



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 (si aplica): CONSTRUCCIÓN DE SOFTWARE INTEGRADOR DE TECNOLOGÍAS ORIENTADAS A SERVICIOS
- Fase del Proyecto (si aplica):Análisis
- Actividad de Proyecto Formativo (si aplica):Desarrollo de procesos lógicos
- Competencia: Construcción del software
- Resultados de Aprendizaje: Codificar el software de acuerdo con el diseño establecido.
- Duración de la Guía de Aprendizaje (horas): 2160

2. PRESENTACIÓN

En el desarrollo de software moderno, la capacidad de conectar aplicaciones con bases de datos y construir servicios backend robustos es una competencia fundamental para cualquier desarrollador Java. La evolución natural después de dominar la programación orientada a objetos es aprender a persistir información de manera estructurada y a desarrollar aplicaciones que sigan estándares profesionales de arquitectura y buenas prácticas.

Con este propósito, la presente guía formativa se orienta al desarrollo de aplicaciones Java con persistencia de datos utilizando JDBC y JPA, así como a la construcción de aplicaciones backend con Spring Boot. A lo largo de este proceso, se fortalecerán competencias relacionadas con arquitectura en capas, separación de responsabilidades, integración con bases de datos relacionales y aplicación de buenas prácticas en el desarrollo empresarial. El objetivo es que el aprendiz comprenda cómo



transformar una aplicación local en un backend estructurado, escalable y preparado para entornos reales de producción.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

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

3.1 Actividades de reflexión inicial:

Descripción de la actividad:

En el desarrollo de aplicaciones empresariales, la construcción de un backend sólido no surge de manera improvisada; requiere análisis, diseño estructurado y el uso adecuado de frameworks que permitan crear soluciones escalables y mantenibles. En el contexto actual, herramientas como Spring Boot facilitan la creación de servicios robustos, pero su correcta implementación depende de comprender arquitectura en capas, persistencia de datos y buenas prácticas de desarrollo.

A partir de esta idea, se propone iniciar un conversatorio en el que los aprendices reflexionen sobre la importancia de investigar, planificar y estructurar adecuadamente una aplicación backend antes de desarrollarla. ¿Por qué no basta con que el sistema funcione? ¿Qué papel cumplen frameworks como Spring Boot en la evolución del software empresarial? ¿Cómo influye una buena arquitectura en la sostenibilidad de un proyecto? Cada aprendiz deberá aportar su análisis, conectando la reflexión con la necesidad de construir aplicaciones organizadas, escalables y preparadas para entornos reales.

.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A



Duración de la actividad: 1 horas.

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

Descripción de la actividad:

Con el inicio del estudio de persistencia de datos con JPA y desarrollo de aplicaciones backend con Spring Boot, es necesario identificar los conocimientos previos que servirán como base para esta nueva etapa. En esta actividad, cada aprendiz responderá de manera individual una serie de preguntas diagnósticas relacionadas con: programación orientada a objetos, conexión a bases de datos, consultas SQL básicas, arquitectura en capas y principios de separación de responsabilidades.

Posteriormente, los participantes se organizarán en grupos de trabajo colaborativo. El instructor proporcionará el esquema de una “rueda de arquitectura backend”, la cual deberán dibujar en un papelógrafo. Cada integrante escribirá en uno de los segmentos un concepto clave relacionado con la temática, por ejemplo: Controlador (Controller), Servicio (Service), Repositorio (Repository), Entidad (Entity), Inyección de Dependencias, JPA, JDBC, API REST, Base de Datos, Arquitectura en Capas. Cada concepto deberá ser definido brevemente dentro del mismo esquema, explicando su función dentro de una aplicación Spring Boot.

Finalmente, cada grupo sintetizará los conceptos trabajados y los socializará ante el curso, explicando cómo se relacionan entre sí dentro de una aplicación backend. Con la orientación del instructor, se construirán conclusiones generales sobre los conocimientos necesarios para desarrollar aplicaciones Java con persistencia de datos, estableciendo una base conceptual sólida antes de iniciar la implementación práctica.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.



3.3 Actividades de apropiación:

Desarrollo de un Sistema Backend para Gestión de Reservas de Servicios

Contexto del Caso

Una empresa que ofrece servicios (asesorías, mantenimiento técnico, clases personalizadas, soporte técnico, entre otros) necesita digitalizar su proceso de gestión de reservas. Actualmente registran la información en hojas de cálculo, lo que genera inconsistencias, duplicidad de datos y dificultad para generar reportes.

Se requiere desarrollar un **backend en Java con Spring Boot**, aplicando arquitectura en capas y buenas prácticas. El sistema evolucionará progresivamente: iniciará gestionando datos en memoria y posteriormente integrará persistencia real mediante JDBC y JPA.

Objetivo Formativo

Que el aprendiz:

- Construya una API REST con Spring Boot.
 - Aplique arquitectura en capas (Controller, Service, Repository).
 - Utilice estructuras de datos para gestionar información en memoria.
 - Integre persistencia con JDBC.
 - Migre posteriormente a JPA aplicando mapeo objeto-relacional.
 - Implemente buenas prácticas de desarrollo backend.
-

Fase 1 – Backend con Spring Boot y Datos en Memoria

Objetivo

Comprender la arquitectura de Spring Boot y la separación de responsabilidades antes de trabajar con bases de datos.

Requerimientos

Construir una API REST que permita:

- Registrar clientes.
- Registrar servicios.



- Crear reservas.
- Consultar reservas.
- Cancelar reservas.

Restricción

En esta fase **NO** se utilizará base de datos.

Los datos se almacenarán en memoria utilizando:

- List
- Map
- ArrayList

Arquitectura obligatoria

- Controller
- Service
- Repository (simulado en memoria)
- Model (POJOs)

Requisitos técnicos

- Uso de @RestController
- Inyección de dependencias
- Respuestas HTTP adecuadas
- Validaciones básicas
- Manejo de excepciones global

El objetivo es entender claramente cómo fluye la información desde el controlador hasta la capa de servicio y repositorio.

Fase 2 – Persistencia con JDBC

Objetivo

Integrar la aplicación con una base de datos relacional utilizando JDBC.

Requerimientos

- Conectar el proyecto a MySQL o PostgreSQL.
- Reemplazar el repositorio en memoria por implementación JDBC.
- Implementar:
 - Registro de clientes.



- Registro de servicios.
- Creación de reservas.
- Consulta de reservas por cliente.
- Cancelación de reservas.

Requisitos técnicos

- Uso de Connection
- PreparedStatement
- ResultSet
- Manejo adecuado de excepciones SQL
- Prevención de SQL Injection
- Configuración de conexión desde application.properties

En esta fase el aprendiz debe comprender cómo funciona la persistencia manual.

Fase 3 – Migración a JPA

Objetivo

Simplificar la persistencia utilizando mapeo objeto-relacional.

Requerimientos

- Implementar entidades con:
 - @Entity
 - @Id
 - @GeneratedValue
 - @OneToMany
 - @ManyToOne
- Reemplazar implementación JDBC por Spring Data JPA.
- Usar JpaRepository.
- Validar generación automática de tablas.
- Mantener arquitectura en capas.

Análisis obligatorio

Comparar:

- Diferencias entre repositorio en memoria, JDBC y JPA.
- Cantidad de código requerido en cada enfoque.



- Nivel de abstracción.
 - Ventajas y limitaciones.
-

Buenas Prácticas Obligatorias

- No colocar lógica de negocio en el Controller.
- Manejo de excepciones centralizado.
- Uso de DTOs (opcional avanzado).
- Separación clara de paquetes.
- Código limpio y organizado.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listade Chequeo

Duración de la actividad: 1058 horas.



3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

Después de haber desarrollado el sistema de gestión de reservas utilizando Spring Boot con datos en memoria, posteriormente integrando JDBC y finalmente migrando a JPA, cuentas ahora con una visión completa de cómo evoluciona una aplicación backend desde una versión básica hasta una solución estructurada con persistencia profesional. Has trabajado arquitectura en capas, separación de responsabilidades, manejo de excepciones y conexión con bases de datos relacionales.

En esta etapa, transferirás estos conocimientos a un nuevo contexto: deberás diseñar y construir un backend similar para un dominio diferente (por ejemplo: sistema de inventario, gestión de citas médicas, control de pedidos o administración académica), aplicando la misma arquitectura, buenas prácticas y estrategias de persistencia aprendidas. El objetivo es demostrar que comprendes el modelo arquitectónico y los principios técnicos, y que puedes adaptarlos a nuevas necesidades, construyendo una solución organizada, escalable y alineada con estándares profesionales de desarrollo backend en Java con Spring Boot.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listado de Chequeo

Duración de la actividad: 1150 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
Analisis	Desarrollo de procesos lógicos	Desarrollo de algoritmos	Evidencias de Producto:	Realiza algoritmos utilizando variables,	EVIDENCIA: DESEMPEÑO conocimiento Técnica de



			Taller de algoritmos	constantes, condicionales, ciclos, bucles, Contadores, acumuladores, pseudocódigo.	Valoración: Observación Directa Instrumento de Evaluación: Listade Chequeo
--	--	--	----------------------	--	---

5. GLOSARIO DE TÉRMINOS

- Aplicación informática: Una aplicación es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas.
- ACTIVIDAD FÍSICA: actividad física es "cualquier movimiento corporal producido por los músculos esqueléticos que produzca un gasto energético mayor al existente en reposo.
- EXPRESIÓN CORPORAL: El cuerpo humano no solamente se mueve, también se comunica. En este sentido, se habla de la expresión corporal como el conjunto de maneras que tiene nuestro organismo de transmitir sentimientos y emociones.
- Los coreógrafos y bailarines tienen que ejercitar la capacidad de sentir el movimiento del cuerpo para que el espectador perciba igualmente unas emociones.
- El cuerpo humano puede adquirir su propio lenguaje, con un ritmo y con una intensidad. Y el resultado es la expresión corporal.
- Factores de riesgo ergonómico en el trabajo: Los factores de riesgo osteomuscular

en el trabajo están en general bien identificados, relacionándose con la manipulación de cargas, las malas posturas de trabajo, los movimientos forzados, los movimientos repetidos, los movimientos enérgicos, la presión mecánica directa sobre los tejidos corporales o las vibraciones, por citar algunos de las situaciones de exposición más comunes.

-
- Higiene Laboral: Conjunto de normas y procedimientos tendientes a la protección de la integridad física y mental del trabajador, preservándolo de los riesgos de salud inherentes a las tareas a su cargo y al ambiente físico donde se ejecutan.
- Higiene Postural: es el conjunto de normas, cuyo objetivo es mantener la correcta posición del cuerpo, en quietud o en movimiento y así evitar posibles lesiones aprendiendo a pro teger princ



ipalm ente la co lum na vertebral, al realizar las actividades diarias, evitando que se presenten dolores y disminuyendo el riesgo.

- desarrollo de actividades y medidas necesarias para prevenir los riesgos
- COREOGRAFIA: Es el arte de crear estructuras en las que suceden movimientos. La estructura de movimientos puede ser considerada como la coreografía.

6. REFERENTES BIBLIOGRÁFICOS

- Lumowell-fitness-es. Entrenamiento Funcional. Recuperado de:
<https://www.youtube.com/watch?v=XsDJr-qDoG4>.
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/798/el-shell-korn.-manual-de-usuario-y-programador>
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/39320/manual-de-citas-y-referencias-bibliograficas---latino--apa--chicago--ieee--mla-y-vancouver--2a-ed.->
- Instituto Nacional de Seguridad e Higiene en el Trabajo. Anuario de estadísticas laborales y de asuntos sociales, 2006. Disponible en:
http://www.mtas.es/insht/statistics/est_anuar.htm.
- Ergonomía en oficinas. Manual de conceptos fundamentales y recomendaciones prácticas. Recuperado de:
<https://www.achs.cl/portal/trabajadores/Capacitacion/CentrodeFichas/Documents/ergonomia-para-oficinas-conceptos-fundamentales-y-recomendaciones-practicas.pdf>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	CARLOS MANUEL NUÑEZ LOPEZ	INSTRUCTOR	SOFTWARE	25/07/2025

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



	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



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 (si aplica): CONSTRUCCIÓN DE SOFTWARE INTEGRADOR DE TECNOLOGÍAS ORIENTADAS A SERVICIOS
- Fase del Proyecto (si aplica):Análisis
- Actividad de Proyecto Formativo (si aplica):Desarrollo de procesos lógicos
- Competencia: Evaluar requisitos de la solución de software de acuerdo con metodologías de análisis y estándares
- Resultados de Aprendizaje: Desarrollar procesos lógicos a través de la implementación de algoritmos
- Duración de la Guía de Aprendizaje (horas): 2160

2. PRESENTACIÓN



Según Gardner todos los seres humanos tenemos un tipo de inteligencia y un método de aprendizaje distinto, entre ellas, destaca la habilidad lógico-matemática, inteligencia considerada como la inteligencia en bruto, es aquella que como su nombre indica es la que vincula a la habilidad ante el razonamiento lógico y la resolución de problemas matemáticos.



Con el objetivo de facilitar el proceso de aprendizaje que permite desarrollar las habilidades lógico – matemáticas la presente guía reúne los elementos necesarios para diagnosticar, resolver problemas, tomar decisiones, capacidad de comunicación, de organización en el trabajo, de gestión de tiempo y de adaptación en diferentes entornos utilizando como pilar el trabajo autónomo.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

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

3.1 Actividades de reflexión inicial:

Descripción de la actividad:

Tomás Andreu (CEO WE TRIBU) explica que la constitución de su empresa “vino precedida de un trabajo enorme de investigación y desarrollo, hasta finalizar un producto sólido de software que fuese capaz de dar solución a las demandas en gestión de empresas de gerentes, directivos y usuarios”. En definitiva, no se trata de inventar la pólvora, pero todavía queda software por inventar –tanto para el entorno empresarial como para el doméstico– y aspectos por modificar en un negocio que, por su propia esencia, está en permanente evolución.

Inicia un conversatorio donde todos los aprendices aporten su concepto sobre la frase dicha por Tomas Andreu.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.



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

Descripción de la actividad:

De manera individual cada aprendiz responderá las siguientes preguntas correspondientes a los conocimientos necesarios para su aprendizaje.

Los participantes se organizan en grupos de trabajo colaborativos, el instructor les facilitará la rueda de la fortuna la que deberán graficar en una hoja de papelógrafo.

Luego cada participante de los grupos de trabajo colaborativo, escribe una palabra que identifique un componente relacionado con la programación de software y Algoritmos.

Esta palabra la deberán ubicar en un recuadro de la rueda de la fortuna, la cual debe ser definida en la rueda. Cada grupo sintetiza los conceptos que aparecen en la rueda de la fortuna.

Los grupos exponen a los demás aprendices los conceptos que están descritos en cada rueda de la fortuna. Con la orientación del instructor los aprendices elaboran las conclusiones sobre la actividad realizada.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.



3.3 Actividades de apropiación:

Descripción de la actividad:

En esta actividad tendrás la oportunidad de desarrollar una serie de ejercicios algorítmicos que te permitirán fortalecer tu capacidad de análisis y tu pensamiento lógico. A través de estos retos prácticos, aprenderás a organizar ideas, plantear soluciones paso a paso y transformar problemas en procesos claros y estructurados. Más allá de aprender a resolver ejercicios, estarás construyendo habilidades que te servirán en tu vida académica, profesional y personal, ya que el pensamiento lógico es la base para enfrentar con éxito cualquier desafío. ¡Cada algoritmo que resuelvas será un paso más hacia convertirte en un pensador creativo, estratégico y capaz de proponer soluciones innovadoras!

Taller 1

- Escribir un Pseudocódigo - la edad y peso de una persona y posteriormente imprimirla.
- Escribir un Pseudocódigo que calcule el área de un triángulo recibiendo como entrada el valor de base y altura. |
- Escribir Pseudocódigo que calcule el área de un círculo.
- Escribir un Pseudocódigo que calcule la conversión de metros a centímetros.
- Elaborar un algoritmo que solicite el número de respuestas correctas, incorrectas y en blanco, correspondientes a postulantes y muestre su puntaje final considerando, que por cada respuesta correcta tendrá 4 puntos, respuestas incorrectas tendrá -1 punto y respuestas en blanco 0 puntos.
- Se requiere un algoritmo para elaborar una planilla de empleados, para ello se dispone de sus horas laboradas en el mes, así como la tarifa por hora.
- Elaborar un algoritmo que permita sacar el promedio de una materia, la cual se compone de 4 notas de 1 a 5.

Taller 2

- Leer un número y mostrar si es positivo, negativo
- Leer un número y mostrar si es par o impar.
- Leer la edad de una persona y decir si es mayor de edad (≥ 18) o menor.
- Leer dos números y mostrar cuál es mayor
- Verificar si un número ingresado es múltiplo de 5
- Leer una nota del 0 al 10 y decir si aprueba (≥ 6) o no.



- Leer un número y verificar si está entre 1 y 100.

Taller 3

- Pide un número. Mostrar "Número válido" si está entre 50 y 100 (inclusive).
- Pide dos notas. Si ambas son mayores o iguales a 6, mostrar "Aprobado". Si una de ellas es menor, mostrar "Debe recuperar".
- Pide si es de noche (booleano) y si es fin de semana (booleano). Si al menos una es verdadera, mostrar "Puedes descansar".
- Pide un número. Si el número es menor que 0 o mayor que 100, mostrar "Fuera de rango"

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listade Chequeo

Duración de la actividad: 1058 horas.



3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

Después de haber desarrollado y practicado diversas actividades algorítmicas, ya cuentas con una base sólida en el pensamiento lógico y en la resolución estructurada de problemas. Estos conocimientos son el punto de partida ideal para dar el siguiente paso: iniciar con un lenguaje de programación. A partir de ahora comenzaremos a trabajar con Java, un lenguaje ampliamente utilizado en la industria del software que te permitirá llevar a la práctica los algoritmos creados, transformándolos en programas funcionales. Este proceso marcará una nueva etapa en tu aprendizaje, en la que podrás aplicar lo aprendido previamente y comenzar a construir soluciones tecnológicas reales.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listade Chequeo

Duración de la actividad: 1150 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
Analisis	Desarrollo de procesos lógicos	Desarrollo de algoritmos	Evidencias de Producto: Taller de algoritmos	Realiza algoritmos utilizando variables, constantes, condicionales, ciclos, bucles, Contadores, acumuladores, pseudocódigo.	EVIDENCIA: DESEMPEÑO conocimiento Técnica de Valoración: Observación Directa Instrumento de Evaluación: Listade Chequeo



5. GLOSARIO DE TÉRMINOS

- Aplicación informática: Una aplicación es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas.
- ACTIVIDAD FÍSICA: actividad física es "cualquier movimiento corporal producido por los músculos esqueléticos que produzca un gasto energético mayor al existente en reposo.
- EXPRESIÓN CORPORAL: El cuerpo humano no solamente se mueve, también se comunica. En este sentido, se habla de la expresión corporal como el conjunto de maneras que tiene nuestro organismo de transmitir sentimientos y emociones.
- Los coreógrafos y bailarines tienen que ejercitar la capacidad de sentir el movimiento del cuerpo para que el espectador perciba igualmente unas emociones.
- El cuerpo humano puede adquirir su propio lenguaje, con un ritmo y con una intensidad. Y el resultado es la expresión corporal.
- Factores de riesgo ergonómico en el trabajo: Los factores de riesgo osteomuscular

en el trabajo están en general bien identificados, relacionándose con la manipulación de cargas, las malas posturas de trabajo, los movimientos forzados, los movimientos repetidos, los movimientos enérgicos, la presión mecánica directa sobre los tejidos corporales o las vibraciones, por citar algunos de las situaciones de exposición más comunes.

-
- Higiene Laboral: Conjunto de normas y procedimientos tendientes a la protección de la integridad física y mental del trabajador, preservándolo de los riesgos de salud inherentes a las tareas a su cargo y al ambiente físico donde se ejecutan.
- Higiene Postural: es el conjunto de normas, cuyo objetivo es mantener la correcta posición del cuerpo, en quietud o en movimiento y así evitar posibles lesiones aprendiendo a pro teger princ ipalm ente la co lum na vertebral, al realizar las actividades diarias, evitando que se presenten dolores y disminuyendo el riesgo.
- desarrollo de actividades y medidas necesarias para prevenir los riesgos



- COREOGRAFIA: Es el arte de crear estructuras en las que suceden movimientos. La estructura de movimientos puede ser considerada como la coreografía.

6. REFERENTES BIBLIOGRÁFICOS

- Lumowell-fitness-es. Entrenamiento Funcional. Recuperado de:
<https://www.youtube.com/watch?v=XsDJr-qDoG4>.
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/798/el-shell-korn.-manual-de-usuario-y-programador>
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/39320/manual-de-citas-y-referencias-bibliograficas---latino--apa--chicago--ieee--mla-y-vancouver--2a-ed.->
- Instituto Nacional de Seguridad e Higiene en el Trabajo. Anuario de estadísticas laborales y de asuntos sociales, 2006. Disponible en:
http://www.mtas.es/insht/statistics/est_anuar.htm.
- Ergonomía en oficinas. Manual de conceptos fundamentales y recomendaciones prácticas. Recuperado de:
<https://www.achs.cl/portal/trabajadores/Capacitacion/CentrodeFichas/Documents/ergonomia-para-oficinas-conceptos-fundamentales-y-recomendaciones-practicas.pdf>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	CARLOS MANUEL NUÑEZ LOPEZ	INSTRUCTOR	SOFTWARE	25/07/2025

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



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)					



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 (si aplica): CONSTRUCCIÓN DE SOFTWARE INTEGRADOR DE TECNOLOGÍAS ORIENTADAS A SERVICIOS
- Fase del Proyecto (si aplica):Análisis
- Actividad de Proyecto Formativo (si aplica):Desarrollo de procesos lógicos
- Competencia: Análisis de la especificación de requisitos del software
- Resultados de Aprendizaje: Modelar las funciones del software de acuerdo con el informe de requisitos.
- Duración de la Guía de Aprendizaje (horas): 2160

2. PRESENTACIÓN

En el desarrollo de software, seleccionar correctamente herramientas de hardware y software es fundamental para garantizar eficiencia, cumplimiento legal y seguridad. Esta guía permitirá al aprendiz analizar necesidades tecnológicas, comprender aspectos de licenciamiento y reconocer riesgos y vulnerabilidades, aplicando buenas prácticas de desarrollo seguro durante todo el ciclo de vida del software.

Asimismo, se fortalecerán los principios de administración de la configuración, comprendiendo la importancia del versionamiento, el control de cambios y la organización de los elementos del sistema. Con ello, el aprendiz podrá mantener trazabilidad, orden y coherencia en los proyectos tecnológicos, asegurando calidad y sostenibilidad en las soluciones desarrolladas.



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

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

3.1 Actividades de reflexión inicial:

Descripción de la actividad:

En el desarrollo de soluciones tecnológicas no basta con elegir herramientas por tendencia o popularidad; es necesario realizar un análisis previo que contemple necesidades reales, aspectos legales de licenciamiento, riesgos de seguridad y sostenibilidad a largo plazo. Asimismo, una gestión adecuada de la configuración permite mantener control y orden en los cambios que se realizan durante el ciclo de vida del software.

A partir de esta idea, se propone iniciar un conversatorio en el que los aprendices reflexionen sobre la importancia de investigar antes de adquirir herramientas tecnológicas, identificar posibles vulnerabilidades y aplicar buenas prácticas de administración de la configuración. Cada aprendiz deberá aportar su punto de vista sobre cómo estas decisiones influyen en la calidad, seguridad y éxito de un proyecto de software.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.

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

Descripción de la actividad:

De manera individual, cada aprendiz responderá una serie de preguntas diagnósticas relacionadas con la selección de herramientas tecnológicas, aspectos de licenciamiento, seguridad digital y administración de la configuración, con el fin de identificar conocimientos previos sobre la temática.



Posteriormente, los participantes se organizarán en grupos de trabajo colaborativo. El instructor proporcionará el esquema de una “rueda de la fortuna”, la cual deberán dibujar en un papelógrafo. Cada integrante escribirá en uno de los segmentos un concepto clave relacionado con adquisición de herramientas, análisis de vulnerabilidades o gestión de configuración (por ejemplo: licenciamiento, control de versiones, riesgo, trazabilidad, actualización, cumplimiento legal). Cada término deberá definirse dentro del mismo esquema. Finalmente, cada grupo sintetizará los conceptos trabajados y los socializará ante el curso, construyendo conclusiones con la orientación del instructor.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.

3.3 Actividades de apropiación:

Descripción de la actividad: Desarrollo de un Sistema de Gestión para Biblioteca Digital (Consola – Java)

Contexto del Caso

Una biblioteca municipal desea modernizar su proceso de registro y préstamo de libros. Actualmente todo se realiza de forma manual, lo que genera errores, pérdida de información y falta de control sobre los préstamos.



La dirección solicita el desarrollo de una **aplicación de consola en Java** que permita administrar libros, usuarios y préstamos, garantizando:

- Validación correcta de datos.
- Control básico de acceso por roles.
- Manejo organizado de configuración del sistema.
- Registro estructurado de la información.
- Aplicación de buenas prácticas de desarrollo.

El sistema será desarrollado bajo paradigma orientado a objetos y utilizando estructuras de control básicas.

Requerimientos Funcionales

El sistema deberá permitir:

Gestión de Libros

- Registrar libro.
- Listar libros.
- Buscar libro por ISBN.
- Actualizar disponibilidad.
- Eliminar libro.

Cada libro debe contener:

- ISBN (único y validado)
- Título
- Autor
- Año de publicación
- Estado (Disponible / Prestado)

Validaciones obligatorias:

- ISBN no puede repetirse.
- Año no puede ser mayor al actual.
- Ningún campo puede estar vacío.

Gestión de Usuarios

- Registrar usuario.
- Listar usuarios.



- Buscar usuario por documento.
- Eliminar usuario.

Cada usuario debe contener:

- Documento (único)
- Nombre completo
- Tipo de usuario (Administrador / Bibliotecario / Lector)

Validaciones:

- Documento único.
 - Tipo de usuario válido.
 - Campos obligatorios completos.
-

Gestión de Préstamos

- Registrar préstamo.
- Registrar devolución.
- Consultar préstamos activos.

Reglas:

- No se puede prestar un libro no disponible.
- Un usuario lector solo puede tener máximo 3 préstamos activos.
- Solo un usuario con rol Bibliotecario o Administrador puede registrar préstamos.

Requerimientos de Seguridad Básicos

El sistema debe implementar:

- Inicio de sesión.
- Validación de usuario y contraseña.
- Control de acceso por rol.
- Bloqueo después de 3 intentos fallidos.

No se permite:

- Acceso a funciones sin autenticación.
- Manipulación directa de datos sin validación.



Requerimientos de Configuración

El sistema debe incluir un archivo de configuración (por ejemplo: config.properties) que contenga:

- Cantidad máxima de préstamos permitidos.
- Nombre de la biblioteca.
- Versión del sistema.
- Número máximo de intentos de login.

El programa deberá:

- Leer estos valores al iniciar.
- Usarlos dinámicamente.
- Mostrar mensaje de error si el archivo no existe.

Requerimiento de Análisis Tecnológico

Antes de desarrollar, cada aprendiz debe:

1. Justificar por qué se utilizará Java para esta solución.
2. Investigar:
 - Tipo de licencia de Java.
 - Ventajas frente a otros lenguajes para este caso.
3. Identificar posibles vulnerabilidades:
 - Inyección de datos por consola.
 - Manipulación indebida de roles.
 - Falta de validación.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listado Chequeo

Duración de la actividad: 1058 horas.



3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

Después de haber desarrollado el sistema de gestión en Java y aplicado principios de programación orientada a objetos, validaciones y estructuras de control, es momento de llevar el aprendizaje a un nivel más profesional. En esta etapa, el enfoque no estará únicamente en que el programa funcione, sino en cómo se gestiona, controla y mantiene el software durante su ciclo de vida, aplicando principios de administración de la configuración, control de versiones y buenas prácticas de seguridad.

A partir de esta actividad, deberás organizar tu proyecto como si formara parte de un entorno empresarial real: documentar versiones, gestionar cambios, controlar adecuadamente los archivos de configuración, justificar las herramientas utilizadas y analizar posibles riesgos de seguridad. El objetivo es que transfieras los conocimientos técnicos adquiridos hacia una visión más integral del desarrollo de software, comprendiendo que la calidad de un sistema no solo depende del código, sino también de su correcta gestión, trazabilidad y control.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listade Chequeo

Duración de la actividad: 1150 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
Analisis	Desarrollo de procesos lógicos	Adminsitracion de la configuracion	Evidencias de Producto: Proyecto biblioteca	Configuracion, software, códigos, validaciones	EVIDENCIA: DESEMPEÑO conocimiento Técnica de Valoración:



					Observación Directa Instrumento de Evaluación: Listade Chequeo
--	--	--	--	--	---

5. GLOSARIO DE TÉRMINOS

- Aplicación informática: Una aplicación es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas.
- ACTIVIDAD FÍSICA: actividad física es "cualquier movimiento corporal producido por los músculos esqueléticos que produzca un gasto energético mayor al existente en reposo.
- EXPRESIÓN CORPORAL: El cuerpo humano no solamente se mueve, también se comunica. En este sentido, se habla de la expresión corporal como el conjunto de maneras que tiene nuestro organismo de transmitir sentimientos y emociones.
- Los coreógrafos y bailarines tienen que ejercitar la capacidad de sentir el movimiento del cuerpo para que el espectador perciba igualmente unas emociones.
- El cuerpo humano puede adquirir su propio lenguaje, con un ritmo y con una intensidad. Y el resultado es la expresión corporal.
- Factores de riesgo ergonómico en el trabajo: Los factores de riesgo osteomuscular

en el trabajo están en general bien identificados, relacionándose con la manipulación de cargas, las malas posturas de trabajo, los movimientos forzados, los movimientos repetidos, los movimientos enérgicos, la presión mecánica directa sobre los tejidos corporales o las vibraciones, por citar algunos de las situaciones de exposición más comunes.

-
- Higiene Laboral: Conjunto de normas y procedimientos tendientes a la protección de la integridad física y mental del trabajador, preservándolo de los riesgos de salud inherentes a las tareas a su cargo y al ambiente físico donde se ejecutan.
- Higiene Postural: es el conjunto de normas, cuyo objetivo es mantener la correcta posición del cuerpo, en quietud o en movimiento y así evitar posibles lesiones aprendiendo a pro teger princ



ipalm ente la co lum na vertebral, al realizar las actividades diarias, evitando que se presenten dolores y disminuyendo el riesgo.

- desarrollo de actividades y medidas necesarias para prevenir los riesgos
- COREOGRAFIA: Es el arte de crear estructuras en las que suceden movimientos. La estructura de movimientos puede ser considerada como la coreografía.

6. REFERENTES BIBLIOGRÁFICOS

- Lumowell-fitness-es. Entrenamiento Funcional. Recuperado de:
<https://www.youtube.com/watch?v=XsDJr-qDoG4>.
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/798/el-shell-korn.-manual-de-usuario-y-programador>
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/39320/manual-de-citas-y-referencias-bibliograficas---latino--apa--chicago--ieee--mla-y-vancouver--2a-ed.->
- Instituto Nacional de Seguridad e Higiene en el Trabajo. Anuario de estadísticas laborales y de asuntos sociales, 2006. Disponible en:
http://www.mtas.es/insht/statistics/est_anuar.htm.
- Ergonomía en oficinas. Manual de conceptos fundamentales y recomendaciones prácticas. Recuperado de:
<https://www.achs.cl/portal/trabajadores/Capacitacion/CentrodeFichas/Documents/ergonomia-para-oficinas-conceptos-fundamentales-y-recomendaciones-practicas.pdf>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	CARLOS MANUEL NUÑEZ LOPEZ	INSTRUCTOR	SOFTWARE	25/07/2025

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



	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



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: Construcción del Software.
- Resultados de Aprendizaje: Construir la base de datos para el software a partir del modelo de datos.
- Duración de la Guía de Aprendizaje (horas): 90 horas Presenciales – 30 autonomo

2. PRESENTACIÓN

Bienvenidos al estudio de los fundamentos de las bases de datos, un componente esencial en el almacenamiento, organización y gestión de la información en los sistemas informáticos actuales. Esta guía formativa ha sido diseñada para introducirte de manera clara y progresiva en los conceptos básicos que permiten comprender qué es una base de datos, por qué es necesaria y cómo se utiliza para soportar aplicaciones y procesos en distintos contextos tecnológicos.

A lo largo de este recorrido, conocerás cómo han evolucionado las bases de datos y aprenderás a identificar los principales tipos existentes, como las bases de datos relacionales y no relacionales, comprendiendo sus características, ventajas y casos de uso más comunes. Asimismo, explorarás el papel fundamental de los sistemas gestores de bases de datos (SGBD), entendiendo su función como intermediarios entre los usuarios, las aplicaciones y los datos, y analizando ejemplos ampliamente utilizados en la industria.

El proceso de aprendizaje se desarrollará de forma estructurada, conectando tus conocimientos previos en informática y lógica de programación con nuevos conceptos como almacenamiento de datos, modelos de bases de datos, persistencia de la información y administración básica de datos.



Esto te permitirá construir una base sólida para estudios posteriores en diseño, modelado y consulta de bases de datos.

Además, se fomentará una actitud activa y reflexiva frente al aprendizaje, promoviendo la investigación, el análisis comparativo y el trabajo colaborativo como herramientas clave para comprender cómo se selecciona y utiliza una base de datos según las necesidades de un sistema. Prepárate para descubrir cómo las bases de datos y sus gestores constituyen el núcleo de soluciones tecnológicas confiables, eficientes y escalables.

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 las bases de datos en la vida cotidiana

Objetivo:

Reflexionar sobre el uso de la información en situaciones reales y reconocer la necesidad de las bases de datos como herramientas fundamentales para organizar, almacenar y gestionar datos de forma eficiente.

Contexto:

En la vida diaria interactuamos constantemente con sistemas que almacenan información: aplicaciones móviles, redes sociales, plataformas educativas, sistemas bancarios, tiendas en línea, entre otros. Detrás de cada uno de estos sistemas existe una base de datos que permite registrar, consultar y mantener la información de manera ordenada y segura.

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

Menciona al menos tres aplicaciones o sistemas que utilices diariamente y que consideres que almacenan información.

¿Qué tipo de datos crees que se guardan en cada uno de esos sistemas? (por ejemplo: datos personales, registros de uso, transacciones, etc.).

¿Qué problemas podrían presentarse si esa información no estuviera organizada o se almacenara de manera desordenada?



¿Cómo crees que una base de datos ayuda a garantizar que la información sea segura, accesible y confiable?

¿Qué diferencias imaginas entre guardar información en archivos simples (como hojas de cálculo o documentos) y utilizar una base de datos?

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.
-



Materiales de formación:

- Guía introductoria sobre **bases de datos relacionales y SQL básico** (PDF o presentación digital) que incluya:
 - ¿Qué es una base de datos?
 - ¿Qué es una base de datos relacional?
 - Conceptos básicos: tablas, registros y campos.
 - Introducción al lenguaje SQL.
 - Ejemplos visuales de modelos simples de bases de datos para análisis.
-

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 los fundamentos de las bases de datos y el uso básico de SQL! En esta actividad, reflexionarás sobre cómo la información que se maneja en distintos sistemas debe ser almacenada y organizada de forma clara y ordenada para facilitar su consulta y mantenimiento. A partir de tus experiencias previas, identificarás ideas



relacionadas con el manejo de información, el uso de datos en aplicaciones cotidianas y nociones básicas de lógica.

De manera individual y colaborativa, analizarás cómo los datos se agrupan y representan dentro de una base de datos, reconociendo conceptos esenciales como registros, campos y tablas. Asimismo, comenzarás a comprender cómo estas estructuras permiten guardar información de manera eficiente y cómo el lenguaje SQL se utiliza para interactuar con los datos mediante operaciones básicas como crear, consultar y modificar información.

La actividad culminará con la elaboración de un esquema o mapa conceptual que integre los conceptos trabajados, sirviendo como punto de partida para comprender qué es una base de datos, cómo se organiza la información y cuál es el rol de SQL como herramienta fundamental para acceder y gestionar los datos. ¡Es el momento de observar, analizar y construir una base sólida para tu aprendizaje!

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:**
Identifica sistemas de información conocidos (académicos, comerciales, administrativos) y reflexiona sobre los tipos de datos que almacenan y cómo se relacionan entre sí.
2. **Preguntas para reflexionar:**
Responde de manera individual o grupal:
 - ¿Qué información es indispensable para un sistema de información?
 - ¿Cómo se puede evitar la duplicación de datos?
 - ¿Por qué es importante definir relaciones entre la información?
3. **Trabajo en equipo:**
En grupos pequeños, construyan un **mapa conceptual** que relacione conocimientos previos con conceptos clave del diseño de bases de datos, tales como:
 - Entidad



- Atributo
- Relación
- Clave primaria
- Clave foránea

4. Conversación grupal:

Socializa el mapa conceptual con el grupo y participa en la discusión guiada por el instructor, quien orientará la comprensión de cómo estos conceptos se aplican en el diseño de una base de datos relacional y su posterior implementación con SQL.

Materiales de formación:

- Guía introductoria (PDF o presentación) proporcionada por el instructor, que incluya:
 - Conceptos básicos de bases de datos relacionales.
 - Modelo Entidad–Relación (MER).
 - Introducción a la normalización.
 - Fundamentos del lenguaje SQL.
- 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**.
- Acceso opcional a un gestor de bases de datos (MySQL, PostgreSQL o SQLite) para observación inicial.
- Pantalla o proyector (en modalidad presencial) para socialización de resultados.

Duración de la actividad:

- **12 horas**
 - Reflexión individual sobre conocimientos previos relacionados con la información.
 - Trabajo colaborativo para la construcción del mapa conceptual.
 - Socialización de resultados y discusión guiada.
 - Cierre del instructor conectando los conceptos con los temas que se abordarán en el diseño de bases de datos y SQL básico.

3.3 Actividades de apropiación:



Descripción de la actividad:

En esta actividad, pondrás en práctica los conceptos básicos sobre bases de datos para comprender cómo se organiza y almacena la información en distintos tipos de sistemas. A partir de situaciones sencillas y cercanas a la realidad, analizarás la información que se gestiona y reflexionarás sobre la forma más adecuada de estructurarla, reconociendo la necesidad de utilizar una base de datos en lugar de otros medios de almacenamiento.

Durante el desarrollo de la actividad, trabajarás de manera individual y colaborativa para identificar los elementos fundamentales de una base de datos, como datos, registros y tablas, así como para reconocer las características principales de los diferentes tipos de bases de datos, especialmente las relacionales y no relacionales. Además, compararás sus usos, ventajas y limitaciones según el contexto del problema planteado.

Esta experiencia te permitirá apropiarte de los conceptos esenciales sobre qué es una base de datos, para qué se utiliza y cómo los distintos tipos de bases de datos responden a necesidades específicas de los sistemas de información, sentando una base sólida para el aprendizaje posterior del diseño de bases de datos y el uso del lenguaje SQL.

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 con explicación de conceptos clave:
 - Modelo Entidad–Relación (MER).
 - Claves primarias y foráneas.
 - Relaciones uno a uno, uno a muchos y muchos a muchos.
 - Sentencias SQL básicas.
 - 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:

- **8 horas**
 - Explicación inicial y modelado guiado con el instructor.
 - Desarrollo individual de un diseño de base de datos.
 - Trabajo colaborativo para mejora e implementación del modelo.
 - Retroalimentación grupal y cierre con conclusiones.

3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

¡Es momento de aplicar los conocimientos adquiridos en un contexto más cercano a la realidad de los sistemas de información! En esta actividad, transferirás lo aprendido sobre **diseño de bases de datos relacionales y SQL básico** al desarrollo de una **solución de base de datos funcional**, orientada a resolver una necesidad concreta del entorno académico, administrativo o comercial.

Diseñarás una base de datos relacional completa a partir de un caso propuesto, identificando correctamente **entidades, atributos, relaciones y claves**, y aplicando principios básicos de **normalización** para evitar redundancia y asegurar la integridad de la información. Posteriormente, implementarás el diseño utilizando **sentencias SQL**, creando las tablas, estableciendo relaciones y realizando consultas básicas para validar el funcionamiento de la base de datos.



Trabajarás en equipo para planificar, diseñar, implementar y probar la solución, transfiriendo los conocimientos adquiridos a un escenario más complejo y realista. Esta experiencia te permitirá comprender la importancia del diseño adecuado de bases de datos y del uso de SQL como herramienta fundamental en el desarrollo de software.

Ambiente requerido:

- Aula presencial o virtual con acceso a computadores y conexión a internet.
 - Espacios para trabajo colaborativo (mesas grupales o salas virtuales en plataformas como **Zoom o Microsoft Teams**).
 - Gestor de bases de datos (**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 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:

- Guía con ejemplos de proyectos simples de bases de datos relacionales (PDF o presentación), como:
 - Sistema de gestión de estudiantes.
 - Base de datos para una biblioteca.
 - Sistema básico de ventas o inventario.



- Documentación básica de SQL:
 - Creación de tablas.
 - Definición de claves primarias y foráneas.
 - Consultas SQL básicas (SELECT, INSERT, UPDATE, DELETE).
 - 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:

- Diagrama entidad–relación final del proyecto.
 - Script SQL completo de creación de la base de datos.
 - Consultas SQL básicas ejecutadas correctamente.
 - Presentación breve (3 a 5 minutos) del proyecto, explicando el diseño y funcionamiento de la base de datos.
-

Instrumentos de evaluación:

- **Rúbrica de evaluación:**
 - Diseño del modelo de datos (40%):
 - ¿El modelo representa correctamente la información y sus relaciones?
 - Implementación en SQL (30%):
 - ¿Las sentencias SQL están correctamente aplicadas y funcionan?
 - Consistencia e integridad de la base de datos (20%):
 - ¿Las tablas y relaciones aseguran integridad de los datos?
 - Trabajo en equipo y presentación (10%):
 - ¿Hubo colaboración y claridad en la explicación del proyecto?
- **Lista de chequeo:**
Verificación de entrega de evidencias (modelo, scripts y presentación).



Duración de la actividad:

- **4 horas**
 - 30 minutos: Introducción al proyecto y conformación de equipos.
 - 2 horas: Diseño del modelo entidad–relación e implementación en SQL.
 - 1 hora: Pruebas, ajustes finales y preparación de la presentación.
 - 30 minutos: Presentación de los proyectos y reflexión grupal guiada por el instructor.

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 en sistemas de información	Lluvia de ideas, preguntas reflexivas y construcción de mapa conceptual sobre bases de datos	Mapa conceptual grupal, participación en discusión	Relación clara entre la información del entorno y los conceptos básicos de bases de datos relacionales	Observación directa, rúbrica de 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)					



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 (si aplica): CONSTRUCCIÓN DE SOFTWARE INTEGRADOR DE TECNOLOGÍAS ORIENTADAS A SERVICIOS
- Fase del Proyecto (si aplica):Análisis
- Actividad de Proyecto Formativo (si aplica):Desarrollo de procesos lógicos
- Competencia: Especificación de requisitos del software.
- Resultados de Aprendizaje: Validar el informe de requisitos de acuerdo con las necesidades del cliente
- Duración de la Guía de Aprendizaje (horas): 2160

2. PRESENTACIÓN

En el desarrollo de software profesional, no basta con que una aplicación funcione correctamente; es fundamental que esté diseñada sobre una arquitectura que permita su crecimiento, mantenimiento y escalabilidad. La arquitectura de software define la estructura del sistema, la manera en que sus componentes interactúan y cómo se gestionan responsabilidades, dependencias y flujos de información.

Con este propósito, la presente guía se centra en el estudio de modelos arquitectónicos ampliamente utilizados en la industria, como la Arquitectura en Capas, la Arquitectura Hexagonal y el enfoque de Microservicios. A lo largo del proceso formativo, se analizarán sus principios, ventajas y escenarios de aplicación, permitiendo comprender cómo cada estilo arquitectónico responde a necesidades específicas de escalabilidad, desacoplamiento y



organización del sistema. El objetivo es desarrollar una visión estructural del software, fortaleciendo la capacidad de diseñar soluciones robustas, mantenibles y alineadas con estándares empresariales..

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

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

3.1 Actividades de reflexión inicial:

Descripción de la actividad:

En el desarrollo de sistemas empresariales, muchas aplicaciones comienzan siendo pequeñas y funcionales, pero con el tiempo crecen en complejidad, usuarios y funcionalidades. Cuando una aplicación no cuenta con una arquitectura bien definida desde el inicio, puede volverse difícil de mantener, escalar o adaptar a nuevos requerimientos. La arquitectura de software no es solo una decisión técnica, sino una estrategia que impacta directamente la calidad, sostenibilidad y evolución del sistema.

A partir de esta idea, se propone iniciar un conversatorio en el que los aprendices reflexionen sobre las siguientes preguntas:

- ¿Qué problemas puede presentar un sistema que fue desarrollado sin una arquitectura clara?
- ¿Por qué una aplicación que funciona hoy podría no ser sostenible en el futuro?
- ¿En qué situaciones sería suficiente una arquitectura en capas y cuándo podría ser necesario migrar hacia arquitectura hexagonal o microservicios?
- ¿Cómo influye la arquitectura en la escalabilidad, mantenimiento y trabajo en equipo?

Cada aprendiz deberá aportar su análisis desde la experiencia previa en desarrollo, identificando la importancia de diseñar la estructura del sistema antes de escribir código. El objetivo es comprender que la arquitectura es la base estratégica sobre la cual se construyen soluciones tecnológicas robustas y preparadas para evolucionar.

.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.

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

Descripción de la actividad:

De manera individual, cada aprendiz responderá una serie de preguntas orientadas a diagnosticar sus conocimientos previos sobre arquitectura de software, tales como: ¿qué es arquitectura de software?, ¿qué diferencias existen entre diseño y arquitectura?, ¿qué es una arquitectura en capas?, ¿qué entiende por arquitectura hexagonal?, ¿qué caracteriza a los microservicios?, ¿qué ventajas y desventajas identifica en cada enfoque?

Posteriormente, los participantes se organizarán en grupos de trabajo colaborativo. El instructor facilitará la dinámica de la “rueda de la arquitectura”, la cual deberá ser graficada en una hoja de papelógrafo. Cada integrante del grupo escribirá en un segmento de la rueda un concepto clave relacionado con la temática, por ejemplo: capas, dominio, puertos y adaptadores, desacoplamiento, escalabilidad, API, contenedores, independencia de base de datos, comunicación entre servicios, entre otros. Cada concepto deberá ser definido de manera técnica y contextualizado dentro de alguno de los estilos arquitectónicos estudiados.

Finalmente, cada grupo sintetizará los conceptos incluidos en su rueda, estableciendo relaciones entre ellos y explicando cómo se articulan dentro de una arquitectura en capas, hexagonal o basada en microservicios. Los grupos expondrán sus conclusiones al resto del curso, y con la orientación del instructor se construirán conclusiones generales que permitan identificar los conocimientos previos y los aspectos que deberán fortalecerse durante el proceso formativo.



Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: N/A

Material de apoyo: N/A

Duración de la actividad: 1 horas.

3.3 Actividades de apropiación:

Descripción de la actividad:

Proyecto Integrador: Evolución de una Aplicación desde Arquitectura en Capas hacia Microservicios

Descripción de la actividad

Los aprendices desarrollarán una aplicación empresarial que inicialmente será implementada bajo una **arquitectura en capas**, y posteriormente será transformada hacia un enfoque basado en **microservicios**, con el fin de comprender de manera práctica las diferencias estructurales, técnicas y organizacionales entre ambos estilos arquitectónicos.

Fase 1: Implementación con Arquitectura en Capas

Se desarrollará una aplicación monolítica, por ejemplo:

Sistema de Gestión de Pedidos (puede adaptarse a inventario, biblioteca, reservas, etc.).

Requisitos mínimos:

- Gestión de usuarios
- Gestión de productos



- Gestión de pedidos
- Persistencia en base de datos
- API REST

Estructura obligatoria:

- Capa de presentación (Controllers)
- Capa de negocio (Services)
- Capa de acceso a datos (Repositories/DAO)
- Capa de modelo (Entities/Domain)

Actividades técnicas:

- Definir responsabilidades claras por capa
- Implementar reglas de negocio únicamente en la capa de servicios
- Evitar dependencias cruzadas indebidas
- Documentar la arquitectura mediante un diagrama de capas
- Identificar ventajas y limitaciones del monolito

Al finalizar esta fase, los aprendices deberán analizar:

- ¿Qué tan acoplados están los módulos?
- ¿Qué pasaría si el módulo de pedidos requiere escalar independientemente?
- ¿Cómo impacta el crecimiento del sistema en el mantenimiento?

Fase 2: Transición hacia Microservicios

A partir del sistema desarrollado, se realizará una descomposición arquitectónica.

Actividades clave:

1. Identificar los bounded contexts o dominios funcionales (usuarios, productos, pedidos).
2. Separar el monolito en servicios independientes:
 - Microservicio de usuarios
 - Microservicio de productos
 - Microservicio de pedidos
3. Implementar comunicación entre servicios (REST).
4. Asignar base de datos independiente por servicio.
5. Implementar un API Gateway (conceptual o técnico).
6. Documentar la nueva arquitectura mediante diagrama de microservicios.



Análisis Comparativo

Los aprendices deberán elaborar un informe técnico donde comparen:

- Acoplamiento
- Escalabilidad
- Despliegue
- Complejidad operativa
- Mantenibilidad
- Gestión de transacciones

Producto Final

- Código funcional en ambas arquitecturas
- Diagramas arquitectónicos
- Documento comparativo técnico
- Exposición donde justifiquen cuándo usar arquitectura en capas y cuándo microservicios

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listade Chequeo

Duración de la actividad: 1058 horas.



3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad:

Después de haber desarrollado e implementado una aplicación bajo arquitectura en capas, ya cuentas con una base sólida en separación de responsabilidades, organización estructural del código y definición clara de reglas de negocio. Estos fundamentos son el punto de partida ideal para dar el siguiente paso: comprender cómo evoluciona un sistema cuando las necesidades de escalabilidad, mantenimiento y despliegue independiente comienzan a crecer.

A partir de ahora iniciaremos la transición hacia una arquitectura basada en microservicios, donde el sistema dejará de ser un monolito para convertirse en un conjunto de servicios autónomos, comunicados mediante APIs y con responsabilidades claramente delimitadas. Esta nueva etapa te permitirá aplicar lo aprendido sobre diseño arquitectónico y, al mismo tiempo, analizar decisiones técnicas reales relacionadas con desacoplamiento, independencia de bases de datos, comunicación entre servicios y gestión de complejidad en entornos distribuidos.

Ambiente requerido: Ambiente de aprendizaje (ambiente 6 – 11)

Estrategias o técnicas didácticas activas: Aprendizaje Basado en Problemas

Materiales de formación: Documento de fundamentos de programación (Word)

Material de apoyo: Diapositivas fundamentos de programación

Evidencias de aprendizaje: Evidencias de Producto: Taller de algoritmos

Instrumentos de evaluación: Listado de Chequeo

Duración de la actividad: 1150 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
Analisis	Desarrollo de procesos lógicos	Desarrollo de algoritmos	Evidencias de Producto: Taller de algoritmos	Realiza algoritmos utilizando variables, constantes, condicionales, ciclos, bucles,	EVIDENCIA: DESEMPEÑO conocimiento Técnica de Valoración:



				Contadores, acumuladores, pseudocódigo.	Observación Directa Instrumento de Evaluación: Listade Chequeo
--	--	--	--	--	--

5. GLOSARIO DE TÉRMINOS

- Aplicación informática: Una aplicación es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas.
- ACTIVIDAD FÍSICA: actividad física es "cualquier movimiento corporal producido por los músculos esqueléticos que produzca un gasto energético mayor al existente en reposo.
- EXPRESIÓN CORPORAL: El cuerpo humano no solamente se mueve, también se comunica. En este sentido, se habla de la expresión corporal como el conjunto de maneras que tiene nuestro organismo de transmitir sentimientos y emociones.
- Los coreógrafos y bailarines tienen que ejercitar la capacidad de sentir el movimiento del cuerpo para que el espectador perciba igualmente unas emociones.
- El cuerpo humano puede adquirir su propio lenguaje, con un ritmo y con una intensidad. Y el resultado es la expresión corporal.
- Factores de riesgo ergonómico en el trabajo: Los factores de riesgo osteomuscular

en el trabajo están en general bien identificados, relacionándose con la manipulación de cargas, las malas posturas de trabajo, los movimientos forzados, los movimientos repetidos, los movimientos enérgicos, la presión mecánica directa sobre los tejidos corporales o las vibraciones, por citar algunos de las situaciones de exposición más comunes.

-
- Higiene Laboral: Conjunto de normas y procedimientos tendientes a la protección de la integridad física y mental del trabajador, preservándolo de los riesgos de salud inherentes a las tareas a su cargo y al ambiente físico donde se ejecutan.
- Higiene Postural: es el conjunto de normas, cuyo objetivo es mantener la correcta posición del cuerpo, en quietud o en movimiento y así evitar posibles lesiones aprendiendo a pro teger princ



ipalm ente la co lum na vertebral, al realizar las actividades diarias, evitando que se presenten dolores y disminuyendo el riesgo.

- desarrollo de actividades y medidas necesarias para prevenir los riesgos
- COREOGRAFIA: Es el arte de crear estructuras en las que suceden movimientos. La estructura de movimientos puede ser considerada como la coreografía.

6. REFERENTES BIBLIOGRÁFICOS

- Lumowell-fitness-es. Entrenamiento Funcional. Recuperado de:
<https://www.youtube.com/watch?v=XsDJr-qDoG4>.
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/798/el-shell-korn.-manual-de-usuario-y-programador>
- <http://www.digitaliapublishing.com.bdigital.sena.edu.co/a/39320/manual-de-citas-y-referencias-bibliograficas---latino--apa--chicago--ieee--mla-y-vancouver--2a-ed.->
- Instituto Nacional de Seguridad e Higiene en el Trabajo. Anuario de estadísticas laborales y de asuntos sociales, 2006. Disponible en:
http://www.mtas.es/insht/statistics/est_anuar.htm.
- Ergonomía en oficinas. Manual de conceptos fundamentales y recomendaciones prácticas. Recuperado de:
<https://www.achs.cl/portal/trabajadores/Capacitacion/CentrodeFichas/Documents/ergonomia-para-oficinas-conceptos-fundamentales-y-recomendaciones-practicas.pdf>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	CARLOS MANUEL NUÑEZ LOPEZ	INSTRUCTOR	SOFTWARE	25/07/2025

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



	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

Taller de Investigación

Fundamentos necesarios antes de implementar una API con Spring Boot

● BLOQUE 1 – Fundamentos Web y APIs

1. ¿Qué es una API y cuál es su propósito dentro de una arquitectura de software?
 2. ¿Qué diferencia existe entre una API REST y una aplicación web tradicional que devuelve HTML?
 3. ¿Qué es un protocolo de comunicación?
Investigue al menos dos ejemplos y explique cuál se usa en las APIs web.
 4. ¿Qué es HTTP y cuál es su función en el modelo Cliente-Servidor?
 5. ¿Quién transporta los datos en una comunicación web?
Explique el papel del protocolo HTTP y del formato de datos (por ejemplo, JSON).
 6. ¿Qué es una petición HTTP (HTTP Request) y cuáles son sus partes?
 7. ¿Qué es una respuesta HTTP (HTTP Response) y qué información contiene?
 8. Explique los métodos HTTP más usados en APIs REST:
GET, POST, PUT y DELETE.
¿Qué tipo de operación representa cada uno?
 9. ¿Qué son los códigos de estado HTTP?
Investigue y explique los siguientes: 200, 201, 400, 404 y 500.
 10. ¿Qué es JSON y por qué es el formato más utilizado en APIs modernas?
-

● BLOQUE 2 – Fundamentos de Spring y Spring Boot

11. ¿Qué es el Spring Framework y qué problema intenta resolver en el desarrollo empresarial con Java?
12. ¿Qué es Spring Boot y en qué se diferencia del Spring tradicional?
13. ¿Qué es un servidor embebido y cuál utiliza Spring Boot por defecto?
14. ¿Qué es el contenedor de Spring (ApplicationContext) y qué es un Bean?
15. Explique qué es Inversión de Control (IoC) y cómo se aplica en Spring Boot.
16. ¿Qué es Inyección de Dependencias (DI) y qué ventajas ofrece frente a crear objetos con `new`?

● BLOQUE 3 – Anotaciones Fundamentales en Spring Boot

17. Investigue y explique el propósito de las siguientes anotaciones:

- @SpringBootApplication
- @RestController
- @RequestMapping
- @GetMapping
- @PostMapping
- @PathVariable
- @RequestBody

18. ¿Cuál es la diferencia entre @Component, @Service y @Repository?
¿Por qué existen si técnicamente hacen algo similar?

● BLOQUE 4 – Arquitectura en Capas

19. ¿Qué es la arquitectura en capas (Layered Architecture)?
Explique el rol de las siguientes capas en una API:

- Controller
- Service
- Repository
- Model

20. ¿Por qué no es buena práctica colocar toda la lógica dentro del Controller?
Explique los problemas que se generan cuando no se separan responsabilidades.

Enunciado del problema

Una empresa llamada **BikeStore** se dedica a la venta de bicicletas y productos relacionados con el ciclismo.

La empresa desea implementar un sistema que le permita llevar el control de los productos que tiene en su tienda y los movimientos de inventario.

En la tienda se venden diferentes tipos de productos, entre ellos:

- Bicicletas
- Accesorios para bicicletas
- Ropa deportiva para ciclismo

Cada producto que se vende en la tienda debe registrarse con su información básica, como nombre, descripción, precio y cantidad disponible en inventario.

La empresa también quiere poder registrar **los movimientos de entrada y salida de los productos**, ya que constantemente llegan nuevos productos al inventario y también se venden productos a los clientes.

Por ejemplo:

- Cuando llegan productos nuevos al almacén se debe registrar una **entrada**.
- Cuando se vende un producto se debe registrar una **salida**.

Además, la empresa quiere saber:

- Qué productos existen en la tienda
- Qué tipo de producto es (bicicleta, accesorio o ropa)
- Cuántas unidades hay disponibles
- Cuándo se agregaron productos al inventario
- Cuándo se vendieron productos

Cada movimiento debe guardar información como:

- Producto
- Tipo de movimiento (entrada o salida)
- Cantidad
- Fecha del movimiento
- Observación del movimiento

El sistema debe permitir llevar un **historial de todos los movimientos de inventario**, para poder consultar posteriormente cuándo entraron o salieron productos.

Actividades que deben realizar

1 ☐ Análisis del problema

A partir del enunciado deben identificar:

- Entidades del sistema
 - Atributos de cada entidad
 - Relaciones entre entidades
-

2 ☐ Construir el Diagrama Entidad–Relación (ER)

Deben crear un **diagrama ER** que incluya:

- Entidades
- Atributos
- Claves primarias
- Relaciones
- Cardinalidades (1:1, 1:N, N:M)

El diagrama puede realizarse en:

- Papel
-

3 ☐ Definir atributos y tipos de datos

Cada campo de las entidades debe tener definido su **tipo de dato**.

Ejemplo:

Campo	Tipo de dato
id_producto	INT
nombre	VARCHAR(100)

Campo	Tipo de dato
precio	DECIMAL(10,2)
fecha_movimiento	DATE

4 ☐ Construir el Script SQL completo (DDL)

A partir del modelo relacional deben crear el **script de creación de la base de datos**, incluyendo:

- Creación de la base de datos
- Creación de tablas
- Claves primarias

Ejemplo de inicio del script:

```
CREATE DATABASE bikestore;  
  
USE bikestore;
```

Luego crear todas las tablas necesarias.

5 ☐ Insertar datos de prueba (DML)

Para **cada tabla creada** deberán insertar **mínimo 5 registros**.

Ejemplo:

```
INSERT INTO productos (id_producto, nombre, precio)  
VALUES (1, 'Bicicleta Montaña', 1200.00);
```

6 ☐ Realizar consultas (SELECT)

Para **cada tabla creada** deben construir **2 consultas SELECT** diferentes.

Ejemplos:

Consulta 1

Mostrar todos los productos

```
SELECT * FROM productos;
```

Consulta 2

Mostrar productos con precio mayor a 500

```
SELECT nombre, precio  
FROM productos
```

Entregables

Cada grupo debe entregar:

1 ☐ Documento con:

- Análisis del problema
- Diagrama Entidad–Relación
- Explicación de entidades

2 ☐ Script SQL con:

- Creación de base de datos
- Creación de tablas
- Relaciones
- Tipos de datos

3 ☐ Script de datos:

- 5 INSERT por tabla

4 ☐ Consultas:

- 2 SELECT por cada tabla