

# EDUKIDS

## SOFTWARE FUNDAMENTALS II

### 1. Equipo

Integrantes:

1. Andreea Ganea.
2. José Rico.
3. Matías Bermejo.
4. Rodrigo Librán.

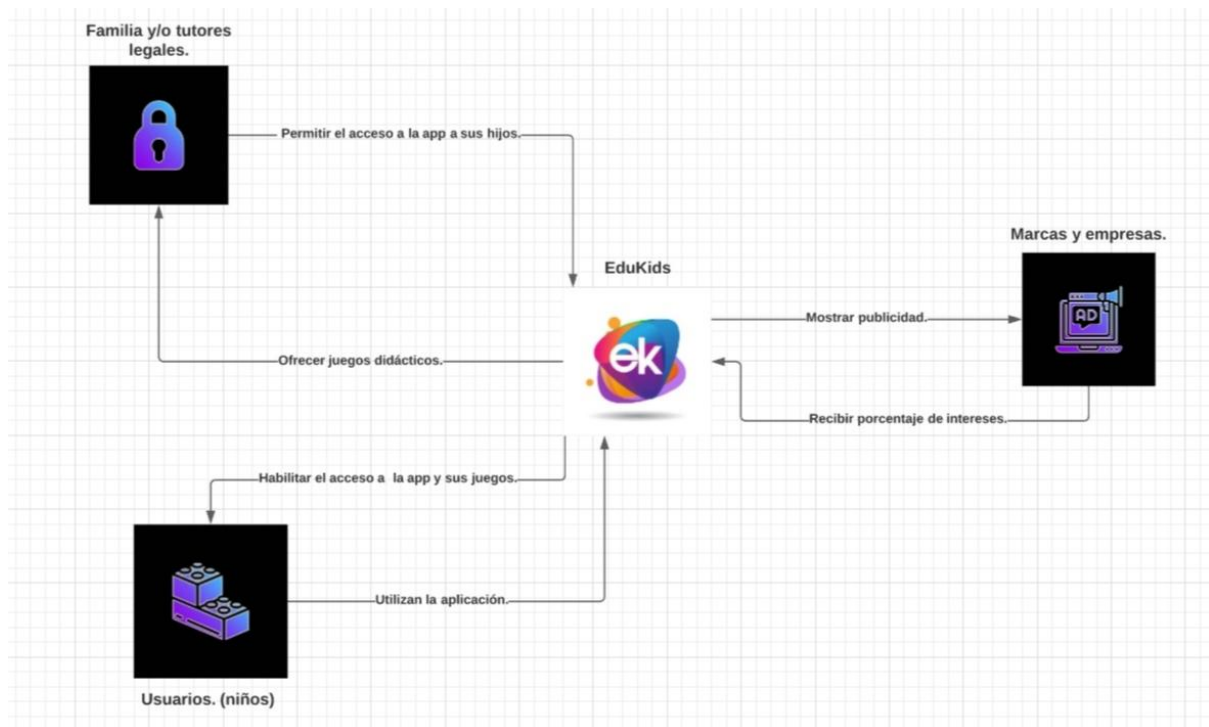
### 2. Registro de control de cambios al documento

Fecha	Versión	Descripción del Cambio	Autor
04/11/2021	0.1	Creación del proyecto, realizada la presentación inicial con las principales características de la app.	Todo el equipo.
10/11/2021	0.2	Desarrollo de la documentación necesaria para el proyecto. (Step by Step, Output, Diagrama de contexto.)	Todo el equipo.
11/11/2021	0.3	Mejorar la documentación y especificarla más. (Editar el diagrama, renovar el Step by Step y refinar el Output.)	Todo el equipo.
11/11/2021 - 16/11/2021	0.4	Diseñar y construir el código de la aplicación.	Todo el equipo.
16/11/2021	0.5	Presentar el resultado final: la aplicación.	Todo el equipo.
29/11/2021	1.0	Actualización de las funcionalidades del programa usando archivos y rehacer step-by-step	Todo el equipo

### 3. Introducción

Aplicación didáctica para ampliar y fomentar la educación infantil apoyándose en los avances tecnológicos.

### 4. Diagrama de Contexto



### 5. Descripción del proyecto

Edukids es una aplicación que funcionará como herramienta para gamificar y mejorar las capacidades y habilidades de los niños, introduciéndose a la educación tecnológica.

Se compondrá de juegos didácticos y entretenidos para los usuarios (los niños) bajo la supervisión parental.

## 6. Guía paso a paso

La aplicación Edukids consta principalmente de dos archivos:

- `main.py`: Aquí se ejecutan las llamadas a las funciones y la ejecución de los bucles.
- `functions.py`: Aquí aparecen todas las funciones que se ejecutan.

Definir las funciones :

. Definir la función `choose_begining()` :

- Dar las opciones que lleven al juego o a terminar el programa y elegir con un

`input`:

. Juego : continuar al siguiente menú definido en la función `choose_course()`

. Salir : terminar el programa con un mensaje

. Definir la función `choose_course()` :

- Mostrar las opciones :

- Matemáticas
- Lengua
- Ciencia
- Casual
- Volver al menú

Para llevar a cabo los juegos específicos de cada una o elegir volver al menú ( definido en la función `choose_begining()`) inicial con un `input`, al escoger una opción diferente a “Volver al menú” se desplegará el siguiente menú de juegos definido en la función `choose_game`.

. Definir la función `choose_game()` de la asignatura elegida :

- El usuario debe elegir con un `input` qué juego ejecutar y resolver o elegir la opción “volver al menú de categorías” (definido en la función `choose_course()`)

Juegos :

. Matemáticas :

- El busca números : genera un número aleatorio del 1 al 100 y al introducir un número mostrará si el número a adivinar es menor, mayor o si has acertado.

- Operaciones : se genera una operación matemática aleatoria y se le deberá dar el resultado.

- Sucesiones numéricas : mostrar un mensaje que diga : “Juegos en proceso” y llevar al usuario al menú anterior

- Problemas matemáticos : mostrar un mensaje que diga : “Juegos en proceso” y llevar al usuario al menú anterior

. Lengua :

- Sinónimos : se da una palabra y tres opciones de las cuales una de ella es correcta , el usuario deberá averiguar cuál de ellas es.

- Tilde : se da una palabra con o sin tilde y el usuario deberá averiguar si está escrita correctamente y donde iría la tilde

- Está bien escrito ? : mostrar un mensaje que diga : "Juegos en proceso" y llevar al usuario al menú anterior

- Ahorcado : mostrar un mensaje que diga : "Juegos en proceso" y llevar al usuario al menú anterior

- . Ciencias : mostrar un mensaje que diga : "Juegos en proceso" y llevar al usuario al menú anterior

- . Casual : mostrar un mensaje que diga : "Juegos en proceso" y llevar al usuario al menú anterior

- . Volver al menú de categorías : devuelve al usuario al menú que le permite elegir la asignatura o ir al menú inicial y decidir si volver a jugar o salir completamente de la aplicación.

- . Definir la función de cada uno de los juegos que se ejecutarán y aparecen en el menú anterior.

- Al elegir el juego se ejecutará la función de determinado juego y se abrirá un input donde el usuario insertará la respuesta.

- Al acabar el juego, el usuario tiene tres opciones a elegir con un input :

- . Cambiar de juego : volver al menú de juegos de la asignatura elegida definido en la función choose\_game().

- . Cambiar de categoría : volver al menú de asignaturas definido en la función choose\_course()

- . Salir de la aplicación : terminar el programa con un mensaje que se despida diciendo: "¡Vuelve pronto!"

## **Nuevas adiciones:**

### **- Sistema de puntos para desbloquear juegos**

Al ver los juegos algunos estarán bloqueados y al entrar en ellos te pedirán una cantidad de puntos para poder desbloquearlos.

función comprar():

- te ofrece dos opciones:

- comprar por (puntos x)

- cambia el valor de 0 a 1 en el .txt

- ejecuta el juego

- volver

- vuelve a la pantalla anterior

Dentro de cada categoria:

- if la categoria2 es igual a (el juego)

- if el juego en el .txt es un 1
  - ejecutar el juego
- elif el juego en el .txt es igual a 0
  - ejecutar la funcion comprar()

## - Estadísticas

Crear la función estadísticas()

- Dar la opción al iniciar sesión que te muestre las estadísticas estadísticas()
- importamos la librería "import matplotlib.pyplot as plt"
- Ponemos fig, ax = plt.subplots() para crear la gráfica de estilo circular
- Lee los datos del .txt con los puntos del usuario en cada categoría de lengua, matemáticas... y los pone en los valores donde está la x en: ax.pie([x, x, x, x, x])
- Con esos datos crea un gráfico personalizado mostrando en qué categoría es mejor y en cual podría mejorar
- Para mostrar el gráfico utilizaremos plt.show()

## - Iniciar sesión

AÑADIDO > Usuario: creamos una nueva función reg\_usuario() dentro de ella habrá un diccionario vacío dic = {"key" : "value"}. Además de desplegarse tres opciones:

- "Iniciar sesión" : creamos la función inicio\_sesion(). Dentro se desplegarán tres opciones, que serán funciones llamadas, que podrá escoger a través de un input que le preguntará si quiere la opción 1, 2 o 3.
  - (1) "Iniciar sesión." : crear una función inicio\_sesion() que contenga:
    - Una variable llamada usuario = con un input que pregunte al usuario su user name. Abrir un bucle while con un if dentro donde mientras usuario está dentro de dic aparezca el siguiente input, else que aparezca un mensaje (un print) que informe de que el user name introducido no existe que vuelva a intentarlo, al estar dentro de un bucle while podrá intentar de nuevo escribirlo correctamente.
    - Una variable llamada contrasena = con un input que pregunte al usuario su password. Abrir un bucle while con un if dentro donde mientras contrasena está dentro de dic se llame a la función volver\_atras()\* que le hará volver al main\_menu(). \*\*En el main\_menu() aparecerán el user name y los puntos correspondientes del usuario. Else, mostrará un mensaje indicando error porque la password no es correcta\*\*. Gracias a que está dentro de un bucle while podrá reintentarlo.

- (2) "Registro." : crear una función register() que contenga:
  - Una variable llamada usuario\_reg = con un input que le pregunte al usuario su user name. Dentro habrá un bucle while con un if, mientras usuario\_reg este dentro del dic, aparecerá un mensaje diciéndole al usuario que ese user name ya existe y que elija otro. Else el nuevo user name se guardará en el diccionario usando dic[usuario\_reg].
  - Una variable llamada contrasena\_reg = con un input que le regunte al usuario su password y se guarde en el diccionario usando dic[usuario\_reg] = contrasena\_reg.
- (3) "Volver."\*Crear una función volver\_atras() que contenga:
  - Un input que le diga al usuario que pulse cualquier tecla para volver al menú. Esto le hará volver al main\_menu() al estar el input introducido por un stop =.
- \*\*Juegos: si escogen esta opción se les mostrará el menú creado choose\_game().
- \*\*Salir: si escogen esta opción se mostrará un mensaje de despedida y se cerrará el programa.

LEYENDA: \* el asterisco se refiere a la función volver\_atras() concretada más abajo.

\*\* El asterisco doble se refiere a funciones ya creadas en la iteración anterior y que ya existen en el código.

## - **Rachas**

Importar datetime.

Crear un diccionario vacío dicc{}

Tras iniciar sesión, o de abrir la aplicación en el caso de tener la sesión abierta, saltará un print que te de la enhorabuena por haberte conectado a la app la cantidad de días que lleves conectando seguidos. La consola sabrá que llevas esos días entrando por la librería datetime más la nueva función creada racha():

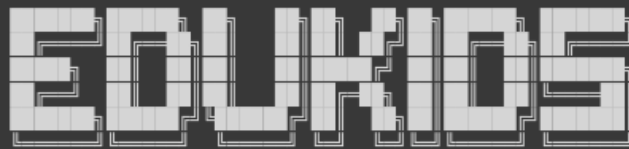
Habrà un bucle for donde mientras el número sea menor o igual que siete, habrá un if por cada día, donde si te has conectado el primero habrá un multiplicador que te sume puntos a tu total de puntos guardados en el diccionario vacío.

Y lo mismo en los 7 siguientes siendo cada día el multiplicador 50 puntos mayor hasta el último que será bastante 200 puntos más.

Este bucle for está dentro de otro bucle for, que será en el que indiquemos que mientras el número de días sea 7+1 el for deberá volver a comenzar, por lo tanto las recompensas que recibes también, es decir, el octavo día tendrá la misma recompensa que el primero y así continuando.

Esto se hará para que las recompensas no se multipliquen infinitamente y para que el 7mo día siga teniendo esa gran recompensa y por lo tanto siga valiendo la pena hacer rachas de más semanas. Como el día 1 tiene su propio valor multiplicador de puntos asignado, cada vez que pierdas la racha, por no haberte conectado, volverá a la recompensa del día 1.

## 7. Ejemplo Output 1



- 1.Juegos
- 2.Iniciar sesión / Registrarse
- 3.Salir

=> Introduce una opción:

Iniciar sesión

- 1.Iniciar sesión
- 2.Registrarse
- 3.Volver

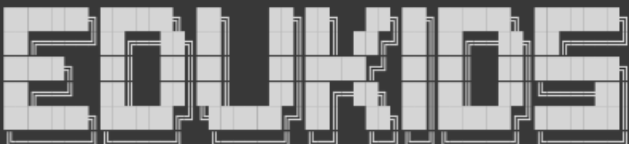
=> Introduce una opción:

Iniciar sesión

--> Introduce tu nombre de usuario:      rodrigo

--> Introduce tu contraseña:      passw

Usuario loggeado con éxito



- 1.Juegos
- 2.Estadísticas
- 3.Salir

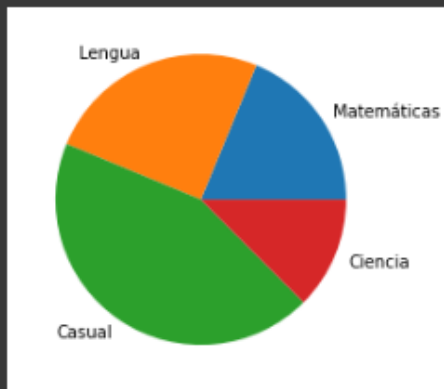
=> Introduce una opción:

- 1.Volver a jugar
- 2.Volver a juegos
- 3.Salir de EDUKIDS

¿Que quieres hacer?:    3

¡¡¡Hasta pronto!!!

## Estadísticas



## 8. Ejemplo Output 2

EDUWIDS

1.Juegos

2.Iniciar sesión / Registrarse

3.Salir

=> Introduce una opción:

Iniciar sesión

1.Iniciar sesión

2.Registrarse

3.Volver

=> Introduce una opción:





## Registrar usuario

--> Introduce tu nombre de usuario:      rodrigo  
--> Introduce tu contraseña:      passw

Usuario registrado con éxito

Presione una tecla para continuar...

# EDUARKIDS

1. Juegos
2. Estadísticas
3. Salir

=> Introduce una opción:

# JUEGOS

Las categorías de juegos son:

1. Matemáticas
2. Lengua
3. Ciencia
4. Casual
5. Volver al menú

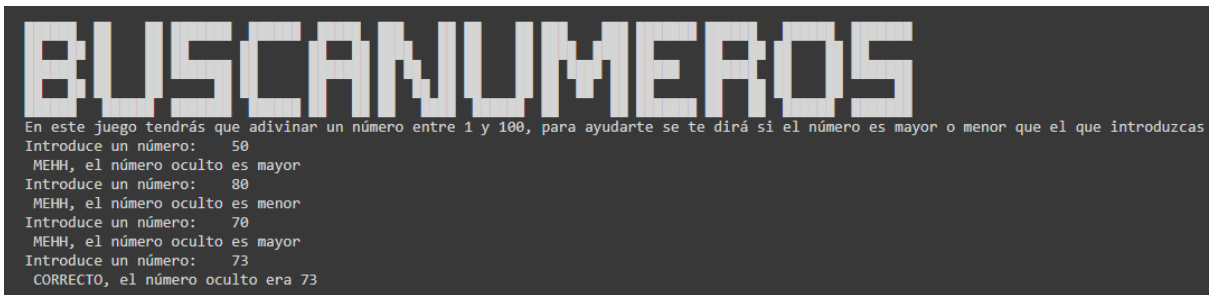
=> Introduce una opción:

# JUEGOS - MATEMATICAS

Los juegos de matemáticas son:

1. El buscanúmeros
2. Sucesiones numéricas
3. Operaciones
4. Problemas matemáticos
5. Volver al menú de categorías

=> Introduce una opción:



## 9. Recursos

1. Código Fuente.
2. [https://lucid.app/lucidchart/966d6f72-fbf5-4b62-9ab6-2f13830bd0b6/edit?viewport\\_loc=-164%2C-28%2C2200%2C1147%2C0\\_0&invitationId=inv\\_4802db3f-39b4-4830-8e6f-620f8c576eac](https://lucid.app/lucidchart/966d6f72-fbf5-4b62-9ab6-2f13830bd0b6/edit?viewport_loc=-164%2C-28%2C2200%2C1147%2C0_0&invitationId=inv_4802db3f-39b4-4830-8e6f-620f8c576eac) Plataforma en la que se realizó el diagrama.
3. Google Colab.
4. Visual Code.