



Hito 02



{desafío}
latam_

Recursos antes de iniciar

- A continuación, se comparten un conjunto de recursos audiovisuales que permitirán abordar los Hitos de nuestro proyecto.
- Antes de ejecutar la solución de los Hitos, es importante que cada estudiante vea estos recursos en video.

Estructura de proyectos Django

La estructura básica de un proyecto en Django sigue una organización modular y sigue el patrón de diseño Model-View-Template (MVT).

- **Directorio principal:** El directorio principal del proyecto generalmente tiene el mismo nombre que tu proyecto. Este es el directorio que contiene el archivo `manage.py`.

Estructura de proyectos Django

- **manage.py**: Archivo de utilidad de línea de comandos para realizar diversas tareas de gestión del proyecto.
- **__init__.py**: Un archivo vacío que indica que el directorio es un paquete de Python.
- **settings.py**: Configuración del proyecto, donde se definen variables como bases de datos, aplicaciones instaladas, configuración de middleware, etc.

Estructura de proyectos Django

- **urls.py**: Configuración de las URL del proyecto, donde se definen las rutas y cómo se deben manejar.
- **asgi.py**: Configuración para el servidor de interfaz de puerta de enlace asincrónica (ASGI) utilizado para manejar conexiones asincrónicas.

Estructura de proyectos Django

Las aplicaciones

- Las aplicaciones son módulos reutilizables que contienen modelos, vistas y plantillas específicos de una funcionalidad del proyecto. Puedes tener múltiples aplicaciones en un proyecto.

```
mi_proyecto/  
├── mi_aplicacion/  
│   ├── __init__.py  
│   ├── admin.py  
│   ├── apps.py  
│   ├── migrations/  
│   ├── models.py  
│   ├── tests.py  
│   └── views.py
```

Estructura de proyectos Django

Las aplicaciones

- **__init__.py**: Indica que el directorio es un paquete de Python.
- **admin.py**: Configuración de la interfaz de administración.
- **apps.py**: Configuración específica de la aplicación.
- **migrations/**: Directorio que contiene archivos de migración para el modelo de la aplicación.
- **models.py**: Definición de modelos de datos.
- **tests.py**: Archivos para escribir pruebas de la aplicación.
- **views.py**: Definición de las vistas (controladores) de la aplicación.

Estructura de proyectos Django

Las aplicaciones

- **Directorio de Plantillas:** Contiene archivos HTML que se utilizan para renderizar las vistas.

```
mi_proyecto/  
└── templates/  
    ├── mi_aplicacion/  
        └── template.html
```


Estructura de proyectos Django

Las aplicaciones

- **Archivos de Configuración adicionales:** Puedes tener archivos adicionales para configurar cosas como la base de datos, archivos estáticos, archivos de registro, etc.

```
mi_proyecto/  
├── requirements.txt  
└──  
otros_archivos_de_configuracion/
```

Revisión de ejercicio

Revisa en plataforma el video “**Muestra de estructura de proyecto Django**”

Repliquemos juntos este ejercicio.

Recurso
audiovisual

Estructura de proyectos Django

Las aplicaciones

- **Directorio de Estáticos:** Contiene archivos estáticos como CSS, JavaScript, imágenes, etc.

```
mi_proyecto/  
├── static/  
│   ├── mi_aplicacion/  
│   │   ├── css/  
│   │   │   └── estilo.css
```

Revisión de ejercicio

Revisa en plataforma el video **“Ejemplo de contenido estático en Django”**

Repliquemos juntos este ejercicio.

**Recurso
audiovisual**



Revisión de ejercicio

Revisa en plataforma el video **“Cómo agregar static files a una plantilla”**

Repliquemos juntos este ejercicio.

**Recurso
audiovisual**



Proyecto: Creación de sitio web para una PYME de venta de pasteles y postres



Para revisar en detalle, dirígete al documento
"Proyecto" del Módulo 6.

Aprendizajes Esperados

- **Despliegue de páginas con contenido dinámico:**
 - Implementa un proyecto Django para servir contenido estático dando solución a los requerimientos.
 - Utiliza templates para la renderización de contenido dinámico en un proyecto Django para dar solución a un requerimiento.
 - Utiliza herencia de plantillas en un proyecto Django para dar solución a un requerimiento.
 - Utiliza instrucciones de control en plantillas de un proyecto Django para dar solución a un requerimiento.

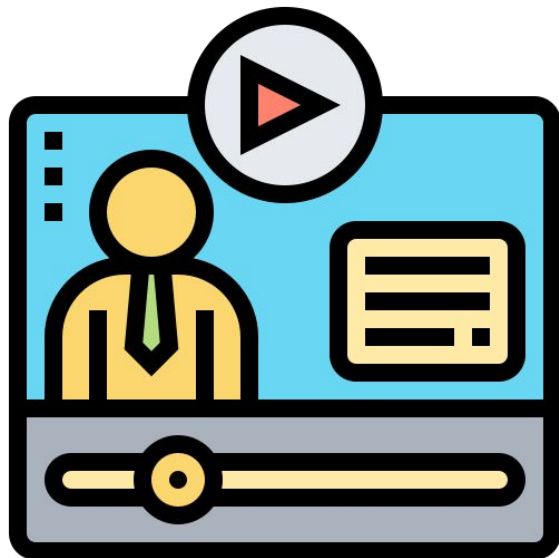
Requerimientos Hito 02



¡Revisemos los
requerimientos del
Hito 02 “Creación
de un sitio web
responsive con
Bootstrap”!

Actividad 01

Iniciación de una App en Django



Iniciaremos una nueva app de Django llamada **web** dentro del proyecto **onlyflans** y la agregaremos a la lista de aplicaciones instaladas, dentro de ella crearemos una carpeta **templates** que contenga un archivo html llamado **index.html** que contenga las estructuras `<html>`, `<header>` y `<body>`. Dentro del `<body>` de este archivo escribiremos el texto “índice”. Crearemos 2 copias del archivo **index.html**, llamadas **about.html** y **welcome.html** y reemplazaremos el texto de sus estructuras `<body>` por “acerca” y “bienvenido cliente” respectivamente. Para guiarte en el proceso, te recomendamos revisar el videotutorial del Hito 01 “Muestra de estructura de proyecto django” y la lectura referida Hito 2 “Creando proyecto y app de django (español)”.

Actividad 02

Habilitación de las urls con una plantilla básica

Habilitaremos las 3 urls distintas con un plantilla básica que muestre solo texto, esto implica:

- Mostrar el texto “índice” en la ruta /
- Mostrar el texto “acerca” en la ruta **/acerca**
- Mostrar el texto “bienvenido cliente” en la ruta **/bienvenido**

Una vez realizado esto, ejecutaremos el sitio web con python manage.py runserver, y crearemos un “pantallazo” de cada una de esas rutas y las guardaremos en formato jpg o png.

Para guiarte en el proceso, revisa la **Lectura Referida del Hito 2**, apartado “**Creando tu página de inicio (español)**”, además del resto de los enlaces disponibles.



Actividad 03

Creación de la plantilla base



Crearemos una plantilla base llamada ***base.html*** que contenga los elementos comunes a todas las rutas necesarias para el sitio web, esto puede ser, como en el caso anterior, simplemente un texto que identifique cada elemento, por ejemplo:

- Mostrar el texto “header” en el lugar donde iría el header
- Mostrar el texto “navbar” en el lugar donde iría la barra de navegación
- Mostrar el texto “footer” en el lugar donde iría el footer.

Para el desarrollo de esta actividad te recomendamos revisar el **videotutorial “Ejemplo de contenido estático en Django”**, así como la **Lectura Referida Hito 2**.

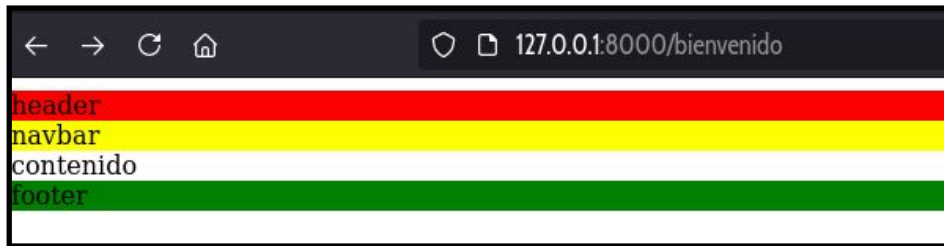
Actividad 03

Creación de la plantilla base

Para esta actividad debes valerte de estructuras como `<div></div>` para separar los distintos elementos, y se recomienda agregarle un color de fondo a cada elemento para guiar de mejor manera la estructura general de lo que estás maquetando.

Una vez maquetado tu archivo base, extiéndelo dentro de cada una de las rutas a disponibilizar, respetando los textos agregados a cada una de las rutas en el texto anterior y crea un “pantallazo” de cada una de ellas y guardalas en formato jpg o png.

(Debiera desplegar algo parecido a este ejemplo)



Actividad 04

Creación de las vistas y plantillas personalizadas

- “Instalaremos” bootstrap a través de incluir la [“plantilla inicial” de bootstrap](#) en el plantilla base, complementando su estructura <body> con lo que ya existe en el archivo. Luego de agregado esto, cambia el contenido de su estructura <title> por “Bienvenido a onlyflans”.
- Al contenido de la web, en los <div> de **header**, **navbar**, **contenido** y **footer** agregaremos la clase css **container**, este cambio debe aplicarse tanto en la plantilla base como en aquellas plantillas que lo extienden.
- Reemplazaremos los contenidos de header, navbar y footer por componentes de bootstrap que nos permitan darle más “estilo” a la web, se recomienda crear una plantilla para cada componente (ejemplo: archivos **header.html**, **navbar.html**, **footer.html** e incluirlos dentro de nuestra base utilizando la [etiqueta include](#).
- A cada elemento del navbar debemos agregarle un enlace que permita dirigirnos a cada una de las url antes descritas al presionar sobre su respectivo nombre en el navbar.

Actividad 04

Creación de las vistas y plantillas personalizadas

Una vez terminado los pasos anteriores, crea un “pantallazo” de cada una de esas rutas y guardalas en formato jpg o png.

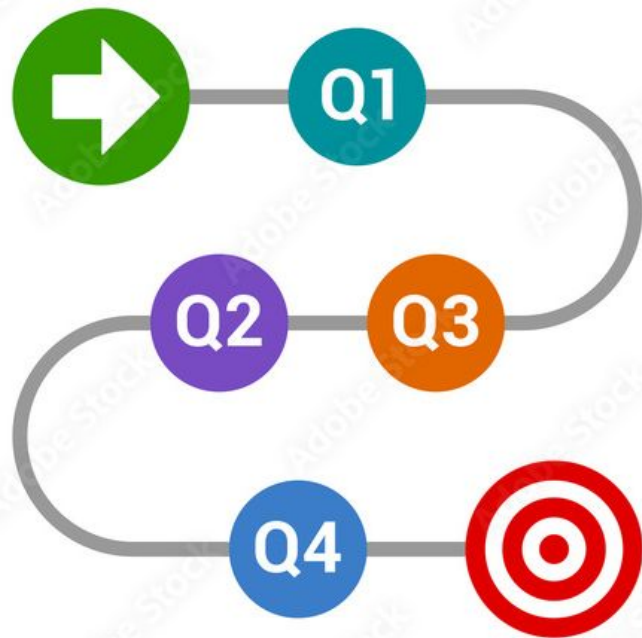
Para más detalles de estos requerimientos, revisa con cuidado la instrucciones contenidas en el Hito 02, así como la **Lectura Referida Hito 2**, particularmente el apartado “**Imprimir variables, condicionales, ciclos. (inglés)**”. También puedes revisar el **videotutorial “Cómo agregar static files a una plantilla”**.



Actividades completadas

Ya llegado a este punto dentro del proyecto, deberías haber logrado lo siguiente:

- Iniciar una App en Django.
- Habilitar las urls con una plantilla básica.
- Crear la plantilla base.
- Crear las vistas y plantillas personalizadas.





*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam