



## 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: Construcción de software integrador de tecnologías orientadas a servicios
- Fase del Proyecto:
- Actividad de Proyecto:
- Competencia: DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO
- Resultados de Aprendizaje Alcanzar: CODIFICAR EL SOFTWARE DE ACUERDO CON EL DISEÑO ESTABLECIDO (Actividad 1)
- Duración de la Guía: 8 horas presenciales dos días a la semana + estudio autónomo

#### 2. PRESENTACIÓN

Codificar el software requiere comprender profundamente la estructura definida en el diseño, seleccionar adecuadamente el patrón de arquitectura y aplicar buenas prácticas en la organización del código. En esta guía se trabajará el patrón **Modelo-Vista-Controlador (MVC)** utilizando **Flask** como framework para el backend con Python. Se realizará una introducción a la codificación organizada, la conexión con bases de datos, uso de rutas, blueprints, y protección de datos mediante encriptación.

El aprendizaje se aplicará sobre un caso de estudio común: "**Lorenzo el Melómano**", donde los aprendices ya desarrollaron las vistas en Bootstrap. Todas las actividades de codificación se basarán en este caso, permitiendo validar que todos los aprendices trabajen en el ambiente y comprendan de forma práctica la codificación estructurada antes de pasar al desarrollo de su **proyecto formativo final**.

#### 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

A continuación, encontrarán las actividades sugeridas para garantizar la apropiación de conocimiento en el área de investigación aplicada a la formación profesional integral



### 3.1 Actividades de Reflexión inicial.

#### Actividad de Reflexión Inicial (15 minutos):

Antes de iniciar con el tema MVC, reflexiona sobre estas preguntas y comparte tus respuestas en una lluvia de ideas en clase:

- ¿Por qué creen que muchos proyectos pequeños terminan siendo difíciles de mantener o escalar?
- Conversatorio sobre la importancia de estructurar el código (buenas prácticas y patrones)

**Actividad:** En sus grupos de trabajo elabora un pdf respondiendo a las preguntas anteriores. En sus portafolios deben crear una carpeta con la fecha del día y ahí deben subir el pdf.

### 3.2. Actividades de apropiación del conocimiento (Conceptualización y Teorización).

#### Actividad 1: Investigativa (9 de junio):

- ¿Cuáles patrones de arquitectura existen en el desarrollo de software además del MVC?
- *Evidencia:* informe en pdf.

Realizan la entrega por la plataforma classroom

#### Actividad 2: Introducción teórica al MVC en Python con Flask (10 de junio):

- temas:
  - Explicación del patrón MVC con ejemplos sencillos.
  - Diagramas de componentes: modelos, vistas, controladores.
- *Evidencia:* mapa del flujo MVC del caso de estudio

#### Actividad 3 Taller de organización de proyecto Flask con MVC - caso Lorenzo el Melómano

(16 de junio):

- Tema:
  - Crear una estructura de carpetas para el proyecto.
  - Crear archivos iniciales: `app.py`, carpeta `templates`, `static`, `controllers`, `models` y `blueprints`.
- *Evidencia:* repositorio en GitHub o zip entregado en Classroom.



#### **Actividad 4 Conexión a base de datos MySQL (16 de junio):**

- Temas:
  - Explicación de la librería `mysql.connector` o SQLAlchemy.
  - Conexión desde Flask a la base de datos del caso de estudio Lorenzo.
- *Evidencia:* archivo funcional de conexión + captura de prueba.

#### **Actividad 5 Creación de rutas y blueprints (17 de junio):**

- Temas:
  - Explicación de rutas simples, organización modular.
  - Implementación de un módulo de autenticación (login/register).
- *Evidencia:* funcionamiento del sistema en local. Link del repositorio

#### **Actividad 6 Encriptación de contraseñas (23 de junio):**

- Temas:
  - Explicación de por qué encriptar datos sensibles.
  - Implementación con `werkzeug.security` o `bcrypt`.
- *Evidencia:* código funcional y pantallazo de prueba.

### **3.3. Actividades de transferencia del conocimiento.**

#### **Actividad 1 (24 de junio):**

- Proyecto final: Con los grupos de trabajos implementar un módulo funcional del caso Lorenzo (ej. artistas, usuarios, discos) usando Flask + MVC.
- Evidencia: repositorio con el módulo terminado. Este trabajo servirá como base para el desarrollo posterior del proyecto formativo.
- Fecha de entrega: 25 de junio.

#### **Actividad 2 (24 de junio):**

- Proyecto Formativo: Con los grupos del proyecto formativo, implementar el MVC del proyecto formativo.
- Fecha de entrega: 1 de julio.



#### 4. AMBIENTE DE FORMACIÓN Y MATERIALES

- Ambiente: 315 (10 computadores con acceso a internet)
- Plataforma Classroom y uso de Git

#### 5. ACTIVIDADES DE EVALUACIÓN

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

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
<b>Evidencias de Conocimiento:</b>  Participación en clase y entrega de evidencias  Evaluación de conocimiento	<ul style="list-style-type: none"><li>• Asistencia y compromiso activo.</li><li>• Comprensión de MVC y estructura y codificación.</li></ul>	Observación directa, lista de chequeo.  Evaluación práctica escrita
<b>Evidencias de Desempeño</b>  Código fuente y estructura web	<ul style="list-style-type: none"><li>• Aplicación adecuada de los conceptos.</li></ul>	Revisión de repositorio y portafolio  Revisión de classroom
<b>Evidencias de Producto:</b>  Informes e investigaciones  Código fuente y estructura web		Revisión de repositorio y portafolio

#### 5. GLOSARIO DE TÉRMINOS

- **MVC:** Patrón de diseño que separa lógicamente la interfaz, la lógica y los datos.
- **Blueprint:** Módulo reutilizable de rutas en Flask.
- **Encriptación:** Proceso para proteger datos transformándolos en cadenas seguras.
- **Controlador:** Componente que conecta la vista con la lógica.
- **Modelo:** Componente que representa la estructura de datos.



## 6. REFERENTES BIBLIOGRÁFICOS

- Documentación oficial de Flask: <https://flask.palletsprojects.com>
- MySQL Connector y SQLAlchemy Docs
- Repositorios de ejemplo en GitHub (educativos)
- FreeCodeCamp, W3Schools, MDN

## 7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Isaura Suarez Novoa	Instructor	Teleinformática	Mayo de 2025

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					