

Proyecto Fin de Carrera

Ingeniería de Telecomunicación

JupyterLab – Creación de un entorno docente para enseñanza semipresencial

Autor: Alejandro Márquez Toscano

Tutor: Alejandro Escudero Santana

**Dpto. Organización Industrial y Gestión de
Empresas II
Universidad de Sevilla**

Sevilla, 2022



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

JupyterLab – Creación de un entorno docente para enseñanza semipresencial

Autor:

Alejandro Márquez Toscano

Tutor:

Alejandro Escudero Santana

Profesor titular

Dpto. de Organización Industrial y Gestión de Empresas II

Universidad de Sevilla

Sevilla, 2022

Proyecto Fin de Carrera: JupyterLab – Creación de un entorno docente para enseñanza semipresencial

Autor: Alejandro Márquez Toscano

Tutor: Alejandro Escudero Santana

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

Antes que nada, quisiera agradecer a mis padres todo el apoyo brindado durante todos los años de estudio. Sin su ayuda no hubiese sido posible llegar hasta aquí.

También me gustaría acordarme de todos los profesores que me han acompañado durante estos años de carrera compartiendo su tiempo y sus conocimientos conmigo. Gracias por su dedicación.

Por último y no menos importante, agradecer a Alejandro, mi tutor de este trabajo, por ayudarme a realizar este proyecto de fin de grado.

Gracias a la tecnología, y a los lenguajes de programación en concreto, disponemos de novedosas herramientas para realizar tareas, análisis y otras actividades de una más forma rápida y sencilla que los métodos usados tradicionalmente. Por ello, ya son varios los grados que incluyen en su plan docente alguna asignatura donde se apliquen métodos de programación.

Un problema con el que se están encontrando algunos docentes es, en líneas generales, la dificultad que tienen la mayoría de los alumnos, no pertenecientes a grados relacionados esencialmente con la programación, para desplegar y adaptarse al entorno (máquinas virtuales, intérpretes/compiladores, editores de código, etc), entorpeciendo la curva de aprendizaje deseada.

El objetivo de este documento es la puesta en marcha de una plataforma docente que permita realizar laboratorios y exámenes de una forma “amigable” con aquellos estudiantes que no posean experiencia previa programando.

Abstract

Thanks to technology, and programming languages in particular, we have new tools to carry out tasks, analyses and other activities more quickly and easily than the methods traditionally used. For this reason, there are already several degrees that include in their teaching plan a subject where programming methods are applied.

A problem that some teachers are encountering is, in general terms, the difficulty that most students, not belonging to degrees essentially related to programming, have in deploying and adapting to the environment (virtual machines, interpreters/compiler, code editors, etc.), hindering the desired learning curve.

The aim of this document is the implementation of a teaching platform that allows laboratories and exams to be carried out in a "friendly" way for those students who have no previous programming experience.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
Acrónimos	xxi
1 Introducción	23
1.1 <i>Motivación</i>	23
1.2 <i>Objetivo</i>	23
1.3 <i>Alcance</i>	24
2 Estado del arte	25
2.1 <i>Programación</i>	25
2.1.1 La programación en la actualidad	25
2.2 <i>Sistemas de gestión de aprendizaje</i>	26
2.2.1 Introducción	26
2.2.2 Tecnicismos sobre los LMS	26
2.2.3 Mercado de los LMS	27
2.2.4 Ventajas de los LMS	27
3 Diseño	29
3.1 <i>Introducción</i>	29
3.2 <i>Especificaciones y análisis de requisitos</i>	29
3.2.1 Requisitos generales	30
3.2.2 Requisitos funcionales	33
3.2.3 Requisitos no funcionales	40
3.2.4 Restricciones técnicas	43
3.3 <i>Diseño y funcionamiento del sistema</i>	44
3.3.1 Introducción	44
3.3.2 Funcionamiento	45
4 Implementación	51
4.1 <i>Introducción</i>	51
4.2 <i>Jupyter Notebook</i>	51
4.2.1 Introducción	51
4.2.2 Distribuciones de JupyterHub	52
4.2.3 Arquitectura de JupyterHub	52
4.3 <i>nbgrader</i>	54
4.3.1 Introducción	54
4.3.2 Arquitectura de nbgrader	54
4.4 <i>Amazon EC2</i>	55
4.5 <i>Instalación y configuración de los servicios</i>	56

4.5.1	Configuración de Amazon EC2	56
4.5.2	Instalación y configuración de JupyterHub	59
4.5.3	Instalación y configuración de nbgrader	61
5	Uso de la aplicación	65
5.1	<i>Inicio de sesión</i>	65
5.2	<i>Gestión de usuarios</i>	66
5.3	<i>Uso de notebooks</i>	67
5.4	<i>Creación, entrega y corrección de un assignment</i>	68
6	Pruebas de ejecución	73
6.1	<i>Despliegue de la aplicación</i>	73
6.2	<i>Pruebas de funcionamiento</i>	73
6.3	<i>Pruebas de rendimiento</i>	75
6.4	<i>Pruebas de seguridad</i>	75
7	Conclusión y líneas de futuro	79
7.1	<i>Conclusiones</i>	79
7.2	<i>Líneas de futuro</i>	80
	Referencias	81

ÍNDICE DE TABLAS

Tabla 3-1 Formato de tabla de requisito.	30
Tabla 3-2 Requisito general 1 "Sistema de inicio de sesión".	30
Tabla 3-3 Requisito general 2 "Existencia de cuenta de superusuario".	30
Tabla 3-4 Requisito general 4 "Editor de código".	31
Tabla 3-5 Requisito general 4 "Sistema de ejecución de código".	31
Tabla 3-6 Requisito general 6 "Calificar a usuarios".	31
Tabla 3-7 Requisito general 6 "Arranque del sistema".	32
Tabla 3-8 Requisito general 7 "Mensajes informativos".	32
Tabla 3-9 Requisito general 8 "Configuración de proyectos".	32
Tabla 3-10 Requisito general 9 "Sistema operativo".	33
Tabla 3-11 Requisito general 10 "Navegador web".	33
Tabla 3-12 Caso de uso 1 "Iniciar sesión".	33
Tabla 3-13 Caso de uso 2 "Cerrar sesión".	34
Tabla 3-14 Caso de uso 3 "Registrar usuario".	34
Tabla 3-15 Caso de uso 4 "Borrar usuario".	34
Tabla 3-16 Caso de uso 5 "Cambiar contraseña".	34
Tabla 3-17 Caso de uso 6 "Crear fichero".	35
Tabla 3-18 Caso de uso 7 "Subir fichero".	35
Tabla 3-19 Caso de uso 8 "Editar fichero".	35
Tabla 3-20 Caso de uso 9 "Eliminar fichero".	36
Tabla 3-21 Caso de uso 10 "Habilitar/Deshabilitar fichero".	36
Tabla 3-22 Caso de uso 11 "Ejecutar fichero".	36
Tabla 3-23 Caso de uso 12 "Entregar fichero".	36
Tabla 3-24 Caso de uso 13 "Calificar actividad".	37
Tabla 3-25 Caso de uso 14 "Publicar calificación".	37
Tabla 3-26 Regla de negocio 1 "Roles de usuario".	39
Tabla 3-27 Regla de negocio 2 "Profesores de un curso".	39
Tabla 3-28 Requisito de información 1 "Almacenamiento sobre usuarios y proyectos".	39
Tabla 3-29 Requisito de fiabilidad 1 "Error durante diferentes procesos".	40
Tabla 3-30 Requisito de fiabilidad 2 "Alta disponibilidad".	40
Tabla 3-31 Requisito de fiabilidad 3 "Nombre de proyectos".	40
Tabla 3-32 Requisito de usabilidad 1 "Manual de instalación".	41
Tabla 3-33 Requisito de usabilidad 2 "Manual de uso".	41
Tabla 3-34 Requisito de usabilidad 3 "Indicaciones de uso".	41

Tabla 3-35 Requisito de mantenibilidad 1 "Reutilización".	42
Tabla 3-36 Requisito de seguridad 1 "Control de acceso a la plataforma".	42
Tabla 3-37 Requisito de seguridad 2 "Control de sesión".	42
Tabla 3-38 Requisito de seguridad 3 "Cambio de contraseña".	42
Tabla 3-39 Requisito de seguridad 4 "Control de acceso a recursos".	43
Tabla 3-40 Restricción técnica 1 "Sistema operativo".	43
Tabla 3-41 Restricción técnica 2 "Navegador".	43
Tabla 3-42 Restricción técnica 3 "Conexión".	44

ÍNDICE DE FIGURAS

Ilustración 2-1 Vista general de un LMS.	26
Ilustración 2-2 Comparativa del mercado de LMS.	27
Ilustración 3-1 Diagrama de casos de uso.	38
Ilustración 3-2 Diagrama de paquetes.	44
Ilustración 3-3 Diagrama de jerarquía de usuarios.	45
Ilustración 3-4 Diagrama de secuencia "Registro de usuario".	46
Ilustración 3-5 Diagrama de secuencia "Inicio de sesión".	47
Ilustración 3-6 Caso de uso "Asignar actividad"	48
Ilustración 3-7 Caso de uso "Calificar actividad".	48
Ilustración 3-8 Caso de uso "Ejecutar fichero".	49
Ilustración 4-1 Arquitectura JupyterHub	53
Ilustración 4-2 Regiones de AWS	56
Ilustración 4-3 Menú EC2: "Instancias" 1	57
Ilustración 4-4 Configuración instancia: "Sistema operativo".	57
Ilustración 4-5 Configuración de instancia: "Tipo de instancia".	58
Ilustración 4-6 Configuración de instancia: "Par de claves".	58
Ilustración 4-7 Configuración de instancia: "Configuraciones de red".	59
Ilustración 4-8 Menú EC2: "Instancias" 2	59
Ilustración 4-9 Menú JupyterHub: "Inicio de sesión".	60
Ilustración 4-10 Menú JupyterHub: "Control Panel".	62
Ilustración 4-11 Escenario "Una clase y un profesor"	63
Ilustración 5-1 Menú JupyterHub: "Inicio de sesión".	66
Ilustración 5-2 Menú JupyterHub: "Página principal".	66
Ilustración 5-3 Menú JupyterHub: "Añadir usuarios".	67
Ilustración 5-4 Menú JupyterHub: "Administrador".	67
Ilustración 5-5 Menú JupyterHub: "Crear Notebook".	67
Ilustración 5-6 Menú JupyterHub: "Notebook".	68
Ilustración 5-7 Menú JupyterHub: "Notebook - celda de código".	68
Ilustración 5-8 Menú JupyterHub: "Ruta del fichero Optimizacion1.ipynb".	69
Ilustración 5-9 Menú JupyterHub: "Notebook - diseño de Optimizacion1.ipynb".	69
Ilustración 5-10 Menú JupyterHub: "Formgrader" 1.	70
Ilustración 5-11 Menú JupyterHub: "Assignments".	70
Ilustración 5-12 Menú JupyterHub: "Formgrader" 2.	71
Ilustración 5-13 Menú JupyterHub: "Formgrader" 3.	71

Ilustración 5-14 Menú JupyterHub: " <i>Assignments</i> " 2.	71
Ilustración 5-15 Menú JupyterHub: "Corrección actividad - directorio".	72
Ilustración 5-16 Menú JupyterHub: "Corrección actividad - archivo".	72
Ilustración 6-1 Menú JupyterHub: "Terminal".	74
Ilustración 6-2 Menú JupyterHub: " <i>Formgrader</i> " 4.	74
Ilustración 6-3 Menú JupyterHub: "Cambiar contraseña".	75
Ilustración 6-4 Menú JupyterHub: "Acceso prohibido".	76
Ilustración 6-5 Menú JupyterHub: "Terminal - permiso denegado".	76
Ilustración 6-6 Menú JupyterHub: "Sesión caducada"	76

Acrónimos

IT	Information Technology
IDE	Integrated Development Environment
LMS	Learning Management System
IEEE	Institute of Electrical and Electronics Engineers
SaaS	Software as a Service
AWS	Amazon Web Services
GCP	Google Cloud Platform
TCP/IP	Transmission Control Protocol/Internet Protocol
HTTP	Hyper Text Transfer Protocol
EC2	Elastic Compute
SSH	Secure Shell
USD	US Dollar
HTML	HyperText Markup Language

1 INTRODUCCIÓN

*“La inspiración existe, pero tiene que encontrarte
trabajando”*

- Pablo Picasso -

Las nuevas tecnologías están permitiendo grandes avances en la mayoría de campos que forman parte en el desarrollo de la vida de los seres humanos. En particular, la docencia se está viendo afectada de manera positiva a través de la introducción de novedosos sistemas y prácticas de aprendizaje gracias a la tecnología.

1.1 Motivación

Actualmente, ya son numerosos los campos profesionales que están aprovechándose de las ventajas y funciones que ofrece la programación para optimizar los modelos de negocios con los que ya trabajan o para disponer de otros nuevos.

A causa de esta nueva realidad, los grados universitarios están actualizando sus planes docentes, incluyendo asignaturas donde hay que realizar tareas de programación.

Sin embargo, los docentes se están encontrando con alumnos que tienen dificultades a la hora de poner en marcha el entorno necesario para programar y realizar las prácticas correspondientes a las asignaturas.

Esto provoca que los alumnos que encuentran más dificultades a la hora de desplegar todo el entorno de desarrollo se desmotiven con gran facilidad, provocando que no le dediquen el empeño necesario para superar la asignatura e incluso abandonandola en algunos casos. Además, también causa que haya que dedicar más tiempo a las sesiones introductorias para conseguir que todos los alumnos puedan avanzar sin quedarse atrapados en fases preliminares.

En este contexto, surge la idea de ayudar a los alumnos para que, de forma autónoma, puedan realizar sus tareas de programación sin complicaciones relacionadas con instalaciones de editores de código, intérpretes o compiladores.

1.2 Objetivo

El objetivo principal de este trabajo es poner en marcha una plataforma donde los alumnos puedan pasar directamente a programar sin tener que pasar por tediosas tareas de preparación del entorno necesario para realizar las actividades o prácticas diseñadas para la asignatura.

Además, se debe buscar un sistema de intercambio de archivos entre el profesor y los alumnos, facilitando al docente la tarea de lanzar actividades a los alumnos, recogerlas una vez realizadas,

corregir dichas actividades y hacerles llegar la calificación obtenida en base a la validez de sus respuestas.

Básicamente, la plataforma debe consistir en una interfaz gráfica en la que se integre un editor de código. Además, la herramienta debe ir acompañada de un kernel que permita la ejecución del código escrito por el usuario en la aplicación web. El trabajo final tiene que ser almacenable en un archivo y exportable para que sea posteriormente entregado por parte de los alumnos.

La aplicación debe estar desplegada en un servidor que permita su acceso remoto, tanto desde la red interna de la Universidad de Sevilla como desde la red pública.

1.3 Alcance

Este proyecto se compone de diferentes fases:

- En primer lugar, se estudiará la realidad actual, explicando que son los sistemas LMS y la situación vigente de ellos.
- A continuación, se explicará el diseño de la herramienta necesaria a poner en marcha para cumplir con el objetivo de este trabajo.
- Seguidamente, se analizará la implementación escogida para hacer realidad el diseño previamente desarrollado junto con las instrucciones de instalación y configuración de las diferentes herramientas.
- Posteriormente, se desarrollará un capítulo explicando los diferentes usos que posee la aplicación.
- Después, se expondrán ciertas pruebas de ejecución realizadas para demostrar el correcto funcionamiento de dicha plataforma.
- Para terminar, se plantearán unas conclusiones alcanzadas con el desarrollo de este proyecto.

2 ESTADO DEL ARTE

*“La vida no se trata de encontrarte a ti mismo,
sino de crearte a ti mismo”*

- Bernard Shaw -

En este punto dedicado al estado del arte se hablará sobre el papel que juega la programación hoy en día y cómo se realizan actualmente en entornos docentes las prácticas cuyo objetivo es desarrollar un código. Además, se comentarán qué son los sistemas LMS, su funcionalidad y sus ventajas. Todo esto nos servirá para comprender el contexto y el objetivo de este trabajo.

2.1 Programación

2.1.1 La programación en la actualidad

En nuestros días, vivimos en la era de la información [1] (también conocida como era digital), iniciada por la revolución tecnológica que se lleva viendo desde finales del siglo XIX. La tecnología y su avance va aumentando de forma exponencial, y ya forma parte esencial de la vida de los seres humanos.

Todo esto está, en gran parte, provocado por la programación y su aplicación a casi cualquier ámbito que maneje el ser humano. Gracias a los ordenadores y los lenguajes de programación, tareas que ocupaban varias horas o incluso días, pueden ser realizadas hoy en día en cuestión de minutos.

A partir de la programación se han desarrollado infinidad de softwares que permiten realizar numerosas actividades a través de los ordenadores y smartphones, como las compras online, la gestión de servicios públicos a través de internet, la comunicación instantánea entre personas via e-mail o chat o el visionado de videos bajo demanda, entre muchas otras posibilidades. Para ilustrar este hecho por medio de un ejemplo, un estudio reciente ha desvelado que actualmente 8 de cada 10 servicios de las administraciones públicas de la Unión Europea se encuentran disponibles de forma *online* [2].

Es por ello que, a día de hoy, se necesitan profesionales con conocimientos de programación para puestos de trabajo de diferentes campos, no relacionados directamente con sectores IT, como por ejemplo en procesos de investigación de la salud, donde está ganando fuerza estos últimos años.

Las universidades no son ajenas a este *boom* que está teniendo la programación, y ya son varios los grados, no relacionados esencialmente con el mundo IT, que incluyen asignaturas donde se tocan conceptos y se realizan prácticas relacionadas con programar.

Por su parte, los docentes utilizan diferentes herramientas para que los alumnos lleguen a cabo las prácticas de la asignatura. Entre las más conocidas se encuentran dos IDEs como Eclipse [3], enfocado para programar con Java, y Spyder [4], para programación científica en Python, Visual Studio Code[5], que no es más que un potente editor de código con un amplio mercado de extensiones que lo

complementan de diferentes tipos de funcionalidad, y, por último, montar un entorno de desarrollo con una máquina virtual corriendo Ubuntu [6] y con Eclipse [7] como editor de código.

2.2 Sistemas de gestión de aprendizaje

2.2.1 Introducción

Un sistema de gestión de aprendizaje [8] (LMS o *learning management system* en inglés) es un software cuyo fin es gestionar cursos educativos para enseñanza no presencial. Estos sistemas permiten el seguimiento, entrega y evaluación de actividades de forma asíncrona y sin necesidad de presencialidad física por parte de los actores. Este software se instala en un servidor web para que sea accedido a través del navegador y, en algunos casos, por medio de aplicaciones web.

A continuación, se adjunta una figura donde se representa de una manera simplificada el funcionamiento de las aplicaciones LMS y de los actores que forman parte en su uso.



Ilustración 2-1 Vista general de un LMS.

En los sistemas LMS se almacenan y se ofrecen, de forma instantánea y remota, elementos de enseñanza como actividades, videos y diapositivas, calificaciones obtenidas durante el curso y sistema de comunicación para la organización interna del curso.

Existen diferentes tipos de usuarios, entre los que se pueden diferenciar a los estudiantes, que harán uso de los contenidos subidos por los profesores, los cuales se engloban en otro grupo de usuarios. Por último, se tiene a los administradores, que se encargan de la configuración, *monitoring* y actualizaciones del sistema.

2.2.2 tecnicismos sobre los LMS

Los sistemas LMS son aplicaciones de tipo web, accedidas normalmente a través de navegadores y, en ciertas ocasiones, cuentan con programas de escritorio para acceder a sus servicios. Estas herramientas suelen estar desarrolladas los lenguajes de programación de Java y PHP. Para el

almacenamiento de información cuentan con bases de datos como MySQL [9].

Tradicionalmente, estos sistemas se instalaban de forma local en un servidor perteneciente a la institución que realizaba la compra de la licencia del software. Actualmente, con las mejoras del hardware y de las conexiones a internet y la popularización de la tecnología *Cloud* [10], se ha popularizado la opción del software como servicio (SaaS) [11], ofreciendo el acceso a la Plataforma LMS a través de internet. Mediante esta opción, el proveedor del servicio deberá proporcionar el hardware y el mantenimiento del mismo.

Cabe destacar que para que una plataforma dedicada a la enseñanza *online* sea considerada un LMS, debe cumplir los estándares estipulados por el IEEE [12], garantizando un mínimo de funcionalidades básicas.

2.2.3 Mercado de los LMS

La siguiente imagen muestra una comparativa de un estudio de mercado realizado en marzo de 2018 sobre los LMS más utilizados.



Ilustración 2-2 Comparativa del mercado de LMS.

Entre los sistemas más populares, se puede destacar a Moodle [13]. Este LMS es de código libre y tiene una gran aceptación entre la comunidad educativa. Desde su web oficial afirman tener 316 millones de usuarios y cuentan con instituciones importantes utilizando sus servicios, como la Universidad de Cambridge.

Por otro lado, en el ranking también se encuentra en buena posición Blackboard [14]. La Universidad de Sevilla ha confiado en esta compañía para desarrollar su LMS, conocido como Enseñanza Virtual [15].

La popularización de estas herramientas ha dado lugar a que los vendedores desarrollen diferentes productos LMS enfocándose según el público objetivo que quieran alcanzar con ellos, los cuales son:

- Académico: pensado para ser usado por la comunidad educativa, como por ejemplo universidades (Blackboard o Moodle).
- Corporativo: para empresas con el fin de formar a empleados y clientes (SuccessFactors).
- Comercial: el objetivo de estos LMS es para individuos puedan vender cursos en líneas para empresas y particulares (Udemy).

2.2.4 Ventajas de los LMS

Entre las ventajas que se pueden destacar de la implementación de los sistemas LMS en cursos académicos se destacan las siguientes:

-
- Posibilidad de controlar y seguir el progreso del curso y el trabajo de los alumnos de forma particular.
 - La programación automática de publicación de actividades, su entrega a distancia de y ofrecer posteriormente la calificación obentida.
 - La inmediatez de acceso a los contenidos publicados.
 - El funcionamiento autónomo que poseen los alumnos para organizar su propio trabajo.
 - La facilidad para la organización de las clases, pudiendo programar de forma automática la publicación de tareas y contando con un sistema de comunicación interno.

3 DISEÑO

“Si la oportunidad no llama, construye una puerta”

- Milton Berle-

Este apartado está dedicado a explicar en profundidad el diseño necesario del software a poner en marcha. A lo largo del apartado se hará una breve introducción al contenido, seguidamente, se estudiarán los requisitos y especificaciones a cumplir con el diseño del programa y, por último, se detallará el funcionamiento básico esperado.

3.1 Introducción

La plataforma a diseñar se debe tratar de una aplicación de tipo web de la que puedan hacer uso tanto alumnos como profesores de asignaturas en las cuales se impartan asignaturas donde haya que realizar tareas de programación.

A grandes rasgos, la herramienta debe permitir crear y desarrollar actividades relacionadas con la programación y que puedan ser asignadas a usuarios con rol de estudiantes. En esta plataforma se debe ser capaz de realizar las actividades propuestas y probar el funcionamiento del código *in situ*. Además, se debe tener la funcionalidad de calificar las entregas realizadas por los alumnos y compartir las puntuaciones con ellos mismos.

Esta plataforma debe ser instalada en un servidor, ya sea en uno local de la Universidad o en un servidor remoto alojado en un proveedor de cloud pública, como AWS [16], Azure [17] o GPC [18].

3.2 Especificaciones y análisis de requisitos

A continuación, se especifican los requisitos que debe cumplir nuestro sistema para cumplir con las necesidades requeridas por los usuarios.

Para describir los requisitos, se usa el siguiente formato de tabla:

IDENTIFICADOR	Nombre del requisito
Dependencias	
Descripción	
Importancia	

Prioridad	
------------------	--

Tabla 3-1 Formato de tabla de requisito.

Las entradas de la tabla significan lo siguiente:

- Dependencias: nombre de los requisitos con los que está relacionado directamente o de los que depende para su implementación.
- Descripción: breve resumen explicando en que consiste.
- Importancia: indica el valor que tiene el requisito para el cliente.
- Prioridad: indica el valor que tiene el requisito para el desarrollador.

Para marcar la relevancia de los parámetros de importancia y prioridad se tienen 3 diferentes grados:

- Alta: requisito imprescindible para el proyecto.
- Media: omitible en situaciones donde suponga la eliminación de un requisito de alta importancia y/o prioridad.
- Baja: se recomienda, pero no es fundamental.

3.2.1 Requisitos generales

RG-01	Sistema de inicio de sesión
Dependencias	RG-02, CU-01
Descripción	Se dispondrá de un panel para que los usuarios accedan usando sus credenciales.
Importancia	Alta
Prioridad	Alta

Tabla 3-2 Requisito general 1 "Sistema de inicio de sesión".

RG-02	Existencia de cuenta de <i>superusuario</i>
Dependencias	RG-01
Descripción	Habrà un usuario con máximos permisos, encargado de administrar la aplicación y del resto de los usuarios.
Importancia	Alta
Prioridad	Alta

Tabla 3-3 Requisito general 2 "Existencia de cuenta de superusuario".

RG-03	Editor de código
Dependencias	-
Descripción	Un editor de código de lenguaje de programación disponible para todos los usuarios estará incorporado en el sistema.
Importancia	Alta
Prioridad	Alta

Tabla 3-4 Requisito general 4 "Editor de código".

RG-04	Sistema de ejecución de código
Dependencias	-
Descripción	Se integrará un sistema que pueda ejecutar código correspondiente a lenguajes de programación.
Importancia	Alta
Prioridad	Alta

Tabla 3-5 Requisito general 4 "Sistema de ejecución de código".

RG-05	Calificar a usuarios
Dependencias	-
Descripción	Se dispondrá de la funcionalidad de puntuar las actividades entregadas por los usuarios y que los alumnos puedan ver la calificación obtenida.
Importancia	Alta
Prioridad	Alta

Tabla 3-6 Requisito general 6 "Calificar a usuarios".

RG-06	Arranque del sistema
Dependencias	-
Descripción	Al arrancar el servicio iniciará el proceso en el puerto correspondiente del servidor, comprobar el estado de la base de datos, habilitar la conexión por http, etc.

Importancia	Alta
Prioridad	Alta

Tabla 3-7 Requisito general 6 "Arranque del sistema".

RG-07	Mensajes informativos
Dependencias	RU-03, RF-01
Descripción	La aplicación proporcionará mensajes al usuario cuando se realicen operaciones ya sean exitosas o erróneas.
Importancia	Media
Prioridad	Media

Tabla 3-8 Requisito general 7 "Mensajes informativos".

RG-08	Configuración de proyectos
Dependencias	RI-01
Descripción	Los archivos sobre los que trabajen los usuarios deben tener los siguientes parámetros: <ul style="list-style-type: none"> • Nombre del archivo. • Propietario. • Fecha de último guardado. • Extensión del fichero (diferentes dependiendo del lenguaje de programación del que se haga uso). • Calificación
Importancia	Alta
Prioridad	Alta

Tabla 3-9 Requisito general 8 "Configuración de proyectos".

RG-09	Sistema operativo
Dependencias	RT-01
Descripción	El servidor del servicio debe ser instalable tanto en sistemas Linux como Windows.

Importancia	Alta
Prioridad	Alta

Tabla 3-10 Requisito general 9 "Sistema operativo".

RG-10	Navegador web
Dependencias	RG-02
Descripción	La aplicación tendrá compatibilidad con los principales navegadores (Google Chrome, Mozilla Firefox, Microsoft Edge, etc).
Importancia	Alta
Prioridad	Alta

Tabla 3-11 Requisito general 10 "Navegador web".

3.2.2 Requisitos funcionales

3.2.2.1 Casos de uso

CU-01	Iniciar sesión
Dependencias	RG-01
Descripción	Los usuarios tendrán que iniciar sesión haciendo uso de las credenciales establecidas para acceder a las funcionalidades internas del sistema. Según el rol asociado a su perfil, tendrán ciertos permisos que les permitirá o no realizar ciertas acciones.
Importancia	Alta
Prioridad	Alta

Tabla 3-12 Caso de uso 1 "Iniciar sesión".

CU-02	Cerrar sesión
Dependencias	RG-01
Descripción	Los usuarios podrán cerrar sesión a través de un botón visible en la interfaz gráfica.
Importancia	Alta

Prioridad	Alta
------------------	------

Tabla 3-13 Caso de uso 2 "Cerrar sesión".

CU-03	Registrar usuario
Dependencias	RG-01, RG-02, RN-01
Descripción	Los usuarios con rol de administradores podrán añadir usuarios al sistema, los parámetros a introducir serán nombre de usuario y rol de administrador (Sí/No).
Importancia	Alta
Prioridad	Alta

Tabla 3-14 Caso de uso 3 "Registrar usuario".

CU-04	Borrar usuario
Dependencias	RG-01, RG-02, RN-01
Descripción	Los usuarios con rol de administrador tendrán la posibilidad de borrar a otros usuarios.
Importancia	Alta
Prioridad	Alta

Tabla 3-15 Caso de uso 4 "Borrar usuario".

CU-05	Cambiar contraseña
Dependencias	RG-01, RS-03
Descripción	Todos los usuarios deben poder cambiar la contraseña con la que acceden al sistema.
Importancia	Alta
Prioridad	Alta

Tabla 3-16 Caso de uso 5 "Cambiar contraseña".

CU-06	Crear fichero
--------------	---------------

Dependencias	RG-01
Descripción	Todos los usuarios tendrán la posibilidad de crear nuevos ficheros sobre los que escribir texto plano y desarrollar código.
Importancia	Alta
Prioridad	Alta

Tabla 3-17 Caso de uso 6 "Crear fichero".

CU-07	Subir fichero
Dependencias	RG-01
Descripción	Cualquier usuario podrá importar archivos ya creados al servidor de la aplicación para poder trabajar con ellos.
Importancia	Alta
Prioridad	Alta

Tabla 3-18 Caso de uso 7 "Subir fichero".

CU-08	Editar fichero
Dependencias	RG-01
Descripción	Los usuarios podrán editar los ficheros existentes en el servidor, ya sean creados, subidos o entregados por el profesor a través del sistema.
Importancia	Alta
Prioridad	Alta

Tabla 3-19 Caso de uso 8 "Editar fichero".

CU-09	Eliminar fichero
Dependencias	RG-01
Descripción	Se podrán eliminar del servidor los ficheros de los cuales se sean propietarios.
Importancia	Alta
Prioridad	Alta

Tabla 3-20 Caso de uso 9 "Eliminar fichero".

CU-10	Asignar actividad
Dependencias	RG-01, RN-01, RN-02
Descripción	Los profesores asignar las actividades que requieran ser entregadas a los alumnos.
Importancia	Alta
Prioridad	Alta

Tabla 3-21 Caso de uso 10 "Habilitar/Deshabilitar fichero".

CU-11	Ejecutar fichero
Dependencias	RG-01
Descripción	Todos los usuarios podrán ejecutar los ficheros de los cuales sean propietarios o tengan permisos.
Importancia	Alta
Prioridad	Alta

Tabla 3-22 Caso de uso 11 "Ejecutar fichero".

CU-12	Entregar fichero
Dependencias	RG-01, RN-01
Descripción	Los alumnos podrán entregar a través de la herramienta sus trabajos una vez consideren que han sido completados.
Importancia	Alta
Prioridad	Alta

Tabla 3-23 Caso de uso 12 "Entregar fichero".

CU-13	Calificar actividad
Dependencias	RG-01, RN-01, RN-02
Descripción	Los profesores tendrán la funcionalidad de corregir y asignar una

	puntuación a las actividades entregadas por los alumnos.
Importancia	Alta
Prioridad	Alta

Tabla 3-24 Caso de uso 13 "Calificar actividad".

CU-14	Publicar calificación
Dependencias	RG-01, RN-01, RN-02
Descripción	Los profesores podrán publicar las calificaciones correspondientes a las actividades realizadas por los alumnos.
Importancia	Alta
Prioridad	Alta

Tabla 3-25 Caso de uso 14 "Publicar calificación".

Gráficamente, se muestran los casos de uso y los actores en la figura 3-1.

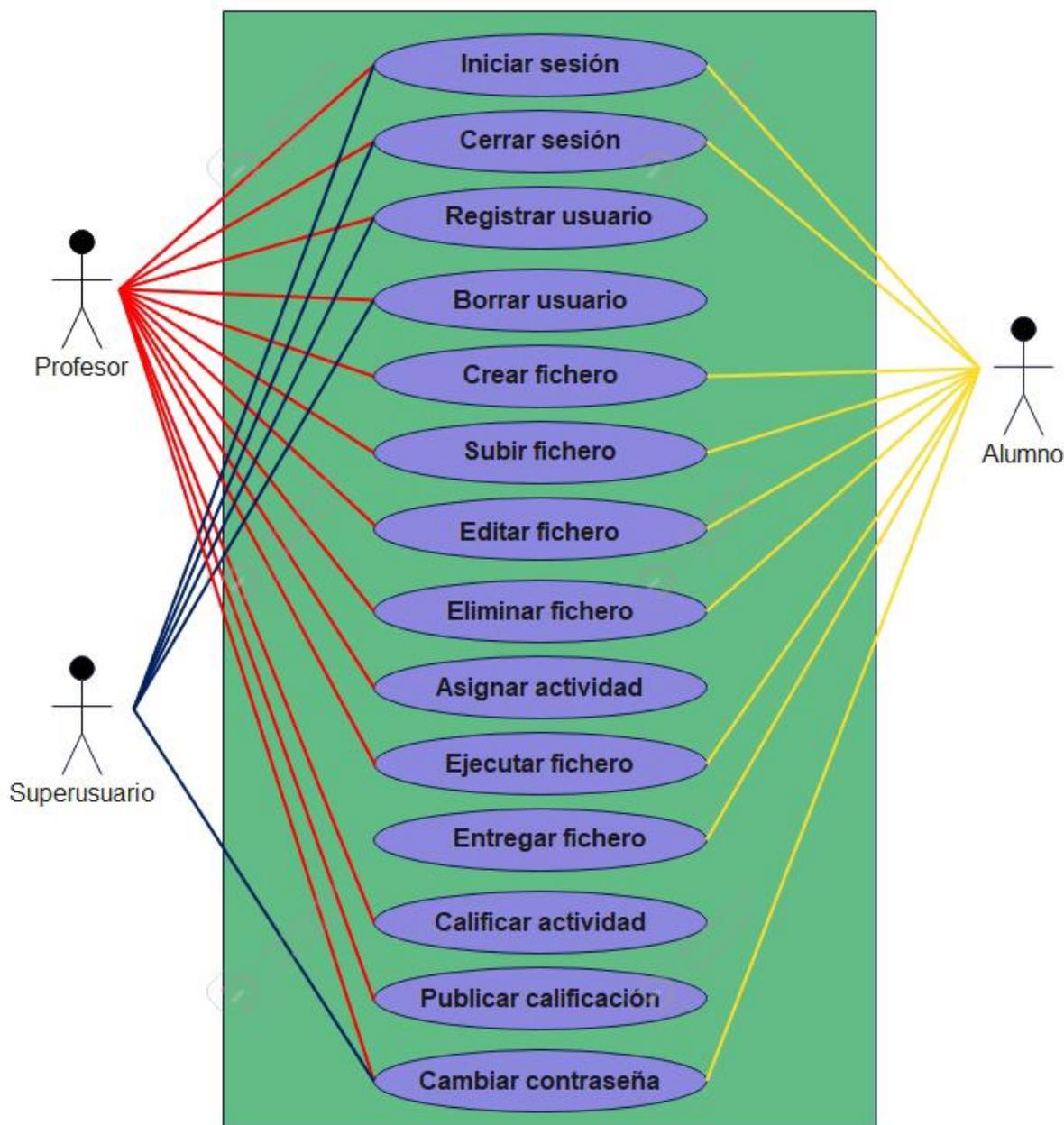


Ilustración 3-1 Diagrama de casos de uso.

3.2.2.2 Reglas de negocio

RN-01	Roles de usuario
Dependencias	RG-02
Descripción	Exisitirán tres tipos de usuarios. Los usuarios con rol de administrador o superusuarios, los que posean rol de profesor y, por último, los que tengan rol de alumno. Las acciones que puedan realizar cada uno vendrán determinado por los permisos que están asociados a su rol.

Importancia	Alta
Prioridad	Alta

Tabla 3-26 Regla de negocio 1 "Roles de usuario".

RN-02	Profesores de un curso
Dependencias	CU-10, CU-13, CU-14
Descripción	En el caso de que se desplieguen varios cursos y haya diferentes profesores por cada curso, cada docente será el responsable de su curso, sin poder intervenir en la creación, asignación o corrección de tareas de los otros cursos.
Importancia	Alta
Prioridad	Alta

Tabla 3-27 Regla de negocio 2 "Profesores de un curso".

3.2.2.3 Requisitos de información

RI-01	Almacenamiento sobre usuarios y proyectos
Dependencias	RG-08
Descripción	<p>La información almacenada sobre los usuarios será:</p> <ul style="list-style-type: none"> • Nombre de usuario • Contraseña • Rol <p>Respecto a los proyectos, la información guardada será:</p> <ul style="list-style-type: none"> • Nombre del proyecto • Propietario • Estado • Curso • Calificación
Importancia	Alta
Prioridad	Alta

Tabla 3-28 Requisito de información 1 "Almacenamiento sobre usuarios y proyectos".

3.2.3 Requisitos no funcionales

3.2.3.1 Fiabilidad

RF-01	Error durante diferentes procesos
Dependencias	RG-07
Descripción	Se buscará que el servicio se restablezca y vuelva a su estado anterior si se producen errores durante los diferentes procesos que tienen lugar en la herramienta.
Importancia	Alta
Prioridad	Alta

Tabla 3-29 Requisito de fiabilidad 1 "Error durante diferentes procesos".

RF-02	Alta disponibilidad
Dependencias	-
Descripción	El servidor debe ser capaz de ofrecer servicio a la totalidad de alumnos que formen parte del curso cuando trabajen de manera simultánea
Importancia	Alta
Prioridad	Alta

Tabla 3-30 Requisito de fiabilidad 2 "Alta disponibilidad".

RF-03	Nombre de proyectos
Dependencias	RG-07
Descripción	No se permitirá la existencia de dos proyectos con el mismo nombre en el directorio de trabajo de un usuario
Importancia	Alta
Prioridad	Alta

Tabla 3-31 Requisito de fiabilidad 3 "Nombre de proyectos".

3.2.3.2 Usabilidad

RU-01	Manual de instalación
--------------	-----------------------

Dependencias	-
Descripción	Se proporcionará un manual de instalación de la herramienta para administradores.
Importancia	Alta
Prioridad	Media

Tabla 3-32 Requisito de usabilidad 1 "Manual de instalación".

RU-02	Manual de uso
Dependencias	-
Descripción	Se proporcionará un manual de uso orientado tanto a profesores como a alumnos.
Importancia	Alta
Prioridad	Media

Tabla 3-33 Requisito de usabilidad 2 "Manual de uso".

RU-03	Indicaciones de uso
Dependencias	CU-10, CU-12, CU-13, CU-14
Descripción	Cada diferente proceso que se lleve a cabo en la aplicación (creación, asignación o calificación de proyectos) irá acompañado de mensajes informativos.
Importancia	Alta
Prioridad	Media

Tabla 3-34 Requisito de usabilidad 3 "Indicaciones de uso".

3.2.3.3 Mantenibilidad

RM-01	Reutilización
Dependencias	-
Descripción	Los proyectos deben poder editarse cuantas veces sean necesarias.
Importancia	Alta

Prioridad	Media
------------------	-------

Tabla 3-35 Requisito de mantenibilidad 1 "Reutilización".

3.2.3.4 Seguridad

RS-01	Control de acceso a la plataforma
Dependencias	RS-02
Descripción	Los usuarios accederán al sistema con la combinación de credenciales formada por nombre de usuario y contraseña.
Importancia	Alta
Prioridad	Alta

Tabla 3-36 Requisito de seguridad 1 "Control de acceso a la plataforma".

RS-02	Control de sesión
Dependencias	RG-01, RN-02, RS-01
Descripción	La sesión de los usuarios caducará cada cierto tiempo, teniendo que iniciar sesión de nuevo para acceder al sistema.
Importancia	Alta
Prioridad	Alta

Tabla 3-37 Requisito de seguridad 2 "Control de sesión".

RS-03	Cambio de contraseña
Dependencias	-
Descripción	Los usuarios registrados podrán cambiar su contraseña si así lo desean.
Importancia	Alta
Prioridad	Alta

Tabla 3-38 Requisito de seguridad 3 "Cambio de contraseña".

RS-04	Control de acceso a recursos
--------------	------------------------------

Dependencias	-
Descripción	Se comprobará los permisos de los usuarios una vez hagan peticiones a direcciones de diferentes recursos internos de la plataforma.
Importancia	Alta
Prioridad	Alta

Tabla 3-39 Requisito de seguridad 4 "Control de acceso a recursos".

3.2.4 Restricciones técnicas

RT-01	Sistema operativo
Dependencias	-
Descripción	El servidor debe ser instalable tanto en sistemas operativos Windows como en Linux
Importancia	Alta
Prioridad	Alta

Tabla 3-40 Restricción técnica 1 "Sistema operativo".

RT-02	Navegador
Dependencias	-
Descripción	Los usuarios deberán poder acceder al servicio con cualquiera de los principales navegadores web que existen.
Importancia	Alta
Prioridad	Alta

Tabla 3-41 Restricción técnica 2 "Navegador".

RT-03	Conexión
Dependencias	-
Descripción	La conexión al servicio deberá ser de tipo TCP/IP sobre el protocolo de aplicación HTTP y dirigida al puerto donde esté abierto ese proceso en el servidor.

Importancia	Alta
Prioridad	Alta

Tabla 3-42 Restricción técnica 3 "Conexión".

3.3 Diseño y funcionamiento del sistema

3.3.1 Introducción

En cuanto a la arquitectura de la plataforma, se deja a continuación un diagrama de paquetes en el que se puede visualizar de forma gráfica la estructura general del sistema.

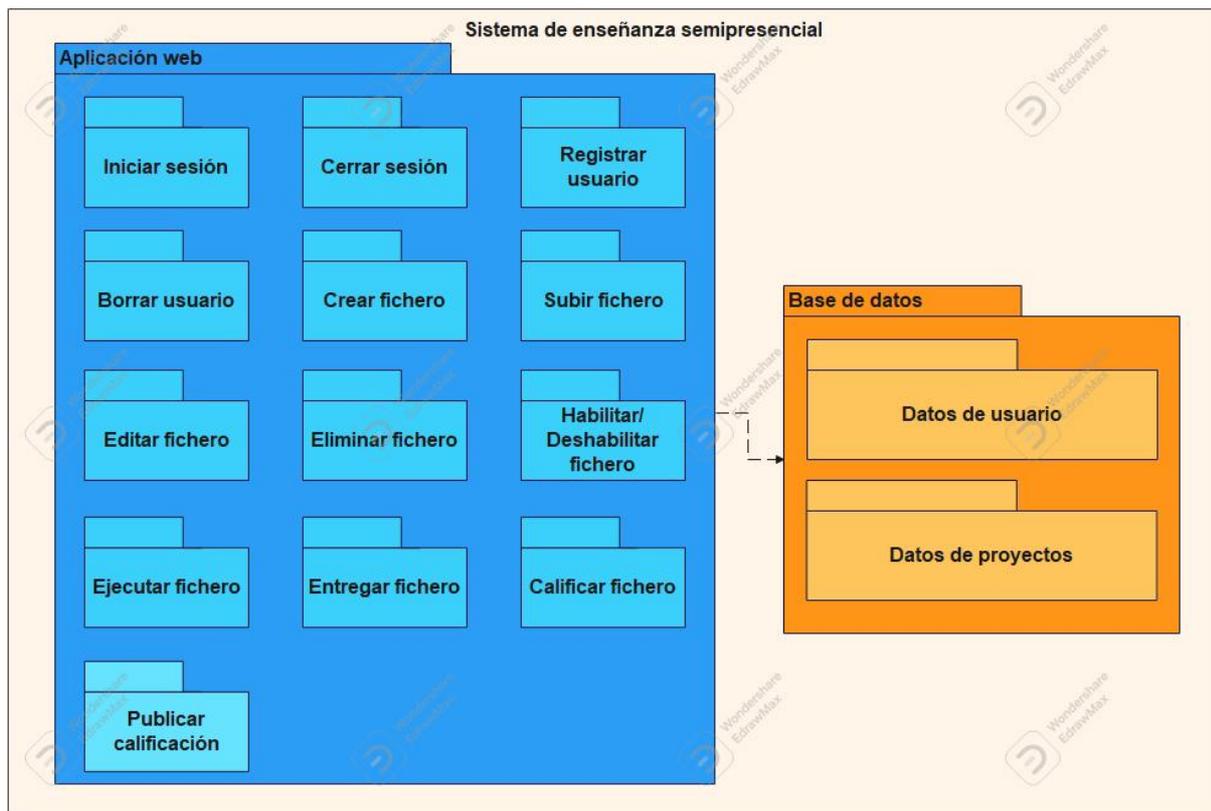


Ilustración 3-2 Diagrama de paquetes.

Respecto a la aplicación, se pueden comentar varios puntos sobre el funcionamiento:

- Se diferenciarán tres tipos de usuarios según su rol:
 - Un superusuario, con todo tipo de permisos y que existirá desde el despliegue de la plataforma,
 - Uno o varios administradores/profesores, los cuales serán creados por el superusuario o por otro administrador y que tendrán privilegios para crear y asignar proyectos y corregirlos posteriormente, y por último,
 - Usuarios de tipo alumno, los cuales serán registrados por el superusuario o por los profesores y dispondrán los permisos necesarios para trabajar con sus proyectos

o los asignados por los profesores.

A continuación, se dispone de un diagrama donde se representa la jerarquía entre usuarios explicada anteriormente.

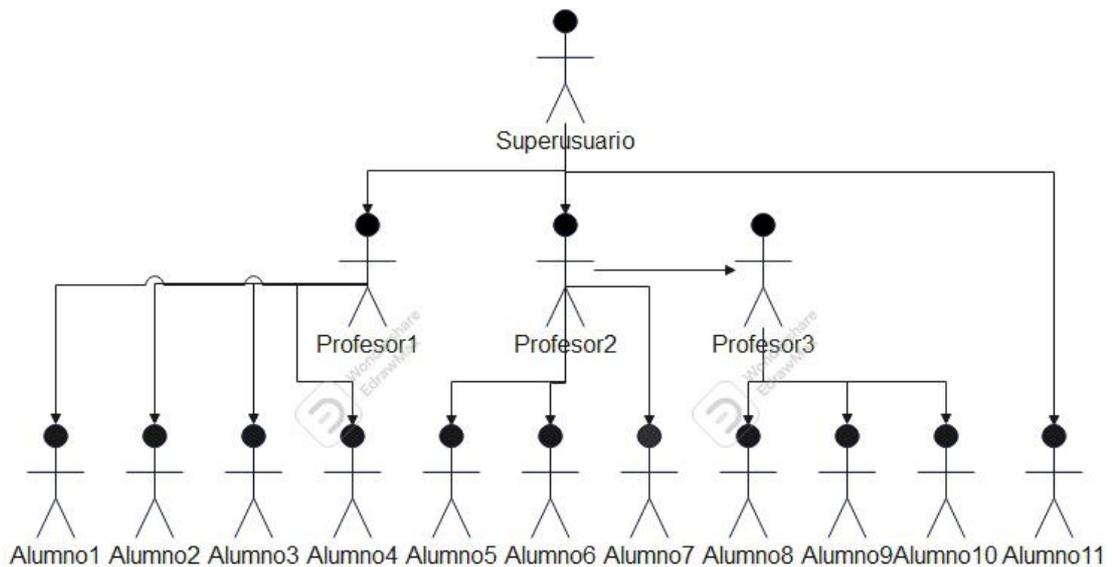


Ilustración 3-3 Diagrama de jerarquía de usuarios.

- En caso de que se quiera desplegar el servicio para varios cursos en una misma instancia de sistema con diferentes profesores por curso, cada profesor solo será capaz de asignar y corregir actividades asociadas a dicho curso, no pudiendo interferir en los otros cursos registrados en la plataforma.
- La información correspondiente a los usuarios (nombre de usuario, contraseña, rol y calificaciones) y a los proyectos (nombre del proyecto, propietario, estado, curso y calificación) se almacenará en una base de datos.

3.3.2 Funcionamiento

En este subapartado se adjuntan diferentes diagramas UML en los que se explican con mayor detalle los diferentes procesos diseñados y que conforman la aplicación.

3.3.2.1 Registro de usuario

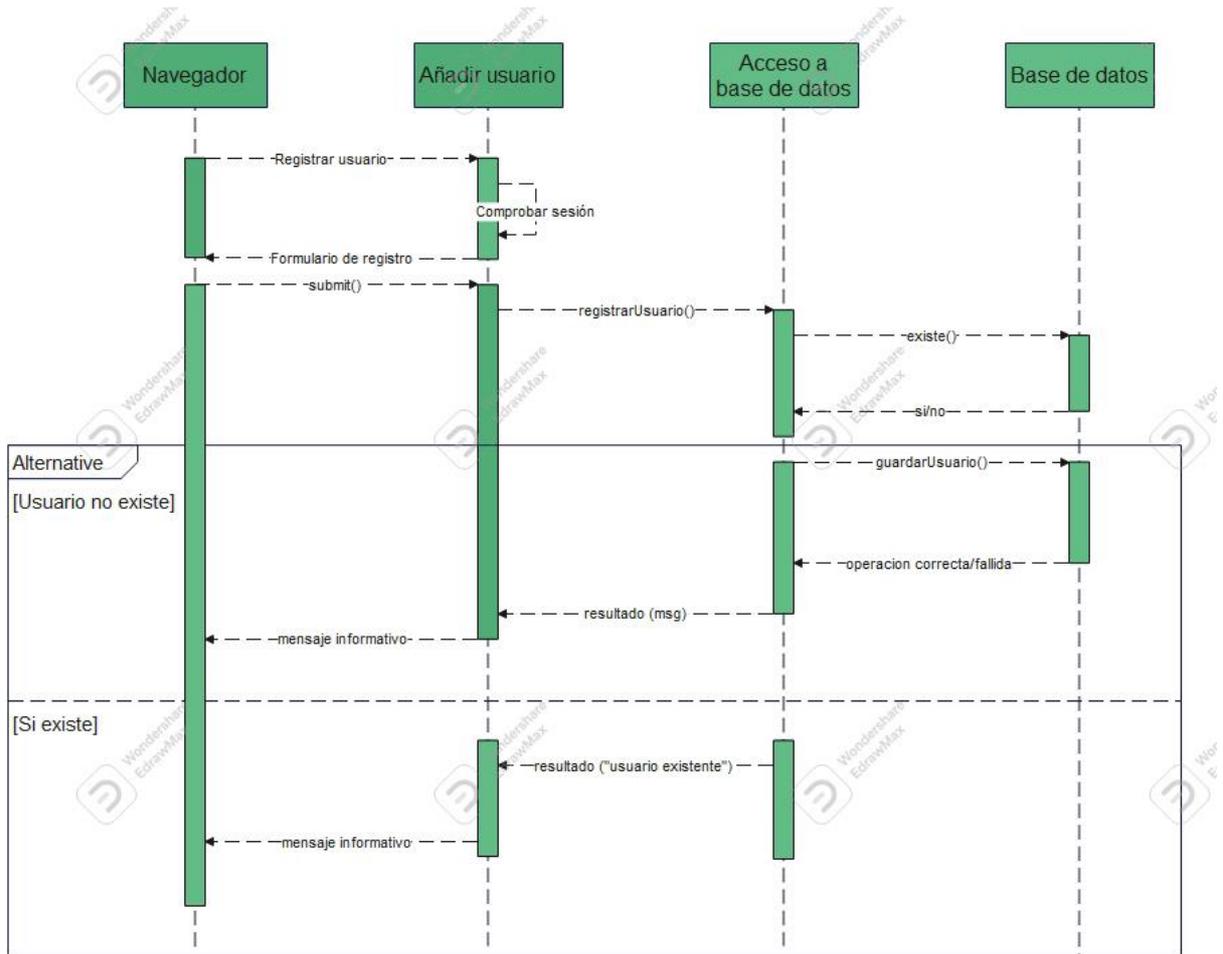


Ilustración 3-4 Diagrama de secuencia "Registro de usuario".

3.3.2.2 Inicio de sesión

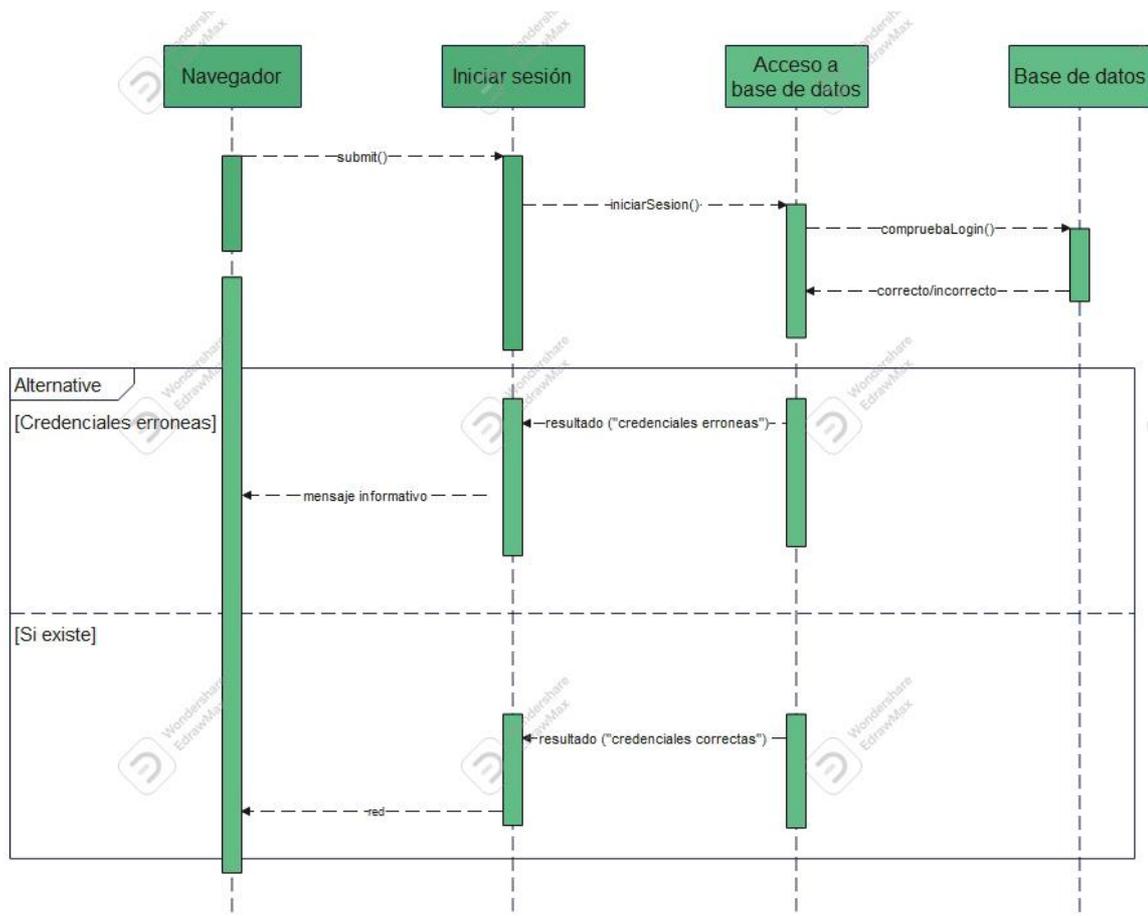


Ilustración 3-5 Diagrama de secuencia "Inicio de sesión".

3.3.2.3 Asignar proyecto

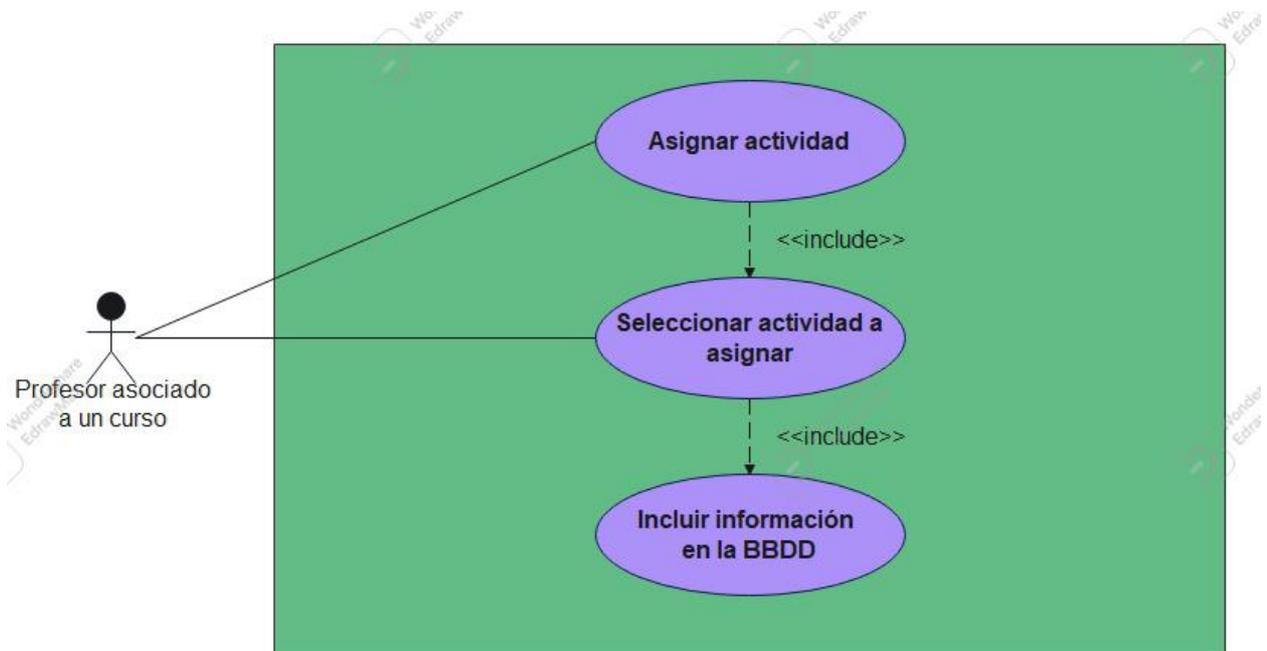


Ilustración 3-6 Caso de uso "Asignar actividad"

3.3.2.4 Calificar actividad

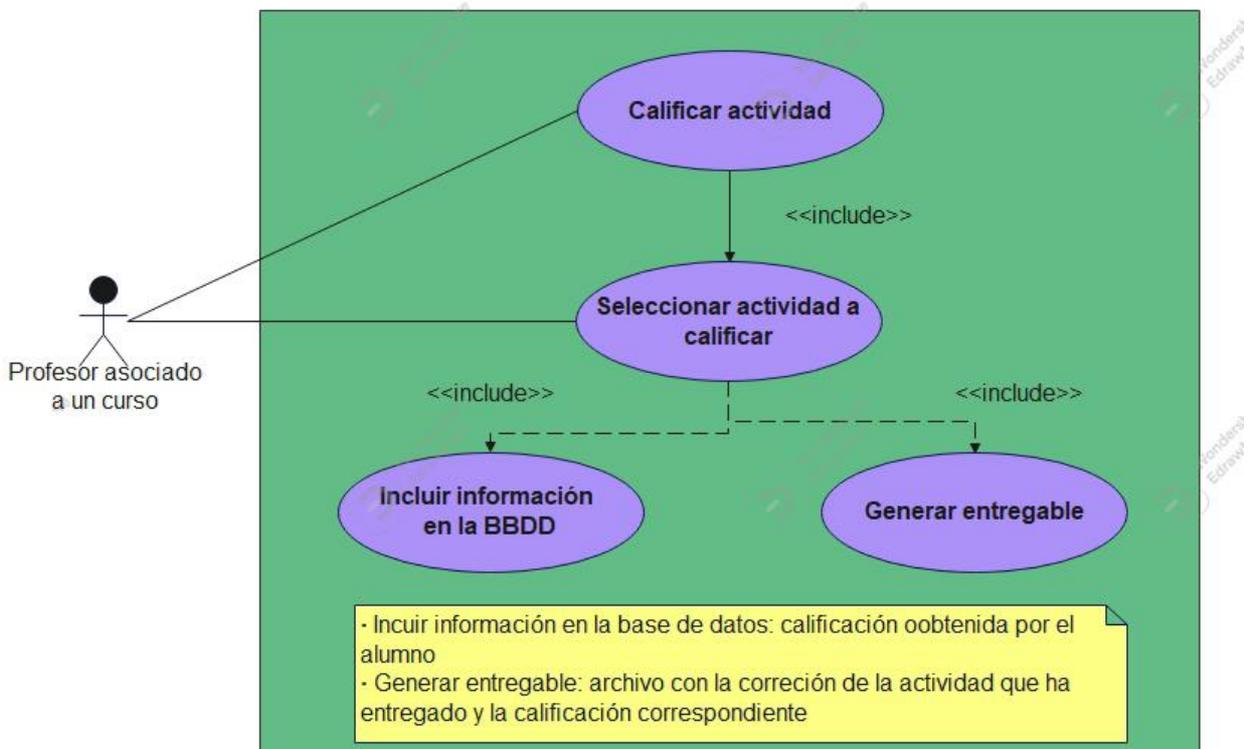


Ilustración 3-7 Caso de uso "Calificar actividad".

3.3.2.5 Ejecutar fichero

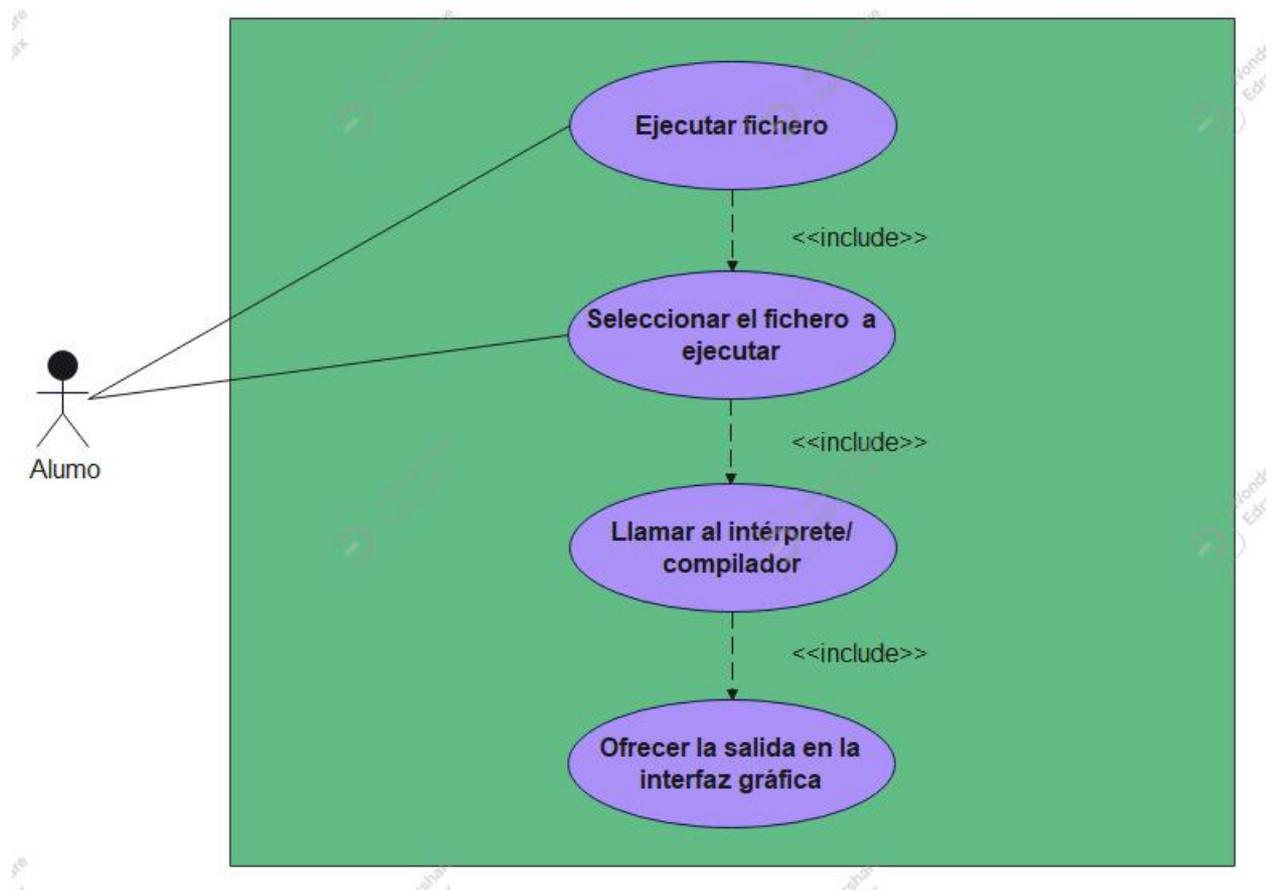


Ilustración 3-8 Caso de uso "Ejecutar fichero".

4 IMPLEMENTACIÓN

“De humanos es errar y de necios permanecer en el error”

- Marco Tulio Cicerón-

Una vez explicado el diseño elaborado para el sistema, podemos pasar a desarrollar cual será la implementación escogida para dicho sistema, explicando los puntos clave que permiten su funcionamiento.

4.1 Introducción

Tras evaluar las necesidades requeridas por los usuarios para este proyecto, se ha llegado a la conclusión que la opción más acertada y que más se ajusta a dichas necesidades es implementar los servicios de JupyterHub [19] combinado de la extensión conocida como nbgrader [20].

Esta solución permite crear una plataforma donde tanto alumnos como profesores podrán desarrollar, ejecutar y compartir código en diferentes lenguajes, además de disponer de un sistema a través del cual se podrá entregar actividades a los alumnos para que sean realizadas, entregadas de vuelta al profesor. Una vez el profesor tenga las entregas de los alumnos, podrá corregir los trabajos, calificarlos y entregar la nota a los alumnos a través del sistema.

Además, el servicio se desplegará en una instancia en el servicio de AWS llamado Amazon EC2 [21], lo cual nos otorgará grandes ventajas como veremos más adelante en este capítulo.

4.2 Jupyter Notebook

4.2.1 Introducción

La primera herramienta a implementar será la aplicación web conocida como Jupyter Notebook. Esta aplicación permite la creación, edición y compartición de documentos, conocidos como *notebooks*, sobre los cuales es posible desarrollar y ejecutar código, incluir texto plano, contenido multimedia y ecuaciones en formato matemático.

Jupyter Notebook es uno de los tres productos pertenecientes al proyecto de código abierto Jupyter. Los otros dos son JupyterLab, el cual se trata de la interfaz de siguiente generación del clásico Jupyter Notebook, y JupyterHub, un servidor sobre el cual se pueden correr múltiples instancias de Jupyter Notebook, permitiendo el uso simultáneo de varios usuarios. Este último producto será el desplegado en el proyecto desarrollado en esta memoria.

Este proyecto surgió en 2014 como la mejora de IPython. Dicho proyecto fue iniciado por los

desarrolladores de software Fernando Pérez y Brian Granger.

El nombre de Jupyter es una referencia a los tres lenguajes de programación que inicialmente soportaba esta herramienta, siendo dichos lenguajes Julia, Python y R. Actualmente, Jupyter Notebook permite integrar en su núcleo una gran variedad de kernels, haciendo posible que se pueda programar en diferentes lenguajes de programación aparte de los tres mencionados anteriormente.

En el repositorio de Jupyter en Github [22], se especifica una lista completa con todos los kernels disponibles para usar en esta aplicación.

La forma tradicional de probar código es a través de la terminal empleando un intérprete o ejecutando un archivo de código previamente compilado (dependiendo del lenguaje de programación utilizado). Jupyter Notebook enmascara todo este proceso por medio de su interfaz web, en la cual existen celdas donde el usuario puede escribir el código que desee y ejecutarlo posteriormente a través de un botón ubicado en la interfaz. Todo esto permite a los usuarios con poca o nula experiencia previa programando simplificar el proceso para desarrollar y probar el código, facilitando sus primeros pasos en el mundo de la programación.

A partir de ahora en adelante, se estudiará el producto de JupyterHub, que es la opción escogida para este trabajo.

4.2.2 Distribuciones de JupyterHub

Actualmente, JupyterHub cuenta con dos tipos de distribuciones:

- The Littlest JupyterHub [23]: es una versión evolucionada del clásico JupyterHub. Es la opción recomendada para un despliegue con una poca cantidad usuarios (entre 0 y 100) haciendo uso del servicio simultáneamente.
- Zero to JupyterHub with Kubernetes [24]: si el proyecto necesita escalabilidad, consumo elevado de recursos, reserva dinámica de servidores y acceso para un gran número de usuarios, Kubernetes permite desplegar JupyterHub con las anteriores características.

Ambas distribuciones cuentan con una buena documentación donde se detalla paso a paso cómo instalar cada una y cómo configurar diferentes aspectos del servicio. Para este trabajo se usará la distribución The Littlest JupyterHub.

4.2.3 Arquitectura de JupyterHub

JupyterHub está compuesto por diferentes módulos que permiten su funcionamiento. Esencialmente, está compuesto por tres módulos diferentes, trabajando en conjunto para ofrecer los servicios deseados. Estos módulos son los siguientes:

- Hub: es el núcleo de JupyterHub, manejando todo los procesos internos que ocurren en el servicio. Se encarga de manejar las cuentas de usuarios, el proceso de autenticación, las bases de datos y las instancias de Jupyter Notebook creadas para cada usuario.
- Proxy: gracias a él, las peticiones HTTP realizadas desde el navegador del cliente son redirigidas al modulo Hub del servidor para que ejecute el proceso solicitado por el usuario.
- Single-User Notebook Server: es la instancia que se corre en el servidor para que cada usuario que inicie sesión disponga de una interfaz gráfica única de Jupyter Notebook. Estas instancias son creadas por un objeto conocido como Spawner.

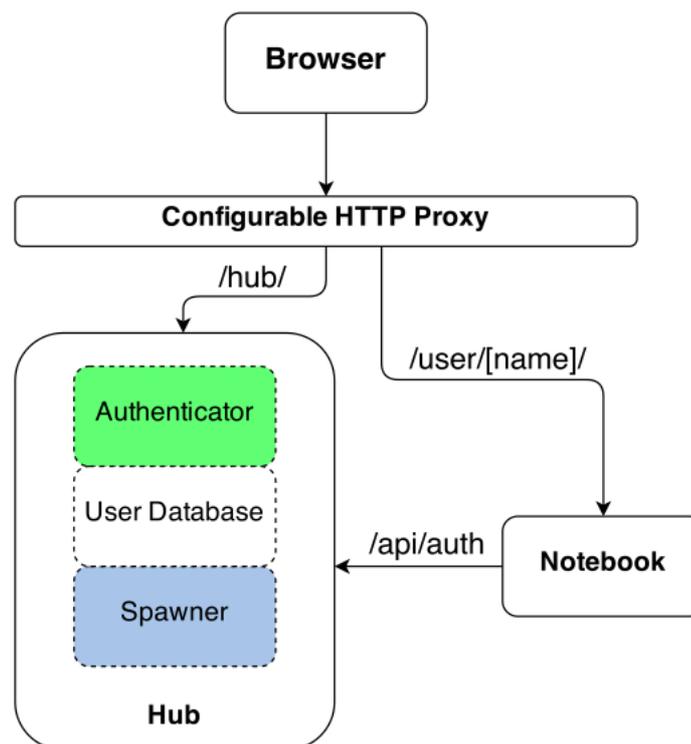


Ilustración 4-1 Arquitectura JupyterHub

En la figura anterior, se puede ver un diagrama en el cual se representa a alto nivel las interacciones de los módulos o subsistemas de Jupyter entre sí. Estos procesos se pueden describir de la siguiente forma:

- El Hub arranca el proxy, el cual escucha las peticiones que apuntan a la dirección pública del sistema, a la dirección privada y a la dirección de *loopback* o *localhost*, ambas en el puerto 8000 por defecto (*http://ip_privada:8000*, *http://ip_publica:8000* o *http://localhost:8000*).
- El proxy reenvía estas peticiones que llegan del cliente al Hub (que se encuentra detrás del proxy en */hub*).
- El Hub maneja el inicio de sesión por medio del submódulo conocido como Authenticator, el cual devuelve el nombre de usuario si las credenciales son correctas.
- Una vez se inicie sesión y el Hub tenga el nombre de usuario en concreto, por medio de otro submódulo, llamado Spawner, arrancará el servidor de único usuario de Jupyter Notebook. Este servidor, por defecto, se iniciará en la misma máquina donde está instalado JupyterHub.
- El Hub configurará al proxy para que las peticiones del cliente con sesión activa sean redirigidas a */user/{username}*. Por lo tanto, todas las acciones dentro del Sistema serán se enviarán al Single-User Notebook server.

Al ser un software de código abierto, todos los módulos son altamente configurables, gracias también a la gran comunidad que posee Jupyter y que colabora con el proyecto. En la documentación de la página oficial se encuentran guías para modificar estos módulos según convenga.

4.3 nbgrader

4.3.1 Introducción

nbgrader es una herramienta o, más concretamente, una extensión de código abierto de Jupyter Notebook que amplía sus propios servicios, permitiendo crear y corregir actividades diseñadas por usuarios en los notebooks. Esto es posible gracias a que nbgrader añade a la clásica interfaz gráfica de Jupyter la funcionalidad necesaria para habilitar al usuario crear cuadernos evaluables con ejercicios de código o de respuesta libre.

En líneas generales, nbgrader implementa unos notebooks de Jupyter con una metadata específica, una interfaz específica de instructor y otra de estudiante y, por último, un sistema para intercambiar y corregir actividades.

nbgrader ha sido desarrollada por la investigadora científica Jessica Hamrick, tratando de mejorar la enseñanza usando su combinación junto con Jupyter. Su experiencia como estudiante y como docente fue lo que le llevo a iniciar este proyecto.

Para este trabajo, se instalará nbgrader para ser usado junto a JupyterHub y conseguir así crear un sistema que se acerque a un LMS, totalmente enfocado a la programación.

4.3.2 Arquitectura de nbgrader

4.3.2.1 Estructura de directorios

En nbgrader y desde el punto de vista del instructor, los ficheros de curso se almacenan en diferentes directorios según la fase en la que se encuentren. La estructura general de directorios viene dada por `{fase_nbgrader}/{id_estudiante}/{id_asignacion}/{notebook}.ipynb`, donde *{fase_nbgrader}* puede ser:

- *source/*: en este directorio se almacenan los archivos en bruto generados por el docente. Para este directorio no se incluye *{id_estudiante}* en el path de directorios.
- *release/*: contiene los notebooks que han sido generados y están listos para asignarse a los estudiantes. Las versiones de notebooks ubicadas aquí ocultan los hidden-tests y bloquean la edición de las celdas marcadas de solo lectura. Como sucede en *source/*, el directorio *{id_estudiante}* no forma parte del path.
- *submitted/*: este directorio almacena los notebooks que ya han sido entregados por los alumnos.
- *autograded/*: contiene los notebooks que han sido ejecutados por la herramienta de autocorrección.
- *feedback/*: a diferencia del resto de directorios, aquí se almacenan ficheros con extensión `.html` con la ejecución del notebook, la puntuación obtenida y posibles comentarios añadidos por el instructor con la corrección manual.

Si estamos usando un servidor compartido como para usar Jupyter, como es el caso de JupyterHub, debe existir un sistema de ficheros de intercambio para que los profesores y alumnos puedan entregarse entre ellos las actividades que se vayan realizando. El directorio padre tendrá en su interior el siguiente Sistema de directorios:

- *outbound/*: contiene las versiones lanzadas del profesor a los estudiantes. Su path de directorios internos es `outbound/{id_asignacion}/{notebook}.ipynb`.
- *inbound/*: almacena las versiones entregadas por los alumnos. Se organiza como `inbound/{id_estudiante}+{id_asignacion}+{timestamp}+{random_string}/{notebook}.ipynb`

b

- *feedback/*: aquí se guardan las puntuaciones de las actividades de los alumnos. La estructura del directorio es *feedback/{hash}.html* donde *{hash}* es la codificación del contenido del notebook y el timestamp.

4.3.2.2 Extensiones de nbgrader

nbgrader tiene diferentes extensiones que pueden ser activadas o desactivadas para dotar de mayor o menor funcionalidad a la herramienta. Estas extensiones son:

- *generate_assignment*: es la extensión dedicada para que el instructor pueda transformar los notebooks brutos en versiones entregables a los alumnos.
- *release_assignment*: permite lanzar a los estudiantes las versiones generadas previamente con la extensión anterior.
- *fetch_assignment*: esta extensión sirve para que los alumnos puedan recopilar los notebooks lanzados por el profesor para una asignación.
- *submit*: su función es entregar los notebooks realizados por el alumno.
- *collect*: se utiliza para copiar los ficheros entregados por los alumnos del directorio de intercambio al directorio submitted, donde el instructor puede visualizar las entregas realizadas.
- *autograde*: ejecuta los notebooks entregados, generando notebook en archive html con la puntuación correspondiente a las celdas que se pueden autocorregir.
- *formgrader*: es el panel gráfico de profesor en JupyterHub. Con él se podrá hacer uso de extensiones como *generate_assignment*, *release_assignment*, *collect* o *autograde* sin hacer uso de la línea de comandos.

4.4 Amazon EC2

Se conoce como Amazon Web Services (AWS de forma abreviada) a la plataforma de nube pública de la conocida multinacional Amazon.com. AWS ofrece más de 200 servicios *cloud computing*, como por ejemplo almacenamiento y gestión de datos, despliegue de servidores virtuales, servicios de correo o herramientas para desarrolladores, aparte de muchos más.

Amazon Web Services fue oficialmente lanzado en el año 2006, y rápidamente fue ganando popularidad hasta llegar a ser lo que conocemos hoy en día. Grandes empresas como Netflix, Coca-Cola, Siemens o NASA han decidido migrar gran parte de sus servicios a AWS, gracias a la ventajosa política de Amazon de “paga por lo que usas”.

AWS posee numerosos *datacenters* [25], ubicados alrededor de todo el mundo. Esto hace posible que las empresas puedan desplegar los servicios que deseen en la zona geográfica que más les convenga.

Entre estos servicios que ofrece AWS, se encuentra Amazon Elastic Compute Cloud (EC2 abreviado). EC2 es un servicio web que proporciona a los usuarios alquilar por horas máquinas virtuales en la nube. Estos usuarios pueden diseñar su servidor al gusto y según sus necesidades, pudiendo elegir el sistema operativo, las prestaciones de memoria y procesador de la máquina y la configuración de red, todo ello desde una interfaz gráfica con un aspecto bastante simple y limpio.

Para alojar el servidor de JupyterHub, se ha decidido usar EC2. Esto nos proporciona grandes ventajas, como no tener que pagar por el hardware de un servidor físico, evitar el mantenimiento y el *monitoring* de dicho servidor, no tener que configurar las opciones de red y, sobre todo, pagar únicamente bajo

demanda, es decir, por tiempo de computación.

En nuestro caso, podremos aprovecharnos del plan gratuito que ofrecen desde AWS y desplegar durante un total de 750 horas de computo una instancia.

4.5 Instalación y configuración de los servicios

4.5.1 Configuración de Amazon EC2

Para comenzar la configuración de EC2, el primer paso a dar es crear una cuenta en AWS (<https://portal.aws.amazon.com/billing/signup#/start/email>). Para ello nos pedirán una dirección de correo y el nombre de usuario del usuario *root* de la consola.

En los siguientes pasos, se nos pedirá información de contacto, además de retenemos 1USD durante un plazo de 3 a 5 días como método de verificación de identidad. Finalmente, se nos mandará un código de confirmación al correo, que habrá que introducir en la web de AWS.

Realizados estos pasos, accederemos a la consola de AWS. Será momento entonces de cambiar la región a París. Esto hará que las instancias EC2 se desplieguen en París, reduciendo así la latencia que podría existir si por ejemplo la instancia se desplegara en Norte América o Asia.

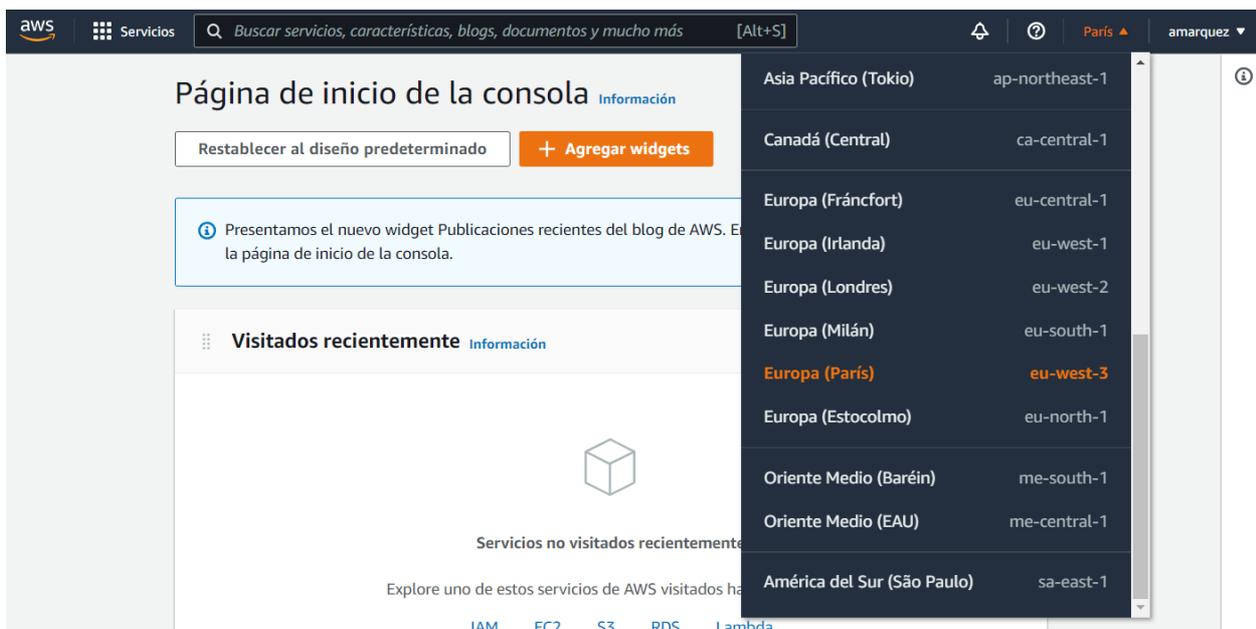


Ilustración 4-2 Regiones de AWS

A continuación, el siguiente paso a dar es buscar “EC2” en la barra de la parte superior. Tras hacer clic en EC2, se nos abrirá dicho servicio. En el menú que se muestra en la parte derecha, habrá que seleccionar Instancias y a continuación Lanzar instancias.

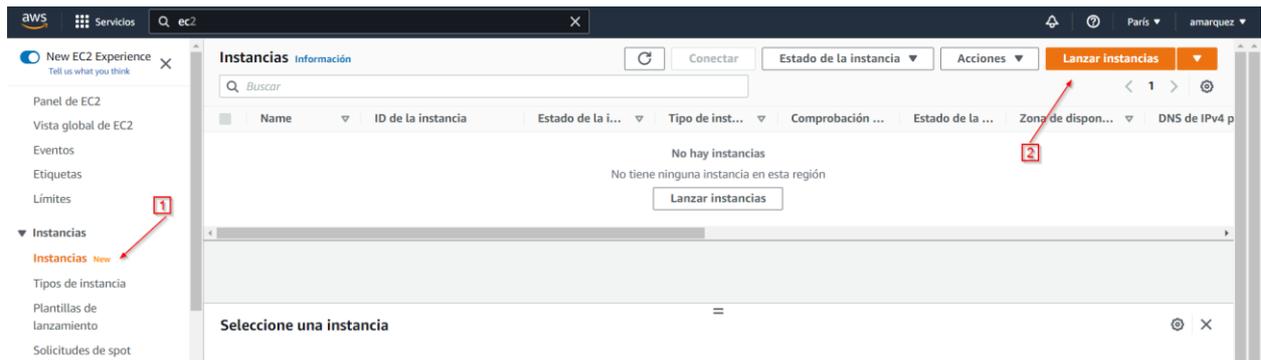


Ilustración 4-3 Menú EC2: “Instancias” 1

Tras esto, se abrirá un menú desde el que se podrá configurar la máquina virtual a desplegar. La opciones escogidas son las siguientes:

- Nombre del servidor: JupyterHub
- Sistema Operativo: Ubuntu Server 18.04 LTS x86

Inicio rápido

Amazon Linux

Ubuntu

Windows

Red Hat

SUSE Linux

[Buscar más AMI](#)

Incluidas las AMI de AWS, Marketplace y la comunidad

Amazon Machine Image (AMI)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type

ami-03214270b6681ff5e (64 bits (x86)) / ami-0a5e546a209458e26 (64 bits (Arm))

Virtualización: hvm Habilitado para ENA: true Tipo de dispositivo raíz: ebs

Apto para la capa gratuita

Descripción

Canonical, Ubuntu, 18.04 LTS, amd64 bionic image build on 2022-06-10

Arquitectura

64 bits (x86)
▼

ID de AMI

ami-03214270b6681ff5e

Proveedor verificado

Ilustración 4-4 Configuración instancia: "Sistema operativo".

- Tipo de instancia: t2.micro

▼ **Tipo de instancia** Información

Tipo de instancia

t2.micro Apto para la capa gratuita ▼ [Comparar tipos de instancias](#)

Familia: t2 1 vCPU 1 GiB Memoria

Bajo demanda Linux precios: 0.0132 USD por hora

Bajo demanda Windows precios: 0.0178 USD por hora

Ilustración 4-5 Configuración de instancia: "Tipo de instancia".

- Par de claves: crear un par nuevo cifradas por RSA y almacenadas en un archivo *.pem*

Crear par de claves ✕

Los pares de claves le permiten conectarse a la instancia de forma segura.

Escriba el nombre del par de claves a continuación. Cuando se lo pida, almacene la clave privada en una ubicación segura y accesible de su equipo. **Lo necesitará más adelante para conectarse a la instancia.** [Más información](#)

Nombre del par de claves

clave1

El nombre puede incluir hasta 255 caracteres ASCII. No puede incluir espacios al principio ni al final.

Tipo de par de claves

RSA
Par de claves públicas y privadas cifradas por RSA

ED25519
Los pares de claves privadas y públicas cifradas ED25519 (no se admite para instancias de Windows)

Formato de archivo de clave privada

.pem
Para usar con OpenSSH

.ppk
Para usar con PuTTY

Cancelar **Crear par de claves**

Ilustración 4-6 Configuración de instancia: "Par de claves".

- Configuración de red: VPC predeterminado (creado por AWS), subred sin preferencia, habilitar asignación automática de IP pública, crear grupo de seguridad y permitir tráfico SSH, HTTP y HTTPS desde internet.

▼ Configuraciones de red Información Editar

Red **Información**
vpc-0d6afe1a3f67a06a7

Subred **Información**
Sin preferencia (subred predeterminada en cualquier zona de disponibilidad)

Asignar automáticamente la IP pública **Información**
Habilitar

Firewall (grupos de seguridad) Información
Un grupo de seguridad es un conjunto de reglas de firewall que controlan el tráfico de la instancia. Agregue reglas para permitir que un tráfico específico llegue a la instancia.

Crear grupo de seguridad Seleccionar un grupo de seguridad existente

Crearemos un nuevo grupo de seguridad denominado "launch-wizard-1" con las siguientes reglas:

- Permitir el tráfico de SSH desde Cualquier lugar
0.0.0.0/0
Ayuda a establecer conexión con la instancia
- Permitir el tráfico de HTTPs desde Internet
Para configurar un punto de enlace, por ejemplo, al crear un servidor web
- Permitir el tráfico de HTTP desde Internet
Para configurar un punto de enlace, por ejemplo, al crear un servidor web

Ilustración 4-7 Configuración de instancia: "Configuraciones de red".

Completados estos pasos, es momento de lanzar la instancia pulsando sobre el botón dedicado para ello, ubicado en la parte derecha de la pantalla. Si se han completado todos los pasos correctamente, se nos debería redirigir de vuelta al menú de Instancias (Ilustración 4-2) y podríamos ver nuestra máquina virtual como en la siguiente figura.

Instancias (1) Información								
<input type="text" value="Buscar"/> < 1 >								
<input type="checkbox"/>	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación ...	Estado de la ...	Zona de dispon...	DNS de IPv4
<input type="checkbox"/>	JupyterHub	i-019d25502c12c33ea	En ejecución	t2.micro	2/2 comprobación	Sin alarmas	eu-west-3c	ec2-35-180-1

Ilustración 4-8 Menú EC2: "Instancias" 2

Llegados a este punto, ya tenemos la máquina virtual con sistema operativo Ubuntu corriendo en la infraestructura de AWS. La siguiente tarea a realizar será instalar JupyterHub

4.5.2 Instalación y configuración de JupyterHub

Como se ha mencionado en el punto 4.2.2, la distribución de JupyterHub elegida para este caso es The Littlest JupyterHub. Siguiendo la documentación oficial de la herramienta, habrá que realizar los siguientes pasos:

1. Ubicar el archivo `.pem`, generado por AWS a la hora de crear la instancia, en el directorio donde se ejecutará SSH.
2. Abrir una conexión SSH contra la instancia de AWS que se acaba de desplegar.

```
ssh -i "clave1.pem" ubuntu@ec2-35-180-114-132.eu-west-3.compute.amazonaws.com
```

“ubuntu” hace referencia al usuario con el que accedemos al servidor y “ec2-35-180-114-132.eu-west-3.compute.amazonaws.com” es el nombre de host de la instancia

3. Actualizar los paquetes y dependencias del sistema.

```
sudo apt-get update
```

4. Instalar los paquetes de python3, python3-dev, curl y git.

```
sudo apt install python3 python3-dev git curl
```

5. Instalar TheLittlest JupyterHub y escoger el username de la cuenta del administrador (sustituyéndolo por <admin-user>).

```
curl -L https://tljh.jupyter.org/bootstrap.py | sudo -E python3 - --admin <admin-user -name>
```

6. Establecer la clave para la cuenta del administrador accediendo al sistema a través del navegador (*http://35.180.114.132*).

Ilustración 4-9 Menú JupyterHub: "Inicio de sesión".

Si se han realizado estos pasos sin ningún inconveniente, se puede decir que se ha instalado satisfactoriamente el servidor de JupyterHub en la instancia de Amazon EC2.

Para hacer cambios en la configuración JupyterHub se provee un comando para ejecutar con las diferentes opciones que nos proporciona la herramienta para personalizarla como se desee. El comando en cuestión es *tljh-config*.

El comando se puede usar de dos formas:

- Por medio de la interfaz gráfica: se inicia sesión con una cuenta de administrador y se abre una terminal.
- Por medio de una conexión SSH contra el servidor: se accede al servidor a través de la cuenta *root*, se llama directamente a */opt/tljh/bin/tljh-config*.

Para usar establecer una cierta configuración, hay que usar el siguiente comando:

```
sudo tljh-config set <property-path> <value>
```

Donde <property-path> es el path, separado por puntos, de la propiedad donde se van a realizar los

cambios, y `<value>` es el valor a establecer en dicha propiedad.

Mediante ese comando se podrán configurar diferentes aspectos del sistema como son usuarios y listas de usuarios, los puertos que utiliza el servicio o el authenticator, entre otros.

Para ver la configuración actual sería por medio de:

```
sudo tljh-config show
```

Cada vez que se realice una modificación en la configuración actual de la plataforma, hay que reiniciar el servicio, lo cual se puede hacer de manera sencilla a través del comando:

```
sudo tljh-config reload
```

Para ver todas las opciones de `tljh-config`, se puede usar la bandera `--help`:

```
sudo tljh-config --help
```

Cuando se ejecute el comando `tljh-config` para realizar una modificación en la configuración, realmente lo que se estará modificando es el archivo `/opt/tljh/config/config.yam`. Modificar el archivo `.yam` directamente no se recomienda, por lo que para realizar cualquier tipo de cambio en la configuración, se aconseja usar exclusivamente el mencionado comando.

Para nuestro caso, no habrá que realizar ningún tipo de configuración, ya que la que trae por defecto en su instalación JupyterHub es válida para alcanzar el objetivo de este trabajo.

4.5.3 Instalación y configuración de nbgrader

Una vez instalado JupyterHub, se puede pasar a instalar nbgrader. Para ello, habrá que abrir una terminal en la interfaz gráfica desde la cuenta del administrador e introducir los comandos que se describen a continuación:

1. Instalar la versión 0.7.1 de nbgrader.

```
sudo -E pip install nbgrader==0.7.1
```

2. Instalar las extensiones de nbgrader para todos los usuarios del sistema.

```
sudo -E jupyter nbextension install --sys-prefix --py nbgrader --overwrite
sudo -E jupyter nbextension enable --sys-prefix --py nbgrader
sudo -E jupyter serverextension enable --sys-prefix --py nbgrader
```

3. Deshabilitar la extensión de Formgrader para el sistema.

```
sudo -E jupyter nbextension disable --sys-prefix formgrader/main --section=tree
sudo -E jupyter serverextension disable --sys-prefix nbgrader.server_extensions.
formgrader
```

4. Deshabilitar la extensión de Formgrader para el sistema.

```
sudo -E jupyter nbextension enable --user formgrader/main --section=tree
```

```
sudo -E jupyter serverextension enable --user nbgrader.server_extensions.formgrader
```

5. Reiniciar el servidor del administrador desde el Control panel de la interfaz gráfica (pulsar sobre *Stop server* y seguidamente sobre *Start server*).

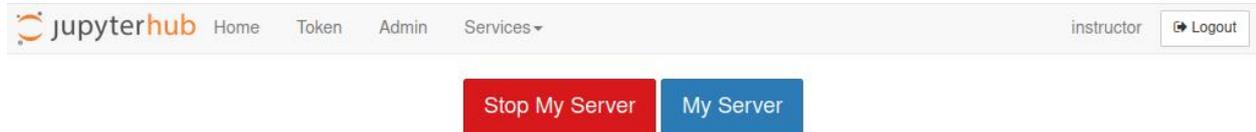


Ilustración 4-10 Menú JupyterHub: "Control Panel".

6. Crear el siguiente directorio y asignarles todos los permisos

```
mkdir /srv/nbgrader
mkdir /srv/nbgrader/exchange
chmod 777 /srv/nbgrader
chmod 777 /srv/nbgrader/exchange
```

Completados estos pasos, la extensión nbgrader estará instalada en el servidor de JupyterHub y se puede pasar a configurar.

Desde nbgrader, nos presentan en su documentación tres escenarios diferentes que se podrán configurar para usar en JupyterHub. Según las necesidades del despliegue, se aconseja implementar uno u otro, ya que la dificultad de su configuración varía según el escenario. El escenario escogido a implementar en este trabajo es el siguiente:

- Un curso y un profesor: para este caso de uso, solo tendremos una clase con una única cuenta de profesor. Suponiendo que el nombre de usuario del profesor es *instructor* y el curso se llama *curso*, se deberá tener un directorio tal que */home/instructor/curso*. Además, habrá que tener un archivo de configuración, */home/jupyter-instructor/nbgrader_config.py*, con el siguiente contenido:

```
c = get_config()
c.CourseDirectory.root = 'home/jupyter-instructor/curso'
```

De esta forma, el módulo de Formgrader estaría corriendo en la cuenta del profesor. Gráficamente, la estructura sería:

One Class, One Instructor

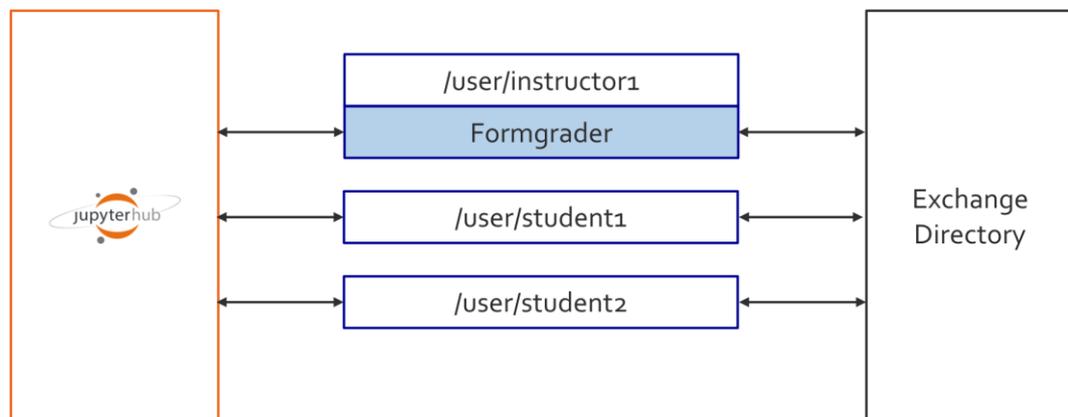


Ilustración 4-11 Escenario "Una clase y un profesor"

5 USO DE LA APLICACIÓN

“Construimos demasiados muros y no suficientes puentes.”

- Isaac Newton-

Comentada la solución elegida para llevar a cabo el diseño de la aplicación realizado, se pasa ahora, en este capítulo, a explicar las funcionalidades o posibles usos que ofrece dicha herramienta.

5.1 Inicio de sesión

Para iniciar sesión, se deberá acceder a la aplicación introduciendo en la barra de búsqueda del navegador una de las siguientes opciones:

- Si estamos en el propio servidor: <http://localhost:8000>
- Si estamos en la misma red privada del servidor: http://IP_privada:8000
- Si estamos en otra red diferente o queremos acceder a través de su IP pública: http://IP_publica:8000

Al estar trabajando con una instancia de Amazon EC2, accederemos por su IP pública.

Una vez se introduzca la dirección, se llegará a una web donde se piden las credenciales para iniciar sesión, con un aspecto como se muestra en la figura de a continuación:

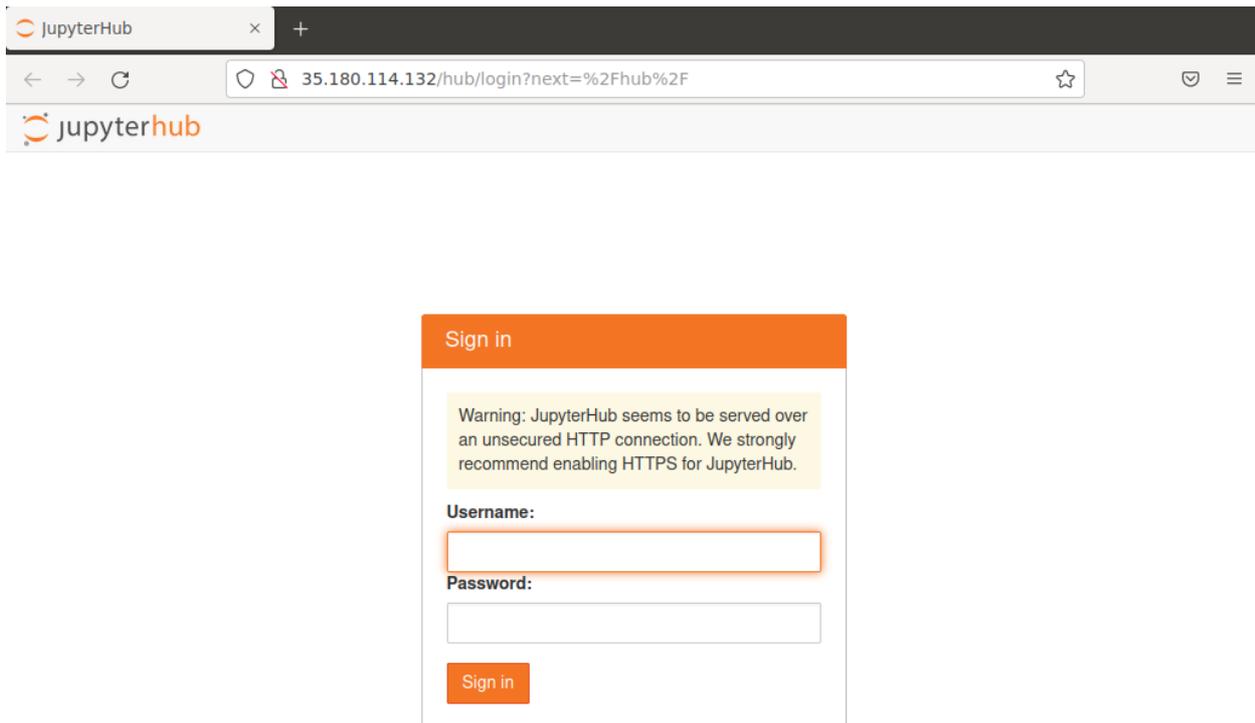


Ilustración 5-1 Menú JupyterHub: "Inicio de sesión".

Si las credenciales introducidas son correctas, el servidor creará un Single-User Notebook Server y nos llevará a la página principal del mismo.

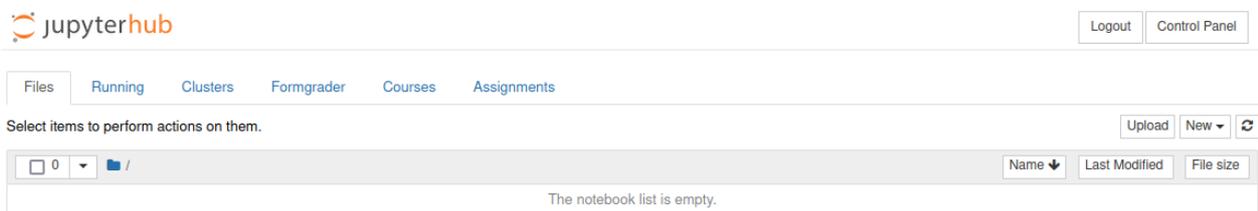


Ilustración 5-2 Menú JupyterHub: "Página principal".

5.2 Gestión de usuarios

Un usuario con permisos de administrador podrá acceder al panel de control de administración a través de *Control Panel* > *Admin*. Desde ese menú, el administrador podrá añadir, editar o eliminar usuarios, parar el servidor de los usuarios que tengan sesión activa en ese preciso momento en JupyterHub o parar el sistema por completo.

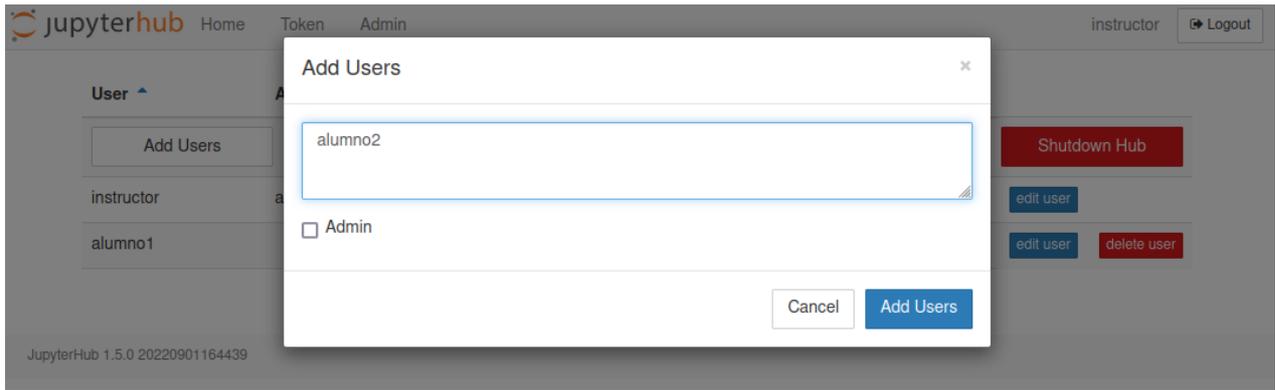


Ilustración 5-3 Menu JupyterHub: "Añadir usuarios".

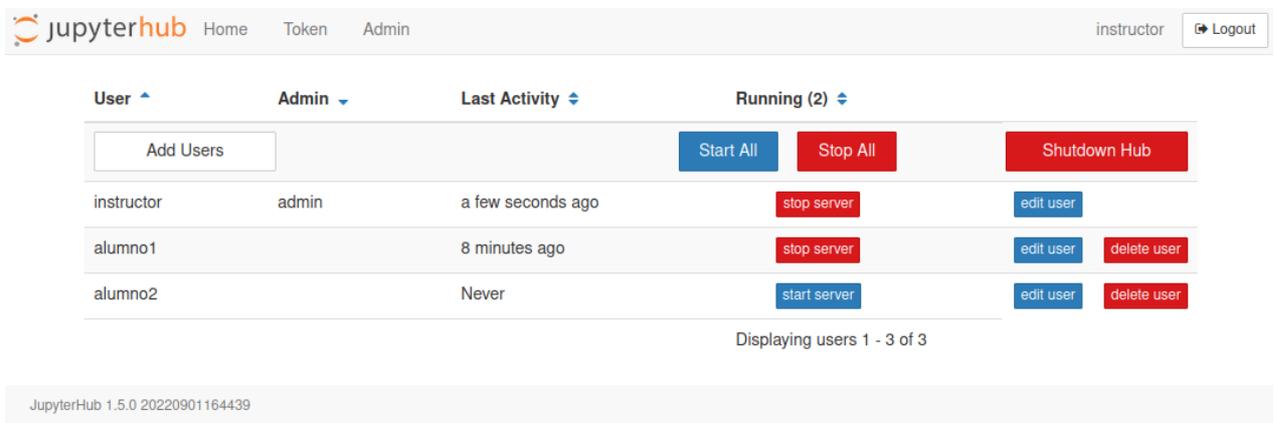


Ilustración 5-4 Menú JupyterHub: "Administrador".

5.3 Uso de notebooks

Para crear un nuevo Notebook basado en Python, hay que pulsar sobre los botones que se muestran en la siguiente figura.

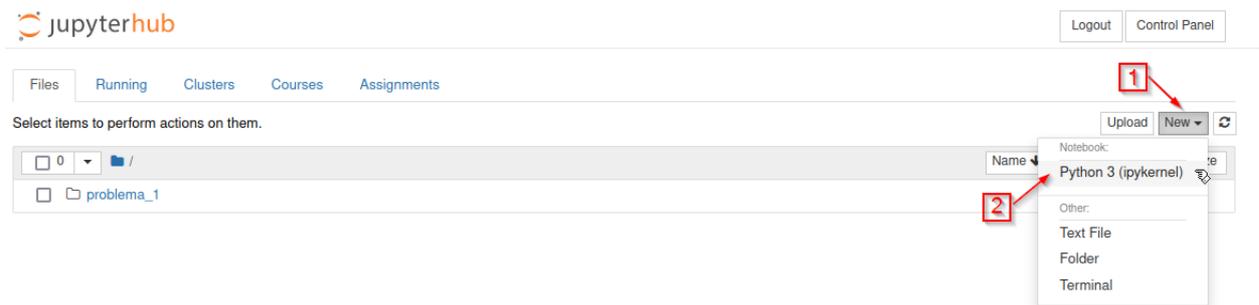


Ilustración 5-5 Menú JupyterHub: "Crear Notebook".

Una vez creado el nuevo notebook, se tendrá una vista como en la figura 5-6.

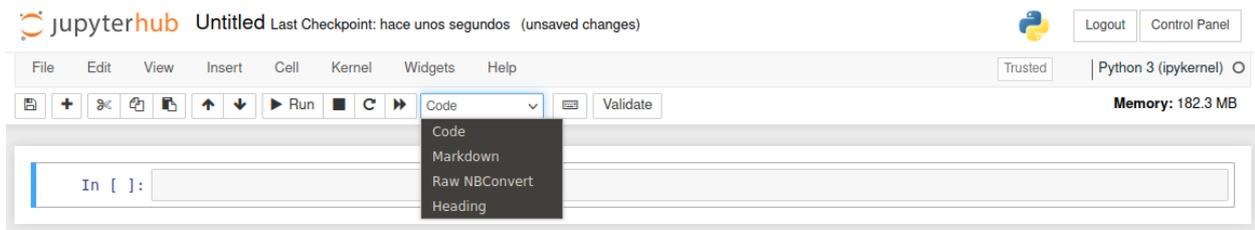


Ilustración 5-6 Menú JupyterHub: "Notebook".

Jupyter permite cuatro tipos de cabeceras:

- *Code*: en estas celdas se desarrollará el código deseado en Python.
- *Markdown*: para celdas donde se escriba texto siguiendo el estándar de Markdown, un lenguaje de marcado.
- *Raw NBConvert*: son celdas que solo serán visibles según la salida elegida. Es decir, una celda raw HTML solo se podrá visualizar cuando el cuaderno se procese a dicho formato.
- *Heading*: para celdas de cabecera.

A través del botón de Run que se ubica en la barra de herramientas del notebook se podrán ejecutar las celdas seleccionadas, ofreciendo la salida justo debajo de la celda correspondiente si se tratase de una celda de tipo Code.

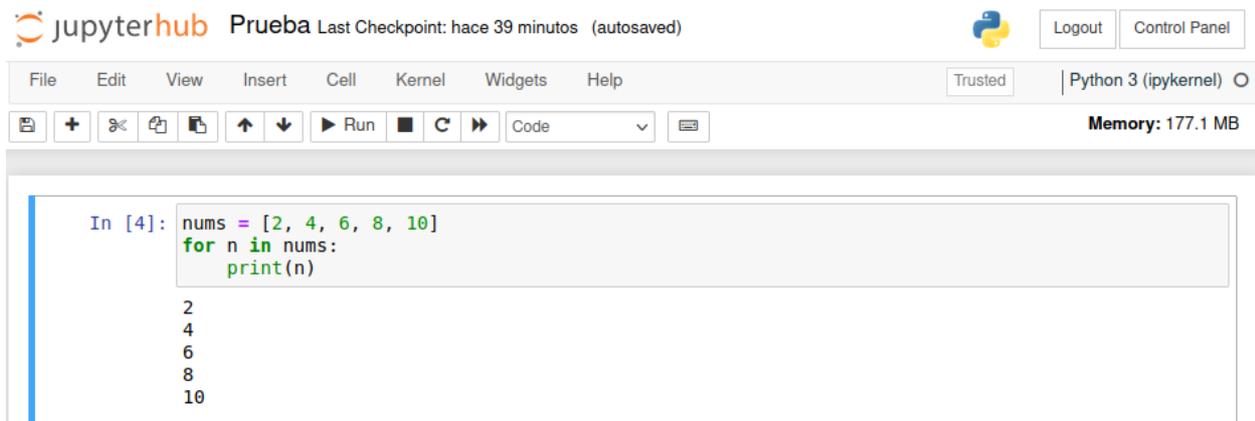


Ilustración 5-7 Menú JupyterHub: "Notebook - celda de código".

5.4 Creación, entrega y corrección de un *assignment*

• Preparación inicial

Para explicar esta parte, se ha diseñado una actividad en un notebook que se ha nombrado *Optimizacion1.ipynb*. Además, se ha creado un curso llamado Optimizacion por medio del comando:

```
nbgrader quickstart Optimizacion
```

Para que nbgrader permita crear la asignación posteriormente, se ha ubicado dicho notebook en el path */home/jupyter-instructor/Optimizacion/source/problema_1/Optimizacion1.ipynb*.

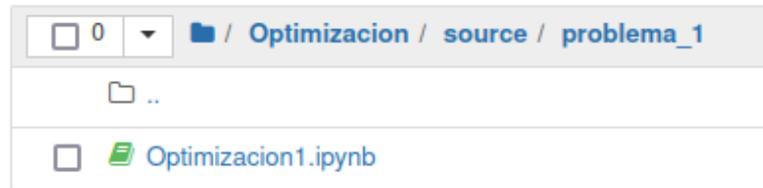


Ilustración 5-8 Menú JupyterHub: "Ruta del fichero Optimizacion1.ipynb".

Además, habrá que modificar con el siguiente contenido el archivo de configuración generado de nbgrader.

```
c = get_config()
c.CourseDirectory.root = '/home/instructor/Optimizacion'
```

El ejercicio diseñado se trata de realizar una función que calcule la distancia euclídea de una ciudad a otra. Para ello habrá que hacer uso de un diccionario con los datos proporcionados de las ubicaciones de las ciudades.

The screenshot shows a JupyterLab notebook titled "Optimizacion1" with the following content:

Ejercicio 1 (2 puntos)
 Realice una función como la anterior, donde se calcule la distancia euclídea entre las ciudades 1 y 2, usando esta vez la representación 2

In [6]:

```
# Calculo de la distancia entre dos puntos siguiendo la representación 2.
def distancia_euclídea_AB_2(A, B, dict_xy):
    # Desarrolle el código necesario.
    ### BEGIN SOLUTION
    return ((dict_xy1[A][0] - dict_xy1[B][0])**2 + (dict_xy1[A][1] - dict_xy1[B][1])**2)**0.5
    ### END SOLUTION

ciudad1 = solucion2[0] # Cambiando solucion2 por solucion3 puede comprobar
ciudad2 = solucion2[1] # los desarrollos para la representación 3.

print(distancia_euclídea_AB_2(ciudad1, ciudad2, dict_xy1))
```

57.87054518492115

In [7]:

```
# Comprueba que la función realizada en la celda anterior es correcta
from nose.tools import assert_equal
assert_equal (distancia_euclídea_AB_2(ciudad1,ciudad2,dict_xy1),57.87054518492115)
```

Ilustración 5-9 Menú JupyterHub: "Notebook - diseño de Optimizacion1.ipynb".

En cada celda habrá que seleccionar en su cabecera el rol que juegan (*Read-only* para el enunciado, *Autograded answer* para el código implementado y *Autograded test* para el test de corrección).

- **Generación y release**

Una vez se haya terminado de diseñar el ejercicio y de asignar valores a las cabeceras de las celdas, se pasa a validar el notebook por si existiese algún error. Esto se hace, como se puede apreciar en la figura anterior, pulsando sobre le botón *Validate*, que se encuentra en la barra de herramientas del notebook.

El siguiente paso a dar sería asignar la actividad a los alumnos del curso. Desde la extensión de *Formgrader*, el instructor puede realizar este paso pulsando sobre el icono ubicado en la columna de

Generate y a continuación sobre el que aparece sobre *Release*.

Manage Assignments



Assignments

Instructions (click to expand)

Search:

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
problema_1	None	released						0		

+ Add new assignment...

Ilustración 5-10 Menú JupyterHub: "Formgrader" 1.

- **Entrega de los alumnos**

Completados estos pasos, los alumnos podrán recuperar estos cuadernos lanzados por el profesor desde el menú *Assignment*. Cuando se realicen las actividades, se entregarán al profesor desde el mismo menú.

jupyterhub Logout Control Panel

Files Running Clusters Courses **Assignments**

Released, downloaded, and submitted assignments for course: Optimizacion

Released assignments

There are no assignments to fetch.

Downloaded assignments

problema_1	Optimizacion	Submit
------------	--------------	--------

Submitted assignments

problema_1	Optimizacion	Fetch Feedback
2022-09-04 19:32:05.973419 UTC		

Ilustración 5-11 Menú JupyterHub: "Assignments".

- **Recogida y corrección**

Llegados a este punto, el instructor menu deberá recolectar las actividades entregadas. Eso se efectua desde el menu *Formgrader* pulsando sobre el icono de *Collect*. Se mostrarán, a continuación, el número de actividades que han sido entregadas bajo la columna *Submitted*.

Manage Assignments



Assignments

Instructions (click to expand)

Search:

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
problema_1	None	released						1		

+ Add new assignment...

Ilustración 5-12 Menú JupyterHub: "Formgrader" 2.

El siguiente paso, será pulsar sobre ese número que se muestra, lo cual redigirá a una ventana donde se puede ver las entregas realizadas por cada alumno y el estado de cada una.

Manage Submissions



Assignments / problema_1

Instructions (click to expand)

Search:

Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback
None, None	jupyter-alumno1	2022-09-04 19:32:05 UTC	needs autograding				

Ilustración 5-13 Menú JupyterHub: "Formgrader" 3.

Ahora, teniendo en cuenta el estado de la actividad que se muestra, habrá que pulsar sobre el botón de *Autograde* para corregir la actividad. Una vez conseguida la puntuación del alumno, se puede generar el feedback y enviárselo al estudiante por medio de los dos respectivos iconos que se muestran en la figura anterior.

Si se realizan estos pasos, el alumno recibirá un archivo HTML con el cuaderno entregado y la puntuación obtenida en cada apartado.

Logout Control Panel

Files Running Clusters Courses **Assignments**

Released, downloaded, and submitted assignments for course: Optimizacion

Released assignments

There are no assignments to fetch.

Downloaded assignments

problema_1	Optimizacion	Submit
------------	--------------	---------------------

Submitted assignments

problema_1	Optimizacion	Fetch Feedback
2022-09-05 12:26:17.636254 UTC view feedback		

Ilustración 5-14 Menú JupyterHub: "Assignments" 2.

Ilustración 5-15 Menú JupyterHub: "Corrección actividad - directorio".

Ejercicio 1 (2 puntos)

Realice una función como la anterior, donde se calcule la distancia euclídea entre las ciudades 1 y 2, usando esta vez la representación 2

In [6]:

```
Student's answer (Top)

# Calculo de la distancia entre dos puntos siguiendo la representación 2.
def distancia_euclidea_AB_2(A, B, dict_xy):
    # Desarrolle el código necesario.
    # YOUR CODE HERE
    return ((dict_xy1[A][0] - dict_xy1[B][0])**2 + (dict_xy1[A][1] - dict_xy1[B][1])**2)**0.5

ciudad1 = solucion2[0] # Cambiando solucion2 por solucion3 puede comprobar
ciudad2 = solucion2[1] # los desarrollos para la representación 3.

print(distancia_euclidea_AB_2(ciudad1, ciudad2, dict_xy1))

57.87054518492115
```

In [7]:

```
Grade cell: cell-cbbd6710ee228e50 Score: 2.0 / 2.0 (Top)

# Comprueba que la función realizada en la celda anterior es correcta
from nose.tools import assert_equal

assert_equal (distancia_euclidea_AB_2(ciudad1, ciudad2, dict_xy1), 57.87054518492115)
```

Ilustración 5-16 Menú JupyterHub: "Corrección actividad - archivo".

6 PRUEBAS DE EJECUCIÓN

“La vida es como un espejo: te sonríe si la miras sonriendo”

- Jim Morrison-

Finalizados los puntos de diseño e implementación, se procede ahora en este capítulo a probar la solución adoptada. Cabe destacar que en la práctica se ha realizado un mayor número de pruebas que no han sido documentadas en esta memoria debido a la considerable extensión que provocaría en el documento, siendo aquí adjuntadas las que se han considerado más relevantes.

6.1 Despliegue de la aplicación

La aplicación se ha instalado en una máquina virtual, hosteada en un servidor ubicado en los datacenter correspondientes a la región de Paris de AWS. El sistema operativo de esta máquina virtual es Ubuntu 18.04 LTS x86 y el tipo de instancia es la t2.micro.

La versión que se ha instalado de JupyterHub es la 3.0.0 correspondiente a la distribución de The Littlest JupyterHub. Por último, nbgrader se ha instalado en la versión 0.7.1 con todas sus extensiones activadas.

6.2 Pruebas de funcionamiento

Las pruebas realizadas en cuanto al funcionamiento se refiere y que han sido completadas de forma satisfactoria son las siguientes:

- Acceso a la plataforma: el sistema es alcanzable a través de la dirección pública (<http://35.180.114.132>) de la instancia (ver *Ilustración 5-1*).
- Cierre de sesión: todos los usuarios pueden *desloguearse* correctamente desde la interfaz gráfica con el botón de *Logout*.
- Gestión de usuarios: se han podido añadir de forma correcta nuevos usuarios a través del panel de administración. Además, se han podido editar y eliminar dichas cuentas como otorgar permisos de administrador (ver *Ilustración 5-3*).
- Gestión de servidores individuales dedicados: otra funcionalidad probada correctamente respecto a la gestión de usuarios es el arranque o pausa de los Sigle-User Notebook servers de las cuentas existentes en la Plataforma (ver *Ilustración 5-4*).
- Uso de notebooks: tanto el instructor como las cuentas de alumnos han podido crear nuevos notebooks. Los cuadernos permiten tener celdas de texto, contenido multimedia, ecuaciones y

código. También se ha comprobado que estos cuadernos puedan ser ejecutados, ofreciendo la salida esperada respecto al código diseñado, funcionando como el intérprete de Python tradicional (ver *Ilustración 5-7*)

- Terminal: Jupyter viene con una terminal integrada, cuyo funcionamiento también se ha probado, resultando funcionar correctamente. Desde esta terminal se ha instalado nbgrader y sus extensiones, entre otros paquetes.

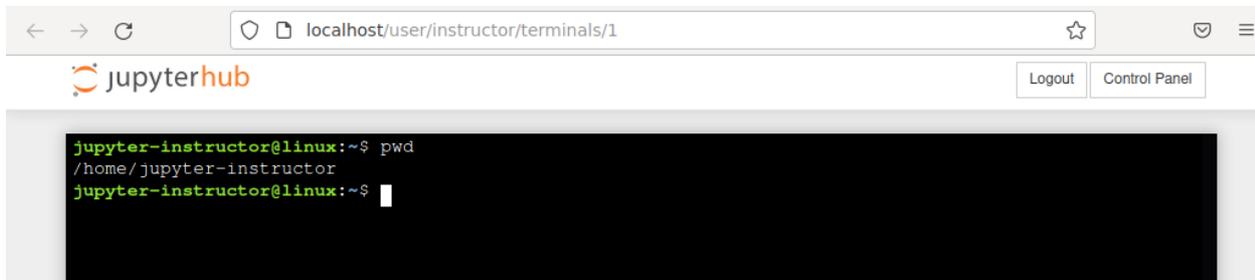


Ilustración 6-1 Menú JupyterHub: “Terminal”.

- Formgrader: ofrece la interfaz gráfica esperada, desde la cual se pueden realizar las tareas de generación, lanzamiento y corrección de actividades, manejo de usuarios que usan nbgrader.

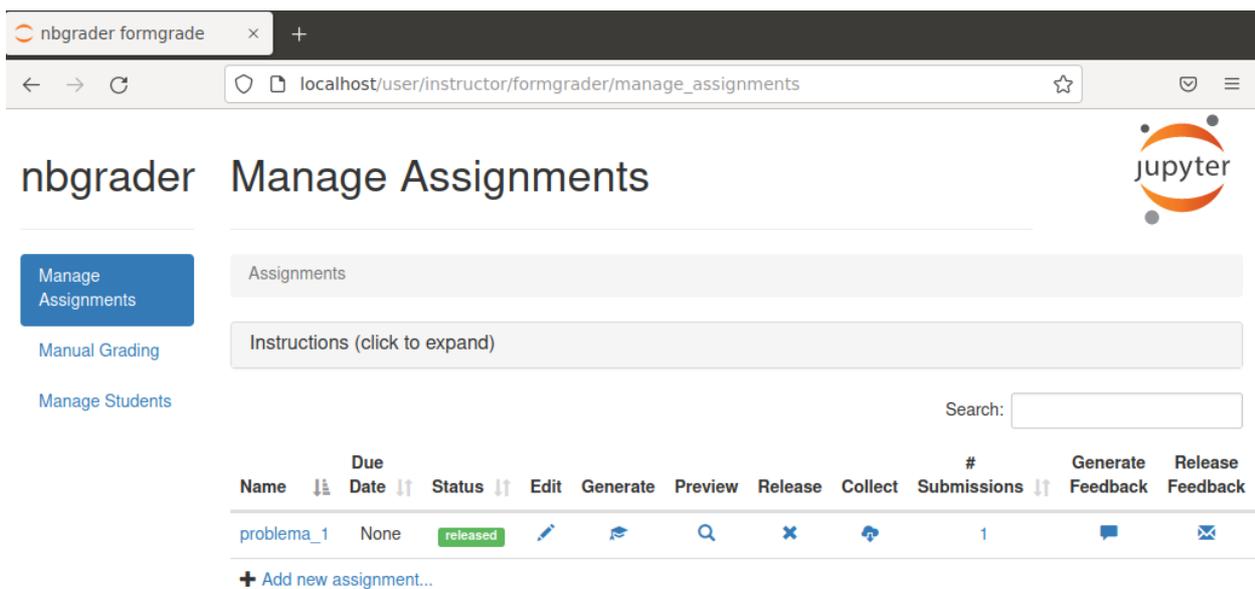


Ilustración 6-2 Menú JupyterHub: “Formgrader” 4.

- Generación y lanzamiento de actividades: se pueden generar y hacer que lleguen a los alumnos del curso las asignaciones preparadas. Esto se puede realizar tanto desde Formgrader como desde la terminal (ver *Ilustración 5-10*).
- Desarrollo y entrega de asignaciones: los alumnos han podido acceder a los notebooks recibidos por parte del profesor, trabajar y ejecutar las celdas en ellos, y una vez se completan, entregarlos de vuelta al profesor para que sean corregidos (ver *Ilustración 5-11*).
- Recolecta de actividades: el profesor, una vez se han entregado los trabajos, ha podido recuperarlos tanto por medio de Formgrader como de la línea de comandos (ver *Ilustración 5-12*).

- Corrección de actividades: a través de la extensión Autograde se ha conseguido puntuar las actividades entregadas, para ser entregadas dichas calificaciones a los alumnos (ver Ilustración (ver *Ilustración 5-13*)).
- Entregar calificaciones: a los alumnos que hubiesen entregado la actividad, al entregar el profesor las calificaciones ha llegado de vuelta un archivo HTML donde han podido visualizar desde su interfaz gráfica la nota obtenida por su trabajo (ver *Ilustración 5-14, 5-15 y 5-16*).

6.3 Pruebas de rendimiento

En cuanto a las pruebas de rendimiento, se han desplegado en paralelo una cantidad de 5 instancias de servidores Notebook dedicados, es decir 5 usuarios con sesión activa a la misma vez, y los usuarios han podido trabajar con normalidad y sin demoras de procesamiento, creando, editando y ejecutando notebooks.

Este número de alumnos es bajo para la cantidad de usuarios que se espera tener trabajando a la vez en un curso universitario. Sin embargo, el rendimiento no supondría un problema para el despliegue diseñado, ya que las instancias de EC2 permiten aumentar la capacidad de memoria y CPU dinámicamente.

6.4 Pruebas de seguridad

Respecto a las pruebas de seguridad, se ha comprobado que funcionan correctamente los siguientes puntos:

- Control de acceso: cualquier persona que quiera acceder al sistema deberá acceder con sus credenciales, habiendo obtenido previamente un usuario con el que acceder por parte del administrador (ver *Ilustración 5-1*).
- Cambio de contraseña: los usuarios son capaces de cambiar la clave de su cuenta por medio de la URL <http://35.180.114.132/hub/auth/change-password>.

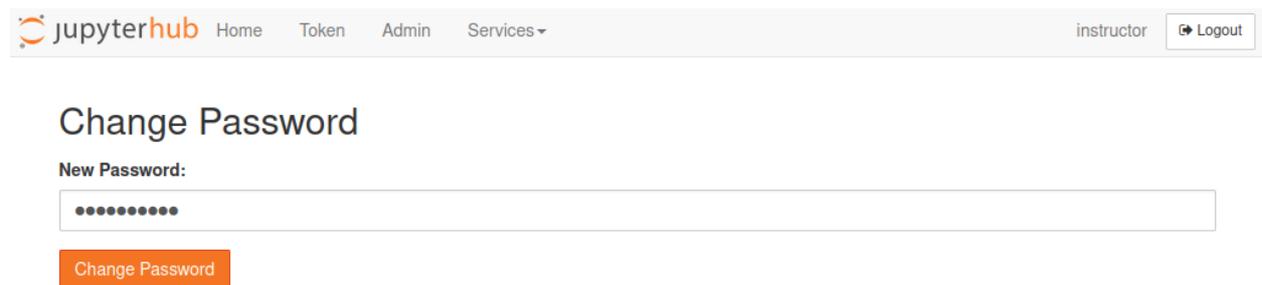


Ilustración 6-3 Menú JupyterHub: "Cambiar contraseña".

- Restricción de accesos: para acceder a los recursos ubicados en el servidor, se deberá pasar un filtro para evitar que usuarios accedan por medio de la URL con la que se accede a dichos recursos. Ni el propio usuario administrador podrá acceder a los *notebooks* del resto de usuarios

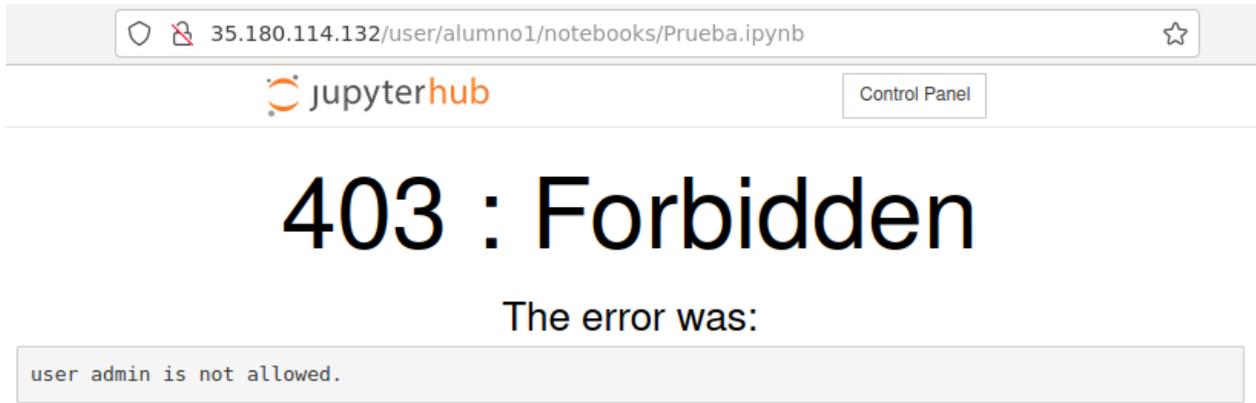


Ilustración 6-4 Menú JupyterHub: "Acceso prohibido".

- Acceso a directorios no permitidos desde la terminal: se prohíbe el acceso a directorios de otros usuarios a través de la terminal integrada con JupyterHub

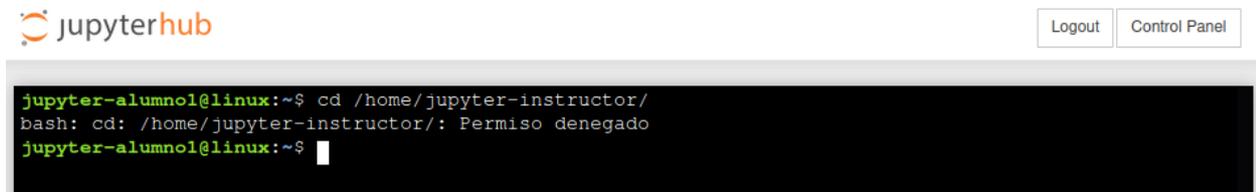


Ilustración 6-5 Menú JupyterHub: "Terminal - permiso denegado".

- Sesión caducada: pasado un tiempo sin hacer uso de la aplicación, el servidor dedicado del usuario en cuestión se pausará y nos aparecerá una pantalla como la de la siguiente ilustración.

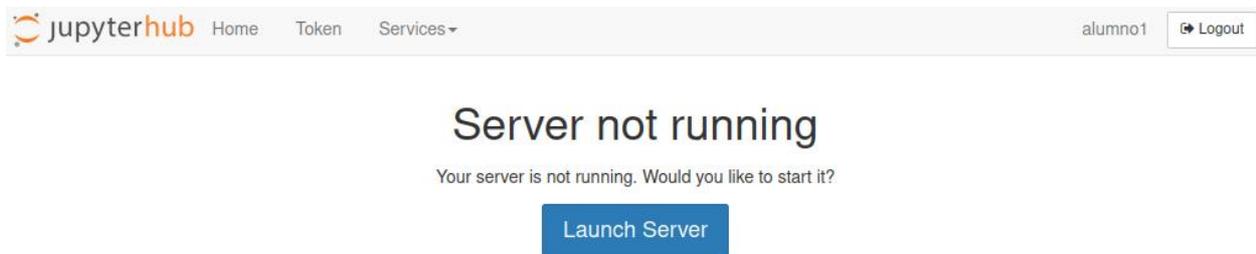


Ilustración 6-6 Menú JupyterHub: "Sesión caducada"

7 CONCLUSIÓN Y LÍNEAS DE FUTURO

“La vida cambia cuando empiezas a enfocarte en ti”

- Anuel -

Legados a este punto, habiendo estudiado la necesidad del proyecto, descrito el objetivo a alcanzar con el trabajo e implementado la solución adoptada en base al diseño realizado, podemos realizar este punto de conclusión reuniendo las ideas más importantes que se pueden extraer de esta memoria, además de plantear líneas de futuro que pueden ser abarcadas por posteriores estudiantes.

7.1 Conclusiones

En primer lugar, podemos hablar de los objetivos que se planteaban al inicio de la memoria y si han sido satisfechos con este desarrollo realizado.

Respecto al entorno de programación por medio de interfaz gráfica interactiva, se puede decir que ha sido claramente cumplido con creces. La solución adoptada, Jupyter Notebook, es una herramienta cuya popularidad y uso en entornos docentes está en crecimiento exponencial, y ya son varias las universidades españolas, como la Universidad Carlos III de Madrid, que tienen un servidor dedicado para que los alumnos puedan trabajar de esta forma cuando se trata de programar sobre todo funciones de análisis de datos.

En cuanto al sistema de intercambios de ficheros con funcionalidades de corrección, también se puede decir que se ha alcanzado añadiendo la extensión nbgrader a Jupyter. El profesor es capaz de lanzar actividades a los alumnos, ellos pueden trabajar en ellas una vez las reciban y posteriormente entregarlas, y finalmente el docente tiene la capacidad a través de la herramienta de corregirlas, puntuarlas y entregar la calificación a dichos alumnos.

Destacar también el gran valor que ofrece desplegar un servicio así en Amazon Web Services, liberando a la Universidad de la gestión, mantenimiento del servidor y modificación de parámetros de red (políticas de seguridad, NAT, routing, etc) en la infraestructura de la Universidad para dar acceso a los alumnos. Solo habría que costear el tiempo de computación que se dedicase en dicha máquina virtual, lo cual no supone un precio elevado (0,0132USD por hora de uso activo).

Faltaría que se implementase en un curso universitario real y se usase durante un periodo de tiempo considerable para obtener conclusiones más certeras acerca del rendimiento de la plataforma y, sobre todo, de la mejora de la calidad de la enseñanza.

En resumen, creo que es una solución que debe adoptar la Universidad de Sevilla, ya que es una herramienta que puede facilitar mucho a los alumnos a iniciarse con la programación. Ofrece una interfaz más amigable que la de un editor de código tradicional y mayor sencillez e inmediatez a la hora de ejecutar el trabajo realizado que la que se tiene desde la línea de comandos.

7.2 Líneas de futuro

Como se ha comentado en la memoria y se puede comprobar en la documentación del sitio oficial de JupyterHub, sería bastante interesante cómo se podría configurar a fondo cada uno de los módulos que ofrece esta herramienta, como el Hub, el Authenticator, el proxy, etc, probando todas las posibilidades que ofrecen y viendo cual es la más conveniente para cada caso.

Además, también se podrían desplegar los otros dos escenarios que ofrece nbgrader, si es que la Universidad se plantease en algún momento desplegar esta herramienta. Estos escenarios ofrecen una funcionalidad más extensa, permitiendo desplegar en un mismo servidor varios cursos y diferentes profesores.

Por último a destacar, se podría ver como se podría estudiar si es viable implementar en el código de nbgrader una función que habilite las entregas de las actividades durante un tiempo determinado. Esto añadiría gran valor a la herramienta, ya que, aunque actualmente las actividades llegan al profesor con un *timestamp*, el docente podría bloquear las entregas una vez pasado el tiempo otorgado a la actividad.

REFERENCIAS

- [1] “Era de la información,” *Wikipedia*. [Online]. Available: https://es.wikipedia.org/wiki/Era_de_la_informaci%C3%B3n
- [2] R. Díaz, “Ocho de cada 10 servicios de la Administración Pública de la UE están disponibles ‘online,’” *Cinco Días*, Sep. 2022, [Online]. Available: https://cincodias.elpais.com/cincodias/2022/09/01/companias/1662043181_093379.html
- [3] “Eclipse.” <https://www.eclipse.org/>
- [4] “Spyder.” <https://www.spyder-ide.org/>
- [5] “Visual Studio Code.” <https://code.visualstudio.com/>
- [6] “Ubuntu.” <https://ubuntu.com/>
- [7] “Emacs.” <https://www.gnu.org/software/emacs/>
- [8] “Sistema de gestión de aprendizaje,” *Wikipedia*. [Online]. Available: https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_aprendizaje
- [9] “MySQL.” <https://www.mysql.com/>
- [10] “Sales Force,” *¿Qué es Cloud Computing?* <https://www.salesforce.com/mx/cloud-computing/>
- [11] “Oracle,” *¿Qué es SaaS (Software como servicio)?* <https://www.oracle.com/es/applications/what-is-saas/>
- [12] “IEEE.” <https://www.ieee.org/>
- [13] “Moodle.” <https://moodle.org/?lang=es>
- [14] “Blackboard.” <https://www.blackboard.com/es-es>
- [15] “Enseñanza Virtual.” <https://ev.us.es/>
- [16] “Amazon Web Services.” <https://aws.amazon.com/es/>
- [17] “Microsoft Azure.” <https://azure.microsoft.com/es-es/>
- [18] “Google Cloud Platform.” <https://cloud.google.com/?hl=es>
- [19] “JupyterHub.” <https://jupyter.org/hub>
- [20] “Nbgrader.” <https://nbgrader.readthedocs.io/en/stable/#>
- [21] “Amazon Web Services,” *Amazon EC2*. <https://aws.amazon.com/es/ec2/>
- [22] “Github,” *Jupyter Kernels*. <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
- [23] “The Littlest JupyterHub.” <https://tljh.jupyter.org/en/latest/>
- [24] “Zero to JupyterHub with Kubernetes.” <https://zero-to-jupyterhub.readthedocs.io/en/stable/>
- [25] “Centro de procesamiento de datos,” *Wikipedia*. [Online]. Available: https://es.wikipedia.org/wiki/Centro_de_procesamiento_de_datos