

# Manual de EDUMON

Juan Camilo Erazo

26 de mayo de 2018

## 1. Introducción

En este manual se encuentran todas las especificaciones, procesos y archivos que se necesitan para ensamblar el juego de EDUMON así como también las explicaciones de como registrar, cambiar las bases de datos y jugar.

## 2. Indicaciones previas

Para comenzar, en general ya sea como administrador o como usuario se debe tener: Un ordenador con cualquier sistema operativo, de preferencia Windows 7/8/10 y un navegador, puede ser cualquiera de preferencia usar Google Chrome o Mozilla Firefox.

## 3. Indicaciones para Administradores

### 3.1. Python

El servidor de EDUMON se maneja a través del lenguaje de programación Python y por eso es necesario instalarlo.

- Primero debemos asegurarnos de tener instalado Python, en caso de no tenerlo procedemos a instalarlo desde este enlace.
- Después de que se haya completado la descarga procederemos a instalarlo y asegurarnos de activar pip.

### 3.2. Flask

Para poder modificar el código base del servidor vamos a necesitar Flask, un framework de Python el cual nos permite crear una aplicación web.

- Fask se instala de una manera muy fácil y rápida, si por alguna razón no se puede llevar a cabo la instalación procederemos a reinstalar Python asegurándonos que todos los campos están marcados.

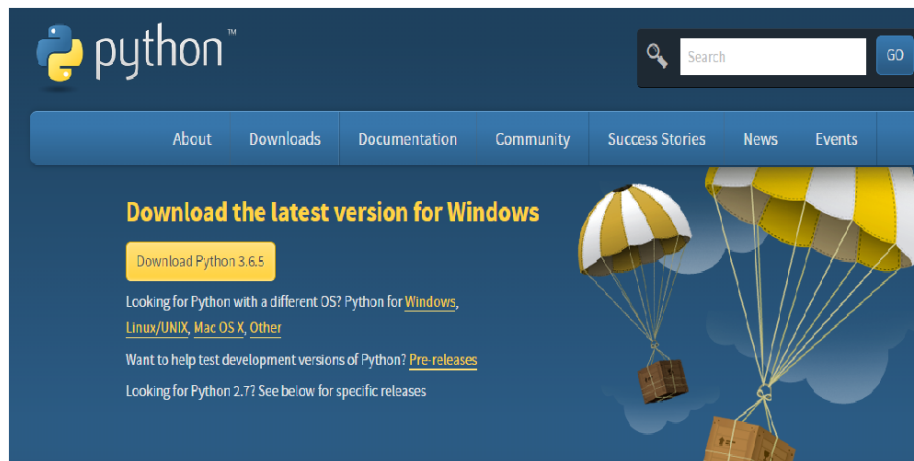


Figura 1: Pagina de instalación Python

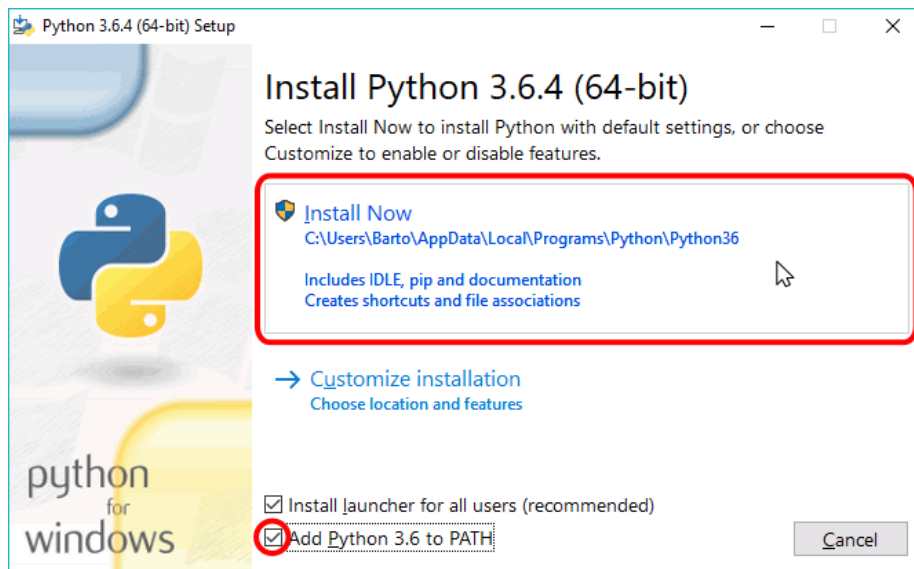
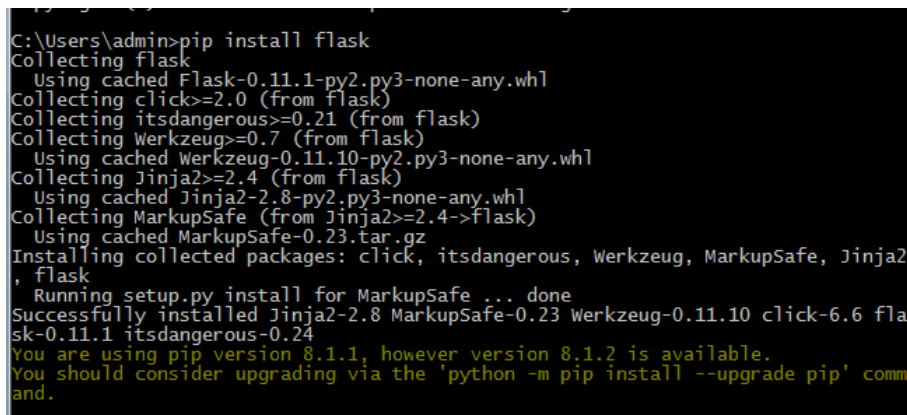


Figura 2: Instalador de Python

- Abrimos la consola de comandos de Windows como administrador.
- Escribimos el comando `-pip install flask` y presionamos la tecla Enter



```
C:\Users\admin>pip install flask
Collecting flask
  Using cached Flask-0.11.1-py2.py3-none-any.whl
Collecting click>=2.0 (from flask)
Collecting itsdangerous>=0.21 (from flask)
Collecting Werkzeug>=0.7 (from flask)
  Using cached Werkzeug-0.11.10-py2.py3-none-any.whl
Collecting Jinja2>=2.4 (from flask)
  Using cached Jinja2-2.8-py2.py3-none-any.whl
Collecting MarkupSafe (from Jinja2>=2.4->flask)
  Using cached MarkupSafe-0.23.tar.gz
Installing collected packages: click, itsdangerous, Werkzeug, MarkupSafe, Jinja2, flask
Running setup.py install for MarkupSafe ... done
Successfully installed Jinja2-2.8 MarkupSafe-0.23 Werkzeug-0.11.10 click-6.6 flask-0.11.1 itsdangerous-0.24
You are using pip version 8.1.1, however version 8.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

Figura 3: Instalación de Flask

## 4. Set-Up del servidor

En la carpeta que contiene el juego podremos encontrar diferentes archivos en diferentes formatos como lo son: html, python, json y txt.

- Los archivos `.txt` son los encargados de almacenar la información de los usuarios registrados así como también se encargan de la información para el ingreso de los administradores a su entorno web de registro.

El admin por defecto debe ser agregado en el archivo de texto manualmente, ya que vendrá vacío a propósito.

- La ruta `/templates` contiene todas las plantillas en formato HTML que utiliza la página web, gracias al motor gráfico *Jinja2* podemos heredar plantillas sobre una plantilla base llamada `base.html`

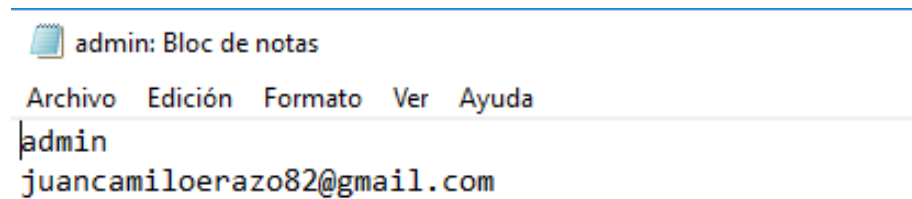


Figura 4: Ejemplo de administrador






Nombre	Fecha de modifica...	Tipo
 admin	17/05/2018 4:17 p.	Chrome HTML Do...
 base	15/05/2018 10:13 a...	Chrome HTML Do...
 edumon	18/05/2018 11:58 a...	Chrome HTML Do...
 edumonAntiguo	5/05/2018 6:02 p. m.	Chrome HTML Do...
 login	18/05/2018 7:10 p.	Chrome HTML Do...

Figura 5: Plantillas de la aplicación

- El archivo llamado preguntas.json está en formato de *JAVASCRIPT OBJECT NOTATION* o mas conocido como json

Python se encarga de obtener su valor y procesarlo con el servidor para que después sea enviado a la página web.

- Los archivos con extensión .py son archivos Python, en estos se almacenan los datos del servidor y las funciones requeridas para la validación de datos. Son la parte mas indispensable del servidor ya que se encargan del backend. Estos pueden ser modificados desde algún editor de texto cualquiera o desde el mismo Python.












 datos_estudiante	18/05/2018 7:12 p.	De
 insertarPregunta	8/05/2018 10:36 p.	Ar
 login	18/05/2018 7:12 p.	De
 main	18/05/2018 11:43 a...	Ar
 mapa1	10/05/2018 3:32 p.	De
 modificar	18/05/2018 11:37 a...	Ar
 pre	14/05/2018 2:02 p.	Ar
 preguntas_json	18/05/2018 11:27 a...	Ar
 registrarUsuario	14/05/2018 3:19 p.	Ar
 respuestas	8/05/2018 10:33 p.	De
 respuestas_correctas	8/05/2018 10:33 p.	De

Figura 6: Archivos con extensión .py

```
app = Flask(__name__)

@app.route('/') #root
def index():
    return render_template('login.html')

@app.route('/login', methods = ['POST', 'GET'])
def game():
    preguntas = obtenerJson()
    if request.method == 'POST':
        usuario = request.form['usuario']
        contrasena = request.form['contrasena']
        print("\n",usuario,"<-- USUARIO INGRESADO")
        print("\n",contrasena,"<-- CONTRASEÑA INGRESADA")
        registro = True

        if validarUsuario(usuario,contrasena)==True:
            if validarAdmin(usuario)==True:
                return render_template('admin.html')
            else:
                datos = listaDeDatos(usuario)
                return render_template('edumon.html',correo=usuario, datos=datos,preguntas1=preguntas)
        else:
            return render_template('login.html',registro=False,usuario=usuario)
    else:
        return render_template('login.html',registro=True)

    return render_template('login.html',registro=True)
```

Figura 7: Archivo principal .py desde Sublime Text 3

## 5. Iniciando el servidor

Hay 2 posibles formas para la inicialización del servidor, la primera es a través del mismo Python pero no se recomienda ya que es susceptible a errores y fallos.

Iniciaremos el servidor desde la **Consola de Comandos de Windows**, po-

demos acceder a ella escribiendo `cmd` y presionando la tecla Enter en la barra para rutas de la carpeta que contiene nuestro servidor.

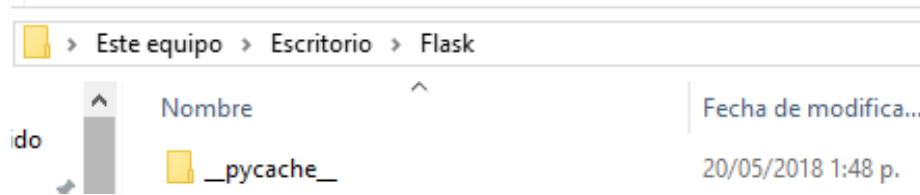


Figura 8: Barra de direcciones de windows

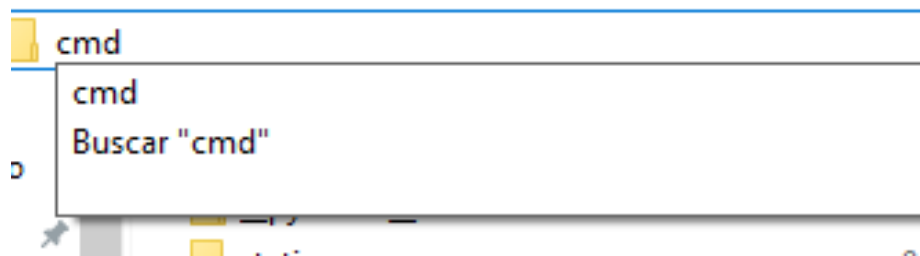


Figura 9: Llamando a la Consola

Después de que la Consola de Comandos haya cargado en la ruta de nuestro juego procedemos a escribir ***python main.py***.

Una vez escrito el comando veremos que aparece la ruta de nuestro servidor, la debemos copiar y ponerla en nuestro navegador.

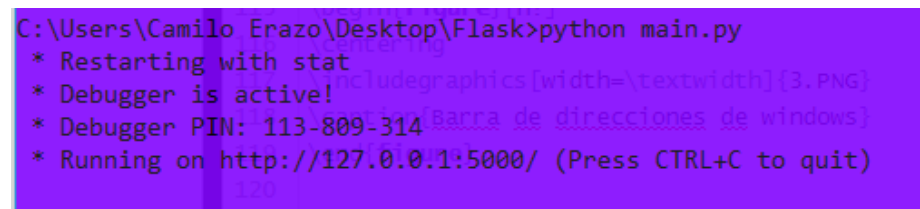


Figura 10: Llamada al servidor

## 6. Registrando usuarios

- Para registrar un usuario debemos ingresar como administradores en el login del servidor, este será desplegado automáticamente apenas abramos la página web.
- Una vez hayamos ingresado como administrador, veremos un menú con los datos necesarios para registrar un estudiante. Deberemos tener mucho cuidado ya que debemos ingresar un correo bien escrito y existente.
- Ahora simplemente debemos asegurarnos de registrar todos los datos asociados a ese usuario teniendo en cuenta las siguientes indicaciones:
  1. El correo no debe existir en la base de datos
  2. El Id de usuario no debe existir en la base de datos
  3. Tenemos que especificar correctamente el grado del estudiante 0-11
  4. Debemos rellenar el campo de administrador y usuario correctamente
- Para estar seguros de que estamos registrando usuarios correctamente podemos verificar el archivo plano que contiene los datos.

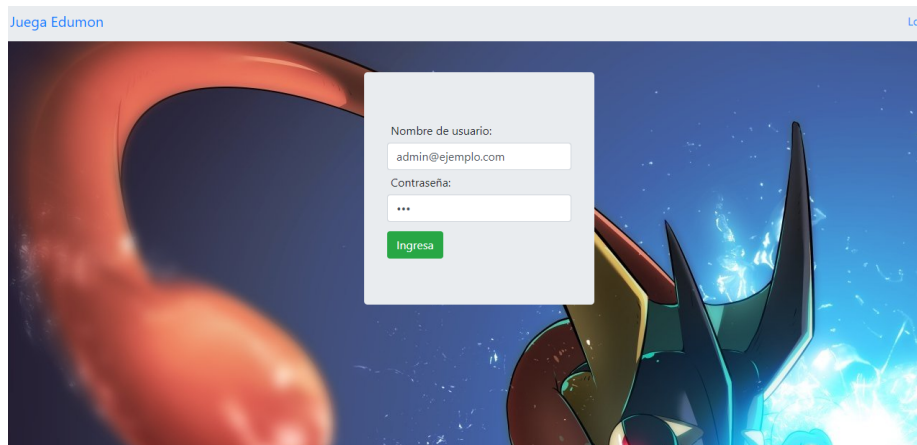


Figura 11: Ingresando como administrador

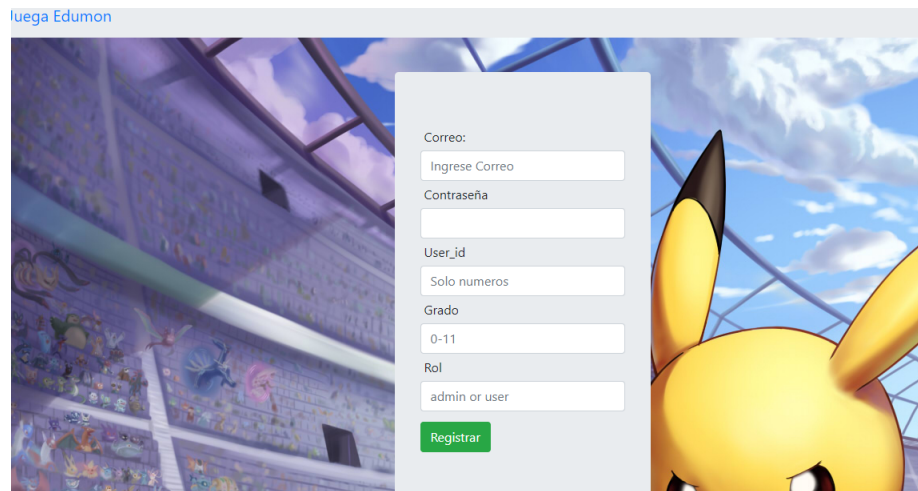


Figura 12: Registrando usuario

## 7. Agregando preguntas

Para agregar una pregunta debemos hacerlo manualmente siguiendo estos pasos:

- Entrar a la carpeta de nuestro servidor
- Abrir el archivo **preguntas.json**
- Insertar correctamente la pregunta en base al modelo del json. La primera llave corresponde al grado "0", la segunda a la materia "matemáticas".

pre	14/05/2018 2:02 p.	Archivo PY	1 KB
preguntas.json	22/05/2018 10:39 a...	Archivo JSON	34 KB
registrarUsuario	20/05/2018 12:20 ...	Archivo PY	5 KB
respuestas	8/05/2018 10:33 p.	Documento de tex	1 KB
respuestas_correctas	8/05/2018 10:33 p.	Documento de tex	1 KB
user_contrasena	20/05/2018 11:39 a...	Documento de tex	1 KB

Figura 13: Ubicación de el archivo contenedor



```
"6":{
  "matematicas": [
    {
      "id": 0,
      "pregunta": "5x = 15 Cual es el valor de x",
      "respuestas": [
        5,
        12,
        4,
        3
      ],
      "correcta": 3
    }
  ]
}
```

Figura 14: Ejemplo de una pregunta en formato json

## 8. Modificar el juego

Para modificar el juego vamos a tener que ingresar al archivo javascript llamado **edumon.html** en él encontraremos la plantilla con los datos de la página web y el código que se encarga de la parte gráfica de la aplicación.

### 8.1. Ingresar a las carpetas del juego

Los archivos estaticos es edumon estan en la siguiente ruta:  
\\flask\\static\\imgs

### 8.2. Mapas

- Para modificar los mapas de Edumon necesitaremos importar nuevas imágenes desde el documento HTML de javascript. modificando su ruta de lectura desde el código base.
- Debemos cerciorarnos de que nuestro mapa esta en una resolución de 1024x608 pixeles ya que estas son las medidas exactas de las matrices que se usaran para su movimiento.
- Los números 1 de la matriz delimitan al personaje, y los números diferentes a 0 y 1 son los indicadores de que hay una materia.
- Para modificar los eventos de entrada a una materia debemos definir el número en su puerta o en el lugar que queramos que el personaje entre, el código se encargara de la validación cuando presione alguna tecla cerca a ese punto.

### 8.3. Preguntas

Lo podemos encontrar en la sección 7

Figura 15: Matriz con los datos de un mapa

Figura 16: Evento para la entrada a una materia

Figura 17: Ubicación de el archivo contenedor

### 8.4. Personajes

Para modificar los personajes debemos hacer el mismo procedimiento que con los mapas, tener una imagen ya preparada e importarla al código base de Javascript así:

```
hero1.src = "{{url_for('static', filename = 'imgs/hero.png')}}"  
hero2.src = "{{url_for('static', filename = 'imgs/hero2.png')}}"  
hero3.src = "{{url_for('static', filename = 'imgs/hero3.png')}}"
```

Figura 18: Ejemplo de los 3 personajes bases del juego

## 9. Link del repositorio

Todos los avances del proyecto están contenidos en un repositorio de github, podemos acceder a el por medio del siguiente enlace.