



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Análisis y Desarrollo de Software.
- **Código del Programa de Formación:** 228118
- **Nombre del Proyecto:** Desarrollo de Software Empresarial Innovador Alineado con las Políticas y Estrategias Gubernamentales Locales y Nacionales.
- **Fase del Proyecto:** Analizar (Fase I)
- **Actividad de Proyecto:** Estructurar procesos logicos para la construcción de algoritmos y bases de datos
- **Competencia:** Construcción del software.
- **Resultados de Aprendizaje Alcanzar:** Planear actividades de construcción del software de acuerdo con el diseño establecido..
- **Duración de la Guía de Aprendizaje:** 120 horas

2. PRESENTACIÓN

Trabajaremos la tecnología de Python a través del framework de Django, que genera un conjunto de objetos que se enlazan a la base de datos de MYSQL, permitiendo a los desarrolladores integrar un lenguaje orientado a objetos con la base de datos en el desarrollo de una aplicación Web.

En esta fase veremos cómo los desarrolladores de software pueden trabajar en un nivel más alto de abstracción cuando tratan con datos, y puede crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales, ya que pueden funcionar en términos de un



modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones. Además, pueden integrar a los proyectos de software Git para el manejo de repositorios y el control de versiones, incluidos proyectos comerciales y de código abierto, donde se almacenan los cambios del software.

Recuerda leer de manera reflexiva la guía de aprendizaje en su totalidad, para que de manera ordenada y manejando los tiempos sugeridos, cumplas con las evidencias e inicies el desarrollo de cada una de las acciones propuestas.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial

Descripción de la actividad: 3.1.1 Actividad: Identificar la importancia del desarrollo web en el mundo actual

Los aprendices ingresan al sitio <https://www.youtube.com/watch?app=desktop&v=YYjLcSft4XM> para visualizar el video: "Importancia del Desarrollo Web en el mundo actual - Introducción al Desarrollo Web".

Luego, responden las siguientes preguntas:

¿Por qué consideras que es importante el desarrollo web en la actualidad?

¿Cuál es el potencial que puede tener el desarrollo de aplicaciones web y en qué campos tiene aplicabilidad?

Al finalizar, se socializan las respuestas y el instructor realiza una retroalimentación final.

Ambiente requerido: Aula de formación con acceso a internet.

Estrategias o técnicas didácticas activas:

- Aprendizaje basado en videos y análisis reflexivo.
- Discusión grupal y socialización de respuestas.



- Preguntas orientadoras para fomentar el pensamiento crítico.
- Retroalimentación del instructor para reforzar conceptos clave.

Materiales de formación: Video explicativo sobre la importancia del desarrollo web, guía con preguntas para reflexión y discusión.

Material de apoyo: Computadores portátiles o de escritorio con acceso a internet, pantalla para proyectar el video.

Duración de la actividad: 3 horas

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje

Descripción de la actividad: 3.2.1 Actividad: Reconocer las ventajas del desarrollo web con Python

Los aprendices ingresan al sitio <https://keepcoding.io/blog/desarrollo-web-con-python/> para identificar las ventajas del desarrollo web con Python, entre ellas:

- ✓ Barreras bajas de entrada.
- ✓ Buena visualización.
- ✓ Desarrollo web fácil de leer.
- ✓ Enfoque de programación menos limitado.
- ✓ Desarrollo web más rápido.
- ✓ Bibliotecas estándar.
- ✓ Frameworks que simplifican el proceso de desarrollo.

Luego, los aprendices elaboran una infografía con los conceptos trabajados.

Al finalizar, se socializan las infografías y el instructor realiza una retroalimentación final.

Ambiente requerido: Aula de formación con acceso a internet.

Estrategias o técnicas didácticas activas:



- Investigación autónoma y análisis de contenido digital.
- Creación de material visual (infografía) para reforzar el aprendizaje.
- Socialización y discusión grupal de los conceptos trabajados.
- Retroalimentación del instructor para consolidar el conocimiento.

Materiales de formación: Artículo sobre desarrollo web con Python, guía de estructura para la infografía.

Material de apoyo: Computadores portátiles o de escritorio con acceso a internet, herramientas digitales para la creación de infografías (Canva, PowerPoint, Microsoft Word), pantalla de TV o proyector para la presentación de las infografías.

Duración de la actividad: 3 horas

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización)

Descripción de la actividad: 3.3.1 Actividad: Reconocer los conceptos generales de Programación Orientada a Objetos (POO) en Python

El instructor realiza una exposición magistral sobre los conceptos fundamentales de Programación Orientada a Objetos (POO) en Python, abordando los siguientes temas:

- ✓ Definición y creación de clases, atributos, métodos y objetos.
- ✓ Métodos (constructores, instancia, clase, estáticos y mágicos).
- ✓ Uso y creación de decoradores.
- ✓ Interfaces y clases abstractas (interfaces formales y virtuales, clases virtuales, colecciones).
- ✓ Encapsulamiento.
- ✓ Herencia (clases y métodos abstractos).
- ✓ Polimorfismo.
- ✓ Manejo de excepciones.



Posteriormente, el instructor presenta casos de estudio relacionados con cada concepto para profundizar en la programación orientada a objetos en Python, mostrando diversas soluciones.

Al finalizar, se asigna un taller a los aprendices, el cual será socializado para reforzar los conocimientos adquiridos.

Ambiente requerido: Aula con acceso a internet, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub).

Estrategias o técnicas didácticas activas:

- Exposición magistral con ejemplos prácticos.
- Análisis de casos de estudio para aplicar los conceptos en situaciones reales.
- Desarrollo de un taller práctico con ejercicios aplicados.
- Socialización y discusión de los resultados del taller.
- Retroalimentación del instructor para aclarar dudas y reforzar conocimientos.

Materiales de formación: Documentos de apoyo con teoría y ejemplos sobre POO en Python, casos de estudio detallados, taller práctico con ejercicios guiados.

Material de apoyo: Computadores con acceso a internet, Editor de texto o entorno de desarrollo (Visual Studio Code), Python instalado, Bootstrap para desarrollo web, MySQL para bases de datos en caso de integración con proyectos prácticos.

Duración de la actividad: 34 horas

Descripción de la actividad: **3.3.2 Actividad:** Reconocer el entorno de desarrollo web y del framework Django

El instructor realiza una demostración del entorno de desarrollo en Python, utilizando Django con el patrón MVT (Modelo-Vista-Template), abordando los siguientes temas:



- Definición de la estructura del proyecto en Django.
- Creación de modelos y migraciones.
- Uso de bases de datos SQL y/o NoSQL y definición de relaciones.
- Manejo de Shell y QuerySets.
- Implementación de plantillas, URLs y vistas.
- Uso de archivos estáticos, CSS, HTML5 y Bootstrap.
- Desarrollo de un CRUD maestro-detalle para listar, crear, actualizar y eliminar registros.
- Uso de funciones y clases en Django.
- Implementación de seguridad con manejo de login y restricciones.
- Creación de menú de navegación.
- Generación de reportes y dashboards.
- Aplicación de pruebas unitarias para asegurar el correcto funcionamiento del sistema.

Ambiente requerido: Aula con acceso a internet, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub).

Estrategias o técnicas didácticas activas:

- Demostración práctica del instructor.
- Desarrollo guiado de un proyecto utilizando Django.
- Implementación progresiva de funcionalidades en el proyecto formativo.
- Aplicación de pruebas unitarias para validación del sistema.
- Socialización del desarrollo del CRUD maestro-detalle con la arquitectura MVT de Django.

Materiales de formación: Documentos de apoyo sobre Django y el patrón MVT, Guía de implementación del CRUD maestro-detalle, recursos sobre seguridad y pruebas unitarias en Django.

Material de apoyo: Computadores con acceso a internet, editor de texto o entorno de desarrollo (Visual Studio Code), Python instalado con Django configurado, Bootstrap para diseño responsivo, MySQL o base de datos compatible con Django ORM.



Duración de la actividad: 34 horas

Descripción de la actividad: **3.3.3 Actividad:** Aplicar los conocimientos del diseño y estructura de un sitio Web, utilizando Django en la implementación de un CRUD con manejo de repositorios

El instructor realiza una demostración práctica en Django para desarrollar un CRUD con autenticación, navegación y acceso a datos, utilizando el patrón MVT (Modelo-Vista-Template).

Posteriormente, cada aprendiz desarrollará un ejercicio individual donde integrará:

- Modelo de datos con Django y bases de datos SQL o NoSQL.
- Implementación del CRUD con autenticación y control de acceso.
- Integración con Visual Studio Code.
- Manejo de repositorios en GitHub, incluyendo:
 - Creación y administración de repositorios.
 - Restablecimiento y configuración.
 - Gestión de ramas.
 - Registro de actividad y administración de datos.

Al finalizar, los aprendices presentarán sus desarrollos y el instructor brindará una retroalimentación sobre la implementación del CRUD y el manejo de repositorios.

Ambiente requerido: Aula con acceso a internet, espacio adecuado para la socialización de avances del proyecto, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub).

Estrategias o técnicas didácticas activas:

- Demostración guiada del instructor.
- Desarrollo individual de un ejercicio práctico.
- Aplicación del control de versiones con GitHub.
- Integración del CRUD con autenticación y seguridad en Django.



- Retroalimentación y socialización de los proyectos.

Materiales de formación: Documentos de apoyo sobre Django y el patrón MVT, guía para la implementación de CRUD en Django, manual de uso de GitHub y control de versiones.

Material de apoyo: Computadores con acceso a internet, editor de texto o entorno de desarrollo (Visual Studio Code), Python con Django instalado, Bootstrap para diseño responsivo, MySQL o base de datos compatible con Django ORM.

Duración de la actividad: 30 horas

Descripción de la actividad: 3.3.4 Actividad: Identificar los conceptos de API-REST en una aplicación desarrollada en Python Django

El instructor realiza una exposición magistral sobre los conceptos fundamentales de API-REST, abordando los siguientes temas:

- Instalación y configuración del entorno.
- Construcción de una API-REST en Django:
- Creación de un proyecto y configuración de rutas.
- Mapeo y registro de endpoints.
- Uso del Django REST Framework.
- Configuración de CORS Headers.
- Integración con la base de datos.
- Implementación de vistas, modelos y migraciones.
- Pruebas de la API con herramientas como Postman o Swagger.

Después de la exposición, el instructor refuerza los conceptos con una retroalimentación y presenta un caso de estudio demostrativo sobre la construcción y uso de una API-REST.



Posteriormente, los aprendices aplicarán lo aprendido desplegando el proyecto formativo desarrollado en Python Django con API-REST.

Ambiente requerido: Aula con acceso a internet, espacio adecuado para la socialización de avances del proyecto, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub), espacio adecuado para el despliegue y pruebas de la API-REST.

Estrategias o técnicas didácticas activas:

- Exposición magistral del instructor.
- Análisis de un caso de estudio para comprender la implementación de API-REST.
- Desarrollo práctico con la creación y despliegue de una API-REST.
- Retroalimentación y socialización de los aprendizajes obtenidos.

Materiales de formación: Documentación de Django REST Framework, guía paso a paso para la construcción de una API-REST, ejemplo práctico de una API-REST en Django.

Material de apoyo: Computadores con acceso a internet, editor de texto o IDE (Visual Studio Code, PyCharm), Python con Django instalado, Django REST Framework, plataforma para pruebas de API (como Postman, rest-client).

Duración de la actividad: 2 horas

Descripción de la actividad: 3.3.5 Actividad: Aplicar los conocimientos relacionados con la implantación de una aplicación desarrollada en Python django

El instructor realiza una exposición magistral sobre los conceptos clave para el despliegue en la nube, abordando las siguientes áreas:

Plataformas de nube populares como AWS, Heroku, Azure, entre otras.



Elementos generales del despliegue de una aplicación desarrollada en Python Django, incluyendo:

- Repositorio Git para el control de versiones.
- Uso de contenedores (Docker, por ejemplo).
- Configuración de variables de entorno.
- Archivos de configuración necesarios para el despliegue.
- Proceso de despliegue y migración de la base de datos.
- Plataformas en la nube para el hospedaje de la aplicación.

Después de la exposición, el instructor brinda retroalimentación sobre los conceptos presentados y realiza un caso de estudio demostrativo de implantación en la nube.

Finalmente, los aprendices aplican lo aprendido para realizar el despliegue de una aplicación en la plataforma seleccionada en la nube, con el uso de los conceptos expuestos.

Ambiente requerido: Aula con acceso a internet, *computadores con herramientas de desarrollo y acceso a plataformas de nube configuradas*, espacio adecuado para la socialización de avances del proyecto, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub)*espacio adecuado para la demostración del despliegue y socialización de resultados.*

Estrategias o técnicas didácticas activas:

- Exposición magistral con ejemplos prácticos sobre despliegue en la nube.
- Análisis de caso de estudio para comprender la implantación de aplicaciones en plataformas de nube.
- Desarrollo práctico para que los estudiantes apliquen los conceptos mediante el despliegue en la nube.
- Retroalimentación del instructor para resolver dudas y mejorar el proceso de despliegue.



Materiales de formación: Documentación sobre despliegue de aplicaciones en la nube. guías para la configuración de repositorios Git, contenedores, y variables de entorno, ejemplo práctico de despliegue en plataformas como AWS o Heroku.

Material de apoyo: Computadores con acceso a internet, editor de texto o IDE (Visual Studio Code, PyCharm), Plataformas en la nube (AWS, Heroku, Azure) para realizar el despliegue, repositorio Git configurado para el proyecto.

Duración de la actividad: 2 horas

3.4 Actividades de transferencia del conocimiento

Descripción de la actividad: 3.4.1 Actividad: Socializar matriz CRUD en el lenguaje de programación seleccionado para verificar la funcionalidad del sistema de acuerdo con el diseño entregado de la fase I de análisis.

Los aprendices presentarán el proyecto en Django, incluyendo los formularios construidos para los procesos del proyecto formativo. Utilizando Django, el patrón MVT y la base de datos, se integrará el desarrollo de los CRUD correspondientes que permiten la creación, consulta, modificación y eliminación en la base de datos. Además, se socializará el despliegue de la aplicación según la plataforma tecnológica seleccionada del proyecto desarrollado con Python Django. La actividad se llevará a cabo en equipos designados del proyecto formativo, con asignación individual de desarrollo, pero integrada de manera colaborativa.

Ambiente requerido: : Aula con acceso a internet, *computadores con herramientas de desarrollo y acceso a plataformas de nube configuradas*, espacio adecuado para la socialización de avances del proyecto, entorno de desarrollo configurado (Visual Studio Code, Python, Django), Plataforma de control de versiones (GitHub) *espacio adecuado para la demostración del despliegue y socialización de resultados.*



Estrategias o técnicas didácticas activas:

- Desarrollo práctico de CRUD utilizando Django y el patrón MVT.
- Socialización en equipos para fomentar el aprendizaje colaborativo.
- Presentación y discusión de casos prácticos relacionados con el proyecto formativo.

Materiales de formación: Documentos de apoyo que incluyan guías y tutoriales sobre Django y el patrón MVT, diagramas de base de datos.

Material de apoyo: Recursos en línea sobre Django y desarrollo web, foros y comunidades de desarrolladores para resolver dudas técnicas, documentación oficial de Django.

Duración de la actividad: 20 horas.

4. ACTIVIDADES DE EVALUACIÓN

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Analizar	Estructurar procesos logicos para la construcción de algoritmos y bases de datos	Reconocer los conceptos generales de Programación Orientada a Objetos (POO) en Python	Taller y socialización de conceptos de programación orientada a objetos en Python	Interpreta el informe de diseño para definir el plan de trabajo en la construcción del software.	Técnica observación Instrumento: rúbrica y/o escala de valoración



		Reconocer el entorno de desarrollo web y del framework Django	<p>CRUD de proyecto formativo utilizando la arquitectura MVT de Django en el desarrollo de un CRUD.</p>	<p>Selecciona y configura herramientas de desarrollo de acuerdo con las condiciones del software a construir.</p> <p>Selecciona y configura herramientas de versionamiento para el control de cambios en el código.</p> <p>Define estándares de codificación de acuerdo con las reglas de la plataforma de desarrollo seleccionada.</p>	<p>Técnica observación</p> <p>Instrumento: rúbrica y/o escala de valoración</p>
		Aplicar los conocimientos del diseño y estructura de un sitio Web, utilizando Django en la implementación de un CRUD con manejo de repositorios	<p>CRUD con el patrón MVT del proyecto formativo, que integra administración de la seguridad y manejo de repositorios.</p>		<p>Técnica observación</p> <p>Instrumento: rúbrica y/o escala de valoración</p>



		Identificar los conceptos de API-REST en una aplicación desarrollada en Python Django	REST-API de una aplicación Python django		Técnica observación Instrumento: rúbrica y/o escala de valoración
		Aplicar conocimientos relacionados con la implantación de una aplicación desarrollada en Python django	Implantación de proyecto desarrollado con python django en la nube		Técnica observación Instrumento: rúbrica y/o escala de valoración
		Socializar matriz CRUD en el lenguaje de programación seleccionado para verificar la funcionalidad del sistema de acuerdo con el	Socialización del proyecto formativo con matriz CRUD en Python django y el despliegue, utilizando el patrón MVT,		Técnica valoración de productos Instrumento rúbrica y/o Lista de chequeo



		diseño entregado de la fase I de análisis.	según los requerimientos de la matriz de historias de usuario.		
--	--	---	--	--	--

5. GLOSARIO DE TÉRMINOS

Algoritmo: Un algoritmo es un conjunto de instrucciones que son tan precisas y tan bien hechas que basta con que una persona sepa leer para ejecutar correctamente una tarea que nunca antes había ejecutado.

BASE DE DATOS: Es un almacenamiento colectivo de las bibliotecas de datos que son requeridas y organizaciones para cubrir sus requisitos de procesos y recuperación de información.

Bootstrap: es un framework web o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.



Hojas de estilo en cascada o CSS (siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

Página Web: Una página web es parte de un sitio web y es un único archivo con un nombre de archivo asignado.

Sitio Web: un sitio web es un conjunto de archivos llamados páginas web.

Python: es un lenguaje de programación de alto nivel, interpretado y de propósito general, conocido por su sintaxis clara y legibilidad. Fue creado en la década de 1990 por Guido van Rossum y ha ganado popularidad por su versatilidad y amplia comunidad de desarrolladores.

Django: es un framework de desarrollo web de alto nivel para Python, diseñado para facilitar la creación de aplicaciones web robustas y escalables. Sigue el patrón de diseño Modelo-Vista-Controlador (MVC) y promueve el desarrollo rápido y limpio.

MVT: La estructura MVT (Modelo-Vista-Template) de Django es una variación del patrón MVC. En este patrón:

- **Modelo:** Representa la estructura de datos y la lógica de negocio de la aplicación.
- **Vista:** Es la interfaz de usuario que presenta los datos al usuario.
- **Template:** Son los archivos que definen la presentación de la vista, generalmente en HTML.

API REST: (Interfaz de Programación de Aplicaciones basada en Transferencia de Estado Representacional) es un estilo arquitectónico para diseñar servicios web. Se basa en principios como la comunicación sin estado, la utilización de métodos HTTP estándar y la representación de recursos en formatos como JSON o XML. Las API RESTful permiten la interacción entre sistemas de manera eficiente y escalable.

CÓDIGO FUENTE: Programa en su forma original, tal y como fue escrito por el programador, el código fuente no es ejecutable directamente por el computador, debe convertirse en lenguaje de máquina mediante compiladores, ensambladores o interpretes.



COMPILADOR: Programa de computadora que produce un programa en lenguaje de máquina, de un programa fuente que generalmente está escrito por el programador en un lenguaje de alto nivel.

INTÉRPRETE: Dispositivo o programa que recibe una por una las sentencias de un programa fuente, la analiza y la convierte en lenguaje de máquina si no hay errores en ella. También se puede producir el listado de las instrucciones del programa.

VARIABLE: En programación es una estructura que contiene datos y recibe un nombre único dado por el programador, mantiene los datos asignados a ella hasta que un nuevo valor se le asigne o hasta que el programa termine.

CONSTANTE: Valor o conjunto de caracteres que permanecen invariables durante la ejecución del programa.

PROGRAMA: Es una colección de instrucciones que indican a la computadora que debe hacer. Un programa se denomina software, por lo tanto, programa, software e instrucción son sinónimos.

FUNCIÓN: En programación, una rutina que hace una tarea particular. Cuando el programa pasa el control a una función, ésta realiza la tarea y devuelve el control a la instrucción siguiente a la que llamó.

INTERFAZ: Una conexión e interacción entre hardware, software y usuario, es decir como la plataforma o medio de comunicación entre usuario o programa.

PROGRAMADOR: Un individuo que diseña la lógica y escribe las líneas de código de un programa de computadora.

PROGRAMADOR DE APLICACIONES: Individuo que escribe programas de aplicación en una organización usuaria. La mayoría de los programadores son programadores de aplicación.

GIT: Git es un sistema maduro de control de revisiones de código abierto que se mantiene activo y lo utilizan miles de desarrolladores en todo el mundo.



6. REFERENTES BIBLIOGRÁFICOS

Parzibyte. (2017, 19 de noviembre). Instalar y configurar Python 3 en Windows 10. Parzibyte. Recuperado de <https://parzibyte.me/blog/2017/11/19/instalar-configurar-python-3-windows-10/>

J2Logo. (s.f.). Virtualenv, pip y librerías en Python. J2Logo. Recuperado de <https://j2logo.com/virtualenv-pip-librerias-python/>

Uniwebsidad. (s.f.). Python. Uniwebsidad. Recuperado de <https://uniwebsidad.com/libros/python>

Python Software Foundation. (s.f.). Descargas. Python.org. Recuperado de <https://www.python.org/downloads/>

Django Girls. (s.f.). Instalación de Python. Django Girls. Recuperado de https://tutorial.djangogirls.org/es/python_installation/

PyBaq. (s.f.). Una guía completa de Django para principiantes - Parte 1. PyBaq. Recuperado de <https://pybaq.co/blog/una-guia-completa-de-django-para-principiantes-parte-1/>

Python.org.ar. (s.f.). Tutorial de Python. Python.org.ar. Recuperado de <https://tutorial.python.org.ar/en/latest/tutorial/3/real-index.html>

McLibre. (s.f.). Python. McLibre. Recuperado de <https://www.mclibre.org/consultar/python/>

J2Logo. (s.f.). args y kwargs en Python. J2Logo. Recuperado de <https://j2logo.com/args-y-kwargs-en-python/>

Full Stack Python. (s.f.). Django URLs: reverse() y lazy() ejemplos. Full Stack Python. Recuperado de <https://www.fullstackpython.com/django-urls-reverse-lazy-examples.html>

APSL. (2009, 31 de agosto). Decoradores en Python. APSL. Recuperado de <https://www.apsl.net/blog/2009/08/31/decoradores-en-python/>

Axiacore. (s.f.). Paginación en Django estilo Digg. Axiacore. Recuperado de <https://axiacore.com/blog/paginacion-en-django-estilo-digg-409/>



Libro:

Martínez, A. (2020). Django.

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Edicccson Quiroz	Instructor	Software	1/03/2024

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Jairo Arboleda	Instructor	Software	31/01/2025	Ajustes contenido y formato
Autor (es)	Edicccson Quiroz	Instructor	Software	31/01/2025	Ajustes contenido y formato