



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: 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 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)					