

DESCRIPCIÓN DE LA ARQUITECTURA CORE

Contenido

1. INTRODUCCIÓN	3
1.1 Propósito	3
1.2 Alcance	3
1.3 Definiciones, Siglas y Abreviaturas	3
1.4 Audiencia del Documento de Arquitectura.	4
2. REQUERIMIENTOS DE ARQUITECTURA	5
3. Modularidad	6
3.1 Módulo de Seguridad	7
3.2 Módulo de Auditoria	8
3.3 Módulo de Servicios de Infraestructura	8
4. Ámbito de la Arquitectura	9
4.1 Capa de Presentación	10
4.2 Capa de Negocio	11
4.3 Capa de Acceso a Datos	12

1. INTRODUCCIÓN

El presente documento corresponde la definición de la arquitectura de software CORE para el Ministerio Público Fiscalía de la Nación, documento que centrará principalmente en la descripción, componentes de la arquitectura y en cómo serán usados en la construcción e implementación de los sistemas.

La arquitectura será basada en una plataforma de soluciones web y funciones construidas e integradas bajo el paradigma de orientación a servicios, implementando procesos de negocio bajo una visión de incorporar elementos e infraestructura que lleven al sistema a una arquitectura completamente orientada a servicios.

1.1 Propósito

El presente documento tiene como principal objetivo presentar el diseño técnico de la arquitectura a utilizar para el desarrollo del proyecto (capas de la arquitectura, componentes de servicios, mecanismo de integración, mecanismos de persistencia, mecanismo de manejo de errores etc.).

Asimismo en este documento se describen las tecnologías a utilizar en la construcción, pruebas y despliegue de la solución.

1.2 Alcance

El presente documento muestra y describe las siguientes consideraciones:

- a) Mecanismo de comunicación entre los módulos.
- b) Patrones de diseños y frameworks a usar por cada capa de la aplicación.
- c) Mecanismo de persistencia y registro de la información en base de datos.

1.3 Definiciones, Siglas y Abreviaturas

Las principales siglas utilizadas a lo largo de este documento, son:

Sigla	Significado
Java EE	Java Enterprise Edition
JVM	Java Virtual Machine
EJB	Enterprise Java Bean
JPA	Java Persistence API
POJO	Plain Old Java Object
LDAP	Lightweight Directory Access Protocol
JAX-RS	Java API para RestFul Web Services
JSF	Java Server Faces
CDI	Context Dependency Injection
WAR	Formato de empaquetado java para aplicaciones web
BCE	Patrón de desarrollo Boundary-Controller- Entity

1.4 Audiencia del Documento de Arquitectura.

La audiencia objetivo del presente documento corresponde a:

- a) EL EQUIPO DE ARQUITECTURA el cual utilizará este documento para seguir las pautas y directivas de implementación del Software, respetando que se cumplan con las convenciones aquí establecidas.
- b) EQUIPO DESARROLLO, los cuales encontrarán en este documento el detalle de la documentación de los bloques básicos y componentes, en los que se construirá gran parte de los elementos del sistema.
- c) EQUIPO DE CALIDAD; Encontrarán la información necesaria para crear y aplicar listas de revisión e inspección, así como para validar el cumplimiento de la arquitectura durante las pruebas no funcionales del Software.

2. REQUERIMIENTOS DE ARQUITECTURA

Se requiere que la solución propuesta sea construida bajo los conceptos y principios de una arquitectura orientada a servicios. La publicación de servicios Web y la reutilización de elementos de la arquitectura para sistemas actuales e interconexión con sistemas externos a la institución serán parte fundamental y que estos puedan integrarse con distintas fuentes de datos pensando en una arquitectura de servicios.

Por razones que la arquitectura debe responder de manera flexible a las necesidades del negocio y proporcionar control de manera adecuada de sus servicios requieren una arquitectura sólida y robusta que cumpla con las siguientes características en capacidad de crecimiento, escalabilidad y plataforma de trabajo.

#	Requisitos Técnicos	Descripción
1	Escalabilidad	La Arquitectura propuesta es escalable, con capacidad para manejar sobrecarga de trabajo de una manera simple y sin complejidades
2	Portabilidad	Esta arquitectura propuesta es de alta portabilidad, de fácil implementación según a los estándares abiertos, eliminando el uso de cualquier herramienta propietaria. La portabilidad del Software representará la factibilidad de ser transportada de una plataforma a otra sin que se necesite modificaciones de importancia en el código fuente, teniendo independencia de los fabricantes de hardware, sistemas Operativos, base de datos y Servidores de Aplicaciones (Java EE 7).
3	Alta Disponibilidad	La Arquitectura propuesta soportara condiciones para poder operar y funcionar de manera continua con una recuperación rápida y transparente en caso de fallas.
4	Integración	La Arquitectura propuesta facilitará la Integración mediante servicios y permitirá interactuar con nuevos sistemas o tecnologías que existen en el mercado, así como con los servicios de otros proveedores externos.
5	Seguridad y Auditoria	Esta arquitectura ofrece una fuerte seguridad y confidencialidad de la información e Integridad de los datos así como la capacidad de ser auditable. Los servicios de acceso y seguridad deben seguir los lineamientos de protocolos de seguridad LDAP versión 3 y seguridad basada en estándares OWASP. La arquitectura deberá considerar registro y la grabación de pistas de auditoría registrando actividades de los usuarios dentro del sistema.
6	Reusabilidad	La arquitectura del software permite la reusabilidad de los Componentes, característica que permitirá reducir el costo de mantenimiento futuro, al evitar tener que recodificar o copiar las porciones de código fuente del sistema.
7	Usabilidad	El software es de fácil de uso e intuitivo con un estándar look-n-feel y con la facilidad de la navegación.
9	Transaccionabilidad	El software esta implementado para procesar grandes volúmenes de información en procesos de negocio, reportes y consultas. Procesos y reportes en batch así como manejo de colas de procesos.
10	Costos en Licenciamiento	Por tener una arquitectura abierta el sistema propuesto tiene la capacidad de ser desplegado en servidores de aplicaciones open source.

3. Modularidad

Como parte fundamental de la solución, la orientación a servicios obedece a la necesidad de convertir la arquitectura CORE en una herramienta de negocio ágil. Esto se logrará en la medida que los servicios soporten los objetivos del negocio, y cuenten con características de diseño tecnológicos para cumplir con el enfoque de una arquitectura orientada a servicios, adicionalmente el producto final será principalmente un conjunto de servicios compuestos y de infraestructura tecnológica congruentes con las metas de flexibilidad, trazabilidad y visibilidad del negocio.

La arquitectura esta implementado en una arquitectura siguiendo el patrón Microservices lo que nos permite agrupar componentes lógicos correspondientes a cada una de las áreas funcionales de la aplicación y descomponer la aplicación en una colección de módulos (WARS) colaborativos independientes cada uno con su propia base de datos y que se comunican entre ellos mediante servicios definidos.

Las ventajas que nos brinda esta arquitectura son:

- Deploy independiente de cada módulo.
- El tamaño de cada módulo es menor lo que permite un tiempo deploy e inicio de la aplicación más corto.
- Mayor nivel de aislamiento y disponibilidad que una aplicación monolítica.

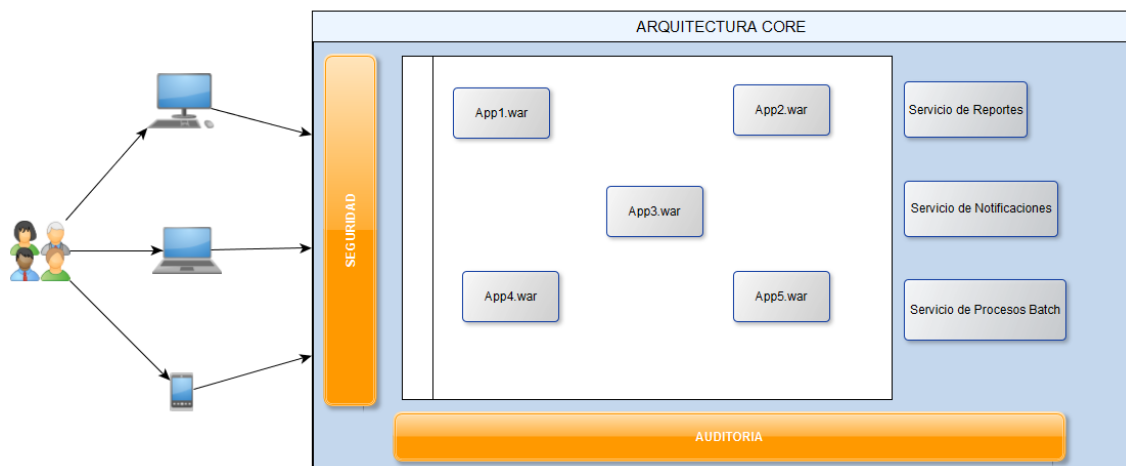


Ilustración 1 Módulos

3.1 Módulo de Seguridad

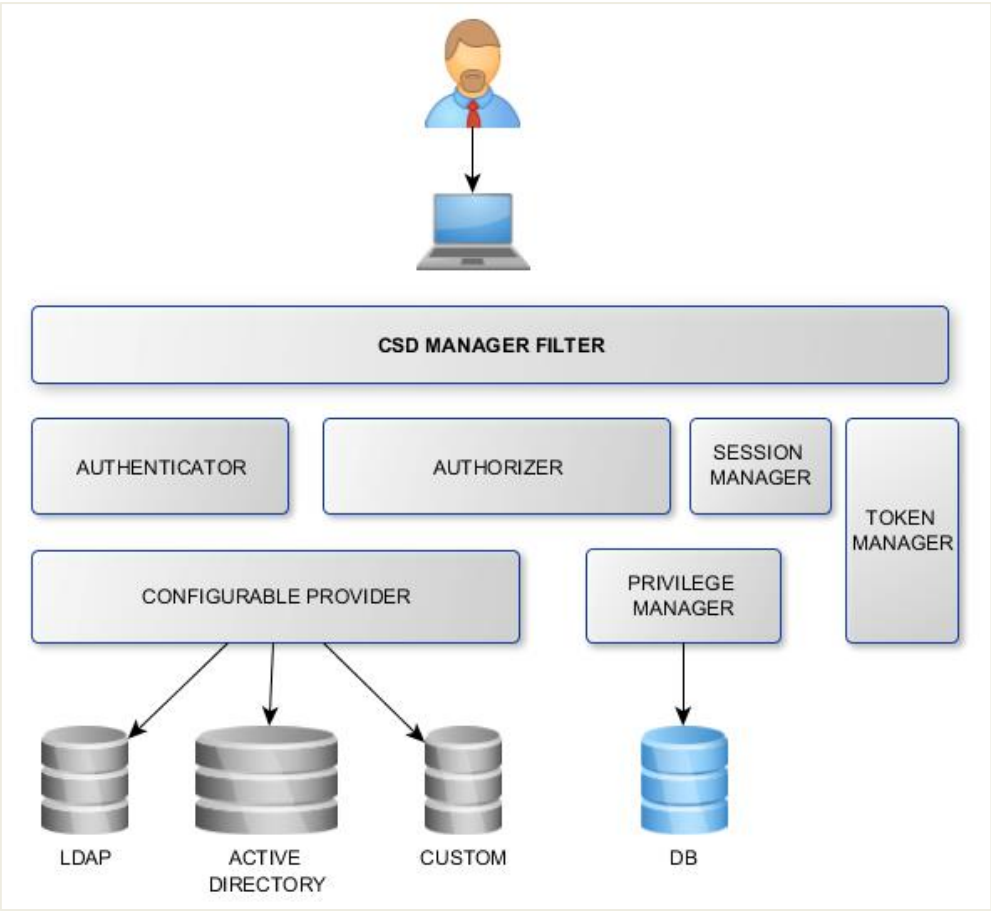


Ilustración 2 Módulo de Seguridad

Componente	Funciones
MANAGER FILTER	<ul style="list-style-type: none">• Encargado de capturar cualquier intento de acceso a los recursos• Verificar si el usuario ya cuenta con una sesión activa.• Verificar si el usuario cuenta con los permisos para acceder al recurso deseado.
AUTHENTICATOR	<ul style="list-style-type: none">• Es el servicio responsable de autenticar las credenciales frente al Directory Server LDAP3 (OpenLdap, Active Directory, ó algún otro proveedor que implemente las interfaces del componente y que este configurado como el proveedor activo de autenticación.
PRIVILEGE MANAGER	<ul style="list-style-type: none">• Servicio responsable de obtener la configuración de los perfiles y privilegios de los usuarios.
AUTHORIZER	<ul style="list-style-type: none">• Encargado de permitir el acceso a recursos o ejecución de acciones utilizando la información del Privilege Manager
TOKEN MANAGER	<ul style="list-style-type: none">• Generar un token interno por cada inicio de sesión del usuario para validar el SSO entre los módulos

3.2 Módulo de Auditoria

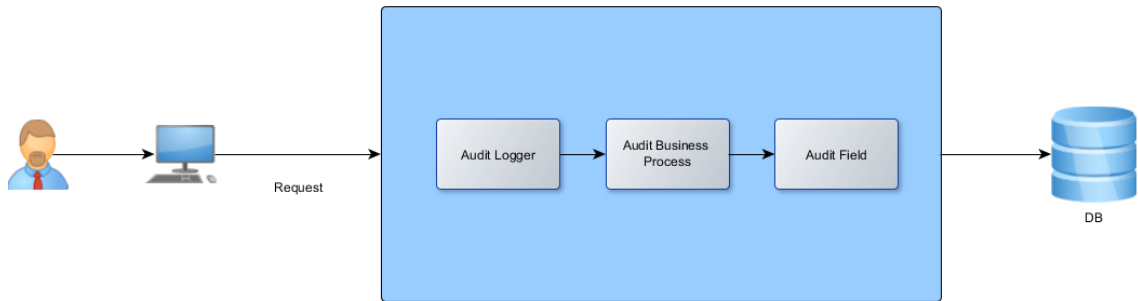


Ilustración 3 Módulo de Auditoria

Componente	Funciones
AUDIT LOGGER	<ul style="list-style-type: none">Encargado de registrar todos los accesos del usuario durante el tiempo que su sesión este activa en el sistema
AUDIT BUSINESS PROCESS	<ul style="list-style-type: none">Encargado de registrar que acciones realizo el usuario en el sistema(que procesos de negocio realizo, los componentes de negocio, servicios ejecutados, reportes, interfaces)
AUDIT FIELD	<ul style="list-style-type: none">Encargado de registrar pistas de auditoria para todos las modificaciones realizadas a nivel de registros de tablas

3.3 Módulo de Servicios de Infraestructura

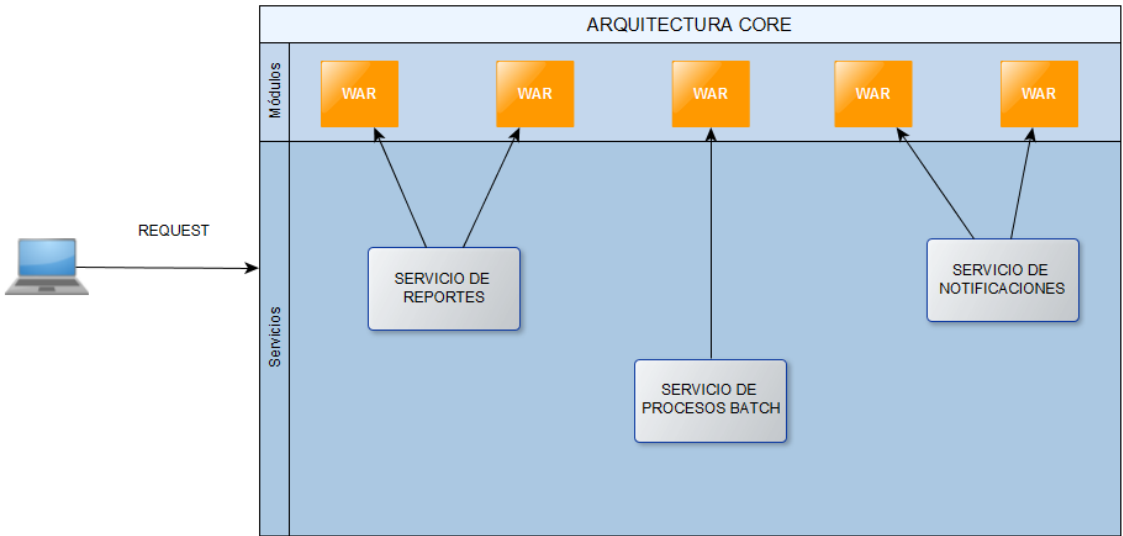


Ilustración 4 Servicios de Infraestructura

4. Ámbito de la Arquitectura

La arquitectura Core esta implementado utilizando el Stack Java EE 7 para poder responder a las necesidades de una aplicación empresarial como lo es escalabilidad, transaccionalidad, concurrencia, portabilidad.

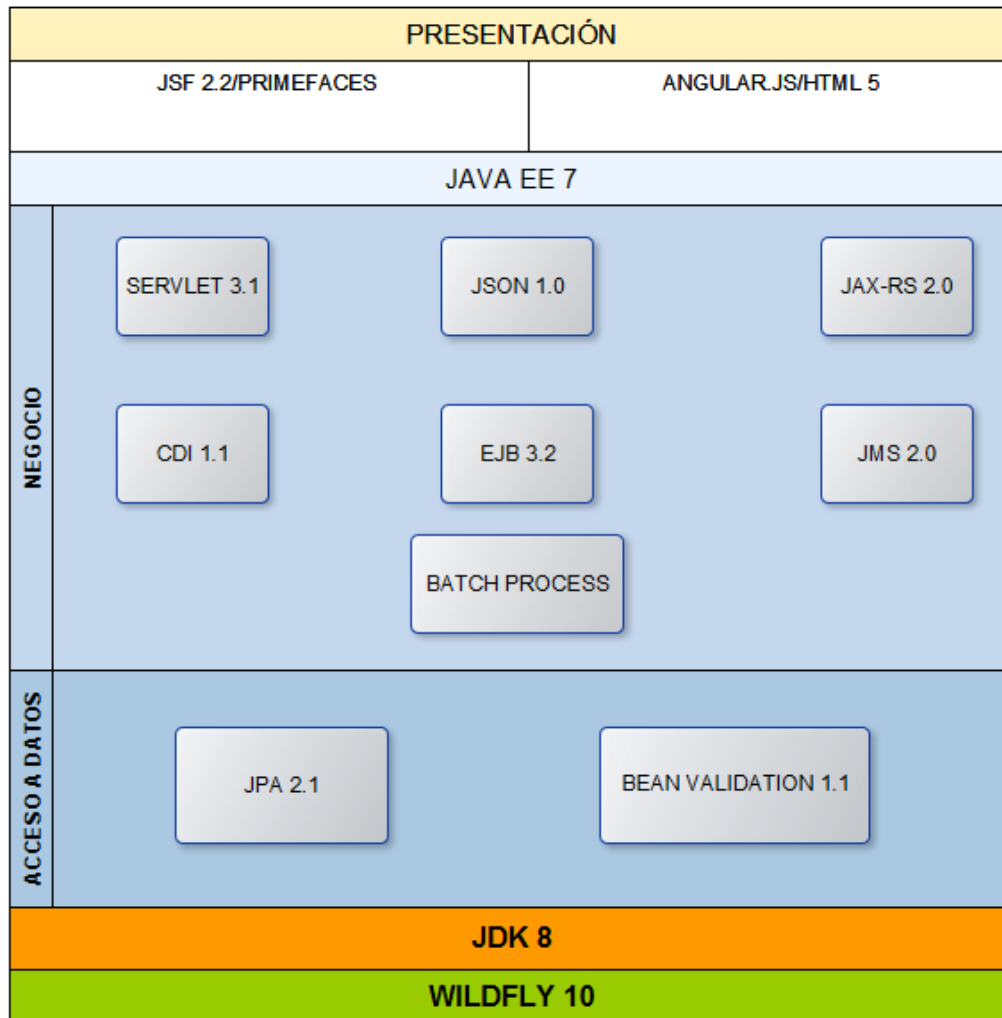


Ilustración 5 Vista Tecnológica

4.1 Capa de Presentación

Características
Implementada con JSF 2.2 utilizando las librerías Primefaces para el uso de componentes y look and feel.
Las páginas son implementadas en HTML 5 utilizando el motor de templates Facelet.
La interacción con la capa de negocio es a través del consumo de los componentes Boundary.

Componente	Funciones
MPFN-JSF-CORE	<ul style="list-style-type: none">• Encargado de la configuración de parámetros comunes de JSF• Encargado de las configuración de parámetros comunes en el web.xml• Encargado de crear las clases Base y Scopes estándares que se utilizaran.• Encargado brindar los templates comunes para la utilización de layouts uniformes en las aplicaciones

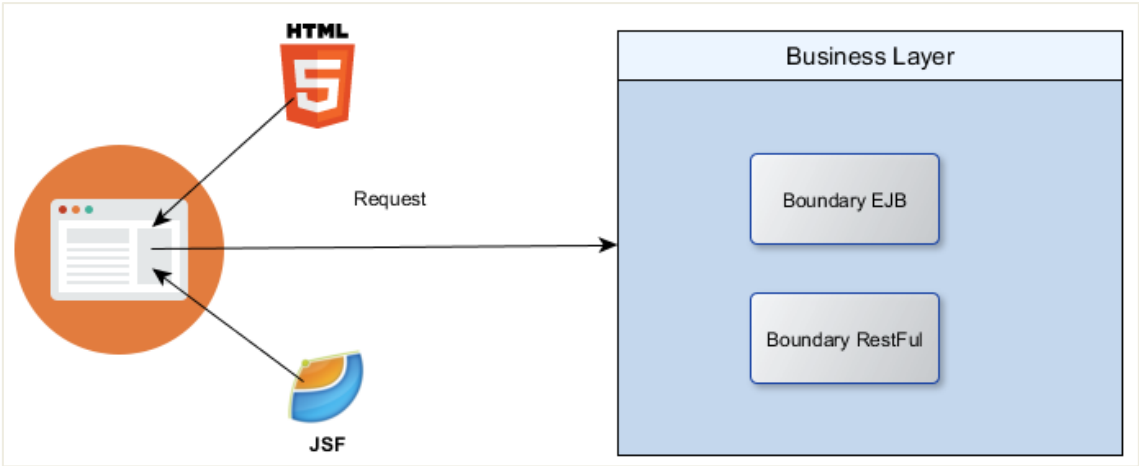


Ilustración 6 Interacción con la capa de Negocio

4.2 Capa de Negocio

Características
Implementado utilizando el patrón BCE(Boundary - Controller - Entity)
Los componentes Boundary (EJB/RestFul) son los únicos que interactúan directamente con la capa de presentación.
El componente Boundary interactúa con uno o más componentes Controller para poder ejecutar el proceso de negocio.

Componente	Funciones
MPFN-CORE	<ul style="list-style-type: none">• Encargado del manejo de excepciones.• Encargado de verificar la seguridad.

Componente	Funciones
MPFN-CONFIGURATION	<ul style="list-style-type: none">• Encargado de la configuración de parámetros de la aplicación utilizando la anotación @Configurable• Encargado de configurar el logging de la aplicación inyectando el componente MpfnLogger

Componente	Funciones
MPFN-SECURITY	<ul style="list-style-type: none">• Encargado de implementar los servicios de encriptación/desencryptación de los tokens de seguridad• Encargado de implementar a la clase UserTrack que pasa la información de auditoria entre las capas de la aplicación

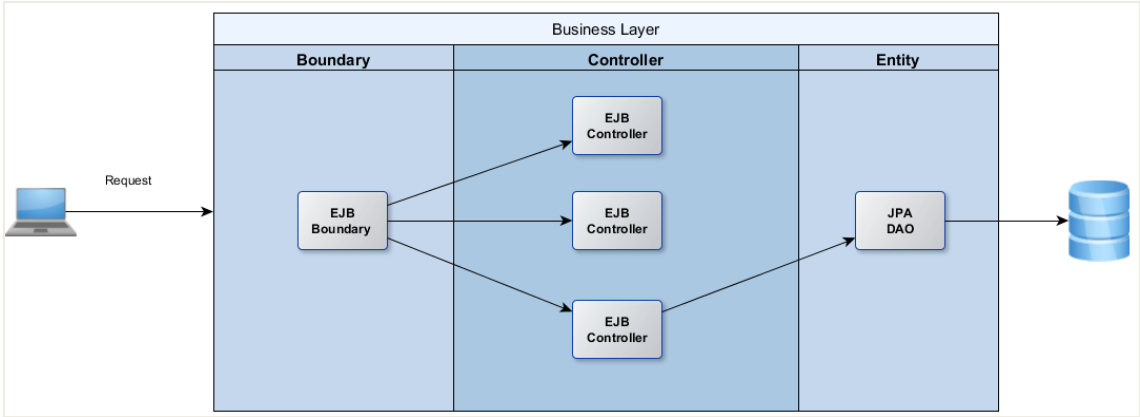


Ilustración 7 Interacción capa de Negocio

4.3 Capa de Acceso a Datos

Características
Implementado con JPA para el mapeo de Objetos a Tablas.
Utiliza Bean Validation para la validación de la información que será persistida.
Manejo de concurrencia de actualizaciones de registros utilizando Optimistic Locking agregando a cada entidad la anotación @Version.

Componente	Funciones
MPFN-ENTITYMANAGER	<ul style="list-style-type: none">• Encargado de manejar la abstracción y reutilización de métodos para realizar las operaciones CRUD en la base de datos utilizando JPA• Encargado de registrar las auditorias de registros obtenidos en la capa de Negocio.• Encargado de brindar el mapping de los nuevos tipos de datos LocalDateTime y LocalDate a Date para poder ser utilizado en el mapeo de JPA.• Encargado de recibir la configuración paramétrica del nombre de la unidad de persistencia

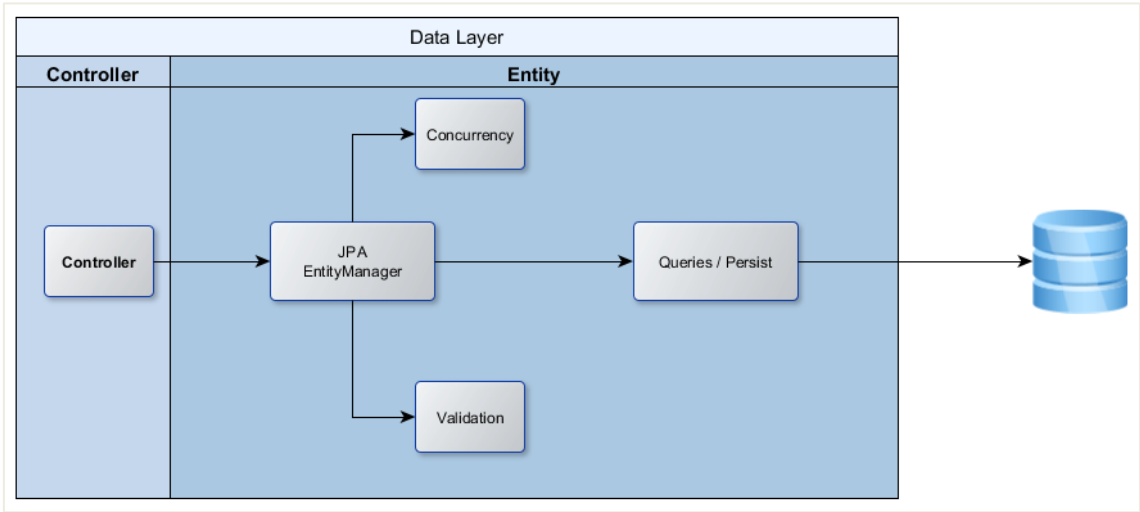


Ilustración 8 Interacción Acceso a Datos