



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

NOMBRE DEL PROYECTO:

DESARROLLO E IMPLEMENTACIÓN DE MÓDULO PARA LA
GESTIÓN Y OPTIMIZACIÓN DE CAPACITACIONES DE
EMPLEADOS

OPCIÓN QUE CORRESPONDA:

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES

PRESENTA:

ALDRIN MICHEL HERNANDEZ LOPEZ

NOMBRE DEL ASESOR:

MTRO. FELIPE PASCUAL ROSARIO AGUIRRE



Apizaco, Tlaxcala, **05/septiembre/2025**

Oficio No. S.C./OPV/0510/25

ASUNTO: **Liberación de proyecto
para titulación integral**

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	HERNÁNDEZ LÓPEZ ALDRIN MICHEL
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	20370982
NOMBRE DEL PROYECTO:	Desarrollo e implementación de módulo para la gestión y optimización de capacitaciones de empleados
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica.
Pensar para Servir, Servir para Triunfar®*

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera
Expediente
MSP/mrol*



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DEPARTAMENTO DE
SISTEMAS Y COMPUTACIÓN

FELIPE PASCUAL ROSARIO AGUIRRE
ASESOR INTERNO



2025
Año de
**La Mujer
Indígena**

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491
Tel. (241) 4172010 Ext. 115, e-mail: sistemas@apizaco.tecnm.mx
tecnm.mx | apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala **08/octubre/2025**

HERNANDEZ LOPEZ ALDRIN MICHEL
EGRESADO(A) DE LA CARRERA DE INGENIERÍA
EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL 20370982.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **DESARROLLO E IMPLEMENTACIÓN DE MÓDULO PARA LA GESTIÓN Y OPTIMIZACIÓN DE CAPACITACIONES DE EMPLEADOS**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®

CECILIA YARELY ARREGUÍN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

ccp. Archivo

CYAL/rrc



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



AGRADECIMIENTOS

En primer lugar, deseo dedicar este trabajo a mi mamá, quien ha sido el motor más grande en mi vida y la razón principal de mis esfuerzos. Su apoyo incondicional, sus consejos llenos de sabiduría y el amor infinito que siempre me ha brindado han sido la base que me sostuvo en los momentos más difíciles y la inspiración que me impulsó a alcanzar cada uno de mis objetivos. Gracias por creer en mí cuando yo mismo dudaba, por enseñarme el valor de la perseverancia y por demostrarme con tu ejemplo que con esfuerzo y honestidad todo es posible. Este logro no es solo mío, también es tuyo.

De igual forma, agradezco profundamente a mi papá, quien, con su ejemplo de trabajo constante, disciplina y compromiso me ha enseñado a enfrentar la vida con responsabilidad y firmeza. Sus consejos, apoyo y confianza en mis capacidades han sido una guía fundamental a lo largo de mi formación. Gracias por mostrarme que el esfuerzo y la dedicación siempre rinden frutos, y por motivarme a dar lo mejor de mí en cada reto. Este proyecto también es reflejo de los valores que me inculcaste y que hoy forman parte de mi carácter y manera de trabajar.

A mis amigos, mi familia elegida durante la universidad, a quienes cariñosamente llamamos Los Mary Janes. Gracias por cada momento compartido, por las risas interminables, por el apoyo en los días de cansancio, por las conversaciones que se convirtieron en refugio y por recordarme que en este camino académico no estaba solo. Ustedes hicieron que cada etapa de la universidad fuera más llevadera, más divertida y llena de recuerdos que siempre atesoraré. Su compañía ha sido una parte fundamental de mi formación, no solo como estudiante, sino también como persona.

También quiero agradecer a aquellas personas especiales que me acompañaron en diferentes momentos de este proceso. Su apoyo, sus palabras de aliento y su confianza en mis capacidades fueron una motivación extra para no rendirme y seguir adelante incluso cuando el camino parecía complicado. A cada una de ellas, les guardo un profundo agradecimiento porque contribuyeron a que hoy este proyecto sea una realidad.

Finalmente, quiero expresar mi sincera gratitud a mis jefes y mentores en el ámbito laboral: Arturo, Ubaldo y Miguel. Trabajar con ustedes ha sido una experiencia invaluable que ha marcado mi desarrollo profesional y personal. Gracias por brindarme la oportunidad de formar parte de su equipo, por transmitirme conocimientos, por compartir su experiencia y por confiar en mis capacidades. Cada enseñanza recibida se ha convertido en una herramienta que hoy aplico no solo en este proyecto, sino en mi vida profesional. Más allá del trabajo, su guía y ejemplo me han demostrado la importancia de la disciplina, la responsabilidad y la pasión por lo que se hace.

ÍNDICE

I. GENERALIDADES DEL PROYECTO	7
1.1. Introducción	7
1.2. Descripción de la empresa/organización	8
1.3. Problemas a resolver	10
1.4. Justificación	11
1.5. Objetivos	12
1.5.1. Objetivo general	12
1.5.2. Objetivos específicos	12
1.6. Alcances y limitaciones	13
II. MARCO TEÓRICO	14
2.1. Definiciones del seguimiento de capacitaciones	14
2.1.1. Capacitación	14
2.1.2. Aprendizaje organizacional	15
2.1.3. Gestión del conocimiento	16
2.1.4. Evaluación de la capacitación	16
2.1.5. Plataformas LMS (Learning Management Systems)	17
2.1.6. Plataformas de e-Learning	18
2.2. Desarrollo web con JAVA	21
2.2.1. Arquitectura de software	21
2.2.1.1. Arquitectura web	23
2.2.1.2. Modelo MVC (Modelo-Vista-Controlador)	25
2.2.1.3. Arquitectura Modelo 2 (JSP Model 2)	28
2.2.2. JAVA	33
2.2.2.1. Java Server Pages (JSP)	34
2.2.2.2. Java Servlets	34
2.2.2.3. Java Persistence API (JPA)	35
2.2.2.4. Hibernate ORM	35
III. DESARROLLO	37
3.1. Introducción	37
3.2. Metodología de desarrollo	38
3.3. Requisitos funcionales y no funcionales	40

3.4.	Recolección de información	42
3.5.	Especificación de Casos de Uso	43
3.6.	Arquitectura del sistema	45
3.7.	Diseño de Base de Datos	47
3.8.	Historias de usuario	47
3.9.	Desarrollo de la lógica del negocio.....	50
3.10.	Diseño y maquetado de Interfaces de Usuario	51
3.11.	Configuración del entorno de desarrollo.....	56
IV.	ANÁLISIS DE RESULTADOS	77
4.1.	Entidades Mapeadas	78
4.2.	Página de Administrador	81
4.3.	Página de Usuario/Empleado	84
4.4.	Página de Búsqueda de capacitaciones asignadas	89
4.5.	Página de reporte de capacitaciones (Administrador)	93
	Conclusiones.....	97
	Competencias desarrolladas	99
	Propuestas de mejora	100
	Bibliografía.....	101

ÍNDICE DE FIGURAS

Figura 1.....	27
Figura 2.....	28
Figura 3.....	30
Figura 4.....	31
Figura 5.....	38
Figura 6.....	44
Figura 7.....	45
Figura 8.....	47
Figura 9.....	50
Figura 10.....	51
Figura 11.....	52
Figura 12.....	53
Figura 13.....	54
Figura 14.....	55
Figura 15.....	57
Figura 16.....	57
Figura 17.....	57
Figura 18.....	58
Figura 19.....	58
Figura 20.....	59
Figura 21.....	59
Figura 22.....	60
Figura 23.....	60
Figura 24.....	60
Figura 25.....	61
Figura 26.....	61
Figura 27.....	62
Figura 28.....	62
Figura 29.....	63
Figura 30.....	63
Figura 31.....	63
Figura 32.....	64
Figura 33.....	65
Figura 34.....	65
Figura 35.....	66
Figura 36.....	66
Figura 37.....	67
Figura 38.....	67
Figura 39.....	68
Figura 40.....	69
Figura 41.....	69
Figura 42.....	70

Figura 43.....	70
Figura 44.....	71
Figura 45.....	71
Figura 46.....	72
Figura 47.....	72
Figura 48.....	73
Figura 49.....	73
Figura 50.....	74
Figura 51.....	74
Figura 52.....	75
Figura 53.....	75
Figura 54.....	76
Figura 55.....	76
Figura 56.....	77
Figura 57.....	78
Figura 58.....	80
Figura 59.....	82
Figura 60.....	83
Figura 61.....	85
Figura 62.....	86
Figura 63.....	87
Figura 64.....	88
Figura 65.....	88
Figura 66.....	90
Figura 67.....	91
Figura 68.....	91
Figura 69.....	92
Figura 70.....	93
Figura 71.....	94
Figura 72.....	95
Figura 73.....	95
Figura 74.....	96

RESUMEN

Este proyecto de residencia profesional, titulado "Desarrollo e Implementación de un Módulo para la Gestión y Optimización de Capacitaciones de Empleados", aborda la necesidad de automatizar y optimizar los procesos de capacitación en POSCO MPPC Puebla. El sistema actual, basado en procesos manuales para la asignación y seguimiento de cursos de capacitación, presenta áreas de oportunidad significativas, como retrasos en la programación, falta de monitoreo del progreso y dificultades para generar reportes confiables. Para superar estos desafíos, se diseñó e implementó un módulo web basado en tecnologías Java EE (Servlets, JSP, JPA) utilizando la arquitectura Modelo-Vista-Controlador (MVC), que garantiza modularidad, escalabilidad y facilidad de mantenimiento.

Los resultados demuestran una mejora sustancial en la eficiencia operativa, reduciendo la carga administrativa y aumentando la precisión de los datos. Además, el sistema facilita la toma de decisiones basada en datos al consolidar métricas de capacitación, alineando el desarrollo del personal con los objetivos corporativos. Este proyecto resalta la importancia de que con las herramientas digitales se puede gestionar mejor la capacitación del personal. El sistema que desarrollado puede adaptarse fácilmente a otras empresas con necesidades parecidas.

Palabras clave: Gestión de capacitaciones, Java EE, arquitectura MVC, automatización, desarrollo de empleados, decisiones basadas en datos.

I. GENERALIDADES DEL PROYECTO

1.1. Introducción

En la actualidad, la capacitación continua de los empleados es clave para el éxito de las organizaciones, ya que no solo mejora las competencias individuales, sino que también fortalece la productividad y la competitividad empresarial manteniendo un equipo actualizado y eficiente. Sin embargo, en muchas empresas, como POSCO MPPC Puebla, los procesos de gestión de capacitaciones se realizan de manera manual, lo que genera ineficiencias, retrasos en la asignación de cursos, falta de seguimiento y dificultades para medir el impacto real de la formación. Por ello surge la necesidad de implementar soluciones tecnológicas que automaticen y optimicen estos procesos, garantizando una gestión más eficiente y estratégica del conocimiento organizacional.

Este proyecto nace con el objetivo de simplificar y mejorar la gestión de las capacitaciones mediante un sistema digital que automatice tareas, facilite el acceso a la información y permita un mejor control del progreso de los empleados. La herramienta busca ser fácil de usar, tanto para el personal de Recursos Humanos como para los trabajadores, y proporcionar datos claros que ayuden a tomar decisiones más informadas.

Para ello, se emplearán tecnologías Java Web, como Servlets, JSP, JPA y AJAX, bajo una arquitectura Modelo-Vista-Controlador (MVC), que garantiza un diseño modular, escalable y de fácil mantenimiento.

La implementación de este módulo no solo mejora la eficiencia operativa del departamento de Recursos Humanos, sino que también proporcionará a los empleados una herramienta para consultar sus cursos asignados y monitorear su avance. Además, la empresa podrá tomar decisiones basadas en datos, alineando la capacitación con sus objetivos estratégicos.

1.2. Descripción de la empresa/organización

Nombre: POSCO MPPC Puebla.

POSCO MPPC Puebla es la planta pionera de POSCO MPPC en México. Fundada en 2006 como parte del Grupo POSCO líder mundial en la producción de acero, con operaciones en más de 53 países, en nuestro país contamos con una red de cuatro plantas estratégicas: Puebla, San Luis Potosí, Celaya y Aguascalientes. La sede de Puebla abastece principalmente a las industrias automotriz, de electrodomésticos y de la construcción, ofreciendo acero laminado de alta calidad respaldado por certificaciones internacionales como IATF 16949 e ISO 14000.

Domicilio

Carretera 190, Km 0.550, Col. Santa Ana Xalmimilulco,

Municipio de Huejotzingo, Puebla, México.

C.P. 74169

Sitio web

<https://poscomppc.com.mx/>

Misión

“En POSCO MPPC nos dedicamos al suministro de acero laminado para la industria, nuestra misión es cumplir con los requisitos de nuestros clientes y partes interesadas, mediante la mejora continua en todos nuestros procesos, identificando nuestros riesgos y oportunidades.”

Visión

“Ser empresa líder elegida por nuestros clientes, ofreciendo calidad en nuestros procesos y servicios, satisfaciendo así sus necesidades, basada en la confianza y el apoyo de sus Colaboradores y contribuyendo al progreso de la comunidad y su entorno.”

Valores

- **Clientes y partes interesadas:** En todas las áreas escuchamos la voz del cliente cumpliendo sus necesidades y expectativas.
- **Ética:** Actuar con responsabilidad y transparencia en cada una de nuestras áreas.

- **Capital Humano:** Valoramos a nuestro personal, cuidando de los derechos de nuestros empleados, demostrando sus habilidades y capacidades, alcanzando un mejor potencial.

Giro de la empresa

El giro de POSCO MPPC, filial de POSCO, es la venta y procesamiento de bobinas, láminas y cintas de acero para diversos sectores, destacando la industria automotriz como su principal mercado.

Área donde se realizó el proyecto

Departamento de sistemas

El proyecto fue desarrollado en el Departamento de sistemas en colaboración y con el apoyo del departamento de recursos humanos, el cuál es el área responsable del seguimiento de capacitaciones para empleados.

1.3. Problemas a resolver

En POSCO MPPC S.A. de C.V., la capacitación de empleados enfrenta serias limitaciones que impiden maximizar su impacto en la productividad y la calidad. Actualmente, la asignación de cursos se lleva a cabo mediante procesos manuales que retrasan tanto la definición de calendarios como la notificación oportuna a los colaboradores. Esta metodología obliga al departamento de Recursos Humanos a coordinar individualmente fechas y participantes, lo que a menudo provoca choques de horario o deja algunos cursos sin asignar.

Además, el seguimiento de las metas formativas está fragmentado: cada planta mantiene sus propios registros de avances, sin indicadores centralizados que permitan identificar a tiempo las brechas de conocimiento o los incumplimientos de objetivos. Esta dispersión informativa impide tener una visión global del estado de la capacitación y retrasa la detección de áreas críticas que requieren refuerzo inmediato.

La ausencia de reportes estructurados y de métricas confiables agrava aún más la situación. Sin datos consolidados, la dirección carece de herramientas para medir el retorno de la inversión en formación y ajustar contenidos o metodologías de manera ágil. Al mismo tiempo, la generación manual de informes convierte en una tarea tediosa la obtención de información relevante para la toma de decisiones, por lo que cualquier análisis profundo de resultados queda prácticamente fuera de alcance.

Paralelamente, Recursos Humanos dedica decenas de horas al recopilar, depurar y unificar información proveniente de múltiples fuentes, con el riesgo constante de errores de transcripción y versiones desactualizadas en las hojas de cálculo compartidas. La falta de un canal institucional único para notificaciones y recordatorios contribuye a que más del 40 % de los empleados retrase o incluso omita la confirmación y el seguimiento de sus capacitaciones, afectando la trazabilidad y la motivación continua.

Por último, no existen mecanismos automáticos para validar datos, detectar duplicados o generar alertas de vencimiento de metas formativas, lo que impide la generación de reportes dinámicos y priva a la empresa de una herramienta proactiva para el análisis de tendencias de desempeño tanto individual, así como, por área.

En conjunto, estas deficiencias limitan la capacidad de POSCO MPPC para alinear su estrategia de capacitación con los objetivos organizacionales, subrayando la necesidad de un sistema centralizado que transforme estos procesos reactivos en una gestión eficiente y estratégica.

1.4. Justificación

En la industria, la capacitación continua es esencial para mantener la calidad, la productividad y la seguridad. Sin embargo, en POSCO MPPC S.A. de C.V. los procesos manuales y descentralizados de asignación y seguimiento de cursos generan duplicidad de tareas, errores de datos y demoras en la toma de decisiones.

Este proyecto propone un módulo centralizado que automatice la asignación de capacitaciones, valide automáticamente la integridad de la información y genere reportes claros para monitorear el avance individual y grupal. Al digitalizar estos procesos, se reducirán las horas anuales que Recursos Humanos dedica a tareas administrativas y se mejorará la visibilidad de los indicadores formativos.

Con datos fiables y accesibles al instante, la dirección podrá medir el retorno de la inversión en formación, ajustar rápidamente los contenidos y alinear el desarrollo de competencias con los objetivos estratégicos de la empresa. En última instancia, este módulo mejorará la eficiencia operativa, facilitará la toma de decisiones basadas en datos y permitirá un seguimiento estratégico y continuo de los programas de capacitación.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar e implementar un módulo centralizado de gestión de capacitaciones que automatice la asignación de cursos, genere dashboards interactivos y reporte el progreso individual y grupal, para optimizar la eficiencia operativa y alinear la formación del personal con los objetivos estratégicos de la empresa.

1.5.2. Objetivos específicos

- Desarrollar una plataforma centralizada de gestión de capacitaciones que unifique la asignación de cursos y permita la carga masiva de información, optimizando la actualización de estados y registros.
- Desarrollar e implementar el módulo de gestión de capacitaciones utilizando tecnologías Java Web (Servlets, JSP, JPA, AJAX y Bootstrap), asegurando una arquitectura MVC y buenas prácticas de codificación.
- Implementar un módulo de visualización y seguimiento que permita a los empleados acceder a sus cursos asignados, consultar progreso individual y visualizar métricas de avance.
- Automatizar los procesos de validación de datos para garantizar la integridad de la información, evitando duplicados y errores en la asignación de cursos.
- Definir roles y permisos de usuario para asegurar la seguridad y el control de acceso a la información.
- Desarrollar un sistema de generación de reportes estructurados que permita analizar el cumplimiento de metas formativas a nivel individual y organizacional, facilitando la toma de decisiones basada en datos.
- Documentar la solución y capacitar al personal de TI para garantizar su mantenimiento, escalabilidad y evolución futura.
- Realizar pruebas funcionales y de rendimiento para asegurar que el sistema cumpla con los requerimientos establecidos y se adapte a las necesidades operativas de la empresa.

1.6. Alcances y limitaciones

El proyecto automatizará la asignación de cursos y el seguimiento de metas formativas en POSCO MPPC mediante el desarrollo de un sistema centralizado. Implementará un módulo de asignación capaz de procesar cargas masivas desde archivos ODS o XLSX, y un dashboard interactivo para que Recursos Humanos y los colaboradores visualicen el avance individual y grupal. Además, generará reportes estructurados exportables en PDF o Excel para facilitar el análisis de resultados.

Quedan fuera de este alcance la creación o edición de contenidos educativos y la integración con plataformas externas de e-learning (LMS). De este modo, el proyecto podrá concentrarse en optimizar los procesos operativos de capacitación sin abordar la producción de materiales didácticos.

II. MARCO TEÓRICO

La capacitación continua es esencial para enfrentar los desafíos que imponen los cambios sociales, tecnológicos, económicos, políticos, ambientales, demográficos y legales. En POSCO MPPC S.A. de C.V., los procesos actuales de capacitación presentan serias limitaciones debido a su naturaleza manual y fragmentada, lo que genera retrasos en la asignación de cursos, falta de seguimiento oportuno y dificultades en la toma de decisiones estratégicas.

El aprendizaje organizacional requiere, por tanto, estructuras y tecnologías que faciliten la asignación automatizada, la centralización de la información y la generación dinámica de reportes. Implementar plataformas digitales que permitan medir y evaluar el impacto formativo es clave para alinear la capacitación con los objetivos estratégicos de la empresa, garantizando eficiencia operativa y fortaleciendo la resiliencia organizativa.

Este marco teórico abordará conceptos clave sobre capacitación, aprendizaje organizacional, modelos contemporáneos de gestión del conocimiento y métricas para evaluar la eficacia de los procesos formativos.

2.1. Definiciones del seguimiento de capacitaciones

2.1.1. Capacitación

Según Chiavenato (2017), la capacitación es un proceso educativo breve, sistemático y organizado, mediante el cual los individuos adquieren conocimientos específicos, desarrollan habilidades y competencias, orientados a cumplir objetivos claramente definidos. Este proceso incluye la transferencia directa de conocimientos relacionados con el trabajo, la adquisición de actitudes positivas hacia la organización y el entorno laboral, y la potenciación de habilidades prácticas y competencias esenciales.

Por su parte, Siliceo (2009) enfatiza que la capacitación es crucial para alcanzar plenamente los objetivos generales de una empresa, ya que proporciona personal debidamente preparado y capacitado. Este enfoque educativo busca cubrir las necesidades presentes y anticipar futuras demandas de habilidades y competencias en los colaboradores, siempre alineadas con las necesidades reales y estratégicas de la organización.

Dessler (2015) considera la capacitación como el acto de proporcionar a empleados nuevos y antiguos las habilidades necesarias para ejecutar eficazmente su trabajo. Esto incluye tanto la formación inicial a nuevos colaboradores, como la actualización continua de habilidades en empleados experimentados, mediante programas prácticos y específicos orientados a mejorar su desempeño laboral concreto.

Finalmente, Alles (2012) define la capacitación como actividades estructuradas en cursos con objetivos claramente establecidos, realizados bajo modalidades presenciales o virtuales, con fechas y horarios definidos. Esta perspectiva subraya la importancia de planificar, organizar y evaluar sistemáticamente la transferencia de conocimientos y habilidades, asegurando así la efectividad y pertinencia del proceso formativo.

2.1.2. Aprendizaje organizacional

El concepto de aprendizaje organizacional es definido ampliamente por Peter Senge (2005), quien lo describe como un proceso mediante el cual las organizaciones adquieren nuevos conocimientos, desarrollan competencias y habilidades que conducen a cambios significativos en sus comportamientos y procesos internos. Este aprendizaje facilita a las organizaciones adaptarse, innovar y sobrevivir eficazmente en contextos dinámicos y altamente cambiantes. Según Senge, para que este aprendizaje sea efectivo, se requieren organizaciones abiertas, capaces de reflexionar sobre sus prácticas, desafiar paradigmas establecidos y promover la colaboración y la creatividad entre sus miembros.

Asimismo, Argyris y Schön (1978) conceptualizan el aprendizaje organizacional como la detección y corrección sistemática de errores a través de ciclos continuos de retroalimentación y reflexión. Para estos autores, el aprendizaje organizacional no es solo la acumulación de conocimientos, sino la capacidad de las organizaciones para modificar activamente sus teorías de acción y marcos operativos, promoviendo así un aprendizaje profundo que garantiza adaptabilidad y mejora continua en respuesta a cambios internos y externos.

2.1.3. Gestión del conocimiento

La gestión del conocimiento es un proceso estructurado mediante el cual las organizaciones crean, almacenan, comparten y aplican conocimientos, con el objetivo de optimizar sus procesos internos, mejorar la toma de decisiones y fortalecer su capacidad innovadora (Nonaka & Takeuchi, 1999). Estos autores destacan que la creación de conocimiento organizacional se produce a través de una interacción dinámica entre conocimiento tácito y explícito. El conocimiento tácito se refiere a experiencias personales, intuiciones y habilidades que son difíciles de expresar formalmente, mientras que el conocimiento explícito es sistemático y fácil de transmitir mediante documentación, manuales o plataformas digitales.

Nonaka y Takeuchi (1999) proponen el modelo SECI, compuesto por cuatro fases: Socialización, Externalización, Combinación e Internalización. La socialización implica compartir experiencias y conocimientos tácitos mediante interacciones directas entre individuos. La externalización convierte el conocimiento tácito en explícito mediante diálogos estructurados, documentación formal o tecnologías digitales. La combinación implica la integración sistemática del conocimiento explícito, utilizando bases de datos, plataformas digitales o repositorios organizacionales. Finalmente, la internalización transforma el conocimiento explícito en conocimiento tácito a través del aprendizaje práctico y experiencias laborales concretas.

Adicionalmente, Davenport y Prusak (2001) enfatizan que la gestión efectiva del conocimiento requiere un entorno organizacional favorable, caracterizado por una cultura de colaboración, confianza y apertura. La gestión del conocimiento no solo implica tecnologías avanzadas, sino también estrategias claras, liderazgo comprometido y procesos organizacionales alineados con objetivos estratégicos definidos.

2.1.4. Evaluación de la capacitación

La evaluación de la capacitación es un proceso fundamental que permite determinar la efectividad y el valor real de los programas formativos implementados en una organización. Según Kirkpatrick y Kirkpatrick (2007), este proceso comprende cuatro niveles de evaluación claramente definidos: reacción, aprendizaje, comportamiento y resultados. La evaluación de reacción mide cómo perciben los participantes la formación recibida,

abordando aspectos como satisfacción y relevancia del contenido. El nivel de aprendizaje evalúa el grado en que los participantes han adquirido conocimientos, habilidades y actitudes previstas en los objetivos formativos.

El nivel de comportamiento analiza hasta qué punto los conocimientos y habilidades adquiridos se aplican realmente en el entorno laboral, identificando cambios en las prácticas laborales y mejoras en el desempeño de los empleados. Finalmente, el nivel de resultados examina el impacto general del programa de capacitación sobre los objetivos estratégicos y operativos de la organización, incluyendo mejoras en productividad, calidad, reducción de errores y aumento en la rentabilidad.

Complementariamente, Phillips y Phillips (2008) introducen un quinto nivel denominado retorno de la inversión (ROI), el cual mide específicamente la relación costo-beneficio del programa formativo, cuantificando monetariamente los beneficios obtenidos frente a los recursos invertidos en capacitación. Este nivel de evaluación proporciona datos esenciales para justificar futuras inversiones en programas formativos y optimizar recursos disponibles.

2.1.5. Plataformas LMS (Learning Management Systems)

Las plataformas LMS (*Learning Management Systems*, o sistemas de gestión del aprendizaje) son herramientas tecnológicas diseñadas específicamente para gestionar, administrar y entregar contenidos formativos de manera eficiente, permitiendo monitorear el progreso y evaluar los resultados del proceso educativo de los usuarios (García Peñalvo, Estado actual de los sistemas e-learning., 2005). Estos sistemas facilitan la creación y organización de cursos, el seguimiento individualizado del aprendizaje y el análisis de datos educativos, constituyendo así un elemento esencial para la capacitación continua en las organizaciones modernas.

Según Rosenberg (2001), una plataforma LMS típica cuenta con funcionalidades clave que permiten administrar usuarios, gestionar contenidos educativos digitales, distribuir material formativo, implementar evaluaciones automatizadas, dar seguimiento al desempeño de los participantes y generar informes detallados para la toma de decisiones estratégicas en materia educativa.

Entre las características principales de los LMS destacan:

- **Gestión centralizada de contenidos educativos:** Permiten almacenar y distribuir una gran variedad de formatos, tales como videos, documentos PDF, presentaciones multimedia y evaluaciones interactivas (García Peñalvo, 2005).
- **Seguimiento y evaluación del aprendizaje:** Las plataformas LMS posibilitan monitorear en tiempo real el avance de los estudiantes, registrar los resultados obtenidos en evaluaciones y facilitar la retroalimentación inmediata, lo que incrementa la eficacia y eficiencia del aprendizaje organizacional (Ghirardini, 2011).
- **Accesibilidad y flexibilidad:** Los sistemas LMS están basados principalmente en tecnologías web, permitiendo acceso desde cualquier lugar y dispositivo con conexión a internet, favoreciendo así modelos flexibles de capacitación, como la formación a distancia, formación híbrida o e-learning (Moodle, 2023).
- **Generación automatizada de reportes y analítica educativa:** Las plataformas LMS ofrecen herramientas avanzadas para generar informes automáticos y dashboards interactivos, facilitando el análisis de indicadores clave como tasas de finalización, rendimiento académico y satisfacción del usuario, esenciales para la mejora continua del proceso formativo (Chamilo, 2023).
- **Compatibilidad con estándares internacionales:** Las plataformas LMS usualmente cumplen estándares como SCORM o xAPI (Tin Can), garantizando la interoperabilidad con distintos proveedores de contenidos educativos digitales y la integración con otras herramientas tecnológicas (Ghirardini, 2011)

2.1.6. Plataformas de e-Learning

Las plataformas de e-learning constituyen entornos virtuales que integran tecnologías de la información y la comunicación (TIC) para diseñar, distribuir y evaluar experiencias formativas flexibles, interactivas y accesibles. A diferencia de un LMS puramente administrativo, estas plataformas ponen el foco en la experiencia pedagógica, combinando recursos multimedia, herramientas de comunicación y analítica de datos para potenciar el aprendizaje autónomo y colaborativo.

Sangrà, Vlachopoulos y Cabrera (2012) definen el e-learning como una modalidad educativa mediada por tecnologías digitales que “integra procesos educativos con tecnologías digitales, caracterizándose por su flexibilidad espacio-temporal, interactividad y comunicación continua” (p. 150). Estos autores subrayan que el e-learning no es solo un canal de distribución de contenidos, sino un ecosistema que fusiona pedagogía, informática y comunicación para ofrecer aprendizaje personalizado y adaptativo.

Cabero (2006) amplía esta visión al señalar que el e-learning “se sustenta en el uso estratégico de las TIC como medio integrador de texto, imagen, audio y video para generar entornos formativos abiertos, accesibles y centrados en el estudiante”. Destaca, además, la importancia de diseñar materiales instruccionales que aprovechen la interactividad y la multicanalidad para favorecer la construcción activa del conocimiento.

Smowl (2022) describe una plataforma de e-learning como «un entorno virtual que permite gestionar, impartir y monitorizar la formación en línea, ofreciendo herramientas para la creación de contenidos, la evaluación automatizada y el seguimiento detallado del progreso del usuario». Esta definición pone el acento en la capacidad de las plataformas para centralizar procesos y generar métricas que faciliten la toma de decisiones formativas basadas en datos.

CognosOnline (2021) añade que el e-learning facilita “la autogestión del aprendizaje desde cualquier dispositivo electrónico, a través de recursos como vídeos, podcasts, artículos interactivos y cursos auto estructurados”, resaltando la democratización del acceso y la personalización de los itinerarios formativos.

En conjunto, estas definiciones coinciden en que las plataformas de e-learning:

1. **Eliminan barreras de espacio y tiempo**, habilitando el acceso 24/7 desde múltiples dispositivos.
2. **Favorecen la interactividad y el aprendizaje colaborativo** mediante foros, chats y videoconferencias.
3. **Incorporan recursos multimedia** que enriquecen la experiencia educativa y aumentan la retención de conocimientos.

4. **Proporcionan analítica educativa en tiempo real**, permitiendo personalizar rutas formativas y medir el impacto.

Tabla 1.
Diferencia entre Sistema LMS y Plataforma e-learning

Criterio	Sistema LMS	Plataforma e-learning
Propósito principal	Administrar, organizar y formalizar procesos de formación (matriculación, calendarios, calificaciones).	Diseñar, distribuir y vivenciar experiencias educativas flexibles y multimedia.
Alcance	Enfoque institucional: controla cursos, usuarios y registros académicos.	Enfoque experiencial: integra contenidos, interacción y analítica de aprendizaje centrada en el usuario.
Enfoque pedagógico	Predomina la gestión; puede incorporar didáctica, pero su núcleo es administrativo.	Centrado en la pedagogía digital: recursos interactivos, aprendizaje autónomo y colaborativo.
Características clave	Gestión de usuarios, roles, inscripción, seguimiento de evaluaciones, informes.	Recursos multimedia, foros, videoconferencia, adaptabilidad, micro-contenidos, gamificación.
Gestión de contenidos	Repositorios estructurados y versiones controladas (SCORM, xAPI).	Contenidos dinámicos y actualizables orientados a experiencias de aprendizaje.
Analítica y métricas	Reportes estandarizados (notas, finalización, tiempo en curso).	Analítica profunda de participación, rutas personalizadas, engagement y retroalimentación en tiempo real.
Escalabilidad	Alta en número de cursos y usuarios, pero dependiente de estructura jerárquica.	Diseñada para multitudes y actualización continua sin rigidez curricular.
Ejemplos	Moodle, Blackboard Learn, Canvas (cuando se usan como LMS).	Coursera, edX, Udemy, Khan Academy.
Ventajas	Control robusto, trazabilidad legal, integración con ERP/HR.	Experiencia de usuario atractiva, flexibilidad, acceso 24/7, aprendizaje adaptativo.
Limitaciones	Puede ser rígido y centrado en gestión más que en experiencia.	Requiere alto compromiso del estudiante y buen diseño instruccional; depende de conectividad.

2.2. Desarrollo web con JAVA

2.2.1. Arquitectura de software

Clements y colaboradores definen la arquitectura de software como “la estructura organizativa de un sistema de software, incluida la descripción de sus componentes, las conexiones entre ellos, las restricciones y la justificación”. (Clements, y otros, 1994)

- **Estructura organizativa:** Se refiere a cómo se agrupan y organizan los módulos o subsistemas dentro del sistema, pudiendo ir desde componentes de bajo nivel (por ejemplo, clases o paquetes) hasta grandes subsistemas autónomos (por ejemplo, un motor de base de datos)
- **Conexiones:** Incluye los mecanismos de interacción entre componentes, como llamadas a procedimientos, colas de mensajes, eventos o APIs REST, que definen el flujo de información y control en el sistema.
- **Restricciones:** Comprende reglas de diseño que limitan cómo pueden organizarse o comunicarse los componentes (por ejemplo, políticas de seguridad, decisiones de ubicación física o dependencias tecnológicas)
- **Justificación:** Documenta las razones de las decisiones arquitectónicas, permitiendo entender por qué ciertos componentes se estructuran de determinada forma y cómo se soportan los requisitos no funcionales (p. ej., respuesta a fallos, escalabilidad, mantenibilidad).

La arquitectura de software es la estructura de alto nivel que define la organización, interacciones y principios de diseño de un sistema informático. Según Garlan y Shaw (1994), representa "las decisiones de diseño relacionadas con la estructura global y el comportamiento del sistema, yendo más allá de algoritmos y estructuras de datos" 1. Esta disciplina sirve como plano rector que:

- Establece componentes, interfaces y protocolos de comunicación.
- Garantiza el cumplimiento de requisitos funcionales (qué hace el sistema) y no funcionales (escalabilidad, seguridad, rendimiento).

- Facilita la toma de decisiones técnicas y la coordinación entre equipos multidisciplinarios.

Para 2030, se proyecta que el mercado global de arquitectura de software alcance \$5,760 millones de dólares, evidenciando su importancia estratégica en la industria digital. (BBVA, 2025)

Tabla 2.

Comparativa de patrones arquitectónicos

Patrón	Principio Central	Ventajas	Limitaciones	Casos de Uso
Cliente-Servidor	Separación entre consumidor (cliente) y proveedor de servicios (servidor).	Simplicidad, bajo acoplamiento.	Riesgo de cuellos de botella en el servidor.	Aplicaciones bancarias tradicionales.
Tres Capas	División en presentación (UI), lógica de negocio y almacenamiento de datos.	Modularidad, mantenimiento eficiente.	Complejidad en sincronización entre capas.	Plataformas de e-commerce como Amazon.
Microservicios	Sistema como servicios independientes comunicados vía APIs.	Escalabilidad granular, tolerancia a fallos.	Complejidad operativa (monitoreo distribuido).	Netflix, Spotify.
Hexagonal (Puertos y Adaptadores)	Aísla lógica de negocio de infraestructura externa.	Pruebas simplificadas, adaptabilidad tecnológica.	Curva de aprendizaje pronunciada.	Sistemas de alta resiliencia.

MVC (Modelo-Vista-Controlador)	Separación entre datos (Modelo), interfaz (Vista) y coordinación (Controlador).	Reutilización de código, desarrollo paralelo.	Riesgo de "código espagueti" en proyectos grandes.	Frameworks web como Ruby on Rails.
---------------------------------------	---	---	--	------------------------------------

2.2.1.1. Arquitectura web

La arquitectura web es la estructura lógica y técnica que organiza los componentes de un sitio web (páginas, contenidos, funcionalidades) para optimizar su rendimiento, usabilidad y posicionamiento. Según Kiwop (2025), representa un "modelo de pensamiento estratégico que define cómo se conectan y presentan los elementos para maximizar la navegación intuitiva". Su evolución refleja cambios tecnológicos y de comportamiento:

- **Web 1.0 (1990s):** Arquitecturas estáticas, basadas en "brochures digitales" con contenido unidireccional.
- **Web 2.0 (2000s):** Surge la interactividad con blogs y redes sociales, exigiendo estructuras dinámicas para contenido generado por usuarios.
- **Web 3.0 (Actualidad):** Enfoque en experiencias omnicanal, integración con IA y arquitecturas descentralizadas para adaptarse a dispositivos móviles y metaversos.

La arquitectura web es el diseño estructural que describe cómo interactúan los distintos componentes de una aplicación en la red, incluyendo el cliente, el servidor, la base de datos y la infraestructura de red para entregar contenido a través de HTTP/HTTPS (hostinger, 2025). Según ThinkSys, se define como “un marco que describe las relaciones e interacciones entre los componentes de una aplicación web, esenciales para un desarrollo eficiente y una experiencia de usuario fluida” (thinksys, 2025).

Componentes clave

Cliente (Front-End)

Consiste en la capa de presentación que corre en el navegador del usuario, implementada con HTML para la estructura, CSS para el estilo y JavaScript (o frameworks como React, Angular o Vue) para la interactividad y dinamismo de la interfaz.

Backend (Servidor de Aplicaciones)

Núcleo de la lógica de negocio y orquestación de procesos. Se desarrolla en lenguajes como Java (Spring Boot, Jakarta EE), Node.js, .NET o Python. Gestiona la validación de datos, aplica reglas de negocio, coordina la persistencia y comunica con servicios externos.

Servidor Web

Atiende las peticiones HTTP/S entrantes. El servidor web (Apache, Nginx) gestiona conexiones, SSL/TLS y entrega archivos estáticos; enruta solicitudes al servidor de aplicaciones para procesar la lógica de negocio.

Base de Datos

Gestiona la persistencia de información. Puede ser una base relacional (MySQL, PostgreSQL, Oracle) con SQL y transacciones ACID, o una base NoSQL (MongoDB, Cassandra) para datos semiestructurados y escalabilidad horizontal.

Capa de API / Middleware

Actúa como intermediario entre front-end y back-end, exponiendo la lógica de negocio a través de servicios RESTful (JSON/HTTP) o GraphQL. Aquí se validan peticiones, se orquestan flujos y se aplica seguridad y control de acceso.

Balanceador de Carga

Distribuye el tráfico entrante entre varias instancias del servidor de aplicaciones, mejorando la escalabilidad y la tolerancia a fallos. Ejemplos: HAProxy, F5, ELB de AWS.

Caché

Sistema de almacenamiento temporal (Redis, Memcached) que guarda resultados frecuentes (consultas a la base de datos, sesiones de usuario) para reducir latencia y carga en los sistemas de origen.

Content Delivery Network (CDN)

Red distribuida de servidores que entrega activos estáticos (imágenes, scripts, estilos) desde la ubicación geográfica más cercana al usuario, minimizando tiempos de carga y ancho de banda.

Mensajería Middleware

Plataforma de mensajería asíncrona (RabbitMQ, Apache Kafka) que permite comunicación desacoplada entre servicios, soportando patrones pub/sub y colas de procesamiento en segundo plano.

DNS y Seguridad

El DNS traduce nombres de dominio legibles (www.miapp.com) a direcciones IP. La seguridad se refuerza con HTTPS/TLS para cifrar el tráfico, WAF (Web Application Firewall) y firewalls perimetrales para bloquear ataques y proteger la capa de aplicación.

2.2.1.2. Modelo MVC(Modelo-Vista-Controlador)

Según TechTarget “Modelo-Vista-Controlador (MVC) es un patrón de arquitectura de software para implementar interfaces de usuario. Divide una aplicación en tres componentes interconectados Modelo, Vista y Controlador para separar las representaciones internas de la información de las formas en que se presenta y se recibe del usuario.” (TechTarget, 2025)

Por otra parte, GeeksforGeeks dice que el “MVC (Modelo-Vista-Controlador) es un patrón de diseño que separa la aplicación en tres capas lógicas: Modelo, Vista y Controlador. El Modelo corresponde a toda la lógica relacionada con los datos con los que trabaja el usuario. La Vista se encarga de la lógica de la interfaz de usuario. El Controlador actúa como interfaz entre los componentes Modelo y Vista.” (GeeksforGeeks, 2025)

Componentes Principales

Modelo

- Representa datos y lógica de negocio.
- Gestiona acceso a la base de datos o servicios externos.
- Notifica Vistas de cambios en su estado, a menudo mediante eventos u observadores (MDN Web Docs, 2023)

Vista

- Presenta los datos del Modelo al usuario.
- Se encarga exclusivamente de la representación HTML/templates, componentes UI, etc.
- Escucha actualizaciones del Modelo y se redibuja en consecuencia. (GeeksforGeeks, 2025)

Controlador

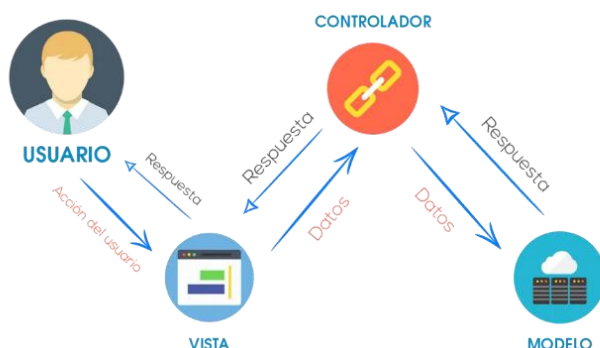
- Interpreta eventos de usuario (clics, formularios, peticiones HTTP).
- Invoca métodos del Modelo para procesar datos y decide qué Vista mostrar.
- Actúa como puente entre Modelo y Vista, evitando que ambos se conozcan directamente. (Tutorialspoint, 2024)

Flujo de Datos

1. **Usuario** realiza una acción en la Vista (por ejemplo, envía un formulario).
2. La **Vista** envía un evento al Controlador.
3. El **Controlador** invoca lógica en el Modelo, actualizando datos o consultándolos.
4. El **Modelo** realiza operaciones (persistencia, cálculos) y notifica el cambio.
5. La **Vista** se actualiza leyendo el Modelo y renderiza la respuesta al usuario

Figura 1

Arquitectura del Patrón Modelo-Vista-Controlador (MVC)



La figura ilustra el funcionamiento del patrón Modelo-Vista-Controlador (MVC) en aplicaciones web. El usuario interactúa con la vista, que envía sus acciones al controlador. Este, a su vez, procesa la solicitud, consulta o actualiza los datos en el modelo y devuelve la respuesta hacia la vista para mostrársela al usuario. Este ciclo asegura la separación de responsabilidades, facilitando la escalabilidad, el mantenimiento y la modularidad del sistema.

Variantes y Patrones Relacionados

- **MVP (Model-View-Presenter):** El Presenter asume parte de la lógica que en MVC estaría en el Controlador y mejora el testeo (MDN Web Docs, 2023)
- **MVVM (Model-View-ViewModel):** Popular en frameworks de UI como Angular o .NET, introduciendo un ViewModel que expone datos del Modelo para la Vista mediante enlaces de datos.
- **HMVC (Hierarchical MVC):** Organiza MVC en múltiples niveles jerárquicos, permitiendo sub-MVC dentro de componentes para mejorar la modularidad.
- **PAC (Presentation-Abstraction-Control):** Similar a MVC pero con componentes más granulares y flujos de control especializados.

Tabla 3.

Ventajas y Desventajas

Ventajas	Desventajas
Separación clara de responsabilidades, facilitando mantenimiento y escalabilidad.	Mayor complejidad inicial y curva de aprendizaje para pequeños proyectos.
Facilita el trabajo en equipo: front-end y back-end pueden avanzar en paralelo.	Requiere más código “plomería” (boilerplate) para enrutar y gestionar flujos.
Permite reutilizar componentes y vistas.	A veces el controlador se convierte en “God Object” con demasiadas responsabilidades.
Fomenta testeo unitario de la lógica de negocio aislada del UI.	Overhead de capas que puede impactar el rendimiento si se abusa del patrón.

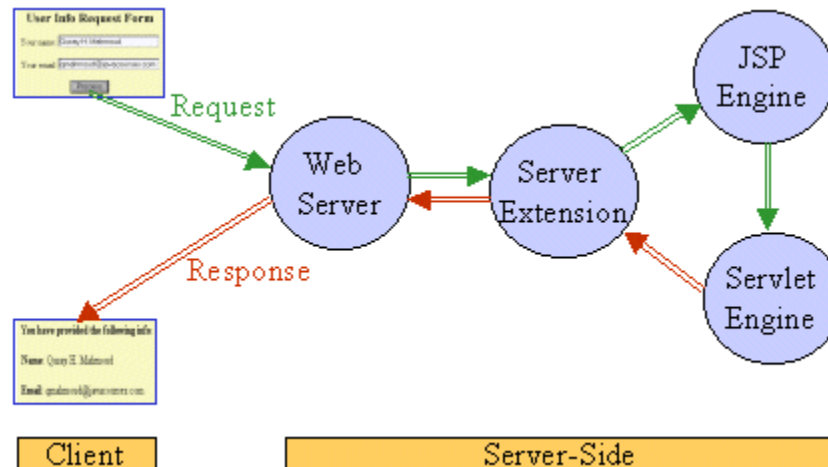
2.2.1.3. Arquitectura Modelo 2 (JSP Model 2)

El JSP Model 2 se define como “un patrón de arquitectura para aplicaciones web Java que separa estrictamente la lógica de control, la lógica de negocio y la presentación, combinando Servlets para el control de flujo y JavaServer Pages para la capa de vista” (Seshadri, 1999).

Según la especificación original de Sun Microsystems, “en Model 2, el servlet actúa como controlador responsable de procesar la petición, crear o rellenar beans y decidir a qué JSP reenviar el flujo, mientras que la JSP se limita a mostrar el resultado” Figura 2. (Oracle, 2025)

Figura 2

Arquitectura de Procesamiento de Servlets y JSP en el Servidor



La figura muestra el flujo de interacción entre un cliente y el servidor durante la ejecución de aplicaciones basadas en Servlets y JSP. El cliente envía una request al Web Server, que la dirige al Server Extension encargado de gestionar la comunicación con los motores de ejecución. Dependiendo del tipo de recurso solicitado, la petición puede ser procesada por el Servlet Engine o el JSP Engine, generando dinámicamente la respuesta. Finalmente, esta se devuelve al cliente como contenido HTML interpretado en el navegador. Este ciclo evidencia cómo se integran los distintos componentes del lado servidor para garantizar aplicaciones dinámicas y eficientes.

Objetivos de la arquitectura

1. Separación de responsabilidades

- Aísla la lógica de negocio en clases Java (POJOs, JavaBeans, entidades JPA).
- Centraliza el flujo de control en un único servlet (Front Controller).
- Mantiene las JSP limpias de cualquier lógica de negocio, usándolas sólo como plantillas de presentación.

2. Colaboración entre roles

- Desarrolladores Java trabajan en servlets, servicios y DAOs.
- Diseñadores web pueden editar JSP sin riesgo de alterar la lógica de negocio.

3. Escalabilidad y mantenibilidad

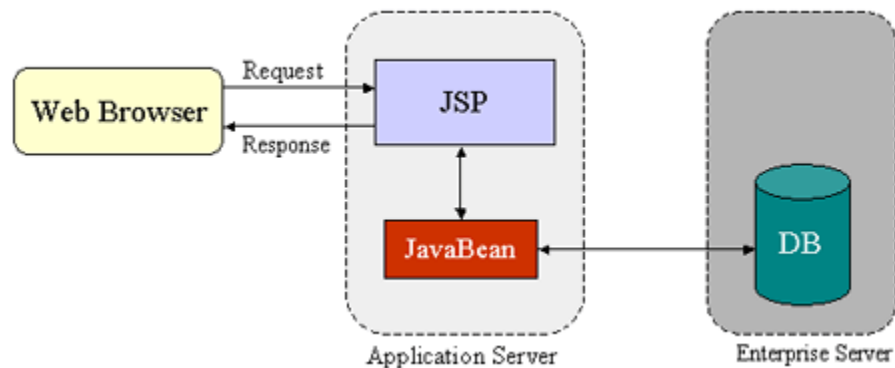
- Facilita pruebas unitarias (los servlets se pueden probar sin JSP).

- Permite evolucionar cada capa de forma independiente (por ejemplo, reemplazar JSP por Thymeleaf).

La especificación de JSP presenta dos enfoques para construir aplicaciones web utilizando páginas JSP: las arquitecturas Modelo 1 y Modelo 2. Estos dos modelos difieren en el lugar donde se realiza el procesamiento. En la arquitectura Modelo 1, como se muestra en la Figura 3, la página JSP es la responsable de procesar las solicitudes y enviar las respuestas de vuelta a los clientes.

Figura 3

Interacción entre JSP, JavaBeans y Base de Datos

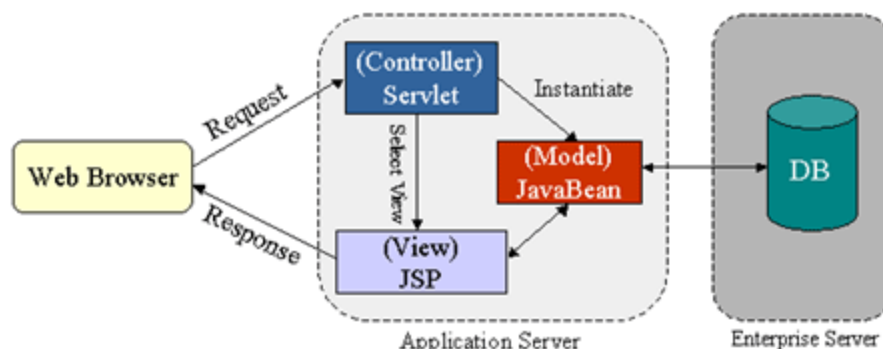


La figura 3 ilustra cómo el navegador web envía peticiones al JSP, el cual se comunica con un JavaBean para la lógica de negocio y este, a su vez, accede a la base de datos (DB). Finalmente, la respuesta generada retorna al navegador, cerrando el ciclo cliente-servidor.

La arquitectura Modelo 2, como se muestra en la Figura 3, integra el uso tanto de servlets como de páginas JSP. En este esquema, las páginas JSP se utilizan para la capa de presentación y los servlets para las tareas de procesamiento. El servlet actúa como controlador, encargándose de procesar las solicitudes y de crear los beans que necesite la página JSP. Asimismo, el controlador decide a qué página JSP reenviar la petición. La página JSP recupera los objetos creados por el servlet y extrae el contenido dinámico para insertarlo en una plantilla.

Figura 4

Arquitectura MVC en Aplicaciones Java EE



El esquema representa la interacción bajo el patrón Modelo-Vista-Controlador (MVC). El Servlet (Controlador) gestiona las peticiones del navegador, instancia los JavaBeans (Modelo) para acceder a la base de datos y determina la JSP (Vista) que mostrará la respuesta al usuario.

A continuación, una tabla comparativa de las arquitecturas JSP Modelo 1 y Modelo 2, resaltando sus diferencias clave:

Tabla 4.

Criterio	Modelo 1	Modelo 2 (MVC)
Lugar del procesamiento	La propia página JSP (o servlet asociado) procesa la petición, la lógica de negocio, validaciones y respuesta.	Un Servlet (Front Controller) recibe la petición y ejecuta la lógica; la JSP solo renderiza la vista.
Controlador dedicado	No existe un controlador central; cada JSP/servlet gestiona su propia lógica y navegación.	Un único servlet actúa como Front Controller, despachando a las JSP correspondientes según la acción.
Separación lógica vs. presentación	Lógica de negocio y presentación suelen mezclarse	Fuerte separación: la lógica reside en el servlet y el modelo,

	en la misma JSP, generando duplicación de código.	y la JSP únicamente en la capa de presentación.
Complejidad y escalabilidad	Adecuado para aplicaciones pequeñas o páginas dinámicas simples; difícil de mantener a gran escala.	Recomendado para aplicaciones medianas y grandes; facilita mantenimiento, testeo y extensiones.
Manejo de la navegación	Navegación basada en hipervínculos directos entre JSPs; cada página debe conocer rutas y lógica de negocio.	El controlador central gestiona la navegación, decidiendo dinámicamente la vista a mostrar.
Modelo de datos	Beans o lógica de negocio insertados directamente en la JSP; no hay capa de modelo formalizada.	POJOs, JavaBeans o entidades JPA gestionados por la capa de servicio/DAO; claros contratos de modelo.
Testabilidad	Baja, pues la lógica está incrustada en las páginas; difícil aislarla para pruebas unitarias.	Alta, al estar la lógica de negocio en clases Java y servlets, independientes de las vistas; facilita pruebas unitarias y de integración.
Uso histórico	Prevaleció en los primeros borradores de la especificación JSP 0.92 (1998) para casos de uso sencillos.	Formalizado como “Model 2” en 1999 y popularizado por frameworks como Apache Struts; sinónimo de MVC en Java Web.

2.2.2. JAVA

James Gosling (1995) describió Java como “un lenguaje de programación de propósito general, concurrente, basado en clases y orientado a objetos, diseñado para ser lo suficientemente sencillo como para que muchos programadores puedan dominarlo”.

Ian Darwin (1996), en The Java Language Specification, afirma que Java es “un lenguaje de programación multiplataforma, basado en clases y orientado a objetos, que abstrae detalles de la plataforma subyacente a través de la JVM”.

IBM (2025) define Java como “un lenguaje y plataforma de software ampliamente utilizado, que opera en miles de millones de dispositivos, con sintaxis derivada de C/C++ y un fuerte énfasis en la portabilidad y la seguridad”.

Britannica (2025) señala que “Java fue creado para permitir la comunicación entre dispositivos electrónicos de consumo y se consolidó como un estándar web en 1995, gracias a su promesa ‘write once, run anywhere’.

ResearchGate (2023) describe Java como “un lenguaje de propósito general que permite a los programadores escribir código universalmente compatible, ejecutable en múltiples plataformas sin recompilación”.

2.2.2.1. Java Server Pages (JSP)

Java Server Pages (anteriormente Java Server Pages) es una especificación de Jakarta EE que permite a los desarrolladores crear páginas web dinámicas basadas en HTML, XML u otros formatos, incorporando código Java directamente en la plantilla. En tiempo de ejecución, cada página JSP se traduce a un servlet, compilándose en bytecode para la JVM. (Coursera Staff, 2024)

Características principales:

- **Separación de presentación y lógica:** Permite mantener la presentación en JSP y la lógica en servlets o JavaBeans, siguiendo el patrón MVC.
- **Expression Language (EL) y JSTL:** Facilitan la extracción de datos y el control de flujo sin usar scriptlets Java en la página.
- **Tag libraries:** JSTL y tags personalizados añaden funciones comunes (iteraciones, condicionales, internacionalización) sin código Java embebido.
- **Alto rendimiento:** La traducción a servlets y el caching de los mismos evitan recompilaciones innecesarias hasta que la JSP cambia.

2.2.2.2. Java Servlets

Un servlet es una clase Java que se ejecuta dentro de un contenedor (por ejemplo, Tomcat o Jetty) y extiende las capacidades de un servidor web mediante el modelo de programación petición-respuesta. Aunque puede manejar diversos protocolos, su uso más común es para solicitudes HTTP(S).

Características principales:

- **Eficiencia y escalabilidad:** Manejan múltiples peticiones creando hilos en lugar de procesos, lo que mejora el rendimiento respecto a CGI tradicionales.
- **Portabilidad:** Al implementarse en Java y ejecutarse en la JVM, funcionan en cualquier servidor compatible sin recompilación.

- **Extensibilidad:** Integración nativa con Java EE (filtros, listeners, seguridad, transacciones) y frameworks MVC (Struts, Spring MVC).
- **Control de ciclo de vida:** Métodos `init()`, `service()` y `destroy()` permiten gestionar recursos y tareas de inicialización/limpieza.

2.2.2.3. Java Persistence API (JPA)

La Java Persistence API es una especificación de Jakarta EE (JSR 338) que estandariza el mapeo objeto-relacional (ORM) y la gestión de la persistencia de datos en aplicaciones Java. Permite definir entidades POJO que sobreviven más allá del ciclo de vida de la aplicación mediante un `EntityManager`. (IBM, 2025)

Características principales:

- **Mapeo declarativo:** Anotaciones (`@Entity`, `@Id`, `@OneToMany`, etc.) o XML describen cómo las clases Java se corresponden con tablas y columnas de la base de datos.
- **API de consultas:** JPQL (Java Persistence Query Language) y Criteria API ofrecen consultas orientadas a objetos, independientes de la base de datos subyacente.
- **Gestión de transacciones:** Integración automática con JTA o transacciones locales, asegurando la atomicidad de las operaciones de persistencia.
- **Cache de primer y segundo nivel:** Optimización del acceso a datos mediante caching a nivel de `EntityManager` y compartido entre instancias.

2.2.2.4. Hibernate ORM

Hibernate ORM es un framework de mapeo objeto-relacional de código abierto para Java, desarrollado originalmente por Gavin King y mantenido por Red Hat. Proporciona una implementación completa de JPA y añade funcionalidades avanzadas como HQL, caching, auditoría y mapeo de herencia. (King, Bauer, Andersen, Bernard, & Ebersole, 2010)

Características principales:

- **HQL (Hibernate Query Language):** Lenguaje de consulta orientado a objetos, similar a SQL pero basado en entidades Java, que abstrae diferencias entre bases de datos.
- **Mapeo flexible:** Soporta mapeo por anotaciones o XML, relaciones uno-a-muchos, muchos-a-muchos, herencia y tipos personalizados.
- **Caching multinivel:** Caching de primer nivel (por sesión) y segundo nivel (por SessionFactory) para mejorar el rendimiento en aplicaciones de gran escala.
- **Integración amplia:** Compatible con Java SE y Java EE, EJB, CDI y frameworks como Spring; facilita la construcción de aplicaciones tanto monolíticas como basadas en microservicios.

III. DESARROLLO

3.1.Introducción

El presente capítulo tiene como objetivo exponer de manera estructurada el proceso de construcción del módulo de gestión de capacitaciones, integrando tanto los fundamentos técnicos como las decisiones prácticas que guiaron su desarrollo. Esta sección se centra en la implementación concreta del sistema, detallando los componentes que lo conforman, la lógica de funcionamiento y las tecnologías utilizadas.

Para asegurar un flujo de trabajo ordenado, flexible y enfocado en la entrega continua de valor, se adoptó la metodología ágil Kanban, que permitió priorizar tareas, visualizar avances y realizar ajustes de forma oportuna durante el ciclo de desarrollo. Esta metodología resultó adecuada dadas las condiciones del entorno y la necesidad de ir realizando cambios con base en pruebas funcionales y retroalimentación interna.

El sistema fue diseñado bajo el patrón arquitectónico Modelo Vista Controlador (MVC), lo que facilitó una clara separación entre la lógica de negocio, la presentación y el control de flujo. Este enfoque permitió trabajar de forma modular y escalable, facilitando futuras ampliaciones o mantenimientos del sistema.

En las siguientes secciones se documenta cada etapa del desarrollo, donde las actividades fueron distribuidas estratégicamente conforme al cronograma oficial aprobado por la institución, abarcando desde la fase de análisis de requerimientos hasta la entrega final y la capacitación al personal encargado del uso del sistema.

Este capítulo constituye el componente operativo del proyecto, en el que se materializan las necesidades detectadas en soluciones funcionales concretas. A través del uso eficiente de tecnologías web basadas en Java, se desarrolló una herramienta que responde directamente a los requerimientos de la organización y contribuye a optimizar sus procesos de capacitación.

3.2. Metodología de desarrollo

Para llevar a cabo el desarrollo del módulo de gestión de capacitaciones en POSCO MPPC Puebla, como ya se mencionó se empleó la metodología ágil Kanban (Figura 5), orientada al control visual del flujo de trabajo y a la entrega continua de funcionalidades. Esta metodología fue seleccionada por su flexibilidad, simplicidad de implementación y efectividad en entornos donde las prioridades pueden cambiar con frecuencia.

El enfoque Kanban permitió estructurar y organizar las actividades del proyecto mediante tableros que representaban las etapas del proceso: por hacer, en proceso y finalizado. Cada tarea fue representada como una tarjeta en el tablero, lo cual facilitó la identificación oportuna de bloqueos, la asignación clara de responsabilidades y la visibilidad del avance en tiempo real.

Figura 5

Tablero Kanban del proyecto



Visualización de las tareas distribuidas por fases y estado, utilizadas durante el desarrollo del módulo de capacitación. Esta herramienta facilitó el seguimiento del avance, la priorización de entregables y la detección de bloqueos en tiempo real.

Las tareas fueron clasificadas y gestionadas de acuerdo con las fases del desarrollo del sistema, organizadas visualmente en un tablero Kanban que permitió llevar un control efectivo del avance. Estas fases incluyeron:

- **Levantamiento de requerimientos:** Incluyó el análisis del proceso actual de asignación de capacitaciones, reuniones con el dueño del proceso en el departamento de recurso humanos y la recopilación de requisitos funcionales y no funcionales.
- **Diseño y arquitectura:** Se definió la metodología de trabajo, se modeló la base de datos y se estructuró la arquitectura del sistema siguiendo el patrón Modelo Vista Controlador (MVC), aprovechando tecnologías como Servlets, JSP y JPA.
- **Diseño de interfaces:** Se diseñaron las vistas principales del sistema, incluyendo las interfaces del administrador, del usuario final y del módulo de estadísticas, priorizando la usabilidad y la coherencia visual con ayuda de Bootstrap.
- **Desarrollo de módulos:** Se implementaron las funcionalidades correspondientes a cada módulo del sistema:
 - Módulo del administrador
 - Módulo de usuarios
 - Módulo de estadísticas y reportesCada módulo fue desarrollado empleando AJAX para mejorar la experiencia del usuario y facilitar la interacción sin recargas.
- **Pruebas y revisión:** Se realizaron pruebas funcionales por cada componente desarrollado, así como actividades de revisión y corrección de errores detectados durante la validación interna.

- **Capacitación y cierre:** Finalmente, se capacitó al responsable operativo sobre el uso del sistema y se brindó soporte post-implementación para garantizar su correcta adopción.

La implementación del tablero Kanban, utilizando la herramienta digital Notion, facilitó una gestión visual efectiva del desarrollo del proyecto. Esto permitió tener visibilidad inmediata del avance, mejorar la coordinación entre las tareas y resolver oportunamente cualquier dificultad presentada. De este modo, fue posible adaptarse con agilidad a modificaciones en los requerimientos y mantener una comunicación fluida con los integrantes involucrados en cada fase del desarrollo.

3.3. Requisitos funcionales y no funcionales

En esta sección se describen de manera detallada los requisitos que se identificaron, así como del análisis exhaustivo del proceso actual de asignación y seguimiento de capacitaciones dentro de la organización. Estos requisitos constituyen la base sobre la cual se construyó y evaluó el módulo desarrollado.

Los requisitos funcionales, los cuales definen claramente las características, comportamientos y funcionalidades específicas que el sistema debe ofrecer a los usuarios finales.

Por otra parte, los requisitos no funcionales, son los que establecen los criterios técnicos, operativos y de calidad que garantizan el correcto desempeño, la estabilidad, la seguridad, la usabilidad y la escalabilidad del sistema implementado, asegurando así la satisfacción de las expectativas del usuario y los objetivos estratégicos de la empresa.

Tabla 5.

Requisitos funcionales

ID	Módulo	Descripción
RF01	Cursos Pendientes	El sistema debe permitir al usuario consultar los cursos asignados correspondientes al mes actual.
RF02	Cursos Pendientes	El sistema debe mostrar el nombre del curso, duración, instructor, fechas de inicio y fin, estatus y porcentaje de avance del curso.
RF03	Cursos Pendientes	El sistema debe mostrar la información personal del usuario como número de control, nombre, planta, departamento y nivel.
RF04	Cursos Pendientes	El sistema debe permitir al usuario filtrar los cursos asignados por rango de fechas.

RF05	Cursos Pendientes	El sistema debe calcular y mostrar gráficamente el avance general de los cursos completados vs pendientes del usuario.
RF06	AdministraCursos	El sistema debe permitir al administrador cargar archivos .ods, .xls o .xlsx que contengan datos de cursos.
RF07	AdministraCursos	El sistema debe procesar el archivo cargado y mostrar en una tabla los cursos contenidos en él.
RF08	AdministraCursos	El sistema debe permitir guardar en la base de datos los cursos tabulados desde el archivo, validando si ya existen registros para actualizarlos.
RF09	AdministraCursos	El sistema debe mostrar mensajes de éxito o error al guardar los cursos procesados.
RF10	EstadisticasCursos	El sistema debe permitir al administrador buscar cursos por número de control, nombre del empleado o clave de curso.
RF11	EstadisticasCursos	El sistema debe mostrar los cursos encontrados con información como nombre, clave, duración, fecha de inicio y estatus.
RF12	EstadisticasCursos	El sistema debe mostrar una tabla con los resultados filtrados o un mensaje si no se encuentran coincidencias.
RF13	EstadisticasCursos	El sistema debe permitir limpiar los filtros de búsqueda para realizar una nueva consulta.
RF14	Cursos Pendientes	El sistema debe restringir el acceso a la vista de cursos únicamente a usuarios autenticados mediante sesión.
RF15	Cursos Pendientes	El sistema debe cargar automáticamente los datos del trabajador desde la base de datos al iniciar sesión.
RF16	Cursos Pendientes	El sistema debe permitir acceder al detalle del curso mediante el clic en tarjetas o elementos de la lista (funcionalidad futura extensible).
RF17	AdministraCursos	El sistema debe validar que el archivo subido no exceda el tamaño máximo permitido de 10MB.
RF18	AdministraCursos	El sistema debe permitir que el botón de "Guardar" solo esté habilitado después de haber cargado correctamente un archivo válido.
RF19	AdministraCursos	El sistema debe actualizar el estatus y porcentaje de un curso si ya existe en la base de datos, en vez de duplicar el registro.
RF20	AdministraCursos	El sistema debe eliminar la lista de cursos en sesión después de un guardado exitoso para evitar registros duplicados.
RF21	EstadisticasCursos	El sistema debe cargar los datos del usuario administrador en el panel lateral para mantener la identidad de sesión.
RF22	EstadisticasCursos	El sistema debe mantener persistentes los filtros de búsqueda después de realizar una consulta para facilitar la navegación del usuario.
RF23	EstadisticasCursos	El sistema debe manejar errores de conversión de parámetros y mostrar mensajes claros si la búsqueda falla por datos mal ingresados.
RF24	General	El sistema debe seguir un diseño responsivo compatible con dispositivos móviles y de escritorio.
RF25	General	El sistema debe mostrar mensajes de error y éxito mediante modales o alertas para mejorar la experiencia del usuario.
RF26	General	El sistema debe permitir una navegación fluida entre módulos para facilitar el uso por parte del usuario y del administrador.

Esta tabla enumera los comportamientos específicos que el sistema debe cumplir desde la perspectiva del usuario final y del administrador. Incluye las acciones que los usuarios pueden realizar (consultar, filtrar, cargar, guardar), así como las respuestas esperadas del sistema (mostrar resultados, actualizar registros, validar archivos, entre otros).

Tabla 6.

Requisitos No Funcionales

ID	Categoría	Descripción
RNF01	Rendimiento	El sistema debe procesar la carga de archivos con al menos 500 registros en menos de 5 segundos.
RNF02	Rendimiento	El sistema debe renderizar las tablas de resultados y gráficos sin afectar la fluidez de la interfaz de usuario.
RNF03	Seguridad	El sistema debe restringir el acceso a los módulos mediante autenticación de usuario mediante sesión activa.
RNF04	Seguridad	Los parámetros recibidos desde el cliente deben ser validados para evitar ataques de inyección o manipulación de datos.
RNF05	Seguridad	Los archivos cargados por los administradores deben ser validados por su extensión y tamaño antes de ser procesados.
RNF06	Usabilidad	El sistema debe contar con una interfaz intuitiva que facilite la navegación por parte de usuarios con conocimientos básicos de informática.
RNF07	Usabilidad	El sistema debe mostrar mensajes claros de validación, error y éxito durante las operaciones realizadas.
RNF08	Mantenibilidad	El código debe estar estructurado en módulos reutilizables y fácil de mantener con separaciones claras entre la lógica de negocio y la vista.
RNF09	Compatibilidad	El sistema debe funcionar correctamente en los navegadores modernos como Chrome, Firefox y Edge.
RNF10	Escalabilidad	El sistema debe permitir la futura integración con módulos adicionales como evaluación por curso o generación de reportes automáticos.
RNF11	Portabilidad	La aplicación debe poder desplegarse en diferentes servidores que soporten Java EE y JSP.
RNF12	Fiabilidad	El sistema debe garantizar que los datos cargados no se pierdan en caso de errores durante el guardado masivo.
RNF13	Accesibilidad	La interfaz debe ser accesible mediante teclado y tener soporte para lectores de pantalla básicos.
RNF14	Localización	El sistema debe mostrar la información en español y utilizar formatos de fecha, número y texto acordes a la región.
RNF15	Integridad de datos	El sistema debe evitar la duplicidad de registros mediante validación antes de insertar o actualizar en la base de datos.

Define los atributos de calidad del sistema, tales como rendimiento, seguridad, usabilidad, compatibilidad, mantenimiento, escalabilidad y fiabilidad. No describen lo que hace el sistema, sino cómo debe hacerlo para garantizar un funcionamiento estable y eficiente.

3.4. Recolección de información

La recopilación de información se realizó a través de diversas reuniones con el encargado del proceso de capacitación dentro de la organización. Estas sesiones permitieron identificar de manera directa las necesidades operativas, los puntos críticos y las áreas de mejora en el manejo de cursos y evaluaciones.

Durante las reuniones, se abordaron temas clave relacionados con la asignación de cursos, el seguimiento del avance de los empleados, así como la gestión y análisis de resultados. Se utilizó un enfoque cualitativo que facilitó la exploración de las percepciones, experiencias y expectativas del responsable del proceso, lo cual fue esencial para el levantamiento de los requerimientos funcionales y no funcionales del sistema.

Estas reuniones se llevaron a cabo conforme a una guía temática previamente diseñada, permitiendo asegurar la recolección estructurada de información relevante. Gracias a este proceso, fue posible definir con claridad los elementos clave para el desarrollo del módulo de administración y seguimiento de cursos.

3.5. Especificación de Casos de Uso

Este apartado tiene como objetivo describir de manera detallada los casos de uso que componen el sistema desarrollado como se muestra en la Figura 6, con el fin de representar claramente las interacciones entre los usuarios (actores) y las funcionalidades que ofrece la aplicación. La especificación de casos de uso permite identificar qué acciones puede realizar cada tipo de usuario dentro del sistema, así como los flujos principales de comportamiento esperados en cada escenario.

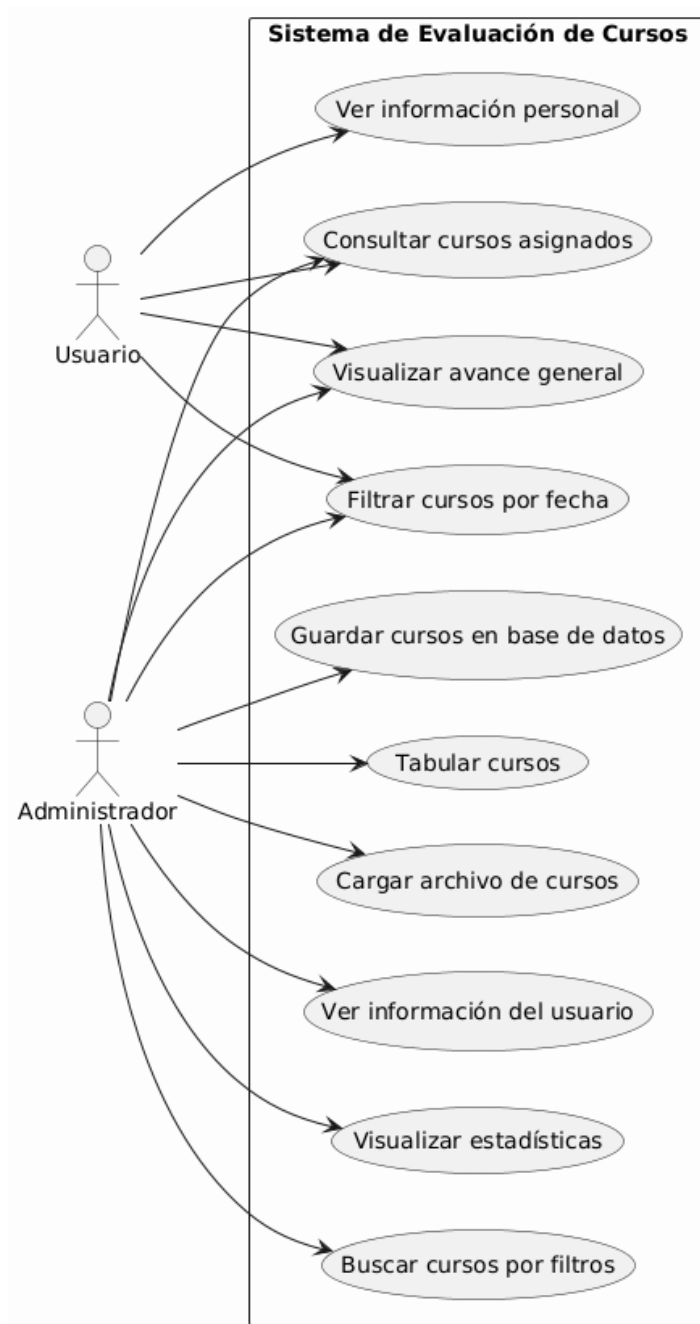
Para su elaboración, se tomó como base la información obtenida durante las reuniones sostenidas con el encargado del proceso de capacitación, lo que permitió definir con claridad los roles involucrados y las funcionalidades requeridas tanto para los usuarios comunes como para el personal administrativo.

Los actores identificados incluyen al Usuario (empleado que consulta sus cursos), y al Administrador (encargado de cargar, gestionar y consultar estadísticas de los cursos). Cada uno cuenta con un conjunto específico de casos de uso que reflejan sus responsabilidades dentro del sistema. La representación gráfica de estos casos se muestra en los diagramas correspondientes, mientras que su comportamiento se detalla mediante descripciones estructuradas que incluyen nombre, actor principal, flujo de eventos, condiciones iniciales y finales, y excepciones.

Esta especificación constituye una herramienta fundamental para asegurar que el desarrollo del sistema cumpla con los requerimientos establecidos y para facilitar su mantenimiento y evolución futura.

Figura 6

Diagrama de casos de usos



El diagrama muestra las principales funcionalidades del sistema clasificadas según los actores Usuario y Administrador, quienes interactúan con los distintos módulos para consultar, gestionar y analizar la información relacionada con los cursos.

El diagrama ha sido elaborado utilizando la notación UML (Unified Modeling Language), la cual permite representar de forma estandarizada las interacciones entre los actores y el sistema. Esta notación es fundamental para comprender, analizar y comunicar de manera clara la funcionalidad del software antes y durante su desarrollo.

3.6. Arquitectura del sistema

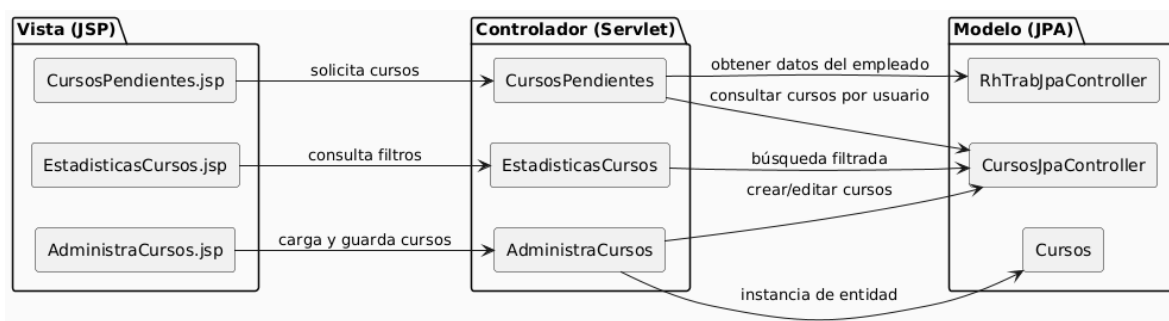
La arquitectura del sistema describe la estructura general de la aplicación, así como la organización de sus componentes y la forma en que interactúan entre sí.

Para el desarrollo de este proyecto se eligió el patrón de arquitectura MVC (Modelo-Vista-Controlador), el cual permite una separación clara de responsabilidades: el Modelo gestiona la lógica y el acceso a datos, la Vista se encarga de la presentación al usuario, y el Controlador actúa como intermediario que procesa las peticiones y coordina la respuesta del sistema.

Esta organización facilita el mantenimiento, la escalabilidad y la reutilización del código a lo largo del ciclo de vida del software.

En la figura muestra de manera visual el diagrama de la arquitectura utilizada en el módulo a desarrollar.

Figura 7



La figura representa la arquitectura del sistema basada en el patrón MVC, donde las vistas (JSP) interactúan con los controladores (servlets), los cuales gestionan la lógica y comunicación con el modelo (entidades JPA y controladores de acceso a datos).

Como se mencionó anteriormente, la arquitectura elegida fue MVC (Modelo-Vista-Controlador), una de las estructuras más utilizadas en el desarrollo de aplicaciones web debido a su claridad, separación de responsabilidades y facilidad de mantenimiento. Además, esta decisión se tomó considerando que el sistema donde se integrará el módulo ya está desarrollado bajo esta misma arquitectura.

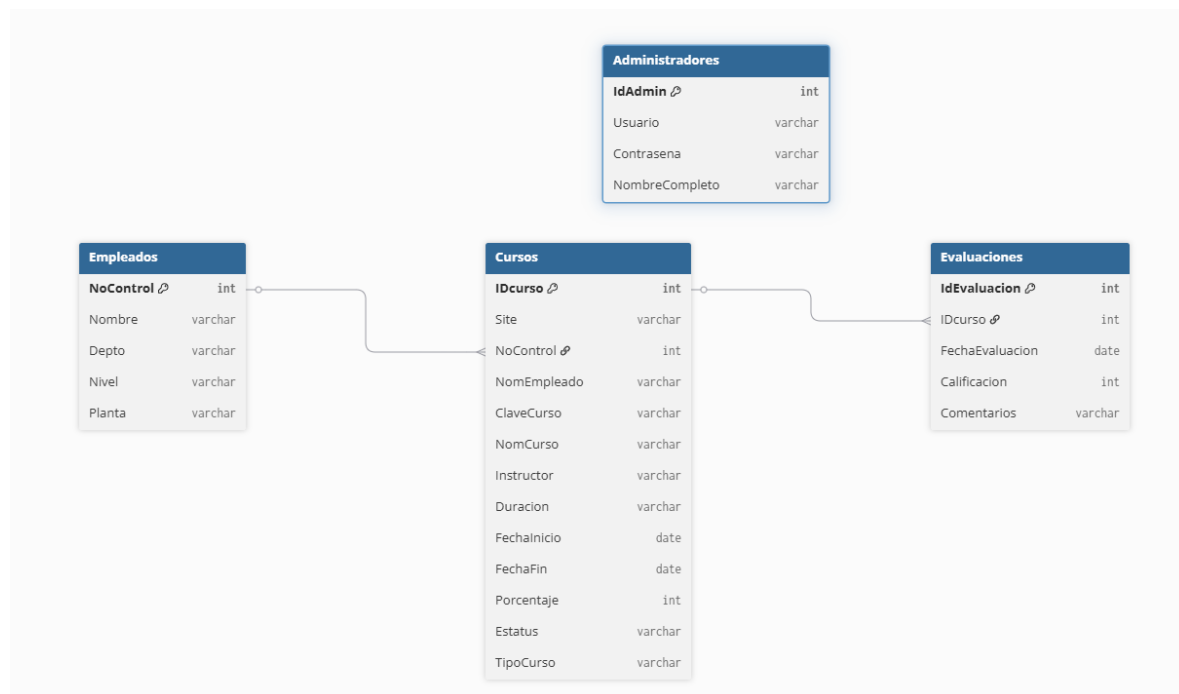
- **Modelo:** Está conformado por las entidades JPA que representan las tablas de la base de datos, tales como la clase Cursos, que encapsula la estructura y comportamiento de los registros relacionados con los cursos asignados. Además, incluye los controladores de persistencia (CursosJpaController, RhTrabJpaController), los cuales gestionan las operaciones de acceso a datos como búsquedas, inserciones y actualizaciones a través de la API de JPA (Java Persistence API).
- **Vista:** Esta capa está constituida por las páginas JSP como CursosPendientes.jsp, AdministraCursos.jsp y EstadisticasCursos.jsp. Estas vistas permiten la interacción con el usuario final (ya sea empleado o administrador), mostrando formularios, tablas de resultados y gráficas interactivas. Se utilizan tecnologías complementarias como JSTL, Bootstrap y Chart.js para enriquecer la interfaz y mejorar la experiencia del usuario.
- **Controlador:** En esta capa se encuentran los servlets como CursosPendientes, AdministraCursos y EstadisticasCursos, los cuales actúan como intermediarios entre la vista y el modelo. Su función principal es recibir solicitudes HTTP desde las vistas, procesar los datos, realizar las operaciones correspondientes en el modelo, y reenviar la información a la vista para su presentación. Cada servlet está diseñado para manejar flujos de datos específicos, como la carga masiva de cursos, la visualización de estadísticas, o la consulta de cursos asignados.

Esta separación lógica permite una mayor escalabilidad, ya que cada componente puede modificarse o ampliarse de manera independiente sin afectar el resto del sistema. Además, facilita el mantenimiento, ya que el código está organizado de forma modular y comprensible, siguiendo buenas prácticas de desarrollo web en entornos Java EE.

3.7. Diseño de Base de Datos

El diseño de base de datos constituye un componente esencial en el desarrollo del sistema, ya que permite estructurar, organizar y relacionar de manera eficiente la información que será gestionada por el módulo. En este apartado se presenta el modelo relacional propuesto, el cual fue construido a partir de los requerimientos funcionales identificados, garantizando la integridad, coherencia y escalabilidad de los datos. Este diseño facilita el almacenamiento de información relevante sobre empleados, cursos asignados, evaluaciones realizadas y usuarios administradores, permitiendo consultas rápidas, mantenimiento eficiente y futuras extensiones del sistema.

Figura 8



Modelo relacional del sistema, donde se representan las principales entidades involucradas, como empleados, cursos y evaluaciones, así como sus relaciones mediante claves primarias y foráneas. Este esquema facilita la integridad y organización de la información gestionada por el módulo.

3.8. Historias de usuario

Las historias de usuario son una técnica ágil utilizada para capturar los requerimientos del sistema desde la perspectiva del usuario final. En esta sección se presentan las historias de usuario planteadas las cuales describen de manera sencilla y estructurada las necesidades y

expectativas de los distintos perfiles que interactúan con el módulo, tales como usuarios administrativos y empleados.

Cada historia se redacta siguiendo el formato “Como [rol], quiero [funcionalidad] para [objetivo]”, lo que permite comprender claramente el propósito de cada función dentro del contexto del sistema de evaluación de cursos. Estas historias sirvieron como base para el diseño de los casos de uso y el desarrollo funcional del sistema.

Tabla 7.

Historia de Usuario 1

ID	Descripción Historia de Usuarios		Criterios de Aceptación
1	Como	Administrador	El sistema debe permitir seleccionar un archivo .ods o .xlsx.
	Quiero	Cargar un archivo con los cursos del personal, especificando su contenido y formato.	Debe validarse el formato y los datos antes de guardar.
	Para	Poder visualizar y almacenar los cursos en la base de datos del sistema.	Se debe confirmar la carga con un mensaje visual.

Esta historia describe la necesidad del administrador de cargar cursos de forma masiva a través de un archivo estructurado, asegurando su validación antes del almacenamiento.

Tabla 8.

Historia de Usuario 2

ID	Descripción Historia de Usuarios		Criterios de Aceptación
2	Como	Administrador	El sistema debe aceptar parámetros de búsqueda.
	Quiero	Buscar cursos mediante filtros como número de control, nombre o clave de curso.	Debe mostrar los resultados en una tabla dinámica.

	Para	Consultar el historial y estadísticas de participación.	Debe permitir limpiar filtros fácilmente.
--	------	---	---

Permite al administrador consultar cursos de manera eficiente mediante filtros personalizables, facilitando el seguimiento y análisis de la capacitación del personal.

Tabla 9.

Historia de Usuario 3

ID	Descripción Historia de Usuarios		Criterios de Aceptación
3	Como	Empleado	El sistema debe listar los cursos activos por mes.
	Quiero	Ver los cursos que me han sido asignados y su avance.	Debe mostrar nombre, fecha de inicio y porcentaje de avance.
	Para	Dar seguimiento a mi participación y cumplimiento.	Solo deben mostrarse cursos asignados al usuario actual.

Enfocada en el empleado, esta historia permite visualizar los cursos asignados con su progreso, brindando una vista clara del cumplimiento mensual de sus responsabilidades formativas.

Tabla 10.

Historia de Usuario 4

ID	Descripción Historia de Usuarios		Criterios de Aceptación
4	Como	Empleado	El sistema debe permitir descargar constancias únicamente de cursos con estatus "Completado"
	Quiero	Descargar las constancias de los cursos que he completado	El archivo debe generarse en formato PDF con los datos del curso y del empleado
	Para	Tener evidencia formal de mi participación y poder presentarla cuando sea requerido	Debe mostrarse un mensaje de confirmación o alerta en caso de error al generar la constancia

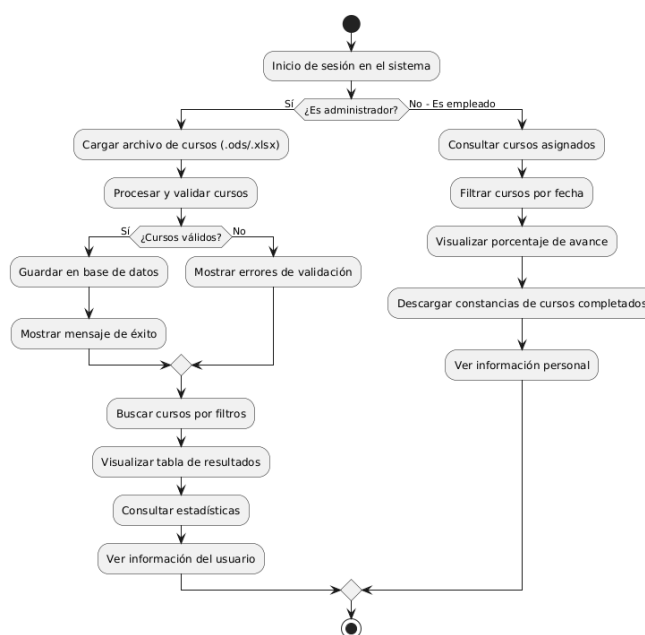
Esta historia cubre la funcionalidad de descargar constancias, permitiendo al empleado obtener comprobantes oficiales de los cursos completados para su respaldo personal o institucional.

3.9.Desarrollo de la lógica del negocio

Con el objetivo de documentar de manera clara y estructurada el funcionamiento interno del módulo desarrollado, se optó por representar el flujo completo del sistema mediante diagramas técnicos que permiten visualizar tanto el orden lógico de las operaciones como la interacción entre los diferentes componentes del sistema.

En primer lugar, se presenta un diagrama de flujo, el cual describe de manera secuencial cada una de las acciones que componen el proceso, desde la carga y visualización de cursos hasta la descarga de constancias por parte del usuario. Este diagrama es fundamental para comprender la lógica general del módulo y para detectar posibles puntos de mejora o validación en el ciclo de ejecución.

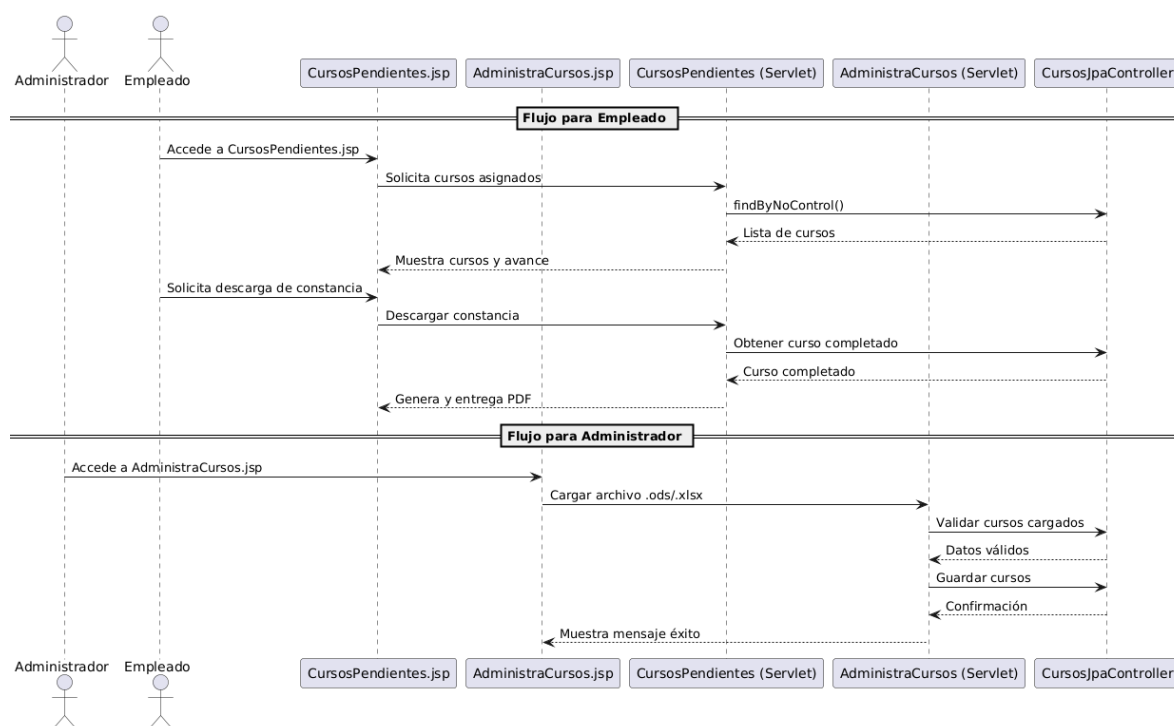
Figura 9



Asimismo, se incluye un diagrama de secuencia, el cual representa las interacciones dinámicas entre los distintos actores (como el administrador y el empleado), las interfaces

(vistas JSP), los controladores (servlets) y el modelo de datos (entidades JPA). Este tipo de diagrama es esencial para visualizar cómo fluye la información dentro del sistema y cómo se orquestan las acciones a través de los distintos niveles de la arquitectura MVC.

Figura 10



Ambos diagramas permiten ilustrar y analizar el comportamiento esperado del módulo, facilitando el entendimiento para desarrolladores, analistas y usuarios técnicos, así como sirviendo como base para futuras mejoras o ampliaciones del sistema.

3.10. Diseño y maquetado de Interfaces de Usuario

El diseño de las interfaces del módulo se realizó pensando en la facilidad de uso, siguiendo principios de experiencia de usuario (UX). Se aseguró que cada pantalla fuera clara, accesible y fácil de manejar tanto para empleados como para administradores.

En esta etapa, se crearon prototipos sencillos con Balsamiq Mockups para probar la estructura y funciones con usuarios reales antes del desarrollo final. Esto ayudó a identificar y solucionar posibles errores de usabilidad desde el principio.

Las imágenes muestran las pantallas principales y cómo se navega entre ellas, dando una idea clara de cómo funciona el sistema. Además, el diseño es responsivo, por lo que se adapta correctamente a distintos dispositivos

Figura 11

Panel de carga del listado de cursos (Administrador)

The screenshot displays the MPPCHub interface. The top navigation bar includes the posco MPPC logo, the title 'MPPCHub', and a user profile section with a home icon and a 'Logout' link. The main content area is divided into a sidebar and a central panel.

Sidebar (Left):

- Información de Usuario:**
 - No. Control: XXXXX
 - Nombre: Juan Perez
 - Planta: Pueblo
 - Departamento: IT
- Avances:** A circular progress indicator showing approximately 75% completion.

Main Panel (Right):

- Listado de Cursos:** A section for uploading files, featuring a dashed box with an upload icon and a 'Choose files to Upload' button. Below this are 'Tabular' and 'Guardar' buttons.
- Datos cargados:** A table listing uploaded data.

Name	File	Category	Author	Status
Project_Vision	PDF	Planning	Alice Johnson	SENT
Market_Analysis	XLSX	Research	James Lee	IN PROGRESS
UX_Design_Protocol	DOCX	Design	Maria Chen	PENDING
Budget_2022	XLSX	Finance	Oliver Smith	SENT
Staffing_Plan	PDF	HR	Hannah Arendt	IN PROGRESS

The footer contains the text 'The Friday Institute' and a link to 'Privacy & Terms'.

Interfaz destinada al administrador para seleccionar y cargar archivos con el listado de cursos en formato .ods o .xlsx. Incluye funciones para validar, procesar y guardar la información en la base de datos, asegurando la correcta asignación de capacitaciones al personal.

Figura 12

Panel de visualización de capacitaciones asignadas

The screenshot displays the MPPCHub web application interface. At the top, there is a browser address bar showing 'http://coursehomepage.com/announcements'. Below this is a header bar with the 'posco MPPC' logo on the left and 'MPPCHub' in the center. A dark blue navigation bar contains a user profile icon labeled 'username', a home icon, and a 'Logout' button. Below the navigation bar, the main content area is divided into several sections. On the left, a 'Información de Usuario' (User Information) sidebar lists details: 'No. Control' (xxxxx), 'Nombre' (Juan Perez), 'Planta' (Puebla), and 'Departamento' (IT). Below this is an 'Avances' (Progress) section with a pie chart showing approximately 75% completion. The central area features a 'Cursos Asignados' (Assigned Courses) section with a search bar and a 'Browse...' button. Below this, a horizontal line indicates 'Mes Actual: Enero' (Current Month: January). Two course cards are displayed: 'INDUCCIÓN A LA EMPRESA' (Company Induction) and 'NOM 001 EDIFICIOS E INSTALACIONES' (NOM 001 Buildings and Installations). Each card shows course details like 'Clave' (Key), 'Instructor', 'Duración' (Duration), 'Inicio' (Start), 'Fin' (End), and 'Progreso' (Progress), along with a 'Completado' (Completed) button.

Curso	Clave	Instructor	Duración	Inicio	Fin	Progreso
INDUCCIÓN A LA EMPRESA	P00-IN01	INSTRUCTOR 1	2 horas	07/01/2025	07/01/2025	100%
NOM 001 EDIFICIOS E INSTALACIONES	P01-N001	Instructor	2 DÍAS	11/04/2025	12/04/2025	100%

Vista dirigida a los empleados para consultar las capacitaciones que les han sido asignadas, mostrando información como nombre del curso, fecha de inicio, fecha de finalización y porcentaje de avance.

Figura 13

Panel de búsqueda y seguimiento de cursos/capacitaciones

The screenshot displays the MPPCHub interface. At the top, there's a header with the 'posco MPPC' logo and the title 'MPPCHub'. Below this is a navigation bar with a user profile icon, the text 'Autoevaluación', 'Cursos', and a 'Logout' button. The main content area is divided into three sections:

- Información de Usuario:** A sidebar on the left showing user details: 'No. Control' (xxxxx), 'Nombre' (Juan Perez), 'Planta' (Puebla), and 'Departamento' (IT).
- Filtros:** A search and filter section on the right. It includes a 'Search' input field, a 'Browse...' button, and date pickers for filtering by date. A 'Guardar' (Save) button is also present.
- Table of Courses:** A table listing course records with columns: Name, File, Category, Author, and Status.

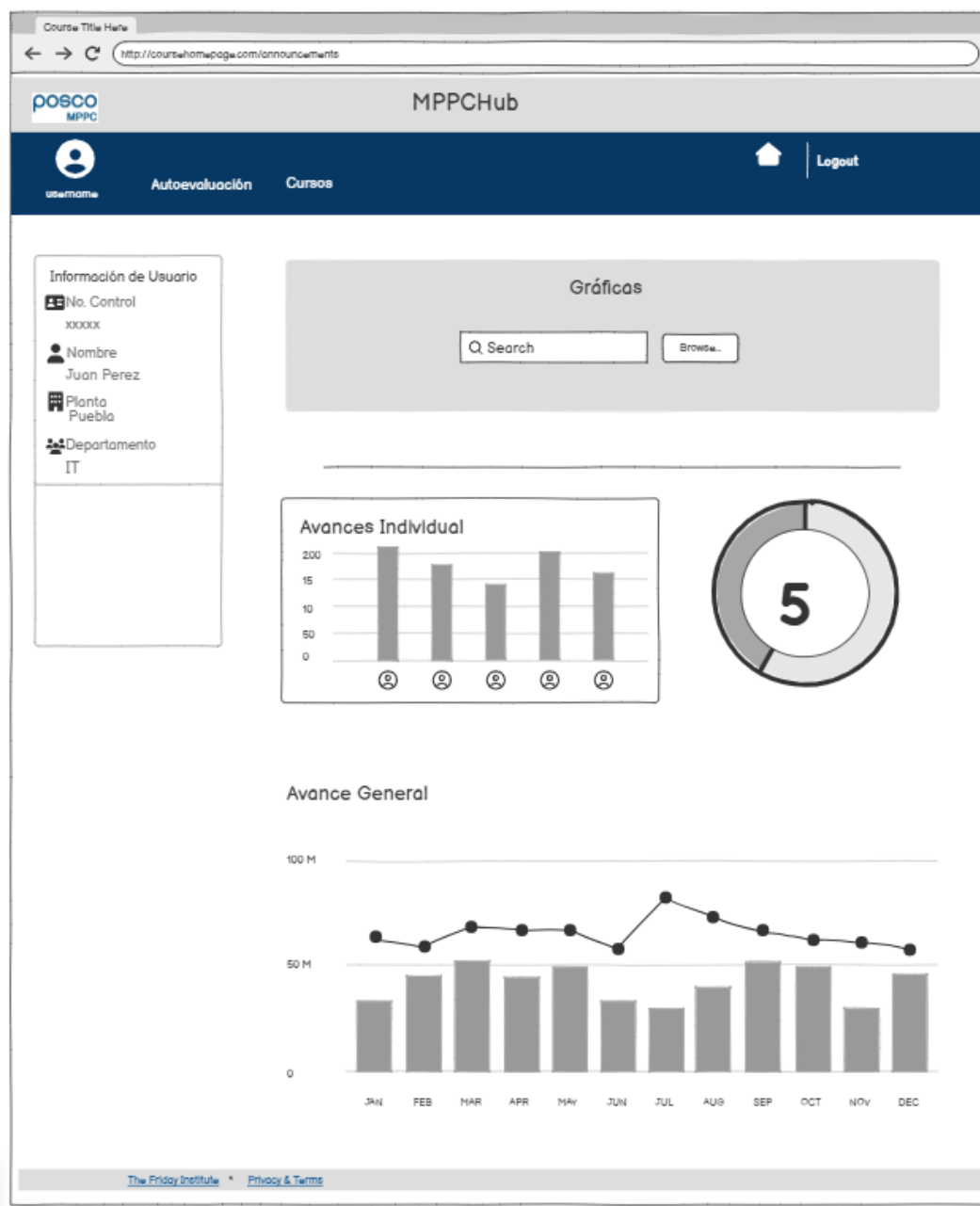
Name	File	Category	Author	Status
Project_Vision	PDF	Planning	Alice Johnson	SENT
Market_Analysis	XLSX	Research	James Lee	IN PROGRESS
UX_Design_Protocol	DOCX	Design	Maria Chen	PENDING
Budget_2022	XLSX	Finance	Oliver Smith	SENT
Staffing_Plan	PDF	HR	Hannah Arendt	IN PROGRESS

At the bottom of the page, there are links for 'The Friday Institute' and 'Privacy & Terms'.

Herramienta de filtrado y búsqueda que permite al administrador localizar cursos por criterios como número de control, nombre del empleado o clave del curso, además de consultar el progreso de cada capacitación.

Figura 14

Panel de reportes y estadísticas de capacitaciones



Pantalla que muestra al administrador gráficos y reportes consolidados sobre la participación y avance de los empleados en las capacitaciones, permitiendo la toma de decisiones basada en datos.

3.11. Configuración del entorno de desarrollo

La correcta configuración del ambiente de desarrollo constituye un pilar esencial para el éxito de cualquier proyecto de software, ya que permite estandarizar las herramientas, versiones y entornos utilizados por los desarrolladores, facilitando así la colaboración, depuración y despliegue del sistema. En el caso del presente módulo de seguimiento de cursos, se implementó un entorno de desarrollo basado en tecnologías Java EE, el cual se integra sin inconvenientes con la arquitectura existente de la empresa.

Esta sección documenta detalladamente los componentes de software requeridos, incluyendo el entorno de desarrollo integrado (IDE), servidor de aplicaciones, sistema gestor de base de datos y configuraciones específicas adoptadas durante la etapa de desarrollo. Asimismo, se describen las versiones empleadas, las dependencias necesarias y los pasos generales para la instalación y puesta en marcha del entorno, garantizando que cualquier integrante del equipo pueda replicarlo con facilidad. Esta base común asegura la consistencia del desarrollo, reduce errores por incompatibilidad y mejora la eficiencia en el mantenimiento y evolución futura del sistema.

Tabla 11.

Herramientas utilizadas para la configuración del ambiente de desarrollo

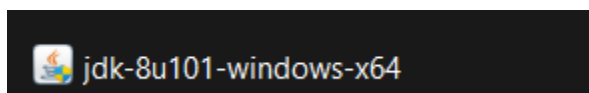
Tipo	Herramienta / Tecnología	Versión	Propósito
IDE	NetBeans	8.2	Desarrollo de aplicaciones Java EE, integración con JPA, JSP y Servlets
Servidor de aplicaciones	Apache Tomcat	9.0.98	Despliegue y ejecución del sistema web
Lenguaje de programación	Java	JDK 8	Desarrollo backend con Servlets, JPA, JDBC
Motor de base de datos	Microsoft SQL Server	2016 – 16.5.3	Almacenamiento y gestión de datos del sistema
Framework de persistencia	JPA	EclipseLink (JPA 2.1)	Mapeo objeto-relacional y acceso a la base de datos
Biblioteca de interfaz web	Bootstrap	5.2.0	Estilos responsivos y diseño de interfaz de usuario

La tabla muestra las principales herramientas y tecnologías utilizadas durante el desarrollo del módulo, especificando su tipo, versión y propósito dentro del sistema. Estas herramientas

fueron seleccionadas por su compatibilidad con la infraestructura existente, su estabilidad y soporte en entornos Java EE, permitiendo una integración eficiente entre la lógica de negocio, la interfaz de usuario y la persistencia de datos.

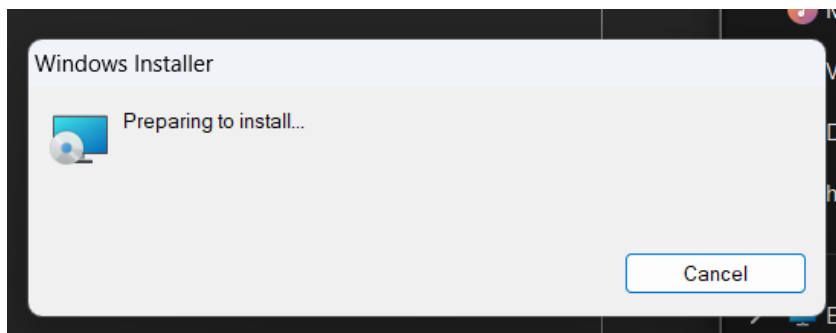
Instalador JDK

Figura 15



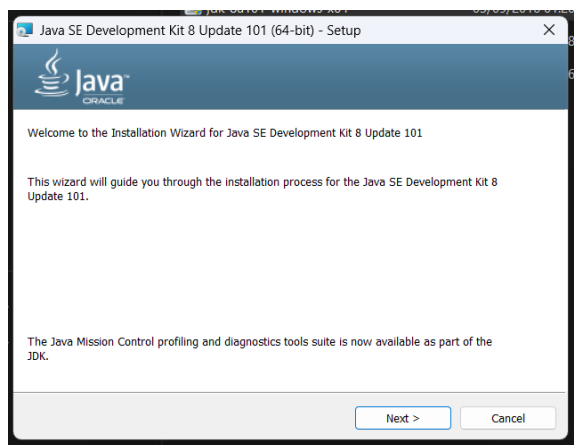
En la imagen se muestra del JDK utilizado para el desarrollo tanto del módulo a implementar, así como el resto del sistema.

Figura 16



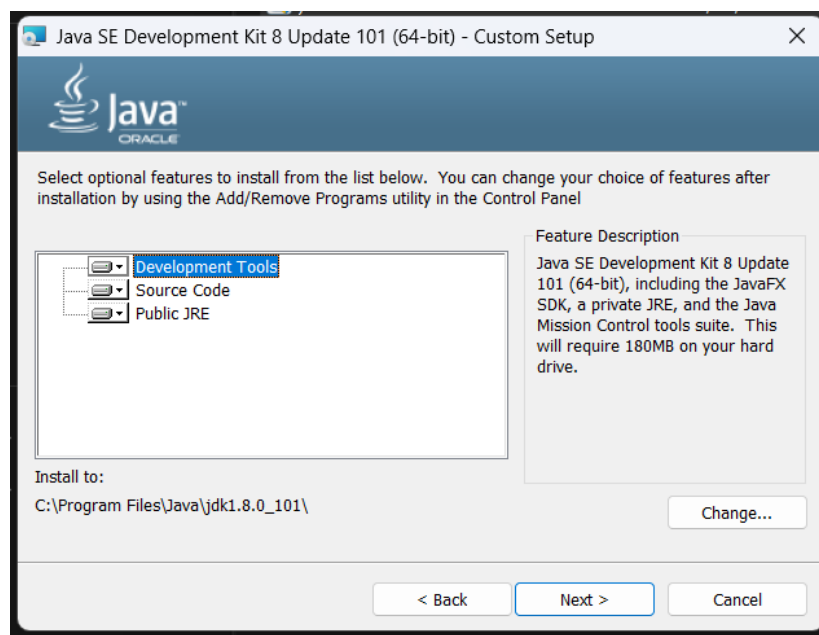
Captura de pantalla que muestra el paso 1 del proceso de instalación o configuración del JDK

Figura 17



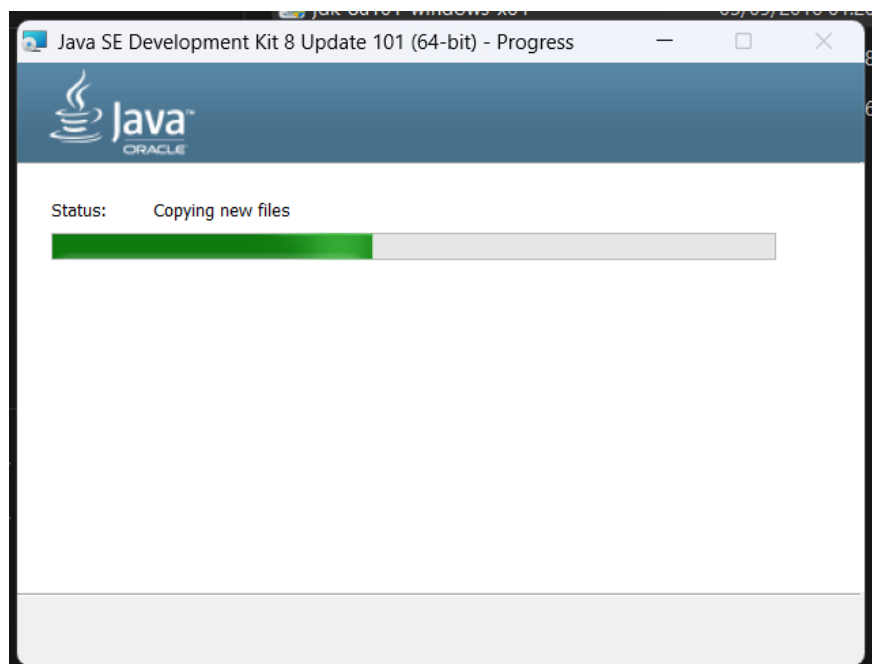
Proceso de instalación del JDK

Figura 18



Proceso de instalación del JDK

Figura 19



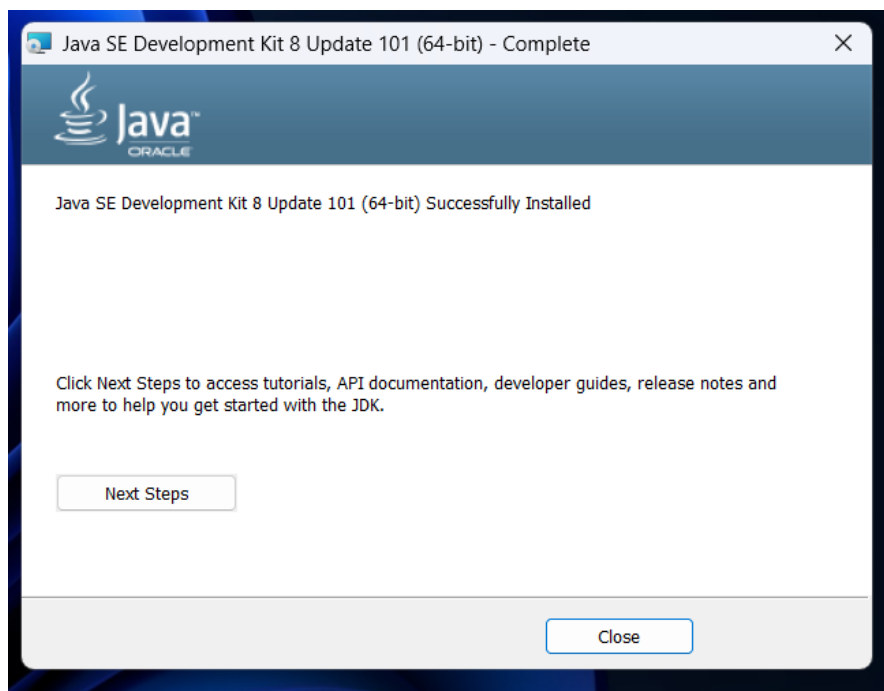
Captura de pantalla del proceso de instalación del JDK

Figura 20



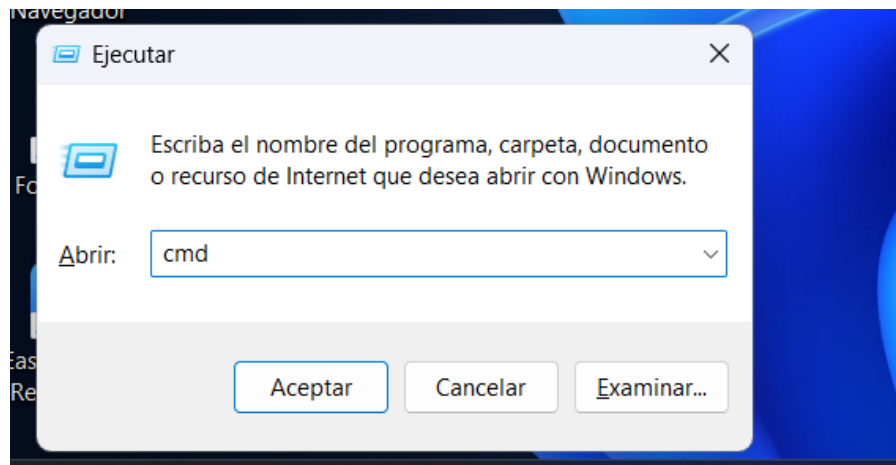
Captura de pantalla del proceso de instalación del JDK

Figura 21



Captura de pantalla del fin del proceso de instalación.

Figura 22



Para verificar que se instaló correctamente la versión de java a ocupar, abrimos la terminal de comandos de Windows con las teclas Windows + R y escribimos “CMD”

Figura 23

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Versión 10.0.26100.4652]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\aldri>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
```

Para ver la versión de Java tenemos que escribir lo siguiente. Java -versión.

Figura 24

Instalador de NetBeans versión 8.2

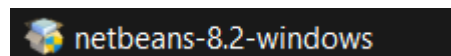
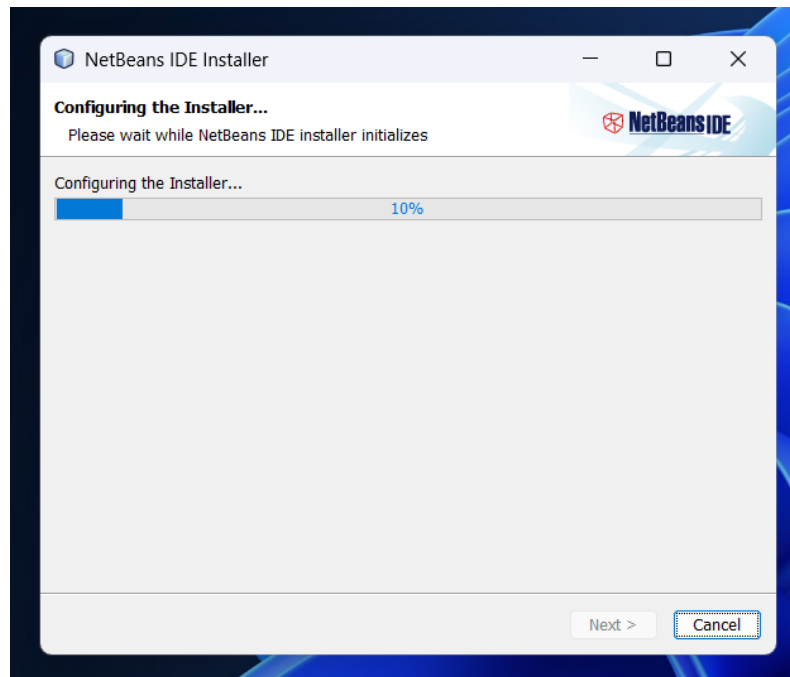
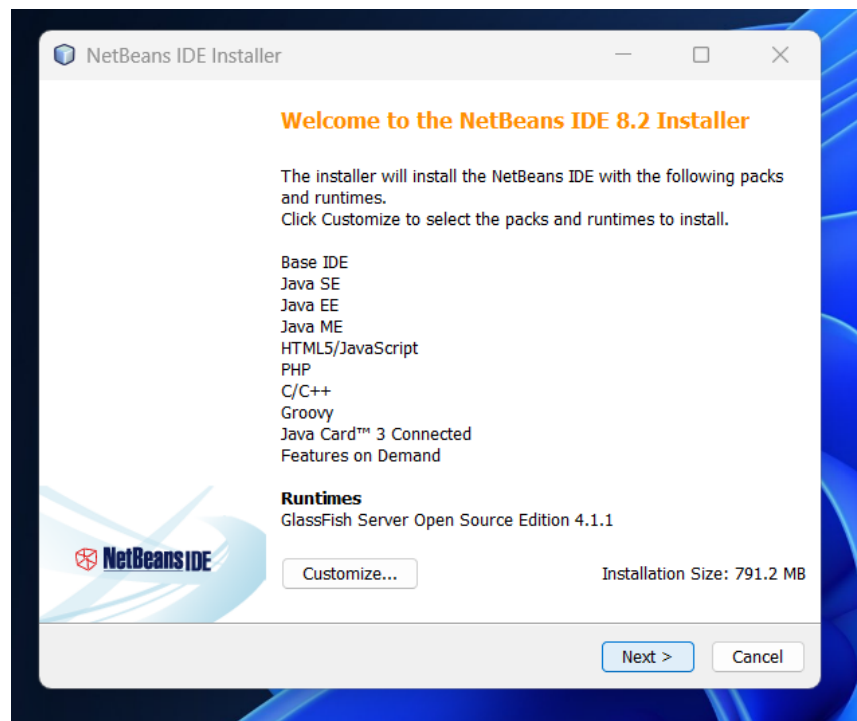


Figura 25



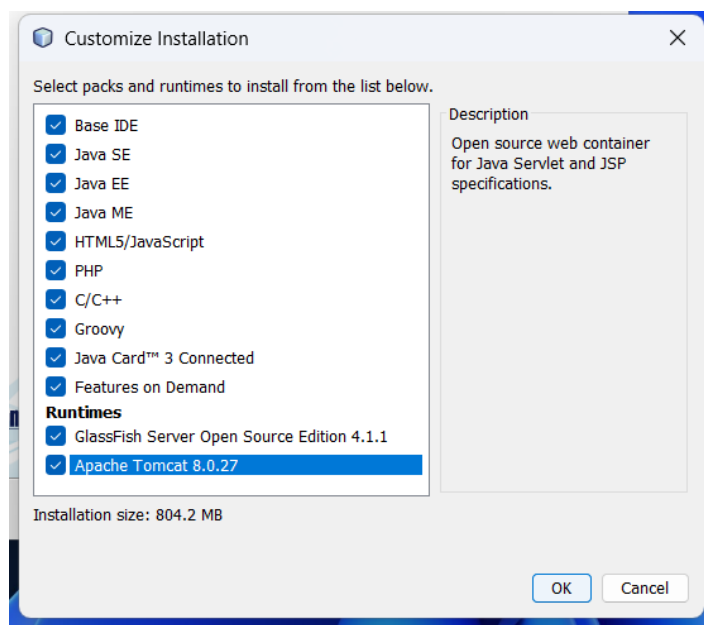
Progreso de instalación en curso, con barra de avance y detalles de los archivos que se están copiando al sistema.

Figura 26



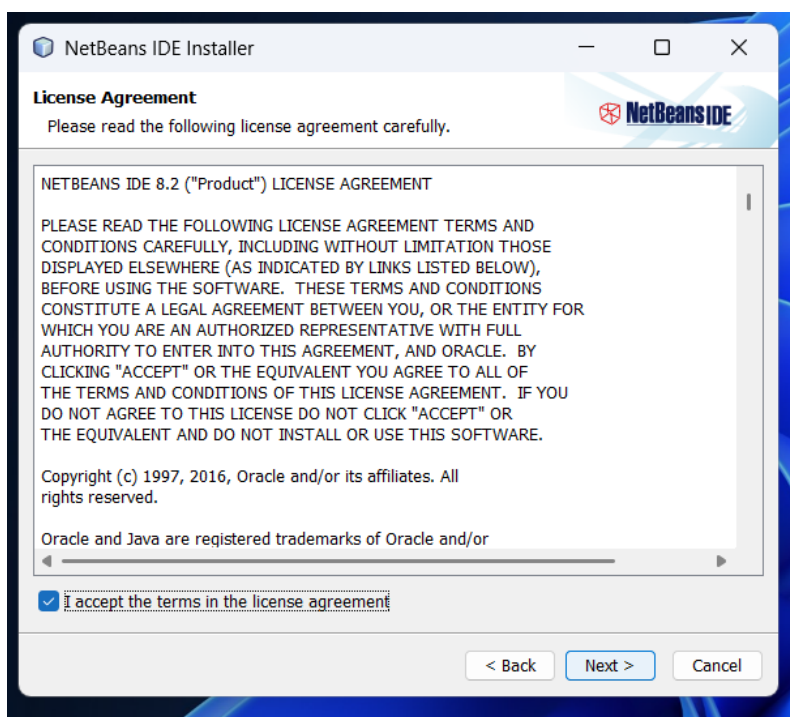
Pantalla de bienvenida una vez iniciado el proceso de instalación del IDE

Figura 27



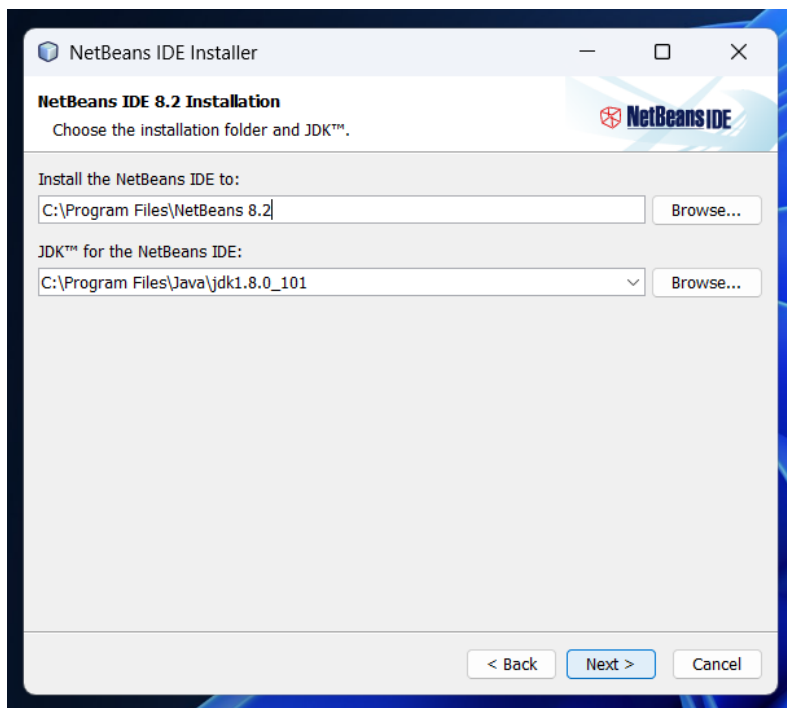
Ventana de selección de componentes a instalar, permitiendo elegir los módulos y herramientas adicionales que se integrarán con NetBeans.

Figura 28



Pantalla de aceptación de términos y condiciones de uso, donde el usuario debe marcar la casilla de conformidad antes de proceder.

Figura 29



Pantalla para definir la carpeta de instalación de NetBeans, con el campo de ruta editable y botón para explorar directorios.

Figura 30

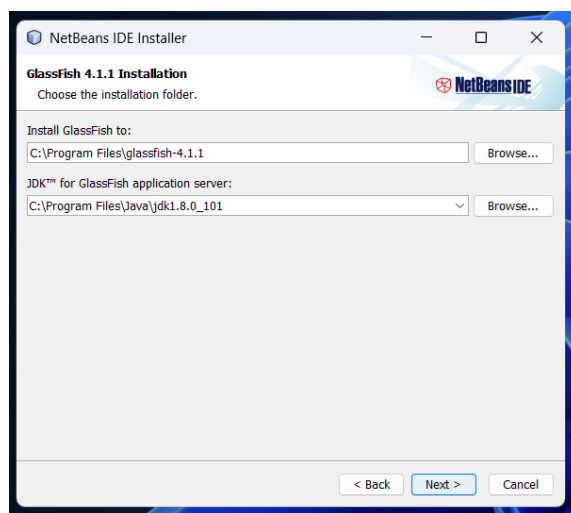
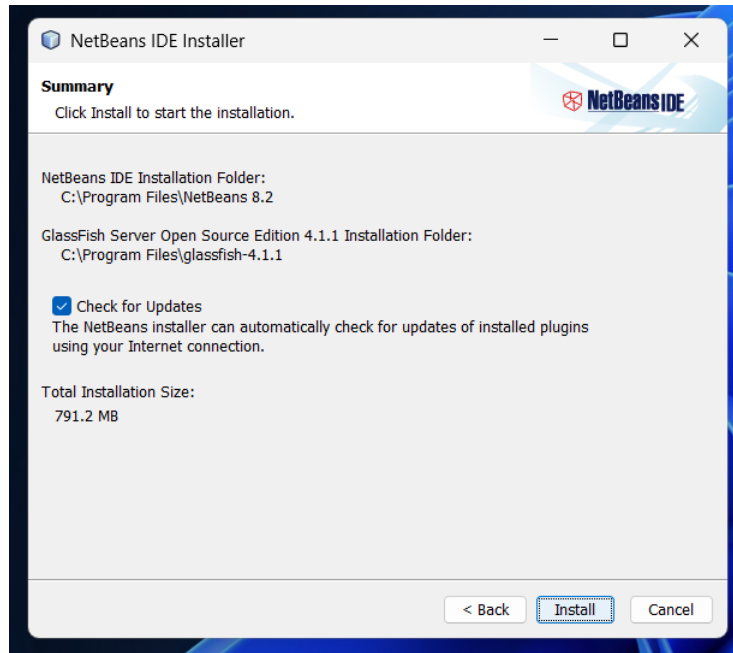


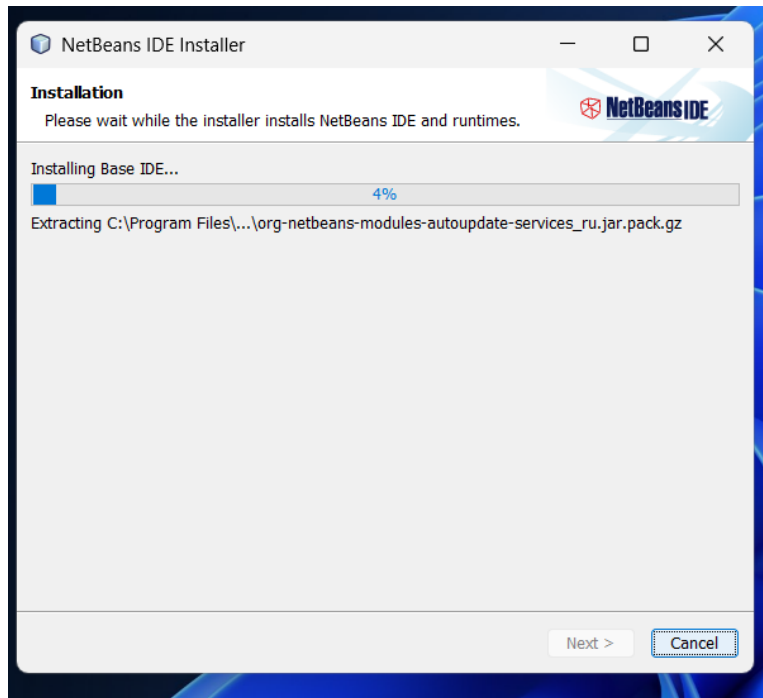
Figura 31

Ventana de selección de la carpeta de instalación para el JDK, necesaria para que NetBeans pueda compilar y ejecutar proyectos Java.



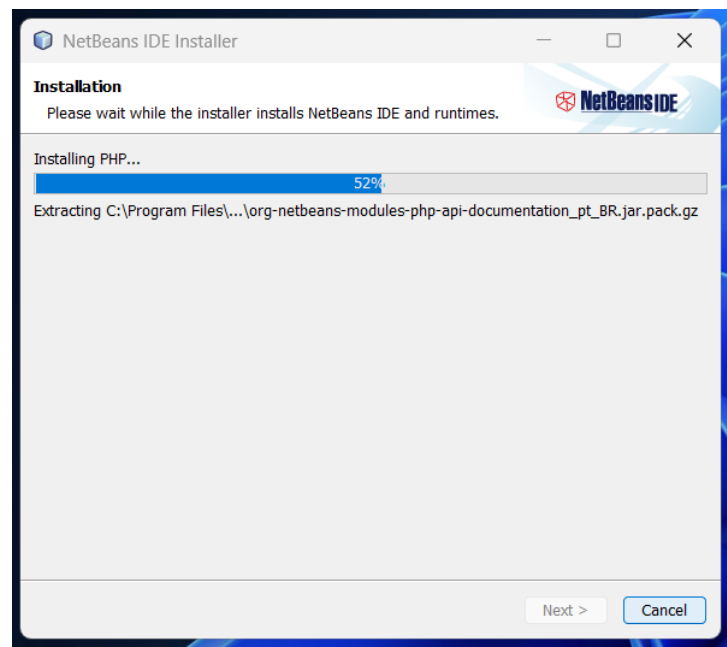
Proceso de instalación del IDE NetBeans para Java

Figura 32



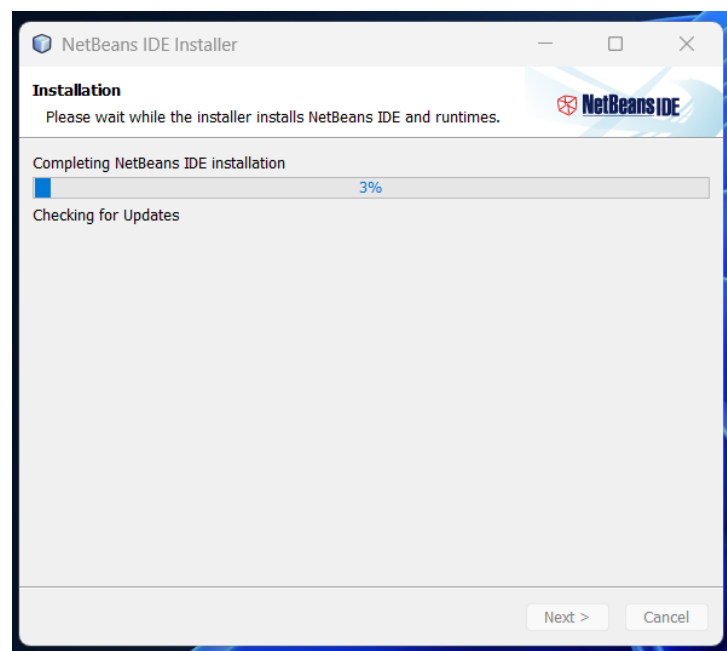
Proceso de instalación del IDE NetBeans para Java

Figura 33



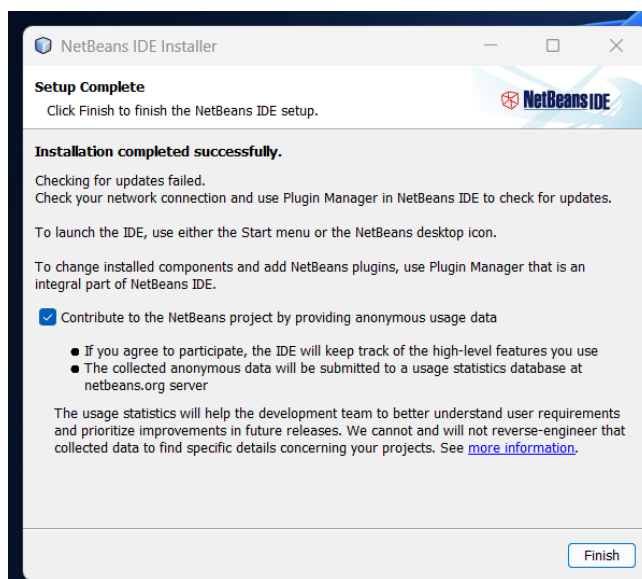
Proceso de instalación del IDE NetBeans para Java

Figura 34



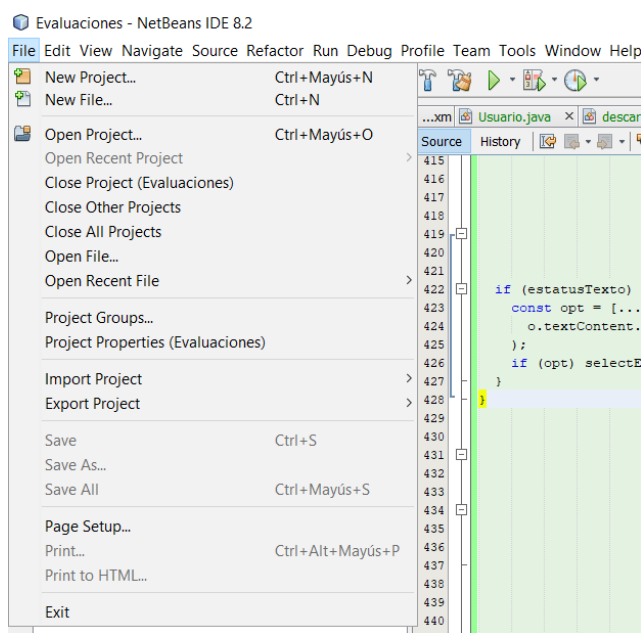
Progreso de instalación en curso, con barra de avance y detalles de los archivos que se están copiando al sistema.

Figura 35



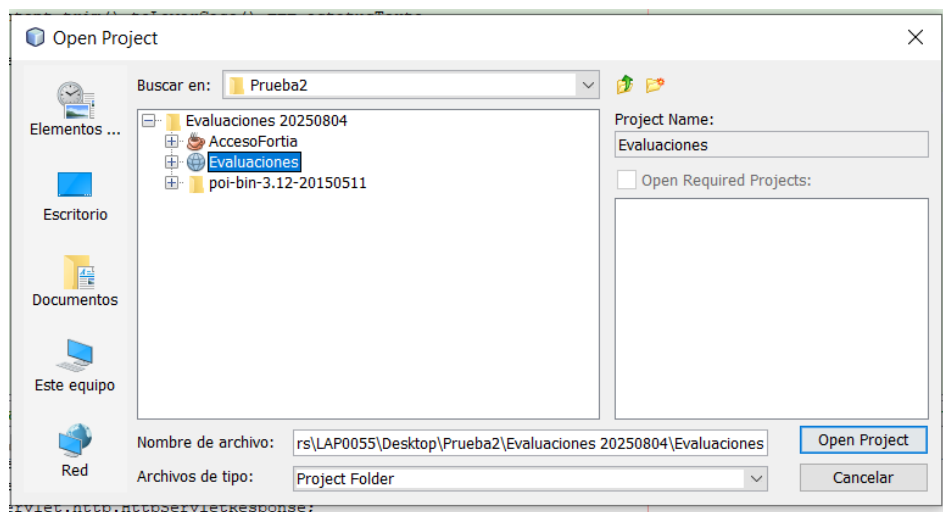
Ventana indicando la finalización exitosa de la instalación de NetBeans, con la opción de ejecutar el IDE inmediatamente.

Figura 36



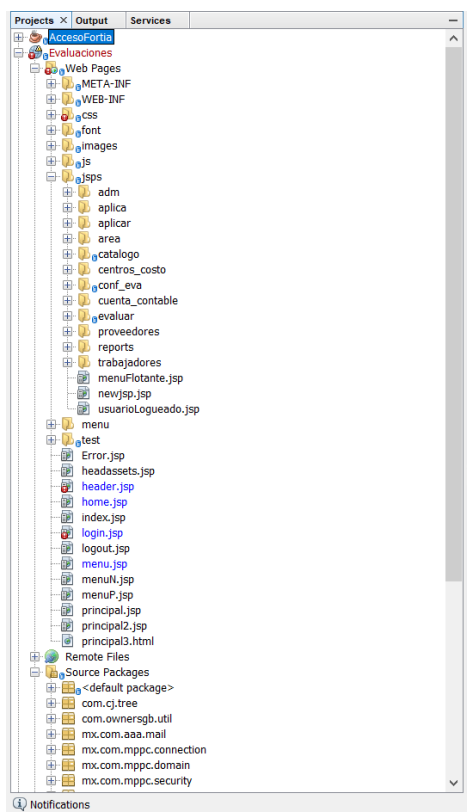
Ventana de opciones para crear un nuevo proyecto, en este caso se hizo una copia completa del proyecto que está en producción para realizar la implementación del nuevo módulo en un entorno de pruebas, basado en el entorno real.

Figura 37



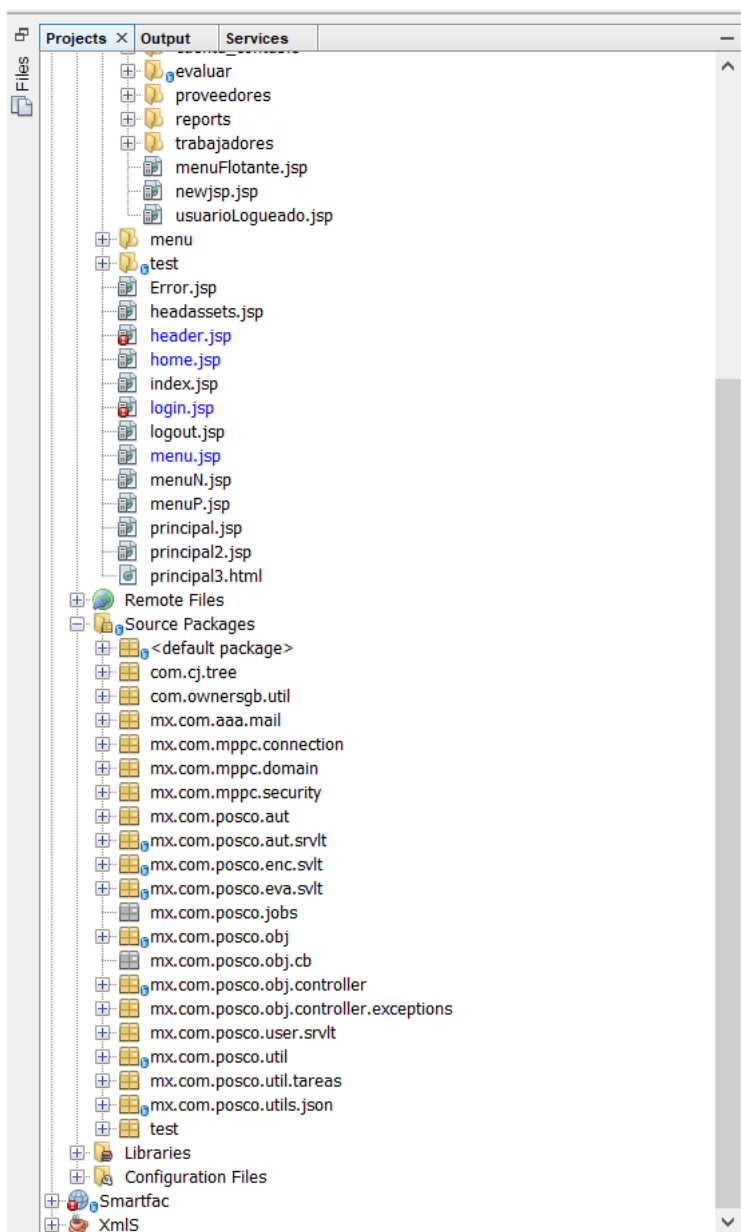
Al darle click en abrir, nos abre una ventana para buscar el proyecto sobre el cual trabajaremos, el proyecto esta nombrado como “Evaluaciones”, una vez identificado el proyecto ya solo le damos click en “Open Project”.

Figura 38



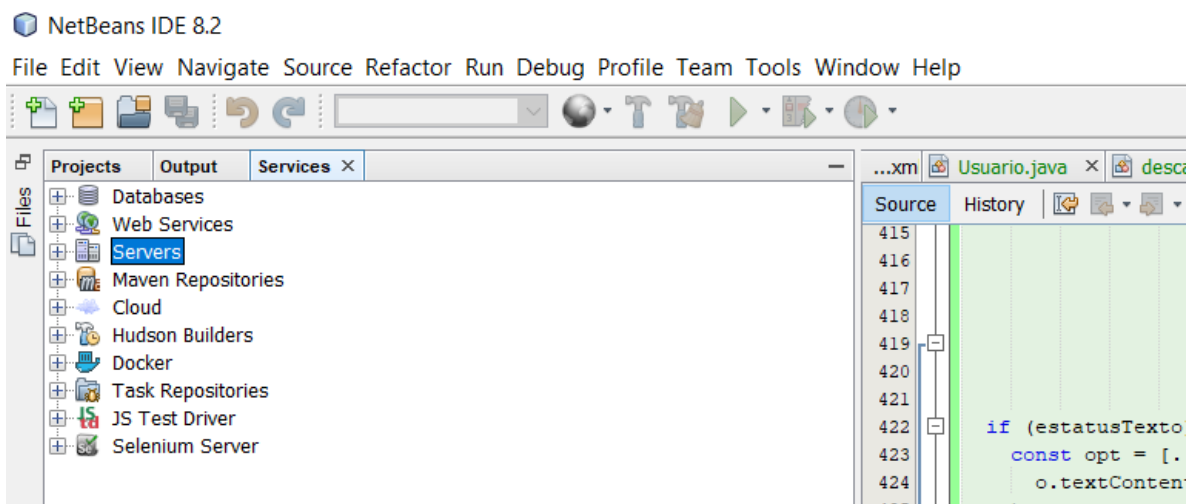
Captura de pantalla de la estructura de carpetas y archivos del proyecto

Figura 39



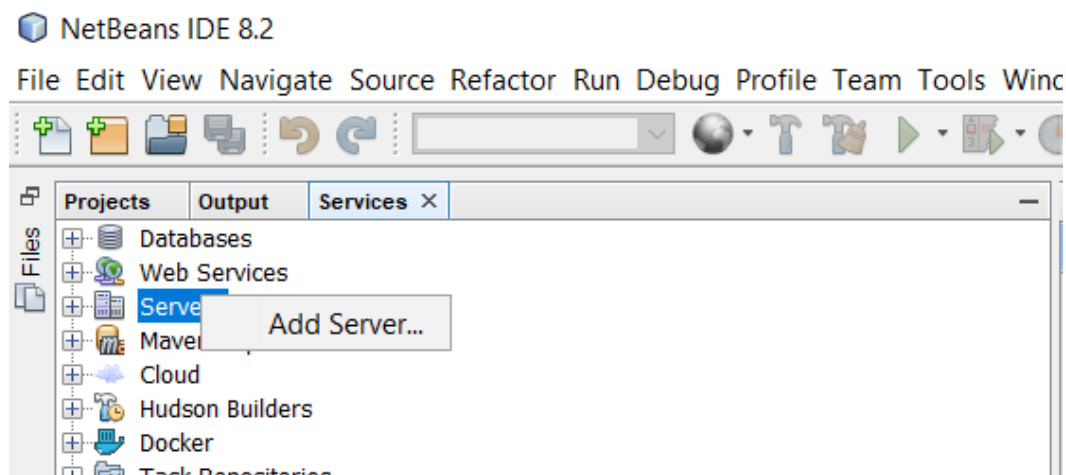
Continuación de imagen de la estructura de carpetas y archivos del proyecto.

Figura 40



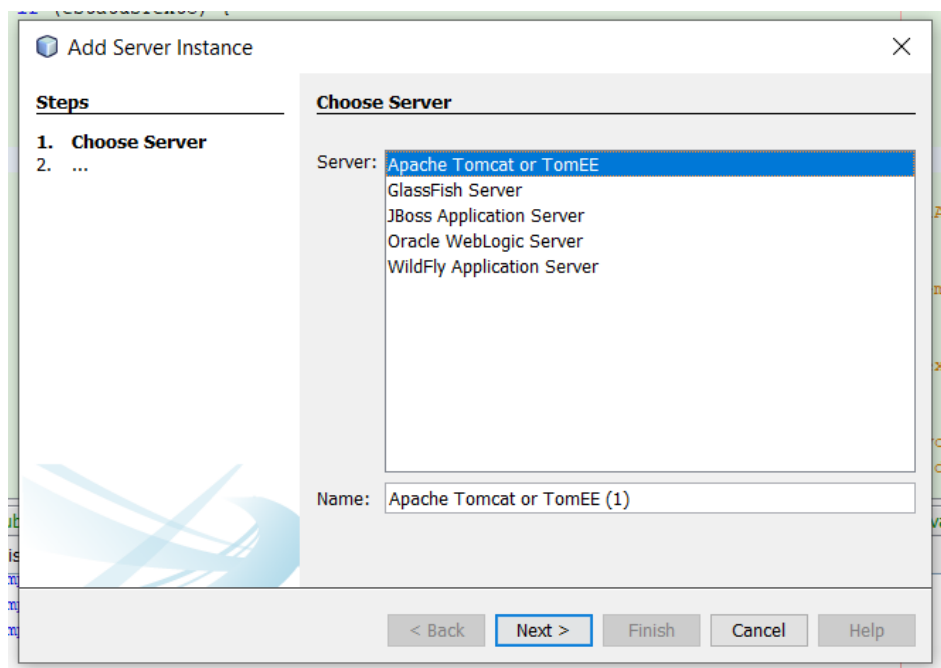
En el apartado de “Services” daremos de alta nuestro servidor el cual será Tomcat, con el cual estaremos corriendo nuestra aplicación web.

Figura 41



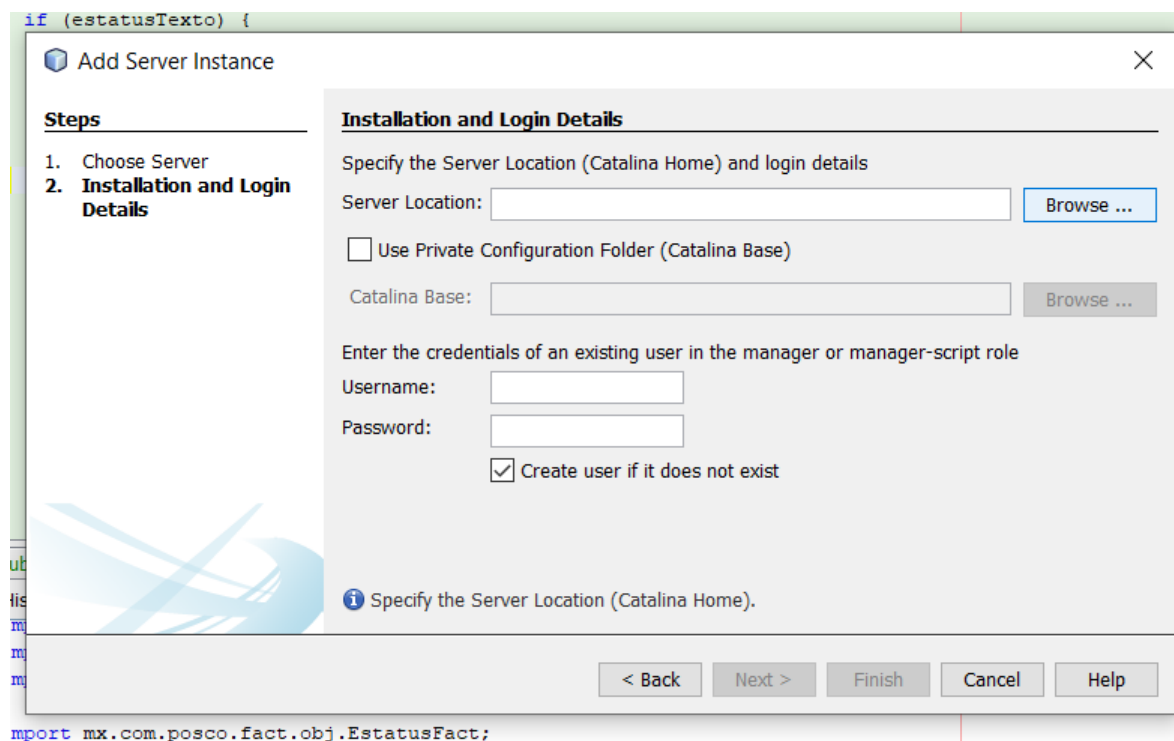
Daremos click en “Add Server”

Figura 42



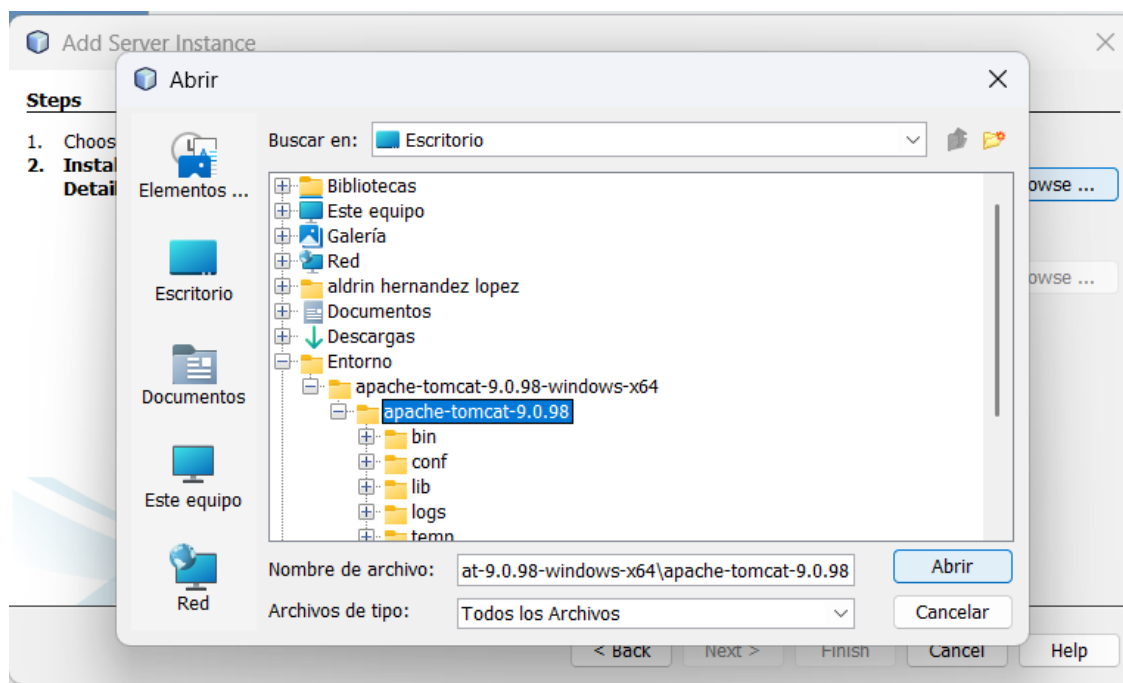
Como se muestra en la figura x seleccionamos Apache Tomcat

Figura 43



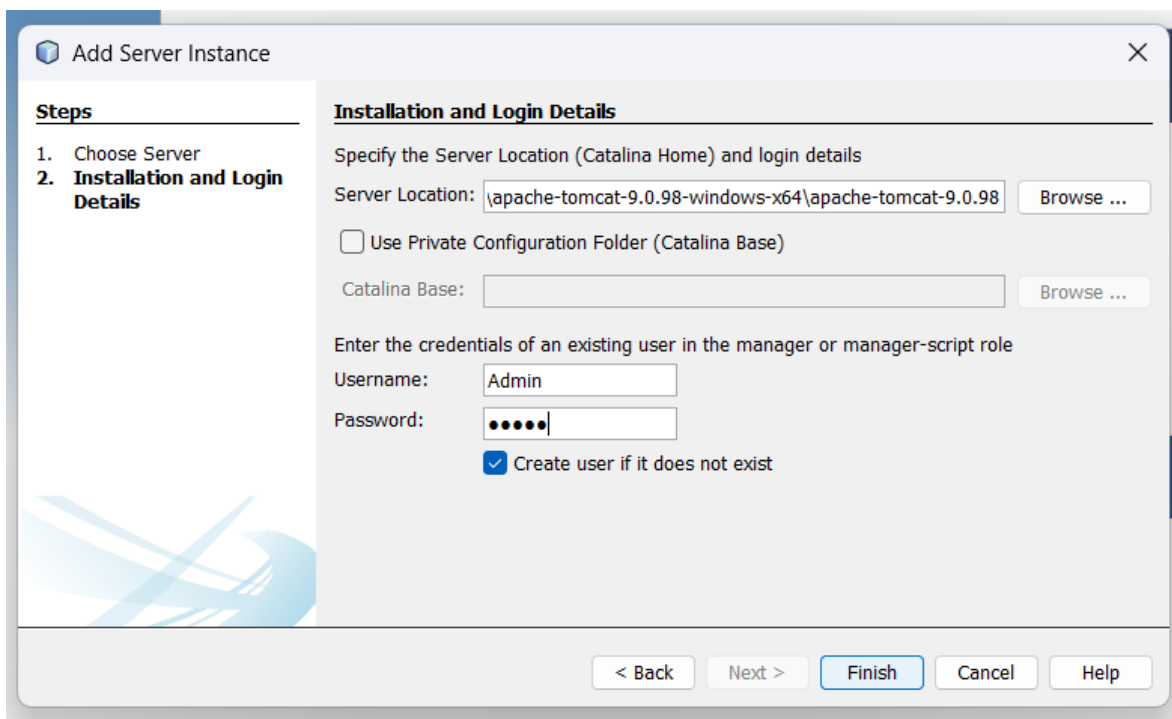
Nos saldrá de manera inmediata la imagen de la figura x donde tenemos que buscar nuestro servidor Tomcat

Figura 44



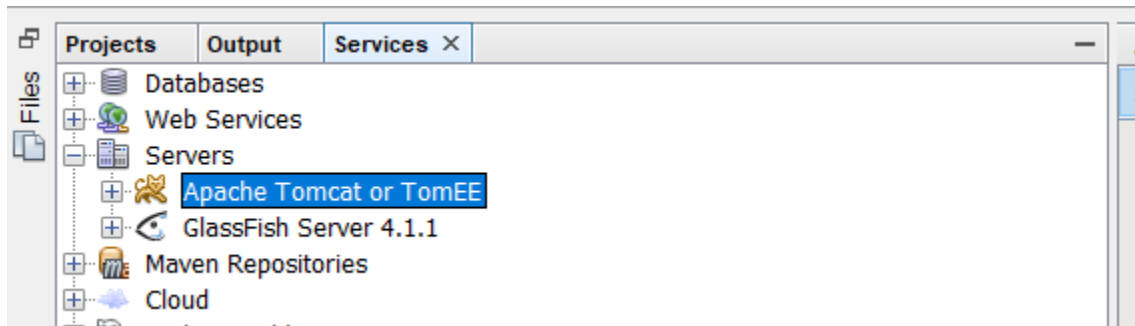
Al buscar las carpetas debemos de escoger la segunda la cual lleva el nombre de “apache-tomcat-9.0.98” y le daremos click en abrir.

Figura 45



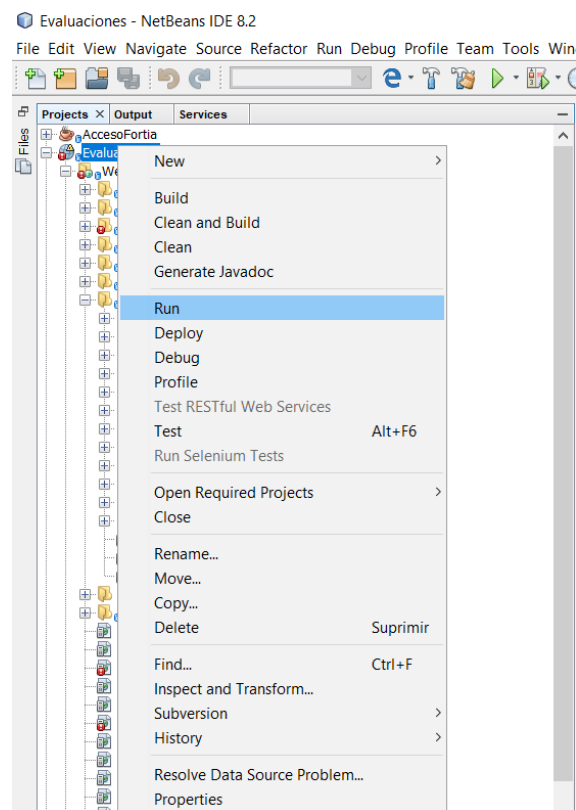
De inmediato nos regresara a esta pantalla, en la cual debemos de establecer un usuario y una contraseña, como observación en caso de que no exista un perfil, debes dejar marcado el Check para que nos cree nuestro perfil nuevo, por último, daremos click en Finish.

Figura 46



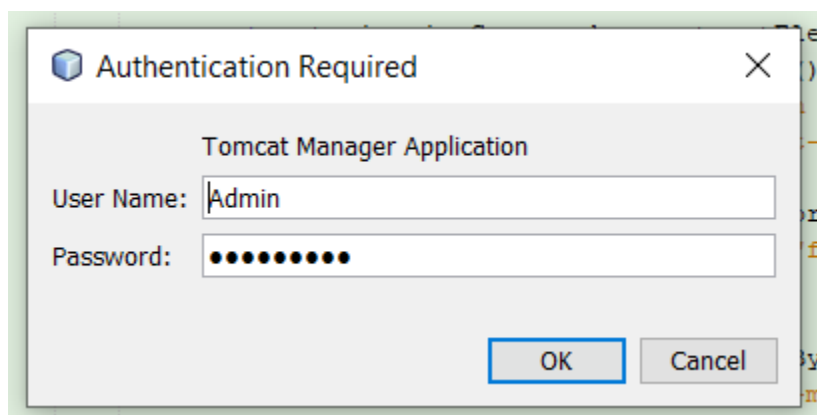
Una vez implementado el servidor, al desplegar el menú de Servers se debe de visualizar como en la figura 46.

Figura 47



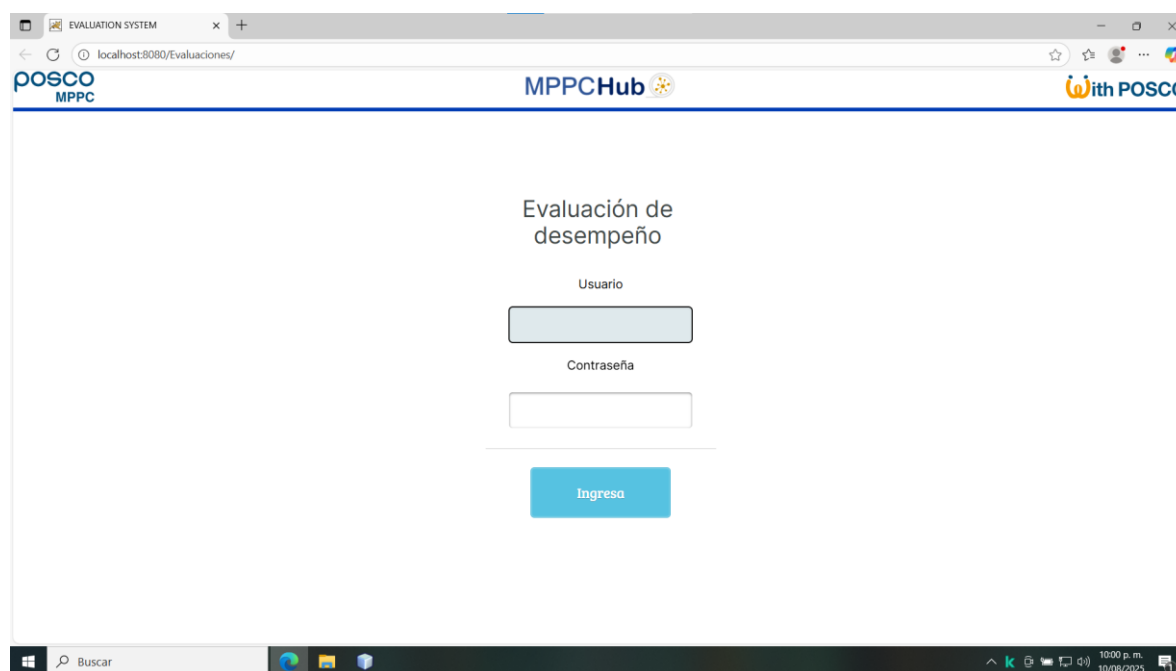
Para comprobar que el servidor esta correctamente instalado se le da click derecho encima del proyecto, en seguida se les desplegará un menú flotante donde daremos click en Run para desplegar el proyecto.

Figura 48



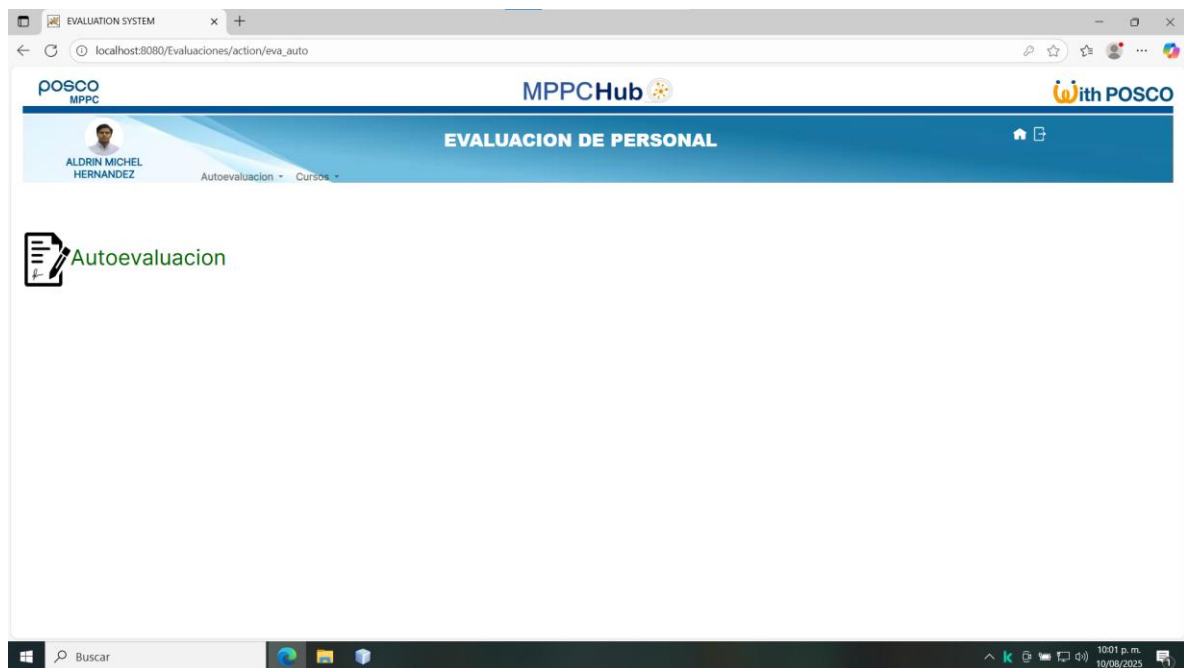
De manera inmediata se lanzará este mensaje donde tendremos que ingresar tanto el usuario como la contraseña que ingresamos antes como se muestra en la figura 45.

Figura 49



Login de usuarios

Figura 50



Home del sistema

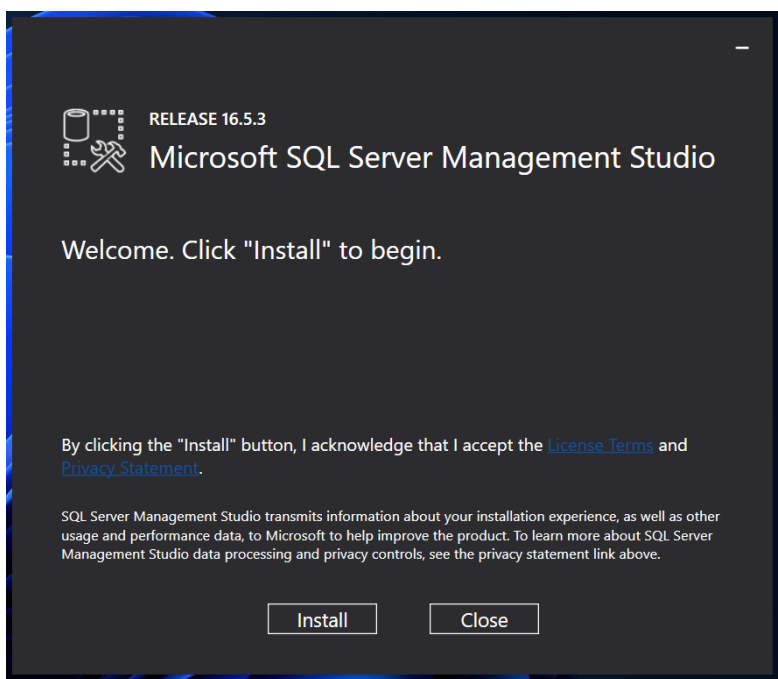
Preparación del motor de base de datos

Figura 51



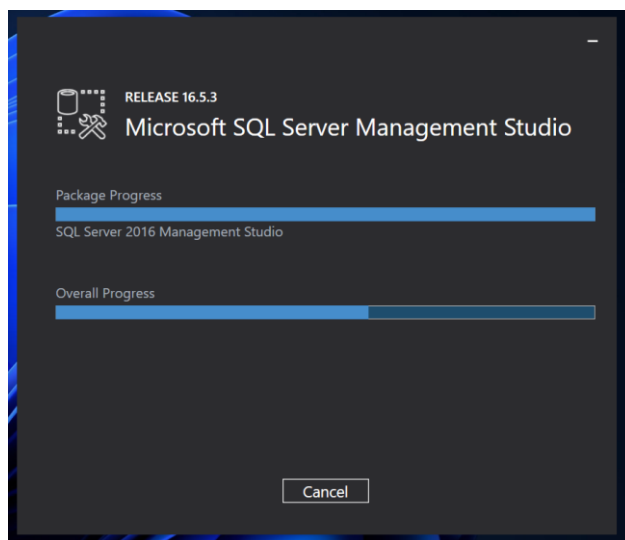
Imagen del instalador del motor de base de datos utilizado

Figura 52



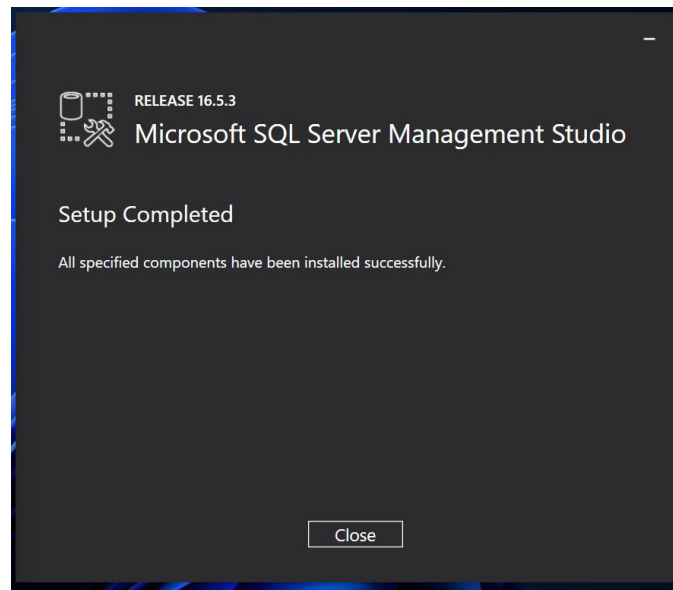
Interfaz inicial del instalador de SQL Server Management Studio (SSMS) 16.5.3, donde se inicia el asistente para la instalación y configuración del entorno de administración de bases de datos.

Figura 53



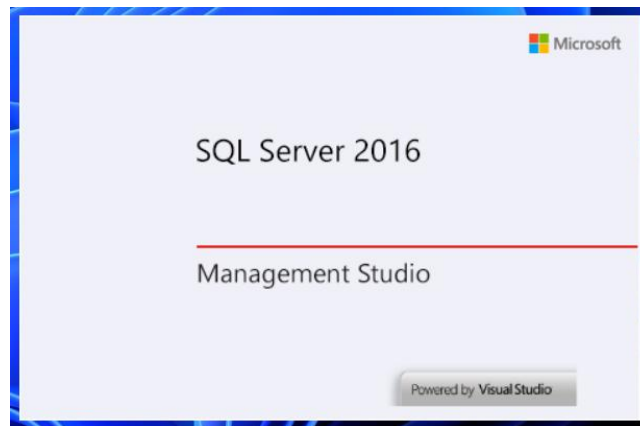
Pantalla de instalación de Microsoft SQL Server Management Studio (SSMS), mostrando la barra de progreso durante la carga de paquetes previos a la configuración del entorno.

Figura 54



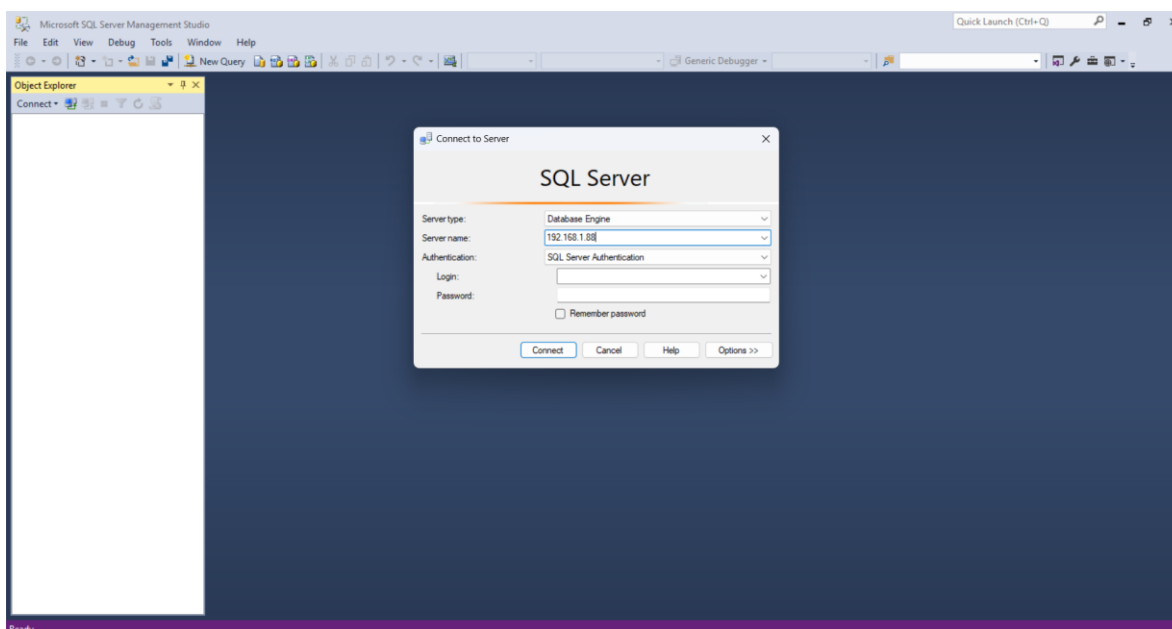
Pantalla final del instalador de Microsoft SQL Server Management Studio (SSMS), confirmando que todos los componentes se han instalado correctamente y el proceso ha finalizado con éxito.

Figura 55



Pantalla de inicio de Microsoft SQL Server Management Studio (SSMS) 2016, mostrando la interfaz de carga del entorno impulsado por Visual Studio antes de acceder a sus herramientas de administración de bases de datos.

Figura 56



Ventana de conexión de Microsoft SQL Server Management Studio (SSMS) mostrando la configuración para acceder a una instancia de SQL Server, con selección de tipo de servidor, nombre de host o dirección IP, método de autenticación y credenciales de inicio de sesión.

IV. ANÁLISIS DE RESULTADOS

El desarrollo del Módulo de Evaluación de Cursos se ejecutó bajo un enfoque iterativo empleando la metodología Kanban, lo que permitió una organización clara de las tareas, una priorización efectiva y un seguimiento constante del progreso del proyecto. Este enfoque facilitó la detección temprana de incidencias, la adaptación rápida a cambios en los requerimientos y la entrega continua de funcionalidades, manteniendo una comunicación fluida con las partes interesadas.

El módulo fue diseñado para cubrir de manera integral las necesidades de gestión y seguimiento de capacitaciones, incorporando funciones como la carga masiva de cursos desde archivos externos, la validación y almacenamiento en base de datos, la consulta filtrada por criterios como número de control, nombre de empleado o clave de curso, el monitoreo

de porcentajes de avance, así como la generación de reportes y estadísticas que facilitan la toma de decisiones.

Cada submódulo y flujo de trabajo se construyó con un enfoque en la usabilidad, la eficiencia y la integridad de los datos, asegurando que tanto administradores como empleados pudieran interactuar con el sistema de forma intuitiva y segura. Por motivos de confidencialidad y seguridad, en este documento se incluyen únicamente fragmentos representativos de código y las pantallas principales del sistema, con la información sensible debidamente censurada, garantizando la protección de datos corporativos y personales.

4.1. Entidades Mapeadas

Figura 57

Entidad mapeada de la tabla cursos

```

 * @author LAP0009
 */
@Entity
@Table(name = "Cursos")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Cursos.findAll", query = "SELECT c FROM Cursos c")
    , @NamedQuery(name = "Cursos.findByIdCurso", query = "SELECT c FROM Cursos c WHERE c.idCurso = :idCurso")
    , @NamedQuery(name = "Cursos.findBySite", query = "SELECT c FROM Cursos c WHERE c.site = :site")
    , @NamedQuery(name = "Cursos.findByNoControl", query = "SELECT c FROM Cursos c WHERE c.noControl = :noControl")
    , @NamedQuery(name = "Cursos.findByNomEmpleado", query = "SELECT c FROM Cursos c WHERE c.nomEmpleado = :nomEmpleado")
    , @NamedQuery(name = "Cursos.findByClaveCurso", query = "SELECT c FROM Cursos c WHERE c.claveCurso = :claveCurso")
    , @NamedQuery(name = "Cursos.findByNomCurso", query = "SELECT c FROM Cursos c WHERE c.nomCurso = :nomCurso")
    , @NamedQuery(name = "Cursos.findByInstructor", query = "SELECT c FROM Cursos c WHERE c.instructor = :instructor")
    , @NamedQuery(name = "Cursos.findByDuracion", query = "SELECT c FROM Cursos c WHERE c.duracion = :duracion")
    , @NamedQuery(name = "Cursos.findByFechaInicio", query = "SELECT c FROM Cursos c WHERE c.fechaInicio = :fechaInicio")
    , @NamedQuery(name = "Cursos.findByFechaFin", query = "SELECT c FROM Cursos c WHERE c.fechaFin = :fechaFin")
    , @NamedQuery(name = "Cursos.findByPorcentaje", query = "SELECT c FROM Cursos c WHERE c.porcentaje = :porcentaje")
    , @NamedQuery(name = "Cursos.findByEstatus", query = "SELECT c FROM Cursos c WHERE c.estatus = :estatus")
    , @NamedQuery(name = "Cursos.findByTipoCurso", query = "SELECT c FROM Cursos c WHERE c.tipoCurso = :tipoCurso")})
public class Cursos implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

```

La figura 57 muestra la entidad JPA Cursos, la cual está mapeada directamente a la tabla Cursos en la base de datos.

En el código se puede observar:

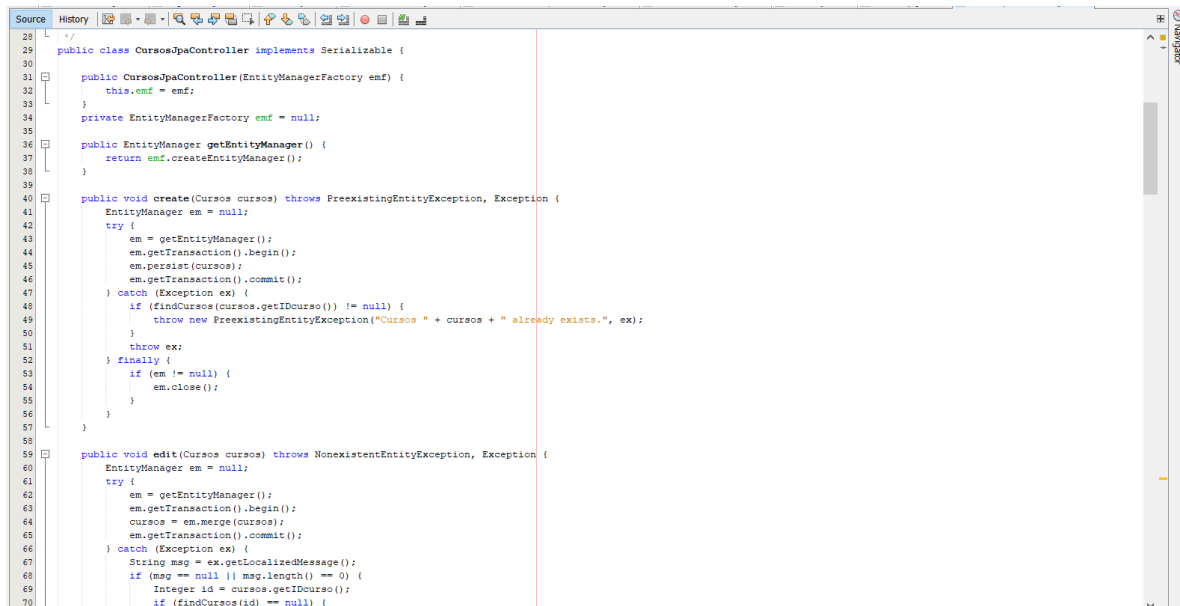
- La clase está anotada con `@Entity` y `@Table(name = "Cursos")`, lo que la define como una representación de la tabla homónima.
- Se utiliza la anotación `@XmlRootElement` para permitir que la entidad pueda ser transformada a XML si es necesario.

- Se definen múltiples consultas nombradas (@NamedQueries), que facilitan la recuperación de información de la tabla Cursos mediante criterios específicos como:
 - findAll: Recupera todos los registros.
 - findByIdCurso: Busca un curso por su identificador.
 - findByNoControl: Busca por número de control del empleado.
 - findByNomEmpleado: Filtra por nombre del empleado.
 - findByClaveCurso, findByNomCurso: Permiten ubicar un curso a partir de su clave o nombre.
 - Consultas adicionales como findByInstructor, findByDuracion, findByFechaInicio, findByFechaFin, findByPorcentaje, findByEstatus y findByTipoCurso permiten cubrir distintos criterios de búsqueda frecuentes dentro del sistema de gestión de capacitaciones.
- La clase implementa Serializable, lo cual garantiza que los objetos de tipo Cursos puedan ser transportados o almacenados de manera segura.
- También se declara un campo id como clave primaria con la anotación @Id y una estrategia de generación automática GenerationType.IDENTITY.

Esta entidad constituye el núcleo de la gestión de cursos dentro del sistema, ya que permite mapear cada registro de la base de datos a un objeto Java y facilita las operaciones de búsqueda, actualización y administración de la información de capacitación de los empleados.

Figura 58

JPA Controller de la Entidad Cursos



```
28
29 public class CursosJpaController implements Serializable {
30
31     public CursosJpaController(EntityManagerFactory emf) {
32         this.emf = emf;
33     }
34     private EntityManagerFactory emf = null;
35
36     public EntityManager getEntityManager() {
37         return emf.createEntityManager();
38     }
39
40     public void create(Cursos cursos) throws PreexistingEntityException, Exception {
41         EntityManager em = null;
42         try {
43             em = getEntityManager();
44             em.getTransaction().begin();
45             em.persist(cursos);
46             em.getTransaction().commit();
47         } catch (Exception ex) {
48             if (findCursos(cursos.getIdCurso()) != null) {
49                 throw new PreexistingEntityException("Cursos " + cursos + " already exists.", ex);
50             }
51             throw ex;
52         } finally {
53             if (em != null) {
54                 em.close();
55             }
56         }
57     }
58
59     public void edit(Cursos cursos) throws NonexistentEntityException, Exception {
60         EntityManager em = null;
61         try {
62             em = getEntityManager();
63             em.getTransaction().begin();
64             cursos = em.merge(cursos);
65             em.getTransaction().commit();
66         } catch (Exception ex) {
67             String msg = ex.getLocalizedMessage();
68             if (msg == null || msg.length() == 0) {
69                 Integer id = cursos.getIdCurso();
70                 if (findCursos(id) == null) {
```

Este controlador es fundamental dentro del sistema de gestión de capacitaciones, ya que abstrae las operaciones CRUD (crear, leer, actualizar, eliminar) sobre la tabla Cursos. Gracias a esta clase, se mantiene un acceso ordenado, seguro y reutilizable a la base de datos, lo que simplifica la lógica de negocio y fortalece la arquitectura del proyecto bajo el patrón Modelo-Vista-Controlador (MVC).

4.2. Página de Administrador

El módulo Administra Cursos del sistema de Evaluación de Personal está diseñado para facilitar la gestión de las capacitaciones asignadas a los empleados, integrando tanto la interfaz de usuario como la lógica de servidor. Este módulo permite que el personal autorizado cargue archivos en formato ODS o Excel (XLS/XLSX) que contienen información de cursos, los visualice en una tabla de previsualización y, posteriormente, los almacene de manera definitiva en la base de datos.

Al acceder al módulo, el usuario encuentra un panel lateral con sus datos personales y laborales como número de control, nombre, planta, departamento y nivel jerárquico lo que contextualiza la información de los cursos que se administran. En la parte central se presenta la sección “Lista de Cursos”, que incluye un formulario para la carga de archivos. Una vez que se selecciona un archivo, el sistema ofrece la opción de tabular los datos, es decir, procesar y mostrar en pantalla cada curso con sus atributos principales: sitio, número de control, empleado, clave del curso, nombre, instructor, duración, fechas de inicio y fin, porcentaje de avance y estado. Esta previsualización permite al usuario verificar que la información es correcta antes de confirmarla.

El proceso técnico de este módulo se sustenta en un servlet denominado Administra Cursos, el cual recibe y procesa las solicitudes provenientes de la interfaz. Cuando un archivo es enviado, el servlet identifica su formato y aplica el procedimiento correspondiente para extraer los datos: si se trata de un archivo ODS, se utiliza la librería ODF Toolkit; en el caso de Excel, se emplea Apache POI. En ambos escenarios, el sistema transforma cada fila del archivo en un objeto de tipo Curso y genera dinámicamente las filas HTML que se mostrarán en la tabla de la vista.

Posteriormente, si el usuario decide guardar la información, el servlet compara los cursos cargados con los ya existentes en la base de datos. Si detecta coincidencias en el número de control y la clave de curso, actualiza el estado y el porcentaje de avance; si no existen, los registra como nuevos. Esta lógica asegura que no se generen duplicados y que el progreso de cada capacitación se mantenga actualizado.

Finalmente, el módulo notifica al usuario mediante mensajes emergentes sobre el éxito o los errores ocurridos en el proceso de guardado. Con esta dinámica, Administra Cursos logra un equilibrio entre facilidad de uso y robustez técnica, garantizando que la información de las capacitaciones se mantenga organizada, actualizada y vinculada correctamente con cada empleado dentro del sistema.

Figura 59

Página de administrador

The screenshot displays the 'Administra Cursos' interface. On the left, a sidebar titled 'Información del Usuario' contains input fields for 'Número de Control', 'Nombre', 'Planta', 'Departamento', and 'Nivel'. The top navigation bar includes the 'posco MPPC' logo, 'MPPCHub', and 'With POSCO'. The main header is 'EVALUACION DE PERSONAL'. Below this, the 'Lista de Cursos' section features a file upload area with a 'Seleccionar archivo .ods' button, a 'Browse' button, and 'Tabular' and 'Guardar' buttons. A table titled 'Datos cargados' shows the following data:

Site	N° Control	Empleado	Clave Curso	Curso	Instructor	Duración	Fecha Inicio	Fecha Fin	% Avance	Estado
PUE	25006	Aldrin	P00-001	Extintores	Ing. Adrian	3 días	11/02/2025		null	

Módulo *Lista de Cursos* del sistema de Evaluación de Personal, mostrando la carga y tabulación de un archivo con información de capacitaciones. En la tabla se visualizan los datos procesados, incluyendo número de control, empleado, clave de curso, nombre, instructor, duración, fechas y avance del curso antes de ser guardados en la base de datos.

Figura 60

Servlet para la vista del módulo administrador

```
@WebServlet(name = "AdministraCursos", urlPatterns = {"/action/AdministraCursos"})
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 2, maxFileSize = 1024 * 1024 * 10, maxRequestSize = 1024 * 1024 * 50)
public class AdministraCursos extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        String action = request.getParameter("action");

        if ("guardar".equals(action)) {
            response.setContentType("application/json;charset=UTF-8");
            guardarCursos(request, response);
            return;
        }

        response.setContentType("text/html;charset=UTF-8");

        try {
            EntityManagerFactory emf2 = Persistence.createEntityManagerFactory("AccesoFortiaPU");
            RhTrabJpaController rtjc = new RhTrabJpaController(emf2);
            Integer use = request.getSession().getAttribute("use") != null ? Integer.parseInt(request.getSession().getAttribute(
                RhTrabNav t = rtjc.getEmpByNcNative(use + "");
            request.setAttribute("tra", t);
        } catch (NumberFormatException e) {
```

Fragmento del servlet *AdministraCursos*, encargado de gestionar las peticiones de la vista del módulo administrador. En este bloque se configuran los parámetros iniciales de la solicitud, se identifica la acción a ejecutar (como el guardado de cursos) y se establece la conexión con la unidad de persistencia.

4.3. Página de Usuario/Empleado

El módulo Cursos Pendientes dentro del sistema de Evaluación de Personal está diseñado para que los empleados puedan visualizar de manera clara y dinámica los cursos que tienen asignados y el avance que registran en ellos. Este módulo combina una interfaz intuitiva con un backend que gestiona la recuperación de datos desde la base de datos de manera filtrada y actualizada.

En la vista JSP, el usuario encuentra en el panel lateral su información básica como número de control, nombre, planta, departamento y nivel. Además, este panel incluye un gráfico circular elaborado con Chart.js que muestra el avance general de los cursos asignados, indicando el número de cursos completados frente a los pendientes, acompañado de un porcentaje global de progreso.

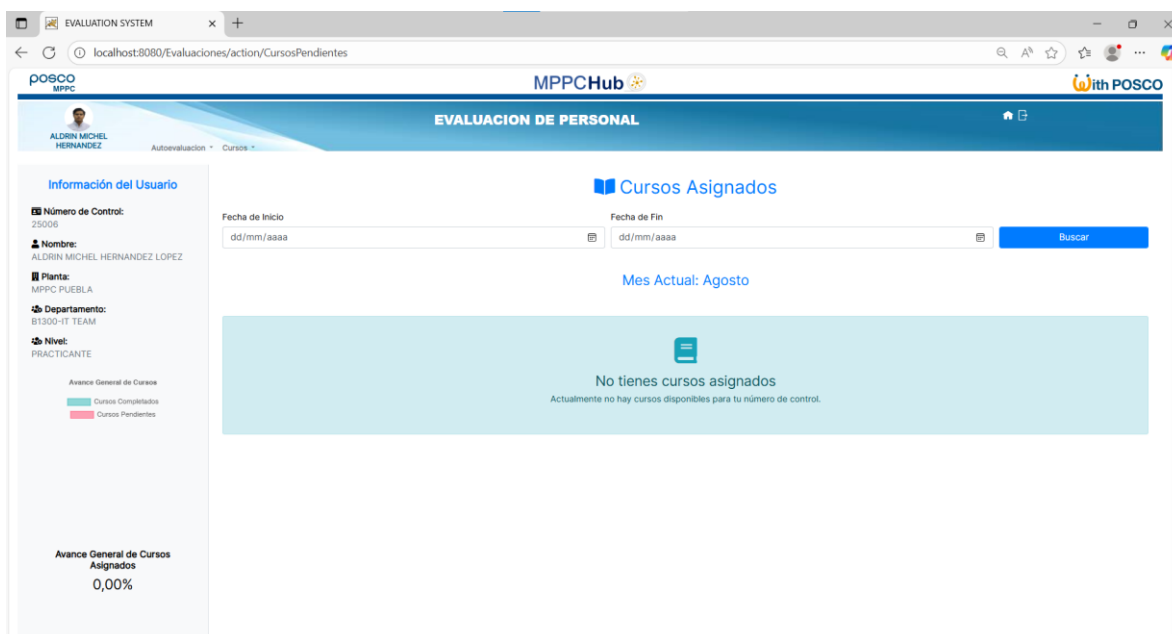
En la parte central, el sistema despliega un título principal “Cursos Asignados” y un formulario que permite filtrar los cursos por rango de fechas de inicio y fin. De manera predeterminada, si el usuario no especifica un rango, se muestran los cursos correspondientes al mes actual, cuyo nombre se presenta de forma destacada en la interfaz.

Los cursos se listan en forma de tarjetas (cards de Bootstrap), cada una con un encabezado de color que varía en función de la clave del curso, lo que permite identificar rápidamente la categoría a la que pertenece (organizacionales, seguridad, legales/financieros, calidad, entre otros). Dentro de cada tarjeta se incluyen detalles como la clave del curso, el nombre del instructor, la duración, las fechas de inicio y fin, así como una barra de progreso que indica el porcentaje de avance. En la parte inferior de cada tarjeta se muestra el estado del curso mediante insignias de color (por ejemplo, verde para cursos completados y amarillo para pendientes). En caso de que el empleado no tenga cursos asignados, el sistema muestra un mensaje informativo con un diseño claro y accesible.

En cuanto a la lógica del backend, esta está implementada en el servlet CursosPendientes. Su función principal es procesar las solicitudes del usuario, recuperar de la sesión el número de control y con ello identificar al empleado en la base de datos. A partir de esta información, el servlet consulta los cursos asignados al usuario, ya sea dentro del mes actual o dentro de un rango de fechas especificado en el formulario de búsqueda.

El servlet además calcula estadísticas: contabiliza cuántos cursos han sido completados y cuántos permanecen pendientes, con lo cual obtiene un porcentaje de avance general que se envía a la vista junto con la lista de cursos. De esta forma, la JSP puede renderizar tanto el gráfico de progreso como las tarjetas con la información detallada de cada curso. Finalmente, en caso de que ocurran errores durante el proceso (como problemas al parsear fechas o recuperar datos), el sistema devuelve mensajes claros que se muestran al usuario.

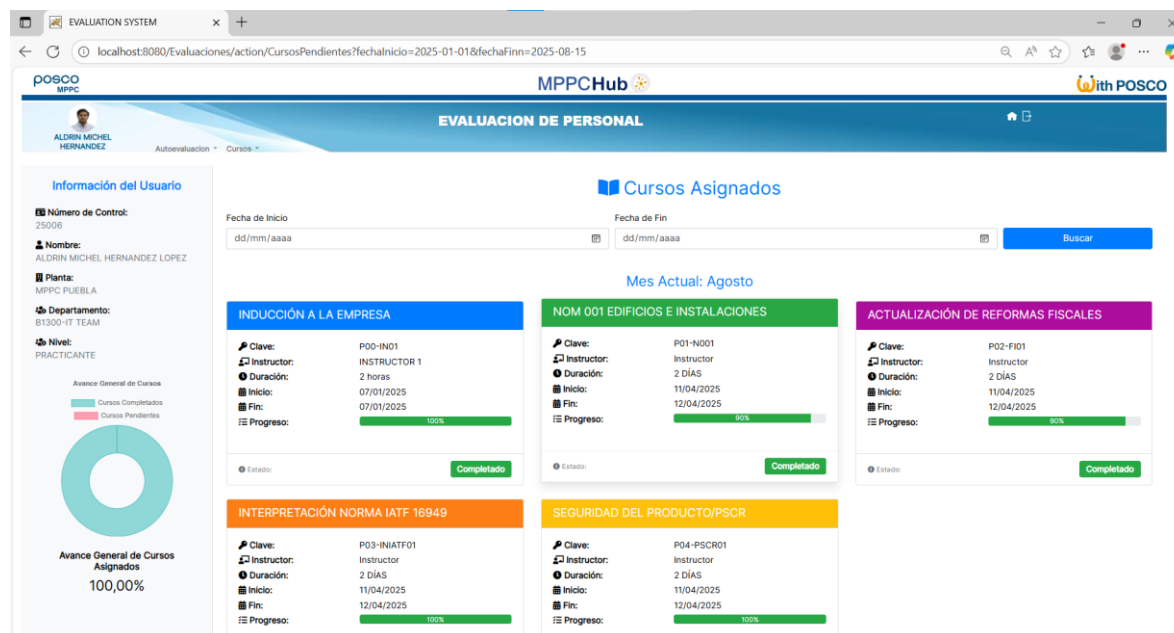
Figura 61



Vista del panel cuando el usuario no tiene cursos asignados. En lugar de mostrar tarjetas de capacitación, el sistema despliega un mensaje informativo que indica la ausencia de cursos disponibles. De forma predeterminada, la consulta se realiza sobre el mes en curso, por lo que, si no existen registros en ese periodo, la pantalla se mantiene vacía salvo por la notificación al usuario.

Figura 62

Dashboard de Empleados con capacitaciones asignadas



Vista del panel principal cuando el empleado tiene cursos asignados. La información se organiza en tarjetas que muestran los detalles de cada capacitación y se complementa con indicadores visuales de estado. En este ejemplo, los datos se presentan aplicando filtros de búsqueda por rangos de fechas, lo que permite al usuario consultar únicamente los cursos programados dentro del periodo seleccionado en la JSP de la página

Figura 63

```

106 <p class="mb-1"><i class="fas fa-users-cog"></i> <strong>Nivel:</strong></p>
107 <p class="text-muted">${tra.NOM_TAB_SUE}</p>
108 </div>
109
110 <!-- Gráfico para mostrar el avance general -->
111 <div class="mt-4">
112 <canvas id="avanceGeneralChart" width="300" height="300"></canvas>
113 </div>
114
115 <!-- Mostrar el porcentaje general debajo del gráfico -->
116 <div class="text-center mt-4">
117 <h4 class="font-weight-bold">Avance General de Cursos Asignados</h4>
118 <p class="h4">
119 <fmt:formatNumber value="${avanceGeneral}" pattern="#0.00"/> <!-- Se tiene que formatear el numero para que no provoque error -->
120 </p> <!-- Muestra el porcentaje con dos decimales -->
121 </div>
122 </div>
123
124 <!-- Contenido principal -->
125 <div class="col-md-9 col-lg-10 p-4">
126 <h2 class="mb-4 text-center text-primary"><i class="fas fa-book-open mr-2"></i>Cursos Asignados</h2>
127
128 <!-- Formulario de búsqueda de fechas -->
129 <form method="GET" action="${pageContext.request.contextPath}/action/CursosPendientes" class="mb-4">
130 <div class="form-row">
131 <div class="form-group col-md-5">
132 <label for="fechaInicio">Fecha de Inicio</label>
133 <input type="date" class="form-control" id="fechaInicio" name="fechaInicio" required>
134 </div>
135 <div class="form-group col-md-5">
136 <label for="fechaFin">Fecha de Fin</label>
137 <input type="date" class="form-control" id="fechaFin" name="fechaFin" required>
138 </div>
139 <div class="form-group col-md-2">
140 <label><input type="submit" class="btn btn-primary form-control">Buscar</button>
141 </div>
142 </div>
143 </form>
144
145 <!-- Mostrar el mes actual -->
146 <
147

```

Fragmento del código fuente de la vista JSP *CursosPendientes*, donde se define la sección principal del módulo. En este bloque se incluye el gráfico de avance general de cursos con su porcentaje, el encabezado de 'Cursos Asignados' y el formulario de búsqueda que permite filtrar los resultados por rangos de fechas de inicio y fin, facilitando la consulta personalizada de las capacitaciones del empleado.

Figura 64

Servlet para la vista de cursos pendientes

```

31  *
32  * @author LAP0009
33  */
34  @WebServlet(name = "CursosPendientes", urlPatterns = {"/action/CursosPendientes"})
35  public class CursosPendientes extends HttpServlet {
36
37      private static final Logger LOG = Logger.getLogger(CursosPendientes.class.getName());
38
39      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
40          throws ServletException, IOException {
41          response.setContentType("text/html;charset=UTF-8");
42          request.setCharacterEncoding("UTF-8");
43
44          EntityManagerFactory emfCursos = null;
45          EntityManagerFactory emfRH = null;
46
47          try {
48              SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
49              EntityManagerFactory emf2 = Persistence.createEntityManagerFactory("AccesoFortisAPU");
50              RHTrabajaPaController rtjc = new RHTrabajaPaController(emf2);
51              Integer use = request.getSession().getAttribute("use") != null ? Integer.parseInt(request.getSession().getAttribute("use").toString()) : null;
52              RHTrabajaPaController rtjc = rtjc.getEmpByNoMatr(use + "");
53              request.setAttribute("tra", t);
54          } catch (NumberFormatException e) {
55              request.setAttribute("error", "Error al procesar la solicitud: " + e.getMessage());
56          }
57
58          try {
59              // Obtener el número de control del usuario desde la sesión
60              Integer noControlUsuario = (Integer) request.getSession().getAttribute("use");
61
62              if (noControlUsuario == null) {
63                  response.sendRedirect(request.getContextPath() + "/login.jsp");
64                  return;
65              }
66
67              // Obtener cursos del usuario
68              emfCursos = Persistence.createEntityManagerFactory("EvaluacionesAPU");
69              CursosPaController cursosController = new CursosPaController(emfCursos);
70
71              // Obtener la fecha actual
72              Calendar calendar = Calendar.getInstance();
73              calendar.set(Calendar.DAY_OF_MONTH, 1); // Primer día del mes

```

Fragmento del servlet *CursosPendientes*, encargado de gestionar la lógica del módulo de cursos pendientes. En este bloque se observa la configuración inicial del servlet, la definición del método `processRequest` y las primeras operaciones: establecer la codificación de la petición, recuperar la información del usuario autenticado desde la sesión y preparar la conexión con la unidad de persistencia. Este servlet permite filtrar los cursos asignados según el rango de fechas o, por defecto, mostrar los correspondientes al mes actual.

Figura 65

```

127  RHTrabajaPaController rhController = new RHTrabajaPaController(emfRh);
128  RHTrabajaPaController rhController = rhController.getEmpByNoMatr(noControlUsuario.toString());
129
130  if (noMatr != null) {
131      request.setAttribute("tra", trabajador);
132  } else {
133      LOG.log(Level.SEVERE, "No se encontró información para el usuario: {0}", noControlUsuario);
134  }
135
136  } catch (NumberFormatException | ParseException e) {
137      LOG.log(Level.SEVERE, "Error en el proceso de curso pendientes", e);
138      request.setAttribute("error", "Error al cargar los cursos. Detalles: " + e.getMessage());
139  } catch (Exception e) {
140      LOG.log(Level.SEVERE, "Error en el proceso de curso pendientes", e);
141      request.setAttribute("error", "Error al cargar los cursos. Detalles: " + e.getMessage());
142  } finally {
143      // Cerrar las EntityManagerFactory
144      if (emfCursos != null) {
145          emfCursos.close();
146      }
147      if (emfRh != null) {
148          emfRh.close();
149      }
150
151      // Redirigir al JSP
152      request.getRequestDispatcher("/jsp/evaluacion/CursosPendientes.jsp").forward(request, response);
153  }
154
155  @Override
156  protected void doGet(HttpServletRequest request, HttpServletResponse response)
157      throws ServletException, IOException {
158      processRequest(request, response);
159  }
160
161  @Override
162  protected void doPost(HttpServletRequest request, HttpServletResponse response)
163      throws ServletException, IOException {
164      processRequest(request, response);
165  }
166
167  @Override
168  public String getServletInfo() {
169      return "Servlet de cursos pendientes";

```

Sección final del servlet *CursosPendientes*, donde se completan las operaciones de validación y cierre de recursos. En este fragmento se incluyen el manejo de excepciones, el cierre de las fábricas de entidades (*EntityManagerFactory*) y el reenvío de la petición a la vista *CursosPendientes.jsp*. Asimismo, se definen los métodos *doGet* y *doPost* para redirigir las solicitudes HTTP al método *processRequest*, centralizando así la lógica del procesamiento de cursos pendientes.

4.4. Página de Búsqueda de capacitaciones asignadas

El módulo Estadísticas de Cursos permite consultar, analizar y actualizar la información de capacitaciones registradas en el sistema mediante filtros y una visualización tabular clara. Al ingresar, el usuario encuentra un panel lateral con sus datos básicos (número de control, nombre, planta, departamento y nivel), lo que contextualiza la consulta y mantiene coherencia con el resto del sistema.

En el área principal se presenta el título “Gestión de Cursos” y un formulario de búsqueda que admite tres criterios: número de control, nombre del empleado y clave de curso. Estos filtros son opcionales y pueden combinarse para refinar los resultados. La interfaz, construida con Bootstrap, ofrece controles de formulario claros y acciones de Buscar y Limpiar.

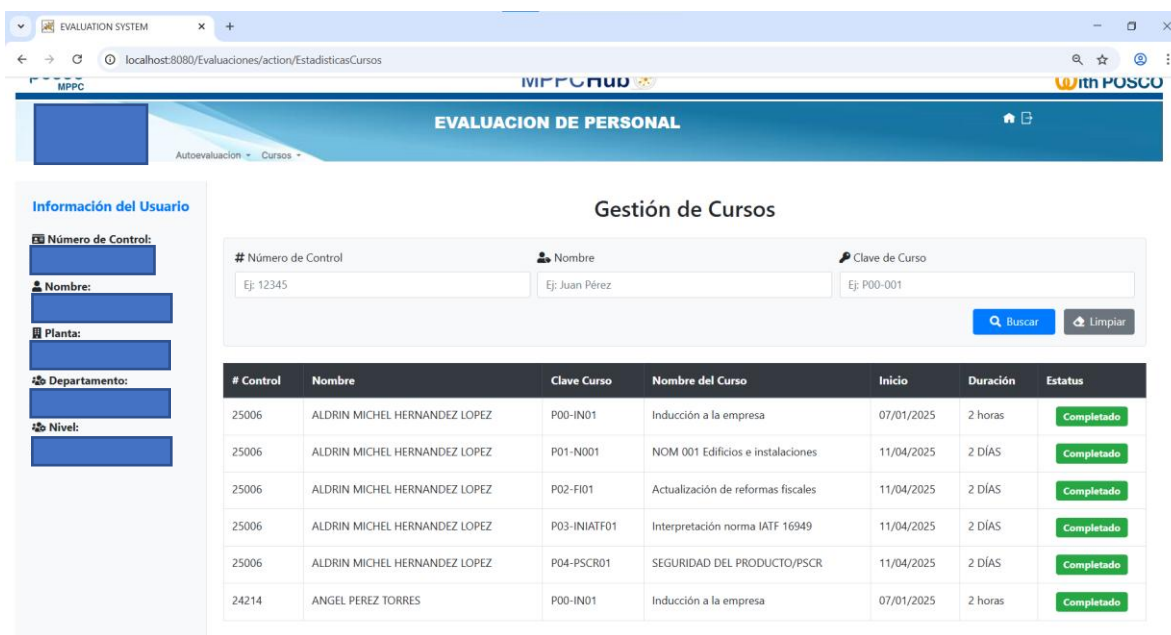
Los resultados se muestran en una tabla responsiva con encabezados estandarizados: número de control, nombre, clave y nombre del curso, fecha de inicio, duración y estatus. Para mejorar la lectura, el estatus se representa con insignias de color (verde para “Completado” y amarillo para otros estados). Si la consulta no arroja coincidencias, se despliega un mensaje informativo indicando que no se encontraron cursos con los criterios especificados.

La lógica de servidor se implementa en el servlet *EstadisticasCursos*. Durante su ciclo de vida, el servlet inicializa un *CursosJpaController* a partir de la unidad de persistencia *EvaluacionesPU*, lo que optimiza el acceso a datos. En cada solicitud, establece la codificación UTF-8, recupera el perfil del usuario desde la unidad de persistencia y lee los parámetros del formulario (número de control, nombre y clave). Con estos criterios invoca el método *findWithFilters(...)* del controlador JPA, que ejecuta la consulta en base de datos aplicando los filtros proporcionados. Los resultados se adjuntan al request y se reenvían a la

vista /jsps/evaluar/EstadisticasCursos.jsp, donde se renderizan. Al destruirse, el servlet cierra ordenadamente el EntityManagerFactory asociado al controlador para liberar recursos.

Figura 66

Filtros de búsqueda para capacitaciones asignadas



Información del Usuario

- Número de Control:
- Nombre:
- Planta:
- Departamento:
- Nivel:

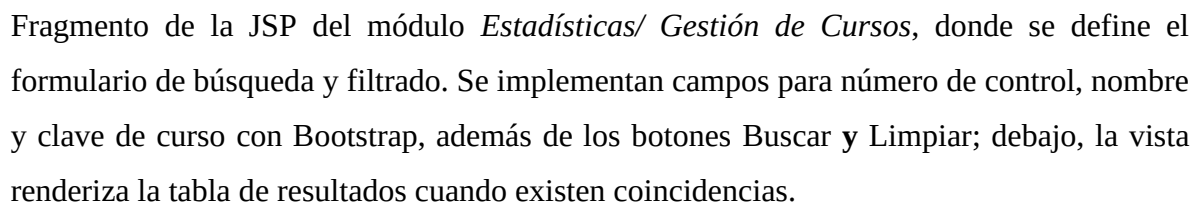
Gestión de Cursos

Número de Control: Ej: 12345 Nombre: Ej: Juan Pérez Clave de Curso: Ej: P00-001

# Control	Nombre	Clave Curso	Nombre del Curso	Inicio	Duración	Estatus
25006	ALDRIN MICHEL HERNANDEZ LOPEZ	P00-IN01	Inducción a la empresa	07/01/2025	2 horas	Completado
25006	ALDRIN MICHEL HERNANDEZ LOPEZ	P01-N001	NOM 001 Edificios e instalaciones	11/04/2025	2 DÍAS	Completado
25006	ALDRIN MICHEL HERNANDEZ LOPEZ	P02-FI01	Actualización de reformas fiscales	11/04/2025	2 DÍAS	Completado
25006	ALDRIN MICHEL HERNANDEZ LOPEZ	P03-INIATF01	Interpretación norma IATF 16949	11/04/2025	2 DÍAS	Completado
25006	ALDRIN MICHEL HERNANDEZ LOPEZ	P04-PSCR01	SEGURIDAD DEL PRODUCTO/PSCR	11/04/2025	2 DÍAS	Completado
24214	ANGEL PEREZ TORRES	P00-IN01	Inducción a la empresa	07/01/2025	2 horas	Completado

Vista del módulo *Gestión de Cursos* con los filtros de búsqueda habilitados para consultar las capacitaciones asignadas. La interfaz permite filtrar por número de control, nombre del empleado o clave de curso, mostrando en la tabla los resultados correspondientes con detalles como nombre del curso, fecha de inicio, duración y estatus de avance.

JSP de la página de búsqueda y filtrado



```

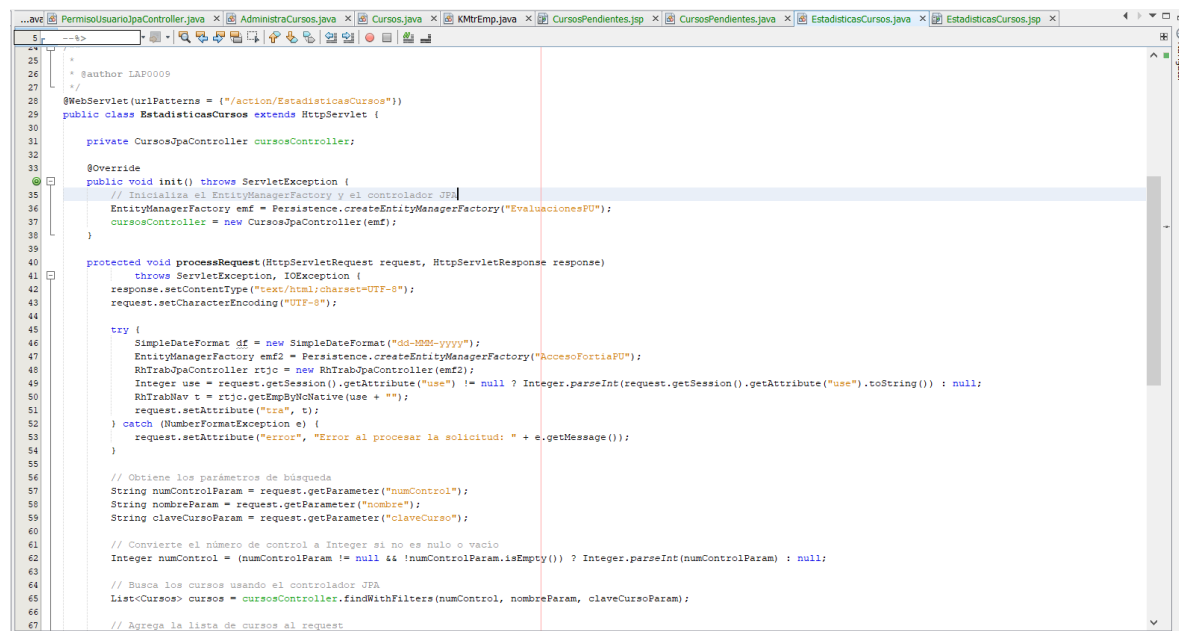
149     </thead>
150     <tbody>
151         <c:forEach items="${cursos}" var="curso" varStatus="loop">
152             <tr>
153                 <td>${curso.noControl}</td>
154                 <td>${curso.noEmpleado}</td> <!-- Corregido a noEmpleado -->
155                 <td>${curso.claveCurso}</td>
156                 <td>${curso.noCurso}</td>
157                 <td><fmt:formatDate value="${curso.fechaInicio}" pattern="dd/MM/yyyy"/></td>
158                 <td>${curso.duracion}</td>
159                 <td class="text-center">
160                     <c:choose>
161                         <c:when test="${curso.estatus eq 'Completado'}">
162                             <span class="badge badge-success">${curso.estatus}</span>
163                         <c:otherwise>
164                             <span class="badge badge-warning">${curso.estatus}</span>
165                         </c:otherwise>
166                     </c:choose>
167                 </td>
168             </tr>
169         </c:forEach>
170     </tbody>
171 </table>
172 </div>
173 <c:when>
174 <c:otherwise>
175     <div class="alert alert-info text-center">
176         <i class="fas fa-info-circle mr-2"></i>No se encontraron cursos con los criterios especificados
177     </div>
178 </c:otherwise>
179 </c:choose>
180 </div>
181 </div>
182 </div>
183 </script>
184
185 // Función para limpiar filtros
186 function limpiarFiltros() {
187     document.getElementById("numControl").value = '';
188     document.getElementById("nombre").value = '';
189 }

```

Fragmento de la JSP del módulo *Estadísticas de Cursos* correspondiente a la tabla de resultados. Aquí se renderizan los cursos obtenidos con los criterios de búsqueda, mostrando campos como número de control, nombre del empleado, clave y nombre del curso, fecha de inicio, duración y estatus. El estatus se presenta con insignias de color verde para cursos completados y amarillo para los pendientes. En caso de no encontrarse coincidencias, se despliega un mensaje informativo al usuario. Además, se incluye la función JavaScript limpiar Filtros() que restablece los campos del formulario a su valor inicial.

Figura 69

Servlet para vista de Búsqueda y filtrado



```

25  *
26  * @author LAP0009
27  */
28  @WebServlet(urlPatterns = {"/action/EstadisticasCursos"})
29  public class EstadisticasCursos extends HttpServlet {
30
31      private CursosJpaController cursosController;
32
33      @Override
34      public void init() throws ServletException {
35          // Inicializa el EntityManagerFactory y el controlador JPA
36          EntityManagerFactory emf = Persistence.createEntityManagerFactory("EvaluacionesPU");
37          cursosController = new CursosJpaController(emf);
38      }
39
40      protected void processRequest(HttpServletRequest request, HttpServletResponse response)
41          throws ServletException, IOException {
42          response.setContentType("text/html; charset=UTF-8");
43          request.setCharacterEncoding("UTF-8");
44
45          try {
46              SimpleDateFormat df = new SimpleDateFormat("dd-MM-yyyy");
47              EntityManagerFactory emf2 = Persistence.createEntityManagerFactory("AccesoFortiaPU");
48              RHTrabJpaController rtjc = new RHTrabJpaController(emf2);
49              Integer use = request.getSession().getAttribute("use") != null ? Integer.parseInt(request.getSession().getAttribute("use").toString()) : null;
50              RHTrabNav t = rtjc.getEmpByNcNative(use + "");
51              request.setAttribute("tra", t);
52          } catch (NumberFormatException e) {
53              request.setAttribute("error", "Error al procesar la solicitud: " + e.getMessage());
54          }
55
56          // Obtiene los parámetros de búsqueda
57          String numControlParam = request.getParameter("numControl");
58          String nombreParam = request.getParameter("nombre");
59          String claveCursoParam = request.getParameter("claveCurso");
60
61          // Convierte el número de control a Integer si no es nulo o vacío
62          Integer numControl = (numControlParam != null && !numControlParam.isEmpty()) ? Integer.parseInt(numControlParam) : null;
63
64          // Busca los cursos usando el controlador JPA
65          List<Curso> cursos = cursosController.findWithFilters(numControl, nombreParam, claveCursoParam);
66
67          // Agrega la lista de cursos al request

```

Fragmento del servlet *EstadisticasCursos*, responsable de la vista de búsqueda y filtrado. Se observa el método `init()` que inicializa el `CursosJpaController` con la unidad de persistencia `EvaluacionesPU`, y el método `processRequest` donde se configura la solicitud, se recupera la información del usuario, se leen los parámetros de filtro (número de control, nombre y clave de curso), se consulta la base de datos mediante `findWithFilters(...)` y se reenvían los resultados a `EstadisticasCursos.jsp` para su renderizado.

Figura 70

```

57  // Obtiene los parámetros de búsqueda
58  String numControlParam = request.getParameter("numControl");
59  String nombreParam = request.getParameter("nombre");
60  String claveCursoParam = request.getParameter("claveCurso");
61
62  // Convierte el número de control a Integer si no es nulo o vacío
63  Integer numControl = (numControlParam != null && !numControlParam.isEmpty()) ? Integer.parseInt(numControlParam) : null;
64
65  // Busca los cursos usando el controlador JPA
66  List<Cursos> cursos = cursosController.findWithFilters(numControl, nombreParam, claveCursoParam);
67
68  // Agrega la lista de cursos al request
69  request.setAttribute("cursos", cursos);
70
71  // Redirige a la página JSP que mostrará los resultados
72  request.getRequestDispatcher("/jsp/evaluar/EstadisticasCursos.jsp").forward(request, response);
73
74
75  @Override
76  protected void doGet(HttpServletRequest request, HttpServletResponse response)
77      throws ServletException, IOException {
78      processRequest(request, response);
79  }
80
81  @Override
82  protected void doPost(HttpServletRequest request, HttpServletResponse response)
83      throws ServletException, IOException {
84      processRequest(request, response);
85  }
86
87  @Override
88  protected void doPut(HttpServletRequest request, HttpServletResponse response)
89      throws ServletException, IOException {
90      processRequest(request, response);
91  }

```

4.5. Página de reporte de capacitaciones (Administrador)

El módulo de Reporte de Avance General constituye una herramienta fundamental dentro del sistema, ya que permite al perfil administrador obtener una visión integral del estado de los cursos registrados en la organización. Su principal función es consolidar la información almacenada en la base de datos y transformarla en indicadores clave que facilitan la supervisión y toma de decisiones estratégicas en materia de capacitación.

A través de este módulo, el administrador puede visualizar el promedio global de avance considerando todos los cursos, así como identificar la distribución de estatus entre los registros, distinguiendo aquellos que se encuentran completados, en proceso o pendientes. De igual forma, se muestra el total de cursos registrados y la cantidad de empleados únicos que participan en ellos, lo que proporciona una perspectiva completa sobre el nivel de cumplimiento en la institución.

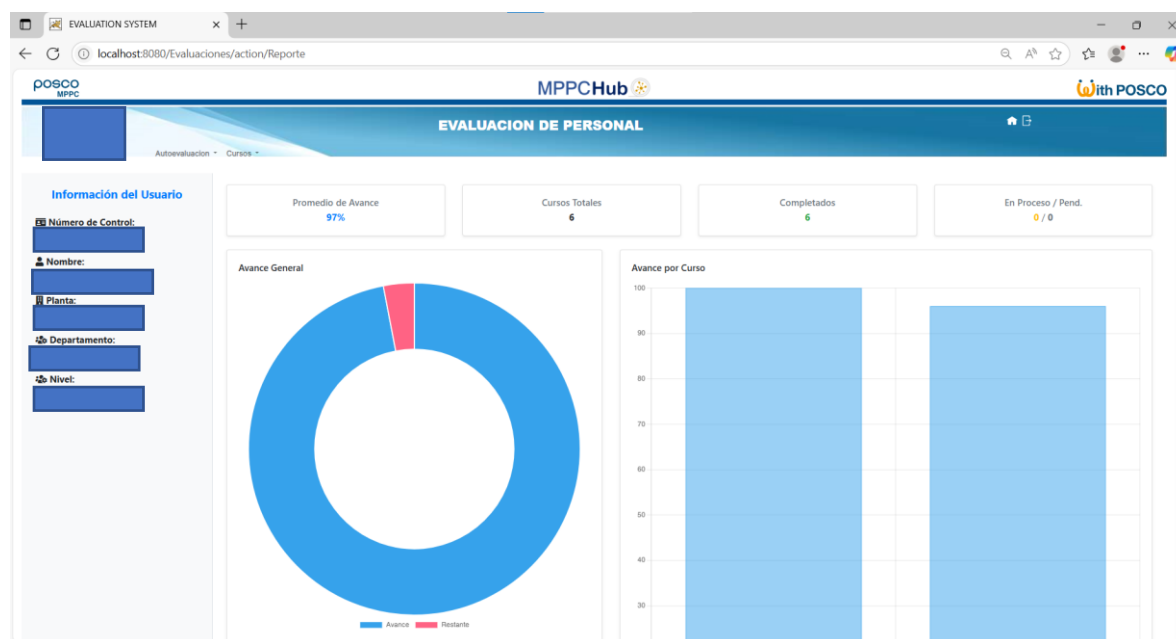
El sistema no se limita a ofrecer métricas generales, sino que también realiza un análisis agregado por empleado. De esta forma, se calculan promedios de avance individuales y se genera un ranking que destaca a los colaboradores con mayor o menor progreso, permitiendo identificar fortalezas y áreas de oportunidad en la formación del personal. Para mejorar la comprensión y facilitar el análisis visual, estos datos se presentan mediante gráficas

dinámicas: una de tipo dona, que refleja el porcentaje de cumplimiento general, y otra de barras, que muestra el avance promedio por empleado, limitando la visualización a los principales casos para evitar la saturación de información.

En conjunto, este módulo transforma los datos dispersos de la base en información estructurada, clara y fácilmente interpretable. Su aporte principal radica en ofrecer al administrador una herramienta de monitoreo y evaluación que respalda la gestión eficiente de los procesos de capacitación, garantizando el seguimiento oportuno y la mejora continua en el desarrollo del talento humano.

Figura 71

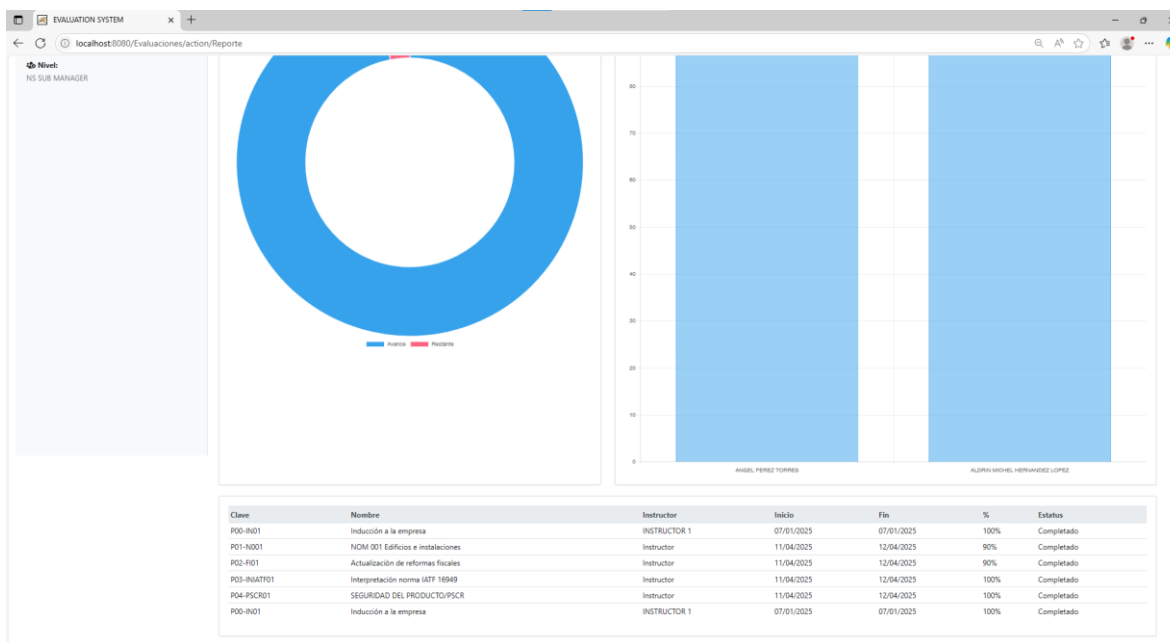
Vista del módulo de reporte



Vista del módulo de *Reporte de Avance General* en el sistema de evaluación de personal. Se muestran indicadores clave como el promedio de avance, cursos totales y estatus de cumplimiento, acompañados de gráficas dinámicas de dona y barras que ilustran el progreso global y por curso.

Figura 72

Vista del módulo de reporte



Interfaz del módulo de *Reporte de Avance General* mostrando la gráfica de progreso global y por empleado, acompañada de una tabla detallada con los cursos registrados. En la tabla se presentan la clave, nombre del curso, instructor, fechas de inicio y fin, porcentaje de avance y estatus de cada capacitación, lo que permite un seguimiento integral y claro del progreso de los colaboradores.

Figura 73

JSP de la vista del Módulo de reportes

```

Source History
141 <div class="progress" style="width: 100%; text-align: center;">
142 <div class="progress-bar" style="width: 100%; height: 10px; background-color: #007bff;">
```

Fragmento de código JSP correspondiente al módulo de *Reporte de Avance General*. En este bloque se definen las tarjetas de indicadores clave (promedio de avance, cursos totales, completados y en proceso/pendientes), utilizando componentes de Bootstrap y expresiones JSP para mostrar dinámicamente los valores obtenidos desde la base de datos.

Figura 74

```

53 EntityManagerFactory emfCursos = null;
54
55 try {
56     // Datos del trabajador logueado, por si los muestran en la barra ---
57     emfRh = Persistence.createEntityManagerFactory("AccesoFortiaPO");
58     RhTrabJpaController rtjc = new RhTrabJpaController(emfRh);
59     Object useAttr = request.getSession().getAttribute("use");
60     Integer use = null;
61     if (useAttr != null) {
62         try { use = Integer.valueOf(useAttr.toString()); } catch (Exception ignore) {}
63     }
64     RhTrabHav t = null;
65     if (use != null) {
66         t = rtjc.getEmpByHollative(String.valueOf(use));
67     }
68     request.setAttribute("tca", t);
69
70     // --- Cursos: ADMIN ve TODO ---
71     emfCursos = Persistence.createEntityManagerFactory("EvaluacionesPO"); // ajusta si aplica
72     CursosJpaController cjc = new CursosJpaController(emfCursos);
73
74     // TODOS los registros de la tabla Cursos
75     List<Cursos> cursos = cjc.findCursosEntities();
76     if (cursos == null) cursos = Collections.emptyList();
77     request.setAttribute("cursos", cursos);
78
79     // --- EEs globales ---
80     double suma = 0.0;
81     int cuentaConPorcentaje = 0;
82     long completados = 0;
83     long enProceso = 0;
84     long pendientes = 0;
85
86     // --- Agregación por empleado ---
87     // Map por noControl -> sumaPorcentajes, contador, nombre
88     Map<Integer, Integer> sumasPorEmpleado = new HashMap<Integer, Integer>();
89     Map<Integer, Integer> contenesPorEmpleado = new HashMap<Integer, Integer>();
90     Map<Integer, String> nombresPorEmpleado = new HashMap<Integer, String>();
91
92     for (int i = 0; i < cursos.size(); i++) {
93         Cursos c = cursos.get(i);
94
95         // Promedio global
96         Integer p = c.getPorcentaje();
97         if (p != null) {
98             suma += p.intValue();
99             cuentaConPorcentaje++;
100         }
101
102         // Estatus global
103         String est = safe(c.getEstatus());
104         if (ip != null && p.intValue() >= 100 || "completado".equalsIgnoreCase(est)) {

```

Fragmento del servlet asociado al módulo de *Reporte de Avance General*. En este código se observa la conexión a la unidad de persistencia, la obtención de datos de los cursos mediante JPA y el cálculo de métricas de avance. Posteriormente, estos resultados son preparados como atributos y enviados al JSP para su visualización en forma de indicadores y gráficas.

Conclusiones

El presente proyecto concluye con la implementación exitosa de un sistema integral de gestión de capacitaciones, cuyo diseño y desarrollo respondió de manera directa a las necesidades institucionales de control, seguimiento y análisis de capacitaciones asignadas al personal. A lo largo del proceso, se construyeron módulos que, además de cumplir con los objetivos planteados en el anteproyecto, aportaron un valor agregado al ofrecer un entorno tecnológico moderno, seguro y adaptable.

El módulo de Administración de Cursos se consolidó como el núcleo del sistema, al permitir la carga masiva de capacitaciones en formatos ODS y XLSX, su validación y almacenamiento en la base de datos. Gracias a este componente, la gestión de la información pasó de ser manual y fragmentada a ser automatizada y confiable, reduciendo los errores humanos y asegurando la integridad de los datos.

El módulo de Cursos Pendientes proporcionó al usuario una interfaz clara y dinámica para consultar las capacitaciones asignadas y visualizar su progreso en tiempo real. La incorporación de filtros por rangos de fechas y la representación gráfica mediante Chart.js fortalecieron la experiencia del usuario al convertir los datos en información visualmente comprensible, lo que facilita el seguimiento de su avance y motiva el cumplimiento de las capacitaciones.

Por su parte, el módulo de Estadísticas de Cursos amplió el alcance del sistema hacia el análisis y la toma de decisiones. Al integrar filtros flexibles el cual permite generar listados dinámicos que reflejan con precisión el estado actual de las capacitaciones en la organización. Con ello, se brinda al área de Recursos Humanos una herramienta analítica que transforma los registros almacenados en información estratégica, indispensable para mejorar la planeación de futuros programas de capacitación.

La adopción de la metodología Kanban resultó clave para el éxito del proyecto, pues facilitó la organización del trabajo en fases claras, el control de los avances y la adaptación a cambios emergentes durante el desarrollo. Esta aproximación permitió entregar resultados funcionales de manera iterativa, manteniendo un ritmo sostenible y fomentando tanto el crecimiento

técnico como el fortalecimiento de habilidades blandas, tales como la comunicación y la colaboración.

En términos tecnológicos, la decisión de utilizar Java EE con Servlets, JSP, JPA y AJAX, complementados con Bootstrap y librerías de apoyo, garantizó la construcción de un sistema robusto, escalable y alineado a estándares reconocidos de la industria. Esta arquitectura asegura la mantenibilidad y la posibilidad de futuras expansiones, como la integración con plataformas LMS o la implementación de notificaciones automatizadas para los empleados.

El proyecto no solo resolvió las necesidades inmediatas de gestión de capacitaciones, sino que sentó las bases para la evolución continua de la organización hacia una cultura de aprendizaje sustentada en datos. La experiencia adquirida en su desarrollo permitió fortalecer competencias técnicas y metodológicas, además de consolidar el compromiso con valores como la responsabilidad, la honestidad y la eficiencia. Este sistema representa un punto de partida sólido para la innovación futura, orientado a potenciar el desarrollo del talento humano mediante la tecnología.

Competencias desarrolladas

El desarrollo del sistema de gestión de cursos representó una experiencia formativa que me permitió fortalecer una amplia gama de competencias técnicas, profesionales y personales. A lo largo del proyecto tuve la oportunidad de aplicar los conocimientos adquiridos durante mi formación académica y, al mismo tiempo, enfrentar retos reales que impulsaron mi capacidad de análisis, resolución de problemas y toma de decisiones estratégicas.

En el ámbito técnico, consolidé el dominio de tecnologías propias del ecosistema Java EE, particularmente en el uso de Servlets, JSP y JPA bajo el patrón de arquitectura MVC, lo que me permitió diseñar un sistema modular, escalable y mantenible. Asimismo, adquirí experiencia en la integración de librerías externas. Del mismo modo, reforcé mis habilidades en el manejo de bases de datos mediante la persistencia con JPA y la validación de datos, asegurando la integridad y consistencia de la información almacenada. También desarrollé competencias en la construcción de interfaces dinámicas y responsivas, aplicando Bootstrap, AJAX y Chart.js, lo que dio como resultado una experiencia de usuario más fluida, intuitiva y visualmente atractiva.

Desde el punto de vista metodológico, el uso de la metodología ágil Kanban me permitió organizar y planificar el trabajo de forma iterativa, priorizando tareas y entregables de acuerdo con los avances del proyecto y los cambios surgidos durante el proceso. Esta experiencia fortaleció mi capacidad de autogestión, mi disciplina en el cumplimiento de objetivos y mi habilidad para adaptar el desarrollo a nuevas necesidades sin perder de vista el alcance original.

En cuanto a las competencias personales y blandas, este proyecto me permitió crecer en aspectos fundamentales para el ejercicio profesional. La comunicación efectiva y la documentación clara fueron esenciales para explicar de manera precisa cada componente desarrollado. Al mismo tiempo, se reforzó mi compromiso con valores como la honestidad y la responsabilidad, procurando siempre la calidad de la solución tecnológica entregada. Igualmente, aprendí a trabajar con una mentalidad de mejora continua, manteniendo una actitud crítica y autocrítica frente a las decisiones tomadas y buscando constantemente alternativas más eficientes.

Las competencias desarrolladas abarcaron desde la técnica especializada, centrada en el diseño y construcción de un sistema robusto, hasta la metodológica y personal, vinculada a la gestión ágil, la resolución de problemas y el compromiso ético. Todo ello se tradujo en un fortalecimiento integral de mi perfil profesional, preparándome no solo para enfrentar proyectos de software de mayor complejidad, sino también para asumir con confianza el reto de aportar soluciones tecnológicas alineadas a las necesidades reales de las organizaciones.

Propuestas de mejora

Si bien el sistema desarrollado cumple con los objetivos planteados y ha demostrado mejorar la eficiencia en la administración de cursos de capacitación, existen diversas áreas de oportunidad que pueden potenciar su funcionalidad y asegurar su evolución a futuro. Entre las principales propuestas de mejora se encuentran:

1. **Integración con un LMS (Learning Management System):** Incorporar el módulo a una plataforma de gestión del aprendizaje que permita centralizar tanto la administración como la impartición de los cursos, facilitando el acceso a contenidos en línea, evaluaciones automatizadas y seguimiento detallado del desempeño.
2. **Notificaciones y recordatorios automáticos:** Implementar un sistema de alertas mediante correo electrónico o notificaciones en la aplicación para recordar a los empleados la fecha de inicio o vencimiento de sus cursos, fomentando la puntualidad y el cumplimiento en los programas de capacitación.
3. **Expansión de funcionalidades de Recursos Humanos:** Integrar el sistema con otros procesos de gestión del talento, como evaluaciones de desempeño, detección de necesidades de capacitación o planes de carrera, alineando la formación con la estrategia corporativa de desarrollo del personal.

Estas propuestas de mejora buscan no solo robustecer la solución ya implementada, sino también proyectar al sistema como una herramienta integral y estratégica para la administración del talento humano, con capacidad de escalar y adaptarse a las necesidades cambiantes de la organización.

Bibliografía

- Alles, M. (2012). *Desarrollo del talento humano basado en competencias (3ª ed.)*. Editorial Granica.
- Argyris, C., & Schön, D. (1978). *Organizational Learning: A Theory of Action Perspective*. Addison-Wesley.
- BBVA. (30 de Junio de 2025). *Arquitectura de 'software' o sistemas: qué es y ejemplos*. Obtenido de <https://www.bbva.com/es/innovacion/arquitectura-de-software-o-sistemas-que-es-y-ejemplos/>
- Britannica. (2025). *Java | Definition & Facts*. Obtenido de <https://www.britannica.com/technology/Java-computer-programming-language>
- Cabero, J. (2006). Bases pedagógicas del e-learning. . *Revista de Universidad y Sociedad del Conocimiento*, 1-10.
- Chamilo. (2023). *Chamilo LMS – Características Principales*. Obtenido de <https://chamilo.org>
- Chiavenato, I. (2017). *Administración de recursos humanos: El capital humano de las organizaciones (10ª ed.)*. McGraw-Hill Education.
- Clements, P., Stafford, J., Bark, C., Garlan, D., Little, R., Nord, R., & Z. (1994). *What is your definition of software architecture?* Software Engineering Institute. Obtenido de https://insights.sei.cmu.edu/documents/2544/2010_010_001_513810.pdf?
- CognosOnline. (2021). *¿Qué es e-learning? Definición y características*. Obtenido de <https://cognosonline.com/que-es-e-learning/>
- Coursera Staff. (6 de Mayo de 2024). *coursera*. Obtenido de <https://www.coursera.org/in/articles/jsp-vs-servlet>
- Darwin, I. (1996). *The Java™ Language Specification*. Addison-Wesley.
- Davenport, T. H., & Prusak, L. (2001). *Conocimiento en Acción: Cómo las organizaciones gestionan lo que saben*. . Pearson Educación.
- Dessler, G. (2015). *Administración de recursos humanos (14ª ed.)*. Pearson Educación.
- García Peñalvo, F. J. (2005). *Estado actual de los sistemas e-learning*. Ediciones Universidad de Salamanca.
- García Peñalvo, F. J. (2005). *Estado actual de los sistemas e-learning*. Ediciones Universidad de Salamanca.
- Garlan, D., & Shaw, M. (1994). *An introduction to software architecture*. Carnegie Mellon University.
- GeeksforGeeks. (2025). *Patrón de diseño MVC*. Obtenido de <https://www.geeksforgeeks.org/system-design/mvc-design-pattern>

- Ghirardini, B. (2011). E-learning methodologies: A guide for designing and developing e-learning courses. *FAO*, 48-63. Obtenido de <https://www.fao.org/3/i2516e/i2516e.pdf>
- Gosling, J., & McGilton, H. (1995). *The Java Language*. Sun Microsystems.
- hostinger. (6 de Junio de 2025). *Web application architecture guide: What it is, and key components*. Obtenido de <https://www.hostinger.com/tutorials/web-application-architecture>
- IBM. (28 de Enero de 2025). *Java Persistence API (JPA)*. Obtenido de <https://www.ibm.com/docs/es/was-liberty/nd?topic=liberty-java-persistence-api-jpa>
- IBM. (2025). *What Is Java?* Obtenido de <https://www.ibm.com/think/topics/java>
- King, G., Bauer, C., Andersen, M. R., Bernard, E., & Ebersole, S. (15 de Septiembre de 2010). *docs.jboss.org*. Obtenido de https://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html_single/
- Kirkpatrick, D. L., & Kirkpatrick, J. D. (2007). *Evaluación de acciones formativas: los cuatro niveles*. Gestión 2000.
- Kiwop. (15 de Mayo de 2025). *Arquitectura Web: cómo escalar tu negocio en Argentina*. Obtenido de <https://www.kiwop.com/blog/arquitectura-web-argentina-2025>
- MDN Web Docs. (2023). *MVC – Glossary*. . Obtenido de <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
- Moodle. (2023). *About Moodle – Open-source Learning Platform*. Obtenido de <https://moodle.org>
- Nonaka, I., & Takeuchi, H. (1999). *La organización creadora de conocimiento: cómo las compañías japonesas crean la dinámica de la innovación*. Oxford University Press.
- Oracle. (2025). *Servlets and JSP Pages Best Practices*. Obtenido de <https://www.oracle.com/technical-resources/articles/javase/servlets-jsp.html>
- Phillips, J. J., & Phillips, P. P. (2008). *ROI en formación y desarrollo: Cómo medir el retorno de la inversión en formación*. Netbiblo.
- ResearchGate. (2023). *A Review on Java Programming Language*. Obtenido de https://www.researchgate.net/publication/371166744_A_Review_on_Java_Programming_Language
- Rosenberg, M. J. (2001). *E-learning: Strategies for delivering knowledge in the digital age*. McGraw-Hill Education.
- Sangrà, A., Vlachopoulos, D., & Cabrera, N. (2012). Building an inclusive definition of e-learning: An approach to the conceptual framework. *International Review of Research in Open and Distance Learning*.

- Senge, P. (2005). *La Quinta Disciplina: El arte y la práctica de la organización abierta al aprendizaje*. Granica.
- Seshadri, G. (29 de Diciembre de 1999). *Understanding JavaServer Pages Model 2 architecture*. Obtenido de <https://www.infoworld.com/article/2164578/understanding-javascript-pages-model-2-architecture.html>
- Siliceo, A. (2009). *Capacitación y desarrollo de personal (2ª ed.)*. Editorial Limusa.
- Smowl. (2022). *¿Qué es una plataforma e-learning? Concepto y para qué sirve*. Obtenido de <https://smowl.net/es/blog/plataforma-elearning/>
- TechTarget. (2025). *¿Qué es el patrón Modelo-Vista-Controlador (MVC)?* Obtenido de <https://www.techtarget.com/whatis/definition/model-view-controller-MVC>
- thinksys. (25 de Junio de 2025). *Web Application Architecture: Working, Components, Types, Trends*. Obtenido de <https://thinksys.com/development/web-application-architecture-complete-guide/>
- Tutorialspoint. (2024). *Design Patterns – MVC Pattern*. Obtenido de https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm