

Además, el *Back-End* se integra con “SQL Server” a través de una conexión para gestionar los datos de manera eficiente y segura a través del mapeado de los procedimientos almacenados mencionados previamente. Esta integración asegura que la información almacenada sea consistente, disponible y accesible para las funcionalidades de la aplicación.

El uso de “Visual Studio” como entorno de desarrollo proporciona un entorno integrado y robusto para la escritura, depuración y prueba del código, contribuyendo notablemente a una mejor productividad y calidad en el desarrollo.

7.4. Controlador

El controlador actúa como el intermediario entre *Back-End* y *Front-End* de la aplicación, gestionando la comunicación y el flujo de datos entre ambas capas. A través de métodos definidos, se manejan las solicitudes del usuario desde la interfaz y se procesan con los métodos del *Back-End*.

Cuando un usuario interactúa con la interfaz de usuario, estas interacciones se envían al controlador en forma de solicitudes. El controlador recibe estas solicitudes, ejecuta la lógica de negocio correspondiente, y accede a los datos necesarios del modelo. Posteriormente, procesa y prepara la respuesta, la cual es enviada de vuelta al *Front-End* para ser presentada al usuario. Este diseño asegura que el *Front-End* permanezca desacoplado de la lógica de negocio y de acceso a datos, facilitando la mantenibilidad y escalabilidad de la aplicación.

A continuación, se explica la lógica de los métodos aplicados para permitir establecer esta comunicación:

- **LoginCE [HomeController]:** encargado de pasar la comunicación de inicio al controlador del centro. Es el único método ubicado en una clase controladora separada con fines de mantener la modularidad entre los usuarios del SOUCAN y los centros educativos. Únicamente existe para poder realizar una ampliación de la implementación en un futuro.
- **LoginCE [CentrosEducativosController]:** método web de tipo *GET*, encargado de recuperar la vista de inicio de sesión, y *POST*, encargado de validar las credenciales de inicio de sesión del usuario y de establecer una variable de sesión para asegurar que los siguientes pasos los realice el mismo usuario. En el caso de que no exista el centro se muestra un mensaje de error.
- **ControladorCentro:** método web de tipo *GET*, encargado de recuperar la vista que muestra el menú de opciones al que puede acceder un centro.
- **EdicionCentro:** método web de tipo *GET*, encargado de direccionar a la vista de edición del centro junto con la información que ya se encuentra registrada del mismo, y *POST*, encargado de recuperar los datos editables de la vista, guardarlos en la base de datos y redireccionar al menú del centro. En el caso de error, muestra un mensaje de error al modificar los datos y vuelve a recuperar la vista inicial.

- **CambioContraseña:** método web de tipo *POST*, encargado de dar la posibilidad a un usuario, a través de un modal (submenú de estilo *popup*), de poder modificar su contraseña de *login*. En el caso de que se modifique correctamente, se redirecciona al menú de inicio de sesión con el fin de mantener la consistencia de los datos. En el caso de error, se muestra un mensaje de error y se mantiene en el menú del centro.
- **RegistroEstudiante:** método web de tipo *GET*, encargado de direccionar a la vista de registro de estudiantes junto con los datos establecidos como activos para el formulario, y *POST*, que recupera los datos introducidos en el formulario, valida que sean correctos, los introduce en la base de datos y redirecciona a la lista de estudiantes del centro en concreto. En el caso de que los datos sean incorrectos, se mostrará un mensaje de error con los campos a corregir y se volverá al formulario vacío. Aparte, realiza la gestión de los nombres de los documentos introducidos con el fin de que se encuentren estandarizados y así se facilite su búsqueda futura. Además, cuenta con una serie de métodos auxiliares que permiten implementar la lógica y operaciones deseadas:
 - **InitFilesViewModel:** encargado de permitir realizar la carga de los documentos introducidos en el registro. Se crea con el fin de permitir realizar un cargado de datos desde la ruta del servidor que se establecerá posteriormente para guardar los archivos.
 - **SetDiagnostico:** encargado de establecer un diagnóstico para ser mandado a una vista parcial.
 - **SetAdaptacion:** encargado de establecer una adaptación asociada a un diagnóstico para ser mandada a una vista parcial.
 - **SetDiagnosticoAdaptacion:** encargado de recuperar un diagnóstico junto con sus adaptaciones para ser mandado a una vista parcial.
- **EstudiantesCentro:** método web de tipo *GET*, encargado de recuperar el listado de estudiantes asociados al centro y direccionar a la vista en la que se muestra.
- **InfoEstudiante:** método web de tipo *GET*, encargado de recuperar los datos registrados de un estudiante del centro y de direccionar a la vista en la que se muestra dicha información.
- **EdicionEstudiante:** método web de tipo *GET*, encargado de recuperar los datos de un estudiante concreto y direccionar a la vista en la que se permite realizar la edición de parte de sus campos, y *POST*, que recupera los datos modificados de dicho estudiante, asegurar que sean correctos y direccionar de vuelta a la vista de listado de estudiantes. En el caso de error, se mostrará un mensaje indicando qué campos son incorrectos.

7.5. Front-End

El *Front-End* de la aplicación ha sido desarrollado utilizando “.NET Framework” en combinación con “Bootstrap” para crear una interfaz de usuario interactiva y responsiva. La estructura del *Front-End* integra varias vistas interconectadas entre sí, proporcionando una experiencia de usuario coherente y fluida.

Las vistas están diseñadas utilizando “HTML 5” y se apoyan en “Bootstrap” para asegurar un diseño responsivo y comprensible. “Bootstrap” proporciona estilos y componentes predefinidos, lo que acelera el desarrollo y garantiza un uso consistente en toda la aplicación.

Para manejar la lógica de presentación y la interacción entre las vistas y los datos, se utilizan *ViewModels*. Estos permiten separar la lógica de la interfaz de usuario de la lógica de negocio, facilitando la gestión de los datos sobre las vistas. Esta arquitectura mejora la mantenibilidad y escalabilidad del código, permitiendo una mejor organización y reutilización de componentes.

El uso de “.NET Framework” en el *Front-End* permite una asociación con el *Back-End*, asegurando una comunicación eficiente y segura entre el cliente y el servidor. “Visual Studio” sirve como el entorno de desarrollo integrado, proporcionando herramientas avanzadas para el diseño, depuración y prueba del *Front-End*, lo que contribuye a una mayor productividad y calidad en el desarrollo de la interfaz de usuario.

A continuación, se presentan imágenes (todas disponibles en el directorio “ImagenesInterfaz” del repositorio de “Github” para que sean más legibles) de las distintas interfaces de usuario de la aplicación, acompañadas de una explicación detallada sobre cómo opera cada una. Estas imágenes ilustran las principales vistas y funcionalidades del *Front-End*, destacando la estructura y el flujo de navegación dentro de la aplicación:



Imagen 6: Menú de Inicio de Sesión

Como se puede comprobar en la Imagen 6, al acceder al servicio se presenta un menú principal. Este menú permite tanto al SOUCAN como a los centros educativos registrados acceder a sus respectivas secciones dedicadas dentro del sistema. Esta estructura facilita la navegación y el acceso a las funcionalidades específicas de cada tipo de usuario, mejorando así la usabilidad y eficiencia del servicio.

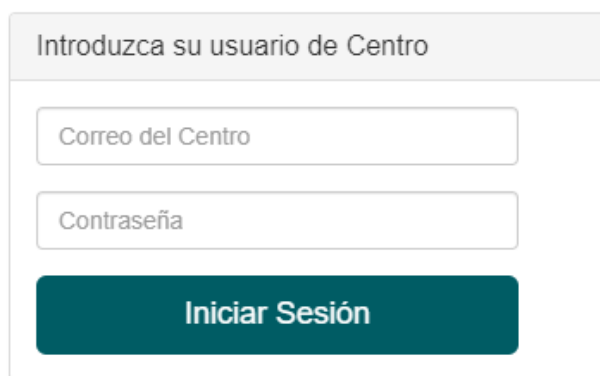
El formulario tiene un encabezado gris con el texto "Introduzca su usuario de Centro". Debajo hay dos campos de entrada: "Correo del Centro" y "Contraseña". El campo de contraseña tiene un ícono de ojo para alternar la visibilidad. En la parte inferior hay un botón de color verde oscuro con el texto "Iniciar Sesión" en blanco.

Imagen 7: Menú de Inicio de Sesión del Centro

Como se puede comprobar en la Imagen 7, se presenta un menú de inicio de sesión para que los centros educativos puedan acceder a sus datos utilizando sus credenciales. El apartado de contraseña muestra los caracteres como puntos para mantener la seguridad y proporciona un aviso sobre el uso de mayúsculas para facilitar la correcta introducción de la contraseña. Si las credenciales son incorrectas, aparece un mensaje de error destacado en rojo, indicando la necesidad de corregir la información ingresada.

Curso de la Convocatoria Actual:

Curso 2024-2025 1º Cuatrimestre 04/03/2024 0:00:00 10/01/2025 0:00:00

REGISTRAR ESTUDIANTE

VER ESTUDIANTES REGISTRADOS

EDITAR INFORMACIÓN CENTRO

CAMBIAR CONTRASEÑA

Imagen 8: Menú de Gestión de Centro

Como se puede comprobar en la Imagen 8, cada centro dispone de un menú de gestión donde se muestra el plazo de registro activo en ese momento, además de las distintas opciones disponibles para trabajar con el servicio. Es importante destacar que, si el momento de uso no coincide con el plazo activo, el botón para registrar estudiantes queda inhabilitado. Esto evita que los centros registren estudiantes cuando no está permitido, asegurando el cumplimiento de los plazos establecidos por el SOUCAN.

Universidad de Cantabria

Curso de la Convocatoria Actual:

Curso 2024-2025 1º Cuatrimestre

04/03/2024 0:00:00

10/01/2025 0:00:00

DNI/NIE/Pasaporte:

12345678B

Nombre Estudiante:

Manolo

Primer Apellido Estudiante:

Gómez

Segundo Apellido Estudiante:

Tartiere

Nombre Completo Primer Tutor:

Juan Gómez

Teléfono Primer Tutor:

942354678

Nombre Completo Segundo Tutor (Opcional):

Teléfono Segundo Tutor (Opcional):

Convocatoria:

☒ Ordinaria ☐ Extraordinaria

ASIGNATURAS PREVISTAS A PRESENTAR EN LA EBAU

FASE 1:

- ☒ Lengua Castellana y Literatura II
- ☐ Lengua Extranjera II: Inglés
- ☒ Lengua Extranjera II: Francés
- ☐ Lengua Extranjera II: Alemán
- ☐ Historia de España
- ☒ Historia de la Filosofía
- ☐ Dibujo Artístico II
- ☒ Análisis Musical II
- ☐ Artes Escénicas II
- ☐ Matemáticas II
- ☐ Matemáticas aplicadas a las Ciencias Sociales II
- ☐ Ciencias Generales
- ☐ Latín II
- ☐ Matemáticas aplicadas a las Ciencias Sociales II

FASE 2 (Opcional):

- ☐ Lengua Castellana y Literatura II
- ☒ Lengua Extranjera II: Inglés
- ☐ Lengua Extranjera II: Francés
- ☐ Lengua Extranjera II: Alemán
- ☐ Historia de España
- ☒ Historia de la Filosofía
- ☒ Dibujo Artístico II
- ☐ Análisis Musical II
- ☐ Artes Escénicas II

DIAGNÓSTICOS

Id Diagnóstico	Nombre Adaptación	Observaciones	Borrar Adaptación
2	Flexo		Borrar
1	Adaptación de textos: tamaño	observación	Borrar

Discapacidad Visual

Adaptación de textos: fuente

Descripción: Solicite la fuente deseada en el campo observaciones

Descripción excepcional: Solicitable en caso de tener una agudeza visual

Observaciones:

[Añadir](#)

DOCUMENTOS

Credenciales de Diagnóstico No se ha s...ún archivo Documentación del Estudiante No se ha s...ún archivo

Observaciones:

[Volver](#)

[Registrar](#)

Imagen 9: Formulario de Registro de Estudiante

Las tres imágenes anteriores se agrupan como Imagen 9, ya que forman parte de la misma funcionalidad y, de manera independiente, perderían su contexto y sentido. Esta agrupación facilita la comprensión del flujo completo de registro por parte de los centros educativos en el sistema.

Como se puede comprobar en la Imagen 9, se aporta un formulario en el que permite a un centro introducir los datos asociados a un estudiante, subdividido en varias secciones:

- **Datos Personales:** información básica personal requerida a modo de cuadros de texto.
- **Convocatoria:** Convocatoria a la que prevé inscribirse para realizar la EBAU, representada mediante *radiobuttons* que permiten seleccionar solo una opción.
- **Asignaturas Previstas:** Asignaturas a las que el estudiante quiere presentarse durante la EBAU. Se establecen mediante *checkboxes*, permitiendo seleccionar las deseadas, y solo se muestran las asignaturas que se encuentren activas en ese momento.
- **Adaptaciones Requeridas:** Utilizando elementos dinámicos, los centros pueden seleccionar los diagnósticos y las adaptaciones correspondientes, junto con un campo de observaciones. Estos datos se adjuntan a una tabla que permite revisar y eliminar filas si se desea realizar modificaciones. Es importante indicar que tanto los diagnósticos como las adaptaciones seleccionables son los que se encuentran activos.
- **Documentos Solicitados:** Permiten adjuntar archivos de tipo PDF que serán almacenados en el servidor con un nombre estandarizado para facilitar su gestión y búsqueda.

Nombre Estudiante	Validado	Ver	Editar
Roberto Bolado García	Pendiente	VER	EDITAR
Manolo Gómez Tartiere	Pendiente	VER	EDITAR

Volver

SOUCAN (Sistema de Orientación de la UC)

Imagen 10: Listado de Estudiantes de Centro

Como se puede comprobar en la Imagen 10, se proporciona a un centro la capacidad de visualizar los estudiantes que haya registrado hasta el momento, junto con el estado actual de la solicitud realizada. En el caso de que un estudiante haya sido validado, el botón de edición se desactiva para evitar incongruencias o ediciones indeseadas.

Esta funcionalidad permite a los centros educativos mantener un seguimiento claro del estado de las solicitudes de sus estudiantes, garantizando una gestión eficiente y precisa del proceso de registro en el sistema.

Universidad de Cantabria

Curso 2024-2025 1º Cuatrimestre

Estado de la Solicitud: **PENDIENTE**

Curso Convocatoria: Curso 2024-2025 1º Cuatrimestre
 Nombre Estudiante: Roberto
 Primer Apellido Estudiante: Bolado
 Segundo Apellido Estudiante: García
 Nombre Completo Primer Tutor: Manuel Bolado
 Teléfono Primer Tutor: 678555666
 Nombre Completo Segundo Tutor (Opcional): Maria Teresa García
 Teléfono Segundo Tutor (Opcional): 634666777
 Convocatoria: Ordinaria

ASIGNATURAS PREVISTAS A PRESENTAR EN LA EBAU

FASE 1:

Lengua Castellana y Literatura II

FASE 2 (Opcional):

DIAGNOSTICOS

Discapacidad Visual

Adaptaciones

Adaptación de textos: fuente Fuente de letra: Comic Sans Se requiere otra fuente **Rechazado**

DOCUMENTOS

Credenciales de Diagnóstico **Pendiente**

Observaciones: Estudiante de Segundo de Bachillerato que se presenta a la convocatoria 24-25

Volver

Imagen 11: Visualización Detallada de Estudiante

Como se puede comprobar en la Imagen 11, se proporciona la capacidad de visualizar la información registrada de cada estudiante, junto con el estado actual de la revisión de este. Esta interfaz tiene un propósito informativo para que los centros puedan realizar los cambios necesarios en el futuro en caso de que una adaptación o documento haya sido rechazado, permitiendo así su modificación.

Esta funcionalidad ofrece a los centros educativos una visión clara del estado de las solicitudes de sus estudiantes y les permite tomar acciones correctivas cuando sea necesario, asegurando un proceso de registro fluido y sin problemas.

Universidad de Cantabria Curso 2024-2025 1º Cuatrimestre

DNI/NIE/Pasaporte: 12345678A
Curso Convocatoria: Curso 2024-2025 1º Cuatrimestre
Nombre Estudiante: Roberto
Primer Apellido Estudiante: Bolado
Segundo Apellido Estudiante: García
Nombre Completo Primer Tutor: Manuel Bolado
Teléfono Primer Tutor: 678555666
Nombre Completo Segundo Tutor (Opcional): Maria Teresa García
Teléfono Segundo Tutor (Opcional): 634666777

ASIGNATURAS PREVISTAS A PRESENTAR EN LA EBAU

FASE 1:

- ☒ Lengua Castellana y Literatura II
- ☐ Lengua Extranjera II: Inglés
- ☐ Lengua Extranjera II: Francés
- ☒ Lengua Extranjera II: Alemán
- ☐ Historia de España
- ☒ Historia de la Filosofía
- ☐ Dibujo Artístico II
- ☒ Análisis Musical II
- ☐ Artes Escénicas II
- ☐ Matemáticas II
- ☐ Matemáticas aplicadas a las Ciencias Sociales II
- ☐ Ciencias Generales
- ☐ Latín II
- ☐ Matemáticas aplicadas a las Ciencias Sociales II

FASE 2 (Opcional):

- ☐ Lengua Castellana y Literatura II
- ☐ Lengua Extranjera II: Inglés

Convocatoria:
☒ Ordinaria ☐ Extraordinaria
DIAGNÓSTICOS

Id Diagnóstico	Nombre Adaptación	Observaciones	Borrar Adaptación
1	Adaptación de textos: fuente	Fuente de letra: Comic Sans	<button>Borrar</button>

Seleccione select

DOCUMENTOS

Credenciales de Diagnóstico Requisitos ...ia ed2.pdf Documentación del Estudiante No se ha s...ún archivo

Observaciones:

Estudiante de Segundo de Bachillerato que se presenta a la

Volver

Guardar cambios

Imagen 12: Edición de Estudiante

Las dos imágenes anteriores se agrupan como Imagen 12, ya que forman parte de la misma funcionalidad y, de manera independiente, perderían su contexto y sentido.

Como se puede observar en la Imagen 12, solo algunos campos registrados de un estudiante son modificables. Esta restricción se implementa para mantener la consistencia de los datos. En caso de necesitar modificar estos campos, el centro correspondiente debe solicitar el cambio al SOUCAN, quien realizará la modificación desde "Microsoft Access". Además de esto, los campos que son modificables tienen la misma funcionalidad que los explicados previamente en la Imagen 9.

Esta medida garantiza la integridad de los datos y asegura que cualquier modificación se realice de manera controlada, manteniendo así la coherencia de la información en el sistema.

Nombre del Centro: Universidad de Cantabria
 Correo del Centro: UC@gmail.com
 Sede Asignada: 1: EDIFICIO INTERFACULTATIVO
 Dirección del centro educativo Calle Honduras 19 (Alfoz de Lloredo)
 Teléfono del Centro:
 942222333

Información del Orientador

Nombre:
 Benito

Apellidos:
 Martínez Toral

Teléfono:
 678333444

Correo:
 benitomartinez@gmail.com

Información del Equipo Directivo

Nombre:
 Mercedes

Apellidos:
 García Sainz

Teléfono:
 612444555

Volver Guardar cambios


Imagen 13: Edición de Información del Centro

Como se puede comprobar en la Imagen 13, se permite a los centros realizar cambios acerca de la información asociada a su cuenta. Esto se lleva a cabo porque, con el paso del tiempo, es posible que cambien diversos aspectos, como los puestos del personal del centro o el número de teléfono de contacto. Permitir a los centros actualizar esta información garantiza que los datos sean precisos y estén actualizados, lo que contribuye a una gestión eficiente de los datos del sistema.

Cambiar contraseña

Contraseña Actual

Nueva contraseña

Confirmar Nueva Contraseña
 

Las mayúsculas están activadas.

Cerrar Guardar cambios

Imagen 14: Modal de Modificación de Contraseña

Como se puede ver en la Imagen 14, los centros tienen la capacidad de modificar la contraseña asociada a su cuenta. Inicialmente, el SOUCAN les proporciona una contraseña para el registro, pero una vez que han iniciado sesión, se les permite cambiarla por una de su elección. Además, los campos se muestran en formato de puntos, similar a la contraseña de inicio de sesión, y se indica que las mayúsculas están activadas para facilitar su introducción. Si los campos introducidos son incorrectos, se muestra un mensaje de error correspondiente. Este proceso asegura la seguridad de la cuenta del centro y brinda flexibilidad para gestionar su acceso al sistema.

8. Pruebas: Unitarias, Integración y Aceptación

En este apartado examinamos las estrategias esenciales para asegurar la calidad del sistema. Las pruebas unitarias se enfocan en unidades individuales de código, las de integración en su interacción y las de aceptación en la validación de los requisitos del usuario. Esta sección destaca el enfoque de prueba utilizado para garantizar un producto final confiable.

8.1. Pruebas Unitarias y Pruebas de Integración

Debido a la magnitud del proyecto, no se han realizado pruebas unitarias y de integración durante la fase de desarrollo inicial. La implementación de estas pruebas se ha pospuesto para futuras fases del proyecto. En el futuro, se planea incorporar un conjunto completo de pruebas unitarias para verificar la funcionalidad de cada componente individual del sistema, así como pruebas de integración para asegurar que los diferentes módulos interactúan correctamente y cumplen con los requisitos del sistema.

A pesar de la ausencia de estas pruebas, se han llevado a cabo diversas comprobaciones para asegurar el correcto funcionamiento del sistema. Estas incluyen:

1. Pruebas Funcionales:

- **Evaluación de funcionalidades clave:** Cada funcionalidad principal del sistema fue evaluada para asegurar que cumple con los requisitos especificados en el diseño del proyecto. Esto incluyó la revisión de todas las funcionalidades implementadas para verificar que funcionan según lo esperado.
- **Pruebas manuales:** Se realizaron pruebas manuales en diferentes escenarios para verificar que cada función del sistema opere correctamente en condiciones normales de uso. Estas pruebas ayudaron a identificar y corregir errores.

2. Pruebas de Usuario:

- **Simulación de casos reales:** Se llevaron a cabo pruebas siguiendo pasos necesarios para casos de uso reales. Esto implicó utilizar el sistema de la misma manera en que lo harían los usuarios finales en un entorno de producción.
- **Validación de usabilidad y funcionalidad:** Se revisó la interfaz de usuario y la interacción general con el sistema para asegurar que sea intuitiva y funcional. Estos casos reales permitieron identificar problemas prácticos que podrían no ser evidentes en pruebas controladas.

8.2. Pruebas de Aceptación

Las pruebas de aceptación son realizadas por el departamento del SOUCAN. Este proceso implica la verificación y validación del sistema para asegurarse de que cumple con todos los criterios y requisitos especificados. Durante estas pruebas, se evalúa el funcionamiento de la aplicación en un entorno real, verificando que todas las funcionalidades operan según lo esperado y que el sistema es capaz de gestionar las tareas para las que fue diseñado de manera eficiente y efectiva. El departamento del SOUCAN se ha encargado de establecer y confirmar que todos los criterios necesarios han sido cumplidos, garantizando que la aplicación satisfaga las necesidades de los usuarios finales. Además, el SOUCAN ha revisado estos criterios y ha comprobado que se cumplen antes de la puesta en producción del sistema.

9. Ejemplo de Uso

Accesible a través de la ruta: [https://youtu.be/ WVuLZ4mcBU](https://youtu.be/WVuLZ4mcBU)

En el vídeo creado para ilustrar el funcionamiento del sistema, se presenta un caso real en el que un centro educativo solicita al SOUCAN el uso del servicio y registra el primer estudiante de la convocatoria actual. El vídeo se divide en las siguientes secuencias:

1. **Primera toma:** El SOUCAN, desde “Microsoft Access” revisa los datos activos en los formularios y registra un nuevo usuario, junto con los datos del centro asociados a ese usuario. Esta toma muestra el proceso de creación de usuarios y la asociación de los datos relevantes del centro educativo, destacando la facilidad de configuración y administración de nuevos registros.
2. **Segunda toma:** El centro, desde el servicio, inicia sesión correctamente con el usuario proporcionado, junto con sus credenciales. A continuación, el usuario cambia la contraseña a su gusto, revisa la información del centro introducida y registra un estudiante en el sistema. Esta secuencia enfatiza la seguridad del sistema mediante el cambio de contraseña y demuestra cómo un centro educativo puede gestionar su información y registrar estudiantes de manera eficiente.
3. **Tercera toma:** El SOUCAN, desde “Microsoft Access” revisa los datos introducidos por el centro acerca del estudiante y valida la solicitud. Aquí se muestra el proceso de verificación y validación de los datos, asegurando que toda la información proporcionada cumple con los requisitos necesarios antes de aprobar la solicitud del estudiante.
4. **Cuarta toma:** El centro, desde el servicio, revisa el estado de la solicitud, comprobando que ha sido validada. En este punto, se muestra que el centro puede únicamente verificar los datos ya que no se permite editarlos una vez que la solicitud ha sido validada. Esto resalta la importancia de la integridad y la protección de los datos, asegurando que una vez validados, los datos no pueden ser alterados inadvertidamente.

Este vídeo proporciona una visión clara y detallada del flujo de trabajo dentro del sistema, demostrando la interacción entre el SOUCAN y los centros educativos y asegurando un manejo eficiente y seguro de la información de los estudiantes.

10. Conclusiones

El desarrollo de este proyecto ha permitido poner en práctica de una amplia gama de conocimientos adquiridos durante el transcurso del Grado en Ingeniería Informática, consolidando habilidades clave y ofreciendo una experiencia integral en el diseño y desarrollo de *software*.

Entre los principales logros y aprendizajes se destacan:

Análisis de requisitos para un proyecto: Se ha realizado un análisis detallado de requisitos, asegurando que todas las funcionalidades necesarias estén claramente definidas y alineadas con las necesidades de los usuarios y objetivos del proyecto.

Creación y gestión de bases de datos: Se ha implementado un modelo de dominio completo que estructura la información de manera clara y lógica. Este permitió la creación y gestión de bases de datos, lo cual ha sido fundamental para la gestión eficiente de la información. La experiencia ha demostrado la importancia de un diseño de base de datos bien estructurado para garantizar la integridad, accesibilidad y seguridad de los datos.

Aplicación del Modelo MVC: La aplicación del patrón de diseño Modelo-Vista-Controlador (MVC) ha sido crucial para el desarrollo modularizado de la aplicación. Este enfoque ha permitido una separación clara de responsabilidades, facilitando la escalabilidad y el mantenimiento del sistema. La implementación del “MVC” ha demostrado ser un modelo para gestionar la complejidad de las aplicaciones y mejorar la organización del código.

Implementación de Métodos Web: Se han aplicado métodos web para la implementación de la aplicación, utilizando tanto métodos *GET* como *POST*. Estos métodos han sido esenciales para la recuperación y envío de datos entre el cliente y el servidor, permitiendo una comunicación eficiente y segura.

Además, se han adquirido y aplicado diversos conceptos durante el desarrollo del proyecto, tales como:

Utilización del Framework .NET: Se ha aprovechado el “.NET Framework” para construir una aplicación robusta y escalable, utilizando sus diversas bibliotecas y herramientas integradas. Comprendiendo el potencial de este *framework* y la serie de facilidades que ofrece para el desarrollo correcto de un proyecto de *software*, aportando herramientas para la implementación tanto del *Back-End* como del *Front-End*.

Creación del Front-End de un servicio: El desarrollo de la interfaz de usuario ha sido un aspecto crucial, garantizando una experiencia de usuario intuitiva y eficiente. Para aplicar estilo a los elementos, se ha empleado “Bootstrap”, lo cual ha permitido un diseño coherente y adaptable a diferentes dispositivos. Además, se ha aprendido a conectar eficazmente las vistas con la lógica del *Back-End* a través del controlador, asegurando una interacción fluida y funcional entre la interfaz de usuario y las operaciones del sistema.

Uso de Microsoft Access: Se ha empleado “Microsoft Access” para la gestión de datos, reforzando la importancia de utilizar herramientas adecuadas para la administración y consulta de bases de datos a nivel local. Además, se ha aprendido a manejar esta herramienta de manera eficiente, mejorando las habilidades en la gestión y manipulación de datos.

Desarrollo basado en un caso real: La implementación de este proyecto en un contexto real ha proporcionado una valiosa experiencia práctica, enfrentando y resolviendo problemas del mundo real.

En cuanto a los objetivos logrados durante el desarrollo, se ha logrado mejorar notablemente la eficiencia y facilidad de uso en la gestión de estudiantes para la realización de la EBAU. Esto se ha conseguido mediante el almacenamiento de datos estandarizado, que no solo facilita la organización y acceso a la información, sino que también permite modificaciones futuras de manera sencilla.

Un logro significativo es que el sistema permite su utilización tanto por el SOUCAN como por los centros educativos, eliminando la necesidad de comunicarse exclusivamente por correo y reducir la carga de trabajo del personal del SOUCAN. Esto agiliza los trámites y mejora la eficiencia en la gestión de solicitudes y datos de los estudiantes.

Además, se ha implementado una interfaz de usuario intuitiva que optimiza la experiencia del usuario y reduce el tiempo necesario para completar tareas administrativas. La utilización de tecnologías modernas y la aplicación de buenas prácticas de desarrollo han contribuido a crear un sistema robusto y escalable, capaz de adaptarse a las necesidades cambiantes de los centros educativos y los estudiantes.

Asimismo, la integración de funciones avanzadas, como la gestión dinámica de plazos y la validación automática de datos, ha aumentado la precisión y fiabilidad del sistema, minimizando errores humanos y mejorando la calidad de los datos registrados. En resumen, el proyecto no solo cumple con los requisitos iniciales, sino que también establece una base sólida para futuras mejoras y expansiones.

En conclusión, este proyecto ha sido una oportunidad valiosa para aplicar y consolidar los conocimientos adquiridos durante el Grado en Ingeniería Informática. La experiencia ha permitido desarrollar una solución para la gestión de registros y solicitudes de estudiantes por parte de los centros, proporcionando una base sólida para futuras iniciativas y desafíos profesionales.

11. Trabajo Futuro

Dada la magnitud del proyecto, se ha optado por desarrollar un Producto Mínimo Viable (PMV) que incluye las funcionalidades esenciales para asegurar la operatividad solicitada del sistema. Este enfoque permite un lanzamiento rápido y la obtención de retroalimentación temprana de los usuarios.

En las siguientes fases, se implementarán pruebas, incluyendo pruebas unitarias, de integración, de rendimiento y de seguridad, para asegurar la calidad del sistema y evitar operaciones indebidas.

Además, se aplicará un estilo atractivo a la parte de *Front-End* a través de hojas de estilo “CSS” y la utilización de “Bootstrap” para mejorar la experiencia del usuario.

Asimismo, se integrará la parte actualmente implementada en “Microsoft Access”, que maneja las operaciones de SOUCAN, al servicio web.

Esta estrategia garantiza que la aplicación pueda ser ampliada y optimizada de manera continua, incorporando nuevas funcionalidades, mejorando el rendimiento y asegurando una presentación profesional y una gestión de operaciones unificada.

Bibliografía

SOUKAN (2024): Recuperado de <https://web.unican.es/unidades/soucan> el 20 de marzo de 2024.

EBAU (2024): Recuperado de <https://web.unican.es/admision/acceso-a-estudios-de-grado/evaluacion-de-bachillerato-para-el-acceso-a-la-universidad> el 27 de marzo de 2024.

UMLet (2024): Recuperado de <https://www.umlet.com/> el 27 de marzo de 2024.

Visual Studio (2024): Recuperado de <https://visualstudio.microsoft.com/es/> 15 de abril de 2024.

Bootstrap (2024): Recuperado de <https://getbootstrap.esdocu.com/> el 13 de mayo de 2024.

Microsoft Access (2024): Recuperado de <https://www.microsoft.com/es-es/microsoft-365/access> el 17 de mayo de 2024.

Imagen 2: Recuperado de <https://www.arquitecturajava.com/patron-mvc-arquitectura-cliente-vs-servidor/> el 10 de junio de 2024