

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



"PLATAFORMA DE RECURSOS DE
APRENDIZAJE PARA PROGRAMACIÓN"

TRABAJO FIN DE GRADO

Junio - 2022

AUTOR: Julio Montesinos Alvarado
DIRECTORA: Kristina Polotskaya

RESUMEN

En la actualidad hay numerosas plataformas destinadas al aprendizaje con el fin de formar, calificar y facilitar la curva de aprendizaje a los alumnos en temas diversos. Cada plataforma ofrece una serie de funcionalidades que quizás otras no lo ofrecen, por lo que están continuamente mejorando la experiencia del usuario con el fin de que utilicen la mayor cantidad de estudiantes posibles su plataforma. Aunque en algunas ocasiones se debe utilizar más de una plataforma diferente de forma simultánea para utilizar funcionalidades que no te ofrece la otra plataforma.

Esta aplicación ha sido desarrollada con el fin de suplir la necesidad de utilizar un editor de código de programación online de forma sencilla e intuitiva dentro de la propia aplicación que ofrece los cursos para mejorar la experiencia del usuario y no tener que utilizar otra aplicación distinta y poder ejecutar el código. Como novedad, se ha utilizado una tecnología nueva que permite ejecutar el código en la parte del cliente y no en la parte del servidor.

Además permite tener diferentes tipos de rol de usuario con la potestad de gestionar cursos, usuarios y contenidos de los cursos, dependiendo del rol que se tenga. Se permite realizar tareas utilizando un editor de código de programación online, entregarlas y ser corregidas con alguna observación por parte del docente.

La aplicación ha sido desarrollada principalmente mediante el lenguaje de programación PHP, JavaScript, HTML5, CSS y en menor medida MySQL, Python, Pyodide, JSON utilizando la técnica de desarrollo web AJAX y la tecnología WebAssembly y CodeMirror. Todo ello siguiendo el patrón MVC.

Por último, el desarrollo está realizado de forma que se puedan ampliar sus funcionalidades de forma sencilla sin tener que hacer grandes cambios en el código.

AGRADECIMIENTOS

Este trabajo pone fin a mi etapa universitaria, rodeado de personas que se han convertido en una parte importante en mi vida y de profesores que me han apoyado mucho a lo largo de toda la carrera.

Agradezco profundamente el apoyo que me han proporcionado mis padres a lo largo de toda mi etapa universitaria, y en especial a mi hermana Andrea, puesto que siempre me han ayudado en todo lo que han podido, sobre todo en momentos difíciles. Además también quiero destacar lo afortunado que he sido al poder tener unos compañeros de clase que puedo considerar amigos para toda la vida, siendo Alejandro Porcar, Dmitry Romanyutenko, Daniel Parres y Javier Del Campo.

Los largos momentos que he pasado estudiando en la biblioteca junto a mis amigos de toda la vida en verano tampoco se me olvidarán nunca.

¡ Este trabajo lo dedico a todos vosotros, ya que sin vosotros no lo hubiera podido conseguir !



ÍNDICE GENERAL

1.- Introducción	9
1.1.- Empresa/Entorno de trabajo	9
1.2.- Justificación del proyecto	10
1.3.- Objetivos	11
1.3.1.- Objetivos principales	11
1.3.2.- Objetivos secundarios	11
1.3.3.- Objetivos personales	12
1.4.- Límites del proyecto	12
2.- Antecedentes y estado de la cuestión	14
2.1.- Situación actual de la organización	14
2.2.- Herramientas disponibles en el mercado	15
2.2.1.- Kaggle	15
2.2.1.1 Cursos	15
2.2.1.2 Repositorio de datos	16
2.2.2.- Moodle	17
2.2.2.1 Tareas	18
2.2.2.2 Roles de usuario	19
2.2.2.3 Calificaciones	21
2.2.3.- Google Colab	21
2.2.3.1.- Jupyter Notebook	22
2.2.4.- DataCamp	22
2.2.4.1- Espacio de trabajo	23
2.2.5.- Resumen	23
2.3.- Análisis de otros entornos de desarrollo	25
2.3.1 Consola Pyodide	25
2.3.2 Cuaderno RetroLab	26
2.3.3 Cuaderno Basthon	27
2.4.- Análisis de editores de código online	30
2.4.1 CodeAnywhere	30
2.4.2 Prism.js	31
2.4.3 Editor de Mónaco	31
2.4.4 CodeMirror	32
2.5.- Resumen de entornos y editores de código	33
3.- Hipótesis de trabajo	37
3.1.- Tecnologías del lado del cliente	38
3.1.1.- HTML	39
3.1.2.- CSS	39

3.1.3.- JavaScript	40
3.1.4.- AJAX	40
3.1.5.- JSON	41
3.1.6.- CodeMirror	42
3.1.7.- Pyodide	42
3.1.8.- WebAssembly	43
3.2.- Tecnologías del lado del servidor	43
3.2.1.- PHP	44
3.2.2.- Python	44
3.2.3.- Base de datos MySQL	45
4.- Metodología y resultados	47
4.1.- Planificación del proyecto	47
4.1.1.- Ciclo de vida	48
4.1.2.- Etapas del proyecto. Diagrama de Gantt	48
4.2.- Captura de requisitos	50
4.3.1.- Diagrama Entidad/Relación y Relacional	55
4.3.2.- Interfaz gráfica	57
4.4.- Implementación	59
4.4.1.- CodeMirror	59
4.4.1.- Pyodide	60
5.- Conclusiones y trabajo futuro	63
5.1.- Conclusiones	63
5.2.- Posibles desarrollos futuros	64
Bibliografía	67
Anexo	73

ÍNDICE DE TABLAS

Tabla 2.1 Resumen comparativo de herramientas estudiadas	25
Tabla 4.1 Rol de usuario no identificado	50
Tabla 4.2 Rol de alumno	51
Tabla 4.3 Rol de profesor	52
Tabla 4.4 Rol de profesor responsable	53
Tabla 4.5 Rol de administrador	54
Tabla A.1 C.U.1: Iniciar sesión	74
Tabla A.2 C.U.2: Registrarse	74
Tabla A.3 C.U.3: Ver cursos disponibles	75
Tabla A.4 C.U.4: Cerrar sesión	75
Tabla A.5 C.U.5: Seleccionar curso	76
Tabla A.6 C.U.6: Ver cursos afectados	76
Tabla A.7 C.U.7: Editar perfil de usuario	77
Tabla A.8 C.U.8: Solicitar inscripción en un curso	77
Tabla A.9 C.U.9: Ocultar y visualizar menú lateral	78
Tabla A.10 C.U.10: Ver detalles de un curso	78
Tabla A.11 C.U.11: Ver tareas	79
Tabla A.12 C.U.12: Consultar nota de una tarea	79
Tabla A.13 C.U.13: Entrar en una tarea	80
Tabla A.14 C.U.14: Ver material	80
Tabla A.15 C.U.15: Hacer clic en la URL del material	81
Tabla A.16 C.U.16: Limpiar salida en pantalla del código programado	81
Tabla A.17 C.U.17: Cerrar ventana modal	82
Tabla A.18 C.U.18: Escribir código de programación	82
Tabla A.19 C.U.19: Ejecutar código de programación	83
Tabla A.20 C.U.20: Guardar y entregar el código programado	84
Tabla A.21 C.U.21: Ver entregas de todos los alumnos del curso	85
Tabla A.22 C.U.22: Asignar nota a entregas	86
Tabla A.23 C.U.23: Editar contenido del curso	87
Tabla A.24 C.U.24: Gestionar profesores no responsables	88
Tabla A.25 C.U.25: Gestionar alumnos	89
Tabla A.26 C.U.26: Añadir profesor	90
Tabla A.27 C.U.27: Eliminar profesor	91
Tabla A.28 C.U.28: Gestionar solicitud del alumno	91
Tabla A.29 C.U.29: Eliminar alumno	92
Tabla A.30 C.U.30: Gestionar cursos	92
Tabla A.31 C.U.31: Gestionar profesores	93
Tabla A.32 C.U.32: Gestionar ediciones	93
Tabla A.33 C.U.33: Añadir cursos	94

Tabla A.34 C.U.34: Eliminar curso	94
Tabla A.35 C.U.35: Editar curso	95
Tabla A.46 C.U.36: Añadir profesor	95
Tabla A.37 C.U.37: Eliminar profesor	96
Tabla A.38 C.U.38: Editar profesor	96
Tabla A.39 C.U.39: Añadir edición	97
Tabla A.40 C.U.40: Eliminar edición	97
Tabla A.41 C.U.41: Editar edición	98



ÍNDICE DE FIGURAS

Figura 2.1 Listado de cursos de Kaggle	16
Figura 2.2 Datasets de Kaggle	17
Figura 2.3 Tareas de Moodle	18
Figura 2.4 Contenido de una tarea de Moodle	19
Figura 2.5 Jupyter Notebook en Google Colab	22
Figura 2.6 Workspace de DataCamp	23
Figura 2.7 Consola de Pyodide en el navegador	26
Figura 2.8 Interfaz de RetroLab	26
Figura 2.9 Interfaz de la consola de Basthon	27
Figura 2.10 Interfaz del cuaderno de Basthon	27
Figura 2.11 Funcionalidades que ofrece CodeAnywhere	30
Figura 2.12 Llamada a la librería Prism desde HTML	31
Figura 2.13 Comparación de 2 códigos en Editor de Mónaco	32
Figura 2.14 Ejemplo de código para el uso de CodeMirror	33
Figura 2.15 Botones en gráfica del cuaderno Basthon	34
Figura 2.16 Pizarra del cuaderno Basthon	34
Figura 2.17 Historial de versiones del cuaderno Basthon	35
Figura 3.1 Esquema del patrón MVC	38
Figura 3.2 Estructura del objeto en JSON	41
Figura 3.3 Estructura del array en JSON	41
Figura 3.4 Estructura del array con espacios en JSON	41
Figura 4.1 Ciclo de vida incremental	48
Figura 4.2 Diagrama de Gantt	49
Figura 4.3 Tipos de usuario y relación de herencia	50
Figura 4.4 Casos de uso del usuario no identificado	51
Figura 4.5 Casos de uso del alumno	51
Figura 4.6 Casos de uso del profesor	52
Figura 4.7 Casos de uso del profesor responsable	53
Figura 4.8 Casos de uso del administrador	55
Figura 4.9 Modelo Entidad/Relación de la base de datos	56
Figura 4.10 Modelo relacional visto desde XAMPP	57
Figura 4.11 Diseño de la parte pública y privada de la aplicación	58
Figura 4.12 Menú lateral izquierdo dependiendo del rol de usuario	58
Figura 4.13 Apariencia de la sección “Gestión Profesores” del administrador	59
Figura 4.14 Importación del código fuente y de python en CodeMirror	60
Figura 4.15 Transformación de textarea a editor de CodeMirror	60
Figura 4.16 Función para cargar Pyodide	61
Figura 4.17 Ejecución de código python con Pyodide	61



Capítulo 1

Introducción

1.1.- EMPRESA/ENTORNO DE APLICACIÓN

El ser humano ha necesitado a lo largo de toda su evolución diferentes técnicas y habilidades para transmitir costumbres y conocimientos a lo largo de cientos de generaciones. Conforme va pasando el tiempo, la sociedad va evolucionando en busca de nuevos métodos con el objetivo de aumentar la sencillez y la utilidad a la hora de enseñar nuevos conocimientos para permitir que la mayor cantidad posible de seres humanos tengan la oportunidad de formarse, logrando un pensamiento crítico y ser competente.

La aparición de la red informática mundial (WWW) ha permitido tener oportunidades que jamás hubiera pensado el ser humano que sería posible, como por ejemplo acceder a todo tipo de información de forma gratuita, poder estar conectado con otra persona independientemente de su localización geográfica, tener todo tipo de entretenimiento, entre otros muchos. Todo ello ha permitido además en muchos casos cambiar la metodología de enseñanza mediante cursos a los alumnos, de manera que ya no es imprescindible asistir presencialmente a clase o

entregar al profesor una tarea compuesta por una cantidad importante de folios escritos a bolígrafo.

Actualmente hay plataformas de enseñanza con infinidad de temáticas que permiten a los alumnos obtener los conocimientos que deseen de forma muy cómoda, en la que se pueden tener clases tutorizadas en formato online mediante videollamadas, tareas programadas y autocorregibles y otros métodos, que sin la aparición de internet sería imposible aplicarlos. Además cabe destacar que en muchas ocasiones se puede estudiar un curso en el que está siendo impartido por un profesor situado en la otra parte del mundo.

Esta aplicación será diseñada para el centro de investigación operativa (CIO) de la Universidad Miguel Hernández para abordar ciertas necesidades en situaciones diferentes.

1.2.- JUSTIFICACIÓN DEL PROYECTO

Hoy en día existen muchas plataformas de enseñanza en su mayoría sujetas a un plan de suscripción, aunque también gratuitas ofreciendo una serie de funcionalidades que permiten una buena metodología de aprendizaje para el alumno. Pero con respecto a cursos de formación en tecnologías, hay muchas plataformas que carecen de un editor de código online, o por el contrario no disponen de un proceso de evaluación al alumno.

Por todo ello, surge la necesidad de crear una aplicación de cursos ofertados por los docentes con la integración de un editor de código online, de manera que los usuarios puedan inscribirse en los cursos, formarse con el material que está adjuntado en los cursos y poner los conocimientos en práctica realizando tareas en un entorno que permite codificar python. Todo ello con el objetivo de aplanar lo máximo posible la curva de aprendizaje del alumno sin tener que usar varias plataformas simultáneas, es decir, evitando así las situaciones en las que el alumno utiliza una plataforma para realizar tareas y obtener calificaciones y además otra plataforma diferente para utilizar el editor de código y programar el resultado de la tarea requerida.

Por otro lado, en muchas ocasiones a los docentes se les presentan oportunidades para realizar mini cursos con alguna temática en concreto, pero con un plazo para la organización del curso muy reducido, por lo que a veces no realizan los cursos de formación por falta de tiempo. Esta aplicación permite reducir mucho el tiempo empleado por el docente para preparar el curso, puesto que está todo más organizado y una vez se haya creado el curso, ya estaría disponible para siempre evitando a los docentes preparar el mismo curso varias veces.

Otra razón por la que se ha realizado esta aplicación es la motivación personal, ya que después de haber estado realizando prácticas para el departamento de estadística, matemáticas e informática (DEMI) de la UMH relacionadas con la investigación de algoritmos de ciencia de datos, quería crear una aplicación que permitiera facilitar el aprendizaje de muchos

conceptos relacionados con la ciencia de datos a estudiantes que empiezan desde cero proporcionándoles un entorno que les permitiera programar algoritmos, crear gráficas y preprocesar grandes cantidades de datos.

1.3.- OBJETIVOS

En el tercer apartado del capítulo 1, se van a explicar los objetivos que se tienen en la realización del proyecto a nivel principal, secundario y personal.

1.3.1.- Objetivos principales

El objetivo principal es el diseño de una aplicación de cursos cuya funcionalidad principal sea ejecutar código en un editor de código online. Se podrán crear cursos de manera que exista un administrador que pueda gestionarlos, docentes que puedan impartirlos y alumnos que tengan la oportunidad de inscribirse en ellos.

Con respecto a la gestión de cursos, se deberá poder editar cierta información de los cursos, así como eliminar o añadir nuevos. Por otro lado se tendrá el control de la gestión de usuarios en todo momento, dependiendo del rol de usuario.

Los usuarios nuevos podrán registrarse, iniciar sesión y solicitar admisión a un curso. La solicitud de admisión será gestionada por el profesor responsable y si se decide aceptarla, el usuario ya será alumno del curso, por lo que podrá acceder a todo su contenido. Además, independiente del rol que tenga el usuario, podrá editar cierta información de su cuenta.

Los docentes de un curso podrán modificar el contenido de la información general de un curso, el material que proporcione al alumno y tendrá la potestad de añadir, eliminar o modificar las tareas que hay en el curso. Si el docente es responsable del curso, podrá además gestionar a los alumnos y a los profesores no responsables de dicho curso.

El administrador es el único usuario que no está controlado por otro usuario, es decir, ningún otro usuario puede modificar o eliminar al usuario con rol de administrador. Será el responsable de activar o desactivar los cursos y gestionar a los profesores, cursos y ediciones de cursos.

1.3.2.- Objetivos secundarios

Como objetivo secundario en la realización del proyecto, se encuentra la creación de un entorno usable mediante una interfaz sencilla e intuitiva, de modo que un usuario con pocos conocimientos en la informática pueda utilizarlo sin dificultades. Además toda la aplicación

se realizará utilizando la técnica AJAX que permite al usuario interactuar con la aplicación sin la necesidad de tener que refrescar la página en ningún momento.

1.3.3.- Objetivos personales

Con respecto a los objetivos personales que tengo a la hora de realizar el proyecto serían ser capaz, empezando desde cero, de conseguir desarrollar una aplicación totalmente funcional aplicando muchos de los conocimientos adquiridos durante la carrera y ser capaz de solventar infinidad de problemas que surgen a lo largo del proceso de desarrollo de la aplicación.

Otro objetivo sería ser capaz de ver en muchos aspectos la utilidad de algunos aprendizajes adquiridos en mi etapa académica en proyectos que resuelvan una problemática real. Esto se debe a que en ocasiones se realizan pequeños ejercicios y se estudia mucha teoría sin tener claro para qué sirven esos conocimientos en la vida real.

Por último, me gustaría ser capaz de superarme a mí mismo, de manera que demuestre ser capaz de desarrollar funcionalidades, dominar técnicas y tecnologías que no se han visto en la carrera universitaria, apoyándome únicamente de la documentación que haya en la web y de foros de comunidades que ayudan a solventar ciertos problemas. Además que todo el desarrollo permita satisfacer las necesidades del usuario que va a utilizarlo.

1.4.- LÍMITES DEL PROYECTO

En este proyecto habrá ciertas funcionalidades que quedan fuera del ámbito del trabajo tanto en el presente como en el futuro:


- **Utilización en dispositivos móviles:** La aplicación de los cursos no podrá ser usable desde un dispositivo móvil.
- **Un único profesor responsable por curso:** Dentro de un curso, no se permitirá que haya más de un profesor responsable. Pueden haber todos los profesores no responsables que se desee.
- **Un usuario no registrado no podrá acceder a un curso:** Si el usuario no está identificado, no podrá en ningún caso ingresar como alumno dentro de un curso.
- **No se incorporarán funcionalidades de terceros con derechos de autor:** Todas las funcionalidades que se creen deberán ser creadas por el propio desarrollador o si se utilizan funcionalidades externas, deberán ser software de código abierto (open source).

- **Un docente de un curso no puede ser alumno de ese curso:** Si un usuario está impartiendo un curso, no puede ser al mismo momento alumno de ese propio curso.
- **No se tendrá acceso a la contraseña de ningún usuario:** Todos los usuarios tendrán única y exclusivamente acceso a su contraseña, por lo que nadie más tendrá acceso a su contraseña. Ni siquiera el administrador tendrá acceso a las contraseñas de los otros usuarios.



Capítulo 2

Antecedentes y estado de la cuestión



En este capítulo se va a realizar un estudio sobre algunas de las plataformas más grandes de ciencia de datos y de creación de cursos que hay en la actualidad. Se va a dividir en 5 apartados, en primer lugar se verá la situación actual de la organización, después se describirán las herramientas que hay disponibles en el mercado, se analizarán los entornos de desarrollo de código, se estudiarán además algunos editores de código online y por último se juzgará mediante un resumen la información expuesta en las herramientas estudiadas y se realizará la selección de la herramienta más adecuada para el desarrollo del proyecto.

2.1.- SITUACIÓN ACTUAL DE LA ORGANIZACIÓN

Actualmente se pueden encontrar infinidad de herramientas y lenguajes de programación que se utilizan para el desarrollo de aplicaciones, herramientas o servicios, aunque es difícil encontrar plataformas que integren en un mismo sitio web dichas herramientas.

La comunidad universitaria dispone de la plataforma “Moodle” para la elaboración de cursos, tareas y exámenes, pero cuando se debe realizar ejercicios que requieran escribir algún lenguaje de programación se necesita utilizar otras plataformas que lo permitan. Por todo ello, surge la necesidad de crear una plataforma de aprendizaje que integre los recursos de programación con el objetivo de facilitar el trabajo a los docentes y mejorar la experiencia de uso del estudiante. En definitiva permitiría realizar ejercicios que requieran recursos de programación dentro del propio curso sin necesidad de salirse de la plataforma.

2.2.- HERRAMIENTAS DISPONIBLES EN EL MERCADO

A continuación se describirán algunas de las herramientas de las plataformas más grandes que hay en el mercado que pueden cubrir, en gran parte, las necesidades que tiene la organización a la que se está haciendo el trabajo.

2.2.1.- Kaggle

Kaggle es una de las plataformas más grandes de Data Science adquirida por Google en marzo de 2017 que permite a los usuarios encontrar y publicar conjuntos de datos, explorar y crear modelos en un entorno de ciencia de datos basado en la web. También permite participar en concursos para resolver desafíos de ciencia de datos y hacer cursos de inicialización en ciencia de datos. Se trata de una plataforma con una comunidad en línea de científicos de datos, profesionales del aprendizaje automático y también de muchos estudiantes con ganas de aprender cosas nuevas.

Kaggle comenzó en 2010 ofreciendo concursos de aprendizaje automático y ahora también ofrece una plataforma de datos públicos, un banco de trabajo basado en la nube para ciencia de datos y educación en inteligencia artificial.

2.2.1.1 Cursos

La plataforma de Kaggle dispone de numerosos cursos para adquirir las habilidades que se necesitan para realizar proyectos independientes de ciencia de datos. Todos los cursos reducen los temas complejos a sus componentes prácticos clave con la finalidad de obtener habilidades y comprender cómo funciona el tema que se está impartiendo en unas pocas horas (en lugar de semanas o meses).

En el momento en el que se está redactando esta documentación se dispone de 16 mini cursos con diferentes secciones. Cada curso se divide en una parte de tutorial, donde se explica el

contenido y se realiza un pequeño ejemplo, y una parte de ejercicio donde se pone en práctica los contenidos vistos en la parte de tutorial. Cada curso tendrá además unos prerrequisitos recomendados de lecciones que se deberían haber visto antes para poder comprender correctamente el mini curso.



Figura 2.1.- Listado de cursos de Kaggle

2.2.1.2 Repositorio de datos

Los repositorios de datos son sistemas de información que preservan y organizan materiales científicos y académicos como apoyo a la investigación y el aprendizaje [2]. Este repositorio estará compuesto por múltiples datasets.

El término Dataset en sí es un término extranjero que se ha incorporado a nuestra lengua como un término más en los países hispanohablantes. Su traducción a nuestra lengua sería conjunto de datos y se refiere a una colección de datos habitualmente tabulados. Un conjunto de datos corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos de estadística, donde cada columna representa una variable en particular y cada fila representa a un dato determinado del conjunto de datos que estamos tratando [3].

En el contexto de Big data, se entiende por dataset un conjunto de datos tan grande que las aplicaciones de procesamiento de datos tradicionales (por ejemplo excel) no los pueden procesar debido a la gran cantidad de datos contenidos en la tabla. Una de las ventajas principales de los datasets es que ya tienen una estructura, a diferencia de los RDD, conocidos como conjuntos de datos desestructurados.

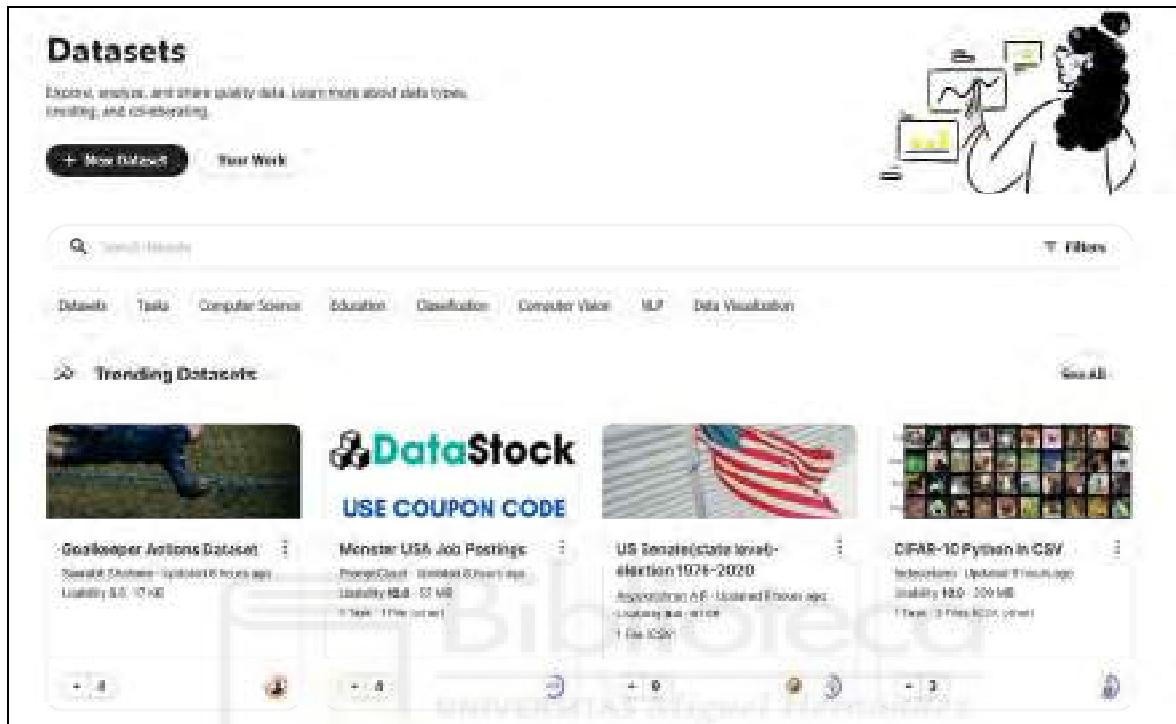


Figura 2.2.- Datasets de Kaggle

Kaggle dispone de una gran cantidad de datasets enviados por numerosos miembros de la comunidad de Data Science con la finalidad de realizar investigaciones, pruebas con diferentes algoritmos y en definitiva, ser útil para cualquiera. Estos datasets son productos de investigaciones de personas (recogiendo los datos de empresas, colegios, hospitales, ...) y normalmente suelen ser los ficheros de tipo CSV, JSON, SQLite, BigQuery [4].

2.2.2.- Moodle

Moodle es una plataforma diseñada para crear y gestionar entornos de formación. Esta plataforma educativa fue desarrollada por el pedagogo e informático Martin Dougiamas, que basó su diseño en las ideas del constructivismo, es decir, el conocimiento no se transmite sino que se construye en la mente del estudiante. El papel del profesor se centra en ayudar a construir ese conocimiento con base en sus habilidades y conocimientos propios.

La primera versión de Moodle se lanzó en el año 2002 y desde entonces no ha dejado de crecer. Con más de 200 millones de usuarios en todo el mundo y casi 160.000 páginas

desarrolladas bajo este software LMS, se puede decir que la plataforma Moodle es el sistema de aprendizaje más popular [5].

2.2.2.1 Tareas

La actividad de tarea de Moodle proporciona un espacio en el que los estudiantes pueden enviar sus trabajos para que los profesores los califiquen y proporcionen retroalimentación. Los envíos de los estudiantes están juntos en una pantalla en su curso. Se les puede pedir que envíen uno o varios archivos y/o que escriban ensayos de texto. Las tareas pueden tener fechas finales y fechas fatales, que pueden extenderse en caso necesario.

Cabe destacar que los estudiantes pueden escribir directamente en Moodle, subir archivos o añadir elementos multimedia. Las opciones están en la sección acerca de Tipos de envíos [6].

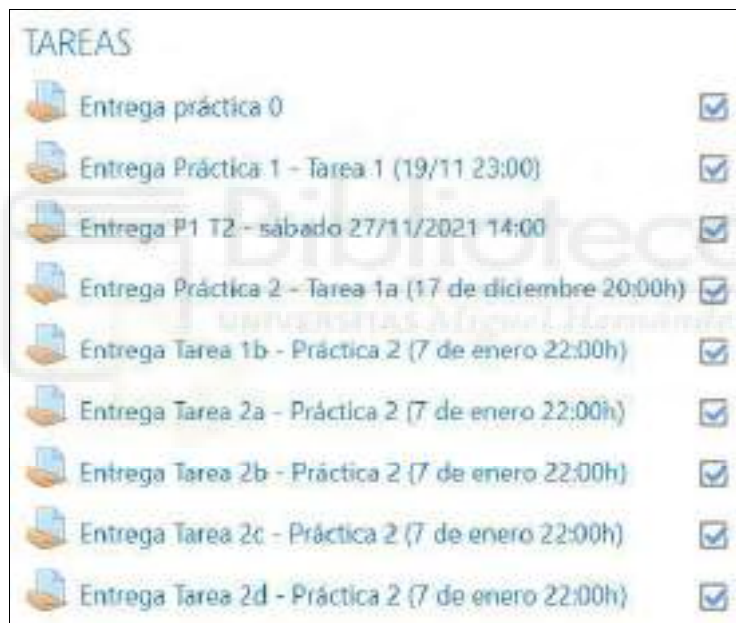


Figura 2.3.- Tareas de Moodle

Como se puede observar en la figura 2.3, Moodle permite crear tareas dentro de un curso con la finalidad de que el usuario pueda realizarlas. En el caso que el usuario realice la tarea, aparecerá un tic en el correspondiente cuadro que hay a la derecha de la tarea.

Si se hace clic en la tarea, aparecerá toda la información necesaria para realizarla. Se puede incorporar un enunciado, imágenes o ficheros. En este ejemplo se adjunta un fichero. También hay apartados que proporcionan información de la entrega al usuario, como “Estado de la presentación”, “Estado de calificación”, “Fecha de vencimiento”, “Tiempo restante”, “Última modificación”, “Envíos de archivo” y por último “Comentarios de envío” (Ver figura 2.4).



Figura 2.4.- Contenido de una tarea de Moodle

2.2.2.2 Roles de usuario

Los roles de usuario son una colección de permisos definida para todo el sistema por la que se puede asignar a usuarios específicos en contextos específicos. La combinación de roles y contexto definen la habilidad de un usuario específico para hacer algo en alguna página [7].

En la plataforma de Moodle existen 8 tipos de roles de usuario diferentes [8]:

- **Administrador del sitio:** Los administradores de Moodle tienen permiso para hacer cualquier cosa. Este rol puede ser asignado por otro administrador, pero el rol en sí mismo no puede ser editado ni eliminado. El administrador primario (creado cuando la plataforma fue creada) no puede ser eliminado del rol de administrador del sitio.
- **Mánager (Gestor):** El rol por defecto de mánager (gestor) permite acceder a cursos y modificarlos, además de realizar ciertos trabajos a nivel administrativo relacionados con cursos, usuarios, configuración de calificaciones, entre otros. A diferencia del rol de administrador, permite editar las capacidades además de ser un rol mucho más seguro para usar debido a los amplios poderes por defecto que tiene un rol de administrador.
- **Creador de cursos:** Los creadores de cursos pueden, como su propio nombre lo sugiere, crear cursos. Tiene la capacidad de editar las configuraciones del curso,

inscribir a otros usuarios, eliminar los cursos y además puede visualizar los cursos ocultos. El rol de creador de curso típicamente puede asignarse a un profesor titular, jefe de departamento o coordinador de programa

- **Profesor:** Los profesores pueden hacer casi todo dentro de un curso, incluyendo el añadir o cambiar las actividades y calificar a los estudiantes. Por defecto, los profesores pueden asignar a otros usuarios el rol de “profesor no editor” y el rol de “estudiante”. Por defecto, a los usuarios no se les asigna el rol de profesor para el contexto de la plataforma completa, sino que solamente son asignados (inscritos/matriculados) como un profesor en un solo curso o clase a la vez. Lo mismo aplica para los estudiantes. Los profesores solamente pueden enseñar en los cursos en donde hayan sido inscritos.
- **Profesor no editor:** Un profesor no editor puede ver y calificar el trabajo de los estudiantes en un curso, pero no puede alterar ni borrar ninguna de las actividades o recursos. Típicamente este rol se le da a un ayudante de profesor.
- **Estudiante:** un estudiante puede participar en actividades del curso y ver recursos, pero no puede alterarlos o ver el libro de calificaciones del grupo. Ellos pueden ver sus propias calificaciones si el profesor lo ha permitido. Los administradores y los profesores determinan cómo se inscribe un estudiante, y qué es lo que pueden hacer o ver en la plataforma de Moodle. Estos permisos pueden variar con cada curso o dentro de cualquiera de los muchos tipos de actividades de Moodle. Por ejemplo, en algunos contextos, los estudiantes pueden corregir o calificar el trabajo de otros estudiantes, ser motivados para explorar todo en un curso e interactuar con otros participantes. En un contexto diferente, el estudiante puede ser guiado hacia una ruta muy definida, con mínima interacción con otros usuarios.
- **Usuario autenticado:** Cuando un usuario entra al sitio y es identificado, se le asigna automáticamente el rol de usuario autenticado. Un usuario tendrá roles adicionales además del rol de usuario autenticado de acuerdo a donde están en Moodle, por ejemplo, estudiante en un curso. Por defecto, los usuarios autenticados tienen permiso para editar su propio perfil, mandar mensajes, escribir en blog y hacer otras cosas además de cursos.
- **Invitado:** Las personas que visitan el sitio pueden entrar como invitados usando el botón para "Entrar/Ingresar como invitado" en la pantalla de entrada al sitio y entrar a cualquier curso que permita el acceso a los invitados. Adicionalmente, los usuarios que hayan ingresado al sitio pueden entrar a cualquier curso que permita el acceso a invitados sin que se les pida inscribirse. Los invitados siempre tienen acceso de “solamente lectura”, por lo que no pueden dejar ninguna publicación, participar en un chat, publicar en foros, enviar una tarea, realizar un examen, recibir calificaciones, entre otros.

2.2.2.3 Calificaciones

La plataforma Moodle permite a los profesores de un curso introducir calificaciones a sus estudiantes, las cuales siempre tendrán un valor comprendido entre 0 y 10, siendo 0 la nota mínima y 10 la nota máxima.

Cuando se realiza un examen tipo test, el profesor tendrá la posibilidad de mostrar la nota automáticamente después de la finalización del examen (corregido de forma automática), mostrar las calificaciones cuando considere oportuno u ocultarlas por tiempo indefinido.

Dentro de una tarea, no existe la posibilidad de que Moodle corrija automáticamente la tarea con una calificación, por lo que será el profesor el encargado de poner dicha nota. Al introducir la calificación desde una tarea tiene la ventaja de que enviará una notificación al estudiante cuando la calificación sea enviada.

Además se puede introducir las calificaciones del estudiante directamente en el libro de calificaciones del curso o modificarlas dentro de la vista individual del estudiante, aunque Moodle no permite que se envíe al estudiante una notificación automáticamente de la nota establecida. El profesor puede poner un porcentaje de peso determinado en cada tarea sobre la nota final y cuando dichas tareas tengan su calificación, Moodle calculará automáticamente la nota final en función de los parámetros de corrección que haya programado el profesor.

2.2.3.- Google Colab

Google Colaboratory, también llamado “Colab” se trata de un servicio cloud basado en Notebooks de Jupyter que permite el uso gratuito de las GPUs y TPUs de Google, con 27 librerías como “scikit-learn”, “PyTorch”, “TensorFlow”, “Keras” y “OpenCV”, entre otros. Todo ello bajo Python 2.7 y 3.6 (Con opción a cambiarlo), R y Scala.

Google Colab es una herramienta ideal para practicar, mejorar los conocimientos en técnicas y herramientas de Data Science y además también permite desarrollar aplicaciones (pilotos) de machine learning y deep learning, sin tener que invertir en recursos de hardware o del Cloud. Con Colab se puede crear cuadernos o importar los que ya se tengan, además de compartirlos y exportarlos cuando se quiera. A la hora de manejar la información también se puede utilizar Google Drive, unidad de almacenamiento local, GitHub e incluso otros sistemas de almacenamiento cloud como por ejemplo el S3 de Amazon.

Una de las ventajas que tiene Google Colab es que permite compartir cuadernos con quien se desee pudiendo ver en tiempo real los cambios que se están realizando, sus ejecuciones y las diferentes versiones creadas. Esto es una función muy útil que ayuda a poder desarrollar modelos y algoritmos más complejos ya que se pueden desarrollar entre varias personas [9].

2.2.3.1.- Jupyter Notebook

Jupyter Notebook proporciona un entorno donde se puede registrar código, ejecutar código, ver resultados, visualizar datos y ver resultados de salida. Estas características lo convierten en una herramienta conveniente para realizar flujos de trabajo de ciencia de datos de extremo a extremo, que se pueden utilizar para la limpieza de datos, el modelado estadístico, la creación y capacitación de modelos de aprendizaje automático, la visualización de datos y muchos otros propósitos [10].

Cuando todavía se está creando un prototipo de proyecto, Jupyter Notebooks es realmente útil, porque su código está escrito en una unidad separada y se ejecuta por separado (Ver figura 2.5). Esto permite a los usuarios probar bloques de código específicos en el proyecto sin tener que ejecutar el código desde el principio del script.

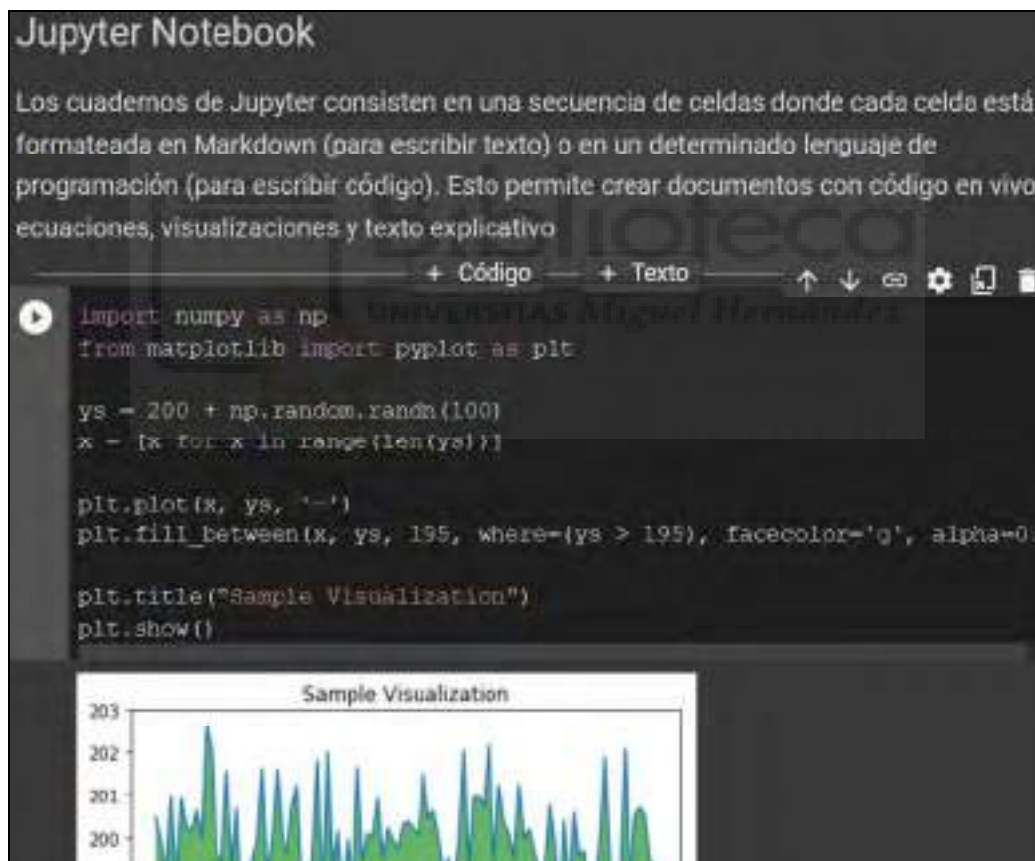


Figura 2.5.- Jupyter Notebook en Google Colab

2.2.4.- DataCamp

DataCamp es una plataforma de autoaprendizaje que trata el área de ciencia de datos, estadísticas y aprendizaje automático. Uno de los puntos fuertes de esta plataforma es la

experiencia de aprendizaje que ofrece al usuario, puesto que dispone de lecciones completamente interactivas de forma virtual y completamente funcional, se dispone de mucho material accesible directamente desde el curso que se está realizando. Dichos cursos disponen de conjuntos de datos con los que se puede experimentar y además están organizados en varias microlecciones, por lo que permite que los usuarios puedan avanzar al ritmo que deseen en función de sus conocimientos previos [11].

2.2.4.1- Espacio de trabajo

El espacio de trabajo de DataCamp, también conocido como Workspace, es un espacio que tiene cada usuario para guardar proyectos, datasets o pruebas realizadas con el objetivo de tener todo el trabajo realizado en un mismo espacio de forma ordenada, eliminando toda la necesidad de buscar entre muchos espacios diferentes, el archivo que se desea encontrar (Ver figura 2.6). Además permite crear un portfolio de perfil en el que se pueda destacar las habilidades que tiene el usuario, mostrar sus méritos y lo que desee [12].



Figura 2.6.- Workspace de DataCamp

2.2.5.- Resumen

En la web se pueden encontrar numerosas herramientas con funcionalidades diferentes con el objetivo en común de prestarle al usuario las mejores funcionalidades posibles. Cuando se

estudia las herramientas que disponen las plataformas mencionadas anteriormente, se aprecia que cada plataforma tiene alguna herramienta que “sobresale” de las otras plataformas que también lo tienen, por lo que se realizará una valoración objetiva de sus pros y contras.

La plataforma Moodle sobresale en las herramientas que permiten la interacción entre profesor - alumno, siendo las herramientas “Tareas” que proporciona el rol de profesor al alumno dentro de un curso, la “Gestión de usuarios” por la existencia de una gran cantidad de roles de usuario y su alta capacidad de edición de los permisos, y por último las “Calificaciones” ya que permite a los profesores realizar calificaciones automáticas de exámenes tipo test, calificaciones con comentarios, aviso al correo electrónico del alumno de tareas y la capacidad de tener calculada la nota final de los alumnos en el curso en función del peso que se le haya dado en porcentaje a cada tarea de la nota final del curso.

Kaggle tiene uno de los repositorios de ciencia de datos de libre acceso más grandes de todo internet, siendo éste el preferido por muchos estudiantes e investigadores de la rama de la ciencia de datos, aunque se debe crear una cuenta para acceder. El repositorio dispone de infinidad de datasets de diferentes temáticas y dentro de ellos aparece un resumen y una gráfica con información de los datos que contiene con el objetivo de una mejor comprensión del usuario. Además dispone de cursos en los que están divididos en varias lecciones con una parte de teoría donde se explica paso a paso el tema de dicha lección y una parte de práctica para que el usuario pueda practicar.

En Google Colab se pueden desarrollar estudios y aplicaciones de ciencia de datos gracias a Jupyter Notebook, puesto que permite al usuario introducir comentarios, registrar código, ejecutar código, ver resultados, visualizar datos y ver resultados de salida. Uno de los puntos fuertes de Jupyter Notebook de Google Colab es que permite integrar más desarrolladores dentro de un mismo cuaderno y la posibilidad de que todos los desarrolladores puedan programar dentro del cuaderno con una sincronización instantánea.

DataCamp tiene una herramienta llamada Workspace (Espacio de trabajo) que permite a los usuarios crear un perfil personalizado con todos sus aportes a la comunidad y logros con la posibilidad de compartirlo por redes sociales. En este espacio de trabajo también estarán todas las pruebas y estudios que ha realizado el usuario para tener un mayor control sobre su progreso con la posibilidad de hacerlo público.

En definitiva, la gran mayoría de las plataformas están en constante evolución y desarrollo, teniendo éstas diferentes herramientas implementadas. En la tabla 2.1 se puede observar una tabla comparativa de la disponibilidad de las diferentes herramientas estudiadas en las plataformas.

FUNCIONES	HERRAMIENTAS			
	Moodle	Kaggle	Colab	DataCamp
Tareas	✓	✗	✗	✗
Cursos	✓	✓	✗	✓
Espacio de trabajo	✗	✓	✓	✓
Jupyter Notebook	✗	✓	✓	✓
Repositorio de datos	✗	✓	✗	✗
Gestión de usuarios	✓	✗	✓	✓
Calificaciones	✓	✗	✗	✗

Tabla 2.1.- Resumen comparativo de herramientas estudiadas

2.3.- Análisis de otros entornos de desarrollo

A pesar de que en la actualidad existen infinidad de herramientas útiles para la realización de cualquier tipo de proyecto, también hay herramientas de libre acceso que permiten su utilización, divulgación y modificación sin tener que pedir permiso, crear una cuenta o pagar dinero.

Para la realización de este proyecto, se necesita la incorporación de un entorno de desarrollo de programación de libre acceso que permita la realización de ejercicios de programación sin salir de la plataforma.

Mediante un proceso de investigación se han encontrado 3 entornos de desarrollo útiles capaces de codificar código python para ejercicios de ciencia de datos. A continuación se expondrán los puntos fuertes y débiles de dichas herramientas.

2.3.1 Consola Pyodide

Pyodide fue creado en 2018 por Michael Droettboom en Mozilla como parte del proyecto Iodide. Iodide es un entorno de cuaderno experimental basado en la web para la comunicación y la computación científica [13].

Cabe destacar que se trata de un proyecto de código abierto independiente e impulsado por la comunidad siendo el proceso de toma de decisiones descrito en la gobernanza del proyecto. Utiliza la versión 2 de la licencia pública de Mozilla [14].

Esta herramienta es fácil de instalar, ligera y muy eficiente, pero uno de sus puntos débiles es su interfaz, ya que no es fácil de utilizar para usuarios principiantes y además no están disponibles las gráficas ni la incorporación de ficheros (Ver Figura 2.7). Se utiliza únicamente con codificación, sin tener opción a hacer clic en ninguna parte.

```
Welcome to the Pyodide terminal emulator 🐍
Python 3.9.5 (default, Mar 14 2022 21:16:01) on WebAssembly VM
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Una operacion realizada en Pyodide es:",3+4*2)
Una operacion realizada en Pyodide es: 11
>>>
```

Figura 2.7.- Consola de Pyodide en el navegador

2.3.2 Cuaderno RetroLab

RetroLab [15] es una distribución alternativa a jupyterLab, creada desde cero, que proporciona una interfaz con un aspecto retro y utiliza el kernel de JupyterLite (Ver figura 2.8).

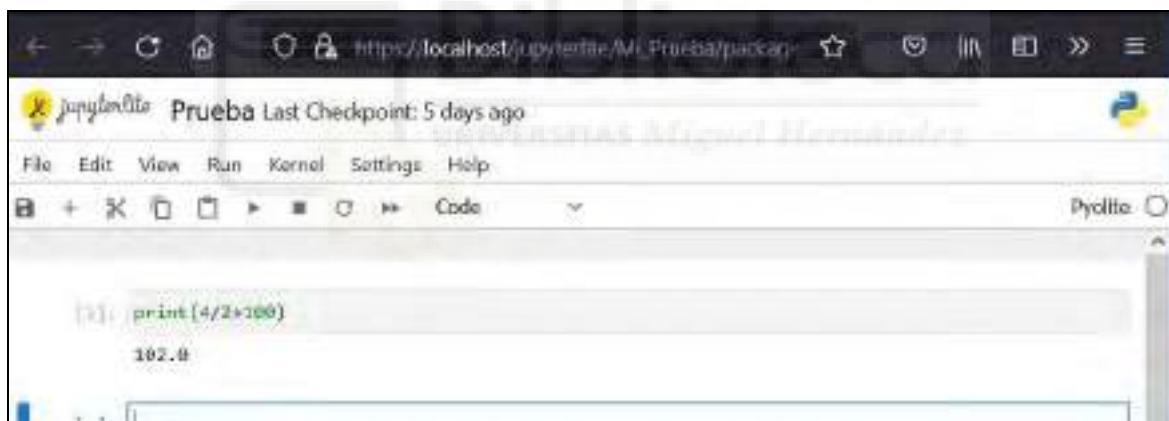


Figura 2.8.- Interfaz de RetroLab

JupyterLite es el resultado de numerosos intentos de hacer una distribución de Jupyter estática completa que se ejecuta en el navegador, sin tener que iniciar “Python Jupyter Server” en la máquina host, generalmente ejecutando JupyterLab o Jupyter Notebook en un terminal. El objetivo principal es proporcionar un entorno de programación ligero y accesible en cuestión de segundos en un navegador web sin la necesidad de instalar nada.

JupyterLite trabaja con Pyodide, que consta del intérprete CPython 3.8 compilado en WebAssembly que permite que Python se ejecute en el navegador. Actualmente está en fase “experimental” por lo que van introduciendo mejoras y nuevas funcionalidades en versiones posteriores.

2.3.3 Cuaderno Basthon

El nombre Basthon [16] viene del acrónimo “Bac À Sable pour pyTHON” ya que parece que se diga “baston” (significado de pelea en francés) haciendo alusión a la “lucha” que en ocasiones puede representar el aprendizaje de la programación, la escritura del código o su depuración.

Basthon es un kernel que esencialmente interpreta código que introduce el usuario, ya sea Python, JavaScript, SQL, OCaml o Bash. Dispone de dos tipos de interfaces diferentes, siendo “Basthon-Console” una interfaz de consola (Ver figura 2.9) y “Basthon-Notebook” una interfaz experimental basada en JupyterLab (Ver figura 2.10).



Figura 2.9.- Interfaz de la consola de Basthon

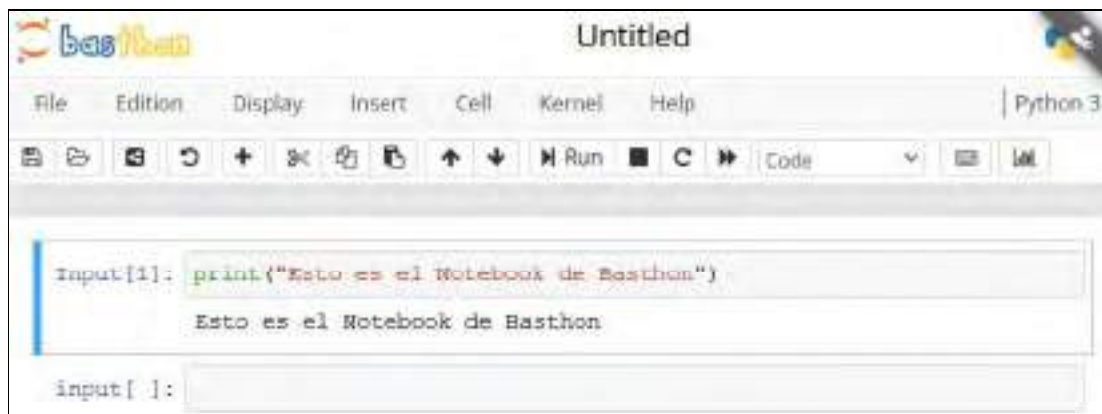


Figura 2.10.- Interfaz del cuaderno de Basthon

Basthon se utiliza para aprender el lenguaje de programación Python 3 sin tener que instalar nada. Todo lo que necesita es un navegador actualizado (Firefox, Chrome/Chromium o Edge)

y una conexión a Internet para acceder a las librerías depositadas en diversos repositorios. El navegador ejecuta completamente el código de Basthon, en el lado del cliente. Basthon fue construido completamente usando software libre y se coloca bajo la licencia gratuita GNU GPL versión 3 [17] o cualquier versión posterior.

La ejecución del código de Python está encomendada a Pyodide, una compilación del intérprete de referencia de Python (CPython) en WebAssembly, por lo que no es necesario tener instalado python en el equipo. Se han realizado algunos cambios para poder usar, entre otros, P5.js, Matplotlib, Sympy, Pandas, Turtle, Folium, Requests, Proj4.js, Graphviz a través del módulo graphviz usando viz.js y PythonTutor. Turtle no está implementado en Pyodide, siendo ésta una versión modificada de la implementación de Brython que se usa en Basthon. A continuación se explicará brevemente las tecnologías y librerías mencionadas anteriormente:

- **OCalm:** Es un lenguaje de programación avanzado de la familia de los lenguajes ML(Metalenguaje). Este lenguaje admite los paradigmas de programación imperativa, programación funcional y programación orientada a objetos. Se compila en código para una máquina virtual o en código de máquina para diferentes arquitecturas y dispone de un análisis de tipos estáticos con inferencia de tipos, con valores funcionales de primera clase, polimorfismo parametrizado, llamada por patrones, manejo de excepciones, recolección de basura y otras características avanzadas [18].
- **Bash:** Es un intérprete de comandos que ejecuta las instrucciones introducidas por el usuario o contenidas en un script y devuelve los resultados. Es decir, actúa como interfaz entre el kernel Linux y los usuarios o programas del modo texto. Además, incorpora numerosas utilidades de programación y mejoras sobre sh, su shell predecesora. Debido a que es una herramienta desarrollada por GNU, suele ser utilizada por defecto en las distros actuales. Por otro lado, cabe destacar que Bash no es una terminal y además tampoco es la única shell disponible[19].
- **p5.js:** Es una biblioteca de JavaScript para la programación creativa, que busca hacer que programar sea accesible e inclusivo para todo tipo de programadores [20].
- **Matplotlib:** Es una librería de Python especializada en la creación de gráficos en dos dimensiones [21].
- **Sympy:** Es una biblioteca de Python para matemáticas simbólicas. Su propósito es llegar a ser un sistema de álgebra por computadora (CAS) completo manteniendo el código tan simple como sea posible para poder ser legible y extensible de manera fácil. SymPy está escrito en Python enteramente [22].
- **Pandas:** Es una librería muy popular de código abierto dentro de los desarrolladores de Python, y sobre todo dentro del ámbito de Data Science y Machine Learning, ya que ofrece unas estructuras muy poderosas y flexibles que facilitan la manipulación y

tratamiento de datos. Esta librería permite cargar, modelar, analizar, manipular y preparar datos [23].

- **Turtle:** Es un módulo de programación gráfica para Python utilizado como método para enseñar programación a través de coordenadas relativas [24].
- **Folium:** Es una poderosa herramienta de visualización ya que esta biblioteca permite la concepción de mapas interactivos [25].
- **Request:** Es una librería Python que facilita enormemente el trabajo con peticiones HTTP [26].
- **Proj4.js:** Es una biblioteca de JavaScript que se centra en la expresión y conversión de la proyección del mapa. Usando una expresión de proyección muy simple y clara, PROJ4 es más simple y más obvia que otras definiciones de proyección. Es fácil ver los parámetros de varios sistemas de coordenadas geográficas y proyecciones de mapas y además tiene una poderosa función de conversión de coordenadas [27].
- **Graphviz:** Es un conjunto de herramientas open-source realizado inicialmente en los laboratorios de investigación de AT&T para el dibujo de gráficos especificados en lenguaje de scripts DOT. Provee librerías para ser usadas por otras aplicaciones. Graphviz es un software libre licenciado bajo CPL (Common Public License) [28].

Basthon-Notebook es una bifurcación (una copia) de Jupyter-Notebook. Gran parte del código detrás de la interfaz del portátil de Basthon proviene de Jupyter. Solo la interfaz con Basthon-Kernel y algunas funcionalidades extra fueron codificadas por el autor de Basthon.

Basthon respeta la privacidad de manera rigurosa, puesto que no se registran datos personales ni se transmiten a través de la red (ni siquiera el código que se escribe), no requiere registro, no hay publicidad, no se recopilan datos estadísticos y no se guardan las cookies. Lo único que se guarda es el último código ingresado y la apariencia de la pantalla, los cuales se guardan en el almacenamiento local del navegador (IndexedDB) [29], por lo que permite a Basthon encontrar el estado en que se quedó el programa después de un fallo en la red, una actualización de la página o el cierre del navegador.

Al comienzo del proyecto, Basthon estaba usando Brython [30]. Varios errores y regresiones hicieron que el autor recurriera a Skulpt [31]. Desafortunadamente, Skulpt no es totalmente compatible con Python 3 y, especialmente, no permite evaluar una expresión de Python, lo que impide su uso en un cuaderno. Fue entonces cuando Pyodide se convirtió en el intérprete de Bashon. Es mucho más pesado que los otros dos pero su soporte para Python es mucho más completo.

2.4.- Análisis de editores de código online

Gracias a los avances de la tecnología informática que está evolucionando constantemente, los usuarios pueden crear un sitio web simple sin necesariamente tener conocimientos particulares de programación, sin embargo, si se quiere realizar un desarrollo de cualquier página web o cualquier otro tipo de programa, entonces es muy recomendable contar con un editor de código.

Los editores de código son herramientas que permiten editar código fuente en diversos lenguajes de programación y ofrecen múltiples características, como por ejemplo el resaltado de la sintaxis, autocompletado y pareo de llaves con el objetivo de facilitar el trabajo y acelerar la escritura del código fuente [32].

2.4.1 CodeAnywhere

CodeAnywhere es un entorno de desarrollo web totalmente online que ofrece numerosas funcionalidades como el autocompletado, la posibilidad de vincular servicios de almacenamiento en la nube para hospedar los archivos que se están editando, ya sea Dropbox, Google Drive o Github [33]. Permite además utilizar los lenguajes de programación PHP, HTML5, Java, JavaScript, Python, Ruby, C++, entre otros (Ver figura 2.11).

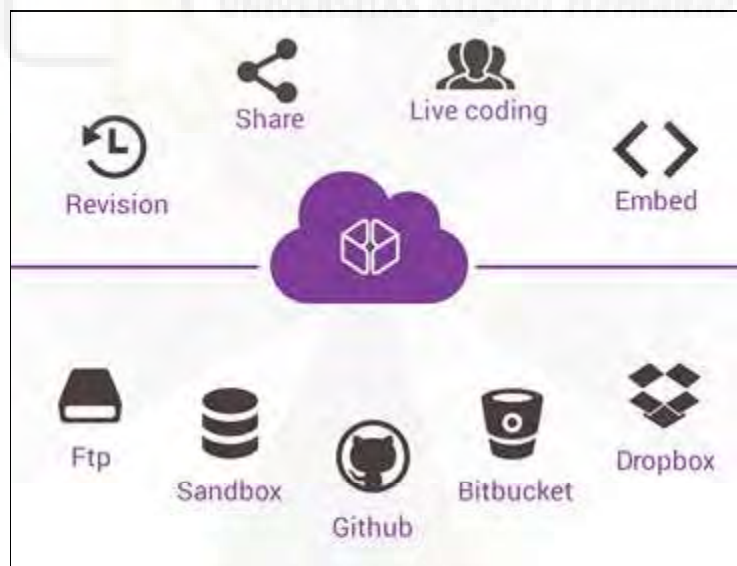


Figura 2.11.- Funcionalidades que ofrece CodeAnywhere

Otra de las opciones que ofrece este editor de código online es la DevBox, en la que se podrá hacer uso de una máquina virtual en la nube con 1 GB de almacenamiento, 256 MB de RAM, acceso a través de la terminal SSH y acceso a MySQL.

La desventaja principal de CodeAnywhere es que la versión gratuita no ofrece muchas funcionalidades, por lo que serían insuficientes sus prestaciones en muchos casos cuando se vaya a realizar un desarrollo de software.

2.4.2 Prism.js

Prism es una librería de JavaScript ligera y extensible construida con estándares web que permite el resaltado de sintaxis de código de programación. Para la utilización de esta librería se debe llamar a “prism.css” y “prism.js” utilizando las etiquetas HTML5 apropiadas (Ver figura 2.12), y en la sección del código donde se desee llamar a la clase de Prism, se deberá llamar a la clase correspondiente [34].



```
<!DOCTYPE html>
<html>
<head>
...
<link href="themes/prism.css" rel="stylesheet" />
</head>
<body>
...
<script src="prism.js"></script>
</body>
</html>
```

Figura 2.12.- Llamada a la librería Prism desde HTML

El código es extremadamente ligero, utilizando únicamente 1.6 KB estando comprimido y por cada lenguaje que se le quiera añadir, se agregará entre 0.3 y 0.4 KB en promedio. Ello hace que vaya muy rápido y además admite el paralelismo a través de Web Workers. Cabe destacar además que las clases CSS del lenguaje son heredables, por lo que sólo se necesita definir las una vez para aplicarlas a varios fragmentos de código.

2.4.3 Editor de Mónaco

Monaco Editor es un editor de código en línea de código abierto desarrollado por Microsoft. Para su uso se deberá llamar a su API y habrá que tener en cuenta que es compatible con los navegadores Edge, Chrome, Firefox, Safari y Opera, pero no es compatible con los navegadores de móvil o frameworks de móvil [35].

Este editor de código permite 28 lenguajes de programación, entre los que están JavaScript, CSS, LESS, C#, R, Python, Java y PHP. También permite realizar una comparación entre 2 códigos en vivo (Ver Figura 2.13).



Figura 2.13.- Comparación de 2 códigos en Editor de Mónaco

Una de las ventajas principales es que es de código abierto y tiene una gran comunidad, por lo que es más sencillo consultar y resolver dudas. Otra de las ventajas que destaca este editor de código con respecto a otros es de su funcionalidad “Monaco Speech Editor” la cual está creada para personas con discapacidad visual. Proporciona funciones de asistencia de voz enriquecidas, que pueden localizar y leer con precisión el contenido especificado por el usuario. Y también proporciona una variedad de modos de voz, como el modo de personaje, el modo de música y el modo global. Estos modos pueden proporcionar una salida de contenido de voz diferenciada de acuerdo con las diferentes necesidades de los usuarios en diferentes etapas de programación. Además, tiene una interfaz de operación incorporada para ciegos, de modo que los ciegos pueden usar el editor normalmente sin usar la pantalla de un ordenador y el ratón [36].

2.4.4 CodeMirror

CodeMirror es un editor de código fuente que funciona en el navegador, ya que está implementado en JavaScript y da soporte a numerosos lenguajes de programación y addons para añadirle funcionalidades de edición más avanzadas. Es de código abierto y su desarrollo y seguimiento de bugs se realizan a través de la página del proyecto en GitHub, por lo que cualquier usuario puede aportar al proyecto [37].

Uno de los puntos fuertes de CodeMirror es que es compatible con todos los navegadores actuales que soportan HTML5. Otro punto fuerte es la cantidad de funcionalidades que proporciona: Soporte de 60 lenguajes de programación, autocompletado en XML, asociaciones de teclas configurables, emparejamiento de llaves y etiquetas, entre otras muchas funcionalidades y addons (Ver figura 2.14).

```
<!-- Create a simple CodeMirror instance -->
<link rel="stylesheet" href="lib/codemirror.css">
<script src="lib/codemirror.js"></script>
<script>
  var editor = CodeMirror.fromTextArea(myTextArea, {
    lineNumbers: true
  });
</script>
```

Figura 2.14.- Ejemplo de código para el uso de CodeMirror

Cabe destacar que las herramientas para desarrolladores de Firefox y Chrome usan el editor CodeMirror.

2.5.- Resumen de entornos y editores de código

Una vez analizados los entornos de desarrollo y los editores de código, se deberá hacer una reflexión y escoger una de estas herramientas para su uso. Primero se hará una comparativa entre los entornos de desarrollo y se escogerá el más apropiado, después se hará otra comparativa con los editores de código y se escogerá de nuevo el más apropiado, y por último, se explicarán los motivos por los que se elige una de estas dos herramientas más apropiadas descritas anteriormente para la realización de este proyecto.

Por un lado, la consola de Pyodide es muy ligera y carga todas las librerías necesarias de forma muy rápida, aunque una gran desventaja sería la ausencia de incorporación de ficheros de datos para realizar estudios y su interfaz de usuario, puesto que se trata simplemente de una consola y puede ser difícil el manejo para usuarios poco experimentados. Por los motivos mencionados anteriormente no se incorporará la consola de Pyodide al proyecto.

Por otro lado, los cuadernos RetroLab y basthon son bastante similares, ya que ambos comparten el mismo kernel de JupyterLite, aunque existen ciertas diferencias entre ambos cuadernos que permitirán saber qué opción es la mejor.

Basthon y RetroLab tienen las mismas funcionalidades, ya que ambos parten del mismo kernel, sin embargo el desarrollador de Basthon le añadió funcionalidades extra que son bastante útiles. En primer lugar se añadió a las gráficas una capa extra con botones que permite cambiar la visión de la gráfica haciendo zoom, moverla de lugar y descargarla en formato PDF, PNG y SVG (Ver Figura 2.15). Los botones extra mencionados anteriormente son realmente útiles y usables ya que permite al usuario de una forma sencilla descargar en el formato deseado las gráficas creadas mediante código.

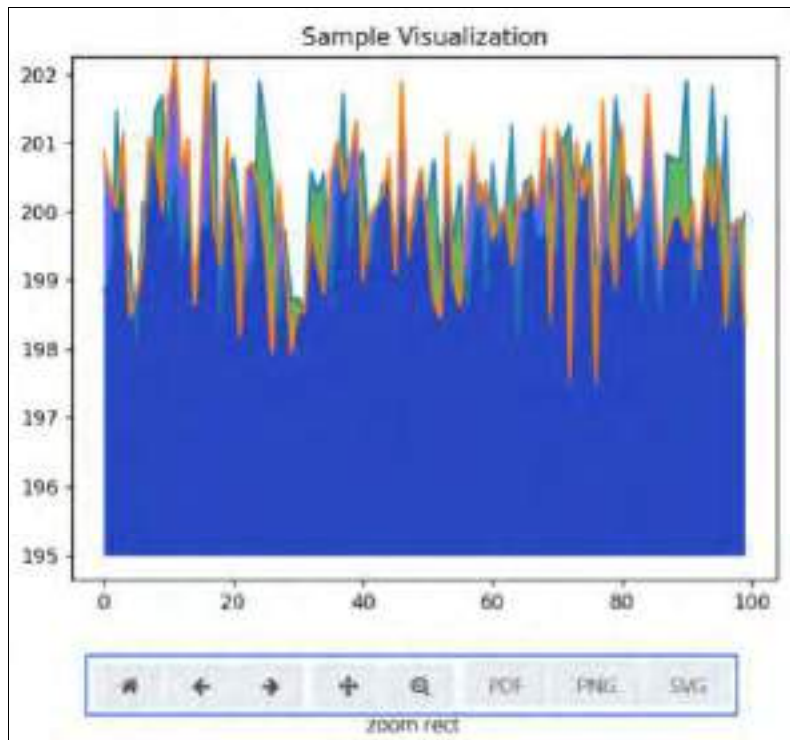


Figura 2.15.- Botones en gráfica del cuaderno Basthon

Otra funcionalidad extra de Basthon es la pizarra, ideal para la enseñanza de profesores a alumnos, puesto que permite escribir con un lápiz, borrar, redondear y manipular cualquier parte del cuaderno, simulando un lápiz virtual (Ver figura 2.16). Su uso es muy sencillo, si se mantiene pulsado el botón izquierdo y se desplaza el ratón, se estará dibujando con el lápiz y si hay que borrar, se realizará de la misma manera pero manteniendo pulsado el botón derecho del ratón. En el caso que no se pulse ningún botón, no se pintará ni borrará nada.

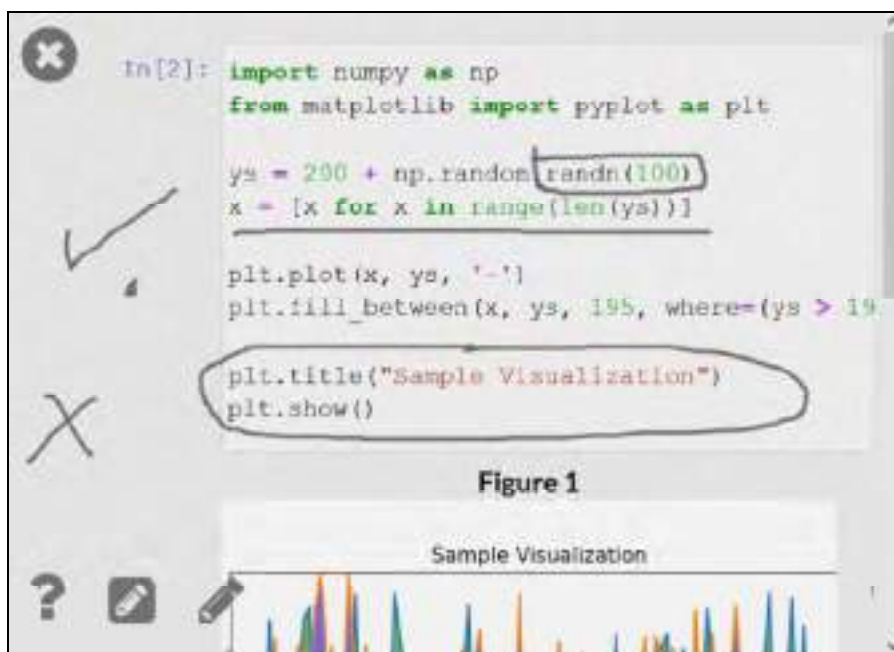


Figura 2.16.- Pizarra del cuaderno Basthon

RetroLab permite volver a la situación anterior con respecto al último cambio en el caso que el usuario se haya equivocado y lo desee, aunque no tiene un “historial” de versiones el cual se pueda elegir el estado del cuaderno que hubo en el pasado. Basthon sí permite esta funcionalidad (Ver Figura 2.17).



Figura 2.17.- Historial de versiones del cuaderno Basthon

Una última diferencia apreciada entre RetroLab y Basthon es la organización de los ficheros que conjuntamente forman el kernel. Esta diferencia no es apreciada por el usuario, pero sí por el desarrollador y es algo importante ya que permite tener una curva de aprendizaje algo menos inclinada.

Por todos los motivos descritos anteriormente, el entorno de desarrollo más apropiado de momento para usar en el proyecto será Basthon. Por otro lado, se debe analizar además los diferentes editores de código descritos anteriormente.

CodeAnywhere es un entorno de desarrollo muy completo, aunque quedaría automáticamente descartado al tratarse en su gran mayoría de funcionalidades de pago. Prism es una librería JavaScript muy ligera y fácil de utilizar, aunque no ofrece tantas funcionalidades como otros editores de código. El editor de Mónaco tiene mucho potencial por su simplicidad a la hora de su uso, por la gran comunidad que tiene y además por las posibles actualizaciones que realice Microsoft, aunque uno de sus puntos negativos es la incompatibilidad con los navegadores de móvil y frameworks, por lo que si se quiere desarrollar una aplicación, no podrá ser usada con ningún dispositivo móvil. Por último se tiene el editor de código CodeMirror el cual presenta muchas funcionalidades, acepta más de 60 lenguajes de programación, es compatible con todos los navegadores y tiene una gran comunidad. CodeMirror se trataría del editor de código más adecuado para la realización del proyecto.

Por último, se llega a la conclusión que entre el entorno de desarrollo y el editor de código más apropiado, Basthon y CodeMirror respectivamente, se utilizará en este proyecto

CodeMirror por su simplicidad y flexibilidad, ya que Basthon es bastante más complejo y además está en fase experimental, por lo que hay muchas funcionalidades que aún no se han implantado.



Capítulo 3

Hipótesis de trabajo

Los modelos y tecnologías de desarrollo web han evolucionado mucho en la última década. Existen multitud de aplicaciones, frameworks, librerías y sistemas de publicación en diferentes versiones que a su vez reciben cambios o mejoras con el tiempo. El progreso también ha tenido lugar en lo relacionado con la administración de sistemas, servicios de alojamiento, técnicas de escalabilidad, monitorización y gestión de centros de procesos de datos.

Esta evolución ha dado lugar a la convergencia de una gran cantidad de tecnologías, herramientas y estilos arquitectónicos para desarrollar sitios web y aplicaciones. A continuación se explicarán todas las herramientas de desarrollo que se van a utilizar para el desarrollo de la plataforma [38].

La arquitectura interna que sigue este proyecto se basa en el patrón MVC (Modelo-Vista-Controlador), en el cual se realiza una separación entre la lógica de negocios y su visualización. Esto proporciona una mejor división del trabajo y mejora el mantenimiento

ya que permite una mejor reutilización del código y la separación de conceptos. Para ello se construyen 3 componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado se definen los componentes para la representación de la información y por otro lado para la interacción del usuario, de modo que si se realiza un cambio en alguno de ellos, no afectará a los otros componentes.

En este patrón, el modelo es la capa que define qué datos debe contener la aplicación. Si el estado de esos datos cambia, el modelo generalmente notificará a la vista para que la pantalla pueda cambiar según lo necesario, y a veces, se notificará al controlador si se necesita una lógica diferente para controlar la vista actualizada. La vista define cómo se deben mostrar visualmente los datos de la aplicación y el controlador tiene una lógica que actualiza el modelo y la vista en respuesta a las entradas de los usuarios en la aplicación [39].

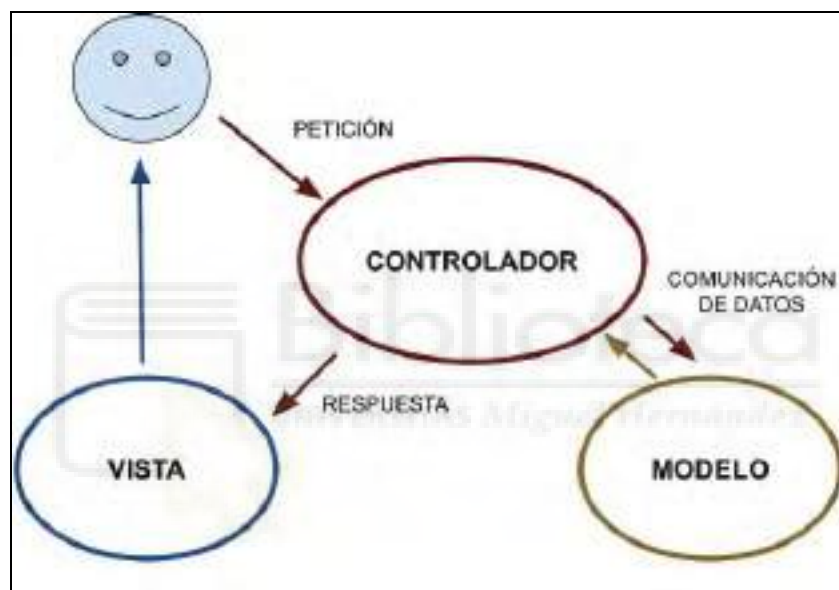


Figura 3.1.- Esquema del patrón MVC

En la Figura 3.1 se puede apreciar el flujo de control que sigue generalmente el patrón MVC. Primero el usuario interactúa con la interfaz de usuario, ya sea escribiendo un texto, haciendo clic en algún botón, entre otros. Después el controlador recibe la notificación de la acción solicitada por el usuario y gestiona el evento accediendo al modelo y actualizándose de forma adecuada a la acción que el usuario ha solicitado y por último el controlador manda una señal a la vista para que modifique la interfaz de usuario. Una vez la vista haya actualizado la interfaz de usuario, se quedaría a la espera de nuevas instrucciones del usuario, por lo que se volverían a realizar los mismos pasos descritos anteriormente.

3.1.- TECNOLOGÍAS DEL LADO DEL CLIENTE

Las tecnologías del lado del cliente son aquellas que permiten crear interfaces de usuario y establecer comunicación con el servidor, actuando el navegador como intérprete. No se

necesita un pretratamiento para que el navegador pueda “entender” la tecnología, ya que la interpreta.

En el desarrollo de aplicaciones web, los lenguajes de programación del lado del cliente son utilizados para la creación de contenidos estáticos y de contenidos dinámicos cuando se utilizan metodologías como AJAX, en el que el propio cliente solicita y procesa datos obtenidos desde el servidor [40].

3.1.1.- HTML

HTML es el lenguaje de marcado estándar que se usa para crear páginas y aplicaciones web. Sus elementos forman los bloques de creación de las páginas y representan texto con formato, imágenes, entradas de formulario y otras estructuras. Cuando un explorador realiza una solicitud a una dirección URL, con independencia de que se obtenga una página o una aplicación, lo primero que se devuelve es un documento HTML. Este documento HTML puede hacer referencia o incluir información adicional sobre su apariencia y diseño en forma de CSS, o el comportamiento en forma de JavaScript [41].

Este lenguaje de programación se ha utilizado para realizar toda la parte de la vista, es decir, crear todo aquello que el usuario puede ver en la página web. Se ha utilizado para crear botones, títulos, contenido, enlazar archivos JavaScript y css, añadir imágenes de fondo, crear las diferentes páginas y la estructura de todas las partes del front end (Header, main, nav y footer).

3.1.2.- CSS

CSS (Hoja de estilos en cascada) se usa para controlar la apariencia y el diseño de los elementos HTML. Los estilos CSS se pueden aplicar directamente a un elemento HTML, o bien definirse por separado en la misma página o en un archivo independiente al que la página haga referencia. Los estilos se aplican en cascada en función de cómo se usan para seleccionar un elemento HTML determinado. Por ejemplo, es posible que un estilo se aplique a todo el documento, pero que se reemplace por un estilo que se aplica a un elemento determinado. Del mismo modo, un estilo específico del elemento se reemplazaría por un estilo que se aplica a una clase CSS aplicada al elemento, que a su vez se reemplazaría por un estilo destinado a una instancia específica de ese elemento [42].

Es recomendable mantener los estilos en sus propios archivos de hoja de estilos independientes y aplicarlos en cascada en función de la selección para implementar estilos coherentes y reutilizables dentro de la aplicación. Se debe evitar colocar reglas de estilo en el código HTML y aplicar estilos a elementos individuales específicos.

3.1.3.- JavaScript

JavaScript es un lenguaje de programación interpretado y dinámico que se ha estandarizado en la especificación del lenguaje ECMAScript. Es el lenguaje de programación de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes, entre otros. JavaScript se puede definir como atributos dentro de los elementos HTML, como bloques de script dentro de una página o en archivos independientes. Al igual que CSS, se recomienda organizar JavaScript en archivos independientes y, en la medida de lo posible, mantenerlos separados del código HTML que se encuentra en las páginas web individuales o vistas de la aplicación [43].

Este lenguaje de programación se ha utilizado para crear la plataforma dinámica, es decir, que permita mostrar, crear o eliminar elementos o datos en base a una petición en ese mismo instante.

3.1.4.- AJAX

El término técnico AJAX es un acrónimo del inglés (Asynchronous JavaScript and XML) y no se trata de un lenguaje de programación, sino una técnica basada en la tecnología JavaScript que cambia la comunicación con el servidor y acelera las aplicaciones web. Las interfaces funcionan más rápido con transferencia de datos retardada (asíncrona). Con AJAX, las aplicaciones web pueden intercambiar datos con el servidor en segundo plano sin necesidad de recargar toda la página [44].

Esta técnica actúa entre el navegador y el servidor garantizando automáticamente que se muestren grandes partes de la página HTML, mientras que las solicitudes de los usuarios se gestionan en el servidor en segundo plano. La interfaz de usuario se actualiza según sea necesario pieza por pieza, por lo que no es necesario recargar todo el contenido de una web con cada clic, sino sólo los datos realmente necesarios.

Las aplicaciones web tradicionales almacenan grandes cantidades de datos para cada entrada de usuario de una sola vez, de modo que las pausas y las esperas se producen durante la ejecución. Mientras que AJAX almacena datos en buffers repetidamente, de modo que sólo hay que volver a cargar partes de la interfaz de usuario y se pierde muy poco durante las interrupciones de la conexión. Anteriormente, la interacción entre los servidores y los usuarios estaba dominada por largas peticiones desde el navegador al servidor y los usuarios tenían que esperar extensos procesos de almacenamiento de datos para obtener la respuesta.

La técnica AJAX elimina cualquier retraso. Funciona directamente en el navegador web y ejecuta operaciones como la validación de datos sin tener que consultar al servidor cada vez. Puede cambiar la interfaz de usuario, incluso cuando envía y recibe datos del servidor. Por lo tanto, el intercambio de datos con el servidor es independiente de la entrada del usuario [45].

3.1.5.- JSON

JSON es un formato que almacena información estructurada y se utiliza principalmente para transferir datos entre un servidor y un cliente. Se trata de una alternativa más simple a XML. Los desarrolladores que trabajan con AJAX utilizan JSON, ya que ambos formatos juntos funcionan muy bien para lograr la carga asincrónica de los datos almacenados, lo que permite que el sitio web pueda actualizar su información sin actualizar la página [46].

En las reglas sintácticas de JSON existen dos tipos de elementos [47]:

- El objeto: Los objetos son listas de parejas nombre / valor. El nombre y el valor están separados por dos puntos (:) y las parejas están separadas por comas. Los objetos se escriben entre llaves ({}), y los nombres de las parejas se escriben siempre entre comillas dobles (") (Ver figura 3.2).

```
{ "nombre": "Alfredo Lopez", "edad": 25, "carnet de conducir": true }
```

Figura 3.2.- Estructura del objeto en JSON

- El array: Se trata de una lista ordenada de valores separados por comas. Los arrays se escriben entre corchetes (Ver figura 3.3).

```
[ 2, "Alfredo", 2.86, "Lopez Hernandez" ]
```

Figura 3.3.- Estructura del array en JSON

Cabe destacar que los espacios en blanco y los saltos de línea no son significativos, es decir, puede haber cualquier número de espacios en blanco o saltos de línea separando cualquier elemento o símbolo del documento con el objetivo de una mayor comprensión (Ver figura 3.4).

```
[  
  {  
    "nombre": "Alfredo Lopez",  
    "edad": 26,  
    "carnet de conducir": true  
  },  
  {  
    "nombre": "Marta Muñoz",  
    "edad": 14,  
    "carnet de conducir": false  
  }  
]
```

Figura 3.4.- Estructura del array con espacios en JSON

3.1.6.- CodeMirror

CodeMirror es una librería JavaScript que permite editar el código fuente implementado y está soportado por todos los navegadores actuales que permiten el uso de HTML5. Es muy completa ya que incluye funciones bastante útiles como por ejemplo poder colorear el código de acuerdo al lenguaje de programación que se está utilizando, indentar o tabular el código para que quede ordenado el código o la función de autocompletado de código [48].

Cabe destacar que dispone de una API de programación y temas CSS para personalizar CodeMirror según lo deseado y añadirles nuevas funcionalidades. Además una de las ventajas es que es Open Source (Código abierto), por lo que se puede utilizar para cualquier fin, seguir el desarrollo y realizar un seguimiento de los bugs a través de la página del proyecto en Github.

3.1.7.- Pyodide

Pyodide es una distribución de python para el navegador y Node.js basada en WebAssembly y gracias a micropip se permite instalar Pyodide y ejecutar los paquetes de python en el propio navegador. Admite cualquier paquete de python, como por ejemplo paquetes de uso general (regex, PyYAML, lxml) y paquetes científicos de python (NumPy, pandas, SciPy, Matplotlib, scikit-learn) y además muchos paquetes con extensiones C [49].

Pyodide viene con una robusta interfaz de función externa para que se puedan mezclar libremente los lenguajes de JavaScript y Python en un mismo código con una “fricción” mínima. Esto además incluye soporte completo para el manejo de errores, entre otros. Cuando se utiliza Pyodide en un navegador, Python tiene acceso completo a las APIs Web.

Si todo lo que Pyodide pudiera hacer es ejecutar el código de Python y escribir de forma estándar, sería una práctica muy sencilla, pero no sería una herramienta práctica para el trabajo real. El poder real proviene de su capacidad para interactuar con las API del navegador y otras bibliotecas de JavaScript a un nivel muy fino. WebAssembly ha sido diseñado para interactuar fácilmente con el lenguaje JavaScript que se ejecuta en el navegador. Dado que se compila el intérprete de Python en WebAssembly, también tiene una integración profunda con el lado de JavaScript [50].

Una de las ventajas para utilizar el navegador en vez de un kernel remoto, como lo hace Jupyter para realizar cálculos de ciencia de datos, es que las visualizaciones interactivas no tienen que comunicarse a través de una red para reprocesar y volver a mostrar sus datos, por lo que se reduce en gran medida la latencia, es decir, el tiempo de ida y vuelta que transcurre desde que el usuario mueve el mouse hasta que se muestra un gráfico actualizado en la pantalla.

Pyodide proporciona además un “puente” entre los ámbitos JavaScript y Python que permite reemplazar JavaScript y usar Python como el lenguaje principal para el desarrollo web. Se puede acceder al alcance JavaScript desde Python a través del módulo “js”. Este módulo da acceso al objeto global “window” y permite manipular directamente el DOM y acceder a las variables y funciones globales desde Python. Además se puede ir en la dirección opuesta y obtener un acceso completo al alcance de Python desde JavaScript a través del objeto “pyodide.globals”. Si se quiere importar la librería “numpy” de Python, se puede usar inmediatamente desde JavaScript, por lo que permitiría a los desarrolladores que prefieren JavaScript aprovechar las bibliotecas de Python [51].

3.1.8.- WebAssembly

WebAssembly (WASM) es un nuevo tipo de código que puede ser ejecutado en navegadores modernos. Es un lenguaje de bajo nivel, similar al lenguaje ensamblador, con un formato binario compacto que se ejecuta con rendimiento casi nativo y provee un objetivo de compilación para lenguajes como C/C++ y Rust que les permite correr en la web. También está diseñado para correr a la par de JavaScript, permitiendo que ambos trabajen juntos. El inconveniente de JavaScript es que es relativamente lento y, en determinados escenarios, se ve limitado. Por eso, el World Wide Web Consortium (W3C) ha impulsado el método WASM [52].

Cabe mencionar que los navegadores web no tienen un sistema de archivos (un lugar para cargar y guardar archivos). Afortunadamente, WebAssembly proporciona un sistema de archivos virtual, escrito en JavaScript, que el intérprete de Python puede usar. De forma predeterminada, estos “archivos” virtuales residen en la memoria volátil en la pestaña del navegador y desaparecen cuando se cierra la página. También proporciona una forma para que el sistema de archivos almacene cosas en el almacenamiento local persistente del navegador, pero Pyodide no lo usa. Al emular el sistema de archivos y otras características de un entorno informático estándar, WASM hace posible mover los proyectos existentes al navegador web con sorprendentemente pocos cambios.

3.2.- TECNOLOGÍAS DEL LADO DEL SERVIDOR

Las tecnologías del lado del servidor son las encargadas de dar respuesta a las solicitudes realizadas por el usuario de manera transparente (sin que se muestre de forma visual). Inicialmente los servidores web permitían visualizar exclusivamente información estática, por lo que cuando surgió la primera actividad publicitaria hubo una limitación importante. Posteriormente surgió una solución para que fueran las páginas más flexibles utilizando lenguajes como Perl o C, aunque rápidamente se descartaron por lo inconvenientes que presentaban en cuestión de seguridad y la carga del servidor. Estas tecnologías del lado del

servidor han ido evolucionando, y en la actualidad se disponen de lenguajes de programación como PHP, Python, entre otros, que cumplen correctamente con los requisitos que necesitan los desarrolladores [53].

3.2.1.- PHP

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Una de las ventajas que tiene este lenguaje de programación es que es de código abierto, por lo que permite a los desarrolladores crear funciones nuevas y corregir posibles vulnerabilidades. PHP es definido como un lenguaje del lado del servidor ya que se aplica en el servidor web responsable de ejecutar la aplicación, por lo que permite cargar los elementos de una página web antes de mostrarlos al usuario que accede al sitio web [54].

El lenguaje PHP se considera uno de los más fáciles de aprender en comparación a otros lenguajes porque tiene una comunidad muy amplia y brinda un apoyo importante a la hora de resolver dudas, crear tutoriales, entre otros muchos materiales de aprendizaje. Como es un lenguaje de código abierto, los programadores no necesitan gastar dinero y no pueden tener ningún problema legal con los fundadores del lenguaje a la hora de desarrollar aplicaciones generales para la web. Además cabe destacar que tiene un buen nivel de compatibilidad con las bases de datos, ya que son la parte fundamental de la estructura [55].

3.2.2.- Python

Python es un lenguaje de programación interpretado, orientado a objetos de alto nivel y con semántica dinámica. Su sintaxis hace énfasis en la legibilidad del código, lo que facilita su depuración y, por tanto, favorece la productividad. Ofrece la potencia y la flexibilidad de los lenguajes compilados con una curva de aprendizaje suave. Aunque Python fue creado como lenguaje de programación de uso general, cuenta con una serie de librerías y entornos de desarrollo para cada una de las fases del proceso de Data Science. Esto, sumado a su potencia, su carácter open source y su facilidad de aprendizaje le ha llevado a tomar la delantera a otros lenguajes propios de la analítica de datos por medio de Machine Learning [56].

Tiene estructuras de datos de alto nivel eficientes y se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa, y en menor medida, la programación funcional. El intérprete de Python y la extensa librería estándar se encuentran disponibles libremente en código fuente y de forma binaria para la mayoría de las plataformas, siendo ésta una tecnología “open source”.

Python fue creado por Guido van Rossum a finales de los años 80 en el Centro de Matemáticas y la Informática en los Países Bajos. Este lenguaje actuó como sucesor del lenguaje de programación ABC, siendo este lenguaje capaz de interactuar con el sistema operativo Amoeba y manejar excepciones [57].

Una característica muy importante de este lenguaje es la resolución dinámica de nombres, es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa. Además usa un tipado dinámico y conteo de referencias para la gestión de memoria. Cabe destacar que el intérprete de Python tiene un modo interactivo en el cual se escriben las instrucciones en una especie de intérprete de comandos, donde los resultados de la evaluación se pueden ver de forma inmediata cuando se introducen las expresiones, lo que permite probar porciones de código en el modo interactivo antes de integrarlo como parte de un programa.

3.2.3.- Base de datos MySQL

MySQL [58] es un sistema de gestión de bases de datos relacionales de código abierto con un modelo cliente-servidor. Se trata del sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

Cabe destacar que cuenta con una doble licencia. Por una parte es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. Las versiones Enterprise, diseñadas para aquellas empresas que quieran incorporarlo en productos privativos, incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial.

Las características que aporta una base de datos MySQL son [59]:

- **Arquitectura Cliente y Servidor:** El funcionamiento de MySQL se basa en un modelo cliente y servidor. Es decir, para un mejor rendimiento los clientes y servidores se comunican entre sí de forma diferenciada. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, entre otros.
- **Vistas:** Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas. Esto se trata de un recurso indispensable para bases de datos de gran tamaño.

- **Compatibilidad con SQL:** MySQL ofrece plena compatibilidad con SQL, por lo que si se ha trabajado en otro motor de bases de datos, no habrá problema en migrar a MySQL.
- **Desencadenantes:** MySQL permite automatizar determinadas tareas dentro de la base de datos.
- **Procedimientos almacenados:** MySQL procesa las tablas a través de procedimientos almacenados y no directamente, por lo que mejora la eficacia de la implementación.
- **Transacciones:** Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. En el caso que haya un error imprevisto, cierre inesperado o fallo de energía, el sistema opta por preservar la integridad de la base de datos resguardando la información.

