



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Aplicación de Gestión de
Aulas mediante API RESTs
de Microsoft**



Presentado por Jorge Gómez Ortiz
en Universidad de Burgos — 19 de julio
de 2020

Tutor: Jesús Maudes Raedo

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	9
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catalogo de requisitos	14
B.4. Especificación de requisitos	17
Apéndice C Especificación de diseño	33
C.1. Introducción	33
C.2. Diseño de datos	33
C.3. Diseño procedimental	35
C.4. Diseño arquitectónico	36
Apéndice D Documentación técnica de programación	39
D.1. Introducción	39
D.2. Estructura de directorios	39
D.3. Manual del programador	40

D.4. Compilación, instalación y ejecución del proyecto	43
D.5. Pruebas del sistema	51
Apéndice E Documentación de usuario	55
E.1. Introducción	55
E.2. Requisitos de usuarios	55
E.3. Instalación	55
E.4. Manual del usuario	57
Bibliografía	69

Índice de figuras

B.1. Diagrama de casos de uso de un usuario sin permisos.	17
B.2. Diagrama de casos de uso de un usuario responsable.	18
B.3. Diagrama de casos de uso de un usuario administrador.	19
C.1. Diagrama Entidad/Relación del proyecto.	34
C.2. Diagrama de secuencias de las tareas principales del proyecto. .	35
D.1. URL de respuesta de nuestra aplicación	44
D.2. Código de aplicación de Azure Active Directory	44
D.3. Código incluyendo la aplicación de Azure Active Directory . . .	45
D.4. Código para conectar con la BD de Azure	46
D.5. Restricción de direcciones IP para la conexión	47
D.6. Uso compartido y permisos	50
D.7. Compartir o borrar	51
E.1. Página portal de Azure.	56
E.2. Activar app service	56
E.3. Página principal.	58
E.4. Página inicial para ver eventos de aulas.	59
E.5. Formulario para consultar eventos.	59
E.6. Tabla donde se muestran los eventos.	60
E.7. Página inicial para ver eventos de aulas y modificarlos.	61
E.8. Formulario para modificar eventos.	61
E.9. Página inicial para ver eventos de aulas y borrarlos.	62
E.10. Selección de eventos a borrar.	62
E.11. Confirmación de borrado.	63
E.12. Pantalla inicial de la reserva.	63
E.13. Pantalla completa antes de la reserva.	64

E.14. Pantalla inicial de la modificación.	64
E.15. Pantalla completa antes de la modificación.	65
E.16. Pantalla para añadir un aula.	65
E.17. Pantalla para ver la tabla de auditorías.	66
E.18. Pantalla inicial de la modificación de propietario.	67
E.19. Pantalla completa antes de la modificación.	67

Índice de tablas

A.1. Coste Desarrollador	9
A.2. Coste Total	10
A.3. Dependencias del proyecto.	11
B.1. Caso de uso 0: Iniciar sesión.	20
B.2. Caso de uso 1: Consultar eventos de aulas.	21
B.3. Caso de uso 1.1: Modificar eventos.	22
B.4. Caso de uso 1.2: Borrar eventos.	23
B.5. Caso de uso 2: Consultar información sobre las aulas.	24
B.6. Caso de uso 2.1: Modificar aulas.	25
B.7. Caso de uso 2.2: Borrar aulas.	26
B.8. Caso de uso 3: Realizar reservas.	27
B.9. Caso de uso 4: Crear aulas.	28
B.10. Caso de uso 5: Ver auditorías.	29
B.11. Caso de uso 6: Ver/editar propietarios.	30
B.12. Caso de uso 6.1: Crear propietario.	31
D.1. Pruebas manuales propietario	52
D.2. Pruebas manuales administrador	53

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Un plan de proyecto recoge la planificación temporal donde se resumen las divisiones de trabajo en periodos de tiempo y un estudio de viabilidad teniendo en cuenta las repercusiones tanto legales como económicas de dicho proyecto.

A.2. Planificación temporal

Para la planificación temporal se ha usado la plataforma GitHub en la que se ha intentado simular una metodología *Scrum*, en la que las divisiones de tiempo para realizar una determinada función se conocen como *Sprints*. La duración de los Sprints en este caso no era especificada, ya que no se tenían unos plazos obligatorios y algunos sprints duraban más de lo planeado y otros menos. Los primeros Sprints están resumidos en unas *issues* de mayor tamaño ya que se centraban en realizar pruebas y generación del comienzo del proyecto.

Dentro de la plataforma de GitHub se utilizó el tablero de Zen-hub para controlar mejor la inicialización y finalización de las *issues*.

El enlace GitHub del repositorio del presente trabajo puede encontrarse en:

<https://github.com/jgo0038/TFG-Reserva-aulas-informatica>

Sprint 0 - 24/02/2020 - 27/02/2020

Primer sprint en el que se inicia el proyecto, realmente comienza antes el proyecto y la investigación pero hasta entonces no se crea el GitHub. Las tareas en este sprint estaban destinadas a la investigación y la preparación para el proyecto:

- Elegir los programas a utilizar.
- Crear un repositorio en GitHub.
- Generar el entorno virtual de trabajo.
- Registrar la aplicación en Azure Active Directory para obtener un identificador de la aplicación.
- Implementación una primera versión simple del login en nuestra web app.
- Empezar a manejar Overleaf para la documentación el LaTeX.
- Investigar sobre el uso de la API de Outlook para Flask.

Una vez se registró la app en Azure, el objetivo principal era poder iniciar sesión en Outlook con los códigos recibidos, utilizando un repositorio de GitHub como [ayuda](#)[4]. Resaltar que aquí creo la primera rama donde se subieron estos primeros pasos del proyecto.

Sprint 1 - 27/02/2020 - 08/03/2020

En este sprint solamente hubo especificada una tarea, que engloba trabajar con la API y empezar a enviar y recibir datos para conocer su funcionamiento. Ya conociendo la forma en la que iba a trabajar y los requisitos que había que conseguir, se comenzó a hacer pruebas sobre la API solucionando los primeros errores y consiguiendo los primeros avances.

Una vez estaba iniciada la sesión de Outlook el objetivo de este Sprint era leer y modificar eventos del calendario de la sesión. Se sube esto a nueva rama, que permite, a parte del inicio de sesión ya comentado en el Sprint 0, la funcionalidad de poder recoger eventos del calendario principal del usuario que ha iniciado sesión en la web app.

Sprint 2 - 08/03/2020 - 05/04/2020

Los objetivos principales en este sprint fueron los siguientes:

- Investigar y comenzar a trabajar con los calendarios compartidos de Outlook.
- Comenzar a trabajar con la base de datos local de MySQL.
- Investigar sobre la seguridad de la aplicación.

En este Sprint, la tarea de investigar la seguridad era secundaria y no corría prisa de momento, por lo que se relegó a mirarlo más tarde cuando tuviera más funcionalidad y páginas creadas. La tarea principal era el trabajo con los calendarios compartidos, en concreto se buscaba poder crear eventos en calendarios compartidos, que son los que hacen la función de albergar los eventos de cada aula.

Sprint 3 - 26/03/2020 - 15/04/2020

Los objetivos que abordamos en este Sprint se centraron en añadir funcionalidades a la aplicación y estudiar el despliegue:

- Controlar el solapamiento de horas de los eventos en los calendarios ya que no lo realiza Outlook.
- Añadir una página en la que se puedan ver los eventos que hay creados en un determinado calendario.
- Estudiar las posibilidades que hay para desplegar la web app. Se opta por utilizar Azure.

Este Sprint se fue compaginando con el anterior, ya que la tarea de investigar sobre la seguridad de la aplicación había quedado pendiente.

Sprint 4 - 05/04/2020 - 27/04/2020

Este Sprint tenía como objetivos principales:

- Controlar que funcione en un entorno multiusuario, que no se pueda reservar a la vez el mismo aula.

- Crear el primer modelo de las tablas y relaciones en la base de datos local.
- Comprobar que las fechas de inicio y fin coincidan en una reserva simple.
- Ofrecer al cliente una opción para elegir de qué calendario quiere ver las horas reservadas de entre aquellos de los que es responsable.

Este Sprint se da por cerrado pese a haber quedado una tarea abierta (sincronizar la web app con la base de datos, de modo que borre los eventos de la base de datos local y los vuelva a guardar recuperándolos directamente desde Outlook, como solución a algún problema o corrupción de los datos de la base de datos), esta tarea fue desestimada en una reunión posterior. La base de datos y los calendarios de Outlook siempre tienen que estar sincronizados, compartiendo los mismos datos en ambos sitios, de forma que si se borra un evento del calendario, también se borre el correspondiente en la base de datos. Si esto fallara, más adelante no se podrían realizar reservas en fechas que en realidad deberían de estar libres.

Sprint 5 - 17/04/2020 - 07/05/2020

Los objetivos que voy a abordar en este Sprint son los siguientes:

- Realizar un cambio en una relación de la base de datos, permite albergar mas de un propietario a un aula.
- Crear el servidor de Azure database, se crea en lenguaje SQL y sin servidor, para reducir el coste al mínimo.
- Subir la base de datos local al servidor de Azure database creado y adaptar el lenguaje al SQL server.
- Conectar la base de datos subida a Azure con nuestra web app, se realiza mediante pyodbc.
- Organizar el proyecto según el estandar establecido por Microsoft y VSCode para poder subir facilmente a Azure.
- Desplegar la aplicación en Azure.
- Primera asignación y restricción de acceso a páginas según la cuenta que acceda.

Sprint 6 - 07/05/2020 - 15/06/2020

Este Sprint se centra en mejorar partes del código ya existentes y facilitar la reserva de las aulas, concretamente los objetivos fueron:

- Solucionar los errores de los formularios, que no se muestran por pantalla correctamente.
- Añadir unos filtros antes de reservar el aula, limitando las aulas que se muestran. Añadir también un campo para el nombre del profesor que lo reserva.
- Crear una nueva página en la que el administrador pueda crear nuevas aulas.
- Crear una nueva página desde la que se pueda ver la información de cada aula.
- En la página de la información, permitir el modificar los datos de las aulas.
- Restringir los datos que se pueden introducir al modificar aulas en campos como propietario, edificio o tipo de aula.
- En la página de la información, permitir una opción para borrar las aulas.
- Añadir las opciones de reserva múltiple (reservar más de un día seguido a la misma hora) y de reserva periódica (reservar un día a la semana a la misma hora durante el tiempo que se quiera).
- Crear una pagina de auditoría para el usuario administrador, desde la que puedan ver las altas, bajas y modificaciones que se realizan sobre los eventos.

Comentar que se actualiza también la base de datos, aplicando la actualización y borrado en cascada. La fecha de finalización se alargó tanto ya que la tarea de la tabla de auditorías quedó pospuesta, de forma que si quedaba tiempo al final para realizarla se haría, por ello fue terminada el 15/06.

Sprint 7 - 20/05/2020 - 17/06/2020

El Sprint 7 se centra en volver a introducir el código con el que se ha estado trabajando en Azure e ir añadiendo la documentación al proyecto, para ello se fijan estos objetivos:

- Actualizar el repositorio, añadiendo todo tal cual está con los cambios subidos a Azure.
- Añadir la carpeta de documentación siguiendo la plantilla proporcionada.
- Agregar la documentación realizada hasta entonces.
- Añadir los casos de uso.

Sprint 8 - 26/05/2020 - 17/06/2020

Este Sprint se centra en tareas más cortas y más visibles, dando estilos a la página y sus elementos y añadiendo funciones para mejorar el funcionamiento. Las tareas más destacables son las siguientes:

- Añadir botón que permita ver las horas ocupadas del aula a reservar.
- Añadir cabeceras a las páginas.
- Añadir un campo nuevo de 'nº de ordenadores' a las aulas, si no tiene ordenadores se indicará con un 0. Establecer este campo como filtro al reservar aula también.
- Añadir campo de correo electrónico del nombre del profesor al que se reserva el aula y enviarle un mensaje cuando la reserva se realice.
- Representar los privilegios de cada usuario en los botones, de forma que se bloquee si no tiene permiso para esa función.
- Mostrar mensaje al modificar el campo propietario recordando al administrador que tiene que darle los permisos al nuevo propietario manualmente desde el calendario de Outlook.
- Aplicar estilos a los elementos de la página.
- Crear la barra de navegación.
- Implementar el cierre de sesión.

- Diseñar la página de inicio de la aplicación.
- Diseñar la página principal al iniciar sesión.
- Crear la barra de navegación en un fichero aparte del que puedan heredar dicho código.

Anótese que los filtros de búsqueda de capacidad y nº de ordenadores en la página de realizar reserva se atienen al mínimo buscado, no es un búsqueda exacta de esas características, sino se limitaría demasiado.

Sprint 9 - 08/06/2020 - 26/06/2020

Este sprint se crea después de una corrección completa del proyecto por el tutor. Anotando los fallos encontrados, las mejoras a realizar y funcionalidades pendientes:

- Mejorar la página de eventos, dejando más claro su funcionamiento y eliminando el apartado de los filtros, convirtiéndolo en un mismo formulario.
- Añadir función de modificar eventos.
- Añadir función de borrar eventos.
- Arreglar el envío de mensaje al realizar una reserva.
- Añadir posibilidad de que el administrador añada propietarios nuevos.
- Añadir posibilidad al administrador de modificar los propietarios existentes.
- Arreglar la búsqueda de la página de reservas.
- Añadir un botón al consultar los eventos que permita mostrar y ocultar los eventos pasados.
- Arreglar los casos de uso tras la revisión realizada por el tutor.
- Rediseñar la página de consulta de eventos, para que sea mas intuitiva y más útil.
- Añadir una funcionalidad para que se puedan borrar más de un evento a la vez.
- Arreglar o mejorar pequeños detalles.

Sprint 10 - 18/06/2020 - 26/06/2020

Realizar los últimos archivos de documentación que quedan por rellenar y completar o modificar algunos ya realizados.

- Realizar el apartado de trabajos relacionados.
- Realizar el manual del programador.
- Realizar el manual del usuario.
- Mejoras sobre la documentación tras revisión del tutor.

Sprint 11 - 01/07/2020 - 16/07/2020

Completar la aplicación con mejoras visuales y de funcionalidad.

- Añadir una opción de seleccionar todos los eventos de la tabla para borrarlos a la vez.
- Añadir un filtro de fechas para la búsqueda en la tabla de auditorías.
- Permitir acceder a la aplicación como un usuario sin registrar y poder realizar la consulta de aulas y de eventos.
- Mejorar la pantalla de ayuda del mensaje al realizar acciones sobre el propietario.
- Permitir una búsqueda de un rango de fechas en la visualización de los eventos.
- Añadir más información al email que se envía al crear una reserva.
- Arreglar la reserva periódica que fallaba en algunos casos concretos.
- Actualizar la suscripción de Azure al haberse finalizado la de estudiante.
- Arreglar la consultas para ver las horas reservadas.
- Arreglar la modificación de la reserva al cambiar la hora a una que coincide con esta misma reserva.

A.3. Estudio de viabilidad

En esta sección trataré la viabilidad del proyecto, tanto en el ámbito económico como legal.

Viabilidad económica

En este apartado trataré de alcanzar el precio total del proyecto si fuera desarrollado en una empresa o comprado por alguien.

Coste de personal

Para realizar este proyecto, se ha necesitado un desarrollador de aplicaciones web para lograrlo. Por lo tanto, el coste del personal se estima en lo siguiente:

Desarrollador

Suponiendo que acaba de comenzar su carrera como desarrollador web, el sueldo medio está entre 17.000 y 22.000. Suponemos un término medio de 20.000 para realizar los cálculos. Lo que supone a la empresa:

$$20,000 \div 12 = 1667 \text{bruto} \quad (\text{A.1})$$

El trabajo se ha realizado de finales de Febrero a finales de Junio, lo que suponen 4 meses.

Concepto	Coste mensual
Salario mensual	1667
Costes seguridad social (23,60 %)	393
Cotización por formación profesional (0'6 %)	10
Coste total (4 meses)	7071

Tabla A.1: Coste Desarrollador

Coste hardware

El coste del hardware utilizado se reduce al coste del portátil usado para desarrollar el proyecto. El precio del ordenador portátil (Asus X541U) con IVA fuese de 600, el desarrollo se estima que durará 4 meses, y se supone una amortización de este mismo de 4 años. Por lo que el precio usado del portátil sería:

$$(600 \div (12 * 4)) * 4 = 50 \quad (A.2)$$

Coste del despliegue

El despliegue se ha realizado en Azure, donde se han ido realizando llamadas a la base de datos y ejecutando la aplicación subida. Esto se realizó con una suscripción a la cuenta de estudiante de Azure en la que se proporciona 100USD gratuitos. El uso de esta aplicación web para pruebas y mejoras ha supuesto un gasto de 100USD de los 100 disponibles, por lo que se agotó la suscripción de estudiante y hubo que mejorarla a una suscripción de pago por uso. Por esta razón el gasto en este caso ha sido 110USD, teniendo en cuenta que ha sido durante 4 meses, el coste mensual se estima en 27.5USD. Pero este coste se aplicaría a una empresa durante este período de desarrollo. Si se quisiera mantener después aumentaría dependiendo del uso que se hiciera de esta aplicación y del plan de pago que se eligiera. Esta cantidad de 110USD pasado a euros a fecha de 17/07/2020 hace un total de 96.28 euros.

Coste total

Concepto	Coste
Coste desarrollador	7071
Hardware	50
Despliegue	96.28
Internet	26*4
Coste total (4 meses)	7321.28

Tabla A.2: Coste Total

Viabilidad legal

En cuanto al código, todo lo utilizado en este proyecto es de libre distribución y dominio público. Se ha utilizado Flask y librerías de Python públicas.

En cuanto a los programas utilizados, todos son públicos también.

Dependencia	Licencia
astroid	LGPL
click	BSD
colorama	BSD
distlib	Python Software Foundation License
dominate	LGPLv3
filelock	Public Domain
Flask	BSD
Flask-Bootstrap	BSD
Flask-SQLAlchemy	BSD
Flask-WTF	BSD
importlib-metadata	Apache Software License
importlib-resources	Apache Software License
isort	MIT
itsdangerous	BSD
Jinja2	BSD
MarkupSafe	BSD
mysqlclient	GPL
pylint	GPL
pyodbc	MIT
requests	Apache 2
six	MIT
SQLAlchemy	MIT
virtualenv	MIT
visitor	MIT
Werkzeug	BSD
wrapt	BSD
WTForms	BSD
zipp	MIT

Tabla A.3: Dependencias del proyecto.

Una vez se han visto las licencias a las que están sometidas las dependencias del proyecto, se ha seguido las recomendaciones de [3] para determinar que este proyecto está bajo la licencia GPL-3.0, ya que es la última versión de la licencia GPL sobre software libre con *copyleft*, lo que permite que todas las versiones sean libres.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apartado de la documentación explicaré los objetivos que se perseguían al comienzo del proyecto, los requisitos funcionales y no funcionales.

B.2. Objetivos generales

- Desarrollar una aplicación web cliente servidor, en la que los usuarios con permisos necesarios puedan reservar aulas para los profesores.
- Ofrecer la posibilidad de reservar un único día, más de uno seguidos o un día de la semana durante un periodo de tiempo, un mismo aula.
- Ofrecer al usuario una vista sobre las aulas existentes, si es administrador podrá modificarlas o borrarlas también.
- Ofrecer al usuario una vista con las franjas horarias reservadas y libres de cada aula.
- Controlar el solapamiento de horas.
- Ofrecer un control al administrador sobre las reservas y la información de las aulas.
- Controlar el acceso de usuarios mediante el inicio de sesión de Microsoft.
- Que las reservas queden registradas en el calendario de Microsoft.

- Que las reservas queden sincronizadas con la base de datos.
- Que las altas bajas y modificaciones de reservas sean rastreables por el administrador mediante un sistema de auditoría.

B.3. Catalogo de requisitos

En esta sección enumeraré los requisitos funcionales de la aplicación y los usuarios que intervienen.

Propietario: Centro o Departamento universitario que tiene asignado el control y mantenimiento de una serie de aulas. **Usuario:** será quién use la aplicación, según los privilegios les hay de tres tipos:

- Sin privilegios / sin registrar: Puede consultar todas las aulas y su ocupación.
- Responsable: Persona designada por un propietario para que pueda realizar, modificar y borrar reservas sobre las aulas asignadas al mismo.
- Administrador: Puede consultar todas las aulas y su ocupación. Además puede realizar reservas sobre todas las aulas. Puede modificar la información de las aulas o borrarlas. Se encarga de asignar aulas en propiedad a los propietarios.

Requisitos funcionales

- **RF.1** Controlar el acceso mediante un inicio de sesión.
 - **RF.1.1** Acceso a la aplicación mediante el login de Outlook.
- **RF.2** La aplicación tiene que poder distinguir entre el tipo de usuario que ha accedido. Distingue cuatro tipos:
 - **RF.2.1** Diferencia los usuarios sin permisos.
 - **RF.2.2** Diferencia los usuarios responsables de aulas.
 - **RF.2.3** Diferencia los usuarios administradores.
 - **RF.2.4** Diferencia los usuarios invitados que no inician sesión.
- **RF.3** La aplicación tiene que poder mostrar información sobre las aulas.
 - **RF.3.1** Permite ver el listado de aulas.

- **RF.3.2** Permite ver características de cada aula.
- **RF.4** La aplicación tiene que poder modificar las aulas.
 - **RF.4.1** Permite modificar las características de las aulas.
 - **RF.4.2** Permite el borrado de aulas.
- **RF.5** La aplicación tiene que poder realizar una reserva de tres tipos y reflejar de forma sincronizada esas reservas en el calendario de Outlook.
 - **RF.5.1** Permite reservar un sólo día de una hora a otra especificada.
 - **RF.5.2** Permite reservar varios seguidos a la misma hora.
 - **RF.5.3** Permite reservar un día a la semana a la misma hora durante el tiempo especificado.
- **RF.6** La aplicación tiene que poder mostrar todos los eventos creados.
 - **RF.6.1** Permite modificar los eventos y reflejar de forma sincronizada esas reservas en el calendario de Outlook.
 - **RF.6.2** Permite borrar los eventos y eliminar de forma sincronizada esas reservas en el calendario de Outlook.
- **RF.7** La aplicación tiene que poder crear un aula nueva para un manejo posterior.
- **RF.8** La aplicación tiene que permitir al administrador ver una tabla de auditoría con las altas, bajas y modificaciones que se realizan en las reservas.
- **RF.9** La aplicación tiene que poder crear y modificar los propietarios y su información.

Requisitos no funcionales

- **RNF.1 Usabilidad** Proporcionar una página web sencilla de usar e intuitiva.
- **RNF.2 Disponibilidad** El tiempo para iniciar o reiniciar el sistema no es mayor a 5 minutos.

- **RNF.3 Seguridad** Se controla que no todos los usuarios puedan acceder sin haber iniciado sesión a realizar modificaciones y una vez iniciada sesión, se controla el acceso a las distintas partes de la aplicación según los privilegios.
- **RNF.4 Mantenibilidad** Es sencillo añadir nuevas funcionalidades.
- **RNF.5 Sincronización con Outlook** Las reservas de las aulas tendrán que estar sincronizadas en la base de datos como en el calendario de Outlook.

B.4. Especificación de requisitos

Diagrama de casos de uso

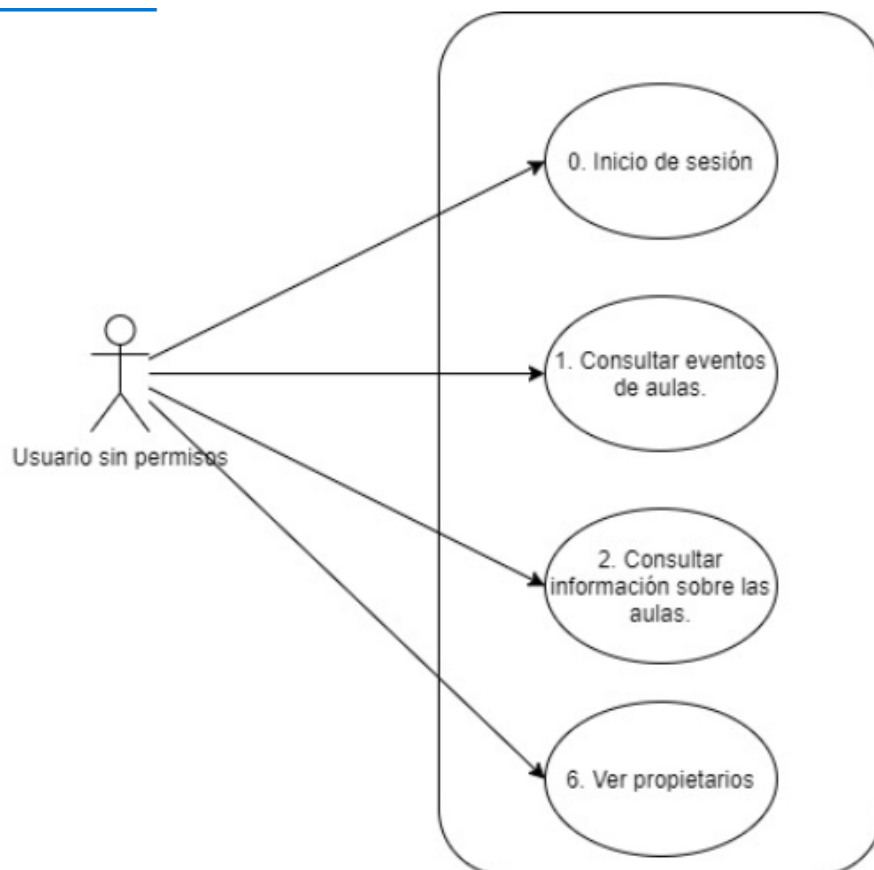


Figura B.1: Diagrama de casos de uso de un usuario sin permisos.

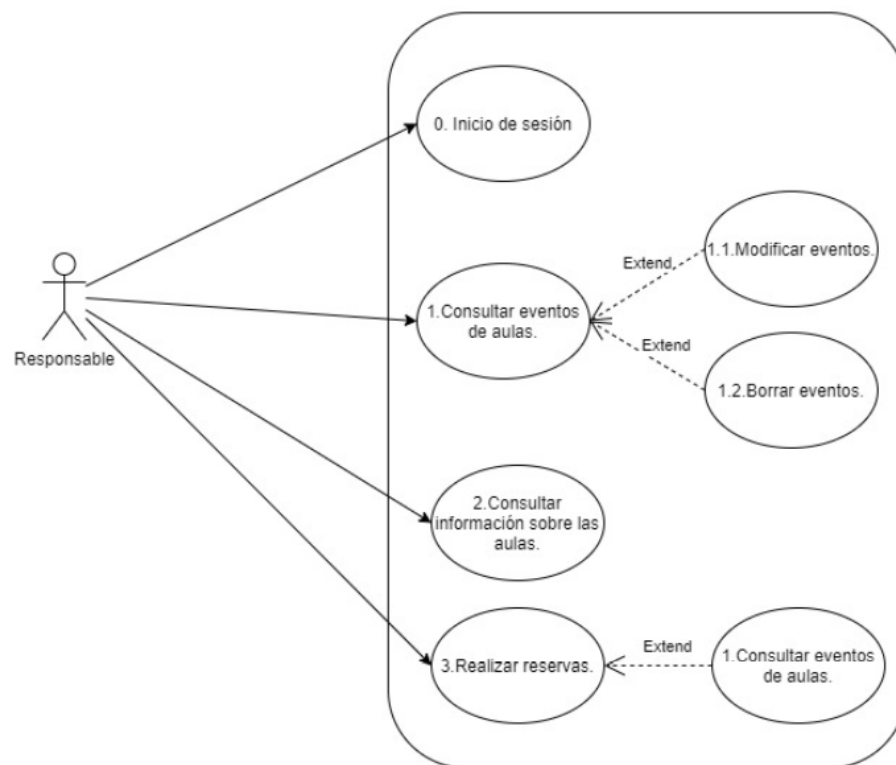


Figura B.2: Diagrama de casos de uso de un usuario responsable.



Figura B.3: Diagrama de casos de uso de un usuario administrador.

Especificación de los casos de uso

Caso de uso 0: Iniciar sesión.		
Descripción	Permite al usuario iniciar sesión en la aplicación.	
Requisitos	RF-1	
	RF-1.1	
	RF-2	
	RF-2.1	
	RF-2.2	
	RF-2.3	
Precondiciones	Disponer de una cuenta de Outlook.	
Secuencia normal	Paso	Acción
	1	El usuario pincha el botón 'Iniciar sesión'.
	2	Se redirige al inicio de sesión de Outlook.
	3	Se introducen los datos de inicio de sesión de la cuenta de Outlook.
	4	Se inicia la sesión si existe la cuenta.
Postcondiciones	Si la sesión se ha iniciado correctamente, se lleva al usuario a la página de inicio. Si no se vuelve a pedir que introduzca una cuenta existente.	
Excepciones	Cuenta de Outlook inexistente.	

Tabla B.1: Caso de uso 0: Iniciar sesión.

Caso de uso 1: Consultar eventos de aulas.		
Descripción	Permite al usuario visualizar los eventos creados en las aulas, facilitando filtros para una búsqueda mas concreta. Permite consultar la disponibilidad de aulas antes de llevar a cabo una petición de reserva al responsable.	
Requisitos	RF-6	
Secuencia normal	Paso	Acción
	1	El usuario pincha el botón 'Consultar ocupación de aulas' de la barra superior de navegación o el menú principal.
	2	Se redirige a una página donde se muestra un desplegable para elegir el edificio. Con un botón de 'Mostrar aulas'.
	3	Se elige el edificio y se pincha en 'Mostrar aulas', aparece entonces un desplegable con las aulas de éste y una primera opción 'Cualquiera'. Debajo tenemos unos cuantos campos para filtrar las aulas por características si elegimos la opción 'Cualquiera' anteriormente.
	4	Si elegimos un aula en concreto solamente se tendrán en cuenta los campos de fecha y hora obligatorios.
	5	Si no se especifica capacidad ni ordenadores, se mostrarán de todas las capacidades y ordenadores, si se especifican se mostrarán eventos con igual o mayor número.
	6	Una vez se ha rellenado el formulario al menos con los campos de fecha y horas se puede pinchar en el botón 'Enviar consulta' para consultar los eventos que coincidan con los especificado anteriormente.
	7	Se muestra la tabla con los eventos futuros que cumplan las especificaciones, mostrando detalles sobre ellos. Existe un botón en la parte inferior de 'Mostrar eventos pasados' que mostrará los eventos pasados coincidentes en color rojo si en la fecha de inicio se puso una anterior a la actual. Una vez mostrados se puede volver a ocultar del botón 'Ocultar eventos pasados'.
Postcondiciones	Permite la visualización de los eventos según el aula elegido, depende de los privilegios del usuario dispondrá de las opciones de modificar (CU1.1) y borrar estos eventos (CU1.2).	
Excepciones	Elegir un aula en concreto en el paso 3. En ese caso ejecutar solo pasos 4, 6 y 7	

Tabla B.2: Caso de uso 1: Consultar eventos de aulas.

Caso de uso 1.1: Modificar eventos.	
Descripción	Permite al usuario modificar los eventos existentes.
Requisitos	RF-6
	RF-6.1
Precondiciones	Haber iniciado sesión (CU0) como administrador o propietario, y haber ejecutado el una consulta de eventos de aulas (CU1)
Secuencia normal	Paso Acción
	1 Se muestra la tabla con los eventos, especificando detalles sobre ellos.
	2 Se ve un botón de 'Modificar' al lado de los eventos sobre los que tenga permiso para modificar.
	3 Se pincha en el botón 'Modificar' y se abre un formulario en la parte superior para cambiar los datos del evento.
	4 Una vez modificados los datos, se pincha sobre el botón 'Guardar' y se realizan las modificaciones.
	5 Si se modifica correctamente, se vuelve a la página anterior.
Postcondiciones	El evento ha sido modificado, y la fecha, horas y aula tras la modificación son compatibles con el resto de eventos. Además la modificación está registrada de forma sincronizada en la BD y en Outlook.
Excepciones	Cambiar a una fecha ya reservada.

Tabla B.3: Caso de uso 1.1: Modificar eventos.

Caso de uso 1.2: Borrar eventos.	
Descripción	Permite al usuario borrar los eventos existentes.
Requisitos	RF-6
	RF-6.2
Precondiciones	Haber iniciado sesión (CU0) como administrador o propietario, y haber ejecutado el una consulta de eventos de aulas (CU1)
Secuencia normal	Paso Acción
	1 Se muestra la tabla con los eventos, especificando detalles sobre ellos.
	2 Se ve un botón de 'Borrar' al lado de los eventos sobre los que tenga permiso para borrar.
	3 Se pincha en el botón 'Borrar' y te pide la confirmación del usuario para borrar el evento.
Postcondiciones	Si se borra correctamente, se muestra un mensaje por pantalla y se vuelve a la página anterior. Si se cancela el borrado te mantienes en la misma.

Tabla B.4: Caso de uso 1.2: Borrar eventos.

Caso de uso 2: Consultar información sobre las aulas.		
Descripción	Permite al usuario conocer la información relevante acerca de las aulas existentes, sea o no responsable de ellas. La información que se facilita es: nombre, edificio, tipo de aula, capacidad, n° de ordenadores y propietario.	
Requisitos	RF-3	
	RF-3.1	
	RF-3.2	
Precondiciones	Haber iniciado sesión.	
Secuencia normal	Paso	Acción
	1	El usuario pincha el botón 'Ver aulas' de la barra superior de navegación.
	2	Se redirige a la página de información sobre las aulas, donde se muestra una opción para elegir el grupo de aulas de entre todos los existentes.
	3	Una vez se elige el grupo de aulas, se pincha en el botón 'Ver aulas'.
	4	Se muestra una tabla con las aulas y su información del edificio elegido en el paso anterior.
Postcondiciones	Permite la visualización de las características de las aulas, depende de los privilegios del usuario dispondrá de las opciones de modificar (CU2.1) y borrar (CU2.2) estas aulas.	

Tabla B.5: Caso de uso 2: Consultar información sobre las aulas.

Caso de uso 2.1: Modificar aulas.	
Descripción	Permite al usuario modificar la información relevante acerca de las aulas existentes.
Requisitos	RF-4 RF-4.1
Precondiciones	Haber iniciado sesión (CU0) como administrador o propietario, y haber ejecutado el una consulta de aulas (CU1)
Secuencia normal	Paso Acción
	1 Se ve un botón de 'Editar' al lado de la información de las aulas que pueda modificar.
	2 Al pinchar sobre éste botón se abre un cuadro en la parte superior de la pantalla donde se pueden introducir los valores a modificar.
	3 Una vez se han modificado los datos correspondientes se pincha en el botón 'Guardar' y se actualizan los datos.
	4 Si se cambia el propietario, se mostrará un mensaje recordando el cambio manual de propietario que tiene que hacer el administrador.
	5 Si se cambia correctamente, se vuelve a la pantalla anterior y se muestra un mensaje por pantalla.
Postcondiciones	El cambio en el aula han sido registrado en la base de datos.
	El cambio de propietario en Outlook queda pendiente de hacer manualmente.
Excepciones	Pinchar en el botón de 'Cancelar', con lo que se cancelan los cambios.
	Dejar sin rellenar partes del formulario de modificación.

Tabla B.6: Caso de uso 2.1: Modificar aulas.

Caso de uso 2.2: Borrar aulas.	
Descripción	Permite al usuario borrar aulas existentes.
Requisitos	RF-4.2
Precondiciones	Haber iniciado sesión (CU0) como administrador o propietario, y haber ejecutado el una consulta de aulas (CU1)
Secuencia normal	Paso Acción
	1 Se muestra una tabla con las aulas y su información del edificio elegido en el paso anterior.
	2 Se ve un botón de 'Borrar' al lado de la información de las aulas que pueda eliminar.
	3 Al pinchar sobre éste botón se abre un cuadro pidiendo confirmación del usuario para borrar el aula. Ofreciendo una comprobación en la que se permite borrar definitivamente o cancelar el borrado.
	4 Si se borra el aula,se muestra un mensaje por pantalla avisando del borrado y se vuelve al formulario anterior.
Postcondiciones	El borrado del aula ha sido registrado en la base de datos y en el calendario de Outlook.
Excepciones	Pinchar en el botón de cancelar a la hora de la confirmación.

Tabla B.7: Caso de uso 2.2: Borrar aulas.

Caso de uso 3: Realizar reservas.	
Descripción	Permite al usuario realizar reservas sobre las aulas que tenga permisos siempre que la hora no esté ocupada.
Requisitos	RF-5
	RF-5.1
	RF-5.2
	RF-5.3
Precondiciones	Haberse logueado como administrador o responsable (CU0).
Secuencia normal	Paso Acción
	1 El usuario (administrador o propietario) pincha el botón 'Realizar reserva' de la barra superior de navegación o el menú principal.
	2 Se muestra un formulario en el que se aprecian 3 divisiones. En el primero se piden la capacidad, nº de ordenadores, edificio y tipo de aula.
	3 Hay que pinchar en el botón 'Consultar' para continuar y se llena el campo 'seleccionar aula' con las opciones según lo especificado al comienzo, si hay más de un aula se añade una opción de 'Cualquiera' a la lista de aulas disponibles.
	4 En la segunda división del cuestionario, se ofrece la posibilidad de cambiar el tipo de reserva. La opción de pinchar en el botón 'Aplicar' conlleva que cambie el número de casillas a rellenar dependiendo de la opción que se haya elegido.
	5 Si se elige 'un día' se queda igual. Si se elige la opción 'días contiguos' se añaden el campo 'Fecha fin'. Si se elige la opción 'Un día por semana' se añaden los campos 'Selecciona día de la semana' y 'Fecha fin'
	6 Si se pincha en el botón 'Consultar ocupación de aulas', se muestran las horas reservadas del aula elegida, si la elección es 'Cualquiera', se muestran los eventos de todas las aulas del desplegable.
	7 Si se pincha en el botón 'Reservar' se realiza la reserva la hora y aula pedida si están libres.
	8 Se permite volver a reservar otro aula recordando la reserva recién realizada por si se quiere una mínima modificación.
	9 Si se realiza correctamente la reserva, se envía un email al profesor que se le ha indicado en el formulario avisando de la reserva realizada.
Postcondiciones	La reserva ha sido registrada en la base de datos y en Outlook.
Excepciones	No existan aulas con los requisitos específicos, con lo que se muestra un mensaje por pantalla indicándolo.

Caso de uso 4: Crear aulas.	
Descripción	Permite al usuario administrador crear aulas nuevas.
Requisitos	RF-7
Precondiciones	Haberse logueado como un usuario administrador.
Secuencia normal	Paso Acción
	1 El usuario (administrador) pincha el botón 'Añadir aulas' de la barra superior de navegación.
	2 Se muestra un formulario en el que se puede seleccionar el edificio donde se ubica el aula, el tipo de aula y el propietario. Se dejan campos a rellenar para proporcionar el nombre del aula, la capacidad y el n° de ordenadores.
	3 Se pincha en el botón 'Crear aula'.
	4 Asignar manualmente los permisos desde Outlook al propietario del nuevo aula.
Postcondiciones	El aula está creada en la base de datos y en el calendario de Outlook, pero pendiente de asignarla permisos a sus responsables desde Outlook.
Excepciones	No llenar todos los campos requeridos

Tabla B.9: Caso de uso 4: Crear aulas.

Caso de uso 5: Ver auditorías.		
Descripción	Permite al usuario administrador ver las altas, bajas y modificaciones realizadas sobre cualquier reserva.	
Requisitos	RF-8	
Precondiciones	Haberse logueado como un usuario administrador.	
Secuencia normal	Paso	Acción
	1	El usuario (administrador) pincha el botón 'Ver auditorías' de la barra superior de navegación.
	2	Se muestra una tabla en la que figuran las acciones sobre las reservas, ya sea una alta, baja o modificación. Se especifica el usuario y la fecha de dicha acción.

Tabla B.10: Caso de uso 5: Ver auditorías.

Caso de uso 6: Crear/editar propietarios.	
Descripción	Permite al usuario administrador ver y modificar la información de los propietarios existentes.
Requisitos	RF-9
Precondiciones	Haberse logueado como un usuario administrador.
Secuencia normal	Paso Acción
	1 El usuario (administrador) pincha el botón 'Ver/Editar propietarios' de la barra superior de navegación.
	2 Se muestra una tabla en la que figuran los propietarios existentes con su información relevante, incluyendo el responsable.
	3 Se da la opción de modificar la información de cada propietario pinchando en el botón 'Modificar' al lado de cada uno.
	4 Se muestra un formulario en el que se permite editar los campos del propietario a modificar .
	5 Se visualizan los cambios realizados.
Postcondiciones	Se actualiza la BD con el nuevo propietario y/o las modificaciones de sus datos.

Tabla B.11: Caso de uso 6: Ver/editar propietarios.

Caso de uso 6.1: Crear propietario.	
Descripción	Permite al usuario administrador crear un nuevo propietario.
Requisitos	RF-9
Precondiciones	Haber ejecutado el CU6 (Ver/editar propietarios).
Secuencia normal	Paso Acción
	1 Se pincha en la opción 'Crear' y el propietario está disponible para ser creado.
	2 Se introducen los datos que se quieran.
	3 Se pincha en el botón de 'Crear' para confirmar los cambios.
	4 Se visualizan los cambios realizados.
Postcondiciones	El nuevo propietario es insertado en la base de datos.
Excepciones	Si se omite algún campo, no se permite la creación y se indica el campo en cuestión necesario.

Tabla B.12: Caso de uso 6.1: Crear propietario.

Apéndice C

Especificación de diseño

C.1. Introducción

En esta sección será descrito el diseño que se ha llevado a cabo en la aplicación, especificando como se han resuelto los objetivos mas relevantes. En concreto, en los siguientes apartados se definirán los datos que maneja la aplicación, los detalles procedimentales y el diseño arquitectónico.

C.2. Diseño de datos

En este apartado se explicarán el conjunto de datos que se utilizan en la aplicación y mostraré el diagrama entidad relación seguido en la aplicación para mostrar la conexión entre los conjuntos de datos.

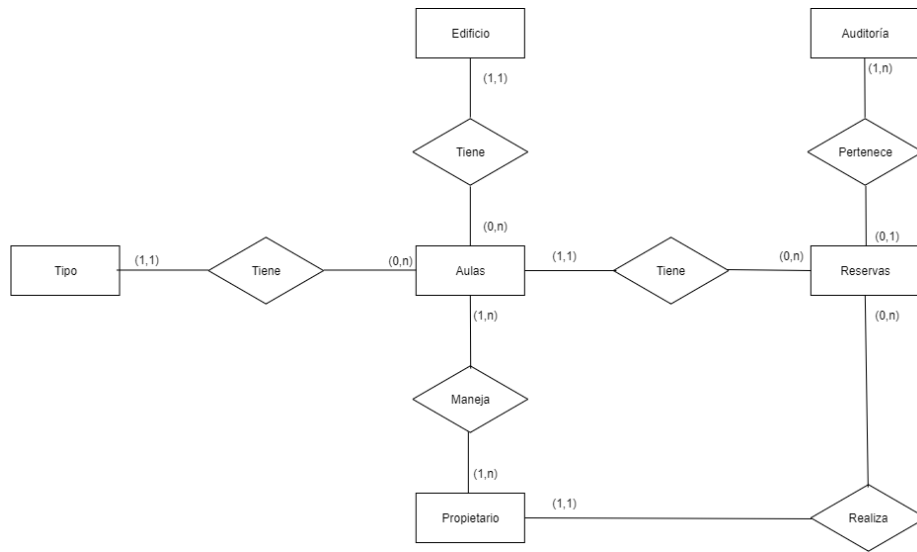


Figura C.1: Diagrama Entidad/Relación del proyecto.

Edificios

Agrupación de aulas que se encuentran en el mismo edificio. En nuestra aplicación se representa como un grupo de calendarios compartido de Outlook gestionados por el administrador.

Aulas

Entidad sobre la que se realizan las reservas, se ubican en un edificio y podrían tener más de un propietario asignado. En nuestra aplicación las aulas se representan como calendarios compartidos dentro de un grupo de calendarios (edificio) sobre los que se va a poder realizar una reserva.

Reservas

Fecha del calendario entre unas horas específicas que se reserva para un usuario, la reserva es única en cada calendario. En nuestra aplicación se representa como un evento creado dentro de un aula (calendario) especificado, al ser único no puede haber más de un evento (reserva) en el mismo calendario a la misma hora o en horas que se solapen, pero sí puede haber eventos con el mismo nombre.

Propietarios

Se utiliza para designar indistintamente a un centro o a un departamento que podrá realizar modificaciones sobre sus aulas asignadas. Cada propietario tendrá un único responsable, éste será un usuario registrado con Outlook que tenga permisos para realizar reservas y modificaciones sobre las reservas de aulas de las que son propietarios.

Auditoría

Tabla en la que se registrarán las altas, bajas y modificaciones de las reservas. Además se guardará información sobre dichas alteraciones en la reserva, será solo visible para el administrador.

C.3. Diseño procedimental

En este apartado se mostrarán los diagramas de secuencias de las interacciones más destacadas de la aplicación.

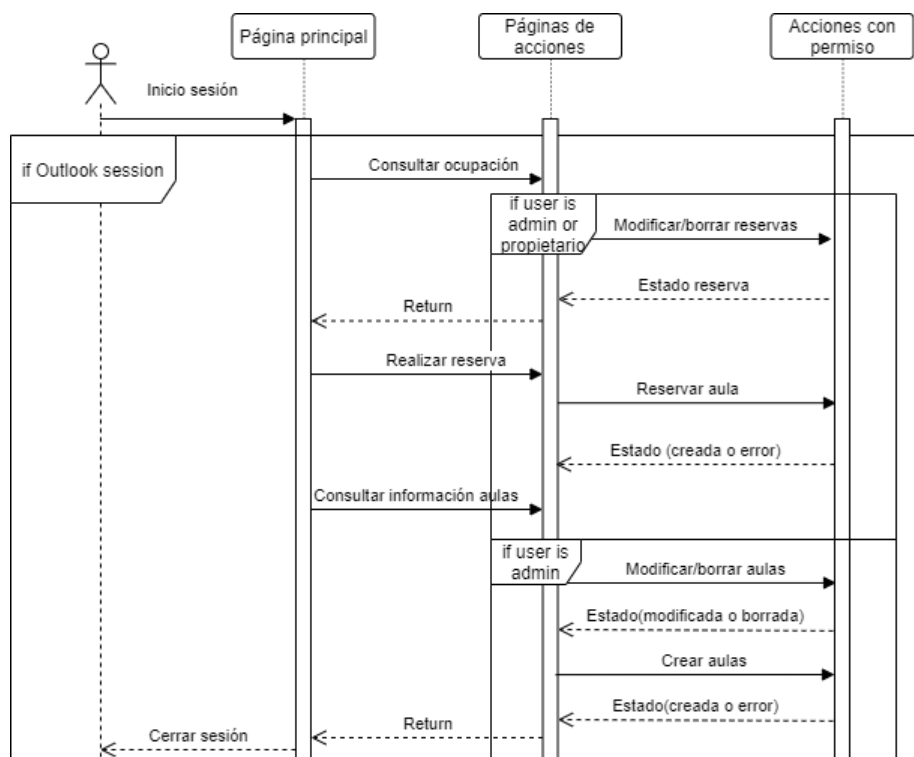


Figura C.2: Diagrama de secuencias de las tareas principales del proyecto.

En el diagrama superior, se puede apreciar las interacciones que se pueden realizar desde la aplicación, aquí cabe resaltar que las transacciones de modificar/borrar y crear reservas trabajan sobre la base de datos y sobre los calendarios de Outlook, por lo que se sincronizan ambas cada vez que se realiza una acción de las mencionadas.

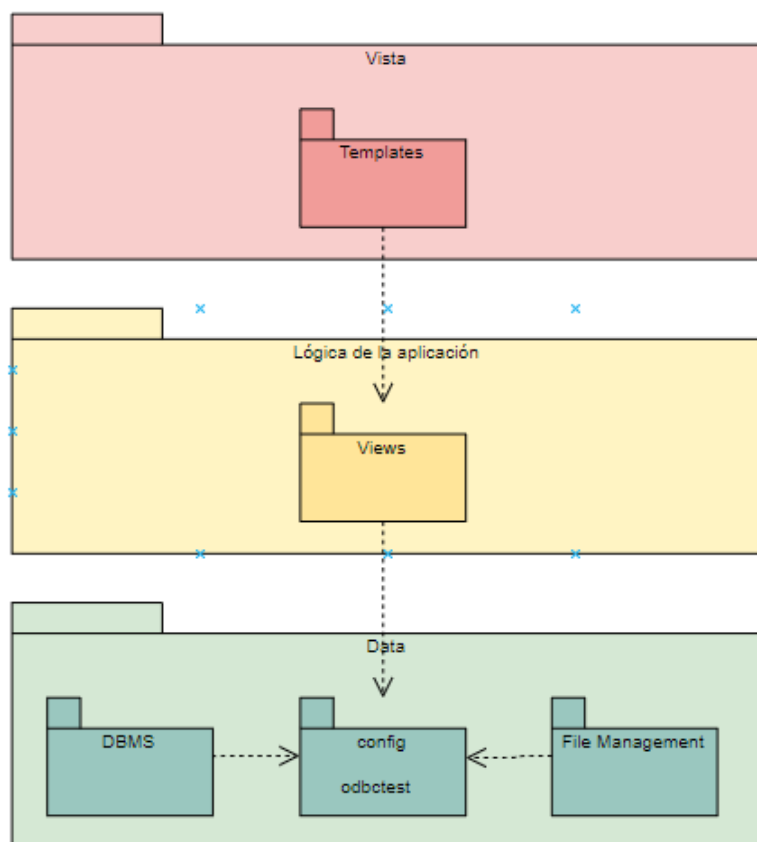
C.4. Diseño arquitectónico

En este apartado se mostrará como está estructurado el código de la aplicación. Al tratarse de una web app se distinguen tres grandes agrupaciones siguiendo el patrón MVC (Modelo - Vista - Controlador). Según este patrón vamos a distinguir en la estructura de los datos el modelo de datos, la interfaz y la lógica de la aplicación en el controlador.

En este proyecto en concreto, podemos distinguir los siguientes tipos de ficheros que realizan operaciones con distintos elementos externos:

- **templates:** Se comunica con los usuarios que acceden a la aplicación mostrándoles la interfaz.
- **views:** Responde a los eventos que realiza un usuario sobre la vista trabajando con los datos recibidos.
- **oauth_helpers:** Conexión con la API REST de Outlook para obtener y enviar datos.
- **getSQLData:** Se comunica con la base de datos para obtener datos de ella.
- **config:** Realiza la conexión necesaria para llevar a cabo la validación de Outlook y los permisos proporcionados.

Entre estos ficheros que acabamos de comentar se distinguen las siguientes dependencias:



La vista, que es la parte de la interfaz se basa en los templates. Los templates dependen del fichero views, que es el que controla la lógica y envía los datos a los templates para mostrar unas u otras cosas y permitir distintos accesos por ejemplo. El views depende de los templates, que le pasan la información que introduce el usuario por pantalla y también depende de los datos recibidos de la base de datos y de los calendarios de Outlook. Las conexiones con el SGBD y con Outlook, a su vez, dependen del fichero odbctest y config respectivamente para poder establecer la conexión.

Apéndice *D*

Documentación técnica de programación

D.1. Introducción

En esta sección describiré como está estructurado el proyecto, que es necesario para poder ejecutarlo desde cualquier ordenador y lo más relevante a la hora de crearlo.

D.2. Estructura de directorios

La estructura de los ficheros alojados en el [repositorio de GitHub](#) es la siguiente:

- **ReservaAulas:** Contiene el proyecto y el entorno virtual y configuraciones.
 - **.vscode:** Contiene la configuración para ejecutar en *debug* desde *VSCode*.
 - **azureEnv:** Entorno virtual creado para el proyecto.
 - **reservaAulas_app:** Contiene el proyecto principal. Dentro se encuentran los ficheros que contienen la funcionalidad del proyecto. También alberga las conexiones con la base de datos, con Azure y con la API.
 - **static:** Contiene imágenes, bibliotecas y ficheros javascript que se usan en el proyecto.

- ◊ **css:** Estilos de Bootstrap y algunos propios para HTML.
- ◊ **img:** Imágenes del proyecto.
- ◊ **js:** Ficheros JavaScript de Bootstrap.
- **templates:** Contiene los ficheros de la parte de la vista del proyecto.
- **requirements.txt:** Contiene los requisitos a instalar dentro del entorno virtual para el funcionamiento del proyecto.
- **startup:** Fichero para hacer el deploy a Azure.

D.3. Manual del programador

Estructura del proyecto

En esta sección se explican las distintas entidades del proyecto y con qué finalidad se han creado. Se trata de explicar los ficheros del proyecto desde el punto de vista del programador para que se entienda el código para futuras mejoras o análisis del mismo. A continuación enumero y explico los ficheros usados:

- **reservaAulas/azureEnv:** Entorno virtual creado para instalar las dependencias necesarias del proyecto.
- **reservaAulas/reservaAulas__app:** Contiene todo el código del proyecto.
 - **static:** Contiene las carpetas de 'css', 'img' y 'js'. En ellas se guardan los estilos de *Bootstrap*, las imágenes que utilizamos y los archivos *JavaScript* necesarios para *Bootstrap* o alguno propio, respectivamente.
 - **templates:** Contiene el código *HTML*, *Jinja2* y *JavaScript* que crean las interfaces que verá el usuario.
 - **reservaAulas__app:** Contiene el código que maneja la lógica de la aplicación, proporciona y recoge los datos desde la API y base de datos.
- **reservaAulas/.deployment:** fichero generado al realizar el deploy en Azure.
- **log_web__app.log:** fichero generado para encontrar errores al hacer el deploy.

- **reservaAulas/createTablesSQLServer.sql**: fichero con el código correspondiente para generar las tablas y relaciones entre ellas necesarias para el proyecto.
- **reservaAulas/requirements.txt**: fichero generado con las bibliotecas necesarias a instalar en el entorno virtual para que funcione la aplicación.
- **reservaAulas/startup.py**: fichero generado para ejecutar desde Azure.
- **reservaAulas/.vscode/launch.json**: configuración para ejecutar el proyecto desde el VSCode en un entorno virtual.
- **reservaAulas/reservaAulas__app/___init___py**: creación de la app de Flask e instancia de la base de datos y el cifrado csrf.
- **reservaAulas/reservaAulas__app/config.py**: configuración de la aplicación con las claves iniciales de Azure para el uso de la API de Outlook y creación de los permisos.
- **reservaAulas/reservaAulas__app/forms.py**: creación de los formularios que se utilizan en las páginas de la aplicación.
- **reservaAulas/reservaAulas__app/getSQLData.py**: fichero con llamadas a la base de datos para ahorrar código en el fichero views.py.
- **reservaAulas/reservaAulas__app/models.py**: modelos utilizados en base de datos en local.
- **reservaAulas/reservaAulas__app/oauth_helpers.py**: funciones que llaman a la API de Outlook para recoger, enviar, borrar o actualizar datos.
- **reservaAulas/reservaAulas__app/odbctest**: conexión mediante pyodbc con la base de datos subida a Azure.
- **reservaAulas/reservaAulas__app/views.py**: fichero que contiene la funcionalidad del servidor, llama a las funciones de otros ficheros para obtener datos y los envía al cliente.
- **reservaAulas/reservaAulas__app/webapp.py**: fichero creado para la creación de la estructura pedida por VSCode y Azure.
- **reservaAulas/reservaAulas__app/templates/anadirAulas.html**: fichero que contiene lo que verá el cliente a la hora de añadir un aula.

- **reservaAulas/reservaAulas__app/templates/anadirPropietarios.html:** fichero que contiene la parte del cliente al intentar crear un propietario nuevo.
- **reservaAulas/reservaAulas__app/templates/auditoria.html:** fichero que contiene el código para que el cliente pueda ver la tabla de auditorías.
- **reservaAulas/reservaAulas__app/templates/errorModificarEvento.html:** fichero que permite al cliente ver el mensaje de error al intentar modificar un evento y que se solapen las horas con algún evento existente.
- **reservaAulas/reservaAulas__app/templates/errorPermisos.html:** fichero que muestra al cliente un mensaje de error cuando se le ha hecho propietario de un calendario pero no se le han dado los permisos manualmente y no puede acceder a sus funcionalidades.
- **reservaAulas/reservaAulas__app/templates/events.html:** página que pide al cliente las características de los eventos que se quieren visualizar .
- **reservaAulas/reservaAulas__app/templates/eventsTable.html:** contiene el código de la parte del cliente que permite ver la tabla con los eventos que coincidan con las especificaciones pasadas, permite modificar y borrar los eventos si se tiene permisos.
- **reservaAulas/reservaAulas__app/templates/home.html:** página de inicio de sesión y página principal si ya se ha iniciado sesión.
- **reservaAulas/reservaAulas__app/templates/mensajePropietario.html:** fichero que permite al cliente ver el mensaje recordatorio al administrador para que asigne los permisos manualmente cuando crea un nuevo propietario o modifica el propietario de un aula.
- **reservaAulas/reservaAulas__app/templates/navBar.html:** contiene código común para la barra de navegación superior y la franja inferior, heredan de ella para ahorrar código duplicado.
- **reservaAulas/reservaAulas__app/templates/reservar.html:** fichero que contiene el código para mostrar al cliente la página desde la que pueden realizar una reserva de un aula.
- **reservaAulas/reservaAulas__app/templates/showEventsTable.html:** contiene la tabla de eventos que se muestra al elegir la opción de ver los eventos de un aula que se va a reservar desde la ventana de reservar.

- **reservaAulas/reservaAulas__app/templates/verAulas.html:** fichero que contiene el código que ve el cliente al ver la lista de aulas.
- **reservaAulas/reservaAulas__app/templates/verPropietarios:** página que muestra una tabla en la que se ven los propietarios existentes.

D.4. Compilación, instalación y ejecución del proyecto

En esta sección será explicada la compilación, lo necesario para realizar la instalación correctamente y la forma de ejecutar el proyecto. Primeramente se explican los pasos que se tuvieron que realizar para poder registrar la aplicación en Microsoft Azure y permitir el acceso a el presente proyecto mediante el login de Microsoft que haremos funcionar al registrar la aplicación previamente.

Comentando el uso de la API de Microsoft para acceder a los datos de Outlook y poder habilitar un inicio de sesión bajo Microsoft, hubo que registrar la aplicación en Azure Active Directory, para ello fueron seguidos estos pasos:

- Acceder al centro de administración de Azure Active Directory.
- Acceder al apartado registro de aplicaciones y crear un nuevo registro.
- Una vez ahí dentro, se pedirá rellenar unos campos con el nombre de la aplicación, cuentas que accederán y lo más importante, la dirección URL de redirección al iniciar sesión mediante Outlook.

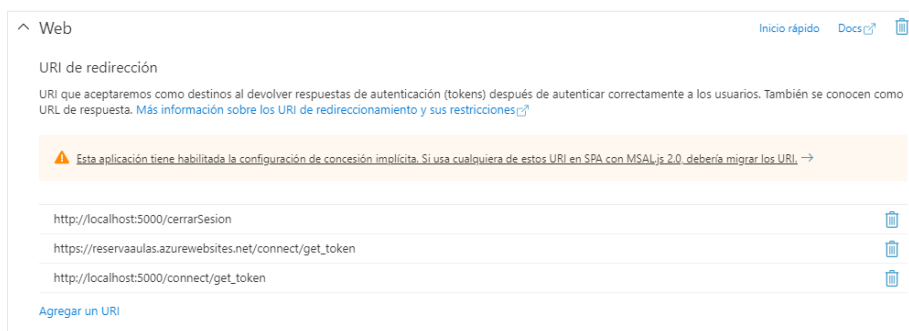


Figura D.1: URL de respuesta de nuestra aplicación

- Una vez se ha creado, recogemos el ID de la aplicación obtenido, que se utilizará para obtener el secreto del cliente, que solamente se mostrará una vez y será necesario para el resto del proyecto. Para obtenerlo se accede a certificados y secretos, se escribe un nombre para registrar la aplicación y se elige una duración límite para que expire el secreto que se recibirá y haya que pedirlo de nuevo, de esta forma se obtiene dicho secreto desde Azure Active Directory.

Esto está especificado en el tutorial de Microsoft[1].

Una vez registrada la aplicación disponemos del código secreto, que no se volverá a repetir, y del identificador de la aplicación que se ve en la figura D.2. Hubo que seguir una serie de pasos para incluirlo en el código y hacer que funcionara:

Nombre para mostrar : Python Outlook Tutorial

Id. de aplicación (cliente) : 0f0cdc95-e08f-4b67-9cb8-62b4350aa9a2

Figura D.2: Código de aplicación de Azure Active Directory

- Agregar al fichero 'config.py' el secreto recibido, el identificador de la aplicación de Azure y la ruta a la que redirigir una vez se ha logeado. Con esto se configura nuestro proyecto para la conexión con la aplicación registrada en Azure que se ha realizado en los pasos anteriores (ver figura D.3) .

```
0365_APP_ID = '0f0cdc95-e08f-4b67-9cb8-62b4350aa9a2'
0365_APP_KEY = 'yvTRWlcnmT@nY4?zIA?N2yc5yeIA0o_'
0365_REDIRECT_URI = os.path.join(SERVER_ADDRESS, 'connect/get_token')
0365_AUTH_URL = 'https://login.microsoftonline.com/common/oauth2/v2.0/authorize'
0365_TOKEN_URL = 'https://login.microsoftonline.com/common/oauth2/v2.0/token'
```

Figura D.3: Código incluyendo la aplicación de Azure Active Directory

- Agregar al fichero 'config.py' el código con los permisos de Outlook que se van a proporcionar al usuario que acceda, en este proyecto son los siguientes: leer el usuario, lectura de calendarios y calendarios compartidos, escritura de calendarios y calendarios compartidos, acceso al email, acceso offline,
- Crear en el fichero 'oauth_helpers.py' unas funciones con llamadas a la API REST para obtener un token de inicio de sesión y otro para refrescarlo. Esto servirá para cada sesión de cada usuario diferente.

Una vez se creó este primer código que permitía acceder mediante el inicio de sesión de Outlook a la página que especificamos en Azure Active Directory como URL de redirección, se pudo empezar a trabajar con llamadas a la API REST para obtener información sobre los calendarios del usuario y sus eventos.

La descarga del proyecto desde GitHub se puede realizar en:

<https://github.com/jgo0038/TFG-Reserva-aulas-informatica>

En el entorno de desarrollo, lo primero que hay que tener es una instalación de *Python* y un entorno virtual, si no se tiene, habrá que crearlo de forma sencilla con el siguiente comando

python3 -m venv /path/to/new/virtual/environment

o utilizar directamente el que se descarga con el repositorio.

Una vez se tenga esto creado, se activa el entorno virtual donde instalaremos las dependencias necesarias, como ya se explicó antes, especificadas en el fichero 'requirements.txt' (ver D.3). Este fichero sirve para realizar una instalación muy simple con un sencillo comando:

\$ pip install -r requirements.txt

Con este comando se instalarán las bibliotecas necesarias para esta versión usada dentro de nuestro entorno virtual. El siguiente paso será la compilación y ejecución del proyecto.

Para la ejecución de nuestra aplicación web, se necesita un servidor web que haga de host para la base de datos. Como ya se ha visto en otros apartados, se ha utilizado Azure, por lo que la conexión con la base de datos ha sido particularizada para este servidor en concreto (ver fichero 'reservaAulas/reservaAulas_app/odbctest.py') que contiene el nombre del servidor, de la base de datos, usuario, contraseña, puerto de Azure (1433) y driver para conectarse exactamente a esta base de datos de SQL Server alojada en Azure. Para que funcione con estas mismas configuraciones, hay que crear la base de datos con los mismos nombres que se muestran en la siguiente imagen, o bien adaptarlo a otra base de datos.

```
1 import pyodbc
2 server = 'reserva-aulas.database.windows.net'
3 database = 'reservaAulas'
4 username = 'user'
5 password = 'Password9'
6 driver= '{ODBC Driver 17 for SQL Server}'
7 cnxn = pyodbc.connect('DRIVER='+driver+';SERVER='+server+';
8 PORT=1433;DATABASE='+database+';UID='+username+';PWD='+ password)
9
```

Figura D.4: Código para conectar con la BD de Azure

Si se quisiera realizar en un servidor instalado en local, se tendría que cambiar la conexión y ajustarla al servidor utilizado. Azure restringe el acceso a las bases de datos desde otros ordenadores mediante la configuración del firewall, en el que se introducen las direcciones IP que pueden acceder a esta base de datos (actualmente solo puede realizarse desde el equipo utilizado por el alumno).

Inicio > reservaAulas (reserva-aulas/reservaAulas) >

Configuración de firewall

reserva-aulas (SQL server)

Guardar Descartar + Agregar IP de cliente

este servidor **SI** NO

i Las conexiones de las IP especificadas a continuación proporcionan acceso a todas las bases de datos en reserva-aulas.

Dirección IP de cliente 213.177.218.250

Nombre de regla	IP inicial	IP final	
			...
ClientIPAddress_2020-6-14_...	213.177.218.250	213.177.218.250	...

Figura D.5: Restricción de direcciones IP para la conexión

Debido a esta particularidad de la configuración del firewall de la base de datos, para poder acceder al proyecto como desarrollador se necesita crear una base de datos con las mismas relaciones que las especificadas en los diagramas y cambiar la conexión del fichero 'odbcetest.py' para conectarse a ella, o bien añadir la IP al firewall de la base de datos de Azure.

Si no se tiene acceso a la base de datos, se puede ejecutar desde el servidor de Azure al que está subida la aplicación accediendo desde el siguiente enlace <https://reservaaulas.azurewebsites.net> como cliente, esto lo veremos en el manual de usuario.

Si se dispone de acceso a la base de datos, se podrá realizar una ejecución en local accediendo a la base de datos del servidor de Azure, para ello propongo dos opciones:

Si se trabaja con VSCode (es el caso del alumno) los pasos a seguir son los siguientes:

- Activar el entorno virtual.
- Ejecutar desde el modo debug que proporciona VSCode, ya está configurado en 'reservaAulas/.vscode/launch.json' el fichero que se tiene que ejecutar en el entorno para que el proyecto se ponga en marcha en el puerto elegido, que es el 5000.
- Acceder a la aplicación desde nuestro localhost.

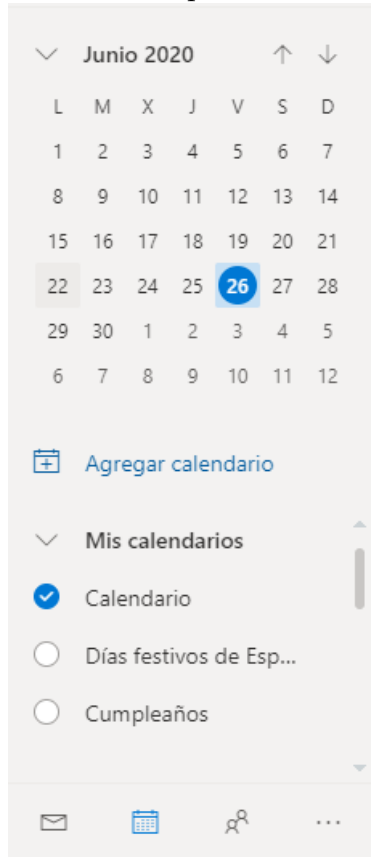
Si por otro lado, se quiere ejecutar desde cualquier otro lugar, hay que establecer manualmente las configuraciones que se acaban de comentar. Esto se hace de la siguiente forma:

- Activar el entorno virtual.
- Establecer la variable de entorno 'FLASK_APP', el comando para hacerlo es el siguiente:
\$ set FLASK_APP=reservaAulas_app.webapp
Si no utiliza Windows, use export en vez de set.
- El siguiente paso es ejecutar el proyecto, para ello ejecutamos la siguiente línea:
python -m flask run
- Acceder a la aplicación desde nuestro localhost.

Una vez se tiene acceso a la aplicación, si la ejecutamos también logeados como el administrador necesitaremos acceder a Outlook para poder compartir los calendarios con los propietarios. A continuación dejo la serie de pasos a seguir para compartir manualmente un calendario con un responsable:

- Iniciar sesión con la cuenta administrador del proyecto (las credenciales se facilitan en el archivo de texto 'EnlacesInteres.txt' adjunto al proyecto).

- Acceder a la pestaña de calendarios de Outlook.



Pestaña de calendarios

- Pinchar en los tres puntos del calendario a compartir y elegir la opción de 'Uso compartido y permisos'.

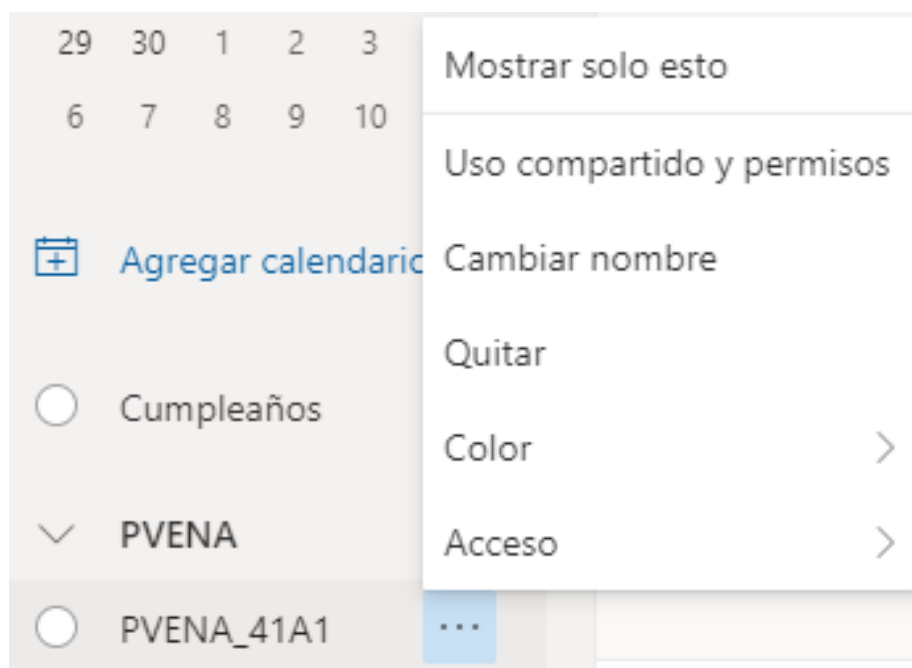


Figura D.6: Uso compartido y permisos

- Escribir la dirección de correo del propietario con el que se quiere compartir el calendario, o seleccionar el icono de la papelera en el caso de querer quitar el permiso a algún propietario.



Figura D.7: Compartir o borrar

D.5. Pruebas del sistema

En esta sección realizaré pruebas manuales de los casos de uso sobre el sistema comprobando que todo funciona correctamente.

Descripción	Respuesta
Iniciar sesión introduciendo los datos de una cuenta de Outlook	Se accede a la aplicación
Consultar eventos de aulas	Se muestra la tabla con los eventos que coincidan con las especificaciones.
Consultar eventos de aulas sin completar los campos obligatorios	Se pide que se rellenen los campos vacíos que sean obligatorios.
Modificar un evento	Se vuelve a la pantalla anterior y se muestra un mensaje de éxito en la modificación. Se modifica la información en el evento del calendario correspondiente de Outlook también.
Modificar un evento cambiando la hora a unas ya ocupadas	Se muestra una pantalla con un mensaje indicando que las horas están ocupadas y se ofrece la posibilidad de volver atrás.
Borrar más de un evento seleccionado y aceptando la confirmación de los deseados	Se muestra un mensaje de éxito en un cuadro de diálogo y se pide confirmación para mantenerse en la misma página con los mismos datos. Se borran los eventos de los calendarios de Outlook también.
Realizar reserva introduciendo todos los campos correctamente y en fechas que estén libres	Se queda en la misma página con el mismo formulario relleno que acabas de reservar y un mensaje en la parte inferior de éxito en la reserva. Se crea el evento en el calendario correspondiente de Outlook y se envía un email de notificación al profesor que solicitó la reserva.
Realizar reserva introduciendo algún campo en blanco	Se pide que se rellenen los campos vacíos.
Realizar reserva introduciendo fechas ya reservadas	Se queda en la misma página con el mismo formulario relleno que acabas de reservar y un mensaje en la parte inferior de error en las horas de la reserva.
Consultar aulas especificando el edificio del que se quieren ver	Se muestra una tabla con las aulas del edificio seleccionado y sus características.

Tabla D.1: Pruebas manuales propietario

En esta tabla realizaré pruebas con funcionalidades únicamente del administrador.

Descripción	Respuesta
Modificar aulas	Se vuelve a la página anterior y se muestra un mensaje por pantalla de éxito en la modificación.
Modificar aulas cambiando el propietario	Se muestra una pantalla en la que se lee un mensaje recordando al administrador de dar enviar el permiso de editar el calendario manualmente desde Outlook.
Borrar un aula aceptando la confirmación	Salta un cuadro de diálogo en el que se indica que se ha borrado correctamente el aula y se vuelve a la página anterior.
Crear un aula con espacios en blanco en el formulario	Se pide rellenar los espacios vacíos dejados.
Crear un aula correctamente	Se mantiene en la misma página mostrando el formulario completo y un mensaje en la parte inferior de éxito al crear el aula.
Añadir un propietario sin completar todos los campos	Se pide que se rellenen los campos vacíos

Tabla D.2: Pruebas manuales administrador

Apéndice *E*

Documentación de usuario

E.1. Introducción

En este apartado se explican los pasos a seguir por un usuario que quiere descargarse y utilizar la aplicación.

E.2. Requisitos de usuarios

Para poder utilizar la aplicación, el usuario tendrá que tener instalado Python.

E.3. Instalación

Los pasos de la instalación para la ejecución de la aplicación son los mismos que se han explicado en el manual del programador. A continuación se explican las formas en las que se puede acceder al proyecto.

Para el acceso de un usuario a la aplicación web, solamente necesita acceder al enlace en el que está subida dicha aplicación en Azure:

<https://reservaaulas.azurewebsites.net>.

Lo único que se necesita es que el *app service* en el que está realizado el deploy este ejecutándose y la base de datos de Azure activa. Estas acciones se explican a continuación:

- Acceder a <https://portal.azure.com/#home> con la cuenta desde la que se registraron los servicios. Se muestra una página como la

siguiente:

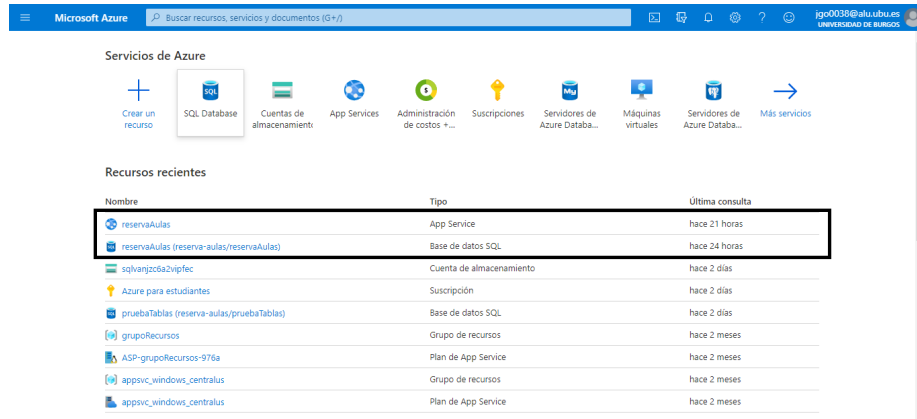


Figura E.1: Página portal de Azure.

- Entrar en el app service llamado 'reservaAulas' y pinchar en el botón de la parte superior de 'Activar' o de 'reiniciar'.

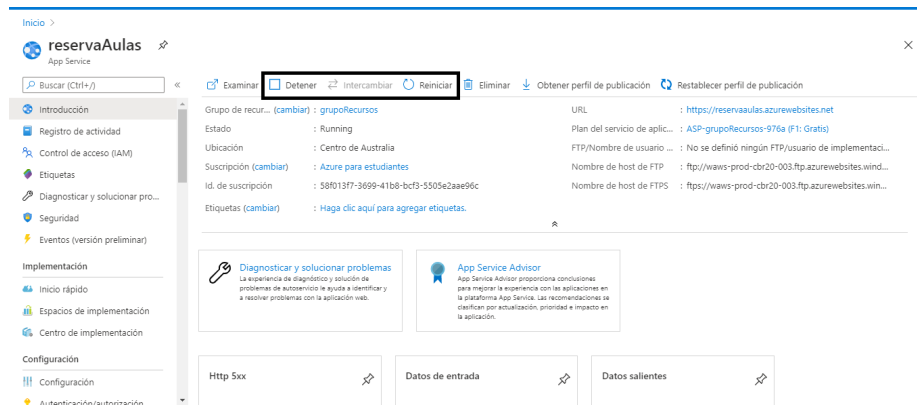


Figura E.2: Activar app service

- La base de datos se mantiene siempre activa para poder trabajar con la aplicación sin problemas.

Una vez se han seguido esos pasos, se asegura que el proyecto funcione ejecutándose desde la URL (<https://reservaaulas.azurewebsites.net>). Para

que esto funcione, fue necesario realizar un despliegue en un servicio de aplicación de Azure que se explica a continuación:

- Lo primero que se necesita es una suscripción a Azure, en mi caso utilicé la suscripción de estudiante, que me proporcionaba 100USD para realizar pruebas, aunque se agotaría el saldo y hubo que cambiar de plan de coste.
- Se necesita también un entorno de Python.
- Una vez tenemos los primeros requisitos, hay que crear un fichero 'requirements.txt' en el que se guarden las dependencias de la aplicación.
- Crear una instancia de un app service en Azure.
- Establecemos el comando de inicio al que nos indica Azure para el funcionamiento, almacenado en 'startup.txt' que es el siguiente:
gunicorn -bind=0.0.0.0 -timeout 600 startup:app
- Realizar el despliegue, en el caso de este proyecto se hizo desde VSCode, mediante la extensión de 'Azure: app service'. Para esto, se inició sesión con Azure en dicha extensión, ahí se muestran los app services dónde se puede realizar el despliegue y se selecciona el creado anteriormente.

Esta explicación se puede ver más detallada en el manual proporcionado por Microsoft del siguiente enlace [2].

Como se puede ver se crean los ficheros y la estructura según pide Microsoft y VSCode a lo largo del proyecto, por lo que el proyecto está creado para ser ejecutado en este entorno y servidor.

E.4. Manual del usuario

A continuación se explicará como realizar las operaciones que proporciona esta aplicación:

Inicio de la aplicación

Para comenzar a utilizar la aplicación se puede hacer de dos formas:

- Sin inicio de sesión (como invitado) que puede ver las reservas realizadas sobre las aulas y las aulas existentes. Para ello, hay que pinchar en el

botón de 'Comience a usar la app' de la página principal (ver figura E.3).

- Iniciando sesión con Outlook. Pinchando sobre el botón de 'Iniciar sesión desde Outlook' e introduciendo los datos de una cuenta de Microsoft (ver figura E.3).



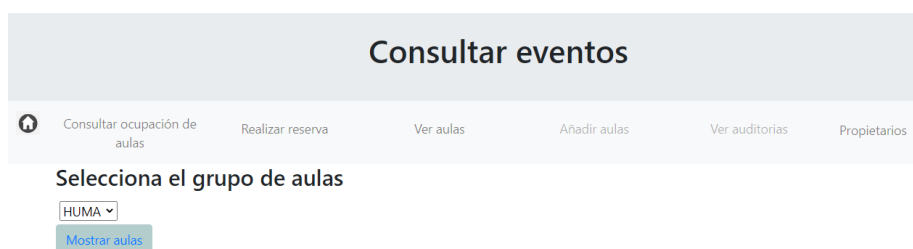
Figura E.3: Página principal.

Consultar ocupación de aulas

Pasos para consultar la ocupación de un aula:

- Pinchar en el botón de 'Consultar ocupación de aulas'
- Escoger el edificio en el que se encuentra el aula que se quiere comprobar(ver figura E.4).
- Se muestra un formulario en el que aparecen dos opciones (ver figura E.5):
 - Seleccionar un aula y rellenar al menos los campos obligatorios para filtrar las horas y fechas de las reservas a mostrar de ese aula.

- Seleccionar 'Cualquiera' en la opción del aula y rellenar al menos los campos obligatorios. Esto filtrará todas las aulas de ese edificio según las fechas especificadas. Además se pueden rellenar los campos opcionales de 'Capacidad', 'Numero de ordenadores' o 'Tipo' para filtrar las aulas según esas características.
- Pinchar en el botón 'Enviar consulta' y se mostrarán los eventos que coincidan en una tabla (ver figura E.6).



Consultar eventos

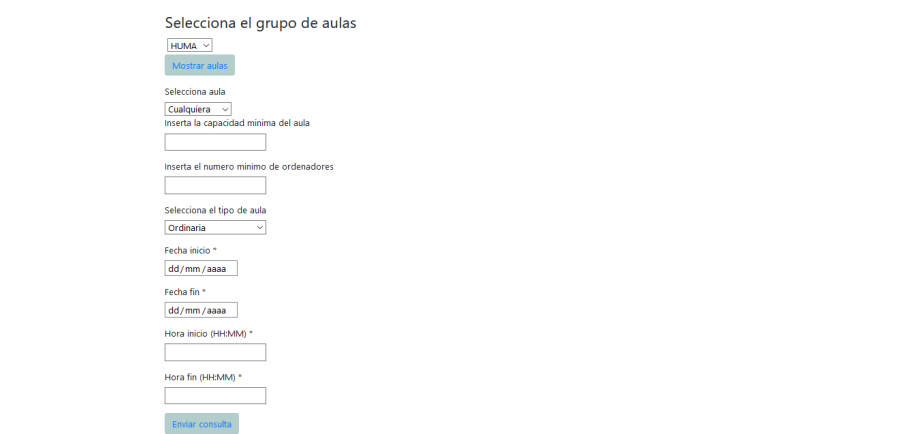
Consultar ocupación de aulas Realizar reserva Ver aulas Añadir aulas Ver auditorias Propietarios

Selecciona el grupo de aulas

HUMA

Mostrar aulas

Figura E.4: Página inicial para ver eventos de aulas.



Selecciona el grupo de aulas

HUMA

Mostrar aulas

Selecciona aula

Cualquiera

Inserta la capacidad mínima del aula

Inserta el número mínimo de ordenadores

Selecciona el tipo de aula

Ordinaria

Fecha inicio *

dd/mm/aaaa

Fecha fin *


dd/mm/aaaa

Hora inicio (HH:MM) *

Hora fin (HH:MM) *

Enviar consulta

Figura E.5: Formulario para consultar eventos.

Tabla de eventos									
<div>  Consultar ocupación de aulas Realizar reserva Ver/Modificar aulas Añadir aulas Ver auditorias Propietarios </div>									
<div> <div>Buscar:</div> </div>									
Nº reserva	Aula	Tema	Usuario	Fecha inicio	Hora inicio	Hora fin	Acciones		
106	HUMA_32A2	prueba invitado	Jorge	2020-07-10	08:00	12:00	Modificar	Selecione para borrar	
114	HUMA_32A2	prueba email	Jorge	2020-07-14	15:00	16:40	Modificar	Selecione para borrar	
115	HUMA_32A2	prueba email	Jorge	2020-07-15	15:00	16:40	Modificar	Selecione para borrar	

Mostrando 1 a 6 de 6 entradas

Leyenda

■ Eventos pasados
■ Eventos futuros

Mostrar eventos pasados
 Borrar eventos seleccionados
 Seleccionar todos

Volver

Figura E.6: Tabla donde se muestran los eventos.

Modificar eventos

Para modificar eventos, es necesario acceder previamente a la página de la consulta de eventos que se ha explicado en la sección anterior, una vez ahí:

- Se pulsa en el botón 'Modificar' de la parte derecha de la tabla (ver figura E.9).
- Se abre un formulario en la parte superior de la página en la que se muestran los datos del evento seleccionado y se puede modificar. Una vez modificado si se pincha en 'Modificar' se guardan los cambios (ver figura E.8).

Tabla de eventos									
<div> Consultar ocupación de aulas Realizar reserva Ver/Modificar aulas Añadir aulas Ver auditorías Propietarios </div>									
<div> <div> <div>Buscar:</div> <input type="text"/> </div> </div>									
Nº reserva	Aula	Tema	Usuario	Fecha inicio	Hora inicio	Hora fin	Acciones		
106	HUMA_32A2	prueba invitado	Jorge	2020-07-10	08:00	12:00	Modificar	Selecione para borrar	
114	HUMA_32A2	prueba email	Jorge	2020-07-14	15:00	16:40	Modificar	Selecione para borrar	
115	HUMA_32A2	prueba email	Jorge	2020-07-15	15:00	16:40	Modificar	Selecione para borrar	

Mostrando 1 a 6 de 6 entradas

Legenda

■ Eventos pasados
■ Eventos futuros

Figura E.7: Página inicial para ver eventos de aulas y modificarlos.

<div> Consultar ocupación de aulas Realizar reserva Ver/Modificar aulas Añadir aulas Ver auditorías Propietarios </div>															
Tema <input type="text" value="prueba invitado"/>		Usuario <input type="text" value="Jorge"/>		Fecha inicio <input type="text" value="2020-07-10"/>		Hora inicio <input type="text" value="08:00"/>									
Hora fin <input type="text" value="12:00"/>		<input type="button" value="Modificar"/>													
<div> <div> <div>Buscar:</div> <input type="text"/> </div> </div>															
Nº reserva	Aula	Tema	Usuario	Fecha inicio	Hora inicio	Hora fin	Acciones								
106	HUMA_32A2	prueba invitado	Jorge	2020-07-10	08:00	12:00	Modificar	Selecione para borrar							
114	HUMA_32A2	prueba email	Jorge	2020-07-14	15:00	16:40	Modificar	Selecione para borrar							
115	HUMA_32A2	prueba email	Jorge	2020-07-15	15:00	16:40	Modificar	Selecione para borrar							

Mostrando 1 a 6 de 6 entradas

Legenda

■ Eventos pasados
■ Eventos futuros

Figura E.8: Formulario para modificar eventos.

Borrar eventos

Para borrar eventos, hay que acceder primero a la página de la consulta de eventos que se ha explicado anteriormente, una vez ahí:

- Se pulsa en el botón 'Selecione para borrar' de la parte derecha de la tabla de los eventos que se quieren borrar (ver figura E.9).
- Se resaltan en rojo estos eventos seleccionados (ver figura E.10).
- Se pincha en el botón de la parte inferior de la página de 'Borrar eventos seleccionados' y sale un cuadro de diálogo en el que se pide la

confirmación para borrar (ver figura E.11). Si se acepta, se borran los eventos seleccionados.

Tabla de eventos										
<div> Consultar ocupación de aulas Realizar reserva Ver/Modificar aulas Añadir aulas Ver auditorías Propietarios </div>										
Buscar:										
Nº reserva	Aula	Tema	Usuario	Fecha inicio	Hora inicio	Hora fin	Acciones			
106	HUMA_32A2	prueba invitado	Jorge	2020-07-10	08:00	12:00	Modificar	Selecione para borrar		
114	HUMA_32A2	prueba email	Jorge	2020-07-14	15:00	16:40	Modificar	Selecione para borrar		
115	HUMA_32A2	prueba email	Jorge	2020-07-15	15:00	16:40	Modificar	Selecione para borrar		

Mostrando 1 a 6 de 6 entradas

Leyenda

■ Eventos pasados
■ Eventos futuros

Mostrar eventos pasados
 Borrar eventos seleccionados
 Seleccionar todos

Volver

Figura E.9: Página inicial para ver eventos de aulas y borrarlos.

Tabla de eventos										
<div> Consultar ocupación de aulas Realizar reserva Ver/Modificar aulas Añadir aulas Ver auditorías Propietarios </div>										
Buscar:										
Nº reserva	Aula	Tema	Usuario	Fecha inicio	Hora inicio	Hora fin	Acciones			
106	HUMA_32A2	prueba invitado	Jorge	2020-07-10	08:00	12:00	Modificar	Selecione para borrar		
114	HUMA_32A2	prueba email	Jorge	2020-07-14	15:00	16:40	Modificar	Selecione para borrar		
115	HUMA_32A2	prueba email	Jorge	2020-07-15	15:00	16:40	Modificar	Selecione para borrar		

Mostrando 1 a 6 de 6 entradas

Leyenda

■ Eventos pasados
■ Eventos futuros

Mostrar eventos pasados
 Borrar eventos seleccionados
 Seleccionar todos

Figura E.10: Selección de eventos a borrar.

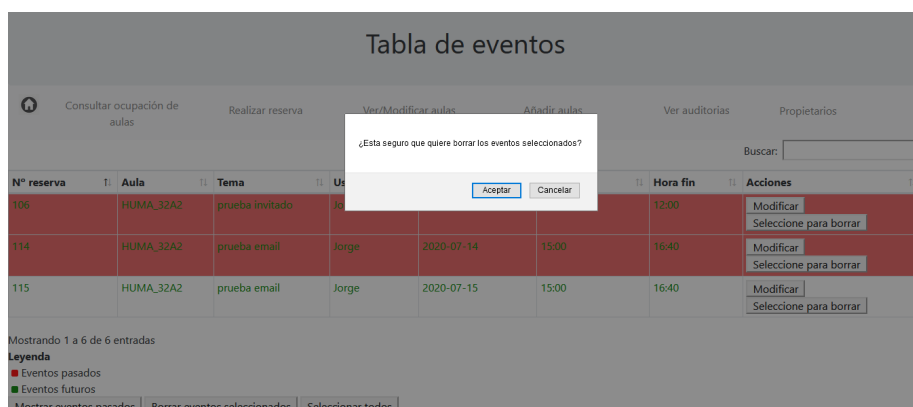


Figura E.11: Confirmación de borrado.

Realizar reserva

Pasos a seguir para realizar la reserva de un aula:

- Seleccionar en el formulario de la izquierda los valores deseados y pinchar en el botón 'Consultar' (ver figura E.12). Esto rellena la columna de la derecha el desplegable de 'Selecciona aula'.
- Elegir un tipo de reserva y pinchar en 'Aplicar'. Varía el número de campos a rellenar.
- Completar la información pedida sobre la reserva en la parte derecha (ver figura E.13).

Figura E.12: Pantalla inicial de la reserva.

Figura E.13: Pantalla completa antes de la reserva.

Modificar aulas

Pasos a seguir para realizar la modificación de un aula:

- Acceder a la ventana de 'Ver/modificar aulas'.
- Seleccionar un grupo de aulas en el que estará el aula que queremos modificar. Se mostrará una tabla con todas las aulas de dicho grupo (ver figura E.14).
- Seleccionar el botón 'Editar', se abrirá un formulario en la parte superior para poder editar los datos del aula. Una vez editados pinchar en el botón 'Guardar'. (ver figura E.15).

Información sobre aulas

Consultar ocupación de aulas

Realizar reserva

Ver/Modificar aulas

Añadir aulas

Ver auditorias

Propietarios

Selección grupo de aulas

Humanidades

Ver aulas

Buscar:

Nombre	Ti	Edificio	Ti	Tipo	Ti	Capacidad	Ti	Nº ordenadores	Ti	Propietario	Ti	Acciones	Ti
HUMA_32A2		Humanidades		Ordinaria		70		0		DEP_INF		<div>Editar</div> <div>Borrar</div>	
HUMA_51A1		Humanidades		Sala de ordenadores		40		20		DEP_INF		<div>Editar</div> <div>Borrar</div>	

Mostrando 1 a 2 de 2 entradas

Figura E.14: Pantalla inicial de la modificación.

Consultar ocupación de aulas | Realizar reserva | Ver/Modificar aulas | Añadir aulas | Ver auditorias | Propietarios

Selecciona grupo de aulas

[Ver aulas](#)

Nombre del aula:
 Selecciona el edificio:
 Tipo de aula:
 Capacidad del aula:
 Numero de ordenadores:
 Propietario del aula:
[Guardar](#)

Buscar:

Nombre	Edificio	Tipo	Capacidad	Nº ordenadores	Propietario	Acciones
HUMA_32A2	Humanidades	Ordinaria	70	0	DEP_INF	Editar Borrar
HUMA_51A1	Humanidades	Sala de ordenadores	40	20	DEP_INF	Editar Borrar

Figura E.15: Pantalla completa antes de la modificación.

Añadir aulas

Pasos a seguir para añadir un aula:

- Acceder a la ventana de 'Añadir aulas'.
- Rellenar el formulario que se presenta indicando los datos pedidos. Pinchar en 'crear aula'.

Añadir aulas

Consultar ocupación de aulas | Realizar reserva | Ver/Modificar aulas | Añadir aulas | Ver auditorias | Propietarios

Selecciona el edificio

 Nombre del aula

Tipo de aula

 Capacidad del aula

 Numero de ordenadores

 Propietario del aula

[Crear aula](#)

Figura E.16: Pantalla para añadir un aula.

- Una vez se ha añadido el aula, se accede a los calendarios de Outlook y se asignan los permisos manualmente al responsable correspondiente de este nuevo aula.

Ver auditorías

Para ver la tabla de las auditorías, que muestra los cambios realizados sobre las reservas de todos los calendarios, hay que ser administrador y pinchar en el botón de 'Ver auditorías'.

Tabla de auditorías					
	Consultar ocupación de aulas	Realizar reserva	Ver/Modificar aulas	Añadir aulas	Ver auditorías
Elige un rango de fechas para filtrar					
Fecha inicio		dd/mm/aaaa	Fecha fin	dd/mm/aaaa	Filtrar
Buscar:					
Nº reserva	Propietario	Fecha realizada	Operación		
46	jgo0038@alu.ubu.es	2020-06-20 11:04	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:07	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:08	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:10	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:11	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:13	MODIF		
46	jgo0038@alu.ubu.es	2020-06-20 11:16	MODIF		
60	estfchad@alu.ubu.es	2020-06-21 13:53	MODIF		

Figura E.17: Pantalla para ver la tabla de auditorías.

Modificar propietario

Pasos a seguir para modificar un propietario:

- Acceder a la ventana de 'Ver propietarios'.
- Se muestra una tabla con los propietarios existentes y un botón de 'Editar' a su derecha (ver figura E.18).
- Aparecerá un formulario en la parte superior de la página con los datos del propietario a editar, ahí se podrá modificar y pinchar en el botón 'Modificar' para guardar los cambios (ver figura E.19).

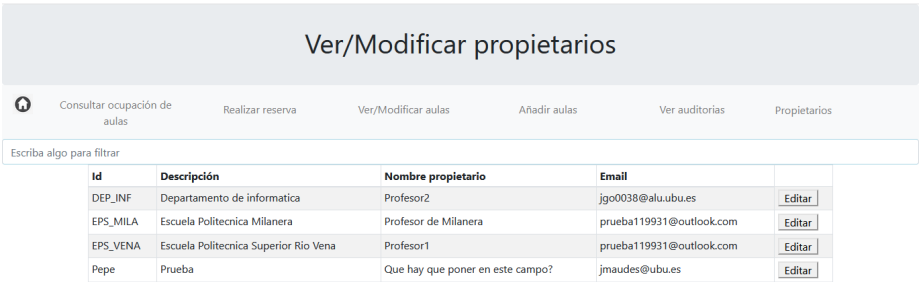


Figura E.18: Pantalla inicial de la modificación de propietario.



Figura E.19: Pantalla completa antes de la modificación.

- Una vez se ha modificado el propietario, si se ha cambiado el responsable, se accede a los calendarios de Outlook y se asignan los permisos manualmente al responsable correspondiente después de la modificación.

Bibliografía

- [1] Microsoft Corp. Python en microsoft graph. <https://docs.microsoft.com/es-es/graph/tutorials/python>.
- [2] Microsoft Corp. Deploy azure app service en vscode. <https://docs.microsoft.com/es-es/azure/developer/python/tutorial-deploy-app-service-on-linux-01>, 2019 (accedido 29/06/2020).
- [3] Inc. Free Software Foundation. Licencias - gnu. <https://www.gnu.org/licenses/license-list.es.html#SoftwareLicenses>, 2019 (accedido 28/06/2020).
- [4] Bryson Tyrrell. Login flask app o365. <https://github.com/brysontyrrell/Office-365-Flask-App>, 2016 (accedido 28/06/2020).