



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS
DE TELECOMUNICACIÓN**

ÁREA DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

Aplicación de flashcards para el estudio de conceptos matemáticos

**DOCUMENTO N°1
MEMORIA**

**D. JUNQUERA FERNÁNDEZ, Paula
TUTOR: D. Entrialgo Castaño, Joaquín
TUTOR: D. Pérez González, Set**

FECHA: Julio, 2020

Índice

1. Introducción	4
2. Producto.....	6
3. Partes de la documentación del proyecto	7
3.1 MEMORIA	7
3.2 ESTUDIO VIABILIDAD DEL SISTEMA	7
3.3 ANÁLISIS DEL SISTEMA DE LA INFORMACIÓN	8
3.4 DISEÑO DEL SISTEMA DE LA INFORMACIÓN	8
3.5 MANUAL DE USUARIO	8
3.6 MANUAL DE INSTALACIÓN	9
3.7 DESARROLLO DEL PLAN DE PRUEBAS.....	9
3.8 PLANIFICACIÓN Y PRESUPUESTO	9
4. Tecnologías y herramientas utilizadas durante el proyecto	10
4.1 ENTORNOS DE DESARROLLO	10
4.1.1 Visual Studio Code	10
4.1.2 Herramientas para desarrolladores de Google Chrome	11
4.1.3 Git.....	11
4.2 LENGUAJES EMPLEADOS.....	12
4.2.1 Python	12
4.2.2 SQL	12
4.2.3 LaTeX	13
4.3 OTROS.....	13
4.3.1 VirtualBox.....	13
4.3.2 Microsoft Office 365	14
4.3.3 Draw.io.....	14
4.3.4 Windows 10	15
4.3.5 Ubuntu.....	15
4.3.6 Ganttter.....	16

4.3.7	Let's Encrypt.....	16
4.3.8	MathJax.....	17
4.3.9	Chart.js.....	17
4.3.10	SM2.....	17
5.	<i>Problemática encontrada</i>	19
5.1	ESTILOS DE LAS VISTAS	19
5.2	LDAP	19
5.3	MATHJAX	20
5.4	COVID-19	20
6.	<i>Posibles mejoras y ampliaciones</i>	21
6.1	CREACIÓN DE GRUPOS	21
6.2	PANTALLA DE RESULTADOS	21
6.3	FLASHCARDS CON IMÁGENES	22
7.	<i>Conclusión</i>	23
8.	<i>Bibliografía</i>	24

1.Introducción

La finalidad de esta memoria es describir el proyecto de realización de una aplicación de flashcards para el estudio de conceptos académicos; en concreto, está enfocada en los conceptos matemáticos.

El proyecto está integrado dentro del proyecto de innovación docente de la Universidad de Oviedo PINN-19-A-088 Flashcards como herramienta de micro aprendizaje para el estudio de conceptos matemáticos.

Esta aplicación tendrá el objetivo de facilitar el estudio de la materia impartida en ciertas asignaturas de la Universidad de Oviedo a sus alumnos.

Las tarjetas sintetizarán los conceptos más importantes a estudiar, como fórmulas, definiciones, normativa, ejemplos o datos, de manera que un alumno use la aplicación para memorizar o realizar repasos rápidos del temario.

La funcionalidad principal de la herramienta es ir mostrando al alumno bloques de cuestiones sobre la materia, para que, posteriormente, éstos realicen una autoevaluación de su conocimiento acerca de la respuesta. El alumno será el encargado de juzgar sus logros respecto a las diferentes preguntas que se le plantean. La autoevaluación es una excelente herramienta de cara a descubrir sus propias debilidades y fortalezas. Asimismo, este procedimiento estimulará la memoria activa de los alumnos relacionando la pregunta que se les hace con la respuesta correcta proporcionada por la tarjeta.

Adicionalmente, para optimizar el aprendizaje de los conceptos más complicados de la asignatura, se mostrarán con menor frecuencia las cuestiones mejor valoradas por los alumnos, creando sesiones personalizadas para cada uno de ellos. Este es un método denominado “repetición espaciada”, que ha demostrado empíricamente que es una de las técnicas más eficaces para incrementar la velocidad de aprendizaje.

Por otra parte, la interacción de los alumnos con los conjuntos de tarjetas permitirá al profesor visualizar de manera fácil que partes del temario poseen mayor dificultad desde el punto de vista de su alumnado. Estas estadísticas disponibles para el profesor, junto con la posibilidad de utilizar fórmulas matemáticas en las tarjetas, tanto en la respuesta como en la pregunta, son las características que hacen a esta aplicación distinta de otras existentes en el mercado. Además, la aplicación se ha diseñado para facilitar al profesor la creación de mazos, permitiendo importarlos a través de ficheros CSV (valores separados por comas), y la importación de alumnos usando el mismo formato.

Otra característica importante de la aplicación es que se ha desarrollado de tal manera que las credenciales de acceso son validadas por el servidor de identidad (LDAP) de la Universidad de Oviedo, por lo tanto, no resulta necesario que los usuarios (tanto profesores como alumnos) tengan que recordar una contraseña más. Este hecho facilitará el acceso a la plataforma a los miembros de la comunidad universitaria.

Por último, se ha diseñado la aplicación para que sea posible traducirla a distintos idiomas, se ha realizado la traducción de los recursos al castellano y al inglés para que pueda ser utilizada en los itinerarios bilingües de la Universidad de Oviedo.

2.Producto

El producto de este proyecto es una aplicación web de tarjetas de estudio compuesta por una aplicación web cliente y dos servidores.

La aplicación está desarrollada en el framework de desarrollo web gratuito y de código abierto Django, escrito en Python. Este framework fomenta el diseño limpio, claro y pragmático y sigue las directrices del patrón “Modelo-Vista-Controlador”.

El producto también está formado por un servidor de aplicaciones llamado Gunicorn, encargado de interactuar con nuestra aplicación. Se encarga de traducir las solicitudes de los clientes de HTTP que llegan desde el navegador web a llamadas Python que puedan ser procesadas por nuestra aplicación. Esta interfaz de aplicación será la encargada de facilitarnos el envío del contenido dinámico, como, por ejemplo, la información de la base de datos, al servidor web.

Además, se configura un servidor web llamado Nginx, responsable del envío del contenido estático a nuestra aplicación. Gunicorn conectará el servidor Nginx con Django para generar el contenido dinámico.

La aplicación dispone de una base de datos de PostgreSQL, un sistema de gestión de base de datos relacional orientado a objetos, multiplataforma, fácil de usar y con la capacidad de manejar un gran volumen de datos.

En esta aplicación los alumnos que tengan acceso a ella podrán visualizar e interactuar con los bloques de flashcards creados por los profesores. Los profesores que tengan permisos de acceso podrán crear, modificar y eliminar los bloques de tarjetas, importar flashcards desde un archivo con formato CSV, dar acceso a conjuntos de alumnos (importando otro CSV con sus nombres de usuario) y visualizar y exportar los resultados de la interacción de los alumnos con los conjuntos de flashcards.

3.Partes de la documentación del proyecto

En esta sección se enumerarán los diferentes documentos y apartados que forman este proyecto.

3.1 MEMORIA

1. Introducción.
2. Producto.
3. Partes de la documentación del proyecto.
4. Tecnologías y herramientas utilizadas durante el proyecto.
5. Problemática encontrada.
6. Posibles mejoras y ampliaciones.
7. Conclusión.

3.2 ESTUDIO DE VIABILIDAD DEL SISTEMA

1. Establecimiento del alcance del sistema.
2. Estudio de la situación actual.
3. Definiciones de requisito del sistema.

4. Alternativas de solución.
5. Solución seleccionada.

3.3 ANÁLISIS DEL SISTEMA DE LA INFORMACIÓN

1. Alcance del sistema.
2. Requisitos.
3. Especificación de subsistemas.
4. Análisis de subsistemas.
5. Modelo de datos.
6. Prototipos de la interfaz.
7. Análisis de consistencia.
8. Especificaciones del plan de prueba.

3.4 DISEÑO DEL SISTEMA DE LA INFORMACIÓN

1. Diseño de la arquitectura del sistema.
2. Última versión de los prototipos de la interfaz.
3. Segunda versión del diseño del plan de pruebas.

3.5 MANUAL DE USUARIO

1. Introducción al sistema.

2. Inicio de sesión.
3. Perfil de profesor.
4. Perfil de alumno.
5. Perfil de administrador.

3.6 MANUAL DE INSTALACIÓN

1. Introducción.
2. Requisitos.
3. Preparación del entorno.
4. Despliegue aplicación producción.

3.7 DESARROLLO DEL PLAN DE PRUEBAS

1. Introducción.
2. Versión final del diseño del plan de pruebas.

3.8 PLANIFICACIÓN Y PRESUPUESTO

1. Planificación.
2. Presupuesto.

4. Tecnologías y herramientas utilizadas durante el proyecto

4.1 ENTORNOS DE DESARROLLO

4.1.1 Visual Studio Code

VS Code es un editor de código fuente sofisticado, rápido y de código abierto que admite numerosas funcionalidades prácticas en el desarrollo de código. Entre ellas se destaca que es multiplataforma, se actualiza constantemente y posee la capacidad para predecir la instrucción que el desarrollador pretende escribir. Además, permite trabajar con diversos lenguajes de programación, la edición de código no está limitada a C# y VB, se puede usar lenguajes como JavaScript, C, C++, CSS, HTML, XML y JSON, entre otros.



Figura 4.1.- Visual Studio Code.

4.1.2 Herramientas para desarrolladores de Google Chrome

Las herramientas que se incluyen en la mayoría de los navegadores modernos son una gran ayuda a la hora de analizar rápidamente el contenido y los recursos de páginas web.

Permiten visualizar el código HTML de la página y las reglas CSS que se han aplicado, posibilitando la modificación de los elementos y el estilo libremente. Además, se puede depurar el código, grabar la página mientras se utiliza de cara a realizar pruebas y medir tiempos de carga para optimizar su rendimiento.



Figura 4.2.- Herramientas desarrolladores Google Chrome.

4.1.3 Git

Este software de control de versiones permite gestionar los diversos cambios que se realizan sobre los elementos del producto, facilitando la administración de sus distintas versiones. Proporciona herramientas para desarrollar software en equipo de manera inteligente y rápida. Git se caracteriza por su rapidez, la gestión distribuida, la gestión eficiente de proyectos de gran volumen y el almacenamiento periódico en paquetes.



Figura 4.3.- Git.

4.2 LENGUAJES EMPLEADOS

4.2.1 Python

Python es un lenguaje de programación multiplataforma y multiparadigma orientado a objetos y de código abierto. Se caracteriza por la simplicidad, versatilidad y rapidez de desarrollo. Está preparado para llevar a cabo cualquier tipo de programa y, además, es un lenguaje interpretado, por lo tanto, no necesita que se compile el código fuente para poder ejecutarlo. Python es interactivo y posee gran cantidad de librerías con tipos de datos y funciones incorporadas en el propio lenguaje y una sintaxis clara.



Figura 4.4.- Python.

4.2.2 SQL

Es un lenguaje declarativo estándar e interactivo de acceso a bases de datos relacionales. Permite realizar consultas para recuperar información de las bases de datos de manera sencilla. Estas consultas toman la forma de un lenguaje de comandos posibilitando seleccionar, actualizar, insertar y encontrar donde se ubican los datos. SQL se puede integrar a lenguajes de programación y es posible usarlo en combinación con cualquier base de datos específica.



Figura 4.5.- SQL.

4.2.3 LaTeX

Sistema de composición de textos (científicos principalmente) orientado a documentos escritos de alta calidad. Es utilizado por un gran abanico de programas, servicios web, artículos y libros científicos.



Figura 4.6.- LaTeX.

4.3 OTROS

4.3.1 VirtualBox

Este software de virtualización para arquitecturas x86 permite instalar sistemas invitados, es decir, sistemas operativos adicionales dentro de otro sistema operativo anfitrión, cada uno con su entorno independiente. Actualmente, es desarrollado por Oracle. Es multiplataforma y en modo anfitrión algunos de los sistemas operativos soportados son Mac

OS X, Windows y GNU/Linux y en su interior se permite virtualizar sistemas operativos como GNU/Linux, Windows, o FreeBSD.



Figura 4.7.- VirtualBox

4.3.2 Microsoft Office 365

Es una herramienta que permite crear acceder y compartir documentos de Excel, Word OneNote y PowerPoint en tiempo real, pudiendo acceder desde cualquier dispositivo que tenga acceso a Internet y OneDrive.



Figura 4.8.- Office 365.

4.3.3 Draw.io

Es una herramienta de creación y edición de diagramas libre y multiplataforma. Es una aplicación web que funciona en un gran abanico de navegadores permitiendo la creación de diagramas. Esta herramienta esta disponible para trabajar tanto en línea como sin conexión.



Figura 4.9.- Draw.io.

4.3.4 Windows 10

Es un sistema operativo vigente desarrollado por Microsoft perteneciente a la familia de sistemas operativos Windows NT para el uso en computadoras personales, acompañado de un conjunto de programas y un sistema de organización de archivos propio.



Figura 4.10.- Windows 10.

4.3.5 Ubuntu

Basada en GNU/Linux, es una distribución Linux que ofrece un sistema operativo de software libre y código abierto enfocado a ordenadores de escritorio (también proporciona soporte a servidores). Se caracteriza por su facilidad de instalación, lanzamientos regulares y libertad y facilidad de uso.



Figura 4.11.- Ubuntu.

4.3.6 Gantter

Software para gestión de proyectos gratuito basado en diagramas de Gantt. Permite crear y editar planes de proyectos integrados con Google y generar diagramas de Gantt con sus tareas, fases e hitos.



Figura 4.12.- Gantter.

4.3.7 Let's Encrypt

Es una CA (Autoridad de Certificación) que permite de manera fácil obtener e instalar certificados TSL/SSL gratuitos para permitir HTTPS en servidores web. Esto lo consigue con Certbot, un cliente de software que automatiza el proceso.



Figura 4.13.- Let's Encrypt.

4.3.8 MathJax

Es un motor de visualización en JavaScript para la notación LaTeX, AsciiMath y MathML, permite visualizar fórmulas matemáticas de cualquier dificultad. Funciona en la totalidad de navegadores web modernos y es de código abierto.



Figura 4.14.- MathJax.

4.3.9 Chart.js

Librería de JavaScript simple, flexible y muy completa que, usando el canvas de HTML5, posibilita la visualización de gráficos en páginas web. Sólo es necesario incluir el plugin en la cabecera de la página, ya que no requiere jQuery.



Figura 4.15.- Chart.js.

4.3.10 SM2

Es un algoritmo de repetición espaciada simple y efectivo. Es el más popular, usado por aplicaciones como Duolingo, Memrise o Anki.

Este algoritmo es el usado para determinar qué tarjetas le corresponde ver a un determinado alumno cada día. Para realizar este cálculo se deben de tener en cuenta tres campos:

- Next due date: Próxima vez que se va a volver a mostrar la tarjeta a un alumno.
- Consecutive correct answers: Número de veces que el alumno ha respondido correctamente una determinada tarjeta.
- Easiness: Facilidad de la tarjeta. Está representada por un número mayor o igual a 1.3. El valor por defecto de este campo es 2.5.

Cuando el alumno selecciona un mazo desde su perfil, las tarjetas que tienen su campo next due date con una fecha previa a la actual, se le mostrarán de manera aleatoria. El estudiante, después de visualizar cada tarjeta, deberá asignarle una calificación de rendimiento. Esta puntuación podrá variar entre 0 (puntuación más baja) y 5 (puntuación más alta), siendo 3 la calificación necesaria para considerar la tarjeta como correcta.

Por último, se actualizarán los tres campos de la tarjeta nombrados anteriormente:

- $Easiness + = -0.8 + 0.28 * Calificación + 0.02 * Calificación^2$
- *Consecutive correct answers* será igual a 0 si la respuesta es incorrecta. Si, por el contrario, es correcta, *Consecutive correct answers* + 1.
- *Next due date* se incrementará en 1 día más si la respuesta es incorrecta. Si ésta es correcta: $Next\ due\ date + (6 * Easiness ^ (Consecutive\ correct\ answers - 1))$.

5. Problemática encontrada

5.1 ESTILOS DE LAS VISTAS

La primera problemática encontrada durante el desarrollo de la aplicación web de flashcards fue el uso de diferentes estilos en las vistas que forman la página web.

Esto es debido a que las pantallas de interacción con las tarjetas desde el perfil de alumno necesitaban estilos completamente diferentes a los usados en el resto de las vistas de la aplicación. Esta situación ocasionó que se mezclasen erróneamente durante un periodo del desarrollo los dos tipos de estilos.

Para solucionarlo las herramientas de Google Chrome fueron de gran ayuda al permitir modificar en tiempo real los estilos de una manera sencilla y visual.

5.2 LDAP

Debido a la poca información sobre LDAP que se puede encontrar en internet, fue costoso conseguir que los usuarios sólo pudiesen iniciar sesión en la aplicación si cumplían estos dos requisitos:

- Encontrarse en la base de datos PostgreSQL del proyecto
- Pertenecer a la Universidad de Oviedo y acceder a la plataforma con sus credenciales.

El problema era que se permitía el acceso a un usuario si introducía sus credenciales de la Universidad de Oviedo o se encontraba en la base de datos de la aplicación, sin comprobarse que se cumpliesen las dos condiciones simultáneamente.

Finalmente se encontró solución a la problemática y sólo es posible acceder a la página si se cumplen ambas.

5.3 MATHJAX

Otro bug encontrado durante el desarrollo de la aplicación fue la importación incorrecta de tarjetas en lenguaje LaTeX. Esto era debido a que no se debía usar los delimitadores que venían por defecto, ya que podían confundirse con otros símbolos que poseía nuestro código y, por lo tanto, no separaba correctamente las sentencias y fórmulas importadas o exportadas. Finalmente se solucionó modificando el delimitador de fórmulas. Para escribir en LaTeX en la aplicación se usará “\$”.

5.4 COVID-19

Debido a la situación en la que nos encontramos, no se han podido realizar todas las pruebas que se tenían programadas con alumnos a gran escala. La aplicación se ha usado con menor frecuencia de la esperada, pero se espera que en el siguiente curso se retome la actividad para que ésta sea utilizada con normalidad y durante un tiempo prolongado para obtener resultados de rendimiento y estadísticas más detalladas.

6. Posibles mejoras y ampliaciones

6.1 CREACIÓN DE GRUPOS

En la versión actual de la aplicación web, un bloque de tarjetas será o no visible para la totalidad de estudiantes que accedan a la plataforma, pudiendo todos ellos interactuar con cualquiera de los mazos que visualicen. Actualmente, el profesor de cada asignatura podría crear sus bloques de tarjetas para que sus alumnos interactuasen con ellas, pero en ningún caso se le podría denegar a un usuario el acceso a los mazos de otras asignaturas, pudiendo éste influir en los resultados de una asignatura que no cursa.

Se propone como ampliación del presente proyecto la división de los alumnos en grupos. De esta manera, los bloques de tarjetas creados por un determinado profesor sólo serán visibles para los estudiantes de las asignaturas que éste imparte.

6.2 PANTALLA DE RESULTADOS

La pantalla de resultados de la versión actual permite al profesor:

- Visualizar la dificultad media por mazo que perciben sus alumnos.
- Visualizar de forma gráfica en un diagrama de araña la “memorización” y “dificultad” de las tarjetas de un mazo, pudiendo divisar con facilidad qué contenido del temario presenta mayor complejidad para sus alumnos.
- Visualizar los campos “Dificultad” y “Respuestas correctas consecutivas” en una tabla para cada pareja alumno-tarjeta.

- Exportar un CSV con más datos acerca de tiempos, valoraciones y sesiones de los alumnos con cada bloque de tarjetas.

Se propone para versiones futuras la ampliación y mejora de esta pantalla, pudiendo mostrar un gráfico de araña por alumno en el que se visualice su agilidad, capacidad de memorización y conocimiento del temario, entre otros. En la versión actual, estas cualidades pueden ser calculadas a partir de los datos exportados en el archivo de formato CSV.

6.3 FLASHCARDS CON IMÁGENES

Otra posible ampliación sería dar la posibilidad de incorporar imágenes o videos en las tarjetas de estudio.

En esta primera versión de la aplicación no se planteó la introducción de este tipo de formatos. Esto es debido a que su objetivo principal era permitir la correcta visualización de expresiones matemáticas en lenguaje LaTeX para ser utilizada por alumnos de las asignaturas de Álgebra y Cálculo de la Universidad de Oviedo.

7. Conclusión

El proyecto ha cumplido sus objetivos, ya que se ha creado y puesto a disposición de los profesores y alumnos una aplicación para flashcards que cumple las especificaciones deseadas.

El presente proyecto está integrado dentro del proyecto de innovación docente de la Universidad de Oviedo PINN-19-A-088 Flashcards como herramienta de micro aprendizaje para el estudio de conceptos matemáticos.

En un principio, la aplicación iba a ser utilizada frecuentemente por los alumnos de las asignaturas de Álgebra y Cálculo de esta universidad durante el segundo semestre del curso 2019-2020, pero, finalmente, debido a la situación en la que nos encontramos causada por la pandemia por COVID-19, se usó con menor frecuencia de la esperada. A pesar de esto, el resultado de la interacción de los alumnos y profesores con la aplicación fue satisfactoria y se espera que el curso que viene se pueda retomar la actividad con otros alumnos para comprobar su rendimiento y funcionalidad a gran escala y durante un tiempo prolongado y, así, poder observar posibles mejoras en las calificaciones obtenidas por los alumnos de las asignaturas.

Por otro lado, me ha servido para ampliar mis conocimientos acerca de Python, JavaScript, HTML y Visual Studio Code, entre otros, y a conocer a fondo el framework Django y desarrollo de aplicaciones web.

A nivel personal, a pesar de haber tenido que compaginar el desarrollo de la aplicación con un trabajo a jornada completa, la experiencia ha sido claramente satisfactoria para mí, ya que este proyecto satisface una necesidad real de la universidad y espero que gracias a él se puedan obtener resultados realmente positivos.

8. Bibliografía

Las diferentes fuentes de información empleadas para este proyecto son:

Django

<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/>

Mathjax

<https://www.mathjax.org/#gettingstarted>

VS Code Remote Development

<https://code.visualstudio.com/docs/remote/>

Diagrama con Chart.js

<https://www.chartjs.org/docs/latest/>

Mensajes de alerta

<https://sweetalert.js.org/guides/>

Let's Encrypt

<https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>

Y, por último, documentación adquirida en las asignaturas cursadas durante la titulación.