

Trabajo Práctico N°8

Simulación

“Autogestión de Alumnos”



UTN FRRO 2020

Grupo N°1

Integrantes:

- Octavio Berruti
- Giovanni Martin

Legajo: 45332

Legajo: 44912

Autogestión de Alumnos

Abstract:

El objetivo de nuestro proyecto es realizar una página web que le sea de gran utilidad a los estudiantes para que se puedan organizar de una manera más eficiente y no se descuiden en los compromisos que deban realizar. Además, queremos proporcionar un espacio de almacenamiento a los datos que tengan los alumnos acerca de cada asignatura, tarea o parcial.

Narrativa:

Nosotros vamos a realizar un sistema web de autogestión de calificaciones para alumnos. Consiste en un sitio donde cada alumno se deberá registrar, registrar las asignaturas que está cursando con los profesores que dictan en dicha comisión, como también los datos de la asignatura como comisión, horarios, etc. Una vez registrado el alumno podrá ingresar tareas o parciales de las distintas asignaturas ingresando la fecha de entrega y una descripción. El alumno podrá consultar por materia los exámenes o tareas y el sistema las mostrará detalladamente, dando las anotaciones que el alumno realizó del mismo (ya sean temas, notas o lo que desee) y una vez finalizado el parcial o la tarea podrá ingresar una nota que luego el sistema calculará el promedio y la tendrá disponible para que el alumno la consulte.

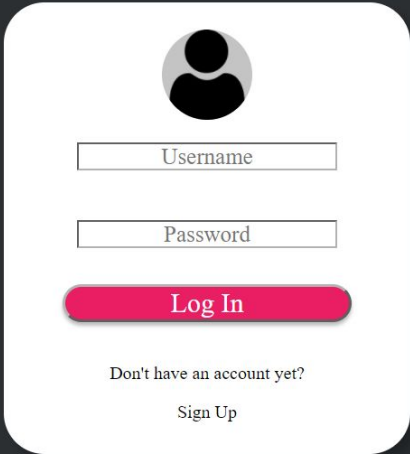
El sistema tiene un apartado llamado calendario que ordenara los compromisos y le irá mostrando al alumno cuales son sus próximos compromisos en cada día/semana o mes.

Condiciones a cumplir:

1. Programación en Python.
2. Utilizar la arquitectura de capas.
3. Bases de datos MySQL.
4. Conexión a la Base de datos con pymysql y ORMAlchemy.
5. Interfaz web utilizando React.
6. Conexión del back end y front end con Flask.

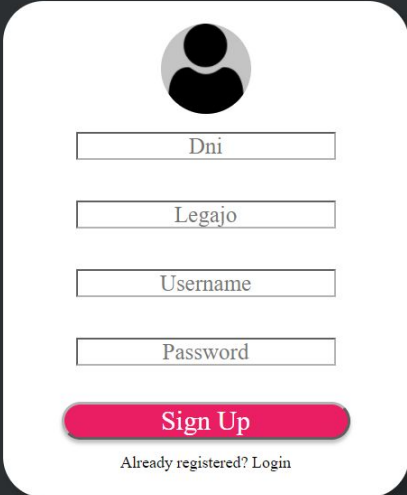
UI de la aplicación:

El sistema comienza con un login para los usuarios ya registrados



A login form UI on a dark gray background. It features a white rounded rectangle containing a user icon at the top. Below the icon are two input fields labeled 'Username' and 'Password'. A prominent pink 'Log In' button is centered below the fields. At the bottom, there is a link that says 'Don't have an account yet?' followed by a 'Sign Up' link.

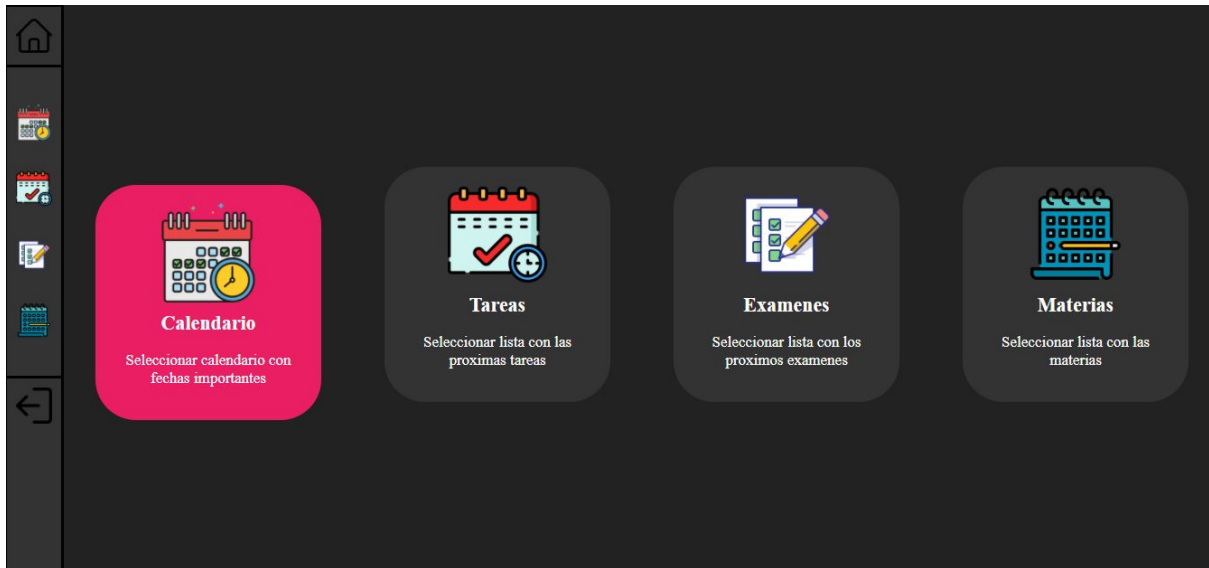
Dando la opción de un sign up para los usuarios nuevos



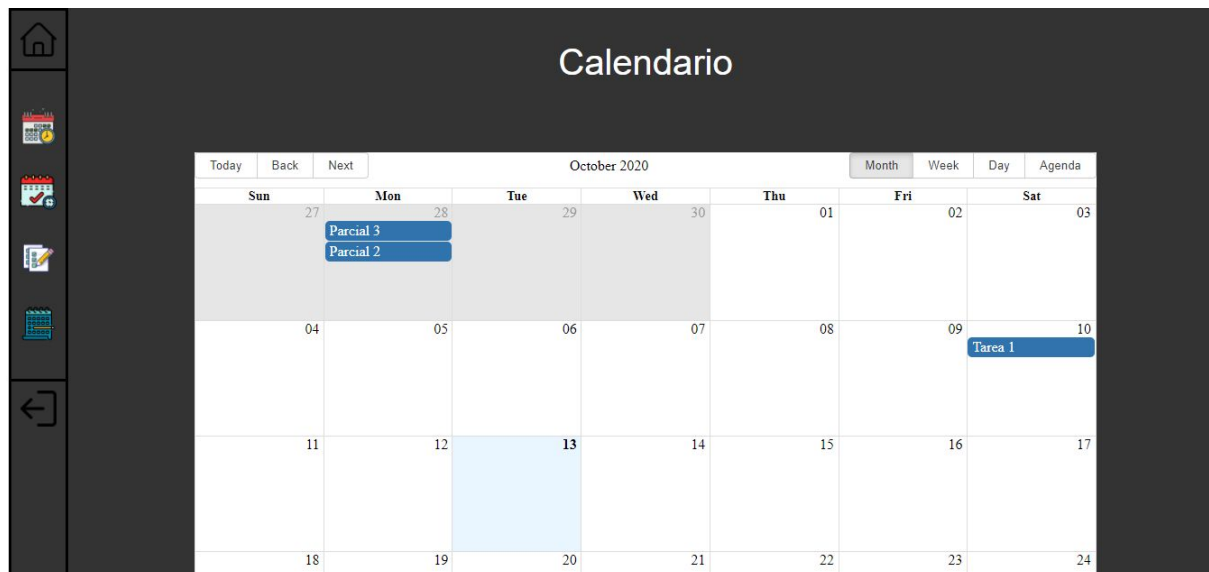
A sign up form UI on a dark gray background. It features a white rounded rectangle containing a user icon at the top. Below the icon are four input fields labeled 'Dni', 'Legajo', 'Username', and 'Password'. A prominent pink 'Sign Up' button is centered below the fields. At the bottom, there is a link that says 'Already registered? Login'.

Al ingresar a la aplicación, esta cuenta con un dashboard que permite acceder a todos los módulos disponibles.

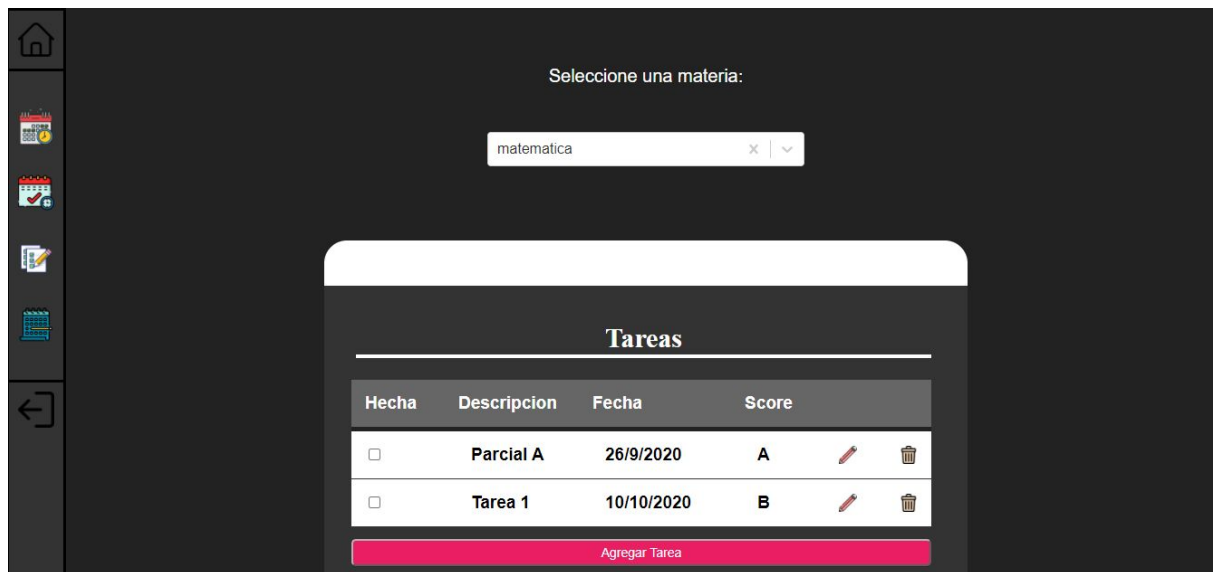
Además, existe una barra de navegación que está presente en todos los módulos.



La sección de calendario cuenta con un calendario flexible entre día/semana/mes



Las secciones de tarea y exámenes muestran los elementos ya ingresados



Dando la posibilidad de agregar nuevos

Agregar tarea

Título tarea*

☐ Esta hecha?

.../.../....

Score

Confirmar Cancelar

La sección de materias cuenta con una tabla que muestra las materias registradas y además se pueden agregar nuevas

Materias

Nombre	Hora teoria	Hora practica	Profesor Teoria	Profesor Practica	Division	Condicion	Score
matematica	15	12	Mario	Juan ignacio	1	aprobado	100

Agregar Materia

Librerias del BackEnd:

- **DateTime:**

The datetime module supplies classes for manipulating dates and times.

datetime.timedelta

A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

- **Flask:**

Flask.redirect

Returns a response object that, if called, redirects the client to the target location.

Flask.session

The session object works pretty much like an ordinary dict, with the difference that it keeps track on modifications.

Flask.url_for

Generates a URL to the given endpoint with the method provided.

Flask.render_template

Renders a template from the template folder with the given context.

Flask.request

The request object used by default in Flask.

It is what ends up as request. If you want to replace the request object used you can subclass this and set request_class to your subclass.

Flask.flash

Flashes a message to the next request. In order to remove the flashed message from the session and to display it to the user, the template has to call get_flashed_messages().

Flask_sqlAlchemy

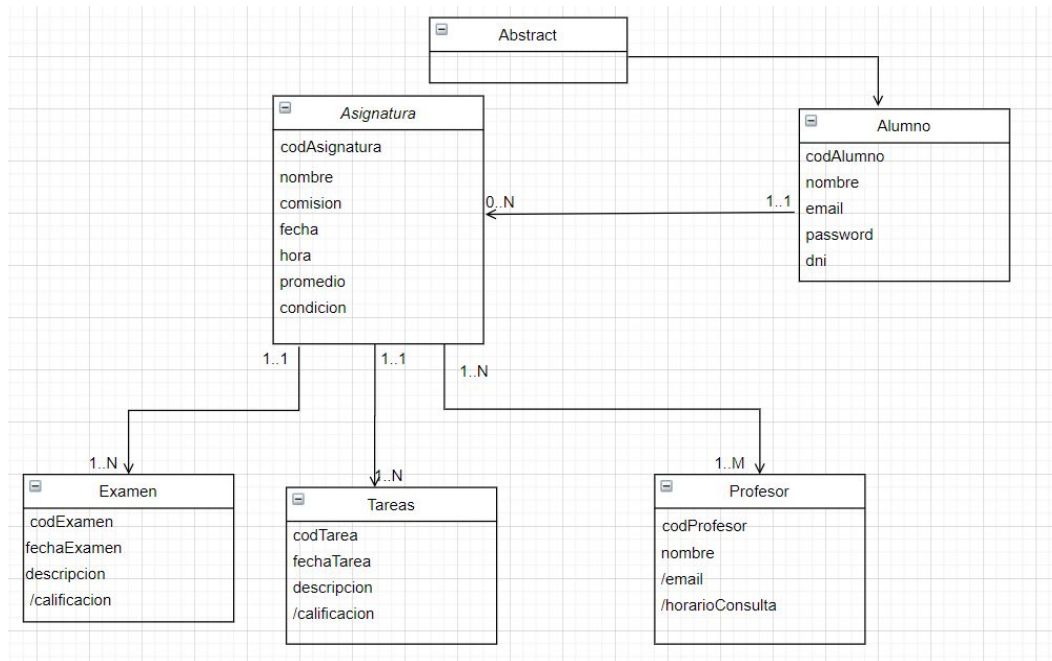
Flask_sqlalchemy is an extension for Flask that adds support for SQLAlchemy to your application. It aims to simplify using SQLAlchemy with Flask by providing useful defaults and extra helpers that make it easier to accomplish common tasks.

Librerías del FrontEnd:

- axios
- eslint
- history
- lodash
- material-ui
- moment
- prop-types
- react
- react-big-calendar
- react-cookie
- react-date-picker
- react-dom
- react-hook-form
- react-router-dom
- react-scripts
- react-select
- typescript

Además, para manejar mejor las dependencias del frontend del proyecto utilizaremos Yarn como nuestro gestor de dependencias.

Modelo de Domino:



Casos de uso:

CUU01 - Registrarse en el sistema

Descripción: Alumno se registra en el sistema

Actores: Alumno, sistema

Precondiciones: --

Flujo normal:

1. El alumno ingresa a la página web.
2. El alumno ingresa sus datos.
3. Sistema comprueba la validez de los datos y registra.
4. Alumno ingresa las asignaturas y los datos correspondientes.

Flujo alternativo:

3.a.1 El sistema comprueba la validez de los datos, si los datos no son correctos o ya están registrados comunica al alumno.

3.a.2 Alumno verifica los datos y los ingresa nuevamente en caso de error.

Postcondiciones: Alumno se ha registrado exitosamente

CUU02 - Ingresar en el sistema

Descripción: Alumno ingresa en el sistema

Actores: Alumno, sistema

Precondiciones: Alumno ya registrado en el sistema

Flujo normal:

1. El alumno ingresa a la página web.
2. El alumno ingresa sus datos.
3. Sistema comprueba la validez de los datos y accede a la pagina web principal.

Flujo alternativo:

- 3.a.1 El sistema comprueba la validez de los datos, si los datos no son correctos lo comunica al alumno.
- 3.a.2 Alumno verifica los datos y los ingresa nuevamente en caso de error.

Postcondiciones: Alumno ha ingresado exitosamente

CUU03 - Registrar examen

Descripción: Alumno registra nuevo examen

Actores: Alumno, sistema

Precondiciones:

Alumno ingresó con su usuario

Flujo normal:

1. Alumno ingresa nuevo examen, asigna una materia y sus anotaciones.
2. Sistema registra nuevo examen

Postcondiciones: Alumno ha ingresado exitosamente

CUU04 - Registrar tarea

Descripción: Alumno registra nueva tarea

Actores: Alumno, sistema

Precondiciones:

Alumno ingresó con su usuario

Flujo normal:

1. Alumno ingresa nueva tarea, asigna una materia y sus anotaciones.
2. Sistema registra nueva tarea.

Postcondiciones: Tarea registrada correctamente.

CUU05 - Registrar nota en examen

Descripción: Alumno registra nota en un examen

Actores: Alumno, sistema

Precondiciones:

Alumno ingresó con su usuario

Examen reigstrado ya finalizado

Flujo normal:

1. Alumno ingresa a la seccion asignaturas y busca el examen realizado.
2. Alumno ingresa la nota.
3. Sistema registra nota.

Postcondiciones: Nota registrada correctamente.

Reglas de Negocio:

- El sistema debe calcular un promedio por asignatura y un promedio total del alumno
- El sistema de acuerdo al promedio de la asignatura debe calcular la condición final del alumno en cada asignatura