



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática

Plataforma de Corrección Automática de Ejercicios en Python



Presentado por Alberto Porres Fernández
en Universidad de Burgos — 7 de julio de 2022
Tutores: Bruno Baruque Zanón
y Roberto Carlos Casado Vara

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	vi
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catálogo de requisitos	14
B.4. Especificación de requisitos	18
Apéndice C Especificación de diseño	41
C.1. Introducción	41
C.2. Diseño de datos	41
C.3. Diseño procedimental	43
C.4. Diseño arquitectónico	47
Apéndice D Documentación técnica de programación	49
D.1. Introducción	49
D.2. Estructura de directorios	49
D.3. Manual del programador	51

D.4. Instalación, ejecución y despliegue del proyecto	73
Apéndice E Documentación de usuario	77
E.1. Introducción	77
E.2. Requisitos de usuarios	77
E.3. Instalación	78
E.4. Manual del profesor	78
E.5. Manual del alumno	97
Bibliografía	105

Índice de figuras

B.1. Casos de Uso Profesor e Inicio de sesión	18
B.2. Casos de Uso Alumno	19
C.1. Diagrama E/R	42
C.2. Diagrama relacional	43
C.3. Diagrama de secuencias de una consulta	44
C.4. Diagrama de secuencias de una inserción / borrado	45
C.5. Diagrama de secuencias autograding	46
C.6. Diagrama de secuencias generación de tarea	47
C.7. Representación MVC	48
D.1. Proceso de instalación 1	51
D.2. Proceso de instalación 2	52
D.3. Proceso de instalación 3	52
D.4. Proceso de instalación 4	52
D.5. Proceso de lanzamiento y creación 1	53
D.6. Proceso de lanzamiento y creación 2	53
D.7. Proceso de lanzamiento y creación 3	54
D.8. Proceso de lanzamiento y creación 4	54
D.9. Proceso de lanzamiento y creación 5	54
D.10. Proceso de lanzamiento y creación 6	55
D.11. Proceso de lanzamiento y creación 7	55
D.12. Proceso de lanzamiento y creación 8	56
D.13. Proceso de lanzamiento y creación 9	56
D.14. Proceso de lanzamiento y creación 10	57
D.15. Proceso de lanzamiento y creación 11	57
D.16. Proceso de lanzamiento y creación 12	58
D.17. Proceso de lanzamiento y creación 13	58

D.18. Proceso de realización y corrección 1	59
D.19. Proceso de realización y corrección 2	59
D.20. Proceso de realización y corrección 3	59
D.21. Proceso de realización y corrección 4	60
D.22. Proceso de realización y corrección 5	60
D.23. Proceso de realización y corrección 6	60
D.24. Proceso de realización y corrección 7	60
D.25. Proceso de realización y corrección 8	61
D.26. Estructura Curso Nbgrader	61
D.27. Tabla Nbgrader 1	63
D.28. Tabla Nbgrader 2	64
D.29. Curso base	65
D.30. Código creación curso	65
D.31. Constructor manager	66
D.32. Corrección de tarea	67
D.33. Generación de tarea para los alumnos	67
D.34. Generación de feedback	68
D.35. Ejecución de Jupyter	69
D.36. Apertura Notebook	69
D.37. Dockerfile	71
D.38. Edición de tarea	72
D.39. Acceso carpetas de cursos	73
D.40. Modo de depuración	74
E.1. Página Inicial	78
E.2. Pagina Inicio de sesión	79
E.3. Página Principal Profesor	80
E.4. Página de Cursos Profesor	81
E.5. Página de Creación de Curso	82
E.6. Página de Gestión de Alumnos de un Curso	83
E.7. Página de Gestión de Contenidos de un curso	84
E.8. Página de creación de sección	85
E.9. Página de secciones no publicadas	86
E.10. Requerimiento de contraseña	87
E.11. Edición de tarea desde la plataforma	87
E.12. Página de creación de tareas no publicadas	88
E.13. Activar vista de edición	89
E.14. Tipos de celda	90
E.15. Ejemplo Tarea	91
E.16. Ejemplo validate	92
E.17. Ejemplo versión alumno	93

E.18.Página de calificaciones global	93
E.19.Página de gestión alumnos	94
E.20.Página de creación de alumno	94
E.21.Página de perfil del alumno	95
E.22.Página de cambio de contraseña del profesor	96
E.23.Página inicial	97
E.24.Pagina Inicio de sesión	98
E.25.Página Principal Alumno	98
E.26.Página de Cursos del alumno	99
E.27.Página de Contenidos del Curso	100
E.28.Página de Calificaciones del Curso	101
E.29.Página de Calificaciones Global del alumno	102
E.30.Página de cambio de contraseña del alumno	102

Índice de tablas

A.1. Tareas sprint 1	2
A.2. Tareas sprint 3	3
A.3. Tareas sprint 3	3
A.4. Tareas sprint 4	4
A.5. Tareas sprint 5	4
A.6. Tareas sprint 6	5
A.7. Tareas sprint 7	6
A.8. Tareas sprint 8	6
A.9. Costes de personal	7
A.10.Costes de hardware	8
A.11.Costes de estructura	9
A.12.Costes totales	9
A.13.Tabla de licencias	10

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este primer apéndice se van a exponer los distintos *sprints* que se han realizado durante el desarrollo del proyecto junto con un estudio de la viabilidad de este dividido en dos apartados: viabilidad económica y viabilidad legal.

A.2. Planificación temporal

Como ya se ha mencionado anteriormente, al desarrollo de este proyecto le ha sido aplicada una metodología *agile*. Para posibilitar la adaptación de esta metodología a un proyecto supervisado, realizado por una sola persona, se han seguido las siguientes indicaciones:

- El desarrollo ha sido dividido en *sprints* de dos semanas de duración.
- Al inicio de cada sprint eran definidas las tareas a realizar dentro de este, añadiéndose nuevas durante el desarrollo del sprint en caso de finalización de las inicialmente definidas o por necesidades surgidas en el desarrollo de otras tareas y pasadas al siguiente sprint en caso de la no finalización de estas.
- Se organizaban reuniones de supervisión alrededor de la fecha de finalización de cada sprint para comprobar el estado de cada tarea y su correcta superación.

- A cada tarea le era asignado un coste estimado en formato de tiempo (horas) que el programador decidía de manera personal al definirse cada tarea.
- Al darse por concluida una tarea se especificaba el coste real que esta había supuesto.

La gestión de tareas y la estimación de sus costes se ha realizado mediante la herramienta *ZenHub*[13] mencionada en el apartado de Técnicas y Herramientas de la memoria.

A continuación se describirán los *sprints* por los que este proyecto ha pasado:

Sprint 1 (22/03/22 - 05/04/22)

Este sprint consistió en una primera toma de contacto con las herramientas de autograding en Python[12]. Momento del proyecto orientado hacia la investigación y aprendizaje ya que se trataba del inicio de este. Adicionalmente, se incluyó en este sprint la búsqueda de un curso[7] introductorio a Python sobre el que basar el que posteriormente sería nuestro curso y la redacción del apartado de objetivos de la memoria:

Tarea	Est	Real
Búsqueda de atuograders, comparativa y elección	1	6
Búsqueda del curso de Python	1	1
Instalación y pruebas documentadas del autograder seleccionado	8	8
Completar apartado Objetivos en la memoria	8	-

Tabla A.1: Tareas sprint 1

Sprint 2 (05/04/22 - 19/04/22)

Al no poder ser realizada / finalizada la redacción del apartado de objetivos en el sprint inicial, esta tarea fue delegada a este sprint número 2 la cual pasó a ser la primera tarea de este.

Una vez realizada la primera toma de contacto con la herramienta de autograding seleccionada *Nbgrader*[2], se definieron para este sprint las tareas de adición de la documentación sobre la instalación y prueba de esta al

Apéndice de Diseño, junto con la realización de una investigación sobre su funcionamiento interno. Además se añadió la tarea de creación de nuestro curso introductorio a Python junto con sus pruebas autocorregibles en un futuro mediante la herramienta Nbgrader.

Tarea	Est	Real
Completar apartado Objetivos en la memoria	8	3
Añadir la documentación de prueba e instalación de Nbgrader a los anexos	2	2
Realizar investigación del funcionamiento interno de Nbgrader	13	12
Creación del curso de Python	13	14

Tabla A.2: Tareas sprint 3

Sprint 3 (19/04/22 - 03/05/22)

Finalizadas de forma exitosa todas las tareas del sprint 2, comenzaba así el sprint número 3 para el que fue definida la tarea de redacción de una explicación del funcionamiento de Nbgrader basada en la información obtenida en la tarea de investigación del sprint anterior.

Adicionalmente y como paso previo al inicio del desarrollo de la plataforma, era necesaria la prueba de ejecución y acceso a Jupyter Notebooks de forma remota en un contenedor Docker[9], tarea que fue incluida en este sprint.

Tarea	Est	Real
Jupyter Notebooks en Docker	13	13
Redactar explicación sobre el funcionamiento de Nbgrader	5	-

Tabla A.3: Tareas sprint 3

Sprint 4 (03/05/22 - 17/05/22)

Debido a que la completitud de la tarea de redacción de explicación sobre el funcionamiento de Nbgrader no había sido posible en el sprint número 3, esta tarea fue delegada a este cuarto sprint, tarea que fue acompañada con el comienzo del desarrollo de la plataforma web la cual contenía: la creación

del login[11] inicial con *Flask*, una prueba de acceso a Jupyter Notebooks[10] ejecutado remotamente en un contenedor docker junto con el proyecto Flask, el desarrollo del diagrama de casos de uso inicialmente pensado para la plataforma y el inicio de la implementación de las vistas y funcionalidades para los alumnos.

Tarea	Est	Real
Redactar explicación sobre el funcionamiento de Nbgrader	5	6
Creación de nuestro primer login con Flask	5	6
Acceso a Jupyter Notebooks ejecutado remotamente desde una aplicación Flask	5	5
Creación del diagrama de casos de uso	2	2
Desarrollo de las vistas de la aplicación para los alumnos	21	-

Tabla A.4: Tareas sprint 4

Sprint 5 (17/05/22 - 31/05/22)

El sprint 5 fue dedicado en su totalidad al desarrollo de las vistas y funcionalidades de la aplicación para los alumnos, tarea de la que surgieron otras sub-tareas en su desarrollo que consistían en: prueba y adaptación de la API de Nbgrader[3] a las funcionalidades implementándose en el momento y construcción de las tablas de la base de datos[5][8] e inicialización de esta.

Tarea	Est	Real
Desarrollo de las vistas de la aplicación para los alumnos	21	24
(sub-tarea) Prueba y adaptación API de Nbgrader	2	2
(sub-tarea) Inicialización de la base de datos	5	5

Tabla A.5: Tareas sprint 5

Sprint 6 (31/05/22 - 14/06/22)

Tras la finalización del desarrollo de las funcionalidades para los estudiantes, se definía en el sprint 6 la tarea de desarrollo de las vistas y funcionalidades de la aplicación para el profesor, de la cual se generaron las sub-tareas de: desarrollo de funcionalidad de gestión completa de alumnos, funcionalidad de visualización de calificaciones por parte del profesor, funcionalidad de gestión completa de cursos.

Adicionalmente, en este sprint también se definieron otras tareas de menor carga de trabajo como el desarrollo del diagrama de clases de la base de datos y el diseño de un estado inicial para la plataforma, con el curso de Python, alumnos y entregas ya realizadas, estado incial que sería incluido en la plataforma posteriormente tras el despliegue vacío de esta.

También fue definida las tareas de creación del landing y adición de estilos y las vistas de la plataforma, siendo esta última únicamente iniciada durante este sprint.

Tarea	Est	Real
Desarrollo de las vistas y funcionalidades de la aplicación para el profesor	21	30
(sub-tarea) Gestión completa de alumnos	5	5
(sub-tarea) Gestión de calificaciones	2	2
(sub-tarea) Gestión completa de cursos	13	13
Desarrollo diagrama de clases de la base de datos	2	2
Establecer el contenido incial que tendrá la plataforma	2	2
Creación del landing de la plataforma	2	2
Dar estilos a toda la plataforma con Bootstrap y landing con CSS	21	-

Tabla A.6: Tareas sprint 6

Sprint 7 (14/06/22 - 28/06/22)

Añadidas todas las funcionalidades incialmente pensadas para la plataforma, el sprint 7 iniciaba con el final de la tarea de adición de estilos y creación de landing con Bootstrap[6] y CSS[1] y en este eran definidas las tareas de: adición de nuevas funcionalidades (obtención de feedback, eliminación de entregas, descarga de tareas enviadas, eliminación de cursos y secciones), despliegue de la app en una instancia EC2 de AWS[4] y completitud de apartados de la memoria del proyecto.

Tarea	Est	Real
Dar estilos a toda la plataforma con bootstrap y landing con CSS	21	21
Añadir funcionalidad de obtención de feedback	3	1
Añadir funcionalidad de eliminación de entregas	2	2
Añadir funcionalidad de eliminación de cursos y secciones	3	3
Añadir funcionalidad de descarga de envíos	2	2
Completar apartado Técnicas y Herramientas en la memoria	5	4
Completar apartado Aspectos Relevantes en la memoria	5	4
Completar apartados de Introducción, Conceptos Teóricos, Trabajos Relacionados y Conclusiones	8	8

Tabla A.7: Tareas sprint 7

Sprint 8 (28/06/22 - 07/07/22)

El sprint número 8 da como concluido el proyecto, incluyéndose en este tareas dedicadas al desarrollo de los anexos y la adición de los contenidos definidos previamente a la plataforma ya desplegada.

Tarea	Est	Real
Completar anexo Plan de Proyecto Software	3	3
Completar anexo Especificación de Requisitos	3	3
Completar anexo Especificación de Diseño	3	3
Completar anexo Documentación del Programador	5	5
Completar anexo Documentación del Usuario	8	8
Adición de contenidos a la plataforma desplegada	2	2

Tabla A.8: Tareas sprint 8

A.3. Estudio de viabilidad

Viabilidad económica

Analizaremos los costes en que habría incurrido el desarrollo de este proyecto de haberse llevado a cabo en un ámbito empresarial o si se hubiese encargado a una entidad externa.

La estructura de costes la desglosaremos en los apartados coste de personal , costes de software y costes del hardware y costes de estructura.

En cuanto a los ingresos o beneficios por la aplicación desarrollada no se realizada ningún análisis. La razón es que no está previsto un uso comercial del producto generado, sino que sería una herramienta utilizable dentro de un marco más general de formación en el que puede haber otras herramientas empleadas. En dicho marco la mayor parte del valor añadido y de la que se han de considerar los ingresos es la actividad realizada por los profesores: desarrollo del material de cursos, evaluaciones, tutorías y comunicaciones con alumnos, gestión de alumnos, etc.

Costes de personal

Para llevar a término el proyecto se ha requerido de un desarrollador durante un periodo de cuatro meses. Si bien el tiempo de desarrollo es inferior al señalado hay que incluir también el dedicado al análisis de herramientas y estudio de soluciones. El coste de personal se desglosa en la siguiente tabla:

Concepto	Coste mensual (€)
Salario bruto	1500,00
Seguridad Social (29,90 %)	449,50
Coste para el empleador	1949,50
Total Coste de personal (4 meses)	7794,00 €

Tabla A.9: Costes de personal

El salario bruto incluye todo el montante del salario del trabajador por este concepto, es decir, integra la parte de prorrata de pagas extra. Para hallar el sueldo mensual neto del trabajador se debe descontar la prorrata de pagas extra (1 / 6 del bruto mensual) y la parte de Seguridad Social a cuenta del trabajador, que es el 6,35 % del bruto mensual (incluyen los apartados contingencias comunes, desempleo y formación profesional).

El apartado de Seguridad incluye la parte a cuenta de la empresa por contingencias comunes (23,60 %), desempleo (5,50 %), FOGASA (0,20 %) y formación profesional (0,60 %), según las tablas de la Seguridad Social para 2022.

Costes de hardware

Para este apartado se considera que se ha utilizado un ordenador portátil y ha sido necesario un teléfono móvil a cuenta de la empresa y disposición

del trabajador. La amortización se consideran 4 años para ambos, con valor residual cero a final de dicho periodo.

Equipo	Coste (€)	Coste amortización 4 meses (€)
Ordenador portátil	900,00	75,00
Teléfono móvil	264,00	22,00
Total	1164,00	97,00

Tabla A.10: Costes de hardware

Costes de software

En este apartado se considera el coste de aquellas licencias de software necesarias y que no sean gratuitas.

En nuestro caso, esto afecta a las licencias de sistema operativo de los equipos hardware empleado. No obstante, en el apartado relativo al hardware el coste total del equipo ya incluye el de la licencia de sistema operativo que llevan instalado de origen. El coste de amortización de dicho software se considera igual que el del equipo, pues no se prevé renovación del sistema operativo del equipo durante su vida útil. Es decir, que el coste del software de está ya incluido en el coste del apartado anterior.

Costes de estructura

En este apartado se incluyen costes aplicables al entorno de trabajo por parte de la empresa y servicios para que la funcionalidad del puesto de trabajo sea operativa.

Se considera que el trabajo se ha realizado en modo teletrabajo, por lo que no se precisa una oficina operativa para el trabajador o un puesto destinado para él en una oficina. Esto resulta en que no se contemplan los costes asociados de alquiler de oficina, instalaciones, suministros eléctricos, limpieza, etc. No obstante, el trabajador necesita que se ponga a su disposición una conexión a Internet. Además necesita una compensación por el gasto de consumo eléctrico que requiera durante su trabajo, pues de otro modo correría por su cuenta.

Concepto	Coste mensual (€)
Conexión a internet	30
Compensación por gasto eléctrico	10
Coste por mes	40
Total Coste de estructura (4 meses)	160 €

Tabla A.11: Costes de estructura

Costes totales

Capítulo	Coste (€)
Personal	7.794,00
Hardware/Software	97,00
Estructura	160,00
Total	8.051,00 €

Tabla A.12: Costes totales

Viabilidad legal

En este apartado se consideran las licencias de software de terceras partes empleado en este proyecto. Se analiza su repercusión o influencia en la viabilidad legal dado que su uso puede aparejar limitaciones o restricciones legales.

La siguiente tabla contiene las licencias de las diferentes herramientas empleadas:

Herramienta	Licencia
Python	PSF
Flask	BSD 3-clause
Flask-Login	BSD 3-clause
Flask-SQLAlchemy	BSD 3-clause
SQLite	No se requiere
WTForms	BSD 3-clause
Nbgrader	BSD 3-clause
JavaScript	GNU-GPL-3.0
Bootstrap	MIT License
Jupyter Notebook	BSD 3-clause
Docker	Apache License 2.0

Tabla A.13: Tabla de licencias

Se describen a continuación estas licencias:

- **BSD:** Las licencias BSD son una familia de licencias de software libre permisivas , que imponen restricciones mínimas sobre el uso y distribución del software cubierto. Actualmente son de uso generalizado licencias BSD modificadas respecto a la licencia BSD que se estableció originalmente para la versión BSD de unix. La licencia BSD simplemente requiere que todo el código conserve el aviso de licencia BSD si se redistribuye en formato de código fuente, o que reproduzca el aviso si se redistribuye en formato binario. La licencia BSD 3-clause contiene un descargo de responsabilidad en la documentación y materiales proporcionados con la distribución. Además no pueden utilizarse sin permiso con fines de respaldo o promoción los nombres del titular de los derechos de autor ni de sus colaboradores.
- **GNU-GPLv3:** Esta licencia permite el uso comercial del software, su distribución, modificación y uso privado. Quien haga alguno de los usos anteriores está obligado a licenciar como código libre cualquier modificación del código o herramientas incorporadas con esta licencia.
- **Apache 2.0:** Esta licencia tiene propiedades iguales que la licencia GPLv3. No obstante, no obliga a que los nuevos desarrollos con dependencias con Apache 2.0 se licencien como código libre.
- **MIT:** La licencia MIT permite uso comercial del software licenciado, su modificación, su libre distribución y el uso privado. No presenta

garantías ni responsabilidad. La única condición que impone es hacer referencia a tal licencia. No obliga a mantener la licencia, ni limita la distribución del software que use productos bajo esta licencia. Esto implica que la licencia final puede ser cualquiera.

- **PSF:** Es una licencia de software libre, que al modo de BSD, es permisiva. Cumple con los requisitos OSI de licencia de software libre y es compatible con la licencia GPL. Al modo de las licencias BSD, permite modificaciones del código fuente y la creación de trabajos derivados, y no obliga a que tales modificaciones ni trabajos derivados tengan que ser a su vez de código abierto.

De todas las licencias descritas con anterioridad, la más restrictiva es la licencia GPLv3 la cual ha sido aplicada a nuestra aplicación una vez desarrollada, permitiendo tanto su uso comercial como privado.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este anexo se van a exponer las especificaciones de requisitos que definen el funcionamiento de la plataforma, comenzando por un repaso de objetivos de esta, exponiéndose los requisitos funcionales y no funcionales derivados de estos y finalizando con la exposición de los casos de uso resultantes de su desarrollo.

B.2. Objetivos generales

Como había sido mencionado previamente en la memoria, el objetivo principal de este proyecto consistía en el desarrollo de una plataforma web orientada a la enseñanza online la cual permita corregir de forma automática ejercicios en Python. Para ello, la plataforma debía basarse en la inclusión de cursos de carácter didáctico por parte de los profesores en los que se encontrarían las tareas y los contenidos teóricos de estos.

B.3. Catálogo de requisitos

En esta sección son enumerados y descritos los requisitos funcionales y no funcionales derivados del objetivo descrito con anterioridad.

Requisitos funcionales de un usuario sin autenticar:

- **RF-1 Inicio de sesión:** Un usuario inicia sesión en la plataforma y es identificado como profesor o estudiante y llevado a la página principal correspondiente.

Requisitos funcionales orientados al profesor:

- **RF-2 Gestionar cursos:** Un profesor debe poder realizar la gestión de sus cursos.
 - **RF-2.1 Gestionar contenidos de un curso:** Un profesor debe poder gestionar los contenidos de cada curso.
 - **RF-2.1.1: Descargar contenidos y tareas:** Un profesor debe poder descargar los contenidos teóricos y tareas tanto en versión alumno como versión profesor de las secciones contenidas en el curso.
 - **RF-2.1.2: Gestionar secciones no publicadas:** Un profesor debe poder acceder a la gestión de las secciones no publicadas de ese curso.
 - ◊ **RF-2.1.2.1: Crear sección no publicada:** Un profesor debe poder crear nuevas secciones no publicadas para un curso.
 - ◊ **RF-2.1.2.2: Editar tarea:** Un profesor debe poder editar las tareas de las secciones no publicadas desde la propia plataforma.
 - ◊ **RF-2.1.2.3: Descargar estado de la tarea:** Un profesor debe poder descargarse el estado actual en el que se encuentran las tareas de las secciones no publicadas.
 - ◊ **RF-2.1.2.4: Eliminar sección no publicada:** Un profesor debe poder eliminar secciones no publicadas.
 - ◊ **RF-2.1.2.5: Publicar sección no publicada:** Un profesor debe poder publicar secciones no publicadas.

- **RF-2.1.3: Eliminar sección de contenido:** Un profesor debe poder eliminar secciones de contenido ya publicadas de un curso.
- **RF-2.1.4: Añadir sección de contenido:** Un profesor debe poder crear secciones de contenido de forma directa.
- **RF-2.2 Gestionar alumnos del curso:** Un profesor debe poder realizar la gestión de alumnos integrantes en cada curso.
 - **RF-2.2.1: Quitar alumno de un curso:** Un profesor podrá quitar alumnos de un curso.
 - **RF-2.2.2: Añadir alumno a un curso:** Un profesor podrá añadir alumnos a un curso.
- **RF-2.3 Crear curso:** Un profesor podrá crear un curso nuevo.
- **RF-2.4 Eliminar curso:** Un profesor podrá eliminar un curso existente.
- **RF-3 Gestionar alumnos:** Un profesor debe poder acceder a la gestión de alumnos integrantes en sus cursos.
 - **RF-3.1: Crear alumno:** Un profesor podrá crear un nuevo alumno, integrándolo en un curso.
 - **RF-3.2: Borrar alumno:** Un profesor podrá eliminar por completo un alumno.
 - **RF-3.3: Acceder a calificaciones del alumno:** Un profesor podrá acceder a la visualización de calificaciones de un alumno en particular.
 - **RF-3.3.1: Eliminar entregas:** Un profesor podrá eliminar entregas de tareas de un alumno desde la vista de sus calificaciones.
 - **RF-3.3.2: Descargar feedbacks:** Un profesor podrá descargarse los archivos feedback de cada entrega / calificación de un alumno.
 - **RF-3.3.3: Descargar entregas:** Un profesor podrá descargarse los archivos tarea de cada entrega / calificación de un alumno.
- **RF-4 Consultar todas las calificaciones:** Un profesor podrá acceder a la visualización total de las calificaciones de los alumnos pertenecientes a sus cursos.
 - **RF-4.1: Descargar feedbacks:** Un profesor podrá descargarse los archivos feedback de cada entrega / calificación de todas las entregas de sus alumnos.

- **RF-4.2: Descargar entregas:** Un profesor podrá descargarse los archivos tarea de cada entrega / calificación de todas las entregas de sus alumnos.

Requisitos funcionales orientados al alumno:

- **RF-5 Acceder a cursos inscrito:** Un alumno debe poder acceder al listado de cursos a los que está inscrito.
 - **RF-5.1 Acceder a los contenidos de un curso:** Un alumno debe poder acceder a los contenidos de cada uno de los cursos a los que está inscrito.
 - **RF-5.1.1 Descargar contenidos y tareas:** Un alumno debe poder descargarse los contenidos teóricos y tareas de cada sección de contenido de un curso.
 - **RF-5.1.2 Entregar tareas:** Un alumno debe poder realizar la entrega de tareas de las secciones de un curso.
 - **RF-5.1.3 Descargar entregas:** Un alumno debe poder descargarse las entregas previamente realizadas de las secciones de un curso.
 - **RF-5.2 Consultar calificaciones de un curso:** Un alumno debe poder consultar sus calificaciones obtenidas en las tareas de un curso en particular.
 - **RF-5.2.1 Descargar entregas:** Un alumno debe poder descargarse el archivo tarea enviado en cada entrega / calificación de un curso.
 - **RF-5.2.2 Descargar feedbacks:** Un alumno debe poder descargarse el archivo feedback obtenido en cada entrega / calificación de un curso.
- **RF-6 Consultar todas mis calificaciones:** Un alumno debe poder consultar la totalidad de sus calificaciones obtenidas en todos los cursos a los que está inscrito.
 - **RF-6.1 Descargar entregas:** Un alumno debe poder descargarse el archivo tarea enviado en cada entrega / calificación.
 - **RF-6.2 Descargar feedbacks:** Un alumno debe poder descargarse el archivo feedback obtenido en cada entrega / calificación.

Requisitos funcionales comunes a profesores y alumnos:

- **RF-7 Cambio de contraseña:** Un usuario profesor o alumno debe poder realizar el cambio de contraseña de su cuenta.
- **RF-8 Cierre de sesión:** Un usuario profesor o alumno debe poder realizar el cierre de sesión de su cuenta.

Requisitos no funcionales:

- **RNF.1 Usabilidad** El diseño de la plataforma busca la facilidad de uso e intuitividad.
- **RNF.2 Disponibilidad** La aplicación siempre ha de estar disponible para su uso, siempre y cuando se tenga conexión a internet.
- **RNF.3 Seguridad** Los datos sensibles pertenecientes a los usuarios permanecen aislados en todo momento.
- **RNF.4 Mantenibilidad** El diseño de la plataforma permite una adecuada mantenibilidad.
- **RNF.5 Escalabilidad** La adición de nuevas funcionales es posible y sencillo de realizar.

B.4. Especificación de requisitos

En esta sección son incluidos los diagramas de casos de uso para usuarios sin autenticar, profesores y alumnos y el desarrollo de cada uno de ellos:

Diagramas de casos de uso

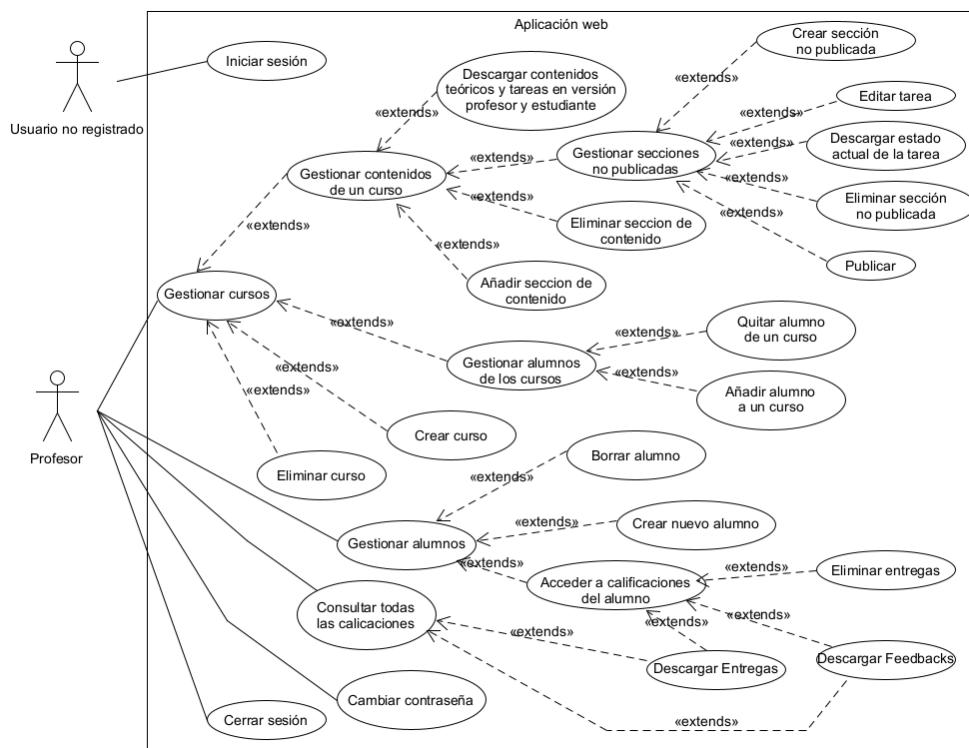


Figura B.1: Casos de Uso Profesor e Inicio de sesión

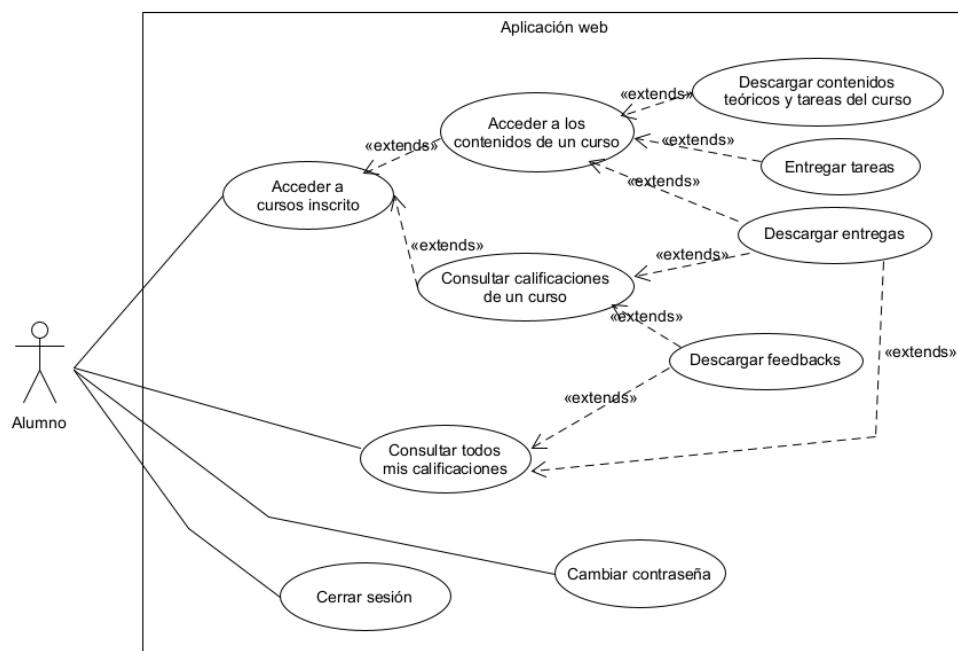


Figura B.2: Casos de Uso Alumno

Casos de uso

Caso de uso 1: Inicio de sesión

Descripción	Permite a un usuario sin autenticar iniciar sesión en la plataforma								
Requisitos relacionados	RF-1								
Precondiciones	El usuario dispone de una cuenta en la plataforma								
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El usuario accede al login de la plataforma</td></tr> <tr> <td>2</td><td>El usuario introduce su usuario y contraseña</td></tr> <tr> <td>3</td><td>Se inicia sesión</td></tr> </tbody> </table>	Paso	Acción	1	El usuario accede al login de la plataforma	2	El usuario introduce su usuario y contraseña	3	Se inicia sesión
Paso	Acción								
1	El usuario accede al login de la plataforma								
2	El usuario introduce su usuario y contraseña								
3	Se inicia sesión								
Postcondiciones	El usuario es llevado a la página principal "Inicio" correspondiente en caso de ser un alumno o un profesor								
Excepciones	Datos erróneos La cuenta no existe								

Caso de uso 2: Gestionar cursos

Descripción	Permite a un profesor realizar la gestión de sus cursos				
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.1, RF-2.1.2, RF-2.1.2.1, RF-2.1.2.2, RF-2.1.2.3, RF-2.1.2.4, RF-2.1.2.5, RF-2.1.3, RF-2.1.4, RF-2.2, RF-2.2.1, RF-2.2.2, RF-2.3, RF-2.4				
Precondiciones	El profesor ha iniciado sesión				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>El profesor accede a la página de "cursos" desde la barra navegacional o desde el botón de "cursos" de la página principal "Inicio"</td></tr> </tbody> </table>	Paso	Acción	1	El profesor accede a la página de "cursos" desde la barra navegacional o desde el botón de "cursos" de la página principal "Inicio"
Paso	Acción				
1	El profesor accede a la página de "cursos" desde la barra navegacional o desde el botón de "cursos" de la página principal "Inicio"				
Postcondiciones	El profesor es llevado a la página de gestión de sus cursos "cursos"				

Caso de uso 2.1: Gestionar contenidos de un curso

Descripción	Permite a un profesor realizar la gestión de contenidos de un curso en particular	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.1, RF-2.1.2, RF-2.1.2.1, RF-2.1.2.2, RF-2.1.2.3 RF-2.1.2.4, RF-2.1.2.5, RF-2.1.3, RF-2.1.4	
Precondiciones	El profesor ha accedido a la vista de gestión de cursos y el curso existe	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón "Acceder" del curso del cual quiere hacer la gestión de contenidos
Postcondiciones	El profesor es llevado a la página de contenidos del curso en particular	
Excepciones	El curso no existe	

Caso de uso 2.1.1: Descargar contenidos y tareas

Descripción	Permite a un profesor descargar los archivos de contenidos teóricos y tareas de las secciones de contenido del curso actual	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.1, RF-2.1.2.2, RF-2.1.2.3 RF-2.1.2.4, RF-2.1.2.5	
Precondiciones	El profesor ha accedido a la vista de contenidos de un curso	
Secuencia	Paso	Acción
	1	El profesor hace click en los botones de descarga de contenidos teóricos y tareas de las secciones del curso
Postcondiciones	El profesor recibe los archivos que ha solicitado descargar	
Excepciones	La sección no existe	

Caso de uso 2.1.2: Gestionar secciones no publicadas

Descripción	Permite a un profesor realizar la gestión de secciones de contenidos no publicadas de un curso	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.1, RF-2.1.2.2, RF-2.1.2.3 RF-2.1.2.4, RF-2.1.2.5	
Precondiciones	El profesor ha accedido a la vista de contenidos de un curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de acceso a las secciones no publicadas del curso
Postcondiciones	El profesor es llevado a la página de secciones no publicadas del curso actual	

Caso de uso 2.1.2.1: Crear sección no publicada

Descripción	Permite a un profesor crear una nueva sección no publicada en el curso actual	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.1	
Precondiciones	El profesor ha accedido a la vista de contenidos de un curso o a la vista de secciones no publicadas de ese curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de creación de sección no publicadas desde la vista del curso o la vista de secciones no publicadas del curso
Postcondiciones	Es creada una nueva sección no publicada y aparecerá en la vista de secciones no publicadas del curso	
Excepciones	Información necesaria no aportada Contenidos pertenecientes a otra sección	

Caso de uso 2.1.2.2: Editar tarea

Descripción	Permite a un profesor editar la tarea de una sección no publicada	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.2	
Precondiciones	El profesor ha accedido a la vista de secciones no publicadas de ese curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón edición de tarea de una sección no publicada
	2	El profesor edita la tarea desde la vista del notebook tarea que es abierta al realizar el paso anterior
	3	El profesor guarda los cambios en el botón de guardado del notebook
Postcondiciones	La tarea es editada	
Excepciones	La sección no publicada no existe	

Caso de uso 2.1.2.3: Descargar estado de la tarea

Descripción	Permite a un profesor descargar el estado actual de las tareas de las secciones no publicadas	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.3	
Precondiciones	El profesor ha accedido a la vista de secciones no publicadas de ese curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de descarga de la tarea de una sección no publicada
Postcondiciones	El profesor recibe el notebook tarea seleccionado	
Excepciones	La sección no publicada no existe	

Caso de uso 2.1.2.4: Eliminar sección no publicada

Descripción	Permite a un profesor eliminar secciones no publicadas	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.4	
Precondiciones	El profesor ha accedido a la vista de secciones no publicadas de ese curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de eliminación de una sección no publicada
	2	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	La sección no publicada es eliminada	
Excepciones	La sección no publicada no existe	

Caso de uso 2.1.2.5: Publicar sección no publicada

Descripción	Permite a un profesor publicar secciones no publicadas	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.2, RF-2.1.2.5	
Precondiciones	El profesor ha accedido a la vista de secciones no publicadas de ese curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de publicación de una sección no publicada
	2	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	La sección no publicada es publicada, aparecerá en la vista de contenido del curso y se genera el archivo tarea correspondiente a esa sección para los alumnos	
Excepciones	La sección no publicada no existe	

Caso de uso 2.1.3: Eliminar sección de contenido

Descripción	Permite a un profesor eliminar secciones de contenido del curso actual	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.3	
Precondiciones	El profesor ha accedido a la vista de contenidos de un curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de eliminación de una sección de contenido del curso
	2	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	La sección de contenido es eliminada	
Excepciones	La sección no existe	

Caso de uso 2.1.4: Eliminar sección de contenido

Descripción	Permite a un profesor crear una sección de contenido de forma directa	
Requisitos relacionados	RF-2, RF-2.1, RF-2.1.4	
Precondiciones	El profesor ha accedido a la vista de contenidos de un curso	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de creación de sección de contenido y es llevado a la vista de creación de sección
	2	El profesor completa el formulario de creación de sección en el que introducirá el archivo de contenido teórico de la sección y el notebook tarea correspondiente a esa sección
	3	El profesor hace click en el botón de crear
	4	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	La sección de contenido es creada y se genera el archivo tarea correspondiente a esa sección para los alumnos	

Caso de uso 2.2: Gestionar alumnos de un curso

Descripción	Permite a un profesor realizar la gestión de alumnos integrantes de un curso en particular				
Requisitos relacionados	RF-2, RF-2.2, RF-2.2.1, RF-2.2.2				
Precondiciones	El profesor ha accedido a la vista de gestión de cursos y el curso existe				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El profesor hace click en el botón de gestión de estudiantes del curso del cual quiere realizar la gestión</td> </tr> </tbody> </table>	Paso	Acción	1	El profesor hace click en el botón de gestión de estudiantes del curso del cual quiere realizar la gestión
Paso	Acción				
1	El profesor hace click en el botón de gestión de estudiantes del curso del cual quiere realizar la gestión				
Postcondiciones	El profesor es llevado a la página de estudiantes integrantes y no integrantes del curso en particular				
Excepciones	El curso no existe				

Caso de uso 2.2.1: Quitar alumno de un curso

Descripción	Permite a un profesor sacar a un alumno del curso actual						
Requisitos relacionados	RF-2, RF-2.2, RF-2.2.1						
Precondiciones	El profesor ha accedido a la vista de estudiantes integrantes y no integrantes del curso en particular						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El profesor hace click en el botón de extracción de un alumno del curso</td> </tr> <tr> <td>2</td> <td>El profesor hace click en el botón de confirmación del modal emergente</td> </tr> </tbody> </table>	Paso	Acción	1	El profesor hace click en el botón de extracción de un alumno del curso	2	El profesor hace click en el botón de confirmación del modal emergente
Paso	Acción						
1	El profesor hace click en el botón de extracción de un alumno del curso						
2	El profesor hace click en el botón de confirmación del modal emergente						
Postcondiciones	El alumno es extraído del curso						
Excepciones	El alumno no pertenecía al curso						

Caso de uso 2.2.2: Añadir alumno a un curso

Descripción	Permite a un profesor añadir a un alumno al curso actual	
Requisitos relacionados	RF-2, RF-2.2, RF-2.2.2	
Precondiciones	El profesor ha accedido a la vista de estudiantes integrantes y no integrantes del curso en particular	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de adición de un alumno al curso
	2	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El alumno es añadido al curso	
Excepciones	El alumno ya pertenecía al curso	

Caso de uso 2.3: Crear curso

Descripción	Permite a un profesor crear un nuevo curso	
Requisitos relacionados	RF-2, RF-2.3	
Precondiciones	El profesor ha accedido a la vista de gestión de cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de creación de un nuevo curso y es llevado a la vista de creación de curso
	2	El profesor completa el formulario de creación de un curso
	3	El profesor hace click en el botón de crear curso
	4	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El curso es creado y visualizable en la vista de gestión de cursos	
Excepciones	Información necesaria no aportada Contenidos pertenecientes a otro curso	

Caso de uso 2.4: Eliminar curso

Descripción	Permite a un profesor eliminar un curso	
Requisitos relacionados	RF-2, RF-2.4	
Precondiciones	El profesor ha accedido a la vista de gestión de cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de eliminación de un curso
	2	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El curso es eliminado	
Excepciones	El curso no existe	

Caso de uso 3: Gestionar alumnos

Descripción	Permite a un profesor realizar la gestión de alumnos pertenecientes a sus cursos	
Requisitos relacionados	RF-3, RF-3.1, RF-3.2, RF-3.3, RF-3.3.1, RF-3.3.2, RF-3.3.3	
Precondiciones	El profesor ha iniciado sesión	
Secuencia	Paso	Acción
	1	El profesor accede a la página de gestión de alumnos desde la barra navegacional o desde el botón de gestión de alumnos de la página principal
Postcondiciones	El profesor es llevado a la página de gestión de alumnos	

Caso de uso 3.1: Crear alumno

Descripción	Permite a un profesor crear un nuevo alumno	
Requisitos relacionados	RF-3, RF-3.1	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de creación de un nuevo alumno y es llevado a la vista de creación de alumno
	2	El profesor completa el formulario de creación del alumno y selecciona un curso al que quiere ingresar al alumno
	3	El profesor hace click en el botón de crear alumno
	4	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El alumno es creado y visualizable en la vista de gestión de alumnos	
Excepciones	Información necesaria no aportada Contenidos pertenecientes a otro alumno	

Caso de uso 3.2: Borrar alumno

Descripción	Permite a un profesor borrar un alumno	
Requisitos relacionados	RF-3, RF-3.2	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el nombre del estudiante accediendo al perfil este en el que se visualizan las calificaciones
	2	El profesor hace click en el botón de eliminación del alumno
	3	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El alumno es eliminado junto con su cuenta	
Excepciones	El alumno no existe	

Caso de uso 3.3: Acceder a las calificaciones de un alumno

Descripción	Permite a un profesor visualizar todas las calificaciones de un alumno en particular	
Requisitos relacionados	RF-3, RF-3.3, RF-3.3.1, RF-3.3.2, RF-3.3.3	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el nombre del estudiante accediendo al perfil este en el que se visualizan las calificaciones
Postcondiciones	El profesor visualiza las calificaciones	
Excepciones	El alumno no existe	

Caso de uso 3.3.1: Eliminar entregas

Descripción	Permite a un profesor eliminar entregas / calificaciones de tareas realizadas por un alumno en particular	
Requisitos relacionados	RF-3, RF-3.3, RF-3.3.1	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el nombre del estudiante accediendo al perfil este en el que se visualizan las calificaciones
	2	El profesor hace click en el botón de eliminación de entrega de una calificación en particular
	3	El profesor hace click en el botón de confirmación del modal emergente
Postcondiciones	El profesor elimina la entrega	
Excepciones	La calificación no existe	

Caso de uso 3.3.2: Descargar feedback

Descripción	Permite a un profesor descargar los archivos de feedback de las entregas / calificaciones de un alumno en particular	
Requisitos relacionados	RF-3, RF-3.3, RF-3.3.2	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el nombre del estudiante accediendo al perfil este en el que se visualizan las calificaciones
	2	El profesor hace click en el botón de descarga de feedback de una calificación en particular
Postcondiciones	El profesor recibe el archivo feedback	
Excepciones	La calificación no existe	

Caso de uso 3.3.3: Descargar entregas

Descripción	Permite a un profesor descargar los archivos entregados de las entregas / calificaciones de un alumno en particular	
Requisitos relacionados	RF-3, RF-3.3, RF-3.3.3	
Precondiciones	El profesor ha accedido a la vista de gestión de alumnos pertenecientes a sus cursos	
Secuencia	Paso	Acción
	1	El profesor hace click en el nombre del estudiante accediendo al perfil este en el que se visualizan las calificaciones
	2	El profesor hace click en el botón de descarga de entrega de una calificación en particular
Postcondiciones	El profesor recibe el archivo de entrega del alumno	
Excepciones	La calificación no existe	

Caso de uso 4: Consultar todas las calificaciones

Descripción	Permite a un profesor acceder a todas las calificaciones de todos los alumnos pertenecientes a sus cursos	
Requisitos relacionados	RF-4, RF-4.1, RF-4.2	
Precondiciones	El profesor ha iniciado sesión	
Secuencia	Paso	Acción
	1	El profesor accede a la página de calificaciones desde la barra navegacional o desde el botón de calificaciones de la página principal
Postcondiciones	El profesor es llevado a la página de calificaciones	

Caso de uso 4.1: Descargar feedback

Descripción	Permite a un profesor descargar los archivos de feedback de las entregas / calificaciones de sus alumnos	
Requisitos relacionados	RF-4, RF-4.1	
Precondiciones	El profesor ha accedido a la vista de calificaciones	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de descarga de feedback de una calificación en particular
Postcondiciones	El profesor recibe el archivo feedback	
Excepciones	La calificación no existe	

Caso de uso 4.2: Descargar entregas

Descripción	Permite a un profesor descargar los archivos entregados de las entregas / calificaciones de sus alumnos	
Requisitos relacionados	RF-4, RF-4.2	
Precondiciones	El profesor ha accedido a la vista de calificaciones	
Secuencia	Paso	Acción
	1	El profesor hace click en el botón de descarga de entrega de una calificación en particular
Postcondiciones	El profesor recibe el archivo de entrega del alumno	
Excepciones	La calificación no existe	

Caso de uso 5: Acceder a cursos inscrito

Descripción	Permite a un alumno acceder a los cursos a los que está inscrito				
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.1, RF-5.1.2, RF-5.1.3				
Precondiciones	El alumno ha iniciado sesión				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno accede a la página de cursos desde la barra navegacional o desde el botón de cursos de la página principal</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno accede a la página de cursos desde la barra navegacional o desde el botón de cursos de la página principal
Paso	Acción				
1	El alumno accede a la página de cursos desde la barra navegacional o desde el botón de cursos de la página principal				
Postcondiciones	El alumno es llevado a la página de cursos				

Caso de uso 5.1: Acceder a contenidos de un curso

Descripción	Permite a un alumno realizar acceder a los contenidos de un curso en particular				
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.1, RF-5.1.2, RF-5.1.3				
Precondiciones	El alumno ha accedido a la vista de cursos y el curso existe				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno hace click en el botón de acceso al curso</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno hace click en el botón de acceso al curso
Paso	Acción				
1	El alumno hace click en el botón de acceso al curso				
Postcondiciones	El alumno es llevado a la página de contenidos del curso en particular				
Excepciones	El curso no existe				

Caso de uso 5.1.1: Descargar contenidos y tareas

Descripción	Permite a un alumno descargar los contenidos teóricos y tareas de las secciones pertenecientes al curso en particular				
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.1				
Precondiciones	El alumno ha accedido a la vista del curso y la sección de contenido existe				
Secuencia	<table> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno hace click los botones de descarga de contenido teórico y tareas de la sección en particular de la que quiere obtener los archivos</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno hace click los botones de descarga de contenido teórico y tareas de la sección en particular de la que quiere obtener los archivos
Paso	Acción				
1	El alumno hace click los botones de descarga de contenido teórico y tareas de la sección en particular de la que quiere obtener los archivos				
Postcondiciones	El alumno recibe los archivos pedidos				
Excepciones	La sección de contenido no existe				

Caso de uso 5.1.2: Entregar tareas

Descripción	Permite a un alumno realizar la entrega de tareas de un curso						
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.2						
Precondiciones	El alumno ha accedido a la vista del curso y la sección de contenido existe						
Secuencia	<table> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno hace click en el botón de exploración de archivos de la sección de contenido en la que quiere entregar la tarea</td> </tr> <tr> <td>2</td> <td>El alumno hace click en el botón de entrega de la tarea</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno hace click en el botón de exploración de archivos de la sección de contenido en la que quiere entregar la tarea	2	El alumno hace click en el botón de entrega de la tarea
Paso	Acción						
1	El alumno hace click en el botón de exploración de archivos de la sección de contenido en la que quiere entregar la tarea						
2	El alumno hace click en el botón de entrega de la tarea						
Postcondiciones	La entrega es realizada, la calificación es calculada y el archivo de feedback es generado						
Excepciones	La sección de contenido no existe La tarea ya ha sido entregada previamente						

Caso de uso 5.1.3: Descargar entregas

Descripción	Permite a un alumno descargar las entregas de las tareas previamente entregadas desde la propia vista del curso	
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.3	
Precondiciones	El alumno ha accedido a la vista del curso y la sección de contenido existe y ha sido entregada su tarea	
Secuencia	Paso	Acción
	1	El alumno hace click en el botón de descarga de la entrega en la sección correspondiente
Postcondiciones	El alumno recibe el archivo de entrega	
Excepciones	La sección de contenido no existe La tarea no ha sido entregada	

Caso de uso 5.2: Consultar calificaciones de un curso

Descripción	Permite a un alumno consultar las calificaciones obtenidas en un curso en particular	
Requisitos relacionados	RF-5, RF-5.2, RF-5.2.1, RF-5.2.2	
Precondiciones	El alumno ha accedido a la vista de cursos y el curso existe	
Secuencia	Paso	Acción
	1	El alumno hace click en el botón de calificaciones del curso
Postcondiciones	El alumno es llevado a la página de calificaciones del curso en particular	
Excepciones	El curso no existe	

Caso de uso 5.2.1: Descargar entregas

Descripción	Permite a un alumno descargar las entregas de las tareas previamente entregadas desde la vista de calificaciones de un curso particular	
Requisitos relacionados	RF-5, RF-5.2, RF-5.2.1	
Precondiciones	El alumno ha accedido a la vista de calificaciones de un curso particular	
Secuencia	Paso	Acción
	1	El alumno hace click en el botón de descarga de entrega de una calificación en particular
Postcondiciones	El alumno recibe el archivo de entrega	
Excepciones	La calificación no existe	

Caso de uso 5.2.2: Descargar feedback

Descripción	Permite a un alumno descargar los archivos de feedback de las entregas / calificaciones desde la vista de calificaciones de un curso particular	
Requisitos relacionados	RF-5, RF-5.2, RF-5.2.2	
Precondiciones	El alumno ha accedido a la vista de calificaciones de un curso particular	
Secuencia	Paso	Acción
	1	El alumno hace click en el botón de descarga de feedback de una calificación en particular
Postcondiciones	El alumno recibe el archivo de feedback	
Excepciones	La calificación no existe	

Caso de uso 6: Consultar todas mis calificaciones

Descripción	Permite a un alumno consultar todas sus calificaciones obtenidas				
Requisitos relacionados	RF-5, RF-5.1, RF-5.1.1, RF-5.1.2, RF-5.1.3				
Precondiciones	El alumno ha iniciado sesión				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno accede a la página de calificaciones desde la barra navegacional o desde el botón de calificaciones de la página principal</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno accede a la página de calificaciones desde la barra navegacional o desde el botón de calificaciones de la página principal
Paso	Acción				
1	El alumno accede a la página de calificaciones desde la barra navegacional o desde el botón de calificaciones de la página principal				
Postcondiciones	El alumno es llevado a la página calificaciones				

Caso de uso 6.1: Descargar entregas

Descripción	Permite a un alumno descargar las entregas de las tareas previamente entregadas desde la vista de calificaciones				
Requisitos relacionados	RF-6, RF-6.1				
Precondiciones	El alumno ha accedido a la vista de todas sus calificaciones				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>El alumno hace click en el botón de descarga de entrega de una calificación en particular</td> </tr> </tbody> </table>	Paso	Acción	1	El alumno hace click en el botón de descarga de entrega de una calificación en particular
Paso	Acción				
1	El alumno hace click en el botón de descarga de entrega de una calificación en particular				
Postcondiciones	El alumno recibe el archivo de entrega				
Excepciones	La calificación no existe				

Caso de uso 6.2: Descargar feedback

Descripción	Permite a un alumno descargar los archivos de feedback de las entregas / calificaciones desde la vista de todas sus calificaciones	
Requisitos relacionados	RF-6, RF-6.2	
Precondiciones	El alumno ha accedido a la vista de todas sus calificaciones	
Secuencia	Paso	Acción
	1	El alumno hace click en el botón de descarga de feedback de una calificación en particular
Postcondiciones	El alumno recibe el archivo de feedback	
Excepciones	La calificación no existe	

Caso de uso 7: Cambio de contraseña

Descripción	Permite a un usuario (profesor o alumno) realizar el cambio de contraseña de su cuenta	
Requisitos relacionados	RF-7	
Precondiciones	El usuario ha iniciado sesión	
Secuencia	Paso	Acción
	1	El usuario hace click en el botón de cambio de contraseña de la página principal y es llevado a la vista de cambio de contraseña
	2	El usuario rellena el formulario de cambio de contraseña
	3	El usuario hace click en el botón de cambio de contraseña
	4	El usuario hace click en el botón de confirmación del modal emergente
Postcondiciones	La contraseña es cambiada y se produce un cierre de sesión llevandose al usuario al login	
Excepciones	Datos de entrada incorrectos	

Caso de uso 8: Cierre de sesión

Descripción	Permite a un usuario (profesor o alumno) realizar el cierre de sesión						
Requisitos relacionados	RF-7						
Precondiciones	El usuario ha iniciado sesión						
Secuencia	<table><thead><tr><th>Paso</th><th>Acción</th></tr></thead><tbody><tr><td>1</td><td>El usuario hace click en el botón de cierre de sesión de la página principal</td></tr><tr><td>2</td><td>El usuario hace click en el botón de confirmación del modal emergente</td></tr></tbody></table>	Paso	Acción	1	El usuario hace click en el botón de cierre de sesión de la página principal	2	El usuario hace click en el botón de confirmación del modal emergente
Paso	Acción						
1	El usuario hace click en el botón de cierre de sesión de la página principal						
2	El usuario hace click en el botón de confirmación del modal emergente						
Postcondiciones	Se produce el cierre de sesión llevandose al usuario al login						

Apéndice C

Especificación de diseño

C.1. Introducción

En este anexo será descrito el diseño, el cual ha sido llevado a cabo en el desarrollo de la plataforma buscando los objetivos y funcionalidades descritos anteriormente. En concreto, viendo el diseño de datos, el diseño procedimental y el diseño arquitectónico.

C.2. Diseño de datos

Las entidades definidas para el funcionamiento de la plataforma y presentes en base de datos son las siguientes:

- Usuario (User): correspondiente a los usuarios del sistema los cuales pueden ser de tipo profesor o de tipo alumno. Cuentan con identificador, nombre, nombre de usuario, contraseña, campo de control de tipo de usuario y campo de control de primer inicio de sesión.
- Curso (Course): correspondiente a los cursos del sistema. Cuenta con identificador, identificador del usuario profesor al que pertenece el curso, nombre y descripción
- Sección (Section): correspondiente a las secciones de los cursos. Cuenta con identificador, identificador del curso al que pertenece, nombre, nombre del archivo de contenido de la sección y nombre de la tarea de la sección.

- Calificación (Calification): Correspondiente a las calificaciones de los alumnos. Cuenta con el identificador del estudiante al que pertenece, identificador de la sección a la que pertenece, nombre de la tarea y el valor obtenido.
- Miembros del curso(Course Members): encargada de guardar las relaciones entre los cursos y los alumnos pertenecientes a estos. Cuenta con el identificador del curso y el del estudiante registrado en el curso.
- Sección sin publicar (Unreleased Section): correspondiente a secciones no publicadas dedicadas a la creación de tareas de forma manual por los profesores desde la plataforma. Cuenta con identificador, identificador del profesor al que pertenecen, identificador del curso al que pertenecen, nombre, nombre del archivo de contenido de la sección y nombre de la tarea de la sección.

Diagrama E/R

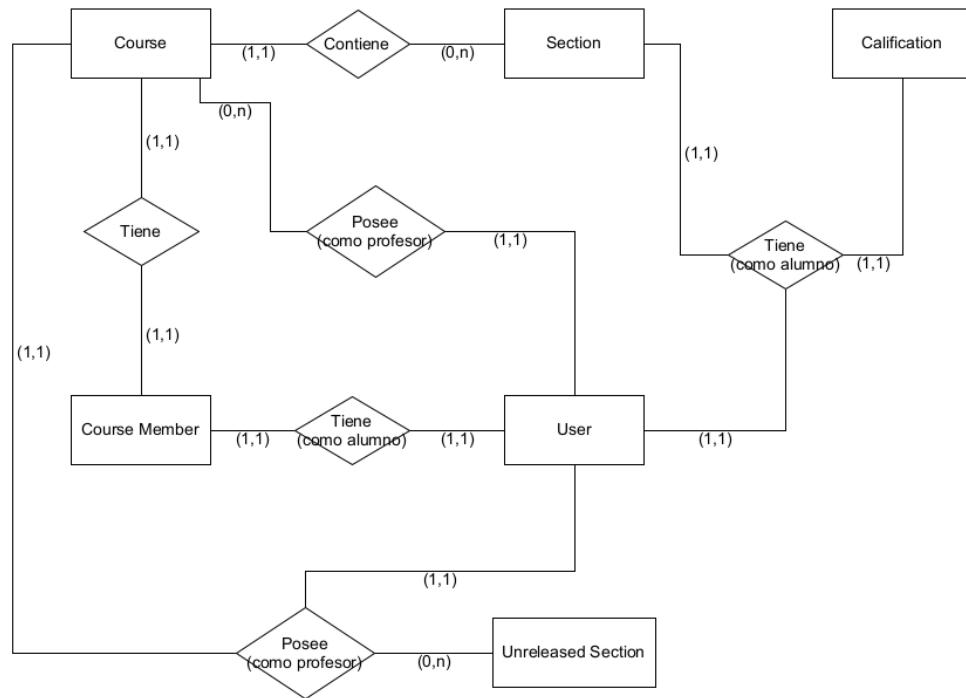


Figura C.1: Diagrama E/R

Diagrama Relacional

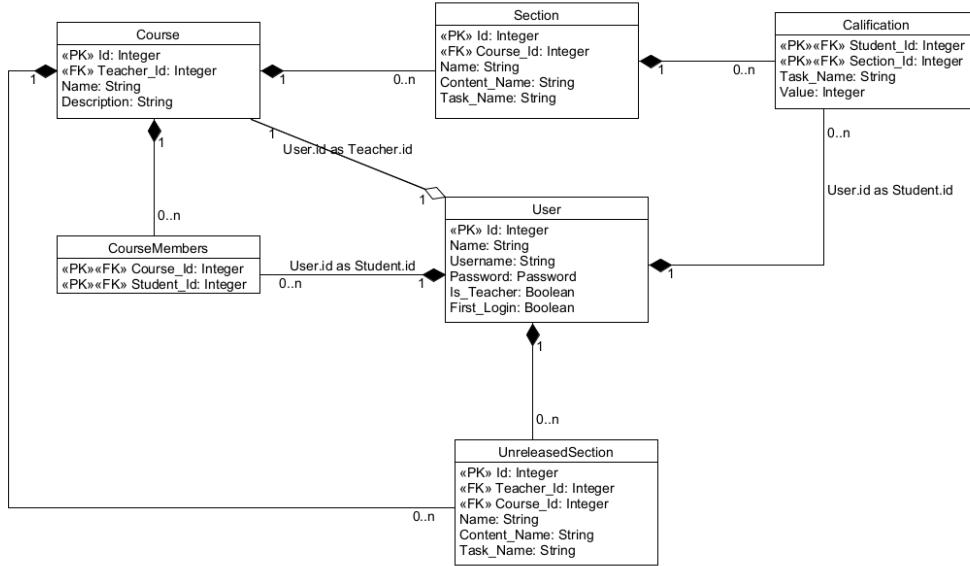


Figura C.2: Diagrama relacional

C.3. Diseño procedimental

En esta sección se van a recoger los aspectos más relevantes del diseño procedimental mediante la representación de los diagramas de secuencias de las funcionalidades fundamentales de la plataforma.

El siguiente diagrama de secuencias representa la realización y muestreo de consultas de datos por parte de un usuario las cuales son realizadas mediante SQLAlchemy, herramienta mencionada con anterioridad. En este tipo de consultas se encuentran la consulta de calificaciones por parte de un profesor o alumno, la consulta de alumnos por parte de un profesor, el muestreo de cursos y el muestreo de contenidos (secciones) de un curso:

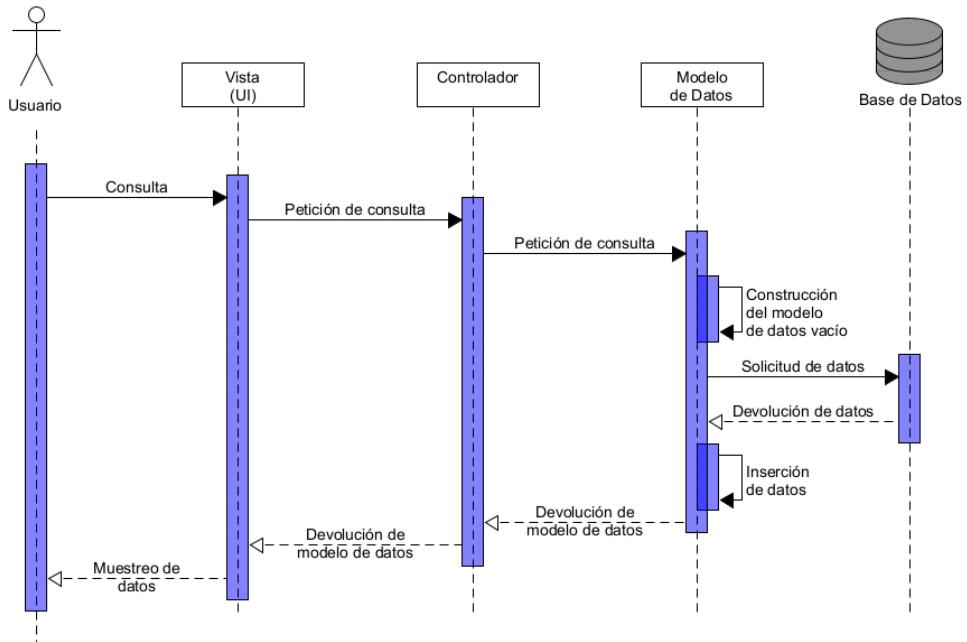


Figura C.3: Diagrama de secuencias de una consulta

El siguiente diagrama de secuencias representa la inserción o eliminación de datos por parte de un usuario, acciones también realizadas gracias a SQLAlchemy. En este tipo de acciones se encuentran la adición de alumnos, cursos y secciones por parte de un profesor, generación de calificaciones tras enviar una tarea por parte de un alumno y la eliminación de todos los tipos de datos anteriores:

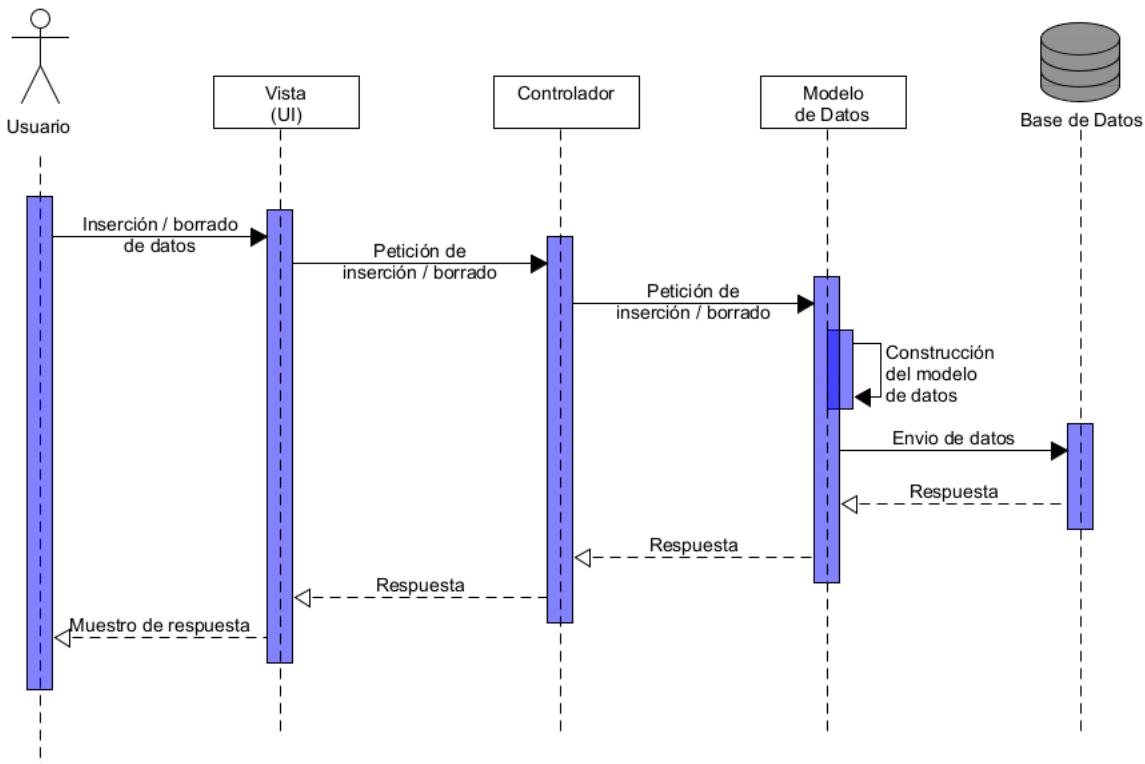


Figura C.4: Diagrama de secuencias de una inserción / borrado

Por último, en los siguientes diagramas se refleja el manejo de las acciones relacionadas con el autograding y generación de tareas mediante Nbgrader, acciones realizadas mediante la implementación de una clase de manejo (manager / gerente) de la API pública de esta herramienta:

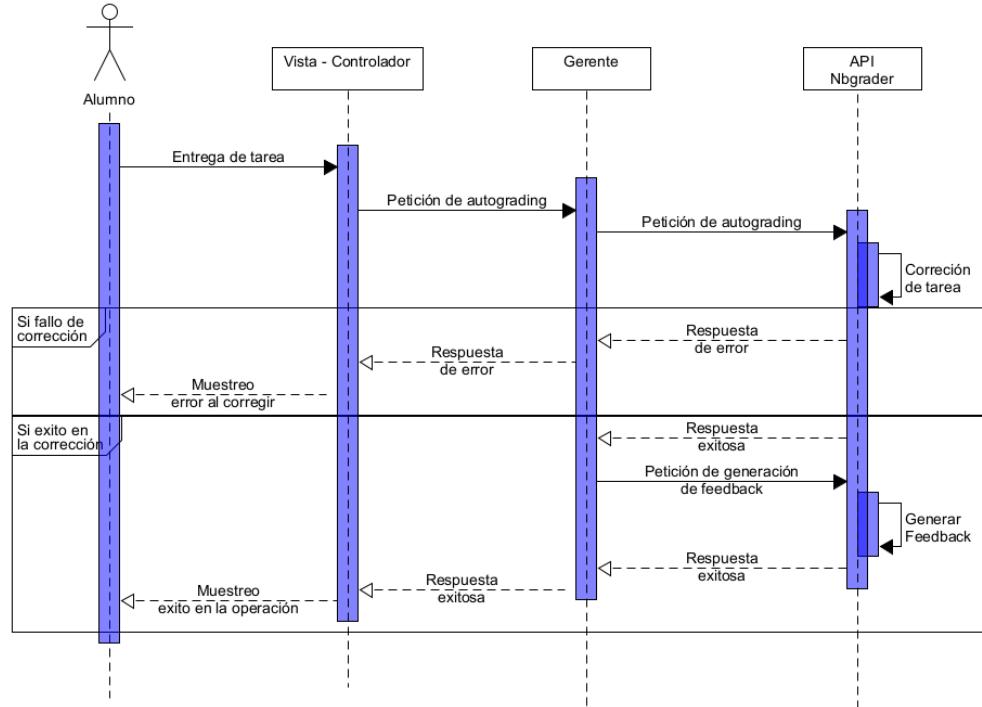


Figura C.5: Diagrama de secuencias autograding

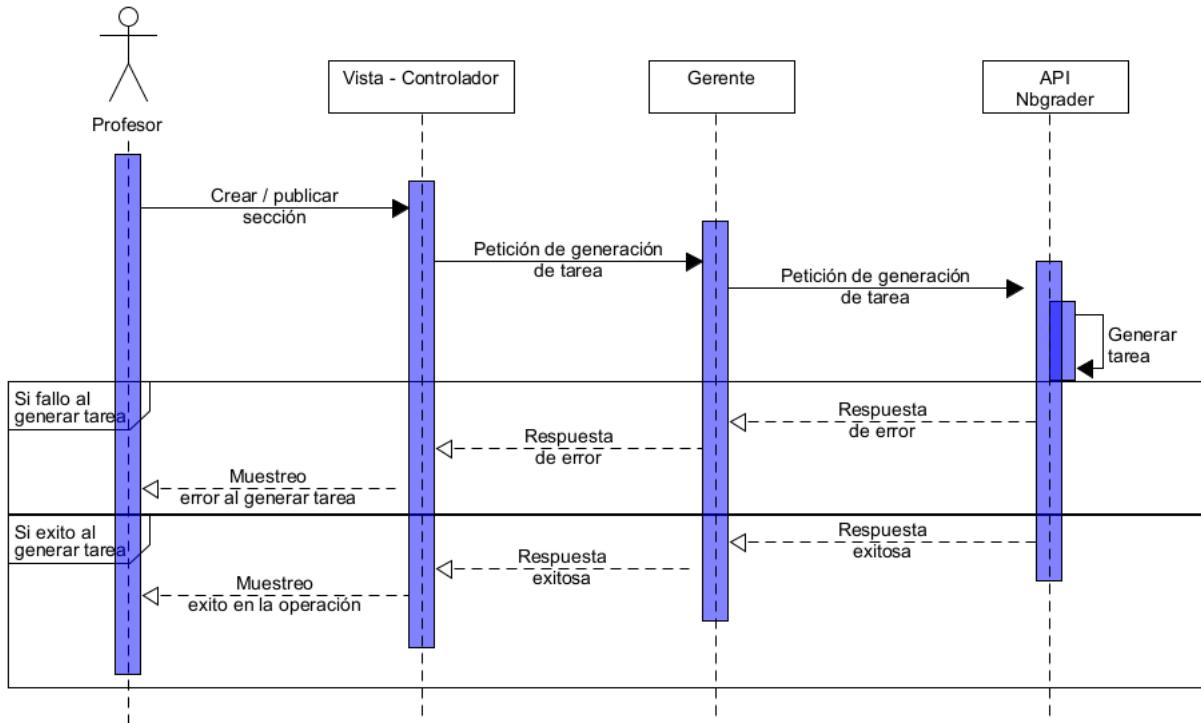


Figura C.6: Diagrama de secuencias generación de tarea

C.4. Diseño arquitectónico

Como ya se ha mencionado con anterioridad, el desarrollo de esta plataforma ha sido realizada mediante Flask. Flask es un framework minimalista no ligado a ningún diseño arquitectónico predefinido, lo cual ofrece a desarrolladores una gran libertad en cuanto a qué patrón arquitectónico dar a su proyecto.

En el caso de este proyecto, el patrón arquitectónico escogido ha sido **MVC (Modelo - Vista - Controlador)**:

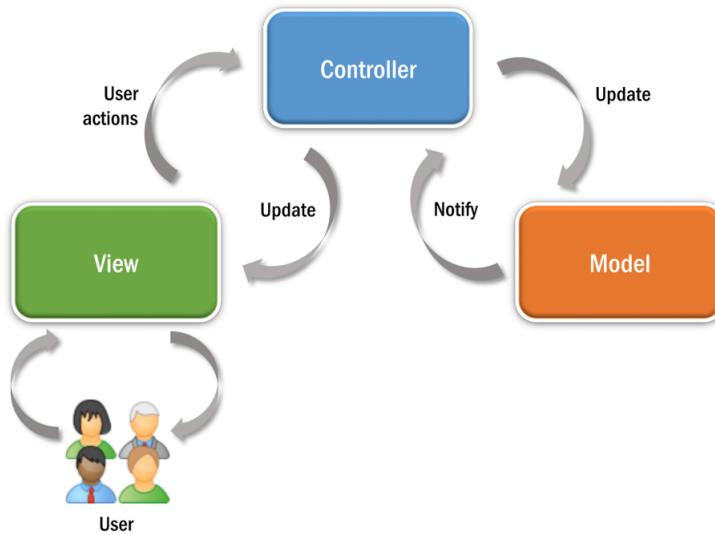


Figura C.7: Representación MVC

Este es un patrón software basado en la división de responsabilidades de una aplicación en tres capas diferentes:

- **Modelo:** Encargada de la representación y acceso a los datos del dominio junto con la lógica de negocio de estos.
- **Vista:** Responsable de la generación de la interfaz de usuario, medio por el que los clientes visualizan la información e interactúan con el sistema.
- **Controlador:** Intermediario entre las capas anteriores, encargado de gestionar el flujo de información entre las mismas.

En lo que a nuestro proyecto respecta, los métodos ruta o endpoints, que desempeñan la labor de Controlador, se encuentran en el fichero **routes.py**. La capa de Vista queda constituida por los directorios **templates** y **static** en los que se encuentran los documentos html, estilos e imágenes que componen la aplicación. La función de renderizado de vistas es desempeñada por el motor de templates Jinja2 sobre el que Flask está montado. Finalmente Flask no tiene predifinido ningún ORM (Asignación objeto-relacional) encargado de la función de Modelo y por ello la herramienta seleccionada para esta función es la mencionada con anterioridad SQLAlchemy.

Apéndice D

Documentación técnica de programación

D.1. Introducción

El propósito de esta sección es la documentación de los componentes internos del proyecto y los aspectos más relevante de su desarrollo a nivel técnico, orientado a aquellas personas con conocimientos en informática interesadas en el desarrollo del mismo.

D.2. Estructura de directorios

El repositorio del proyecto, **accesible en la siguiente URL: <https://github.com/AlbertoPorres/autograder-python>**, cuenta con la siguiente estructura de directorios:

- **/**: Directorio raiz del proyecto. Contiene el fichero README y LICENSE para el proyecto, el archivo *app.py* de ejecución de la aplicación, el archivo de requirements y Dockerfile para su instalación dentro de un contenedor Docker, el archivo de configuración de Jupyter Notebook y el resto de directorios del proyecto.
- **/env**: Carpeta correspondiente a VirtualEnv donde se encuentran instaladas todas las dependencias de la aplicación para su correcta ejecución en el ambiente de desarrollo.
- **/migrations**: Directorio de migraciones de la Base de Datos realizadas durante el desarrollo del proyecto.

- **/src:** Directorio de código fuente del proyecto. En él se encuentra el archivo `__init__.py` de configuración del proyecto flask, el archivo `forms.py` correspondiente al manejo de formularios mediante Flask-Forms, el archivo `management.py` correspondiente al manejo de la API de Nbgrader, el archivo `models.py` correspondiente a los modelos de datos de SQLAlchemy, el archivo `routes.py` donde se realizan las labores de controlador y la base de datos `database.db` de la aplicación.
- **/src/courses:** se encuentra el curso base que es utilizado como plantilla para la creación de nuevos cursos. En este directorio se almacenarán todos los cursos que se creen en la aplicación mediante el copiado y renombrado del curso base.
- **/src/courses/base_course:** curso base que contiene todos los directorios necesarios para el funcionamiento de Nbgrader (`autograded`, `content`, `feedback`, `release`, `source` y `submitted`), un notebook vacío utilizado para la creación de nuevas tareas dentro del curso mediante el copiado y renombrado del mismo, el archivo de configuración de Nbgrader `nbgrader_config.py`, la base de datos de Nbgrader `gradebook.db` y un directorio adicional `content` en el que se almacenan los archivos de contenido teórico de las secciones del curso.
- **/src/static:** Contenidos estáticos de la aplicación. En él se encuentra el directorio de imágenes y el archivo `base.css` de estilos.
- **/src/static/imagenes:** Directorio de imágenes.
- **/src/templates:** Directorio de archivos HTML.
- **/doc:** Directorio documentación del proyecto.
- **/doc/tex:** Directorio de archivos .text de la documentación.
- **/doc/img:** Directorio de imágenes de la documentación.
- **/Curso-Python:** Este directorio contiene los contenidos teóricos y tareas del curso de introducción a Python el cual ha sido añadido a la plataforma tras su despliegue vacío. Este contiene las 6 secciones de contenidos correspondientes a este curso (Introducción, Funciones, Listas y Tuplas, Control de Flujo, Diccionarios y Orientación a Objetos) divididas cada una en un directorio particular con su respectivo nombre. Dentro de cada directorio-sección se encuentran dos notebooks (archivos ipynb), el primero correspondiente al archivo de contenido teórico de la sección y cuyo nombre empieza por "T_" (teoría) y el segundo correspondiente al archivo tarea en versión del profesor y cuyo nombre empieza por "EV_" (evaluación).

D.3. Manual del programador

Este manual tiene como objetivo dar información a futuros programadores que trabajen en el proyecto sobre los aspectos más importantes del código y herramientas de este.

Prueba Nbgrader

Tras considerar diversas herramientas de autocorrección de ejercicios *Nbgrader* fue seleccionada como la herramienta de autograding a utilizar en nuestro proyecto. En esta sección queda documentada la primera toma de contacto con la herramienta en la que se realizó la instalación y prueba de esta. A diferencia de nuestro proyecto, en el cual las acciones de generación y corrección automática de tareas se realizan mediante el uso de la API pública de Nbgrader, en este primer manejo de la herramienta se hizo uso de la interfaz de usuario adicional que esta proporciona denominada Formgrader. Pese a ello, esta prueba demuestra la estructura de directorios de la que Nbgrader depende y nos proporciona nociones importantes sobre su funcionamiento:

Instalación y preparación de ficheros

Puesto que Nbgrader es un software de autocorrección de pruebas dependiente de Jupyter Notebook, el proceso de preparación de la herramienta para esta prueba fue el siguiente:

1. Instalación de la herramienta desde el terminal de Jupyter:

```
PS C:\Users\alber\Proyectos\NBgrader> conda install -c conda-forge nbgrader
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

PS C:\Users\alber\Proyectos\NBgrader>
```

Figura D.1: Proceso de instalación 1

2. Debido a la forma en la que nbgrader busca las tareas y test y opera con ellos, se debe de crear una estructura concreta de ficheros en el directorio raíz. En primer lugar, se crea el fichero “submitted” (enviados) en el que se almacenarán las tareas enviadas por los alumnos:

```
PS C:\Users\alber\Proyectos\NBgrader> mkdir ~/submitted

Directorio: C:\Users\alber

Mode          LastWriteTime    Length Name
----          -----          ---- 
d----        19/03/2022      17:26      submitted
```

Figura D.2: Proceso de instalación 2

3. Dentro del directorio “submitted” se debe de incluir un fichero para las entregas de cada alumno, en este caso solo tendremos al alumno “alum1”:

```
PS C:\Users\alber\Proyectos\NBgrader> mkdir ~/submitted/alum1

Directorio: C:\Users\alber\submitted

Mode          LastWriteTime    Length Name
----          -----          ---- 
d----        19/03/2022      17:32      alum1
```

Figura D.3: Proceso de instalación 3

4. Dentro de los ficheros individuales de cada alumno se debe crear un fichero por cada tarea encargada, en este caso definimos una única tarea, “task1”:

```
PS C:\Users\alber\Proyectos\NBgrader> mkdir ~/submitted/alum1/task1

Directorio: C:\Users\alber\submitted\alum1

Mode          LastWriteTime    Length Name
----          -----          ---- 
d----        19/03/2022      17:34      task1
```

Figura D.4: Proceso de instalación 4

5. Los ficheros tareas enviadas por los alumno almacenadas en estas carpetas tendrán el mismo nombre que la carpeta de esa tarea correspondiente con una extensión .ipynb (notebook de jupyter).

Lanzamiento y creación de una tarea

1. Una vez creada la estructura de ficheros reiniciamos jupyter notebook:

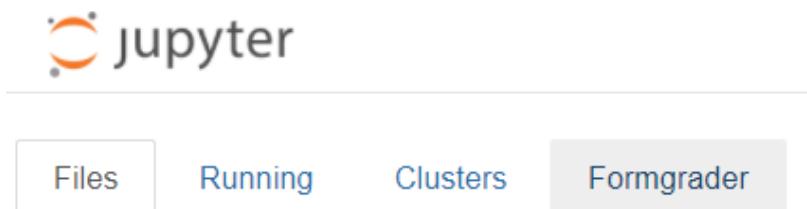


Figura D.5: Proceso de lanzamiento y creación 1

2. Al reiniciarlo nos habrá aparecido una nueva sección (Formgrader). En esta sección podremos gestionar la creación de tareas para los alumnos a través de la UI Formagrader:

A screenshot of the nbgrader 'Manage Assignments' interface. On the left, there is a sidebar with buttons for 'Manage Assignments' (highlighted in blue), 'Manual Grading', and 'Manage Students'. The main area is titled 'Manage Assignments' and contains a table with the following columns: Name, Due Date, Status, Edit, Generate, Preview, Release, Collect, # Submissions, Generate Feedback, and Release Feedback. A yellow box highlights a message: 'The exchange directory at / does not exist and could not be created. The "release" and "collect" functionality will not be available. Please see the documentation on Setting Up The Exchange for instructions.' At the bottom, there is a button labeled 'Add new assignment...'.

Figura D.6: Proceso de lanzamiento y creación 2

3. Para crear una nueva tarea, hacemos click en la opción “Add new assignment” y rellenamos los campos:

Add New Assignment

Name	Task1
Due date (optional)	dd/mm/aaaa --::--
Timezone as UTC offset (optional)	

Save **Cancel**

Figura D.7: Proceso de lanzamiento y creación 3

Name	Due Date	Status	Edit	Generate	Preview
task1	None	draft			
+ Add new assignment...					

Figura D.8: Proceso de lanzamiento y creación 4

4. Para editar la tarea hacemos click en el nombre:

jupyter

Files Running Clusters Formgrader

Select items to perform actions on them.

task1

Upload New

The notebook list is empty.

Figura D.9: Proceso de lanzamiento y creación 5

5. En esta sección creamos una nueva tarea (notebook) y editamos su nombre. Dentro del apartado de “view” (vista) podemos activar el modo de creación de tareas:

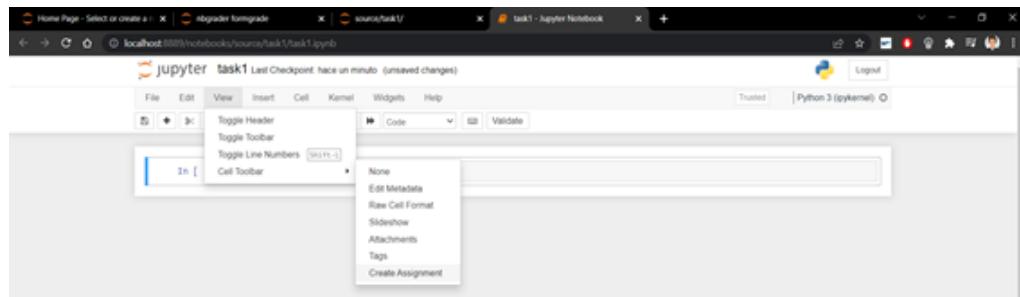


Figura D.10: Proceso de lanzamiento y creación 6

6. De esta manera las celdas tendrán diferentes formatos para la creación de la tarea:

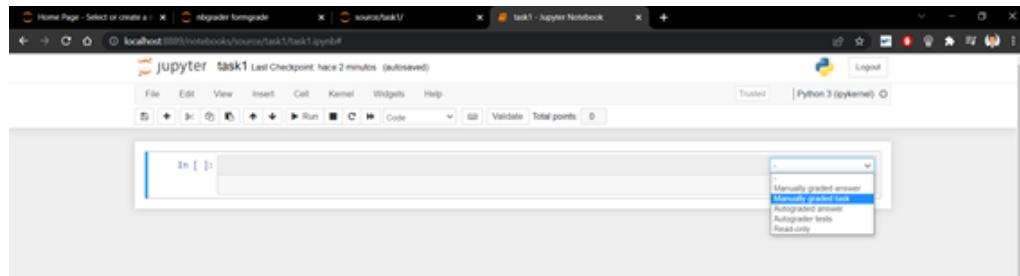


Figura D.11: Proceso de lanzamiento y creación 7

7. A continuación se puede crear nuestra primera prueba; se ha creado una prueba muy sencilla en la que se manda al alumno crear una función suma, para ello se crea una celda tipo “Autograded answer” en la que el profesor crea la función y deja entre los comentarios “###BEGIN SOLUTION” y “###END SOLUTION” la parte de código que el alumno ha de implementar. En la celda tipo “Autograded tests” se crean las pruebas para dicha función, estas pruebas están basadas en la ejecución de sentencias assert y se le ha de asignar una puntuación.

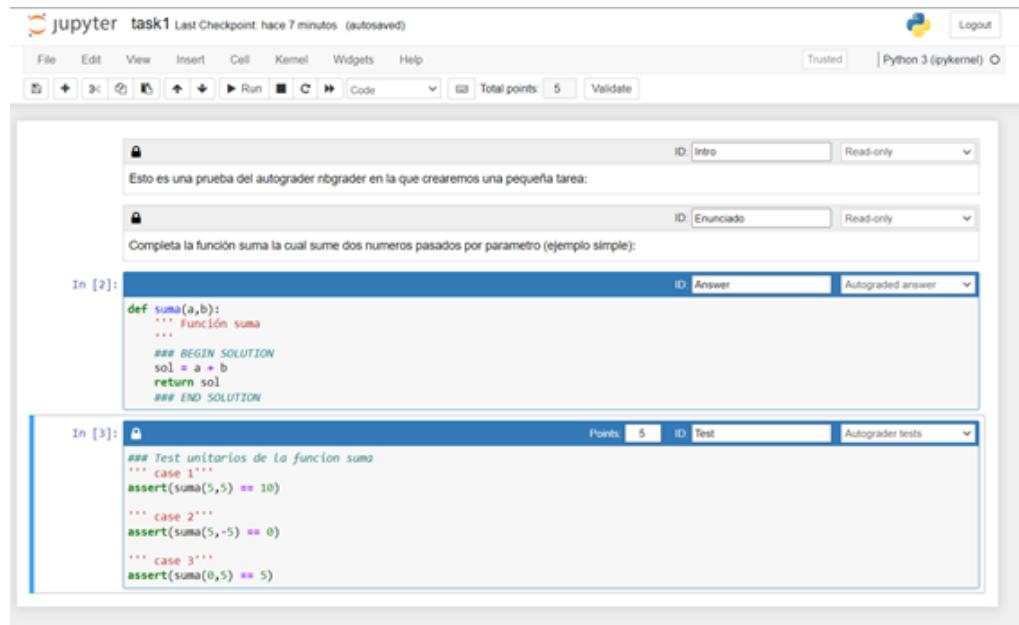


Figura D.12: Proceso de lanzamiento y creación 8

8. Una vez creadas las pruebas se ha de validar que el código del profesor es correcto y funciona por lo que se ha de hacer click en la casilla “validate” y en caso de que sea correcto recibiremos el siguiente mensaje:

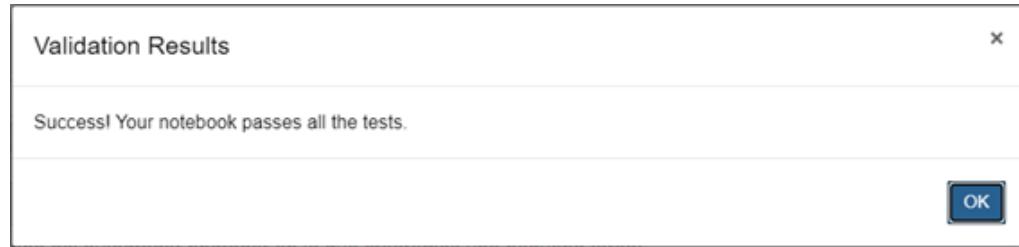


Figura D.13: Proceso de lanzamiento y creación 9

9. Una vez creado, volvemos a la sección de manejo de tareas:

Figura D.14: Proceso de lanzamiento y creación 10

10. Para generar la tarea que sería entregada a los alumnos hacemos click en el icono de “generate” para la tarea correspondiente y recibiremos, en caso de funcionar correctamente, el siguiente mensaje:

Figura D.15: Proceso de lanzamiento y creación 11

11. Una vez creado hacemos click en el apartado “preview” para previsualizarlo como un alumno lo vería:

```

jupyter task1 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Logout Not Trusted Python 3 (ipykernel) ○
Esto es una prueba del autograder nbgrader en la que crearemos una pequeña tarea:
Completa la función suma la cual sume dos numeros pasados por parametro (ejemplo simple):
In [ ]: def suma(a,b):
    """ Función suma
    ...
    # YOUR CODE HERE
    raise NotImplementedError()

In [ ]: ### Test unitarios de la función suma
    case 1
    assert(suma(5,5) == 10)
    case 2
    assert(suma(5,-5) == 0)
    case 3
    assert(suma(0,5) == 5)

```

Figura D.16: Proceso de lanzamiento y creación 12

12. Como vemos en la casilla de la función suma, el contenido del código se ha modificado automáticamente para ser completado por el alumno. Una vez hechos todos estos pasos ya tendríamos creada la tarea. En este momento nbgrader habrá generado automáticamente en el directorio raíz de tu equipo una nueva carpeta llamada “release” con la tarea correspondiente:

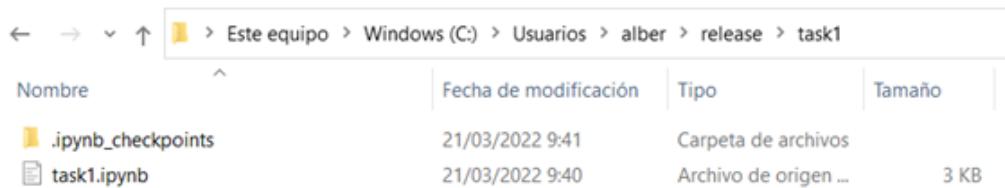


Figura D.17: Proceso de lanzamiento y creación 13

13. Este notebook será entregado al alumno para la realización de la tarea.

Realización de la tarea y corrección

1. Una vez el alumno recibe la tarea y la completa:

The screenshot shows a Jupyter Notebook interface with the title 'jupyter task1'. The notebook contains two code cells:

```
In [1]: def suma(a,b):
    """ Función suma
    """
    # YOUR CODE HERE
    return a + b
```

```
In [2]: """ Test unitarios de la función suma
    """
    case 1:
        assert(suma(5,5) == 10)
    """
    case 2"""
        assert(suma(5,-5) == 0)
    """
    case 3"""
        assert(suma(0,5) == 5)
```

The notebook header indicates it is 'Last Checkpoint hace unos segundos (autosaved)' and shows 'Trusted' status with 'Python 3 (ipykernel)' selected.

Figura D.18: Proceso de realización y corrección 1

2. Este puede comprobar su resultado haciendo click en el icono “validate” en caso de tener Nbgrader instalado:

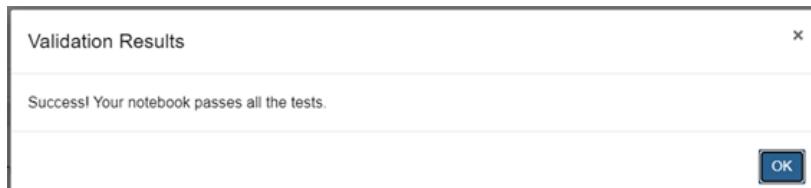


Figura D.19: Proceso de realización y corrección 2

3. Una vez haya acabado la tarea, la guardará y enviará de nuevo al profesor quien almacenará esta en el archivo dentro de submitted creado anteriormente para esa tarea:

A file explorer window showing the path: 'Este equipo > Windows (C:) > Usuarios > alber > submitted > alum1 > task1'. The table below lists the contents of the 'task1' folder:

Nombre	Fecha de modificación	Tipo	Tamaño
task1.ipynb	21/03/2022 9:53	Archivo de origen ...	3 KB

Figura D.20: Proceso de realización y corrección 3

4. Ahora el profesor accederá de nuevo al apartado de gestión de tareas:

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
task1	None	draft						1		
+ Add new assignment...										

Figura D.21: Proceso de realización y corrección 4

5. Y accederá al apartado de submissions de la tarea:

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
None, None	alum1	None					

Figura D.22: Proceso de realización y corrección 5

6. Hacemos click en la opción “autograde”:

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
None, None	alum1	None		5/5			

Figura D.23: Proceso de realización y corrección 6

7. Y como vemos se corrige automáticamente, en este caso el alumno ha aprobado. Adicionalmente si hacemos click en “generate feedback” nbgrader creará una carpeta “feedback” con documentos html con retroalimentación para el alumno:

			<< alberto > feedback > alum1 > task1		Buscar en task1
NOMBRE			Nombre	Fecha de modificación	Tipo
task1.html			task1.html	21/03/2022 10:57	Chrome HTML Do... 283 KB

Figura D.24: Proceso de realización y corrección 7

```

task1 (Score: 5.0 / 5.0)
  1. Test cell (Score: 5.0 / 5.0)

In [1]: Student's answer
def suma(a,b):
    """ Function suma
    ...
    return a+b

In [2]: Grade cell Test
Score: 5.0 / 5.0 (100%)
    
```

Figura D.25: Proceso de realización y corrección 8

Funcionamiento Nbgrader

Como se ha visto en la prueba anterior, Nbgrader necesita una serie de directorios y archivos para su funcionamiento. La estructura de un curso sobre el que va a funcionar Nbgrader es la siguiente:

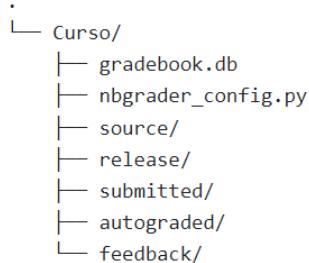


Figura D.26: Estructura Curso Nbgrader

- **gradebook.db:** Base de datos en la que Nbgrader almacena información sobre tareas, alumnos y entregas.
- **nbgrader_config.py:** Archivo de configuración de Nbgrader.

- **source:** Directorio en el que se almacenan las tareas (notebooks) creadas por el profesor. Estos notebooks serán adaptados por Nbgrader para la generación de estos en versión alumno.
- **release:** Directorio en el que se almacenan las tareas en versión del alumno tras ser generadas por Nbgrader.
- **submitted:** Directorio en el que se almacenan las tareas entregadas por los alumnos.
- **autograder:** Directorio en el que se almacenan las tareas corregidas.
- **feedback:** Directorio en el que se almacenan los archivos feedback generados por Nbgrader de las tareas enviadas y corregidas.

En la siguiente tabla se muestran las diversas etapas en las que consiste el funcionamiento de Nbgrader con la información sobre los actores que realizan estas acciones, comandos utilizados, estructura de carpetas, archivos implicados y observaciones sobre estas:

Etapa del proceso de evaluación	Paso	Agente al cargo		Orden nbgrader en línea de comandos	Manejo del paso			Estructura de carpetas y archivos			Comentarios
		Profesor	alumno		Consola	Jupyter	Api + programación (ad hoc)	Carpetas implicadas	Archivos implicados		
Desarrollo ejercicios	Crear contenedor del curso	x			x		x	Carpeta curso	curso_id (= .)		Crear manualmente carpeta en donde ejecutar por defecto jupyter notebook o los comandos manuales
	Configurar opciones de ejecución	x		nbgrader generate_config	x		x	Carpeta curso	curso_id (= .)	nbgrader_config.py	Opcional. Archivo de python con opciones de configuración (carpetas específicas, formatos, directivas de ejecución, etc.)
	Definición de tareas	x		nbgrader db	x	x	x	Subcarpeta de las fuentes de la tarea	./source/task_i	gradebook.db	La creación de las carpetas de tareas se puede hacer manualmente o con Jupyter
	Elaboración de ejercicios	x				x		Subcarpeta de la tarea con notebooks fuente	./source/task_i	notebooks_ij (versión del profesor)	Creación de los notebooks de ejercicios de cada tarea. Contienen soluciones el alumno debe elaborar por su cuenta y soluciones
	[Validación soluciones y tests]	x	x	nbgrader validate	x	x	x			notebooks_ij (versión del profesor)	Paso opcional. Sirve para comprobar que los notebooks del profesor (tests y soluciones propuestas) funcionan
	Elaborar versión alumno	x			x	x	x	Subcarpeta de la tarea con notebooks para alumnos	./release/task_i	notebooks_ij (versión para el alumno)	Para cada tarea generan los notebooks de ejercicios, versión del alumno, a partir de los de la versión del profesor
	[Vista previa de la versión del alumno]	x				x				notebooks_ij (versión para el alumno)	Permiten comprobar que lo que va a recibir el alumno es lo esperado
Resolución	Distribución a alumnos	x		nbgrader release_assignment	x	x	x	Carpetas de exportación	ruta_export/outbound/task1	notebooks_ij (versión para el alumno)	Poser la versión del alumno en la carpeta de distribución (o en una aplicación de distribución que se desarrolle al efecto)
	Descarga de tareas/ejercicios		x		x	x	x	Carpetas del curso en equipo del alumno y subcarpeta de tarea	mi_course/task_i	notebooks_ij (versión para el alumno)	El alumno se crea una carpeta del curso y se descarga los notebooks de ejercicios
	Trabajo personal		x	nbgrader validate	x	x	x			notebooks_ij con respuesta	El estudiante soluciona los ejercicios
	[Validación soluciones y tests]	x	x		x	x	x			notebooks_ij con respuesta	El estudiante puede comprobar que su resolución de ejercicio pasa los tests (los que permite el profesor, pues puede haber errores)
	Entrega de respuestas		x	nbgrader submit	x	x	x	Carpetas de importación de resultados	ruta_export/inbound/task_i	notebooks_ij con respuesta	El alumno sube o carga los notebooks resueltos de cada tarea
Evaluación	Recogida de respuestas	x		nbgrader collect	x	x	x	Subcarpeta de tareas a evaluar	/submitted/alumnoid_k/task_	notebooks_ij con respuesta archivo timestamp.txt (nota 1)	Las entregas se deben disponer en un esquema de archivos con el que nbgrader pueda trabajar
	Evaluación automática	x			x	x	x	Subcarpetas de tareas evaluadas por alumno	/autograded/alumnoid_k/task_i	notebooks_ij con respuesta con corrección automática (nota 1)	Autograding: Se hacen correr los tests automáticos
	Evaluación manual	x		nbgrader autograde	x	x	x	Subcarpetas de tareas evaluadas por alumno	/autograded/alumnoid_k/task_i	notebooks_ij con respuesta con corrección automática y manual (y comentarios)	Si el notebooks de ejercicios tiene celdas cuya evaluación debe hacerse manualmente por el profesor, este califica tales celdas e incluye sus comentarios para el feedback
Feedback	Generación de feedback	x		nbgrader generate_feedback	x	x	x	Subcarpeta de feedback con subcarpetas de la alumno y sus tareas evaluadas	/feedback/alumnoid_k/task_	notebooks_ij.html de feedback (nota 1)	Se generan archivos html con el feedback de la evaluación: resultados, solución propuesta por el profesor, comentarios, etc
	Entrega de feedback	x			x	x	x	Carpetas de exportación	ruta_export/feedback	notebooks_ij.html de feedback	Los archivos de feedback se sitúan en la carpeta de descarga para el alumno
	Recogida de feedback		x	nbgrader fetch_feedback	x	x	x	Carpetas de exportación	ruta_export/feedback	notebooks_ij.html de feedback	El alumno accede a los archivos de html de feedback

Figura D.27: Tabla Nbgrader 1

Etapa del proceso de evaluación	Paso	Agente al cargo		Orden nbgrader en línea de comandos	Manejo del paso			Estructura de carpetas y archivos			Comentarios
		Profesor	alumnado		Consola	Jupyter	Api + programación (ad hoc)	Carpetas implicadas	Archivos implicados		
otros pasos de Gestión y apoyo	Gestión de alumnos	x		nbgrader db	x	x	x	Carpetas curso	gradebook.db	Base de datos con tablas diversas: alumnos, tareas, datos de notebooks, datos de celdas, datos de entregas, comentarios, Exportación de las calificaciones a un archivo csv para su gestión	El profesor instala nbgrader y carga en jupyter las extensiones del profesor. Si el proceso se realiza en el entorno de red carpetas adecuado, el alumno se instala las extensiones para descarga y entrega de notebooks de ejercicios El alumno opcionalmente puea instalar una extensión para validar sus soluciones antes de entregarlas
	Recabar puntuaciones	x		nbgrader export	x		x	a discreción	datos en gradebook.db y archivo csv con calificaciones		
	Configuración de jupyter para nbgrader	x	x	nbgrader extension	x		x				

Figura D.28: Tabla Nbgrader 2

Creación de Cursos

Ya se ha visto la estructura de archivos y directorios que necesita Nbgrader para su funcionamiento. Debido a esta dependencia estructural, la forma en la que ha sido implementada la creación de cursos dentro del proyecto ha consistido en el copiado y renombrado del directorio de un curso base el cual contiene todos los archivos y directorios necesarios para el funcionamiento de Nbgrader y la creación de tareas. Este curso se encuentra dentro de la carpeta *courses* que es la propia carpeta en la que el resto de cursos serán añadidos:

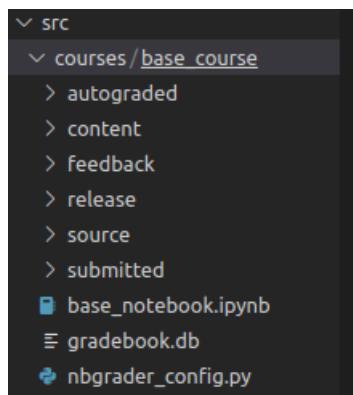


Figura D.29: Curso base

Adicionalmente, este curso base contiene el directorio *content* cuya función es el almacenamiento de los archivos teóricos pertenecientes al curso.

El endpoint o método-ruta en el que es realizada esta acción es el denominado **teacher_create_course()** y concretamente la sección de código en la que se realiza esta acción es la siguiente:

```
path = "courses/" + name
path_ = "courses/base_course"
source = os.path.join(os.path.abspath(os.path.dirname(__file__)), path)
destination = os.path.join(os.path.abspath(os.path.dirname(__file__)), path)
shutil.copytree(source, destination)
```

Figura D.30: Código creación curso

Manejo de Nbgrader

El manejo de Nbgrader dentro del proyecto se realiza mediante la clase **NbgraderManager** dentro del módulo **Management.py**.

Construcción

El constructor de esta clase es el siguiente:

```
# PRUEBA DE MANEJO DE LA API Y BASE DE DATOS DE NBGRADER
class NbgraderManager:
    """ Class responsible of handling the Nbgrader API and DB conexion usage.

    """

    def __init__(self, course_name):
        """ Initialization/constructor method.

        Parameters:
        - course_name: (string) courses name

        """

        root_chain = "src/courses/" + course_name
        db_chain = "sqlite:///src/courses/" + course_name + "/gradebook.db"

        config = Config()
        config.CourseDirectory.root = root_chain
        config.CourseDirectory.course_id = course_name

        self.api = NbGraderAPI(config = config)
        self.dbConnection = Gradebook(db_chain, course_name)
```

Figura D.31: Constructor manager

En este se está estableciendo la conexión con la base de datos *gradebook.db* de Nbgrader dentro de la carpeta del curso pasado por parámetro sobre el que se van a realizar las acciones de Nbgrader. Adicionalmente se crea la variable de instancia de la API de Nbgrader por la que se ejecutarán las funciones de este.

Acciones más importantes

Las operaciones más importantes del manager son las siguientes:

- **Grade:** Manda a la API de Nbgrader la realización de la corrección de una tarea accediendo a esta en la carpeta submitted del alumno pasado por parámetro y, en caso de tener éxito en la operación, accede

a la base de datos para obtener la calificación obtenida, redondeándola sobre 10:

```
def grade(self, task, student):
    """ Grades a task.

    Parameters:
        - task: (string) tasks's name
        - student: (string) student's username

    Returns:
        - False if the operation did not succeed
        - The calification the student recievied

    """
    result = self.api.autograde(task,student)
    if not result["success"]:
        return False
    submision = self.dbConnection.find_submission(task,student)
    # nota sobre 10
    grade = submision.score * 10 / submision.max_score
    return round(grade,2)
```

Figura D.32: Corrección de tarea

Este método es llamado desde el endpoint `student_course()` al realizarse la entrega de una tarea por parte de un alumno.

- **Create assigment:** Manda a la API de Nbgrader la generación de una tarea en versión de estudiantes. Para ello, accede a la tarea dentro de la carpeta source del curso:

```
def create_assigment(self, task):
    """ Creates an assigment for the students.

    Parameters:
        - task: (string) tasks's name

    Returns:
        - True
        - False

    """
    result = self.api.generate_assignment(task)
    if not result["success"]:
        return False
    else:
        return True
```

Figura D.33: Generación de tarea para los alumnos

Este método es llamado cuando un profesor crea una sección de contenido de forma directa en la que entrega el notebook tarea creado previa-

mente, acción realizada en el endpoint **teacher_create_section(course)**. O cuando el profesor ordena la publicación de una sección de contenido no publicada, acción realizada en el endpoint **publish_section(course,section)**.

- **Generate feedback:** Manda a la API de Nbgrader la generación del documento feedback de una tarea entregada por un alumno:

```
def generate_feedback(self, task, student):
    """ Generates student's submission feedback file.

    Parameters:
        - task: (string) tasks's name
        - student: (string) student's username

    """
    self.api.generate_feedback(task, student)
```

Figura D.34: Generación de feedback

Este método es llamado tras la correcta corrección automática de una tarea al ser entregada por un alumno en el endpoint **student_course()**.

Creación de tareas desde la plataforma

Ya ha sido mencionado con anterioridad el hecho de que una de las funcionalidades de la plataforma consistía en permitir a los profesores la creación de tareas desde la propia plataforma. Para lograr esto fueron añadidas las denominadas **Secciones no publicadas**. Estas secciones son creadas por el profesor dentro de uno de sus cursos y son solo visualizables por él y no por los alumnos, hasta su publicación o descarte. Al crearse una sección no publicada por un profesor, este define el nombre que tendrá el notebook tarea la cual es incluida en la carpeta *source* del curso. Posteriormente, el profesor puede ver el listado de las secciones no publicadas del curso y acceder a la tarea que deseé editar, momento en el que se le abrirá en su navegador una nueva pestaña en la que se está accediendo, mediante la ejecución remota de Jupyter Notebook, al notebook específico. Para lograr esto, Jupyter Notebook junto con Nbgrader se encuentra instalado dentro del propio proyecto y es ejecutado remóticamente desde el momento inicial en el que la app es iniciada. Los comandos para lograr esto son los siguientes:

```
# KEEP JUPYTER NOTEBOOK RUNNING
# kill any jupyter current instance
os.system("pkill -f -1 jupyter*")
# run jupyter
os.system("jupyter notebook --ip='0.0.0.0' --no-browser --allow-root --port=8888 --notebook-dir=src/courses &")
```

Figura D.35: Ejecución de Jupyter

Para lograr la apertura del notebook tarea concreto que el profesor desea editar, se incluye la siguiente línea de código dentro del template de vistas de secciones sin publicar **unreleased_sections.html**:

```
<a class="btn btn-outline-primary" target="_blank"
| href="http://...:8888/notebooks/{{ course.name }}/source/{{ section.task_name }}/{{ section.task_name }}.ipynb">
```

Figura D.36: Apertura Notebook

Esta se encarga de abrir en una nueva pestaña el notebook específico, introduciendo la URL de este. Para ello se ha tenido que introducir la IP específica de la ruta de ejecución remota de nuestro Jupyter Notebook (la cual ha sido ocultada en esta imagen por seguridad), por ello, se debía conocer con anterioridad la IP sobre la que nuestra plataforma ha sido desplegada.

Cuando un profesor decide publicar una sección no publicada, la orden de generación de tarea es llamada, cogiendo el archivo correspondiente a esa sección en el fichero *source* y generando dentro de *release* la versión correspondiente a los alumnos.

Librerías y métodos relevantes utilizados

En esta sección van a ser descritas algunas de las librerías y métodos más relevantes utilizados en el proyecto junto con su utilidad dentro del mismo:

- **os:** El módulo de Python os permite el uso de funcionalidades de sistema operativo de forma versatil. En nuestro proyecto, los métodos y módulos derivados de este utilizados han sido:
 - **system:** Ejecución de comandos de sistema operativo. Ha sido utilizado para el lanzamiento de Jupyter Notebook desde el proyecto.

- **remove:** Método de borrado de ficheros dentro de un directorio. Ha sido utilizado para el borrado de envíos, borrado de archivos de contenido teórico y tareas dentro de los directorios *source* de los cursos en la cancelación de secciones, borrado de archivos de contenido teórico en la eliminación de secciones y borrado de tareas en la entrega de las mismas por los alumnos en caso de no haberse podido realizar el autograding por problemas en ella.
- **path:** Modulo de manejo de rutas de archivos y directorios.
- **mkdir:** Método de creación de directorios. Ha sido utilizado para la creación de directorios fundamentales para el funcionamiento de Nbgrader.
- **rename:** Método de renombrado de archivos. Ha sido utilizado para el renombre de archivos en la creación de tareas de secciones no publicadas, tras realizarse la copia del notebook base encontrado en cada curso.
- **shutil:** El módulo de Python *shutil* permite realizar Operaciones de archivos de alto nivel. En nuestro proyecto, los métodos y módulos derivados de este utilizados han sido:
 - **rmtree:** Método de borrado de rutas de directorios. Ha sido utilizado para el borrado de carpetas de alumnos dentro de los directorios *feedback* y *submitted* en el borrado de alumnos o extracción de los mismos de un curso, borrado de directorios que habían sido creados con anterioridad en la creación de secciones que han dado error de generación de tarea, cancelación de secciones no publicadas, borrado de cursos, eliminación de secciones y actividades similares.
 - **copytree:** Método de copiado de árboles de archivos. Ha sido utilizado en la creación de nuevos cursos.
 - **copy:** Método de copiado de archivos. Ha sido utilizado en la creación de nuevas tareas.
- **flask.send_from_directory():** Método de Flask para el envío de ficheros de un directorio. Utilizado para la descarga de archivos por parte de los usuarios.

Montaje en Docker

El Dockerfile a partir del que se ha generado la imagen del contenedor en el que ha sido instalada nuestra plataforma es el siguiente:

```

FROM ubuntu:20.04

# desabilitar prompt en la instalacion de paquetes
ARG DEBIAN_FRONTEND=noninteractive

# instalar pip3
RUN apt update && apt install -y python3-pip python3-dev

WORKDIR /project

COPY . /project

# extensiones necesarias para el funcionamiento de Nbgrader
RUN pip3 --no-cache-dir install -r requirements.txt
RUN jupyter nbextension install --sys-prefix --py nbgrader --overwrite
RUN jupyter nbextension enable --sys-prefix --py nbgrader
RUN jupyter serverextension enable --sys-prefix --py nbgrader
COPY jupyter_notebook_config.py /root/.jupyter/jupyter_notebook_config.py

```

Figura D.37: Dockerfile

En este se está ordenando la copia del proyecto junto con la instalación de todas sus dependencias software para su correcto funcionamiento. Adicionalmente, es introducida en la ruta de configuración de Jupyter Notebook el fichero **jupyter_notebook_config.py** el cual ha sido modificado deshabilitando la apertura de terminales y el botón de parada del kernel de ejecución para, en caso de que un profesor intente pararlo, este no pueda ya que esto generaría un problema en la edición de tareas desde la plataforma.

Una vez creada la imagen del contenedor a partir del dockerfile mediante la ejecución del comando **docker build -t "nombre_imagen"**. Este ha de ser ejecutado realizando el mapeo de puertos adecuado. Nuestra aplicación flask es lanzada a través del puerto local 5000 y Jupyter Notebook es ejecutado a través del 8888, por ello, el comando de construcción del contenedor con el mapeo de puertos correcto sería: **docker run -d -t -name=*nombre* -p *puerto de salida deseado para flask*:5000 -p *puerto de salida deseado para Jupyter*:8888 *nombre_imagen***. Una vez esté corriendo el contenedor con el mapeo de puertos correcto, se ha de acceder a este mediante el comando: **docker exec -ti *nombre* bash**.

Al ser ejecutado Jupyter de forma remota y accedido en un navegador web por primera vez, Jupyter pide al usuario que introduzca un token que este genera de forma automática, o una contraseña previamente establecida como forma de seguridad ante el intento de acceso a usuarios no autorizados. Puesto que los profesores que accedan de forma remota a Jupyter no tendrán acceso a este token generado automáticamente dentro del contenedor Docker,

se ha de establecer una contraseña previa al lanzamiento de la aplicación. Para ello, una vez instalado el proyecto dentro del correspondiente contenedor Docker, se ha de establecer una contraseña mediante el comando **jupyter notebook password**. En el caso de nuestra plataforma, la contraseña que los profesores tendrán que introducir la primera vez que accedan a jupyter al crear o editar una tarea de forma remota será la palabra "teacher".

Una vez realizado el paso previo nuestra aplicación estaría lista para su ejecución desde el contenedor, acción que es realizada mediante la ejecución del comando **python3 app.py**. Tras esto nuestra aplicación podrá ser accesible de forma local en un navegador en la dirección **http://localhost:5000**.

Riesgo de la ejecución remota de Jupyter Notebook

El hecho de estar ejecutando jupyter notebook de forma remota con el fin de hacer posible la edición de tareas supone un inconveniente de seguridad. Este inconveniente consiste en que en el momento en el que un profesor está editando una tarea, este puede retroceder en el árbol de archivos de jupyter notebook, viendo así carpetas correspondientes al funcionamiento interno de los cursos:

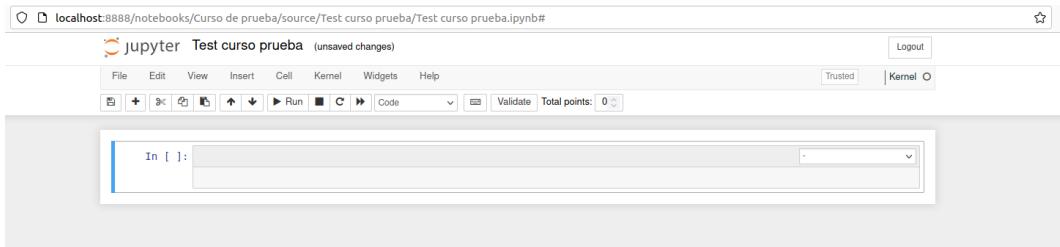


Figura D.38: Edición de tarea

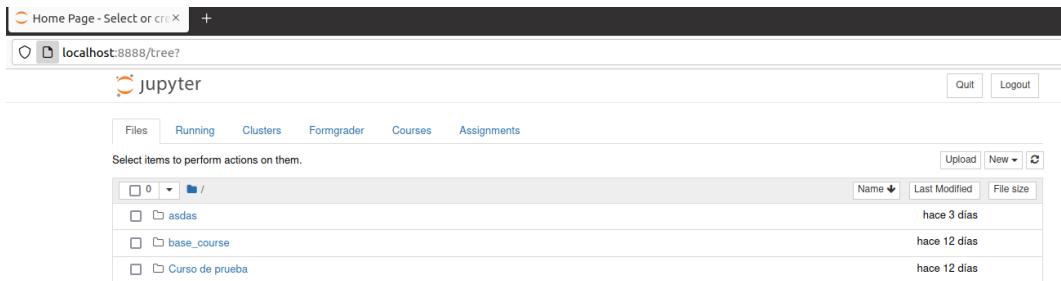


Figura D.39: Acceso carpetas de cursos

Como se puede ver en las imágenes anteriores, el profesor ha pasado de editar una tarea a ver las carpetas de los cursos de la página retrocediendo en la ruta de archivos de Jupyter. Este incoveniente de seguridad no es especialmente preocupante al ser los propios profesores quienes pueden acceder a estos archivos, pero es la principal razón por la que no se permite a los alumnos contestar a las tareas desde la propia web, sino que han de resoverlas de forma local habiéndose descargado la tarea. Si un alumno pudiera realizar esta acción, tendría acceso a todos los contenidos de los cursos incluidas las versiones de las tareas del profesor, lo que podría dar lugar a la realización de trampas para superar las tareas o incluso problemas de funcionamiento de la plataforma en el caso de que estos decidieran eliminar algún directorio o archivo. La resolución de este problema ha sido propuesta como una posible futura línea de trabajo en el apartado de conclusiones y futuras líneas de trabajo de la memoria.

D.4. Instalación, ejecución y despliegue del proyecto

Instalación

Puesto que se ha hecho uso de la herramienta VirtualEnv la instalación de nuestro proyecto **en un ambiente de desarrollo Ubuntu / Linux** es una tarea muy sencilla. Para ello simplemente se ha de clonar el repositorio del mismo el cual tiene incluida la carpeta env correspondiente a esta herramienta.

Ejecución desde el entorno de desarrollo

Para la ejecución de nuestra aplicación de forma local en el ambiente de desarrollo se ha de activar el ambiente virtual mediante el comando **source env/bin/activate** desde el directorio raíz del proyecto. Una vez hecho esto, nuestra aplicación puede ser lanzada mediante el comando **python3 app.py**.

Adicionalmente, puesto que el estado final en el que se encuentra el proyecto dentro del repositorio está pensado para el despliegue directo de la plataforma, se puede activar dentro del archivo de lanzamiento **app.py** el modo de depuración, asignando el valor "True" al parámetro "Debug" del método de ejecución Flask de la aplicación:

```
app.py
1 """ Flask Project app.py module
2 """
3 from src import app
4
5 if __name__ == "__main__":
6     app.run(host="0.0.0.0", port="5000", debug= True)
7
```

Figura D.40: Modo de depuración

Despliegue

La plataforma ha sido desplegada en una instancia EC2 de Amazon Web Service. En este apartado se va a explicar el proceso seguido para lograrlo en caso de que futuros programadores quieren realizar el mismo tipo de despliegue:

- Creación de la instancia t2.micro con imagen Amazon Linux gratuita de arquitectura 64 bits, par de claves RSA y formato de archivo pem y permitiendo el tráfico HTTP desde internet.
- Acceso a la instancia como cliente SSH.
Comando: **ssh -i *Tu_Archivo_Clave.pem* ec2-user@ec2-3-89-140-10.compute-1.amazonaws.com**
- Instalación de Docker y Git dentro de la instancia. Comandos: **sudo amazon-linux-extras install docker** y **sudo yum install Git**
- Clonación del repositorio. Comando **Git clone *url***

- Asignación de permiso al usuario ec-2. Comando **sudo usermod -a -G docker ec2-user**.
- Salida y vuelta a entrar a la instancia para la correcta asignación de permisos.
- Creación de la imagen mediante docker build.
- Lanzamiento del contenedor dentro de la instancia con el mapeo correcto de puertos. El puerto de salida de la plataforma Flask sobre el que se accederá a esta queda a elección del programador, en nuestro caso ha sido seleccionado el puerto 80. El puerto de salida de Jupyter Notebook ha de ser el 8888 por obligación ya que es hacia el que la plataforma redirecciona al abrir las tareas.
- Acceso al contenedor, establecimiento de la contraseña de Jupyter Notebook y lanzamiento de la aplicación como ha sido descrito en el apartado de Montaje en Docker del Manual del programador.
- Adición de las dos reglas de entrada correspondientes a los dos puertos de nuestra plataforma dentro del grupo de seguridad de la instancia EC2. Tipo de reglas TCP personalizado y puertos 8888 (Jupyter Notebook) y el asignado a la salida de la aplicación Flask.
- Salida de la instancia.

Una vez realizados los siguientes pasos nuestra plataforma será accesible en la url: **http://*IP_INSTANCIA*:80**.

Apéndice E

Documentación de usuario

E.1. Introducción

Esta apendice contiene la información orientada a los usuarios, tanto profesores como alumnos, para el correcto acceso y manejo de la plataforma.

E.2. Requisitos de usuarios

Requisitos de profesores

Para poder dar un uso adecuado a la aplicación se recomienda a los profesores la instalación de **Jupyter Notebooks** y **Nbgrader** en su entorno de trabajo personal para la creación de tareas. Estos pueden prescindir de ello ya que la plataforma permite edición de tareas desde esta pero, en el caso de que requieran visualizar las entregas de los alumnos descargadas, necesitaran de al menos **Jupyter Notebooks** para hacerlo.

En cuanto al acceso a la plataforma no se requiere de ningún requisito previo debido a la naturaleza web de esta. Estos podrán acceder a ella en el siguiente enlace <http://3.89.140.10/>.

Requisitos de alumnos

En lo que a los alumnos respecta, estos deberán tener instalado de forma obligatoria **Jupyter Notebooks** en su entorno de trabajo para la resolución de tareas.

Como se ha mencionado previamente, no se requiere de ningún requisito previo en cuanto al acceso a la plataforma web. Estos podrán acceder a ella en el siguiente enlace <http://3.89.140.10/>.

E.3. Instalación

Al tratarse de una plataforma web no es necesaria ninguna instalación previa para el acceso a esta. Alumnos y profesores podrán acceder a la plataforma en el siguiente enlace: <http://3.89.140.10/>.

E.4. Manual del profesor

A continuación se explicará el manejo de la plataforma orientado a los profesores:

1 Acceso a la Plataforma

El acceso a la plataforma es realizado a través del siguiente enlace: <http://3.89.140.10/> el cual nos lleva a la página inicial de esta.

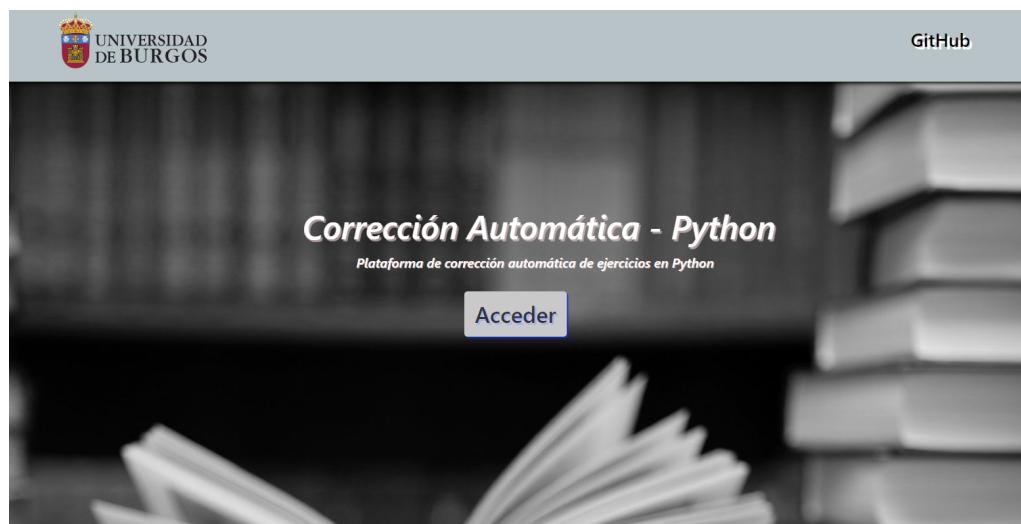


Figura E.1: Página Inicial

2 Inicio de Sesión

El proceso de inicio de sesión contiene los siguientes paso:

- Una vez accedido a la Página incial hacer click en el botón **Acceder** lo cual nos llevará a la página de inicio de sesión:



Figura E.2: Pagina Inicio de sesión

- Introducir los credenciales de inicio de sesión y hacer click en el botón **Login**. Han sido incluido en el fichero README del repositorio los credenciales de acceso de los 5 profesores definidos incialmente para la aplicación.
- Si los credenciales son correctos será llevado a la página principal del profesor:



Figura E.3: Página Principal Profesor

3 Gestión de Cursos

Para ir a la página de gestión de cursos del profesor se ha de hacer click en el botón **Ir a mis Cursos** de la **Página Principal del Profesor**(E.3) o en el botón **Cursos** de la barra navegacional superior. Esto nos llevaría a la página de gestión de cursos en la que se encuentran todos los cursos creados por este profesor:

The screenshot shows a web application interface for managing courses. At the top, there is a menu bar with links to 'Inicio', 'Cursos', 'Calificaciones', and 'Estudiantes'. Below the menu, the title 'Sus Cursos:' is displayed. A table lists two courses:

Curso	Descripción			
Curso 1	Soy el curso número 1	Acceder	Gestion de estudiantes	Eliminar
Curso 2	Soy el curso número 2	Acceder	Gestion de estudiantes	Eliminar

Below the table is a button labeled 'Crear curso'. At the bottom of the page, there is a footer section with links to 'UNIVERSIDAD DE BURGOS', 'REPOSITORIO' (GitHub and Nbgrader), 'ENLACES', and 'CONTACTO' (email: apf1002@alu.ubu.es). The footer also includes a copyright notice: '© 2022 Copyright'.

Figura E.4: Página de Cursos Profesor

3.1 Creación de cursos

Para la creación de un nuevo curso:

- Hacer click en el botón **Crear curso** de la **Página de Cursos del Profesor**(E.4) lo cual le llevará al formulario de creación de un nuevo curso:

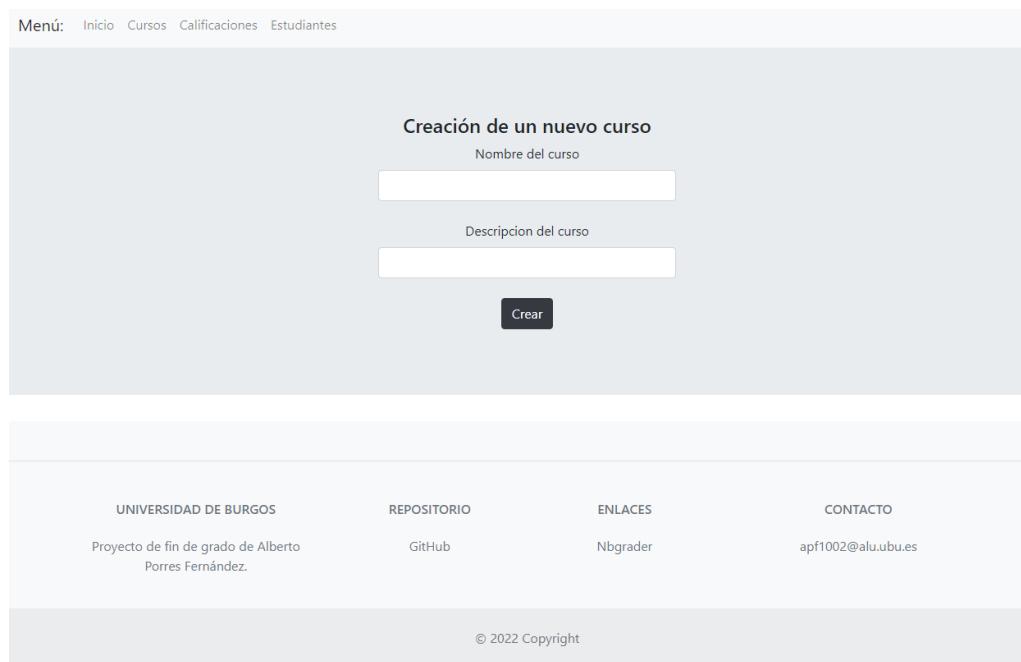


Figura E.5: Página de Creación de Curso

- Introducción de los datos (nombre y descripción) del curso a crear.
- Hacer click en el botón **Crear**.
- Hacer click en el botón **Confirmar** del modal de confirmación desplegado.
- Si los datos no pertenecen a otro curso su curso será creado y visualizable desde la **E.4 Página de Cursos del Profesor**.

3.2 Borrado de cursos

Para el borrado de cursos se ha de hacer click en el botón **Eliminar** del curso deseado desde la **Página de Cursos**(E.4), esto desplegará un modal de confirmación de acción que borrará el curso al hacer click en **Confirmar**.

3.3 Gestión de alumnos de un curso

El proceso de gestión de alumnos de un curso, que permite el ingreso o expulsión de alumnos a un curso, se realiza de la siguiente manera:

- Hacer click en el botón **Gestión de estudiantes** del curso deseado desde la **Página de Cursos(E.4)**. Esto nos llevará a la página de gestión de alumnos del curso:

The screenshot shows two main sections: 'Alumnos dentro del curso Curso 1:' and 'Alumnos fuera del curso Curso 1:'. Both sections have columns for 'Nombre' and 'Nombre de Usuario'. In the first section, there are two rows: one with 'Alumno 2' and 'Alumno 2' with a red 'Sacar del curso' button, and another with 'Alumno 3' and 'Alumno 3' with a red 'Sacar del curso' button. In the second section, there is one row with 'Alumno1' and 'Alumno1' with a green 'Ingresar en el curso' button.

Alumnos dentro del curso Curso 1 :		Alumnos fuera del curso Curso 1 :	
Nombre	Nombre de Usuario	Nombre	Nombre de Usuario
Alumno 2	Alumno 2	Alumno1	Alumno1
Alumno 3	Alumno 3		

UNIVERSIDAD DE BURGOS

Proyecto de fin de grado de Alberto
Porres Fernández.

REPOSITORIO

GitHub

ENLACES

Nbgrader

CONTACTO

apf1002@alu.ubu.es

© 2022 Copyright

Figura E.6: Página de Gestión de Alumnos de un Curso

- Mediante los botones de **Sacar del curso** y **Ingresar en el curso** podrá expulsar alumnos matriculados o ingresar alumnos no matriculados al curso. La confirmación de estas acciones se hace haciendo click en el botón **Confirmar** del modal desplegado al hacer click en los botones anteriores.

3.4 Acceso y gestión de contenidos de un curso

Para ir a la página de un curso en particular, desde la que se puede realizar la gestión de su contenido, se ha de hacer click en el botón **Acceder** del curso en la **Página de Cursos del Profesor(E.4)**. La página/vista de un curso particular es la siguiente:

Menú: Inicio Cursos Calificaciones Estudiantes

Curso 1

Sección 1

Descargar contenido teórico Descargar tarea versión profesor Descargar tarea versión estudiante

Eliminar

Sección 2

Descargar contenido teórico Descargar tarea versión profesor Descargar tarea versión estudiante

Eliminar

Crear sección Crear sección no publicada Ir a las secciones no publicadas

Figura E.7: Página de Gestión de Contenidos de un curso

Los cursos se organizan en **secciones de contenido** cada una de las cuales está **constituida por un documento teórico** cuyo formato queda a elección del profesor (PDF, DOCX, IPYNB, ...) y **una tarea corregible automáticamente en su entrega** en formato IPYNB creada mediante Jupyter Notebook y Nbgrader por el profesor.

Desde esta vista podrá:

- Descargar contenidos teóricos, tareas en versión profesor y tareas en versión alumno a través de los botones **Descargar contenido teórico**, **Descargar tarea versión profesor** y **Descargar tarea versión alumnos** de cada sección de contenido actual del curso.
- Eliminar secciones actuales del curso mediante el botón **Eliminar** de cada una de ellas.
- Ir a la creación de sección de forma directa, creación de sección no publicada o gestión de las mismas (estos procesos serán descritos a continuación).

3.4.1 Creación de sección de forma directa

Para crear una sección de contenido de forma directa se ha de hacer click en el botón **Crear sección** de la **Página de gestión de contenidos de un curso**(E.7) en particular, lo que nos lleva al formulario de creación de sección:

Menú: Inicio Cursos Calificaciones Estudiantes

Creación de sección de contenido

Nombre:

Nombre de la tarea:

Archivo de contenido teórico:
[Seleccionar archivo] Ninguno archivo selec.

Archivo tarea:
[Seleccionar archivo] Ninguno archivo selec.

El archivo tarea será renombrado al nombre de la tarea

Crear sección

Figura E.8: Página de creación de sección

En este formulario se han de introducir el nombre de la sección, el **archivo de contenido teórico**, el **archivo tarea creado previamente mediante Jupyter Notebook y Nbgrader** y el **nombre que va a ser asignado a ese archivo tarea**. Debido al funcionamiento interno del sistema, todos los campos deben de ser únicos y no estar presentes en alguna otra sección de algún curso, de ser así la creación de la sección no será posible.

Para crear la sección se ha de hacer click en el botón **Crear sección** que será visualizable desde la **Página de Gestión de Contenidos del Curso**(E.7).

3.4.2 Creación y gestión de secciones no publicadas

Dentro de un curso también puede haber incluidas **secciones no publicadas** las cuales no son visualizables por los alumnos. La función de estas consiste en permitir a profesores la **edición de las tareas desde la propia plataforma mediante Jupyter Notebook y Nbgrader**. Haciendo click en el botón **Ir a las secciones no publicadas** de la **Página de Gestión de Contenidos del curso**(E.7) irá a la página en la que se encuentran estas secciones:

The screenshot shows a web interface for managing unpublished sections. At the top, there's a menu bar with 'Menú: Inicio Cursos Calificaciones Estudiantes'. Below it is a title 'Secciones sin publicar' with a help link 'Ayuda creación de tareas'. A table lists a single section: 'Sección de prueba'. To its right are four buttons: 'Publicar' (green), 'Editar' (blue), 'Descargar estado actual de la tarea' (blue), and 'Eliminar' (red). At the bottom of the main content area is a blue button 'Crear sección con tarea creada manualmente'. The footer of the page includes links for 'UNIVERSIDAD DE BURGOS', 'REPOSITORIO', 'ENLACES', and 'CONTACTO', along with specific contact details like 'Proyecto de fin de grado de Alberto Porres Fernández.', 'GitHub', 'Nbgrader', and 'apf1002@alu.ubu.es'. A copyright notice '© 2022 Copyright' is also present.

Figura E.9: Página de secciones no publicadas

En esta vista podrá **publicar secciones, descargar el estado actual de las tareas, eliminar secciones, crear nuevas secciones o editar las tareas desde la propia plataforma mediante el uso de los botones correspondientes visualizables en la imagen anterior**.

Al hacer click en el botón **Editar** se le abrirá en una nueva pestaña el Notebook de Jupyter correspondiente a la tarea de la sección, permitiendo su edición. En caso de ser la primera vez que acceda desde un navegador a la edición de tareas le será **requerida una contraseña**. Esta contraseña es la palabra **"teacher"**:

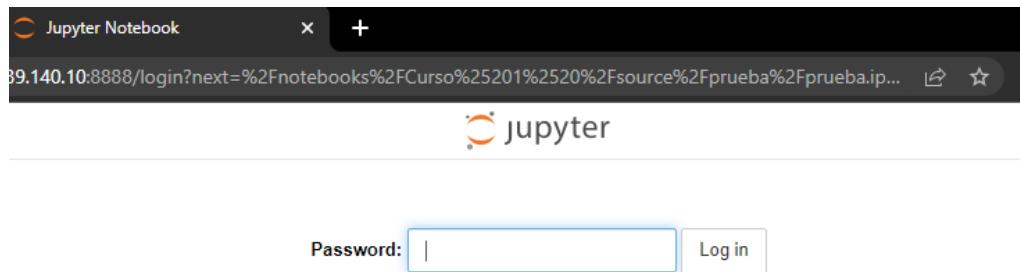


Figura E.10: Requerimiento de contraseña

Una vez introducida podrá visualizar y editar la tarea correspondiente:

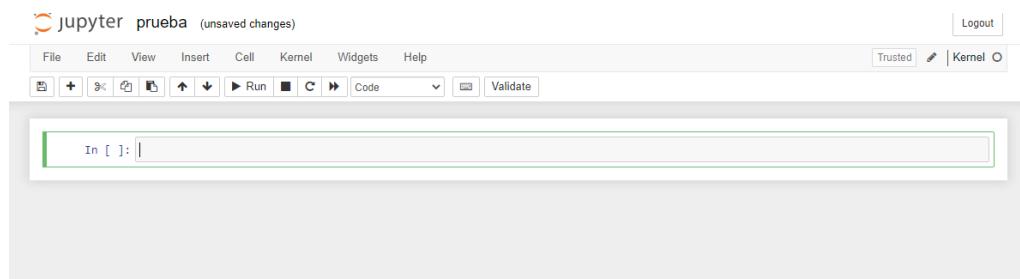


Figura E.11: Edición de tarea desde la plataforma

Para ir a la creación de secciones no publicadas se puede acceder haciendo click en el botón **Crear sección no publicada** de la **Página de Gestión de Contenidos de un curso**(E.7) o haciendo click en el botón **Crear sección con tarea creada manualmente** de la **Página de secciones no publicadas**(E.9). Estas acciones le llevarán al formulario de creación de secciones no publicadas:

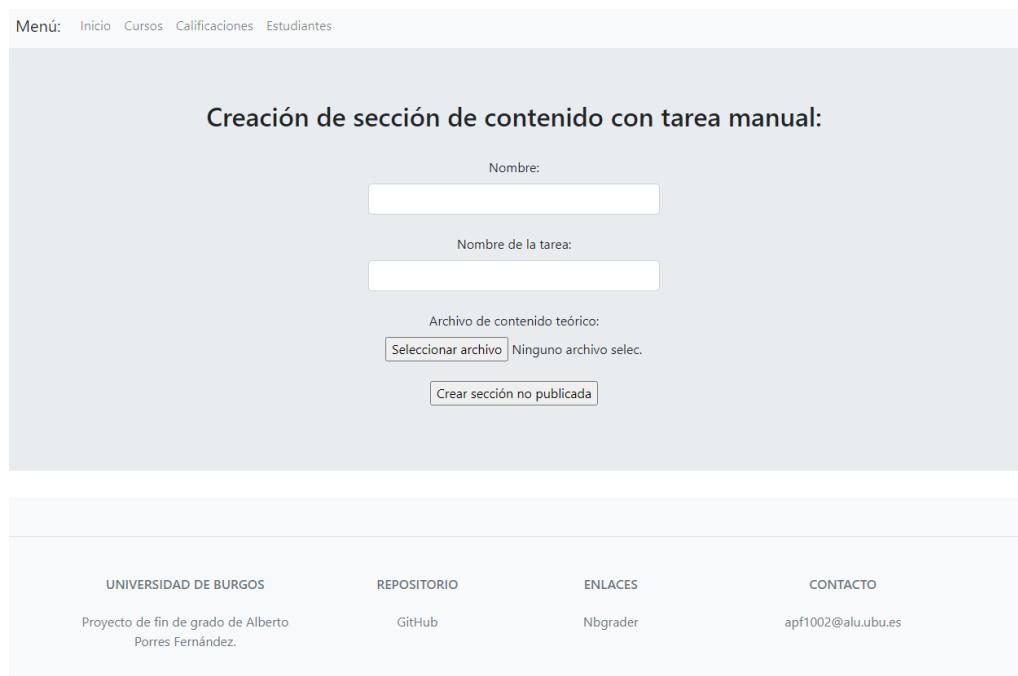


Figura E.12: Página de creación de tareas no publicadas

De la misma manera que en la creación de secciones de forma directa, todos los campos y contenidos han de ser únicos. En esta vista deberá escoger **nombre de la sección**, **nombre el cual recibirá la tarea** y entregar el **archivo de contenido teórico contendrá la sección**. La creación se realiza haciendo click en el botón **Crear sección no publicada** la cual será visualizable desde la **Página de secciones no publicadas**(E.9).

4 Crear tareas con Nbgrader

La herramienta de la que se está haciendo uso para la creación y corrección automática de tareas dentro de la plataforma es Nbgrader. Dentro de la página de secciones no publicadas se puede acceder a la guía de esta herramienta para la creación de tareas a través del botón **Ayuda creación de tareas** pero debido a que en esta guía se cubre funcionalidades que no utilizamos, a continuación se explicará todo lo que un profesor necesita saber para la creación de tareas:

Una vez abierto el Notebook correspondiente a la tarea, ya sea desde la opción de edición proporcionada por la plataforma o de forma local

teniendo instalado Jupyter Notebook y Nbgrader, se ha de activar la vista de edición de tareas en **View/Cell Toolbar/Create Assignment** en la barra navegacional del Notebook:

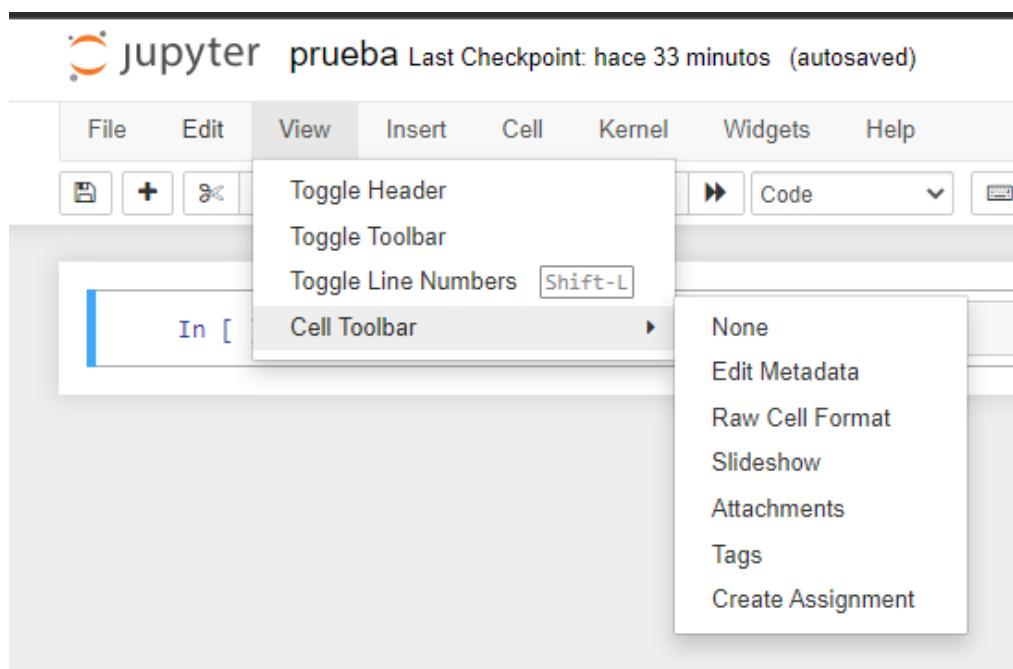


Figura E.13: Activar vista de edición

Lo que le permitirá asignar identificadores de celda a cada una de las celdas del notebook:



Figura E.14: Tipos de celda

Los tipos de celda que nos interesan son:

- **Read-Only** Solo lectura. Estas celdas no permitirán su edición cuando sea generada la versión del alumno de la tarea.
- **Autograded Answer** Celdas de Respuesta. En estas celdas los alumnos tendrán que implementar su código de respuesta a los ejercicios contenidos en la tarea.
- **Autograded Test** Celdas de Test. Estas celdas contienen los test que los alumnos han de superar. A estas celdas es asignada una puntuación que los alumnos obtendrán (para ese ejercicio en particular) en caso de superar esos tests.

Vamos a ver un ejemplo de como sería una tarea sencilla con un ejercicio a resolver y a continuación explicaremos lo que se ha hecho para comprender mejor el proceso de creación de tareas. Este ejercicio consiste en la implementación de una función que suma dos números pasados por parámetro:

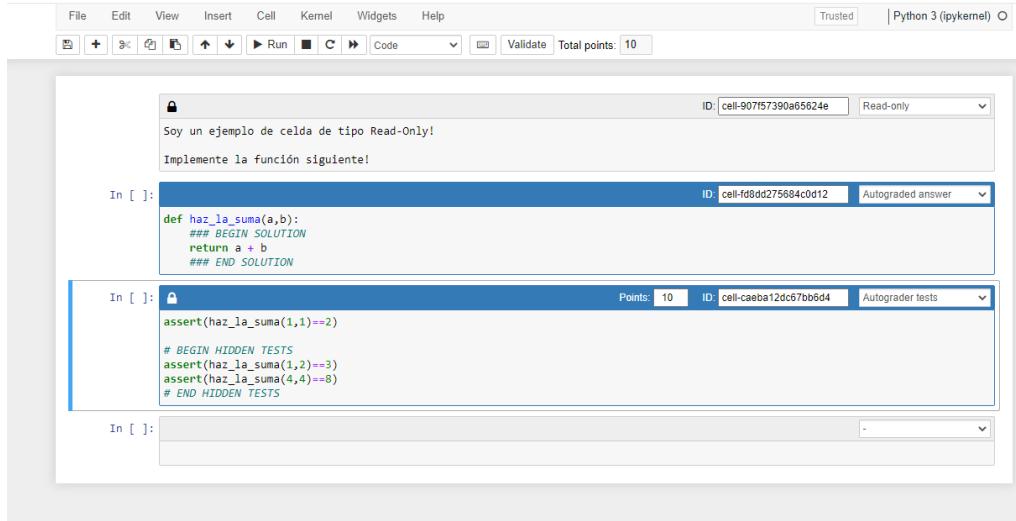


Figura E.15: Ejemplo Tarea

En la segunda celda se encuentra la función que el alumno ha de completar, Nbgrader tiene la opción de definir secciones dentro de las celdas de tipo Autograded Answer entre los comentarios **### BEGIN SOLUTION** y **### END SOLUTION** en las que el profesor puede implementar el código solución. Estas secciones son ocultadas en la versión de la tarea para el alumno pero son de utilidad en el desarrollo de las tareas. De forma similar, en las celdas de tipo Autograded Tests, como la tercera celda, se pueden definir secciones de test entre los comentarios **### BEGIN HIDDEN TESTS** y **### END HIDDEN TESTS** los cuales serán ocultados a los alumnos. Esto es de gran utilidad ya que la posibilidad de visualizar los tests pueden dar pistas sobre la resolución de los problemas a los alumnos. Se tendrá cuidado al definir estos tipos de secciones ya que un fallo en la ortografía de los comentarios provocará que la tarea **no se genere**.

Como podemos ver en la imagen superior, a la tercera celda le ha sido asignada una puntuación como se ha descrito anteriormente. Nbgrader no redondea la puntuación total de todos los ejercicios de una tarea, sin embargo, se ha adaptado la plataforma para que la calificación obtenida cuando un alumno envía una tarea sea **sobre 10** pese a que la suma total de puntuaciones de una tarea pueda ser diferente.

Adicionalmente y como podemos ver en la barra navegacional del Notebook, Nbgrader añade un botón **Validate** que permite a los profesores

comprobar si el código que ha implementado dentro de los comentarios de solución pasan los tests.

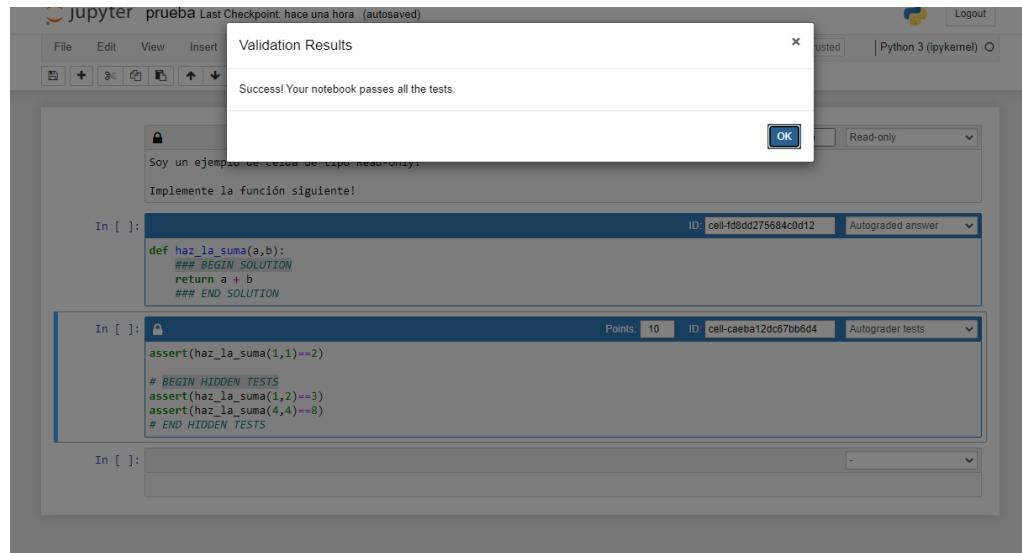


Figura E.16: Ejemplo validate

La forma en la que Nbgrader comprueba la superación de los ejercicios al ser ejecutada la orden de corrección es mediante la detección de errores. Cuando una celda es de tipo **Autograded Test** y la ejecución de estos ha resultado en algún error, la puntuación obtenida en ese ejercicio es 0 o la asignada a la celda en caso contrario. Por ello, queda a decisión de los profesores la forma en la que implementan sus tests. En el ejemplo anterior se está haciendo uso del método **assert** de Python.

La versión de los estudiantes de la tarea anterior, tras ser generada al crear una sección de forma directa o al ser publicada una sección no publicada, es la siguiente:

Soy un ejemplo de celda de tipo Read-Only!

Implemente la función siguiente!

```
In [ ]: 1 def haz_la_suma(a,b):
          2     # YOUR CODE HERE
          3     raise NotImplemented()
In [ ]: 1 assert(haz_la_suma(1,1)==2)
In [ ]: 1
```

Figura E.17: Ejemplo versión alumno

Como podemos ver, el código implementado por el profesor y los tests ocultos no son visualizables.

5 Acceso a todas las calificaciones

Para poder acceder a la página calificaciones global de todos los alumnos en sus cursos se ha de hacer click en el botón **Ir a las calificaciones** de la **E.3 Página principal del profesor** o en el botón **Calificaciones** de la barra navegacional superior. Esta página es la siguiente:

Calificaciones de sus alumnos:

Curso	Sección	Alumno	Usuario	Tarea	Calificación		
Curso 1	Sección de prueba	Alumno 2	Alumno 2	prueba	10	Descargar entrega	Descargar feedback

UNIVERSIDAD DE BURGOS
Proyecto de fin de grado de Alberto
Porres Fernández.

REPOSITORIO
GitHub

ENLACES
Nbgrader

CONTACTO
spf1002@eluubu.es

© 2022 Copyright

Figura E.18: Página de calificaciones global

En ella tendrá acceso a la descarga del archivo tarea entregado por el alumno mediante el botón **Descargar entrega** y a la descarga del documento feedback generado tras la calificación automática mediante el botón **Descargar Feedback**

6 Gestión de alumnos

Para poder acceder a la página gestión de alumnos click en el botón **Ir a la gestión de alumnos** de la **Página principal del profesor**(E.3) o en el botón **Estudiantes** de la barra navegacional superior. Esta página es la siguiente:

The screenshot shows a web page titled "Estudiantes pertenecientes a alguno de sus cursos:". Below the title is a table with two columns: "Nombre" and "Nombre de usuario". The table contains three rows, each with "Alumno 2" and "Alumno 2" respectively, and "Alumno 3" and "Alumno 3" respectively. At the bottom of the table is a button labeled "Crear nuevo alumno". Below the table is a footer navigation bar with four links: "UNIVERSIDAD DE BURGOS", "REPOSITORIO", "ENLACES", and "CONTACTO". Under "UNIVERSIDAD DE BURGOS", it says "Proyecto de fin de grado de Alberto Porres Fernández.". Under "REPOSITORIO", it says "GitHub" and "Nbgrader". Under "CONTACTO", it says "apf1002@alu.ubu.es".

Figura E.19: Página de gestión alumnos

6.1 Creación de alumnos

Al hacer click en el botón **Crear nuevo alumno** de la página de estudiantes irá al formulario de creación de alumno:

The screenshot shows a form titled "Creación de un nuevo alumno". It has four input fields: "Nombre del alumno", "Nombre de usuario del alumno", "Contraseña temporal", and "Repita la contraseña". Below the fields is a dropdown menu labeled "Elija un curso al que registrar al alumno" with the option "-- Seleccione una opción --". At the bottom is a "Crear" button.

Figura E.20: Página de creación de alumno

En ella deberá introducir su **nombre, el nombre de usuario de su cuenta, la contraseña temporal de esta y el curso al que quiere registrar al nuevo alumno.**

Si el nombre de usuario escogido para el alumno no pertenece a otro alumno este será creado de forma correcta y visualizable desde la **Página de gestión alumnos**(E.20).

6.2 Perfil del alumno

Al hacer click en el nombre un alumno en la **Página de gestión alumnos**(E.20) irá al perfil de calificaciones del alumno:

Curso	Sección	Tarea	Calificación			
Curso 1	Sección de prueba	prueba	10	Descargar entrega	Descargar feedback	Eliminar entrega

Calificaciones de Alumno 2:

UNIVERSIDAD DE BURGOS
Proyecto de fin de grado de Alberto
Porres Fernández.

REPOSITORIO
GitHub

ENLACES
NbGrader

CONTACTO
apf1002@alu.ubu.es

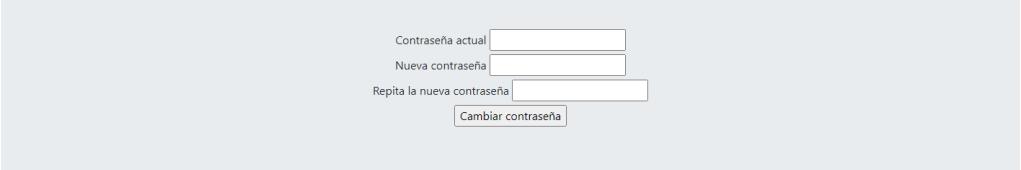
Figura E.21: Página de perfil del alumno

Desde esta podrá consultar las calificaciones de este, descargar los documentos de entrega y feedback como en la página de calificaciones global y adicionalmente:

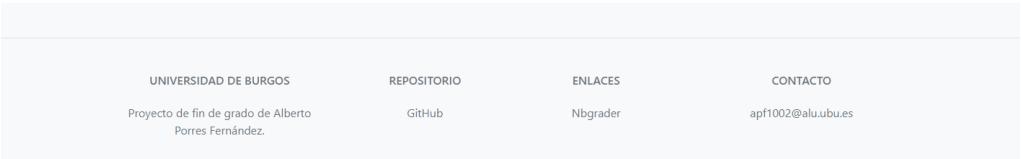
- **Eliminar entregas del estudiante** Al hacer click en el botón **Eliminar entrega** de la calificación, lo que permitirá al alumno volver a entregar la tarea de nuevo.
- **Eliminar la cuenta del alumno** Al hacer click en el botón **ELIMINAR ALUMNO**. Se ha de tener en cuenta al realizar esta acción que un alumno puede estar inscrito a cursos de otros profesores.

7 Cambio de contraseña

Para acceder al cambio de contraseña de su cuenta se ha de hacer click en el botón **Cambiar Contraseña** de la **Página principal del profesor**(E.3). Esto le llevará al formulario de cambio de contraseña:



Contraseña actual	<input type="text"/>
Nueva contraseña	<input type="text"/>
Repita la nueva contraseña	<input type="text"/>
<input type="button" value="Cambiar contraseña"/>	



UNIVERSIDAD DE BURGOS Proyecto de fin de grado de Alberto Porres Fernández.	REPOSITORIO GitHub Nbgrader	ENLACES	CONTACTO apf1002@alu.ubu.es
---	-----------------------------------	---------	--------------------------------

Figura E.22: Página de cambio de contraseña del profesor

Al completar el formulario y aceptar el modal de confirmación emergente, su sesión será cerrada siendo redireccionado al login.

8 Cierre de sesión

El cierre de sesión se realiza haciendo click en el botón **Cerrar Sesión** de la **Página principal del profesor**(E.3) y aceptando el modal de confirmación emergente.

E.5. Manual del alumno

A continuación se explicará el manejo de la plataforma orientado a los alumnos:

1 Acceso a la plataforma

El acceso a la plataforma es realizado a través del siguiente enlace: <http://3.89.140.10/> el cual nos lleva a la página inicial de esta.

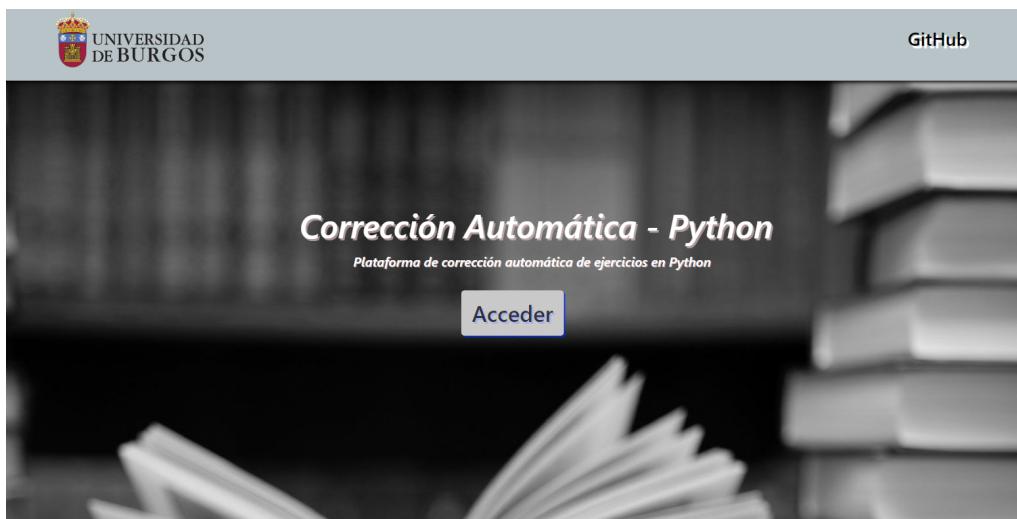


Figura E.23: Página inicial

2 Inicio de Sesión

El proceso de inicio de sesión contiene los siguientes paso:

- Una vez accedido a la Página incial hacer click en el botón **Acceder** lo cual nos llevará a la página de inicio de sesión:



Figura E.24: Pagina Inicio de sesión

- Introducir los credenciales de inicio de sesión y hacer click en el botón **Login**.
- Si los credenciales son correctos será llevado a la **Página principal del alumno**:

The screenshot shows the student main page. At the top, there is a menu bar with links for 'Inicio', 'Cursos', and 'Calificaciones'. The main content area has a title 'Bienvenido Alumno 2'. Below the title is a paragraph of text explaining the platform's purpose and how it works. There are two blue buttons at the bottom of this section: 'Ir a mis Cursos' and 'Ir a mis Calificaciones'. At the very bottom of the page, there is a footer with links to 'UNIVERSIDAD DE BURGOS', 'REPOSITORIO', 'ENLACES', and 'CONTACTO', along with some additional text and a copyright notice.

Figura E.25: Página Principal Alumno

- En caso de ser su **primer inicio de sesión en la plataforma** deberá **realizar el cambio de contraseña** por lo que será redireccionado directamente a esta página.

3 Acceso a Cursos

Para acceder a los cursos a los que el alumno está inscrito ha de hacer click en el botón **Ir a mis cursos** de la **Página principal del alumno**(E.25) o en el botón **Cursos** de la barra navegacional superior. Esto le llevará a la **Página de cursos del alumno**:

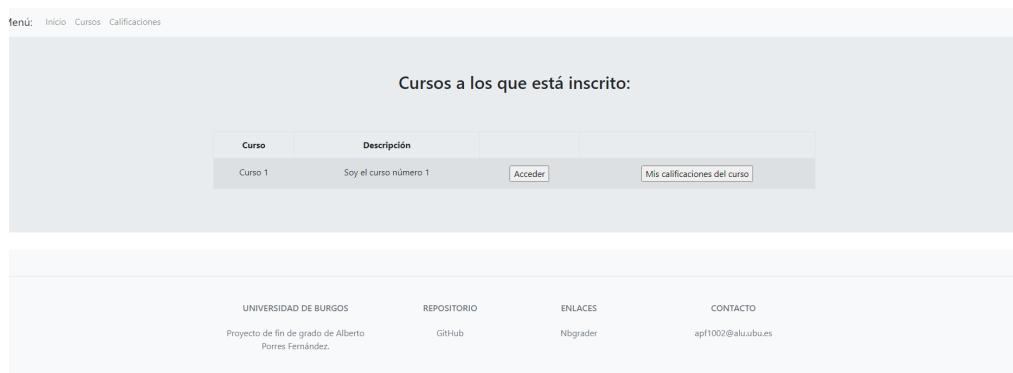


Figura E.26: Página de Cursos del alumno

3.1 Acceso a contenidos del curso

Una vez en la **Página de cursos del alumno** a los que este está inscrito, para acceder al contenido del curso ha de hacer click en el botón **Acceder** del curso deseado. Esto le llevará a la **Página de contenidos del curso**:



Figura E.27: Página de Contenidos del Curso

Los cursos están constituidos por secciones de contenido cada una de las cuales contiene un documento teórico y una tarea a entregar como se puede ver en la imagen anterior. En esta página podrá:

- Descargar los contenidos teóricos y tareas de cada sección mediante los botones **Descargar contenido teórico** y **Descargar tarea** de cada una.
- Realizar la entrega de tareas mediante el formulario de entrega de cada sección.
- Descargar los documentos tarea previamente entregados en aquellas secciones en las que la tarea ya se encuentre enviada mediante el botón **Descargar entrega** de la sección. En el caso de que un profesor elimine una entrega tendrá habilitada de nuevo el envío de la tarea para la sección.

3.2 Acceso a calificaciones del curso

Para consultar las calificaciones obtenidas en un curso específico ha de hacer click en el botón **Mis calificaciones del curso** en la **Página de cursos del alumno**(E.26). Esto le llevará a la **Página de calificaciones del curso**:

Sus calificaciones del curso Curso 1 :

Sección	Tarea	Calificación		
Sección de prueba	prueba	10	Descargar entrega	Descargar feedback

UNIVERSIDAD DE BURGOS
Proyecto de fin de grado de Alberto
Porres Fernández.

REPOSITORIO
GitHub

ENLACES
Nbgrader

CONTACTO
apf1002@alu.ubu.es

© 2022 Copyright

Figura E.28: Página de Calificaciones del Curso

En esta página tendrá acceso a los documentos tarea entregados y feedback de cada calificación mediante el botón **Descargar entrega** y **Descargar feedback**.

4 Resolución de tareas

La resolución de tareas debe realizarse mediante la herramienta Jupyter Notebooks la cual el alumno tiene que tener instalada en su ordenador de trabajo.

5 Consulta de todas las calificaciones

Para consultar todas las calificaciones obtenidas ha de hacer click en el botón **Ir a mis calificaciones** de la **Página principal del alumno**(E.25) o haciendo click en el botón **Calificaciones** de la barra navegacional superior. Esto le llevará a la **Página de calificaciones global**:

The screenshot shows a user interface for viewing grades. At the top, there is a menu bar with links to 'Inicio', 'Cursos', and 'Calificaciones'. Below this, a section titled 'Sus calificaciones:' displays a table with one row of data. The table has columns for 'CURSO', 'Sección', 'Tarea', and 'Calificación'. The data row shows 'Curso 1', 'Sección de prueba', 'prueba', and '10'. To the right of the table are two buttons: 'Descargar entrega' and 'Descargar feedback'. At the bottom of the page, there is a footer with links to 'UNIVERSIDAD DE BURGOS', 'REPOSITORIO' (GitHub), 'ENLACES' (Nbgrader), and 'CONTACTO' (email: apf1002@alu.ubu.es). A copyright notice at the very bottom reads '© 2022 Copyright'.

Figura E.29: Página de Calificaciones Global del alumno

Desde esta también tendrá acceso a los documentos de entrega y feedback mediante los respectivos botones.

6 Cambio de contraseña

Para acceder al cambio de contraseña de su cuenta se ha de hacer click en el botón **Cambiar Contraseña** de la **Página principal del alumno**[E.25](#). Esto le llevará al formulario de cambio de contraseña:

The screenshot shows a password change form. It contains three input fields: 'Contraseña actual', 'Nueva contraseña', and 'Repita la nueva contraseña'. Below these fields is a single button labeled 'Cambiar contraseña'. At the bottom of the page, there is a footer with links to 'UNIVERSIDAD DE BURGOS', 'REPOSITORIO' (GitHub), 'ENLACES' (Nbgrader), and 'CONTACTO' (email: apf1002@alu.ubu.es).

Figura E.30: Página de cambio de contraseña del alumno

Al completar el formulario y aceptar el modal de confirmación emergente, su sesión será cerrada siendo redireccionado al login.

7 Cierre de sesión

El cierre de sesión se realiza haciendo click en el botón **Cerrar Sesión** de la **Página principal del alumno**(E.25) y aceptando el modal de confirmación emergente.

Bibliografía

- [1] Css - cascading style sheets. <https://www.w3.org/Style/CSS/Overview.en.html>, 1996.
- [2] Nbgrader. <https://nbgrader.readthedocs.io/en/stable/>, 2015-2017.
- [3] Nbgrader. https://nbgrader.readthedocs.io/en/stable/api/high_level_api.html, 2015-2017.
- [4] Amazon. Aws. <https://aws.amazon.com/es/>, 2006.
- [5] Michael Baye. Sqlalchemy. <https://www.sqlalchemy.org/>, 2006.
- [6] Jack Dorsey. Bootstrap. <https://getbootstrap.com/docs/4.0/getting-started/introduction/>, 2011.
- [7] R.G. Duque. *Python para todos*. mundogeek.net, 2016.
- [8] D. Richard Hipp. Sqlite database engine. <https://www.sqlite.org/index.html>, 2000.
- [9] Solomon Hykes. Docker. <https://www.docker.com/>, 2013.
- [10] Fernando Pérez. Jupyter notebooks. <https://jupyter.org/>, 2014.
- [11] Armin Ronacher, David Lord, Adrian Mönnich, and Markus Unterwaditzer. Flask-login. <https://flask-login.readthedocs.io/en/latest/>, 2010.
- [12] Guido van Rossum. Python programming language. <https://www.python.org/>, 1991.

- [13] Axiom Zen. Zenhub. <https://www.zenhub.com/>, 2014.