

Gestor de Pasantías

El prototipo tiene como idea brindar un mejor control a la Secretaría de Asuntos Universitarios (SAU) acerca de gestionar las pasantías , permitiendo un dinamismo a la hora de realizar acciones como, registrar un alumno, registrar una empresa, generar y tener control sobre las pasantías y llevar a cabo un registro de informes post-pasantías.

Requerimientos funcionales:

- El sistema debe permitir que los usuarios ingresen con su correo electrónico y contraseña.
- El sistema debe permitir registrar un nuevo alumno.
- El sistema debe permitir registrar una nueva empresa.
- El sistema debe permitir registrar un puesto de trabajo dentro de una empresa.
- El sistema debe permitir registrar una pasantía.
- El sistema debe permitir cancelar una pasantía activa.
- El sistema debe permitir emitir un informe sobre las pasantías finalizadas.

Requerimientos no funcionales:

- El sistema debe ser accesible únicamente desde dispositivos de escritorio, bloqueando el acceso o mostrando un aviso cuando se intente utilizar desde dispositivos móviles.
- El sistema debe permitir el acceso únicamente mediante autenticación con correo electrónico y contraseña previamente definidos, sin admitir otros métodos de autenticación.

Proceso de Negocio BPMN:

Gestor de pasantía:

- <https://drive.google.com/file/d/1Ec8g8Ft8ctTONSpwQ4Vo4g-H7ahOfff0/view>

Sub procesos:

- Registrar alumno:
<https://drive.google.com/file/d/1RWMw7943605nqLrBSsCv8nupvh1vnloS/view>
- Registrar empresa y puesto de trabajo:
<https://drive.google.com/file/d/1lcx0Ve46EA4ZWHj2reuAuLuFJCFTfVmu/view>
- Registrar pasantía:
<https://drive.google.com/file/d/17fMZVUQvX3dET8daO2jSQKdS9R-Vu2X-/view>

Diagrama caso de uso:

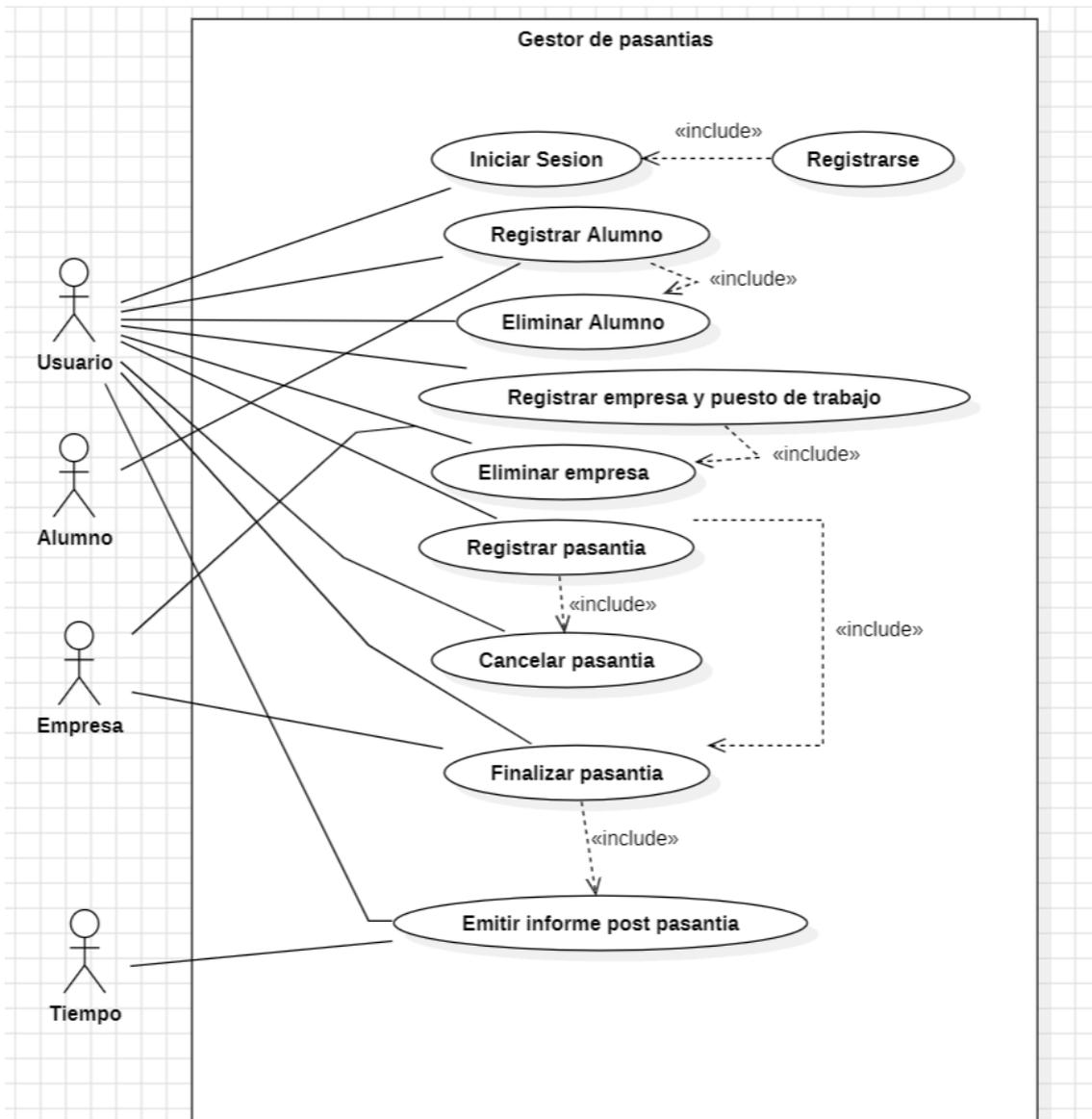
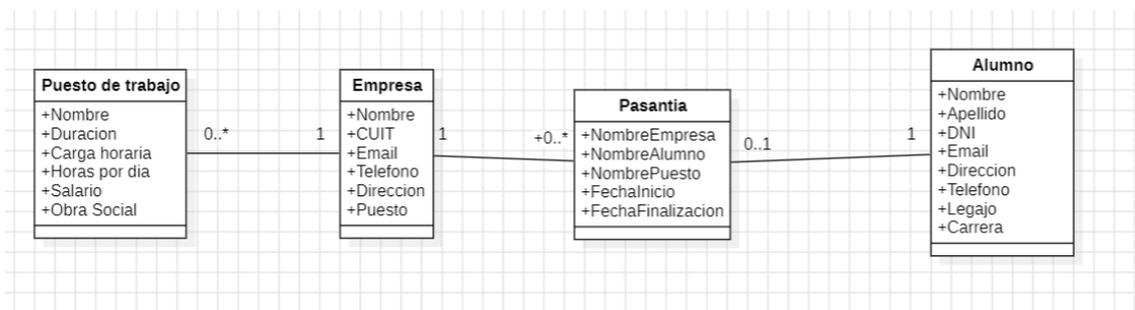


Diagrama de clases:



Descripción de casos de uso:

Caso de Uso: Iniciar sesión

ID: 1

Fecha:

Descripción: El objetivo de este caso de uso es permitir al usuario iniciar sesión dentro de la aplicación.

Actores Principales: Usuario

Actores Secundarios:

Observaciones:

Precondiciones: -

Post- Condiciones **Éxito:** El usuario logra iniciar sesión.

Fracaso:

Flujo PRINCIPAL

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario ingresa a la aplicación y se le pide iniciar sesión.

2. El usuario ya tiene una cuenta creada y procede a poner sus datos.

3. Se verifica los datos ingresados y son correctos. Se muestra pantalla de "Inicio".

4. Fin del caso de uso.

2.1 El usuario no tiene cuenta creada y se llama al caso de uso **Registrarse**.

3.2 Se verifica los datos ingresados y no son correctos.
3.2. Se vuelve al paso 2.

Caso de Uso: Registrar alumno

ID: 1

Fecha:

Descripción: El objetivo del caso de uso es permitir al usuario registrar un alumno al sistema.

Actores Principales: Usuario

Actores Secundarios: Alumno

Observaciones:

Precondiciones: -

Post- Condiciones

Éxito: Se logra registrar un nuevo alumno al sistema.

Fracaso:

Flujo PRINCIPAL

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario desea registrar un **alumno** al sistema en la pagina "Alumnos".

2. El usuario clickea sobre el botón de "Registrar".

3. El usuario ingresa Nombre, Apellido, DNI, Email, Telefono, Dirección, Legajo y Carrera del **Alumno**.

4. Se validan que los datos sean correctos y lo son. Se registra un nuevo **Alumno**.

4.1 Se validan que los datos sean correctos y no lo son.

4.2 Se vuelve al paso 3.

5. Fin de caso de uso.

Caso de Uso: Eliminar alumno

ID: 1

Fecha:

Descripción: El objetivo del caso de uso es permitir a usuario eliminar un alumno registrado del sistema.

Actores Principales: Usuario

Actores Secundarios:

Observaciones:

Precondiciones: -

Post- Condiciones

Éxito: Se logra eliminar al alumno del sistema.

Fracaso:

Flujo principal

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario desea eliminar un **Alumno** registrado.

2. Se muestran en pantalla los **alumnos** registrados en la página "Alumnos".

3. El usuario elige al **Alumno** por eliminar y se borra del sistema.

4. Fin de CU.

Caso de Uso: Registrar empresa y puesto de trabajo**ID:** 1**Fecha:****Descripción:** El objetivo del caso de uso es permitir al usuario registrar una empresa al sistema.**Actores Principales:** Usuario**Actores Secundarios:** Empresa**Observaciones:****Precondiciones:** -**Post- Condiciones****Éxito:** Se logra registrar una nueva empresa al sistema.**Fracaso:****Flujo principal****Flujo Alternativo**

1. **El caso de uso comienza cuando** el usuario desea registrar una **empresa** en el sistema en la página "Empresas".

2. El usuario clickea sobre el botón de "Registrar".

3. El usuario ingresa Nombre, CUIT, Email, Teléfono, Dirección, Nombre del puesto, Duración, Carga Horaria, Horas por día, Salario y Obra social de la **Empresa** y el **Puesto** de trabajo.

4. Se validan que los datos sean correctos y lo son. Se registra un nuevo **Empresa**.

4.1 Se validan que los datos sean correctos y no lo son.
4.2 Se vuelve al paso 3.

5. Fin de caso de uso.

Caso de Uso: Eliminar empresa**ID:** 1**Fecha:****Descripción:** El objetivo del caso de uso es permitir a usuario eliminar una empresa registrada del sistema.**Actores Principales:** Usuario**Actores Secundarios:****Observaciones:****Precondiciones:** -**Post- Condiciones** **Éxito:** Se logra eliminar a la empresa del sistema.**Fracaso:**

Flujo principal

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario desea eliminar una **Empresa** registrada.

2. Se muestran en pantalla las **empresas** registradas en la página "Empresas".

3. El usuario elige a la **Empresa** por eliminar y se borra del sistema.

4. Fin de CU.

Caso de Uso: Registrar pasantía

ID: 1

Fecha:

Descripción: El objetivo del caso de uso es permitir al usuario registrar una pasantía al sistema.

Actores Principales: Usuario

Actores Secundarios:

Observaciones:

Precondiciones: -

Post- Condiciones

Éxito: Se logra registrar una nueva pasantía al sistema.

Fracaso:

Flujo principal

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario desea registrar una **pasantía** al sistema en la página "Pasantías".

2. El usuario ingresa NombreEmpresa, NombreAlumno, NombrePuesto, FechaInicio, FechaFinalizacion.

3. El usuario clickea el botón "Registrar Pasantía", se validan que los datos sean correctos y lo son. Se registra una nueva **Pasantía**.

3.1 Se validan que los datos sean correctos y no lo son.

3.2 Se vuelve al paso 2.

4. Fin de caso de uso.

Caso de Uso: Cancelar pasantía**ID:** 1**Fecha:****Descripción:** El objetivo del caso de uso es permitir al usuario cancelar una pasantía registrada del sistema.**Actores Principales:** Usuario**Actores Secundarios:****Observaciones:****Precondiciones:** -**Post- Condiciones** **Éxito:** Se logra cancelar la pasantía.**Fracaso:**

Flujo principal

Flujo Alternativo

1. **El caso de uso comienza cuando** el usuario desea cancelar una **Pasantía** registrada.2. Se muestran en pantalla las **pasantías** registradas en el apartado de "pasantías activas" en la página "Pasantías".3. El usuario elige la **pasantía** y se cancela. Se muestra en apartado de "pasantías canceladas".

4. Fin de CU.

Caso de Uso: Finalizar pasantía**ID:** 1**Fecha:****Descripción:** El objetivo del caso de uso es permitir al usuario finalizar una pasantía registrada en el sistema.**Actores Principales:** Usuario**Actores Secundarios:****Observaciones:****Precondiciones:** -

Post- Condiciones	Éxito: Se logra finalizar una pasantía en el sistema.
	Fracaso:
Flujo principal	Flujo Alternativo
1. El caso de uso comienza cuando el usuario desea finalizar una pasantía registrada.	
2. Se muestran en pantalla las pasantías registradas en el apartado de "pasantías activas" en la página "Pasantías".	
3. El usuario elige la pasantía y procede a finalizarla. Se muestra en el apartado de "pasantías finalizadas".	
4. Fin de CU.	

Prototipo:

<https://drive.google.com/file/d/1Cif7LU0qCAnY74GptWmAoCCPYMmCqY2N/vi>
[ew](#)

Casos de Testing de la primera Iteración:

En esta primera iteración se busca comprobar:

- Correcto funcionamiento del inicio de sesión
- Correcto funcionamiento a la hora de registrar un alumno
 - Validar el correcto ingreso de datos
 - Mostrar los alumnos registrados
- Correcto funcionamiento a la hora de registrar una empresa
 - Validar el correcto ingreso de datos
 - Validar que el registro del puesto cumpla con las condiciones para la pasantía
 - Mostrar las empresas registradas
- Permitir eliminar un alumno.
- Permitir eliminar una empresa.

ID del caso: 1	Título: Inicio de sesión
-----------------------	---------------------------------

Descripción: Lograr que el usuario inicie sesión dentro de la aplicación en caso de tener una cuenta creada, como el la primera vez que se utiliza se verifica que el registro valide un mail y una contraseña y que luego con esos mismos datos se puede iniciar sesión.	
Pasos para reproducir:	
<ol style="list-style-type: none"> 1. Entrar a la aplicación 2. Verificar que el registro valide un mail y una contraseña 3. Verificar que solo se pueda ingresar con los datos verificados 4. Utilizar los datos registrados para entrar a la aplicación 5. Ingresar a la página "Inicio" 	
Resultado: El sistema de registro valida la entrada de un mail y una contraseña pero no avisa si estás colocando un texto invalido en el casillero del mail. Luego el botón de inicio de sesión solo valida el inicio con los datos ingresados y luego envía correctamente a la pagina "Inicio" de la aplicación.	
Estado: Éxito.	Nota: Agregar mensaje para cuando no se ingresa un valor tipo mail a la hora de registrarse.

ID del caso: 2	Título: Validar datos ingresados del alumno.
Descripción: Se busca validar que los datos ingresados acerca del alumno a registrar sean correctos y concisos.	
Pasos para reproducir:	
<ol style="list-style-type: none"> 1. Ir a la página "Alumnos". 2. Presionar "Registrar". 3. Ingresar datos y validar que solo tome datos correctos. 4. Ingresar datos correctos y verificar que permita registrar al alumno. 	
Resultado: El sistema redirige al usuario correctamente a la página "Alumnos". El botón de registrar muestra en pantalla una ventana con un formulario para completar. Los datos a ingresar se validan correctamente y luego permite registrar al alumno de manera exitosa.	
Estado: Éxito.	Nota: Agregar mensajes que indiquen los casilleros con datos invalidados.

ID del caso: 3	Título: Validar datos de la empresa y condiciones del puesto de trabajo.
Descripción: Se busca validar que los datos ingresados acerca de la empresa a registrar sean correctos y concisos. Además de validar que el puesto de trabajo cumpla con las condiciones para la pasantía.	
Pasos para reproducir: <ol style="list-style-type: none"> 1. Ir a la página “Empresas”. 2. Presionar “Registrar”. 3. Ingresar datos de la empresa y validar que solo tome datos correctos. 4. Ingresar datos de puesto de trabajo y validar que sean correctos y cumplan con las condiciones. 5. Ingresar datos correctos y verificar que permite registrar una empresa. 	
Resultado: El sistema redirige al usuario correctamente a la página “Empresas”. El botón de registrar muestra en pantalla una ventana con un formulario para completar. Los datos de la empresa a ingresar se validan correctamente al igual que los datos del puesto de trabajo que se valida que cumplan con las condiciones. Luego permite registrar a la empresa de manera exitosa.	
Estado: Éxito.	Nota: Agregar mensajes que indiquen los casilleros con datos invalidados.

ID del caso: 4	Título: Mostrar alumnos registrados.
Descripción: Se busca verificar que en la página “Alumnos” se muestre aquellos que fueron registrados.	
Pasos para reproducir: <ol style="list-style-type: none"> 1. Ir a la página “Alumnos” 2. Registrar un alumno 3. Verificar que en pantalla se muestre el alumno registrado. 	
Resultado: El sistema luego de registrar un alumno muestra correctamente al alumno registrado en pantalla, incluso cuando se cambia de pagina sigue allí guardado. El problema es al reiniciar la página ya que el alumno registrado desaparece de pantalla pero en el almacenamiento del código sigue estando.	
Estado: Fallo.	Nota: Arreglar error a la hora de reiniciar página y que el alumno desaparezca de la pantalla.

ID del caso: 5	Título: Mostrar empresas registradas.
Descripción: Se busca verificar que en la página “Empresas” se muestre aquellas que fueron registradas.	
Pasos para reproducir: <ol style="list-style-type: none"> 1. Ir a la página “Empresas” 2. Registrar una empresa 3. Verificar que en pantalla se muestre la empresa registrada. 	
Resultado: El sistema luego de registrar una empresa muestra correctamente a la empresa registrada en pantalla, incluso cuando se cambia de pagina sigue allí guardada. El problema es al reiniciar la página ya que la empresa registrada desaparece de pantalla pero en el almacenamiento del código sigue estando.	
Estado: Fallo.	Nota: Arreglar error a la hora de reiniciar pagina y que la empresa desaparezca de la pantalla. Agregar también datos sobre el puesto de trabajo de la empresa.

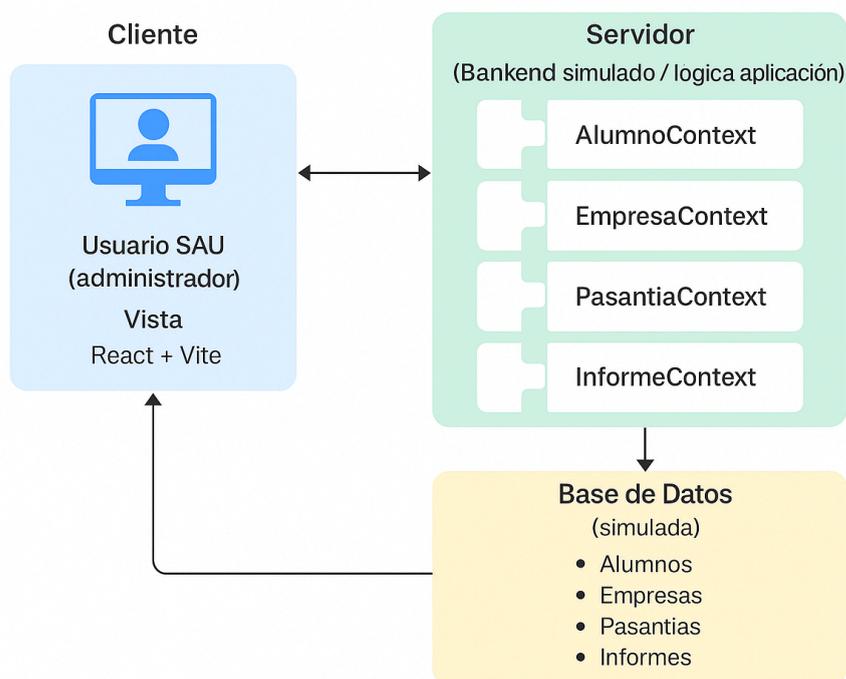
ID del caso: 6	Título: Eliminar alumno registrado.
Descripción: Se busca verificar que se pueda eliminar a un alumno registrado.	
Pasos para reproducir: <ol style="list-style-type: none"> 1. Ir a la pagina “Alumnos” 2. Registrar un alumno 3. Verificar que en pantalla se muestre al alumno registrado. 4. Verificar que aparezca el botón eliminar. 5. Verificar que el botón elimine definitivamente al alumno. 	
Resultado: El sistema luego de registrar un alumno y mostrarlo en pantalla, muestra consigo el botón eliminar. Al presionarlo, cumple su función de eliminar al alumno definitivamente.	
Estado: Éxito.	Nota: -

ID del caso: 7	Título: Eliminar empresa registrada
Descripción: Se busca verificar que se pueda eliminar a una empresa registrada.	
Pasos para reproducir: <ol style="list-style-type: none"> 1. Ir a la página “Empresas” 2. Registrar una empresa 3. Verificar que en pantalla se muestre la empresa registrada. 4. Verificar que aparezca el botón eliminar. 5. Verificar que el botón elimine definitivamente a la empresa. 	
Resultado: El sistema luego de registrar una empresa y mostrarla en pantalla, muestra consigo el botón eliminar. Al presionarlo, cumple su función de eliminar a la empresa definitivamente.	
Estado: Éxito.	Nota: -.

Segunda Iteración:

Arquitectura:

Mi aplicación está basada en una arquitectura cliente-servidor. El cliente, desarrollado con React, ofrece una interfaz web enfocada exclusivamente en el uso por parte del personal de la Secretaría de Asuntos Universitarios (SAU). Esta interfaz permite gestionar alumnos, empresas y pasantías a través de componentes reutilizables, validaciones de formularios y navegación entre vistas usando React Router.



Clasificación de errores:

ID del caso: 5

Título: Mostrar empresas registradas.

Descripción: Se busca verificar que en la página “Empresas” se muestre aquellas que fueron registradas.	
Pasos para reproducir: 4. Ir a la página “Empresas” 5. Registrar una empresa 6. Verificar que en pantalla se muestre la empresa registrada.	
Resultado: El sistema luego de registrar una empresa muestra correctamente a la empresa registrada en pantalla, incluso cuando se cambia de pagina sigue allí guardada. El problema es al reiniciar la página ya que la empresa registrada desaparece de pantalla pero en el almacenamiento del código sigue estando.	
Estado: Fallo.	Nota: Arreglar error a la hora de reiniciar pagina y que la empresa desaparezca de la pantalla. Agregar también datos sobre el puesto de trabajo de la empresa.
Clasificación de error: Funcional	

ID del caso: 5	Título: Mostrar empresas registradas.
Descripción: Se busca verificar que en la página “Empresas” se muestre aquellas que fueron registradas.	
Pasos para reproducir: 7. Ir a la página “Empresas” 8. Registrar una empresa 9. Verificar que en pantalla se muestre la empresa registrada.	
Resultado: El sistema luego de registrar una empresa muestra correctamente a la empresa registrada en pantalla, incluso cuando se cambia de pagina sigue allí guardada. El problema es al reiniciar la página ya que la empresa registrada desaparece de pantalla pero en el almacenamiento del código sigue estando.	
Estado: Fallo.	Nota: Arreglar error a la hora de reiniciar pagina y que la empresa desaparezca de la pantalla. Agregar también datos sobre el puesto de trabajo de la empresa.
Clasificación de error: Funcional	

Endpoints

Método	Endpoint	Descripción
GET	/alumnos	Listar todos los alumnos.
GET	/alumnos/:legajo	Obtener un alumno por legajo.

POST	/alumnos	Registrar un nuevo alumno.
PUT	/alumnos/:legajo	Actualizar los datos de un alumno.
DELETE	/alumnos/:legajo	Eliminar un alumno registrado.
GET	/empresas	Listar todas las empresas.
GET	/empresas/:cuit	Obtener una empresa por CUIT.
POST	/empresas	Registrar una nueva empresa.
PUT	/empresas/:cuit	Modificar los datos de una empresa.
DELETE	/empresas/:cuit	Eliminar una empresa.
GET	/pasantias	Listar todas las pasantías.
GET	/pasantias/:id	Obtener detalles de una pasantía específica.
POST	/pasantias	Registrar una nueva pasantía.
PUT	/pasantias/:id	Actualizar información de una pasantía.
DELETE	/pasantias/:id	Cancelar (eliminar) una pasantía.

Tercera Iteración:

Implementacion: <https://candid-nougat-d4e11f.netlify.app/>

Para verificar el inicio de sesion utilizar:

- admin@gmail.com (Email)
- admin123 (Contraseña)

Nota a la hora de registrar empresa: En la parte del puesto en Carga Horaria Semanal ingresar valor 20.

Testing con la implementación:

ID del caso: 1	Título: Verificar e iniciar sesión de manera correcta.
Descripción: Se busca verificar que a la hora de iniciar sesión solo acepte los datos del admin y permita ingresar a la página "Inicio".	
Pasos para reproducir:	
<ol style="list-style-type: none"> 10. Ir a la página "Inicio de Sesión" 11. Ingresar datos incorrectos para corroborar la validación. 12. Ingresar datos correctos para corroborar la validación. 13. Ingresar a la página "Inicio" 	
Resultado: El sistema luego de ingresar a la página "Inicio de sesión" y colocar datos incorrectos corrobora correctamente que esos datos no son válidos. Luego de colocar los datos correctos la página valida los mismos y nos envía a la página "Inicio".	
Estado: Exito.	Nota: -
Clasificación de error: -	

ID del caso: 2	Título: Registrar y mostrar alumnos
Descripción: Se busca verificar que a la hora de registrar un alumno, se guarde en la base de datos y se muestre en pantalla.	
Pasos para reproducir:	
<ol style="list-style-type: none"> 14. Ir a la página "Alumnos" 15. Presionar botón "Registrar Alumno" 16. Ingresar datos (corroborar que sean correctos) y registrar alumno 17. Verificar que el alumno registrado se muestre en pantalla. 	
Resultado: El sistema luego de ingresar a la página "Alumno" y presionar el botón "Registrar Alumno", despliega una ventana con un formulario para completar con datos el cual verifica que los datos sean válidos. Sin embargo, no permite guardar al alumno en la base de datos. Los alumnos generados previamente en la base de datos se muestran en pantalla.	
Estado: Fallo.	Nota: Arreglar error: Failed to load resource: the server responded with a status of 404 () que no permite que los alumnos se guarden.

Clasificación de error: Funcional.

ID del caso: 3	Título: Eliminar alumnos.
Descripción: Se busca verificar que el botón "Eliminar" cumpla con su función y borre al alumno de la base de datos.	
Pasos para reproducir: 18. Ir a la página "Alumnos" 19. Buscar un alumno registrado y presionar el botón "Eliminar" 20. Verificar que el alumno fue eliminado correctamente.	
Resultado: El sistema luego de ingresar a la página "Alumno" y presionar el botón "Eliminar" en un alumno registrado, despliega un cartel preguntando si queremos confirmar la eliminación del alumno. Al clickear confirmar, el alumno se borra exitosamente tanto en pantalla como en código.	
Estado: Exito.	Nota: -
Clasificación de error: -	

ID del caso: 4	Título: Registrar y mostrar empresas
Descripción: Se busca verificar que a la hora de registrar una empresa, se guarde en la base de datos y se muestre en pantalla.	
Pasos para reproducir: 21. Ir a la página "Empresas" 22. Presionar botón "Registrar Empresa" 23. Ingresar datos (corroborar que sean correctos) y registrar empresa. 24. Verificar que la empresa registrada se muestre en pantalla.	
Resultado: El sistema luego de ingresar a la página "Empresa" y presionar el botón "Registrar empresa", despliega una ventana con un formulario para completar con datos el cual verifica que los datos sean válidos. El sistema guarda exitosamente a la empresa y la muestra en pantalla.	
Estado: Exito	Nota:
Clasificación de error:	

ID del caso: 5	Título: Eliminar empresa.
Descripción: Se busca verificar que el botón "Eliminar" cumpla con su función y borre a la empresa de la base de datos.	
Pasos para reproducir: 25. Ir a la página "Empresa" 26. Buscar un alumno registrado y presionar el botón "Eliminar" 27. Verificar que la empresa fue eliminada correctamente.	
Resultado: El sistema luego de ingresar a la página "Empresa" y presionar el botón "Eliminar" en una empresa registrada, despliega un cartel preguntando si queremos confirmar la eliminación del alumno. Al clickear confirmar, el sistema tira un error.	
Estado: Fallo.	Nota: Arreglar error: Failed to load resource: the server responded with a status of 404 () que no permite que los alumnos se guarden.

Clasificación de error: Funcional.

ID del caso: 6	Título: Registrar y mostrar pasantías en “Activas”
Descripción: Se busca verificar que a la hora de registrar una pasantía, se guarde y se muestre en la sección “Activas”	
Pasos para reproducir: 28. Ir a la página “Pasantías” 29. Presionar botón “Registrar pasantía” 30. Ingresar datos (corroborar que sean correctos) y registrar pasantía. 31. Verificar que la empresa registrada se muestre en pantalla en la sección “Activas”.	
Resultado: El sistema luego de ingresar a la página “Pasantías” y presionar el botón “Registrar pasantía”, despliega una ventana con un formulario para completar con datos el cual verifica que los datos sean válidos. El sistema guarda exitosamente a la pasantía y la muestra en pantalla en la sección “Activas”.	
Estado: Exito	Nota: -
Clasificación de error: -	

ID del caso: 7	Título: Verificar la función del botón “Cancelar” en Pasantías.
Descripción: Se busca verificar que a la hora de cancelar una pasantía, el botón funcione correctamente y muestre la pasantía cancelada en la seccion “Canceladas”.	
Pasos para reproducir: 32. Ir a la página “Pasantías” 33. Presionar el botón “Cancelar” en una pasantía activa. 34. Verificar que la pasantía se cancela y se muestra en la sección “Canceladas”.	
Resultado: El sistema luego de ingresar a la página “Pasantías” y presionar “Cancelar” en una pasantía activa, se despliega una ventana preguntando si queremos confirmar la cancelación de la pasantía. Al presionar “Cancelar”, la pasantía se cancela exitosamente y se muestra en la sección “Canceladas”.	
Estado: Exito	Nota: -
Clasificación de error: -	

ID del caso: 8	Título: Verificar la función del botón “Finalizar” en Pasantías.
Descripción: Se busca verificar que a la hora de finalizar una pasantía, el botón funcione correctamente y muestre la pasantía finalizada en la seccion “Finalizadas”.	
Pasos para reproducir: 35. Ir a la página “Pasantías” 36. Presionar el botón “Finalizar” en una pasantía activa. 37. Verificar que la pasantía se finaliza y se muestra en la seccion “Finalizadas”.	
Resultado: El sistema luego de ingresar a la página “Pasantías” y presionar “Finalizar” en una pasantía activa, se despliega una ventana preguntando si queremos confirmar la finalización de la pasantía. Al presionar “Finalizar”, la pasantía	

se finaliza exitosamente y se muestra en la sección "Finalizadas".
Además muestra un botón de emitir informe no funcional.

Estado: Exito

Nota: -

Clasificación de error: -

Cuarta Iteración:

Requerimientos no funcionales:

- 1. El sistema debe ser accesible únicamente desde dispositivos de escritorio, bloqueando el acceso o mostrando un aviso cuando se intente utilizar desde dispositivos móviles.**

El sistema está diseñado para ser utilizado en un entorno institucional donde se espera que el personal trabaje desde computadoras de escritorio. Limitar el acceso desde dispositivos móviles garantiza una mejor experiencia de usuario, evita problemas de compatibilidad con pantallas pequeñas y reduce riesgos de seguridad derivados del uso de dispositivos externos o no controlados. Para su implementación, se utiliza detección del tamaño de la pantalla mediante **media queries en CSS** o validaciones en **JavaScript** para ocultar la aplicación principal y mostrar un mensaje de aviso cuando el sistema se abra en dispositivos móviles.

- 2. El sistema debe permitir el acceso únicamente mediante autenticación con correo electrónico y contraseña previamente definidos, sin admitir otros métodos de autenticación.**

El sistema no está orientado a un uso masivo ni público, sino que será empleado únicamente por personal autorizado de la universidad. Por esta razón, no se requiere un mecanismo complejo de registro de usuarios, sino un acceso controlado y limitado mediante credenciales fijas. Esto evita el ingreso de usuarios no autorizados y simplifica la gestión de la aplicación. La autenticación se implementa comparando el correo y contraseña ingresados con las credenciales previamente definidas en el sistema. Esto puede lograrse mediante un módulo de login en el frontend que valide los datos contra un backend o un archivo seguro. Aunque inicialmente se podría almacenar en memoria o en una base mínima, la mejor práctica es gestionarlo desde un

backend con almacenamiento seguro y cifrado de contraseñas, para reforzar la seguridad.

Quinta Iteración:

Problemas de seguridad

1. Almacenamiento inseguro de credenciales en el frontend

Actualmente el sistema guarda usuario y contraseña en localStorage. Esto es riesgoso, ya que cualquier persona con acceso al navegador puede visualizar o modificar estos datos.

Contra medida: Implementar autenticación basada en **tokens JWT** generados por el backend, con expiración temporal. Los tokens deben almacenarse en **sessionStorage** o, preferentemente, en **cookies seguras (HttpOnly + Secure)**.

Implementación técnica:

- Añadir un **backend con validación de credenciales** (ej. Node.js + Express).
- Integrar librerías como jsonwebtoken para firmar y verificar tokens.
- Modificar el frontend para enviar el token en cada request y validar la sesión en el servidor.

2. Exposición a ataques de Cross-Site Scripting (XSS)

Los formularios de alumnos, empresas o pasantías permiten ingresar datos que podrían contener scripts maliciosos (ejemplo: `<script>alert("hack")</script>`).

Contra medida: Validar y sanitizar todas las entradas de usuario. Además, configurar **Content Security Policy (CSP)** en los headers HTTP para limitar la ejecución de scripts externos.

Implementación técnica:

- Usar librerías de validación como validator.js en el frontend y backend.
- Configurar en el backend cabeceras de seguridad (helmet en Express).
- Revisar que React escape automáticamente HTML y evitar el uso de dangerouslySetInnerHTML.

3. Falta de control de acceso a recursos protegidos

Actualmente, cualquier usuario que conozca la URL de un endpoint puede acceder a datos (ejemplo: /alumnos). No existe verificación de roles ni autenticación en el backend.

Contramedida: Implementar **middleware de autorización** que valide el token JWT y restrinja accesos según el rol del usuario (ej. administrador vs. usuario común).

Implementación técnica:

- Backend con middleware de validación (jwt.verify).
- Definir roles básicos (ej. "Administrador SAU") en la carga útil del token.
- Modificar los endpoints para verificar permisos antes de devolver datos.

4. Falta de protección frente a ataques CSRF (Cross-Site Request Forgery)

Si se usa autenticación con cookies, un atacante podría engañar al navegador para ejecutar acciones en nombre del usuario autenticado.

Contramedida: Implementar **tokens CSRF** únicos por sesión y cabeceras personalizadas que verifique el backend.

Implementación técnica:

- Usar librerías como csurf en Express para generar y validar tokens.
- Incluir el token en formularios o cabeceras al enviar peticiones desde el frontend.
- Asegurar que el backend verifique la validez del token antes de procesar la acción.

5. Transmisión insegura de datos

Si la aplicación no utiliza **HTTPS**, las credenciales, tokens o datos personales de alumnos/empresas pueden ser interceptados mediante ataques “man-in-the-middle”.

Contra medida: Habilitar **TLS/SSL** en el servidor para que todas las comunicaciones sean cifradas.

Implementación técnica:

- Configurar un certificado SSL (ej. gratuito con **Let's Encrypt**) en el servidor de despliegue (Render, Railway o Netlify).
- Forzar redirecciones de `http://` a `https://`.
- Verificar en el frontend que todas las llamadas fetch usen `https://`.