



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Analisis y Desarrollo de software
- Código del Programa de Formación: 228118
- Nombre del Proyecto Formativo: CONSTRUCCIÓN DE SOFTWARE INTEGRADOR DE TECNOLOGÍAS ORIENTADAS A SERVICIOS
- Fase del Proyecto : Planeacion
- Actividad de Proyecto Formativo: Diseñar la base de datos para su proyecto formativo.
- Competencia: Elaboración de la propuesta técnica del software.
- Resultados de Aprendizaje: Elaborar propuesta técnica del software de acuerdo con las especificaciones técnicas definidas
- Resultados de Aprendizaje: Plantear problemas matemáticos a partir de situaciones generadas en el contexto social y productivo.
- Resultados de Aprendizaje: Identificar modelos matemáticos de acuerdo con los requerimientos del problema planteado en contextos sociales y productivo.
- Resultados de Aprendizaje: Resolver problemas matemáticos a partir de situaciones generadas en el contexto social y productivo.
- Resultados de Aprendizaje: Proponer acciones de mejora frente a los resultados de los procedimientos matemáticos de acuerdo con el problema planteado.
- Duración de la Guía de Aprendizaje (horas): 90 horas Presenciales – 30 autonomo

2. PRESENTACIÓN

Bienvenidos al estudio de la Programación Orientada a Objetos (POO) en Java, las estructuras de datos y los fundamentos de programación, pilares esenciales para el desarrollo de software moderno. Esta guía ha sido diseñada para fortalecer tu comprensión de cómo se construyen



soluciones tecnológicas desde una perspectiva estructurada, utilizando clases, objetos, encapsulamiento, control de flujo y organización lógica del código para resolver problemas reales.

A lo largo de este proceso, profundizarás en la creación y uso de clases, atributos y métodos en Java, así como en la implementación de estructuras de datos como listas y colecciones para gestionar información de manera eficiente. Además, reforzarás conceptos fundamentales como variables, condicionales, ciclos y validaciones, integrándolos dentro de un enfoque orientado a objetos. El aprendizaje se desarrollará de manera progresiva, conectando la lógica algorítmica con la construcción de aplicaciones funcionales, promoviendo el análisis, la organización del código y la aplicación de buenas prácticas en el desarrollo de software.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es)**

3.1 Actividades de reflexión inicial:

Descripción de la actividad

Tema: La importancia de la Programación Orientada a Objetos y las estructuras de datos en el desarrollo de software

Objetivo

Reflexionar sobre cómo los fundamentos de programación, la POO y las estructuras de datos permiten construir aplicaciones organizadas, mantenibles y eficientes, comprendiendo su relevancia en el desarrollo de soluciones reales.

Contexto

En la vida cotidiana interactuamos con múltiples aplicaciones: sistemas de mensajería, plataformas educativas, videojuegos, aplicaciones bancarias y tiendas en línea. Detrás de cada una de ellas existe código estructurado mediante clases, objetos y estructuras de datos que permiten organizar la información y el comportamiento del sistema. Sin estos principios, el software sería difícil de mantener, escalar o mejorar.



Instrucciones

De forma individual o en pequeños grupos, analiza las siguientes preguntas y registra tus respuestas de manera clara y argumentada:

1. Menciona al menos tres aplicaciones que utilices frecuentemente y describe qué funcionalidades crees que están organizadas mediante clases u objetos.
2. ¿Qué tipo de datos consideras que gestionan estas aplicaciones y cómo podrían almacenarse usando estructuras como listas o colecciones?
3. ¿Qué problemas podrían surgir si un programa no estuviera organizado con POO y utilizara únicamente código desordenado o repetitivo?
4. ¿Por qué crees que las estructuras de control (if, switch, ciclos) son esenciales para que una aplicación tome decisiones correctamente?
5. ¿Qué ventajas aporta utilizar estructuras de datos frente a manejar variables sueltas sin organización?

Finalmente, se realizará una socialización grupal donde cada equipo compartirá sus conclusiones, promoviendo el análisis crítico sobre la importancia de escribir código estructurado, organizado y mantenible en el desarrollo de software.

Ambiente requerido:

- Aula presencial con acceso a computadores.
- Conexión a internet para consulta de ejemplos y recursos digitales.
- Espacio físico o digital para trabajo colaborativo (pizarra, papelógrafo o tableros virtuales como **Miro, Jamboard o Canva**).
- Proyector o pantalla para socialización de ideas y resultados (en modalidad presencial o virtual).

Estrategias o técnicas didácticas activas:

- **Lluvia de ideas:**
Los aprendices compartirán ejemplos de sistemas que utilizan diariamente y analizarán qué tipo de información almacenan (usuarios, productos, transacciones, calificaciones, registros).



- **Aprendizaje basado en preguntas (ABQ):**
Se plantearán preguntas orientadoras como:
 - ¿Qué sucede si la información de un sistema no está bien organizada?
 - ¿Por qué es importante evitar datos duplicados?
 - ¿Cómo se relacionan los datos entre sí dentro de un sistema?
 - **Construcción colaborativa:**
En pequeños grupos, los aprendices elaborarán un **mapa conceptual** que relacione los conceptos cotidianos de información con términos básicos del diseño de bases de datos relacionales, como **tabla, registro, campo, clave primaria y relación**.
 - **Discusión guiada:**
Los grupos socializarán sus mapas conceptuales y, con la orientación del instructor, reflexionarán sobre cómo una correcta estructura de datos facilita el acceso, la seguridad y el mantenimiento de la información en un sistema.
-

Material de formación:

- Guía introductoria sobre **POO** (PDF o presentación digital)
-

Material de apoyo:

- Hojas de trabajo o plantillas digitales para la elaboración del mapa conceptual.
 - Marcadores, papel o acceso a herramientas digitales como **Jamboard, Miro o Canva**.
 - Proyector o pantalla compartida para presentar ejemplos y resultados grupales.
 - Acceso opcional a un gestor de bases de datos o simulador en línea para observación inicial.
-

Duración de la actividad:

- **4 horas**
 - Introducción y lluvia de ideas individual.
 - Trabajo en grupos pequeños para reflexión y construcción del mapa conceptual.
 - Socialización de resultados y discusión guiada.
 - Cierre con reflexiones finales y conexión con los conceptos iniciales de diseño de bases de datos y SQL básico.



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

Descripción de la actividad:

¡Es momento de contextualizar y conectar tus conocimientos con la Programación Orientada a Objetos (POO) y los fundamentos de la programación en Java! En esta actividad reflexionarás sobre cómo los sistemas reales se modelan a partir de clases y objetos, y cómo la información y el comportamiento se organizan de manera estructurada para facilitar el mantenimiento, la reutilización y la escalabilidad del software.

De manera individual y colaborativa, analizarás cómo los elementos del mundo real pueden representarse mediante clases, atributos y métodos, comprendiendo conceptos esenciales como encapsulamiento, responsabilidad de clase y relación entre objetos. Asimismo, comenzarás a identificar cómo las estructuras de datos permiten agrupar y gestionar objetos dentro de una aplicación, y cómo las estructuras de control hacen posible que el sistema tome decisiones y ejecute procesos de forma lógica.

La actividad culminará con la elaboración de un esquema o mapa conceptual donde integres los conceptos trabajados: clase, objeto, atributo, método, encapsulamiento, colecciones y flujo de control. Este ejercicio servirá como punto de partida para comprender cómo se diseña una aplicación orientada a objetos y cómo estos principios constituyen la base para construir soluciones tecnológicas organizadas y eficientes.

Ambiente requerido:

- Espacio de trabajo presencial o virtual con acceso a computador.
- Conexión a internet para consulta de recursos digitales.
- Espacios para trabajo colaborativo (mesas grupales o salas virtuales en plataformas como **Zoom o Microsoft Teams**).
- Acceso a herramientas colaborativas digitales como **Jamboard, Miro o Canva** para la elaboración del mapa conceptual.

Estrategias o técnicas didácticas activas:

1. **Lluvia de ideas:**
2. **Preguntas para reflexionar:**
3. **Trabajo en equipo:**



4. Conversación grupal:

Materiales de formación:

- Guía introductoria (PDF o presentación) proporcionada por el instructor
- Ejemplos visuales de modelos de bases de datos simples para análisis.

Material de apoyo:

- Hojas o plantillas digitales para la elaboración del mapa conceptual.
- Lápices, marcadores o acceso a herramientas digitales como **Jamboard, Miro o Canva**.
- Pantalla o proyector (en modalidad presencial) para socialización de resultados.

Duración de la actividad:

- **12 horas**

3.3 Actividades de apropiación:

Descripción de la actividad:

Desarrollo de un Sistema Básico para Tienda de Bicicletas – Java (Consola)

En esta actividad desarrollarás una aplicación de consola que permita gestionar la información básica de una tienda de bicicletas. El objetivo es modelar un sistema real aplicando Programación Orientada a Objetos y utilizando estructuras de control para manejar la lógica del programa.

El sistema deberá permitir registrar bicicletas, clientes y ventas, organizando la información mediante clases y gestionándola a través de estructuras de datos como ArrayList. De esta manera, pondrás en práctica la creación de clases, encapsulamiento, constructores, métodos y la interacción entre objetos.

Requerimientos del Sistema



Gestión de Bicicletas

Cada bicicleta debe tener:

- Código (único)
- Marca
- Modelo
- Tipo (Montaña, Ruta, Urbana)
- Precio
- Cantidad en inventario

El sistema debe permitir:

- Registrar bicicleta.
 - Listar bicicletas.
 - Buscar bicicleta por código.
 - Actualizar inventario.
-

Gestión de Clientes

Cada cliente debe tener:

- Documento (único)
- Nombre
- Teléfono

El sistema debe permitir:

- Registrar cliente.
 - Listar clientes.
 - Buscar cliente.
-

Registro de Ventas

Cada venta debe:

- Asociar un cliente.
- Asociar una bicicleta.
- Registrar cantidad comprada.
- Calcular total de la venta.
- Descontar del inventario.



Reglas:

- No se puede vender más cantidad de la disponible.
- No se puede vender una bicicleta inexistente.
- No se puede registrar venta sin cliente.

Aspectos Técnicos Obligatorios

El proyecto debe incluir:

- Menú interactivo usando while y switch.
- Validaciones con if.
- Uso de ArrayList para almacenar objetos.
- Encapsulamiento (atributos privados + getters/setters).
- Métodos para calcular valores (por ejemplo total de venta).
- Separación básica en clases:
 - Bicicleta
 - Cliente
 - Venta
 - SistemaTienda (lógica principal)

Ambiente requerido:

- Aula presencial o virtual con acceso a computadores y conexión a internet.
- Espacio para trabajo individual y colaborativo (mesas grupales o salas virtuales en plataformas como **Zoom o Microsoft Teams**).
- Gestor de bases de datos instalado o en línea (**MySQL, PostgreSQL, SQLite o similar**).
- Herramienta de administración de bases de datos (**MySQL Workbench, pgAdmin, DBeaver o similar**).

Estrategias o técnicas didácticas activas:

- **Aprendizaje basado en problemas (ABP):**
Resolverás ejercicios prácticos como el diseño de una base de datos para un sistema sencillo (biblioteca, control de estudiantes, ventas), identificando las entidades y sus relaciones.



- **Modelado guiado:**
Seguirás ejemplos paso a paso con el instructor para construir diagramas entidad–relación y convertirlos en tablas relacionales.
- **Trabajo colaborativo:**
Trabajarás en pareja o en pequeños grupos para diseñar y mejorar un modelo de base de datos, promoviendo el análisis conjunto y la retroalimentación.
- **Práctica con SQL:**
Implementarás el diseño en un gestor de bases de datos utilizando sentencias SQL básicas como:
 - CREATE TABLE
 - INSERT
 - SELECT
 - UPDATE
 - DELETE

Materiales de formación:

- Documento o presentación
- Ejemplos de modelos de bases de datos resueltos.
- Guía de ejercicios prácticos de diseño e implementación SQL (proporcionada por el instructor).

Material de apoyo:

- Gestor de bases de datos y herramienta de administración (MySQL Workbench, pgAdmin, DBeaver).
- Guía rápida de SQL básico (PDF o enlace digital).
- Plantillas para diagramas entidad–relación.
- Videos cortos o tutoriales recomendados por el instructor (opcional).

Evidencias de aprendizaje:

- Diagrama entidad–relación del problema planteado.
- Script SQL de creación de tablas y relaciones.
- Consultas SQL básicas ejecutadas correctamente.
- Participación activa en el trabajo colaborativo y retroalimentación grupal.



Instrumentos de evaluación:

- **Rúbrica de evaluación:**
 - Diseño del modelo de datos (40%):
¿El diagrama entidad–relación representa correctamente el problema?
 - Aplicación de SQL básico (30%):
¿Se utilizan correctamente las sentencias SQL básicas?
 - Organización y consistencia de la base de datos (20%):
¿Las tablas y relaciones están correctamente definidas?
 - Trabajo colaborativo (10%):
¿Participa activamente en el trabajo en equipo?
- **Lista de chequeo:**
Verificación de entrega de evidencias (modelo, scripts y consultas SQL).

Duración de la actividad:

- **100 horas**

3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

¡Es momento de aplicar los conocimientos adquiridos en un escenario más cercano a la realidad del desarrollo de software! En esta actividad transferirás lo aprendido sobre Programación Orientada a Objetos, estructuras de datos y estructuras de control al diseño y mejora de una solución funcional más estructurada y organizada.

A partir del sistema básico de tienda desarrollado previamente, deberás evolucionar la aplicación incorporando mejoras propias de un entorno real: reorganizar el proyecto en paquetes, fortalecer el encapsulamiento, validar correctamente todas las entradas del usuario, evitar duplicación de código y aplicar principios básicos de diseño como responsabilidad única en las clases. Además, deberás optimizar la gestión de colecciones de objetos y mejorar la lógica del sistema mediante estructuras de control más robustas.

Trabajarás en equipo para analizar el diseño actual, identificar oportunidades de mejora y proponer una versión optimizada del sistema. Esta experiencia te permitirá comprender que



desarrollar software no consiste únicamente en que el programa funcione, sino en que esté correctamente estructurado, organizado y preparado para crecer. Con ello, consolidarás tu capacidad de modelar problemas reales mediante clases, objetos y lógica estructurada, fortaleciendo tu visión profesional del desarrollo en Java.

Ambiente requerido:

- Aula presencial o virtual con acceso a computadores y conexión a internet.
-

Estrategias o técnicas didácticas activas:

- **Aprendizaje basado en proyectos (ABP):**
Diseñarás e implementarás en equipo una base de datos relacional completa a partir de un caso práctico propuesto por el instructor.
 - **Resolución de problemas:**
Analizarás el problema planteado, identificarás la información necesaria y definirás las relaciones entre los datos para construir una solución coherente.
 - **Trabajo colaborativo:**
Distribuirás roles dentro del equipo (análisis, diseño, implementación y validación) para fomentar la organización, la cooperación y la responsabilidad compartida.
 - **Presentación y reflexión:**
Expondrás la solución desarrollada, explicando el diseño de la base de datos, las relaciones definidas y las consultas SQL utilizadas, reflexionando sobre el proceso y los aprendizajes obtenidos.
-

Materiales de formación:

- Documentación básica
 - Instrucciones del proyecto con objetivos claros y criterios mínimos, proporcionadas por el instructor.
-

Material de apoyo:

- Gestor y herramienta de administración de bases de datos.



- Plantillas para diagramas entidad–relación.
- Acceso a recursos en línea como **MDN Web Docs**, **W3Schools SQL** o documentación oficial del gestor de base de datos.
- Pizarra física o digital (**Jamboard**, **Miro**) para organizar el diseño del modelo de datos.

Evidencias de aprendizaje:

- Script SQL completo de creación de la base de datos.
- Consultas SQL básicas ejecutadas correctamente.

Instrumentos de evaluación:

- **Rúbrica de evaluación:**
- **Lista de chequeo:**
Verificación de entrega de evidencias (modelo, scripts y presentación).

Duración de la actividad:

- **4 horas**

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Planear (Reflexión y contextualización)	Análisis de la información y su organización	Lluvia de ideas, preguntas reflexivas y construcción de	Mapa conceptual grupal,	Relación clara entre la información del entorno y	Observación directa, rúbrica de



	en sistemas de información	mapa conceptual sobre bases de datos	participación en discusión	los conceptos básicos de bases de datos relacionales	reflexión, lista de chequeo
Hacer (Apropiación)	Diseño e implementación de bases de datos relacionales	Aprendizaje basado en problemas, modelado entidad-relación y práctica con SQL básico	Diagrama entidad-relación, scripts SQL, consultas ejecutadas	Correcta identificación de entidades, atributos y relaciones; uso adecuado de SQL básico	Rúbrica (diseño 40%, SQL básico 30%, organización del modelo 20%, trabajo colaborativo 10%), lista de chequeo
Verificar (Transferencia)	Desarrollo de un proyecto de base de datos relacional	Aprendizaje basado en proyectos, trabajo colaborativo y presentación	Base de datos implementada, script SQL completo, presentación del proyecto	Funcionalidad de la base de datos, integridad y consistencia de la información	Rúbrica (modelo de datos 40%, implementación SQL 30%, trabajo en equipo 20%, presentación 10%), lista de chequeo

5. GLOSARIO DE TÉRMINOS

Atributo

Característica o propiedad que describe una entidad dentro de una base de datos, representada como una columna en una tabla (Elmasri & Navathe, 2016).

Base de datos

Conjunto organizado de datos relacionados entre sí, almacenados de forma estructurada para facilitar su acceso, gestión y actualización (Connolly & Begg, 2015).



Base de datos relacional

Tipo de base de datos que organiza la información en tablas relacionadas entre sí mediante claves primarias y foráneas, siguiendo el modelo relacional (Date, 2004).

Clave foránea

Campo o conjunto de campos que establece una relación entre dos tablas, haciendo referencia a la clave primaria de otra tabla (Elmasri & Navathe, 2016).

Clave primaria

Atributo o conjunto de atributos que identifica de manera única cada registro dentro de una tabla (Connolly & Begg, 2015).

Entidad

Objeto del mundo real o conceptual sobre el cual se desea almacenar información dentro de una base de datos (Elmasri & Navathe, 2016).

Lenguaje SQL (Structured Query Language)

Lenguaje estándar utilizado para definir, manipular y consultar datos almacenados en bases de datos relacionales (Date, 2004).

Modelo Entidad–Relación (MER)

Modelo conceptual utilizado para representar gráficamente las entidades, atributos y relaciones de una base de datos antes de su implementación (Connolly & Begg, 2015).

Normalización

Proceso de organización de una base de datos con el objetivo de reducir la redundancia y mejorar la integridad de los datos (Elmasri & Navathe, 2016).

Tabla

Estructura básica de una base de datos relacional que organiza los datos en filas (registros) y columnas (atributos) (Date, 2004).

6. REFERENTES BIBLIOGRÁFICOS

Construya o cite documentos de apoyo para el desarrollo de la guía, según lo establecido en la guía de desarrollo curricular. (**BIBLIOGRAFÍA / WEBGRAFÍA**).

Libros

Connolly, T., & Begg, C. (2015).

Database systems: A practical approach to design, implementation, and management (6th ed.). Pearson Education.



Date, C. J. (2004).
An introduction to database systems (8th ed.). Addison-Wesley.

Elmasri, R., & Navathe, S. B. (2016).
Fundamentals of database systems (7th ed.). Pearson.

Recursos web y documentación

Oracle. (2024).
SQL tutorial. <https://docs.oracle.com/en/database/>

PostgreSQL Global Development Group. (2024).
PostgreSQL documentation. <https://www.postgresql.org/docs/>

W3Schools. (2024).
SQL tutorial. <https://www.w3schools.com/sql/>

Mozilla Developer Network. (2024).
MDN Web Docs – Databases. <https://developer.mozilla.org>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	CARLOS MANUEL NUÑEZ	Instructor	CCYT	8 de octubre del 2025

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					