

Curso:

Desarrollo Web en

Python con Django

Plan de estudios



Nuestro objetivo

Brindar herramientas que faciliten la inserción laboral en el sector Informática (IT), y en particular, fomentar la participación de las mujeres, con el fin de mejorar su empleabilidad.

Desarrollo Web en Python con Django

Ideal para quienes poseen conocimientos de programación Full Stack con Python, y quieran profundizar sus conocimientos en el desarrollo integral de un sitio. Trabajas en el Web Framework más potente y popular del mercado Python. Aprenderás a desarrollar aplicaciones web escalables y seguras de forma profesional. Al finalizar el curso, y a través del trabajo en equipo, habrás desarrollado el backend y el frontend para un sitio web completo y escalable que se conectará a una base de datos, tendrá su propio sitio de administración, respetará los aspectos básicos de seguridad.

Modalidad

Clases online de una hora y media en vivo 2 veces por semana

Duración

20 semanas

Herramientas y tecnologías que aprenderemos



Visual Studio Code



HTML



CSS



Bootstrap



GIT



GitHub



JavaScript



PostgreSQL



Python



Django

Contenido clase a clase

0. Presentación del curso

- Sobre Codo a Codo.
- Objetivo del curso y lineamientos de cursada.
- Conceptos necesarios identificando cuales se revisarán en el curso.
- Herramientas a utilizar.
- Instalación del software.
- Presentación del proyecto integrador grupal y modalidad

1. Arquitectura cliente servidor

- Repaso conceptos de la web
- Concepto Cliente/Servidor
- Front-end y Back-end
- Protocolo http/https, request y response
- Especificación WSGI con Django

2. Git con Github

- Comandos básicos de Git.
- Estrategias de Branches
- Workflows de trabajo.
- Presentación repositorio del Curso

3. Python 1 - Introducción

- Repaso de fundamentos del lenguaje, condicionales e iteradores.
- Debug en Python con VS Code.
- Entorno virtual, módulos y librerías.
- Tipos de datos y funciones

4. Python 2 – Diseño POO

- Diseño de clases en el paradigma orientado a objetos (draw.io, EA, visual paradigm)
- Modelo de dominio, diagrama de clases.
- Comportamiento y estado.
- Relaciones entre clases, agregación, composición y herencia.
- Concepto de polimorfismo

5. Python 3 – Herencia y encapsulamiento

- Clases y objetos, constructores, variables de instancia y de clase
- Visibilidad de atributos y métodos (públicos y privados).
- Generalización, herencia simple y múltiple
- Ejemplo polimorfismo.
- Clases abstractas.

6. Python 4 - Excepciones

- Manejo de excepciones
- Árbol de herencia en excepciones
- Creación de excepciones propias
- Lanzar excepciones de Python y propias
- Buenas prácticas en el manejo de excepciones

7. Python 5 - Integración de contenidos

- Resolución de ejercicios para la integración de contenidos

8. Python 6: Módulos y Paquetes

- Pip
- venv
- Pypi
- setuptools
- Instalación de ambiente de django

9. Django: Introducción

- Patrón MVT
- Estructura de directorios de Django
- Proyecto, configuraciones, aplicaciones, requirements
- Url Dispatcher

10. Django: Views - 1

- Creación de rutas
- Múltiples vistas y urls
- HttpResponse
- Vistas parametrizadas

11. Django: Views - 2

- Url y vistas parametrizadas.
- Analizando el HttpRequest
- Tipos de HttpResponse
- Html desde Django

12. Django: Templates - 1

- Templates backend
- Templates loader
- Configuración (distintas prácticas)
- Integración con vistas
- Contexto
- Variables

13. Django: Templates - 2

- Tags (csrf, url, now, etc)
- Etiquetas de condicionales y bucles

- Filter
- Comments

14. Django: Templates - 3

- Herencia de templates
- Plantilla base (distintas prácticas)
- Extend, bloques
- Estáticos: Css, js, imágenes, etc
- Incorporando Bootstrap al proyecto.

15. Django: Integración de contenidos

- Resolución de ejercicios para la integración de contenidos.

16. Django: Forms - 1

- Creación de formularios.
- Get y Post en formularios.
- Integración con templates y vistas
- Campos de formularios

17. Django: Forms – 2

- Widgets
- Validaciones en formularios (is_valid(), cleaned_data)
- Errores en formularios

18. DB - 1

- Repaso de base de datos (relacionales y no relacionales, dbms, db).
- Pasando de un modelo de clases a un modelo de datos relacional (der).
- Instalación PostgreSQL, PgAdmin.
- Repaso SQL, lenguaje DDL y DML.
- Repaso claves primarias, claves foráneas, relaciones.

19. DB - 2

- Vincular PostgreSQL con Django.
- Models first
- ORM, Django ORM, introducción a Models.
- Migraciones de Django (migrate).

20. Exposición parcial de proyectos y consultas

- Exposición de estado de proyectos integradores grupales
- Consultas varias.

21. Django: Models - 1

- Crear modelos
- Campos y atributos (DjangoFields).
- Django Shell

- Migraciones de Modelos (makemigrations, migrate).

22. Django: Models - 2

- Relaciones entre modelos.
- Herencia de Modelos
- QuerySets (lazyloading)

23. Django: Models – 3

- Crud sobre modelos.
- Integración de Modelos con Vistas y Templates
- Vistas basadas en clases.

24. Django: Models - 4

- Asociar un Modelo a un Formulario (ModelForm).
- Seleccionando campos a utilizar.
- Herencia de formularios.

25. Django: Integración de contenidos

- Resolución de ejercicios para la integración de contenidos.

26. Django: Admin - 1

- Configuración de DjangoAdmin
- Creación del superusuario.
- Integrando los modelos.

27. Django: Admin - 2

- Trabajando con modelos muchos a muchos
- Personalizando el DjangoAdmin.

28. Django: Authentication - 1

- Autenticación y autorización.
- Instalación
- Usuarios y grupos
- is_authenticated
- Decorator login_required
- Limitar acceso en templates

29. Django: Authentication - 2

- Configurando vistas y templates de autenticación.
- Registro de usuarios
- Login de usuarios
- Reinicio contraseña.
- Logout de usuarios

30. Django: Integración de contenidos

- Resolución de ejercicios para la integración de contenidos

31. Django: Rest framework

- WebSite vs WebApi.
- Instalación y configuración del django rest framework.
- Serializadores
- Vistas y Viewsets.
- Routers.
- Autenticación

32. Django: Despliegue en servidor

- Tipos de infraestructuras (IaaS, PaaS, SaaS)
- Load Balancer
- Python Anywhere
- Heroku
- Amazon EC2
- Docker

33. Exposición de proyectos y consultas

- Exposición de proyectos integradores grupales
- Consultas varias.

34. Exposición de proyectos y consultas

- Exposición de proyectos integradores grupales
- Consultas varias.

35. Exposición de proyectos y consultas

- Exposición de proyectos integradores grupales
- Consultas varias

36. Cierre de cursada

- Recomendaciones para el EFI.
- Retrospectiva: fortalezas y oportunidades de mejora.
- Próximos pasos. Después del curso, ¿qué puedo hacer? ¿Qué otras opciones se ofrecen en Codo a Codo?
- Cierre de cursada.

Proyecto a desarrollar durante la cursada

Se realizará un proyecto integral en equipos a lo largo de la cursada, el cual servirá para validar la adquisición de los conocimientos, y se hará en paralelo con el proyecto desarrollado por el/la docente. Las pautas del proyecto se les informará durante la cursada. El proyecto será una excelente herramienta para enriquecer tu portfolio y así poder aumentar tus probabilidades de empleabilidad en el sector IT.

Proyecto Integrador Grupal

En grupos de 4 o 5 personas, emulando un equipo de trabajo profesional con un líder de proyecto que será la vía de comunicación con el docente en representación del equipo, se desarrollará una aplicación web con Django con conexión a una Base de datos PostgreSQL. La web a desarrollar será un sistema para la gestión de datos. El sistema será tipo CRUD, denominado así por Create (Crear registros), Read (Leer/recuperar registros), Update (Actualizar registros), Delete (Borrar registros). Este sistema nos permitirá gestionar sin problemas nuestra base de datos, teniendo autenticación, páginas y formularios para poder integrar los conceptos correspondientes al desarrollo Front y Back-End así como también poner en práctica el uso de Django, librerías y el paradigma orientado a objetos.

Requerimientos

Requerimientos para cursar

- Contar con los siguientes conocimientos: conocimientos básicos de inglés, conocimientos de desarrollo Fullstack con Python (html, css, python, sql)
- Acceso a Internet, manejo básico de exploradores web (Chrome, Firefox, Edge, Opera, Safari), manejo básico de la computadora.

Requisitos para obtener el diploma

- Asistir al 75% de las clases en vivo (clases sincrónicas).
- Acceder semanalmente al Aula Virtual.
- Realizar los ejercicios obligatorios semanales.
- Aprobar la exposición del Proyecto Integrador Grupal.
- Aprobar el EFI (Examen Final Integrador).

Modelo de diploma

