

TALLER:

“Proyectos Web en Python”

django



Jorge Díaz M.
Jorge.diazma@sansano.usm.cl
Segundo Semestre 2019



**¿Cuáles son los objetivos
del taller?**





Objetivos del Curso

Comprender las etapas necesarias para desarrollar un Proyecto Web

En este taller, definiremos los enfoques *backend* y *frontend*, aprendiendo las nociones básicas que los distinguen mientras trabajamos la misma cantidad de sesiones en cada aspecto.

Desarrollar un Proyecto Web en Django

Dado que Python es un lenguaje que nos es tan natural (lo utilizamos en programación), en este taller aplicaremos el framework Django para desarrollar nuestro sitio web.

Cabe mencionar, que las slides y las explicaciones estarán hechas para Python 3.6 en adelante, ya que el django para Python 2.7 se quedará sin soporte el próximo año.

Para mayor información sobre Django <https://www.djangoproject.com/start/>



Objetivos del Curso

Entornos Virtuales

Desde el inicio del taller (desde esta sesión), veremos la importancia de los entornos virtuales en la realización de un trabajo más prolijo. Además, notaremos la relevancia de estos cuando desplaguemos nuestro proyecto en el servidor.

Patrón MVC, HTML y CSS

Es evidente que necesitaremos trabajar un poco en el lenguaje de etiquetado como lo es HTML, pero lo haremos más robusto utilizando css y plantillas. Todo esto, estará acompañado de cómo el patrón MVC está presente en el framework Django

Para mayor información sobre Django <https://www.djangoproject.com/start/>



Objetivos del Curso

Deploy en servidor de los proyectos

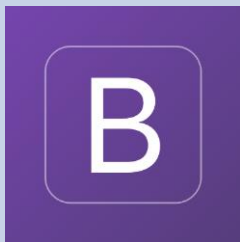
En la fase final de nuestro taller, pondremos en práctica todo lo que aprenderemos subiendo nuestro software a un servidor y esto, junto con la compra de un dominio, podremos pasar por todas las etapas del desarrollo de un proyecto real en Django.

Trabajo Freelance y Boletas de Honorarios

Como extra, veremos en que consiste el trabajo freelance, cómo emitir boletas de honorarios, qué se necesita para esto y qué es el 10% de descuento en dichas boletas.



Para mayor información sobre Django <https://www.djangoproject.com/start/>



NGINX

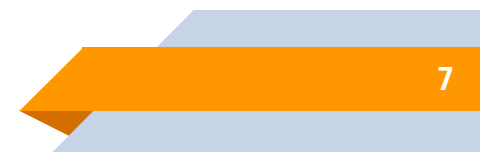


¿Qué aprenderemos en este taller?



¿Cómo será evaluado el Taller?

- 1) Actividades en clases (9 en total y se entregan hasta las 13:55 hrs del jueves siguiente al día que se dió inicio a la actividad)
- 2) Proyecto del Taller (se explicará más adelante)
- 3) Asistencia al taller (10 sesiones)





Formula de Aprobación

■ Promedio Actividades Evaluadas (AE)

■ Presentación Final Proyecto (PF)

■ Asistencia (A)

$$NF = 10\% * A + 40\% * AE + 50\% * PF$$

Si $NF \geq 55$ se aprueba con NF, en caso contrario $\text{MIN}(54, NF)$

(*) 80% asistencia mínima al taller para aprobar, sino se reprueba con $\text{MIN}(54, NF)$



Calendario

SESIÓN	FECHA	CLASE
1	22-08-2019	INTRODUCCIÓN, MOTIVACIÓN Y NIVELACIÓN
2	29-08-2019	PATRÓN MVC Y DJANGO
3	05-09-2019	HTML Y CSS
4	12-09-2019	MODELOS DE DATOS Y ADMIN DJANGO
5	26-09-2019	BOOTSTRAP Y JAVASCRIPT
6	03-10-2019	FORMULARIOS Y SESIONES
7	10-10-2019	PRESENTACIÓN DE AVANCE DE PROYECTOS
8	17-10-2019	SERVIDORES, HOSTING, DNS Y DOMINIOS
9	24-10-2019	DEPLOY EN SERVIDOR
10	07-11-2019	PRESENTACIÓN FINAL

“

¿En qué consiste el Proyecto del Taller?



Puede ser Individual o en pareja

A person with short brown hair, seen from the back, is looking at a wall covered in various papers, photos, and diagrams. The papers include sketches of people, flowcharts, and photographs of interior spaces. The person is wearing a grey and black striped sweater. The text "¿Qué debería tener mi proyecto?" is overlaid in yellow on the person's head and shoulders.

¿Qué debería tener mi proyecto?



ELECCIÓN DE PROYECTO

Programación

Bases de
Datos

Análisis y
Diseño

Página Web Básica

Si eres un(a) estudiante que no ha cursado aún base de datos, te recomiendo que autodefinas un Proyecto que consista en la creación de una página web básica, incluyendo modelos de datos estaticos no relacionados y formulario de contacto. Ejemplos: Blog, página corporativa, etc

Plataforma Web Intermedia

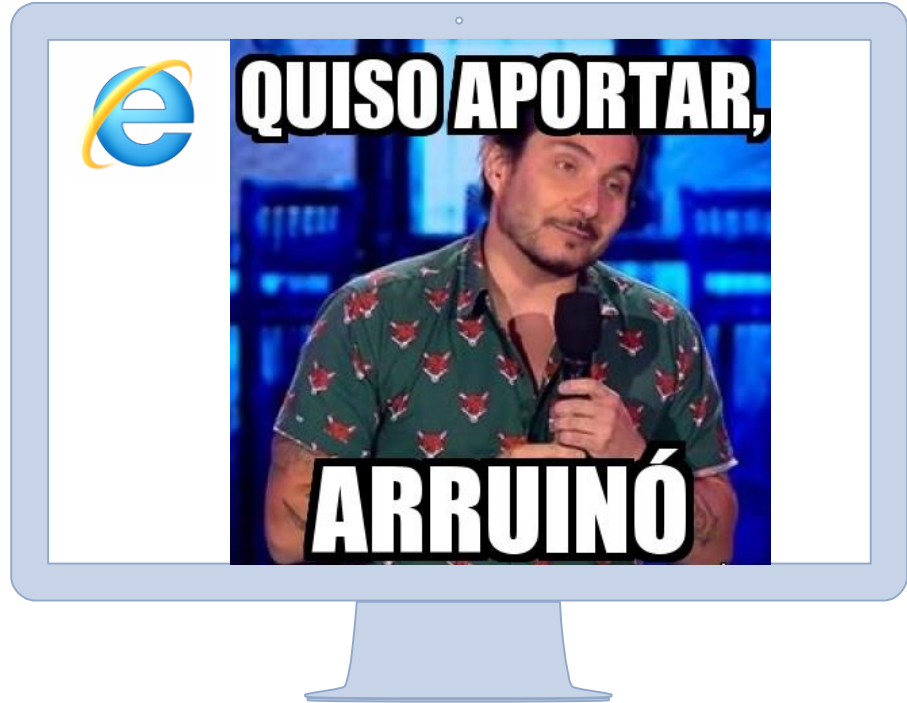
Si ya cursaste Bases de Datos, te recomiendo que definas un Proyecto que incluya modelos de datos relacionados, sesiones y todo lo que conlleva hacer una página web básica. Ejemplos: Instagram, Foro, etc

Sistema Web

Si ya cursaste Análisis y diseño de software, te recomiendo que hagas una Sistema Web, incluyendo todo lo de un Proyecto intermedio, pero utilizando un motor de bases de datos diferente a sqlite. Además, puedes intentar hacer un Proyecto con enfoque de microservicios (~~muy avanzando, pero cabe mencionarlo~~) utilizando el framework Rest junto a Django. Ejemplos: Página de ventas, Sistema de inventario con diferentes tipos de usuario, Servicio basado en microservicios, etc.

Nota

Recuerda que el proyecto del taller lo subiremos a un servidor y adquirirás un dominio que deberá (después de que se le enseñe) configurarlo y enlazarlo con su proyecto.





**¡Y eso fue todo por
hoy!**



Mentira, manos a la obra

Ups, aún no me presento...



¡Bienvenidos!

Jorge Díaz M.

Jorge.diazma@sansano.usm.cl

Full Stack Developer Django



Experiencia Trabajando



Ayudantía de Estructuras Discretas 2018-2 y 2019-1

Ayudantía Optimización 2019-2

Sesión 0

Motivación



Motivos para hacer un framework en Django

- Es un framework MVC puro
- Es para Python, y bueno, es Python
- Favorece las metodologías ágiles
- Es altamente demandado en el mercado
- Está con soporte vigente y se aproximan nuevas versiones
- Simple y sencillo de comprender
- Porque, literalmente, es el mejor para comenzar en el mundo del desarrollo de software

Ofertas de trabajo Django

<https://www.getonbrd.cl/empleos/tag/django>



Sesión 1.1

Proyectos y sus involucrados

Backend, Frontend, dominios y
servidores



¿Qué es un Proyecto Web?

- Es algún programa que está en la web
- Comúnmente soluciona un problema o satisface una necesidad
- Es Web debido al auge de la Internet y la necesidad de estar continuamente conectado

En síntesis, “algo” que se puede visualizar en la web (para el enfoque de este taller)

El ciclo de vida de un Proyecto

Lo ideal vs lo que siempre ocurre





En un mundo ideal

Idea Inicial



"Alguien"
necesita
solucionar un
problema

CLIENTE



"Alguien"
desarrolla un
Proyecto Web
que soluciona
su problema

DESARROLLADOR

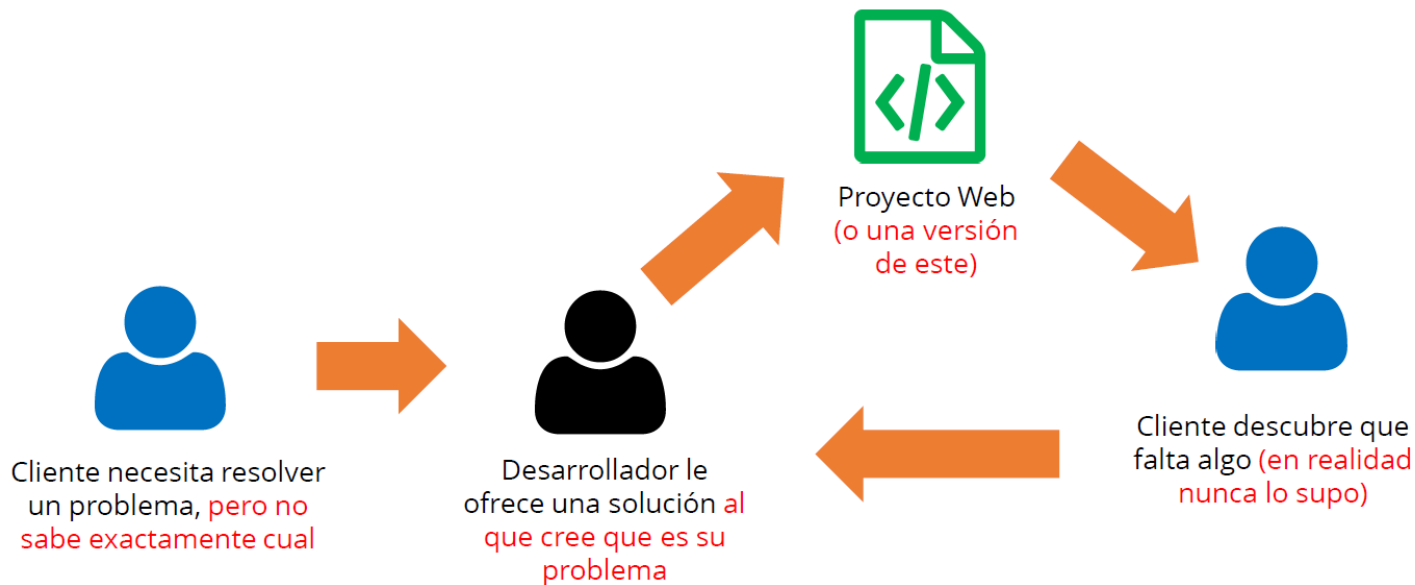


Proyecto Web
Terminado

CÓDIGO



Lo que realmente pasa



Proyectos sin fin



Pero en realidad está bien

Varias de las metodologías de desarrollo de software se basan en un proyecto básico o modular que va creciendo hasta tener el proyecto final que soluciona el problema, siempre y cuando todo lo intermedio se hizo correctamente.





Tipos de desarrolladores

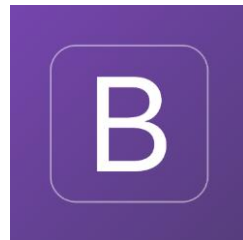
django

NGINX

Backend

Proyecto
Full Stack

Frontend





Por último antes de codear

Servidores

- Es simplemente un computador que siempre está encendido (debería) y tiene instalada la programación backend del Proyecto.
- Tiene una IP (identificador alfa numérico que permite reconocer dicho computador en la red).

Dominios

- Frase alfanumérica que permite acceder a un computador con una determinada IP.
- Existen dominios de todos los gustos: .pro .cl .vivachile, etc.



Explorador hace la consulta para el dominio sudominio.com

Dónde está sudominio.com?



Se encuentra en la IP 192.168.0.100



Servidor DNS

Sesión 1.2

Nivelación Python y creación de entorno virtual

Clases, objetos, pip y virtualenv



¿Qué es un objeto?

- Es un “molde” o “esquema” con el que podemos crear entidades llamadas objetos.

Ejemplo:

```
class Persona( ):
    nombre = “ ”
    categoria = “ ”
    pais = “ ”
```

Objeto

```
yo= Persona( )
yo.nombre = “Luke Skywalker”
yo.categoria = “Jedi”
yo.pais = “Tatooine ”
```

Nombre clase



```
class Persona( ):
```

```
    nombre = “ ”
    categoria = “ ”
    pais = “ ”
```

Atributos



Método



```
    def hablar (self):
        print (“Que la fuerza esta contigo”)
```



Métodos especiales

- Constructor `__init__`

```
class Persona( ):
    def __init__(self, nom, pais):
        self.nombre=nom
        self.pais=pais
```



Atributos obligatorios para
crear la instancia de
objeto de esta clase

- To String `__str__`

```
Class Persona( ):
    def __str__(self):
        return self.nombre
```



Lo que muestra si intento
“*printear*” el objeto



Ejemplo

Class Persona():

```
def __init__(self, nom, pais):  
    self.nombre=nom  
    self.pais=pais
```

vader = Persona ("Darth Vader", "Tatooine")

vader = Persona (nom="Darth Vader", pais="Tatooine")

Print (vader)

> "Darth Vader"



Herencia

```
Class Persona( ):
```

```
    def __init__(self, nom, pais):
```

```
        self.nombre=nom
```

```
        self.pais=pais
```

```
Class SuperHeroe( ):
```

```
    def __init__(self, poder):
```

```
        self.poder=poder
```



Por ahora son
clases no
relacionadas



Herencia

```
Class Persona( ):
```

```
    def __init__(self, nom, pais):  
        self.nombre=nom  
        self.pais=pais
```

```
Class SuperHeroe(Persona,object ):
```

```
    def __init__(self, poder,nom,pais):  
        self.poder=poder  
        self.nombre=nom  
        self.pais=pais  
        super(SuperHeroe,self).__init__(nom,pais)
```

Ahora, al crear el objeto SuperHeroe, se hereda el método de creación del objeto Persona



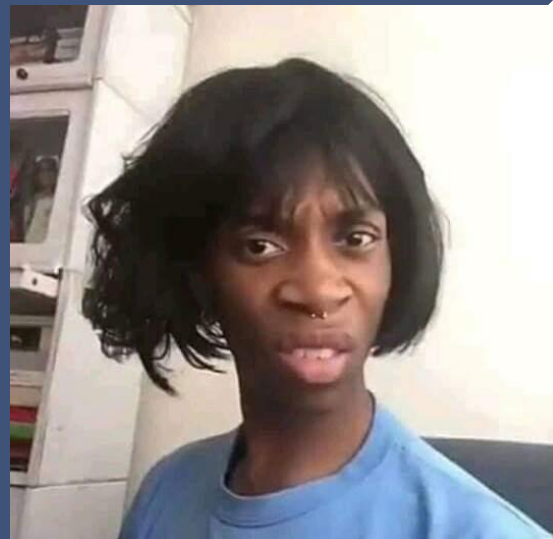


Actividad Evaluada 0 (optativa)

Cree un programa en Python 3 que posea al menos una herencia de clases con los métodos `__str__` y `__init__`

“

Entornos Virtuales



“

Antes de eso, PIP





Comencemos

python --version



Nos dice qué versión de Python tenemos instalada

Pip es un instalador de paquetes desarrollados por terceros que se encuentren en Python Package Index (PyPI)

pip3 install nombre-paquete

pip3 uninstall nombre-paquete

“

Entornos Virtuales

Ahora sí





Entorno virtual (paquete virtualenv)

Un entorno virtual es un directorio que encapsula paquetes instalados para una determinada versión de Python.

Son muy útiles para resolver problemas cuando necesitamos tener diferentes versiones del mismo framework en nuestro computador (y son necesarios para montar nuestro proyecto en el servidor)

```
py -m venv nombre_env
```



Entorno virtual (paquete virtualenv)

Para utilizar nuestro entorno virtual debemos activarlo con el comando:

Unix

```
source nombre_env/bin/actívale
```

Windows

```
nombre_env\Scripts\activate.bat
```




Entorno virtual (paquete virtualenv)

Para generar un archivo que nos diga todos los paquetes que están instalados en nuestro entorno, utilizamos el siguiente comando:

pip3 freeze > requirements.txt

requirements.txt

```
Django==1.11.5
```

```
pytz==2017.2
```



Actividad Evaluada 1

1. Instale virtualenv en su equipo y cree un entorno virtual.
2. Luego actívelo e instale django.
3. Genere el archivo requirements.txt
4. Por último, súbalo a Moodle.



Gracias por venir!

¿Preguntas?

Estamos a una clase de tener nuestro hola mundo 

Jorge.diazma@sansano.usm.cl