



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)					