

## Material Imprimible

### Curso de Aplicaciones Prácticas de Python

#### Interacción con Excel

Se puede colocar como una tarea programada en los sistemas operativos y que corra cada una hora como ejemplo.

Dos herramientas que utiliza Python de la colección de paquetes que desarrolla la comunidad (empresas que desarrollan esto)

Herramientas:

- 1) **Pandas**, para que Python pueda leer un archivo de Excel y escribir
- 2) **Requests**, para adquirir en tiempo real la cotización del dólar en Argentina, se podría ver cómo hacer en otros países
- 3) **Openpyxl windows 10 (64 bit)**. Para comenzar a interactuar con Excel y otras hojas de cálculo.
- 4) **Xlrd** Para extraer información de hojas de cálculo (en los formatos. xls y. xlsx) de Microsoft Excel.

**Pandas** es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para hacer que trabajar con datos estructurados (tabulares, multidimensionales, potencialmente heterogéneos) y series temporales sea fácil e intuitivo. Su objetivo es ser el bloque de construcción de alto nivel fundamental para hacer un análisis de datos práctico y real en Python. Además, tiene el objetivo más amplio de convertirse en la herramienta de análisis / manipulación de datos de código abierto más potente y flexible disponible en cualquier idioma. Ya está bien en camino hacia este objetivo.

Pandas es muy adecuado para muchos tipos diferentes de datos:

- Datos tabulares con columnas heterogéneamente tipadas, como en una tabla SQL o una hoja de cálculo de Excel.
- Datos de series temporales ordenados y desordenados (no necesariamente de frecuencia fija).
- Datos de matriz arbitrarios (tipados o heterogéneos de forma homogénea) con etiquetas de fila y columna.
- Cualquier otra forma de conjuntos de datos observacionales/estadísticos. Los datos en realidad no necesitan ser etiquetados en absoluto para ser colocados en una estructura de datos pandas.

Las dos estructuras de datos principales de pandas, Series (1-dimensional) y DataFrame (2 dimensiones), manejan la gran mayoría de los casos de uso típicos en finanzas, estadísticas, ciencias sociales y muchas áreas de ingeniería. Para los usuarios de R, DataFrame proporciona todo lo que proporciona data.frame de R y mucho más. pandas está construido sobre NumPy y está destinado a integrarse bien dentro de un entorno de computación científica con muchas otras bibliotecas de terceros.

Estas son solo algunas de las características de los pandas:

- Fácil manejo de los datos que faltan (representados como NaN) en punto flotante, así como datos de punto no flotante
- Mutabilidad del tamaño: las columnas se pueden insertar y eliminar de DataFrame y objetos de mayor dimensión
- Alineación de datos automática y explícita: los objetos se pueden alinear explícitamente con un conjunto de etiquetas, o el usuario puede simplemente ignorar las etiquetas y permitir que Series, DataFrame, etc. alineen automáticamente los datos en los cálculos
- Grupo potente y flexible por funcionalidad para realizar operaciones de combinación de aplicaciones divididas en conjuntos de datos, tanto para agregar como transformar datos
- Facilita la conversión de datos irregulares e indexados de forma diferente en otras estructuras de datos de Python y NumPy en objetos DataFrame

- Corte inteligente basado en etiquetas, indexación de lujo y subconjunto de grandes conjuntos de datos
- Combinación y unión intuitivas de conjuntos de datos
- Remodelación y pivote flexibles de conjuntos de datos
- Etiquetado jerárquico de ejes (posible tener varias etiquetas por tick)
- Robustas herramientas de E/S para cargar datos de archivos planos (CSV y delimitados), archivos de Excel, bases de datos y guardar/cargar datos desde el formato HDF5 ultrarrápido
- Funcionalidad específica de series temporales: generación de intervalo de fechas y conversión de frecuencia, estadísticas de ventanas móviles, cambio de fecha y retraso.

Muchos de estos principios están aquí para abordar las deficiencias frecuentemente experimentadas utilizando otros lenguajes / entornos de investigación científica. Para los científicos de datos, trabajar con datos normalmente se divide en varias etapas: munging y limpieza de datos, analizar / modelar, a continuación, organizar los resultados del análisis en una forma adecuada para trazado o visualización tabular. Pandas es la herramienta ideal para todas estas tareas.

Para instalar Pandas y Requests desde pip en el intérprete de comandos

```
pip install pandas o python -m pip install modulo_a_instalar
```

```
pip install requests
```

```
pip install openpyxl
```

```
pip install xlrd
```

Creamos la tabla de Excel y la guardamos en el directorio raíz donde se encontrará nuestro archivo Python.

Abrimos sublime y lo guardamos con la extensión .py

```
import pandas
```

```
import requests
```

```
import statistics (tercera librería)
```

Importamos las tres librerías.

Primero vamos a hacer que el programa obtenga en tiempo real la cotización, aunque el orden es indistinto

```
r= requests.get("https://api.recursospython.com/dollar")
```

Este requests nos sirve para obtener información de algún servicio web, son como páginas webs que se acceden a través de un navegador preparados para que acceda una persona física.

En cambio, los servicios web están pensados para que accedan los programas, pero al igual que las páginas webs también tiene url, es decir, dirección web.

recursospython.com

API - Recursos Python

Use GET /dollar para obtener la cotización del dólar en Argentina. La información se actualiza cada diez minutos desde el Banco Nación. Si tiene un token, use, en su lugar, GET /api/dollar?token=SU-TOKEN-AQUÍ.

El servicio es gratuito. No obstante, establecemos un límite de 100 peticiones diarias para usuarios sin token generado. Utilice el siguiente formulario para obtener uno (¡también gratis!): [Generar token](#).

¿Ahora de ese requests que quiero obtener? La cotización del dólar.

```
cotizacion= r.json().values()
```

**Json** es el formato en el que viaja la información, puede ser xml, texto, etc. Pero los servicios web trabajan con este formato y se encarga requests de interpretarlo.

**Values** nos trae los valores de esa cotización (cotización de compra y de venta). Podríamos elegir uno, pero vamos a hacer un promedio de estas dos cotizaciones, de ahí la librería statistics

```
cotizacion= statistics.mean(r.json().values())
```

Una de las funciones statistics es mean “promedio o media de una lista de números”

```
print (cotizacion) # lo corremos
```

Ahora, leer el Excel, la columna de precio en dólares, hacer la conversión a partir de la cotización que ya está guardada y que lo muestre en precio en pesos.

### Generamos una función

```
def dolar_a_pesos(precio_en_usd): # le damos los parámetros que tenemos en Excel
```

y lo convertimos a pesos

```
return precio_en_usd * cotizacion # retorna el valor convertido
```

Ahora el último paso de cómo usamos esta función:

Leemos el Excel desde Python

```
excel=pandas.read_excel("productos.xlsx")
```

La propiedad read\_ puede leer varios formatos no solo Excel como por ejemplo html, Json, csv, clipboard, sql, table, etc.

Ahora nos ubicamos en los campos de Excel

```
excel["Precio en Dolares"].apply(dolar_a_pesos)
```

Ubicamos la columna entre corchetes y aplicamos la función a cada fila con el método apply.

Agregamos en que campo donde queremos almacenar los resultados de la función con la columna “Precio en dólares”

```
excel["Precio en pesos"]=excel["Precio en Dolares"].apply(dolar_a_pesos)
```

Hasta acá Python hizo toda la conversión en la memoria, debemos reflejarlo en un sistema de archivos que puede ser el mismo Excel o en un archivo nuevo.

Lo exportamos

```
excel.to_excel("ProductosPesificados.xlsx", index=False)
```

### **Nuestro programa terminado quedaría así**

```
import pandas
import requests
import statistics
r= requests.get("https://api.recursopython.com/dollar")
cotizacion= statistics.mean( r.json().values() )
print (cotizacion)
def dolar_a_pesos(precio_en_usd):
    return precio_en_usd * cotizacion
excel=pandas.read_excel("productos.xlsx")
excel["Precio en pesos"]=excel["Precio en Dolares"].apply(dolar_a_pesos)
excel.to_excel("ProductosPesificados.xlsx", index=False)
```

### **Base de datos alumnos**

Proyecto inscripción de alumnos desde mesa de entrada de un instituto.

Para aplicaciones web se utilizan varias tecnologías como html, css, JavaScript, etc.

Con Python tenemos una herramienta para utilizar sin necesidad de utilizar nada de estas herramientas dichas anteriormente y es Web2py <http://www.web2py.com/> que nos permite crear aplicaciones web de forma totalmente gratuita y segura.

A este tipo de tratamiento se le llama DRAP (Desarrollo rápido de aplicaciones)

- Descargamos el web2py

No seguro | www.web2py.com Traducido

[Casa](#) [acerca de](#) [Descargar](#) [Documentos y recursos](#) [Apoyo](#) [Colaboradores](#)

## WEB2PY Marco Web

Marco de pila completa de código abierto gratuito para un rápido desarrollo de *Seguro* y aplicaciones portátiles basadas en bases de datos. Escrito y programable en Python (versión 3 y 2.7).



Descarga ahora  
Ejemplos rápidos  
Pruebo ahora en línea  
Sitios impulsados por web2py

InfoWorld.com  
★ 2012 AWARDS ★  
**TECHNOLOGY OF THE YEAR**

Versión actual: 2.20.4-stable+timestamp:2020.05.03.05:18:50 (Licencia LGPLv3)

### Baterías incluidas

Todo lo que necesita en un paquete, incluido el servidor web multiproceso rápido, la base de datos SQL y la interfaz basada en web. No hay dependencias de terceros, pero funciona con herramientas de terceros.

### IDE basado en web

Cree, modifique, implemente y administre aplicaciones desde cualquier lugar con su navegador. Una instancia web2py puede ejecutar varios sitios web utilizando bases de datos diferentes. Pruebe el [tour interactivo](#).

### Documentos extensos

Comience con algunos ejemplos rápidos, luego lea el Manual y el [Documento de la EstrellaRojo](#). Videos, y únase a un grupo de usuarios para la discusión. Aproveche la [Plataforma Python AnarchoV](#).



## web2py™ Descargar

Para Python 3.7	Para Python 2.7	Para probadores	Para desarrolladores
<a href="#">Binarios de Windows</a>	<a href="#">Binarios de Windows</a>	<a href="#">Binarios de Windows</a>	<a href="#">Repositorio Git</a>
<a href="#">Mac binarios</a>	<a href="#">Mac binarios</a>	<a href="#">Mac binarios</a>	<a href="#">Manual</a>
<a href="#">Código fuente</a>	<a href="#">Código fuente</a>	<a href="#">Código fuente</a>	<a href="#">Documentos de código</a>
		<a href="#">Cambiar registro</a>	<a href="#">Informar de un error</a>

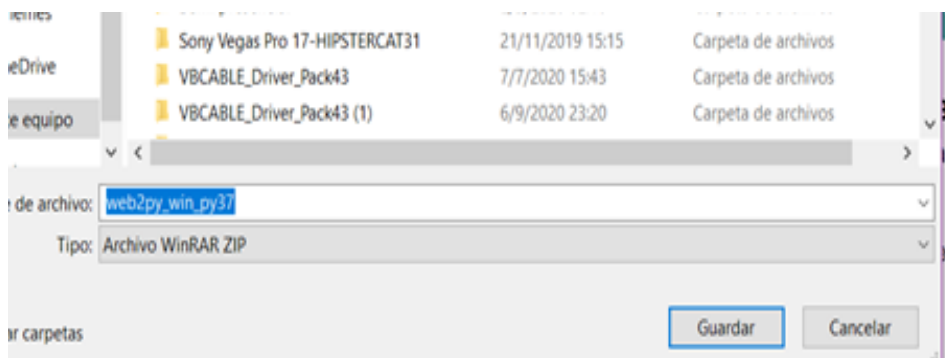
La versión del código fuente funciona en Windows y la mayoría de los sistemas Unix, incluidos **Linux**, **BSD** y **Mac**. Requiere Python 3.5+ (recomendado para nuevos proyectos) o Python 2.7+ (estable, para su uso con aplicaciones heredadas) ya instalado en su sistema.

También hay paquetes binarios para Windows y MacOs. Incluyen la versión 3.7.4 o 2.7.16 del intérprete de Python, por lo que no es necesario tenerlo preinstalado.

### Instrucciones

Con los paquetes binarios, después de la descarga, simplemente descomprimirlo y luego haga clic en web2py.exe (Windows) o web2py (MacOs).

Tenga en cuenta que en las versiones recientes de MacO (10.12+) podría tener problemas en la ejecución del programa

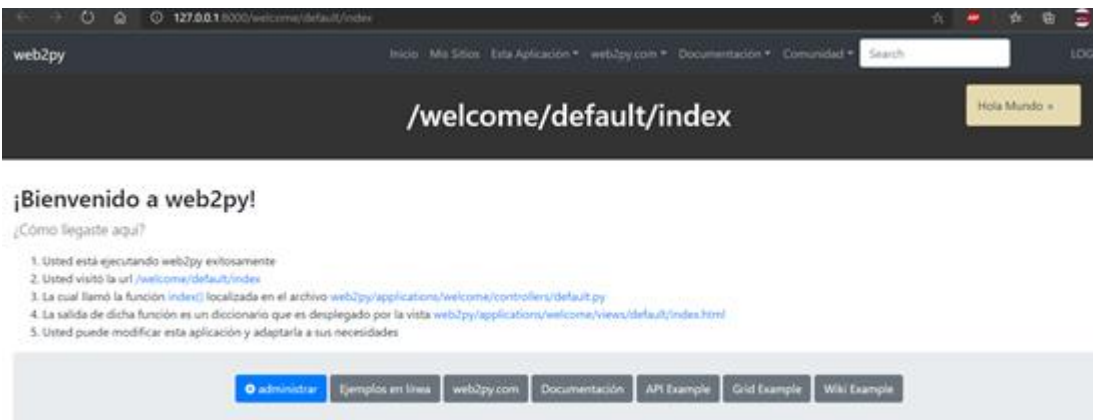


- Descomprimos el archivo
- Y ejecutamos la instalación





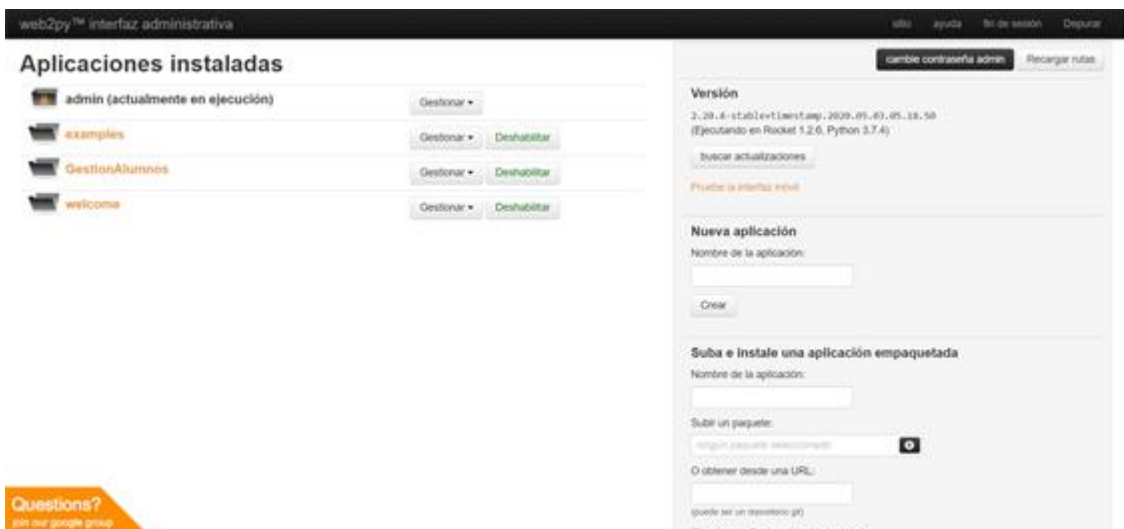
- Elegimos una contraseña
- Y clicamos **start server** y nos abre una nueva página en el navegador



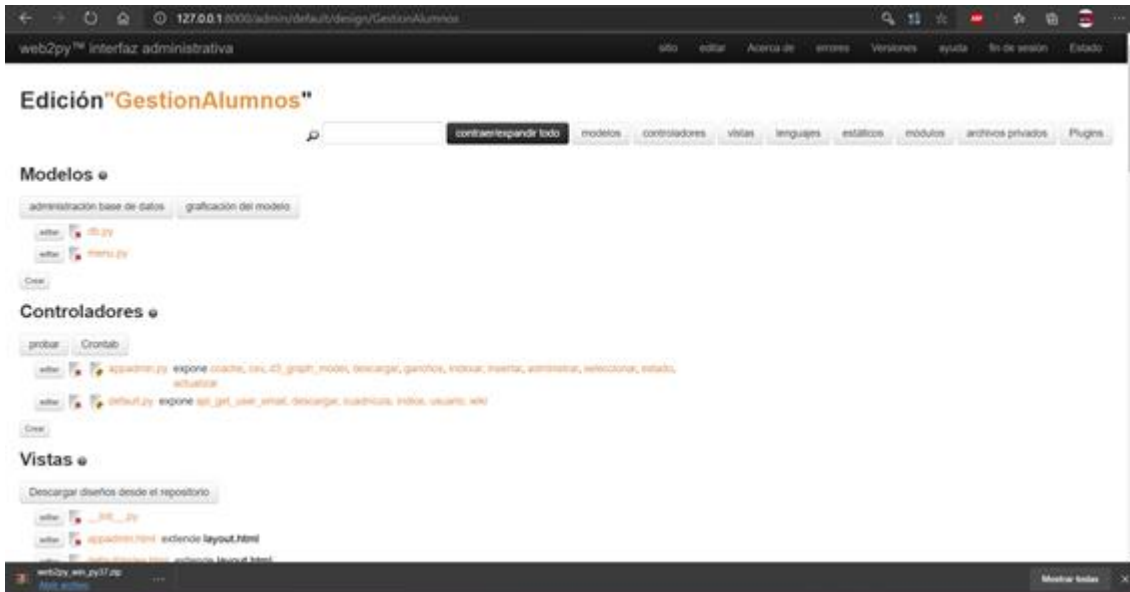
- Clicamos en **administrar**



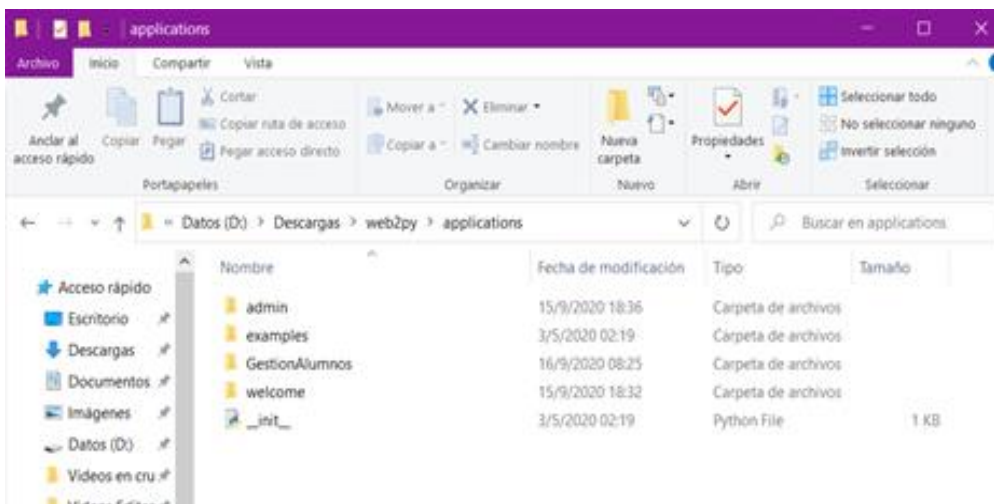
- Y cuando colocamos la contraseña, nos abre la siguiente ventana donde pondremos en la casilla “nombre de aplicación” la aplicación de la base de datos que vamos a generar. En este ejemplo usaremos **GestionAlumnos** y presionamos **generar**



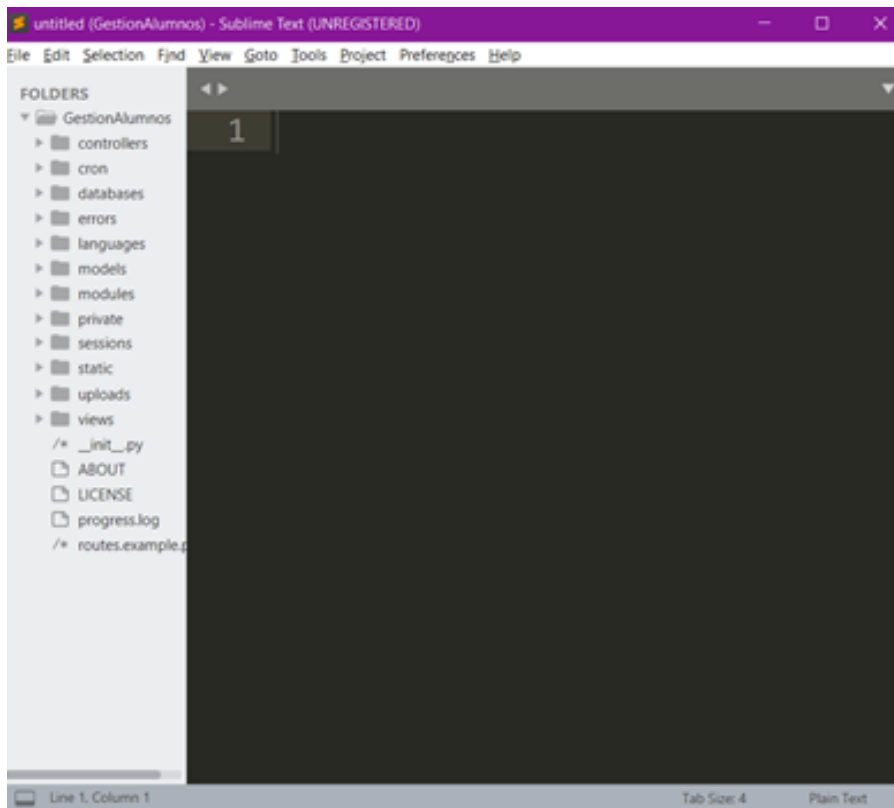
- Nos muestra la siguiente ventana y lo dejamos así.



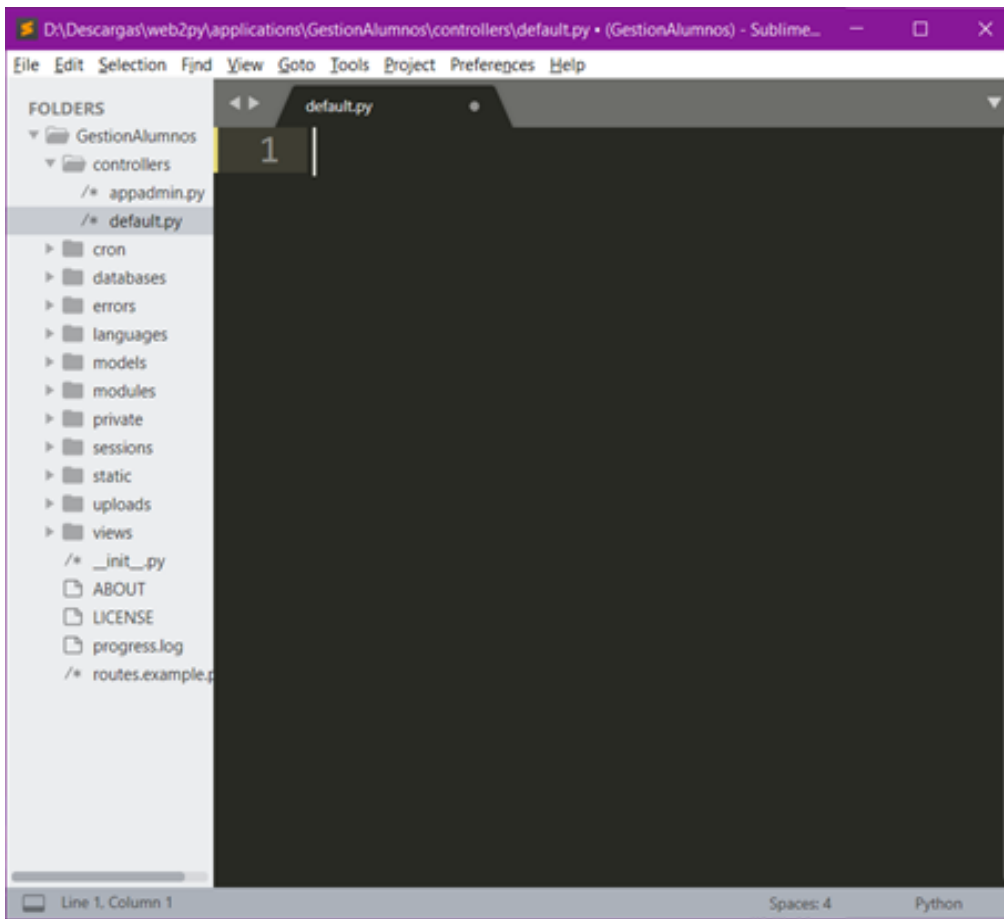
- Cuando creamos una base de datos nos crea además una carpeta que contiene datos e información de la creación, dentro de la carpeta donde instalamos el Web2py dentro de applications



- Arrastramos la carpeta **GestionAlumnos** al sublime



- Nos trae varias carpetas y la que necesitamos expandir es la carpeta **controllers**. Ubicamos sobre la solapa del sublime el que dice **default.py**, luego borramos todo su contenido para poder trabajar con nuestro proyecto dejando la hoja **default.py** en blanco



Ahora definimos una tabla como si fuera una tabla en Excel, pero se creará desde una aplicación web:

```
db.define_table( #codigo para definir la tabla
    "alumno", #nombre de la tabla
    Field("dni", type="integer"), #campos donde tiene el nombre y el tipo de dato que tendrá
    Field("nombre", type="string"),
    Field("fecha_alta", type="date")
)
```

Luego debemos generar un ingreso para la administración de los alumnos a través del navegador web, para eso hacemos una función:

```
def alumnos():
```

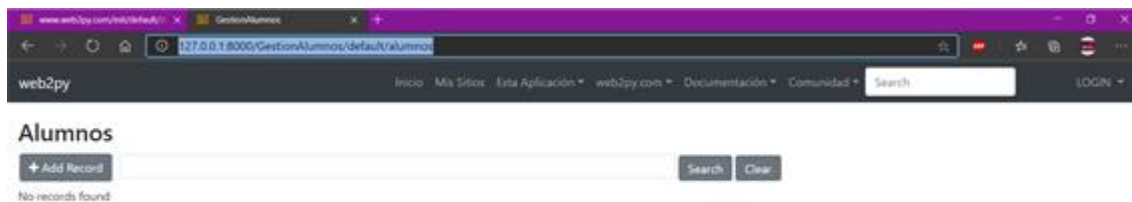
---

```
return {"grid": SQLFORM.grid(db.alumno, user_signature=False)}
```

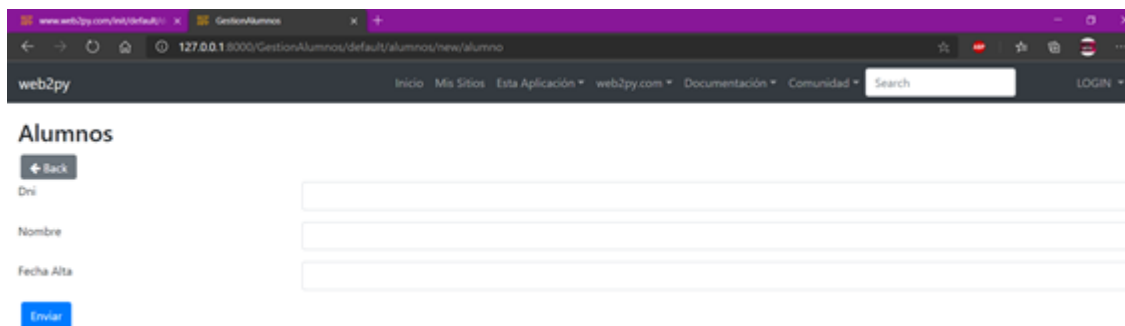
vamos al navegador y escribimos

<http://127.0.0.1:8000/GestionAlumnos/default/alumnos>

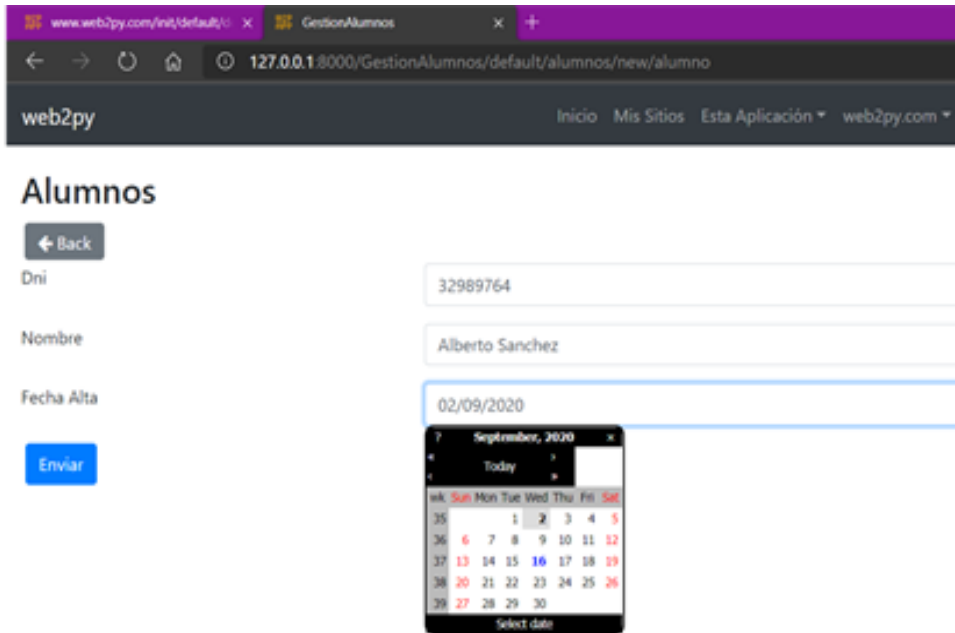
127.0.0.1:8000 esto es porque esta aplicación está corriendo en nuestra computadora, toda la base estará instalada en nuestro disco. Para verla online se debe contratar un servidor web o hosting donde alojar nuestra base de datos



Hasta el momento no tenemos registros, para hacer el registro. Presionamos **Add Record**, este nombre podemos modificarlo luego para que diga, como, por ejemplo: "Nuevo Alumno"



Completamos los cuadros de textos y veremos que tenemos la posibilidad de utilizar un calendario para utilizar en lugar de ingresar manualmente los valores, en **fecha de alta**.



www.web2py.com/init/default/... GestionAlumnos

127.0.0.1:8000/GestionAlumnos/default/alumnos/new/alumno

web2py Inicio Mis Sitios Esta Aplicación web2py.com

## Alumnos

← Back

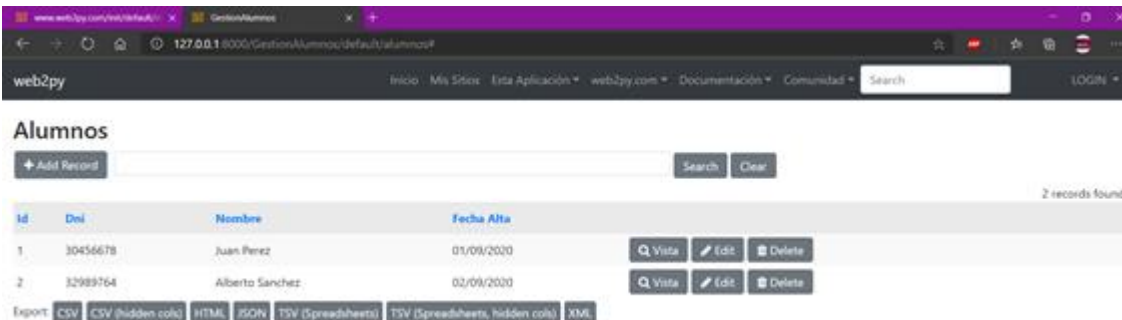
Dni: 32989764

Nombre: Alberto Sanchez

Fecha Alta: 02/09/2020

Enviar

Ahora veremos cuando presionamos **enviar la lista**, los que fuimos ingresando:



web2py Inicio Mis Sitios Esta Aplicación web2py.com Documentación Comunidad Search LOGIN

## Alumnos

+ Add Record Search Clear

2 records found

Id	Dni	Nombre	Fecha Alta	
1	30456678	Juan Perez	01/09/2020	View Edit Delete
2	32989764	Alberto Sanchez	02/09/2020	View Edit Delete

Export: CSV CSV (hidden cols) HTML JSON TSV (Spreadsheets) TSV (Spreadsheets, hidden cols) XML

Si probamos escribir un texto en el campo **DNI**, veremos que no nos lo permitirá.

Esto se debe a que colocamos como tipo de campo **integer**, es decir, "entero" y Web2Py genera automáticamente una política de validación para que solo se puedan ingresar números enteros.

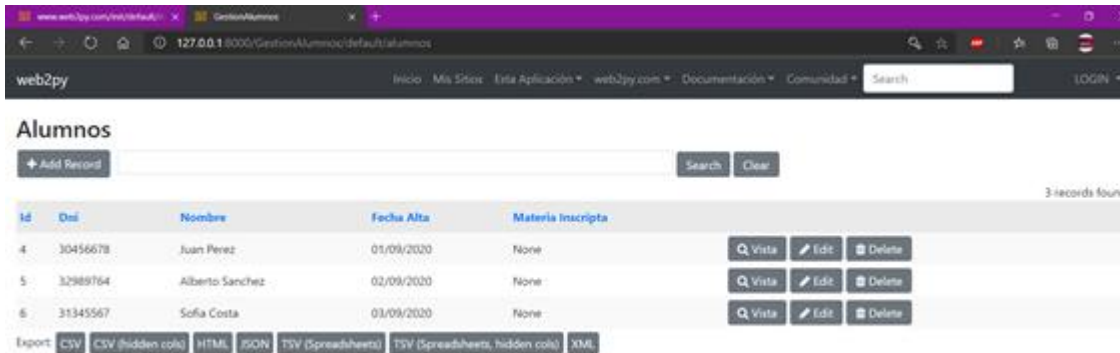
Colocamos tres datos en nuestra base.

Bien, ahora, intentaremos hacer que se pueda ingresar también, a qué curso se pueden inscribir los alumnos.

Agregamos en el **sublime** un campo mas:

```
Field("Materia_Inscripta", type="string", requires=IS_IN_SET(["Python", "Java", "JavaScript"])),
```

En esta oportunidad no queremos que el operador tenga la opción de ingresar datos en ese campo, sino que seleccione de una lista desplegable de una múltiple de opciones.



En este caso, veremos qué materia inscripta está vacío o en este caso dice ninguna.

Podemos también realizar ordenamientos si presionamos los diferentes nombres de los campos. Función que agrega Web2py.

### Actividad:

Realicen los edit. necesarios para modificar los valores de materia inscripta.

Veremos que podemos generar una vista previa para impresión por usuario, además de editar y eliminar.

Podemos utilizar las opciones de exportar en diferentes formatos, como, por ejemplo CSV, que lo podrá leer como un archivo de Excel.

### CSV:

Las siglas "csv" son un acrónimo que significa "comma separated values", que en español significa "valores separados por columnas". Los formatos de tipo csv son bastante sencillos, y suelen representar un tipo de caracteres muy concretos y determinados.

### HTML:

Formato web



**JSON** es el acrónimo para JavaScript Object Notation, y aunque su nombre lo diga, no es necesariamente parte de JavaScript, de hecho, es un estándar basado en texto plano para el intercambio de información, por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros

### **TSV:**

El formato de archivo TSV significa "Valores separados por tabulaciones", y estos archivos valores separados por tabulaciones se crean y utilizan muchas aplicaciones de hoja de cálculo.

### **XML**

Formato web

Tenemos una opción de búsqueda. Claro, para grandes cantidades de datos es mucho más útil que utilizar con tres datos como el ejemplo.

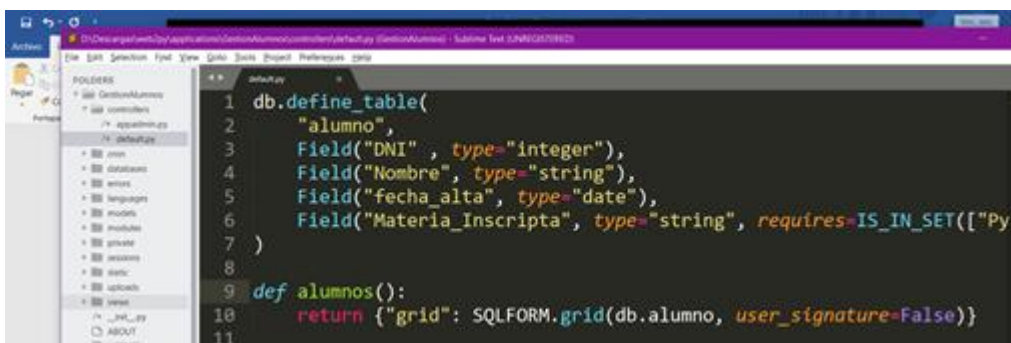
Web2py es una herramienta muy ágil para realizar estas bases de datos, se puede realizar con otros lenguajes diseñadores web, pero sería muy arduo trabajo y demasiado código.

Miren el código con la opción "ver origen de la página" realizando clic derecho con el mouse o, según el navegador, "ver código fuente".

Si quisiéramos vender una base de datos como proyecto propio se puede realizar y cambiaríamos la estética como para darle el estilo propio. Pero debemos tener algo de conocimiento en desarrollo web para generar cambios en la interface.

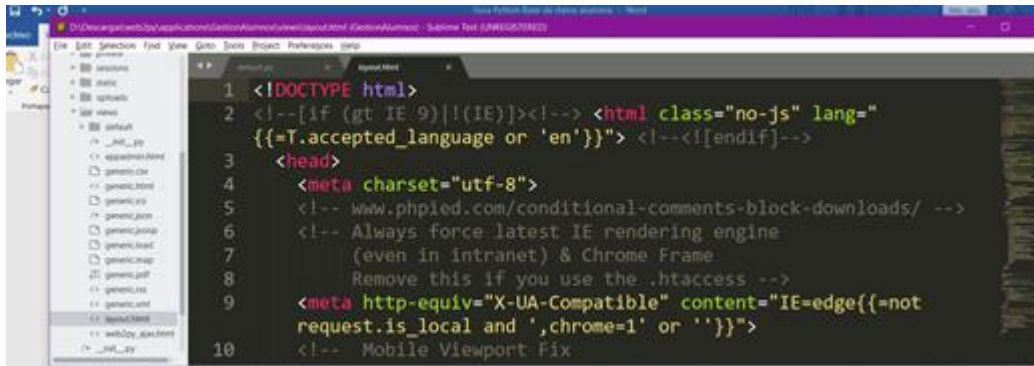
Vemos un ejemplo:

Clic en views, carpeta dentro de nuestra base creada



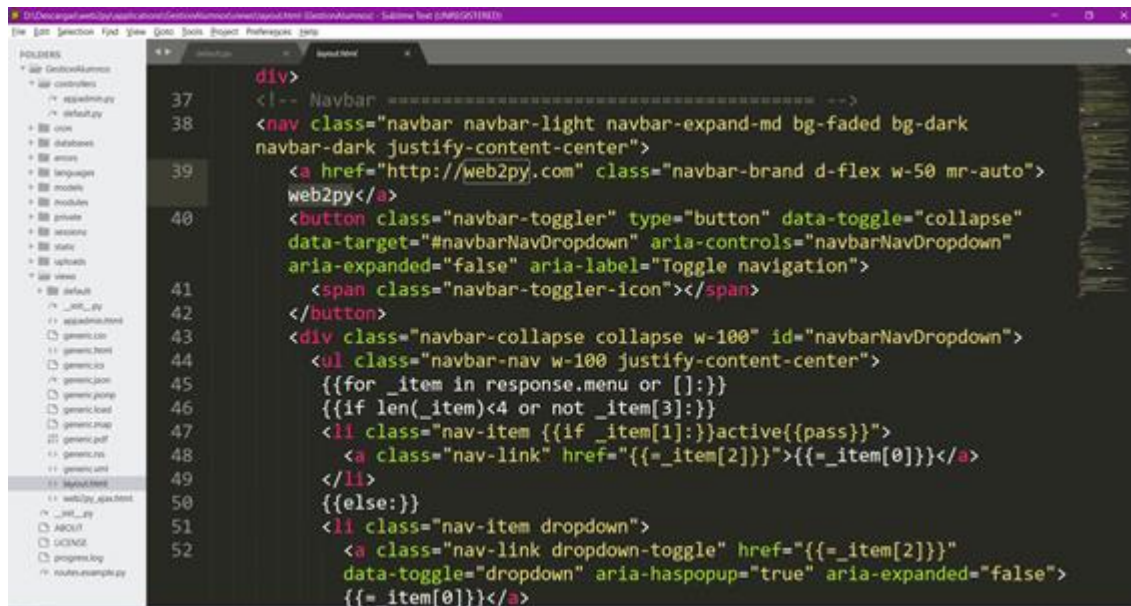
```
1 db.define_table(
2     "alumno",
3     Field("DNI", type="integer"),
4     Field("Nombre", type="string"),
5     Field("fecha_alta", type="date"),
6     Field("Materia_Inscripta", type="string", requires=IS_IN_SET(["Py
7 ]
8
9 def alumnos():
10     return {"grid": SQLFORM.grid(db.alumno, user_signature=False)}
11
```

Luego a la opción de Layout.html donde está nuestro entorno gráfico, por así decirlo, o nuestro diseño web.



```
1 <!DOCTYPE html>
2 <!--[if (gt IE 9)]!(IE)]><!--> <html class="no-js" lang="
  {{T.accepted_language or 'en'}}"> <!--<![endif]-->
3 <head>
4   <meta charset="utf-8">
5   <!-- www.phpied.com/conditional-comments-block-downloads/ -->
6   <!-- Always force latest IE rendering engine
7     (even in intranet) & Chrome Frame
8     Remove this if you use the .htaccess -->
9   <meta http-equiv="X-UA-Compatible" content="IE=edge{{=not
  request.is_local and ',chrome=1' or ''}}">
10  <!-- Mobile Viewport Fix
```

Buscamos web2py dentro de Navbar un comentario dentro del código (Línea 39)



```
37 <div>
38   <!-- Navbar ***** -->
39   <nav class="navbar navbar-light navbar-expand-md bg-faded bg-dark
  navbar-dark justify-content-center">
40     <a href="http://web2py.com" class="navbar-brand d-flex w-50 mr-auto">
  web2py</a>
41     <button class="navbar-toggler" type="button" data-toggle="collapse"
  data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown"
  aria-expanded="false" aria-label="Toggle navigation">
42       <span class="navbar-toggler-icon"></span>
43     </button>
44     <div class="navbar-collapse collapse w-100 id="navbarNavDropdown">
45       <ul class="navbar-nav w-100 justify-content-center">
46         {{for _item in response.menu or []:}}
47         {{if len(_item)<4 or not _item[3]:}}
48         <li class="nav-item {{if _item[1]:}}active{{pass}}">
49           <a class="nav-link" href="{{=_item[2]}}"{{=_item[0]}}</a>
50         </li>
51         {{else:}}
52         <li class="nav-item dropdown">
53           <a class="nav-link dropdown-toggle" href="{{=_item[2]}}"
  data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  {{=_item[0]}}</a>
```

Cambiamos por el logo de nuestro curso, al igual que el link y colocamos, por ejemplo Google, sino iría la dirección de nuestra página web.

```

37 <!-- Navbar
38 <nav class="navbar navbar-light navbar-expand-md bg-faded bg-dark
   navbar-dark justify-content-center">
39 <a href="http://web2py.com" class="navbar-brand d-flex w-50 mr-auto">
   Capacitarte Python</a>
40 <button class="navbar-toggler" type="button" data-toggle="collapse"
   data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown"
   aria-expanded="false" aria-label="Toggle navigation">
41 <span class="navbar-toggler-icon"></span>
42 </button>
43 <div class="navbar-collapse collapse w-100 id="navbarNavDropdown">
44 <ul class="navbar-nav w-100 justify-content-center">
45 {{for _item in r(Definition:
46 {{if len(_item)<
47 <li class="nav-item static/css/calendar.css:5:tive{{pass}}">
48 <a class="nav-link" href="{{=_item[2]}}">{{=_item[0]}}</a>
49 </li>
50 {{else:}}
51 <li class="nav-item dropdown">
52 <a class="nav-link dropdown-toggle" href="{{=_item[2]}}>

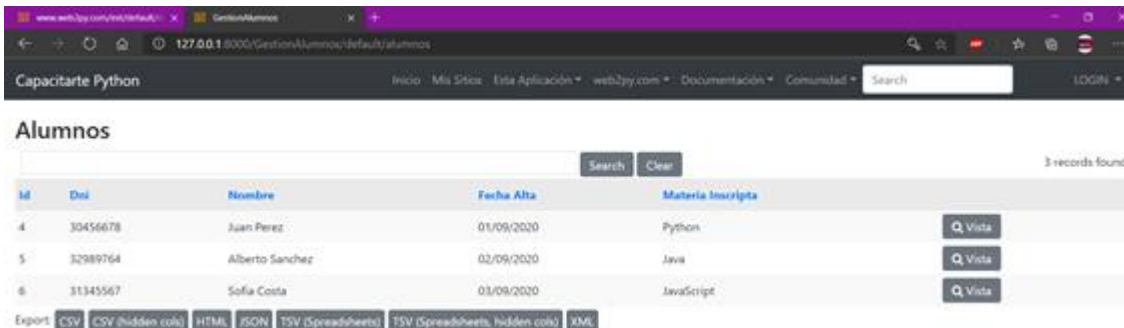
```

## Ahora veremos la sintaxis:

**user\_signature=False** que escribimos al final de nuestro código cuando generamos la base de datos.

Con esta aplicación cualquier persona con acceso a la dirección web puede acceder a dar de altas, bajas o modificaciones de los alumnos sin ninguna restricción. Esto no se puede permitir, si vieron en la parte superior derecha tenemos un login, algo que Web2py tiene en cuenta al momento de realizar aplicaciones web.

Por defecto si no ponemos la opción **user\_signature=False** no podremos modificar nada de la página. Quidemos esta sintaxis.



Id	Dni	Nombre	Fecha Alta	Materia Inscrita
4	30456678	Juan Perez	01/09/2020	Python
5	32989764	Alberto Sanchez	02/09/2020	Java
6	31345567	Sofia Costa	03/09/2020	JavaScript

Se darán cuenta de que solo podemos ver los alumnos en general y en forma individual.

Para que se pueda generar un login, colocamos antes de la función **def alumnos**, la siguiente sintaxis:

**@auth.requires\_login()** # esto indica que para ingresar a la página web debe estar logueado.

### Ahora agregamos el sistema de login de usuario

En [web2py.com/book](http://web2py.com/book) capítulo 9 control de acceso, está la documentación para generar todos los códigos necesarios para generar usuarios y restricciones.

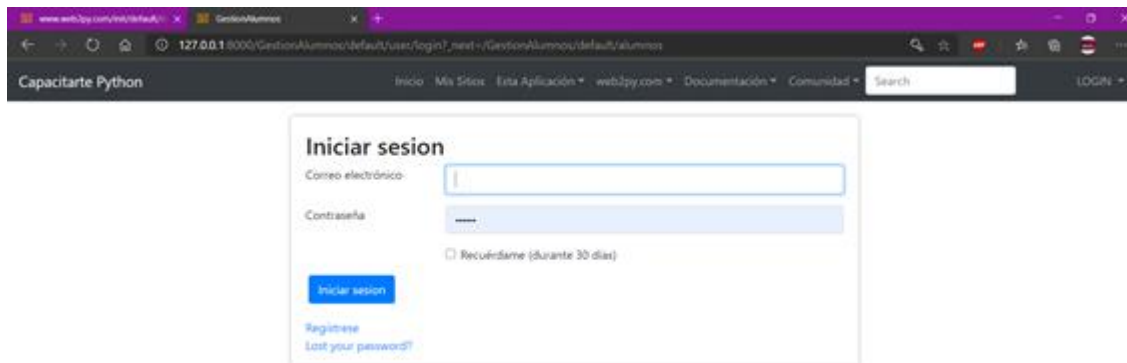
Escribimos esta sintaxis antes de la autenticación requerida recién realizada:

```
def user():  
    return {"form":auth()}
```

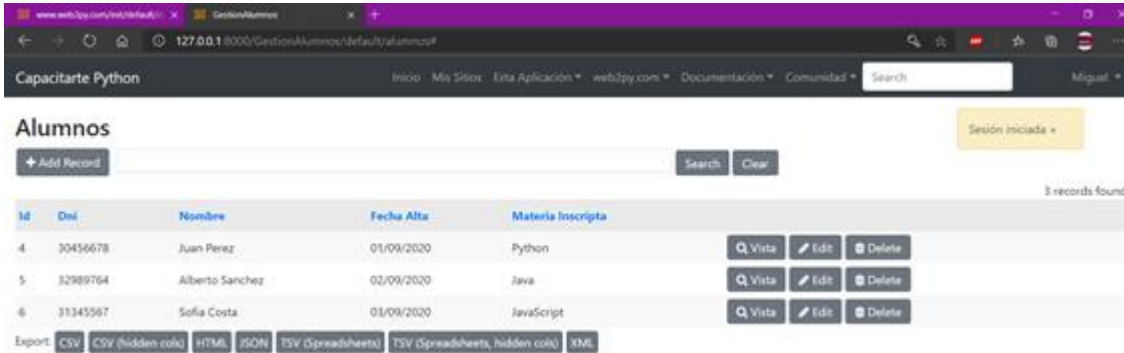
Ahora redireccionamos de la página principal a **alumnos** debajo de la función alumnos:

```
def index():  
    redirect(URL("alumnos"))
```

Ahora actualizamos la página web:



Nos aparece el login de usuarios predeterminado. Ahora no podemos acceder a nuestra base de datos, en este caso, vamos a registrarnos. Veremos que nos muestra el nivel de seguridad de nuestra contraseña en colores, pero nos deja registrar igual si ponemos una contraseña fácil.



Id	Dni	Nombre	Fecha Alta	Materia Inscripta
4	30456678	Juan Perez	01/09/2020	Python
5	32989764	Alberto Sanchez	02/09/2020	Java
6	31345567	Sofia Costa	01/09/2020	JavaScript

Ya nos deja ingresar para realizar las modificaciones y, en esta oportunidad, veremos cómo cambia el login por nuestro nombre.

Nos deja registrarnos sin tantos requisitos como confirmar correo electrónico u otro medio de confirmación, pero se puede configurar.

Lo que debería pasar es generar un usuario y contraseña a los encargados o encargadas de ingresar a los alumnos:

```
db.define_table( #codigo para definir la tabla
    "alumno", #nombre de la tabla
    Field("dni", type="integer"), #campos donde tiene el nombre y el tipo de dato que tendrá
    Field("nombre", type="string"),
    Field("Materia_Inscripta", type="string", requires=IS_IN_SET(["Python", "Java", "JavaScript"])),
    Field("fecha_alta", type="date")
)
@auth.requires_login()
def alumnos():
    return {"grid": SQLFORM.grid(db.alumno)}

def user():
    return {"form":auth()}
```

```
def index():  
    redirect(URL("alumnos"))
```

fuentes: <https://ichi.pro/es/12-increibles-funciones-de-pandas-y-numpy-38368672630200>