



Universidad Tecnológica Nacional
FRRO

Tecnologías de desarrollo de software IDE

Informe Trabajo Práctico Academia

AÑO: 2021 - Comisión 3K7

Alumnos

Nombre	Legajo
Fonzo, Franco	46882
Tano, German	46873
Del Coro, Lara	46978
Poratti, Candela	47072
Colarte, Federico	47536

Docentes

Porta, Ezequiel
Joaquín, Andrés

Índice

Índice	2
Introducción	3
Tecnologías utilizadas	4
Arquitectura	6
Análisis y Diseño	8
Fragmentos de código	14
Integración con Azure DevOps	15

Introducción

El sistema Academia permite registrar y administrar actividades académicas tales como inscripciones, registro de notas, generación de informe, creación de cursos, ABMC de diferentes entidades, entre otras cosas.

El sistema puede ser utilizado por cualquier persona que posea un usuario con el rol de alumno, docente o administrador (dirección académica).

El mismo se ejecuta en web a través de Azure App Services, o en escritorio. Los datos pueden ser obtenidos tanto por medio de una conexión local, usando el motor de base de datos SQL Server, como por conexión remota a Azure SQL Server.

El diseño está unificado en ambas plataformas y una vez que un usuario ingresa, sólo puede realizar acciones permitidas según su rol.

Los Usuarios poseen un usuario y contraseña, una Persona asociada, y pueden ser habilitados o inhabilitados.

Las Personas pueden tomar el rol de Administrador, Alumno, o Docente, y tienen datos en común tales como Legajo, Nombre, Apellido, Dirección entre otros. Las Materias pertenecen a un Plan y tienen cantidad de horas de cursado semanales y totales. Los Planes son de una Especialidad.

La relación entre una Personas del tipo Docente y un Cursos, se llama DocenteCurso, y se la conoce como Dictado. La relación entre una Persona del tipo Alumno y un Curso, se llama AlumnoInscripción y se conoce como Inscripción.

Tecnologías utilizadas

- IDE: **Visual Studio - Versión 2019**
- Framework: **.NET Framework 4.7.2**
- Motor de Base de Datos: **Microsoft SQL Server / Azure SQL Server + SQL Database**

Al comenzar el desarrollo utilizamos Microsoft SQL Server como motor de base de datos de la aplicación de escritorio. Una vez desarrollada la capa de presentación web, e integrado el proyecto con Azure DevOps y Azure App Service, optamos por utilizar Azure SQL Server con SQL Database para que los datos puedan ser consultados tanto por instancia local, como remota.

- Versionado de Código: [GitHub](https://github.com)

Github nos permitió llevar el versionado del proyecto de una manera organizada. Además posibilitó y facilitó el trabajo en equipo, de manera que pudimos desarrollar al mismo tiempo e implementar nuevas partes de la aplicación gracias a la creación de distintas ramas y el posterior merge a la rama principal sin correr riesgos de arruinar el trabajo ya realizado.

- Acceso a datos: **Entity Framework**

El acceso a la base de datos a través de Entity Framework permite un código mucho más corto y limpio. Esto se ve reflejado en el proyecto Data.Database, el cual contiene en cada archivo adapter.cs breves líneas de código que permiten traer a la aplicación los datos de cada entidad.

- Integración y Distribución web: **Azure Devops (CI/CD)**

La integración y distribución continua que brinda Azure DevOps para la aplicación web fue implementada como integración del primer trabajo práctico realizado por el grupo. Esta tecnología agiliza la integración y distribución de la aplicación, ya que al realizar un commit en la rama de producción del repositorio de GitHub, Azure DevOps se encarga de manera automática de compilar la aplicación a través de pipelines, y enviar la nueva versión del proyecto a su App Service para que cualquier persona pueda acceder a través de su URL.

- Acceso a aplicación web: **Azure App service**

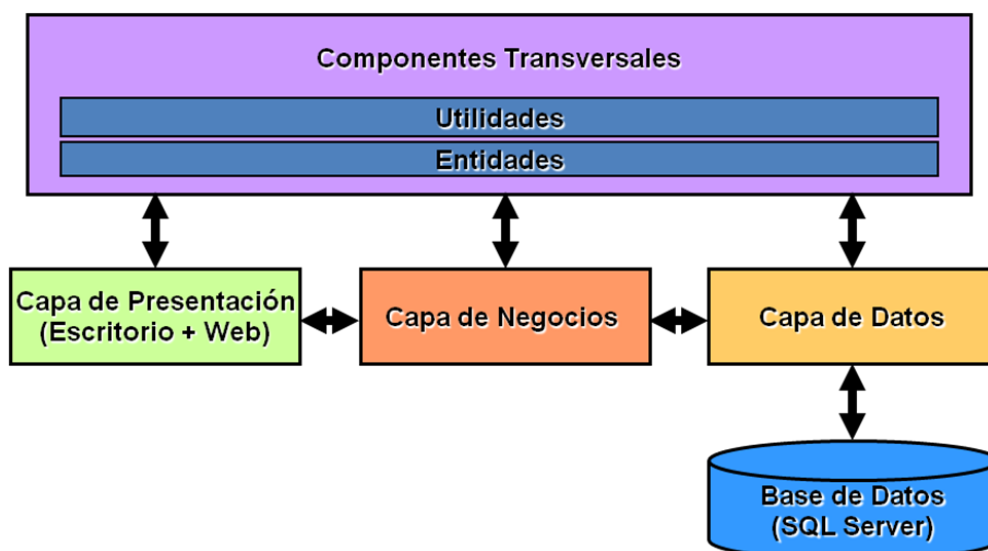
Azure App service, también implementada como integración del primer trabajo práctico, nos brindó la posibilidad de hospedar nuestra web ya creada para que la misma sea accesible desde cualquier navegador con acceso a internet.

Arquitectura

La solución implementa una arquitectura basada en capas.

Capas y proyectos de la solución:

Proyecto	Capa
Data.Database	Capa de Datos
Business.Logic	Capa de Negocios
UI.Web	Capa de Presentación: Web
UI.Desktop	Capa de Presentación: Escritorio
Business.Entities	Entidades
Util	Utilidades



Capa de Datos: El proyecto Data.Database centraliza la conexión a la base de datos utilizando el motor de base de datos Microsoft SQL Server, o Azure SQL Server. Para la consulta de los datos se utiliza Entity Framework.

Capa de Negocios: El proyecto Business.Logic es la capa intermediaria entre la capa de Presentación y la capa de Datos.

Capa de presentación

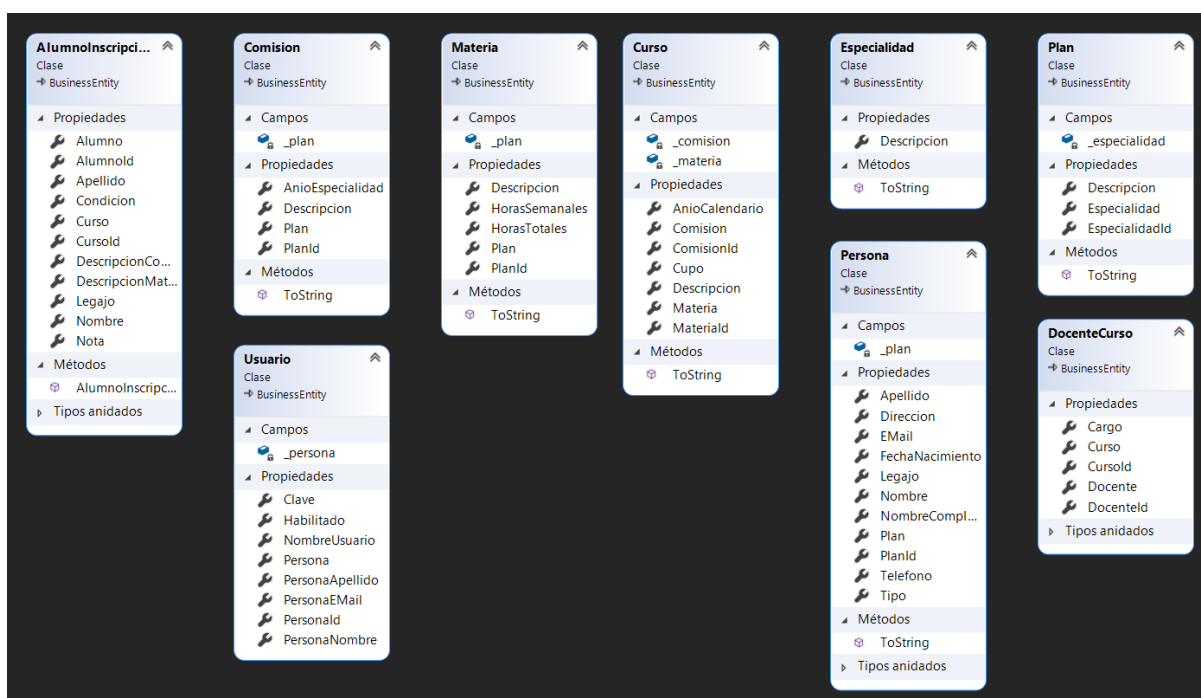
Web: El proyecto UI.Web utiliza WebForms con Master Page. El mismo contiene un Login que al iniciar sesión redirige al usuario a la página principal, con un menú de navegación.

Escritorio: El proyecto UI.Escritorio utiliza WinForms que inicia con un Login, que al iniciar sesión, redirige al usuario a la página principal con un menú.

Componentes Transversales

Entidades: El proyecto Business.Entities contiene la definición de cada entidad. Las mismas se implementan como Objetos.

Diagrama de navegación de clases



Análisis y Diseño

LOGIN

Escritorio:



A desktop application window titled "LOGIN" with a dark theme. The left side features a teal background with a circular logo containing a graduation cap and the word "ACADEMIA" below it. The right side is a dark gray panel with the title "LOGIN" at the top right. It contains two input fields labeled "Usuario" and "Contraseña", a blue button labeled "Iniciar sesión", and a link at the bottom that says "¿Has olvidado la contraseña?".

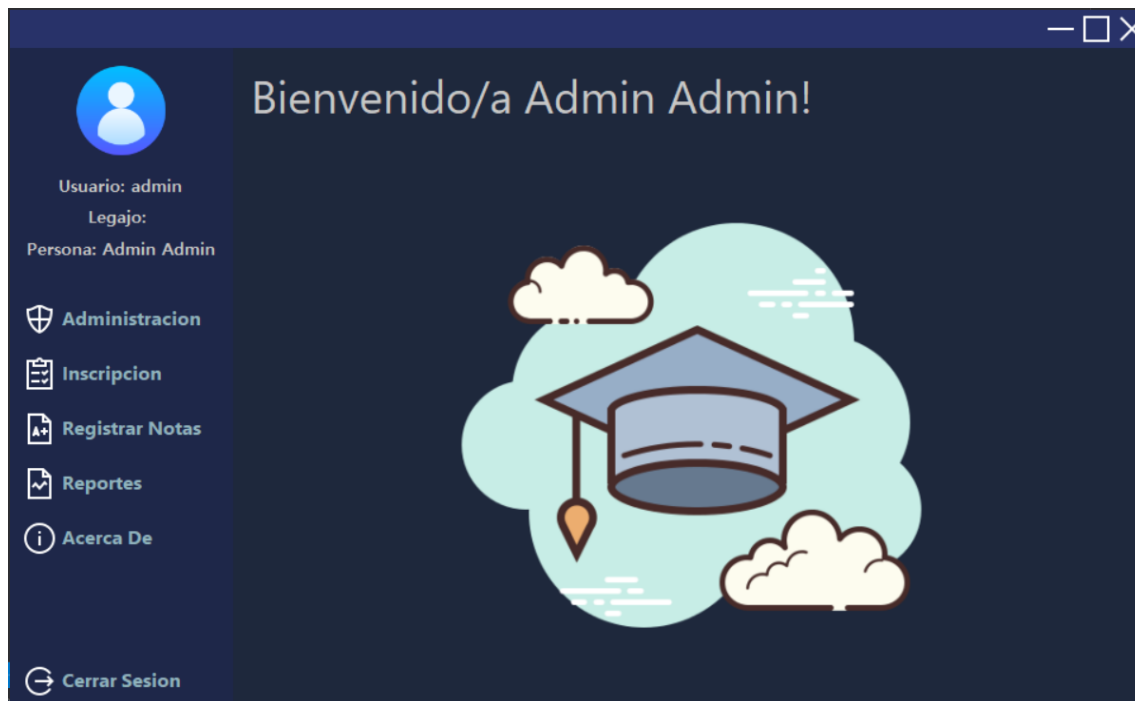
Web:



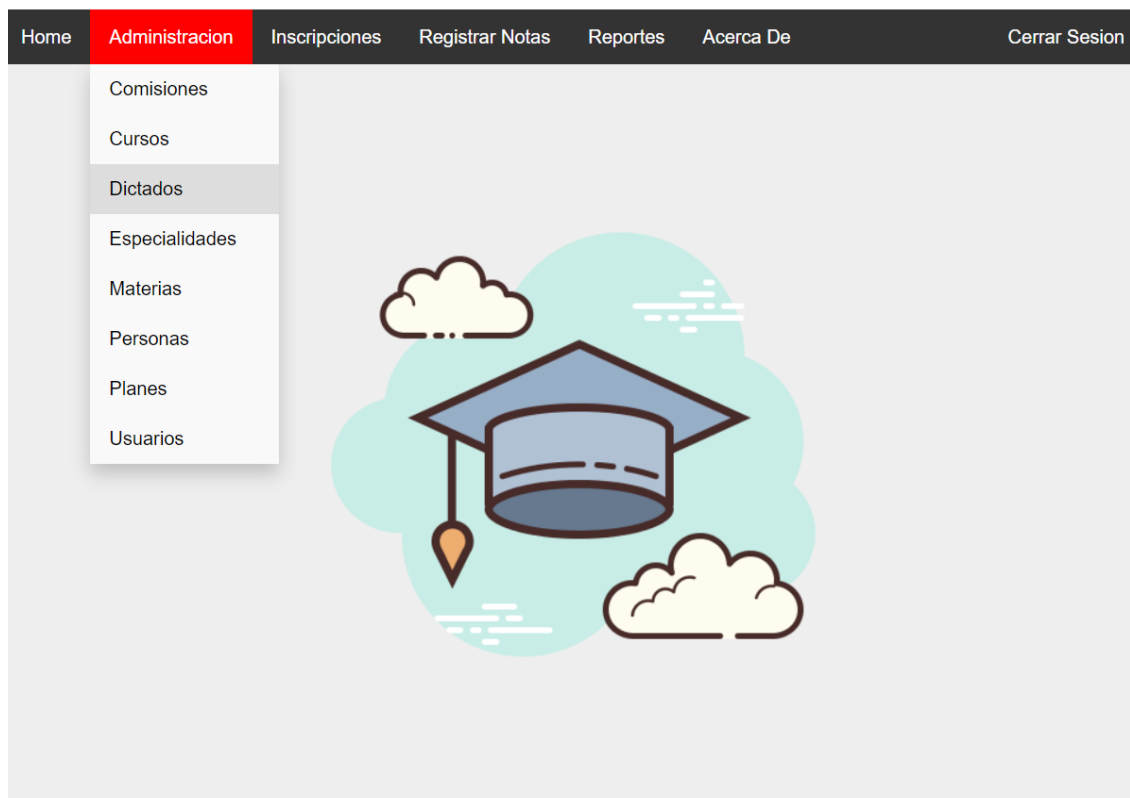
A web application login form. The top has a dark gray header with the word "Academia" in white. Below the header is a light gray background. In the center is a white rectangular box containing the login fields. The box has labels "Usuario" and "Contraseña" above their respective input fields. Below the fields is a blue button labeled "Iniciar Sesión" and a link at the bottom that says "¿Has olvidado la contraseña?".

Menú de acceso

Escritorio: (vista de administrador)

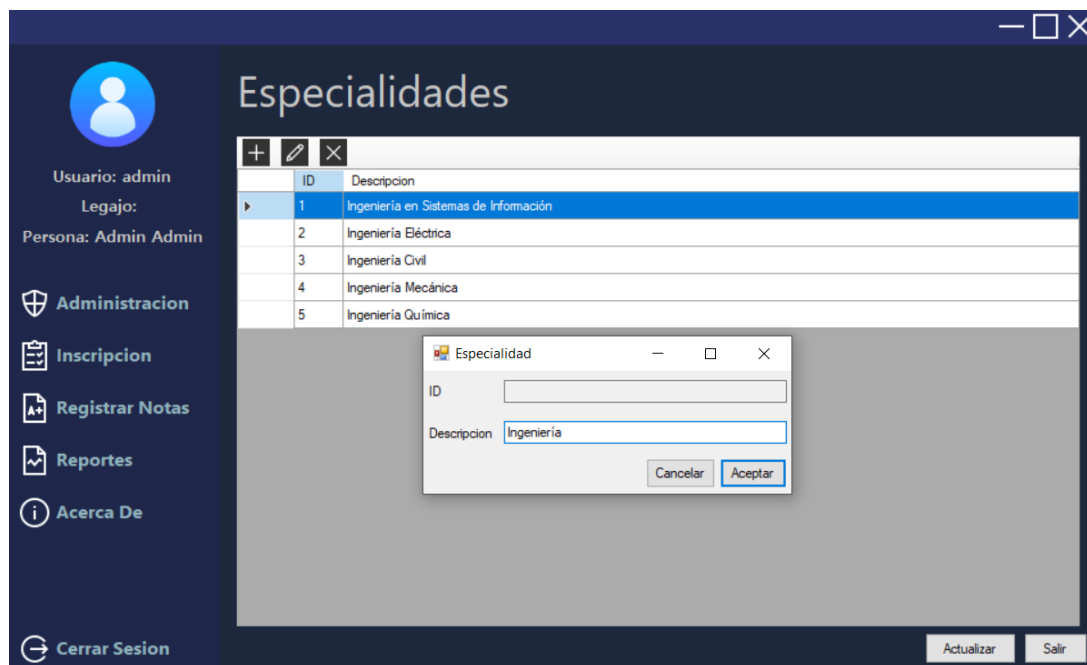


Web: (vista de administrador)

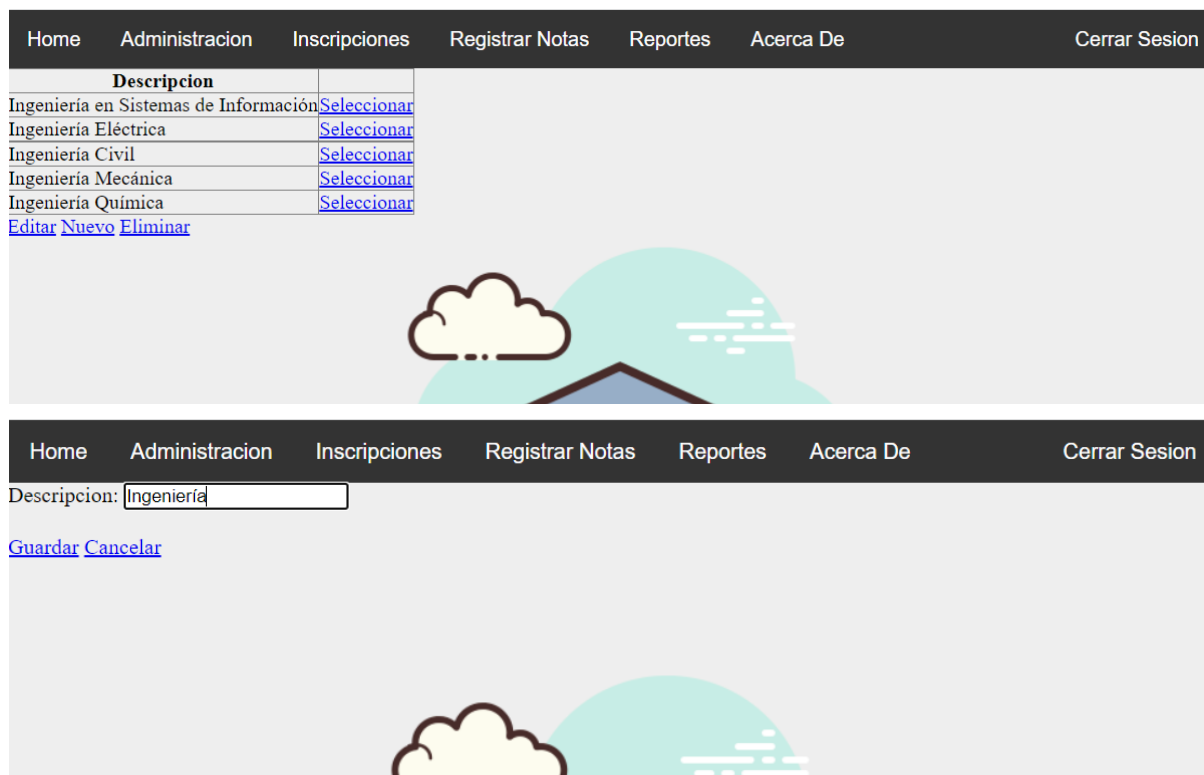


ABMC de entidades

Escritorio:



Web:



Generación de reportes

Escritorio:

Cursos



ID Curso: 2 Año Calendario: 2021 Cupo: 40

ID Materia: 2 Descripción: Software del Desarrollo IDE

Comisión: 3

ID	Alumno	Legajo
4	Franco Fonzo	46882
8	Federico Colarte	47536
7	Lara Del Coro	46978
6	German Tano	46873
9	Juana Garcia	47502
12	Sofia Rodriguez	46980

Planes



ID Plan	Descripción	ID Especialidad	Desc Esp
2	Plan de Estudio 2008	1	Ingeniería en Sistemas de Información

Inscripciones

Escritorio:

Estado Academico

Usuario: candeporatti
Legajo: 47072
Persona: Candela Poratti

Inscripcion

Acerca De

Cerrar Sesión

Materia	Comision	Condicion	Nota
Algoritmos y Estructura de Datos	104	Inscripto	

Inscripcion

Alumno: Candela Poratti Condicion: Inscripto

Cursos

Materia	Comision	Año calendario
Software del Desarrollo IDE	305	2021
Paradigmas de Programación	201	2021
Paradigmas de Programación	202	2021
Matemática Discreta	103	2021
Análisis de Sistemas	201	2021
Sintaxis y Semántica de los Le...	305	2021
Matemática Superior	305	2021

Cancelar Inscribir

Actualizar Salir

Web:

Home Administracion Inscripciones Registrar Notas Reportes Acerca De Cerrar Sesión

Alumnos: Juana Garcia

Materia	Comision	Condicion	Nota
Física 2	103	Regular	4
Álgebra y geometría analítica	101	Inscripto	1
Software del Desarrollo IDE	305	Aprobado	10

[Nuevo Eliminar](#)

Home Administracion Inscripciones Registrar Notas Reportes Acerca De Cerrar Sesión

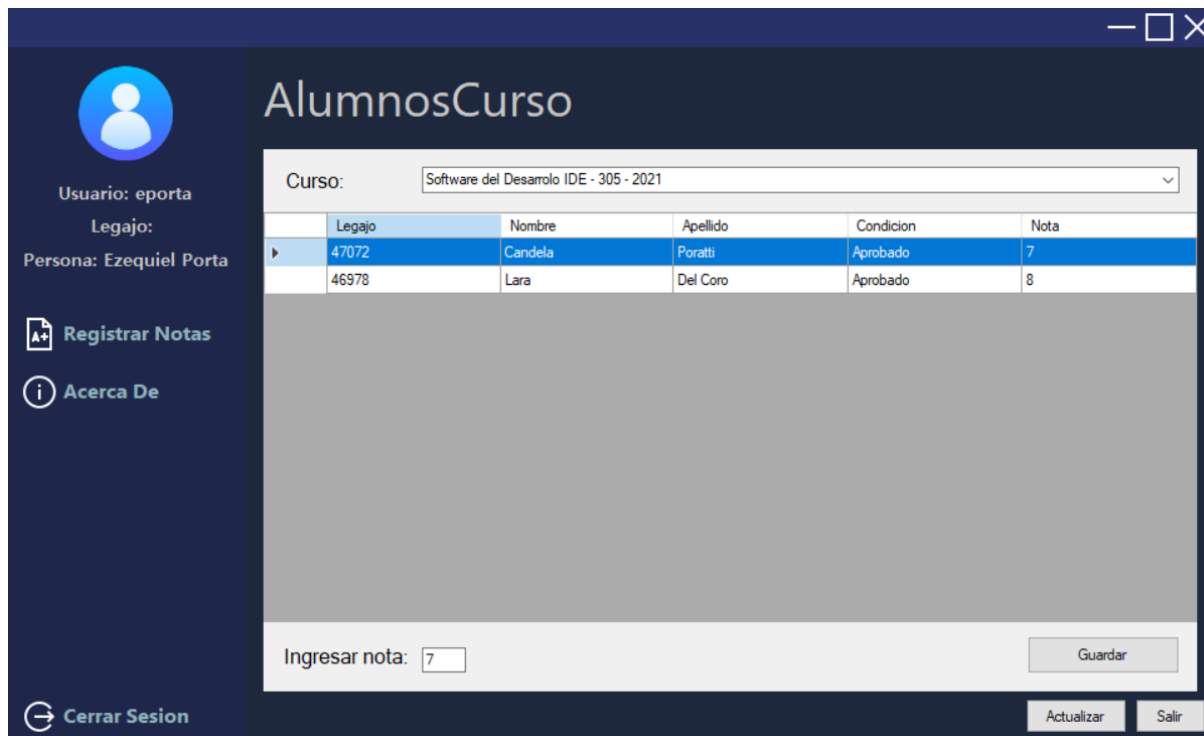
Cursos:

Materia	Comision	Año Calendario
Algoritmos y Estructura de Datos	104	2021
Software del Desarrollo IDE	305	2021
Paradigmas de Programación	201	2021
Paradigmas de Programación	202	2021
Matemática Discreta	103	2021
Análisis de Sistemas	201	2021
Sintaxis y Semántica de los Lenguajes	305	2021
Matemática Superior	305	2021
Física 2	103	2021
Álgebra y geometría analítica	101	2021

[Cancelar Guardar](#)

Registro de notas

Escritorio:



The desktop application 'AlumnosCurso' has a dark blue sidebar on the left with a user profile icon and the following text: 'Usuario: eporta', 'Legajo: Persona: Ezequiel Porta', 'Registrar Notas' (with a plus icon), 'Acerca De' (with an info icon), and 'Cerrar Sesion' (with a back icon). The main area has a title 'AlumnosCurso' and a dropdown menu for 'Curso:' set to 'Software del Desarrollo IDE - 305 - 2021'. Below this is a table with columns: Legajo, Nombre, Apellido, Condicion, and Nota. The table contains two rows: one for Legajo 47072 (Candela Poratti, Aprobado, Nota 7) and one for Legajo 46978 (Lara Del Coro, Aprobado, Nota 8). At the bottom, there is a field 'Ingresar nota:' with the value '7' and a 'Guardar' button. In the bottom right corner, there are 'Actualizar' and 'Salir' buttons.

Legajo	Nombre	Apellido	Condicion	Nota
47072	Candela	Poratti	Aprobado	7
46978	Lara	Del Coro	Aprobado	8

Web:



The web application 'AlumnosCurso' has a dark header with links: 'Home', 'Registrar Notas', 'Acerca De', and 'Cerrar Sesion'. Below the header is a dropdown menu for 'Curso:' set to 'Software del Desarrollo IDE - 305 - 2021'. Below this is a table with columns: Legajo, Nombre, Apellido, Condicion, and Nota. The table contains two rows: one for Legajo 47072 (Candela Poratti, Inscripto, Nota 7) and one for Legajo 46978 (Lara Del Coro, Inscripto, Nota 8). Below the table, there is a field 'Ingresar nota:' with a dropdown menu showing '7' and buttons for 'Cancelar' and 'Guardar'. At the bottom, there is a large illustration of a graduation cap (mortarboard) with a tassel, surrounded by clouds and a light blue background.

Legajo	Nombre	Apellido	Condicion	Nota
47072	Candela	Poratti	Inscripto	7
46978	Lara	Del Coro	Inscripto	8

Fragmentos de código

Entity Framework:

Ejemplo de método GetAll() y GetOne() para Usuarios, aplicando Entity Framework:

```

1.     public List<Usuario> GetAll()
2.     {
3.         try
4.         {
5.             using (AcademiaContext context = new AcademiaContext())
6.             {
7.                 return context.Usuario.Include(u =>
u.Persona).ToList();
8.             }
9.         }
10.        catch (Exception ex)
11.        {
12.            throw new Exception("Error al recuperar los
usuarios.", ex);
13.        }
14.    }
15.
16.
17.    public Usuario GetOne(int id)
18.    {
19.        try
20.        {
21.            using (var context = new AcademiaContext())
22.            {
23.                return context.Usuario.Include(u =>
u.Persona).FirstOrDefault(u => u.ID == id);
24.            }
25.        }
26.        catch (Exception ex)
27.        {
28.            throw new Exception("Error al recuperar el usuario:
/nMensaje.", ex);
29.        }
30.    }

```

Integración con Azure DevOps

Creamos un proyecto de Azure DevOps en el que usamos el repositorio de Github en el cual fuimos desarrollando la aplicación a lo largo del año.

Para esto, creamos un pipeline de compilación de builds. Con un trigger, el pipeline se ejecuta cada vez que se realiza un commit en la rama Main del repositorio Github asociado.

Al terminar de ejecutarse el primer pipeline se crea un artefacto, que gracias al uso de la plantilla de asp.net, toma el proyecto web y lo envía a una pipeline de release. Una vez que se ejecuta esa segunda pipeline, Azure envía el artefacto, es decir, todo el proyecto compilado, a un App service para poder acceder a la aplicación.

El App service y la aplicación están conectadas a la base de datos cargada en Azure, por lo que los datos están unificados e integrados.

De esta manera obtenemos una integración y distribución continua del proyecto a través de los servicios de Azure.

[Link al proyecto Azure DevOps aquí](#)

[Link a la aplicación web aquí](#)