**INFORME TÉCNICO**

**A:**  **ING. GUIDO NOGALES SUAZNABAR**

**DIRECTOR DE LA CARRERA DE INGENIERIA INDUSTRIAL**

**DE:** **ING. JUAN CARLOS MAMANI LAURA**

**CONSULTOR**

**REF: INFORME TÉCNICO CONSULTORIA POR PRODUCTO**

**ANTECEDENTES**

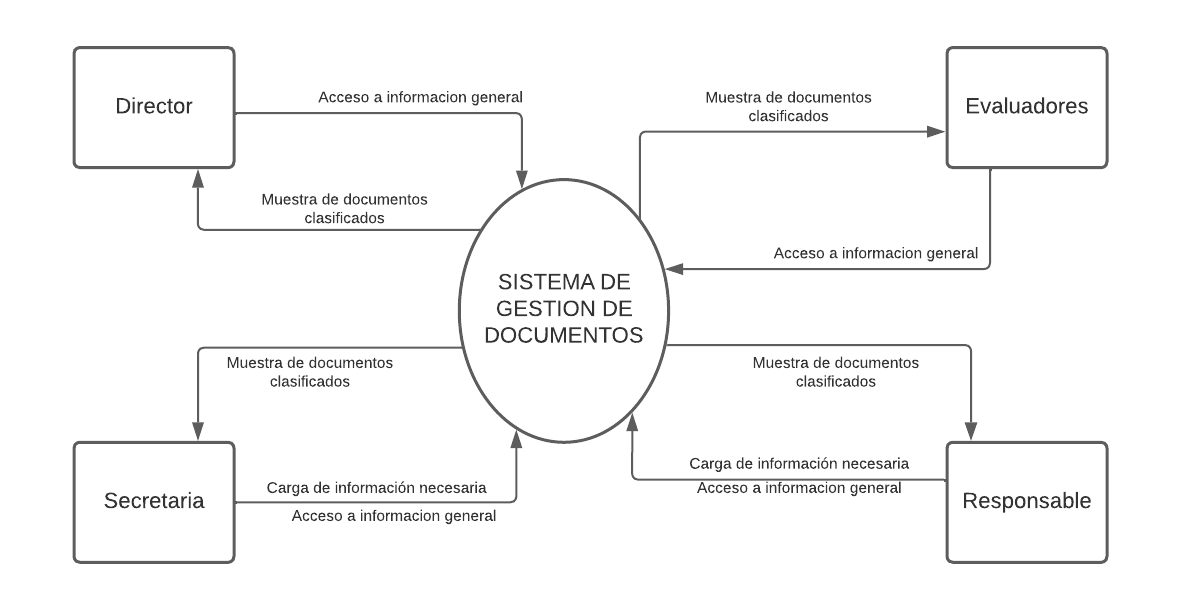
La carrera de ingeniería industrial es una de las carreras pioneras en la acreditación a la CUB y a MERCOSUR por lo cual en las últimas evaluaciones por parte de los pares evaluadores escribieron una serie de mejoras para la carrera en lo que concierne acreditación, una de las recomendaciones es la pronta digitalización de todos los documentos que hacen parte de la acreditación, en este sentido la dirección de la carrera decidió la construcción de un sistema que gestione la documentación de la acreditación en lo cual se llevó a cabo reuniones técnicas con la Unidad de Sistemas de Información y Comunicación de la UAP donde se definió el perfil del consultor para la construcción del sistema.

# **1.- DESARROLLO:**

Bajo las necesidades de la Carrera de Ingeniería Industrial se estableció el siguiente modelo de negocio obtenidos mediante una entrevista.

## **MODELO DE NEGOCIO**

Se describe la estructura operacional apoyada en un conjunto de diagramas que dan luz de forma clara y precisa toda la estructura funcional del Sistema de gestión de documentos.



El modelo define los límites del sistema por partes operativas donde se muestra gráficamente la interacción de entidades involucradas con el sistema.

|  |  |
| --- | --- |
| ENTIDADES | INTERES |
| Director | Su principal función del director obtener información requerida de su interés, así como la asignación de roles a otros usuarios |
| Responsable | Su función del responsable es de agregar documentos que se presentes físicamente, además de cargar documentos escaneados y realizar modificaciones. |
| Secretaria | Su función de secretaria es la misma que la de responsable y entra en función cuando no se encuentra la responsable. |
| Evaluadores | Su función es tener acceso visual a toda información requerida de su interés con la autorización del director. |

Ya definido la relación entre los usuarios y el sistema de gestión de documentos, se estableció los siguientes requerimientos funcionales y no funcionales.

REQUERIMIENTO FUNCIONAL

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Descripción** | **Código** |
| **Director** | El Sistema permitirá habilitar nuevos usuarios | RF-1 |
| El Sistema permitirá crear nuevos documentos | RF-2 |
| El Sistema mostrará la organización de documentos por niveles | RF-3 |
| El Sistema permitirá cargar documentos escaneados | RF-4 |
| El sistema permitirá realizar cambios en la información | RF-5 |
| El Sistema permitirá hacer búsqueda de documentos | RF-6 |
| **Responsables** | El Sistema permitirá crear nuevos documentos | RF-2 |
| El Sistema mostrará la organización de documentos por niveles | RF-3 |
| El Sistema permitirá cargar documentos escaneados | RF-4 |
| El sistema permitirá realizar cambios en la información | RF-5 |
| El Sistema permitirá hacer búsqueda de documentos | RF-6 |
| **Secretaria** | El Sistema permitirá crear nuevos documentos | RF-2 |
| El Sistema mostrará la organización de documentos por niveles | RF-3 |
| El Sistema permitirá cargar documentos escaneados | RF-4 |
| El sistema permitirá realizar cambios en la información | RF-5 |
| El Sistema permitirá hacer búsqueda de documentos | RF-6 |
| **Evaluadores** | El Sistema mostrará la organización de documentos por niveles | RF-3 |
| El Sistema permitirá hacer búsqueda de documentos | RF-6 |

REQUERIMIENTO NO FUNCIONAL

|  |  |  |
| --- | --- | --- |
| **Nro.** | **Descripción** | **Código** |
| **Sistema** | El Sistema restringirá el acceso por tipo de usuario | RNF-1 |
| El Sistema validará los tipos de datos al crear nuevos documentos | RNF-2 |
| El Sistema debe **poseer un diseño “Responsive”** a fin de garantizar la adecuada visualización en múltiples computadores personales y dispositivos tableta. | RNF-3 |
| El Sistema se ejecutará en un servidor Ubuntu server (Linux) | RNF-4 |
| E Sistema será desarrollado en un leguaje “**open source**”. | RNF-5 |

**ESTRUCTURA DEL SISTEMA**

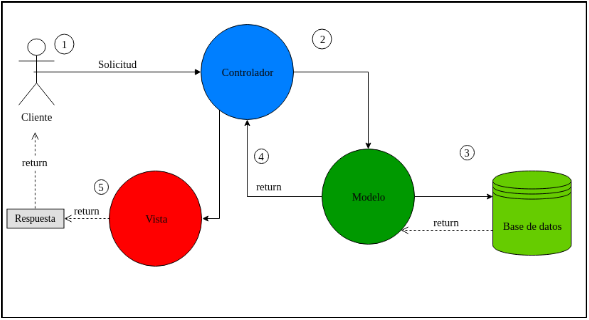
**BACK-END:** Se utilizará el framework Spring Boot 2 ya que es una herramienta versátil, seguro, optimo en el código, tiene una buena integración con los API REST y posee un mantenimiento seguro por PIBOTAL.

**FRONT-END:** Se utilizará el motor de plantillas Thymeleaf por su modularidad, velocidad de carga, trabajará con el BACK-END el cual tiene acceso con el Gestor de Base de Datos PostgreSQL, para complementar la seguridad de la información.

El sistema compondrá estas dos tendencias de desarrollo de software para garantizar la operatividad y modularidad, seguridad y factibilidad en el funcionamiento del sistema.

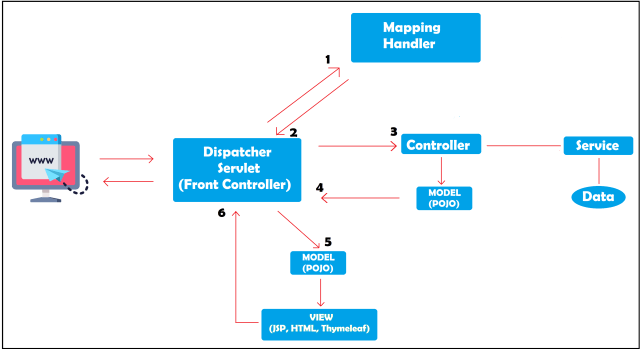
**ARQUITECTURA DEL SISTEMA**

El modelo vista controlador (MVC) es un patrón de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Este modelo de lo ve en aplicaciones web, donde la vista es el HTML y el código que provee de datos dinámicos a la página

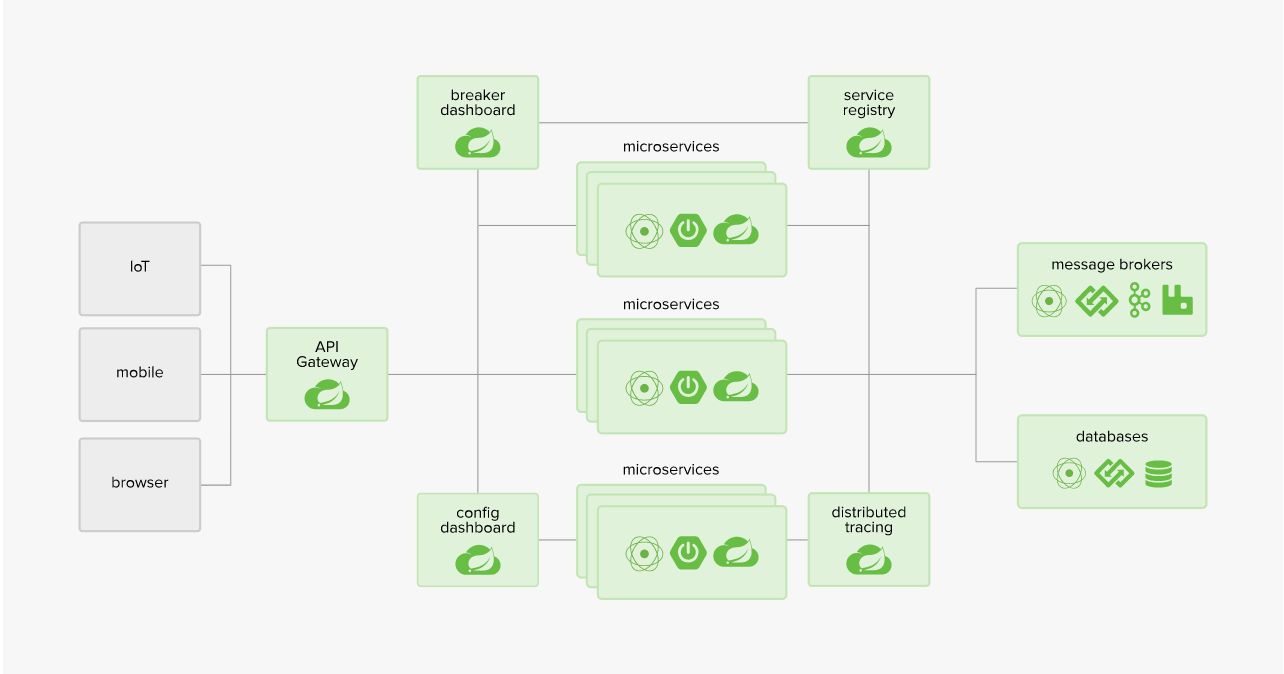


El sistema trabaja bajo las siguientes plataformas libres de desarrollo.

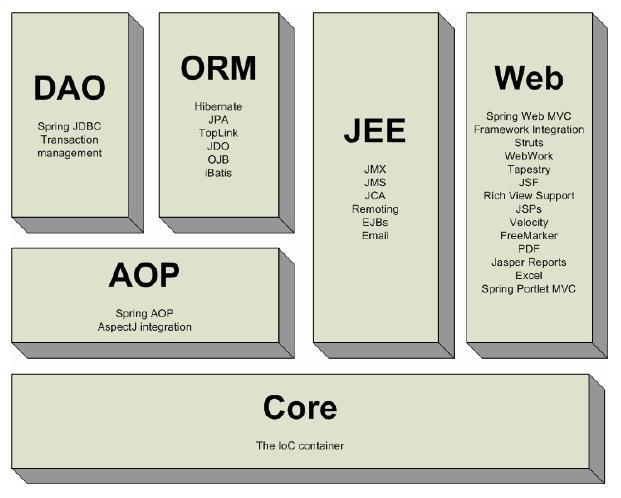
|  |  |
| --- | --- |
| Lenguaje de programación | JAVA |
| Motor de Base de Datos | Postgresql |
| Framework | Spring Boot |
| ORM | JPA Hibernate |
| Motor de Plantillas | Thymeleaf |



**ARQUITECTURA SPRING BOOT**



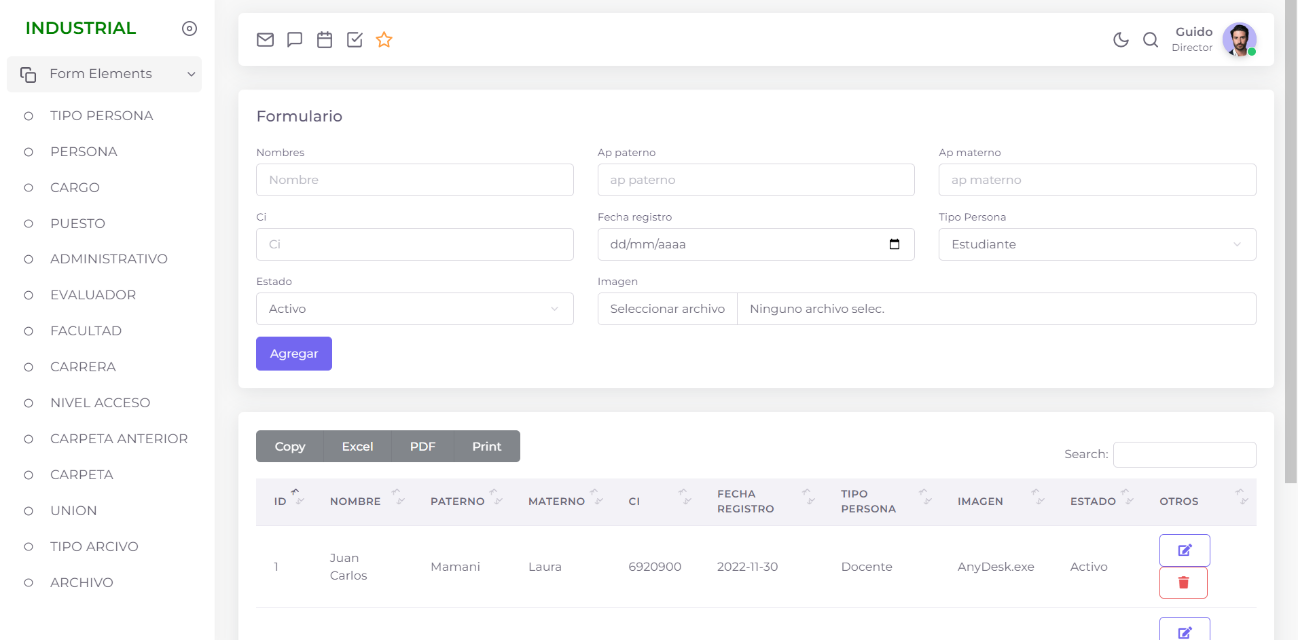
Spring es un framework contenedor liviano basado en la técnica de Inversión de Controles (IoC) y una implementación de desarrollo según el paradigma de Orientación a Aspectos (AOP).



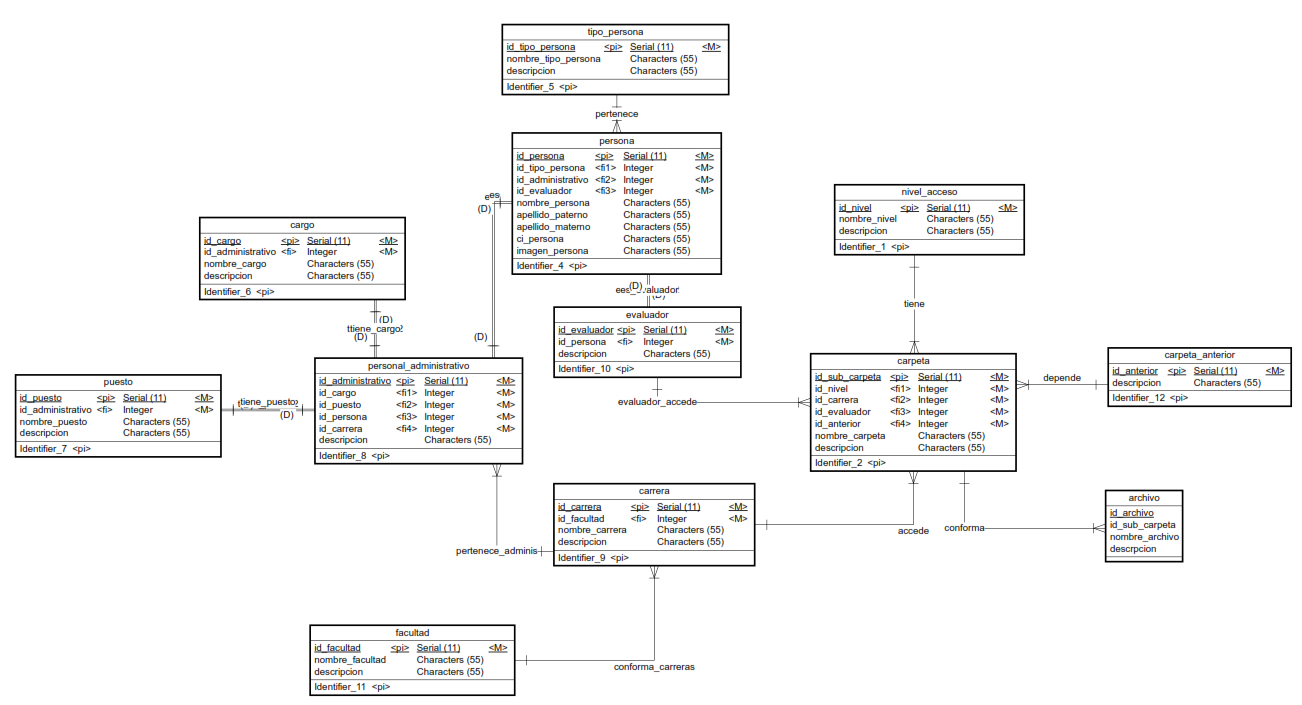
* DAO: Accesos JDBC con manejo de transacciones (desde el módulo AOP).
* ORM: Integración con Hibernate, JDO, etc.
* JEE: Acceso e interacción con servicios enterprise.
* Web: Provee un contexto apropiado para el desarrollo de aplicaciones web e integración con otros frameworks. (Struts, JSF, Tapestru, etc).
* POJO: (Plain Old Java Object) Revalora la simplicidad de las clases Java aportando manejo de transacciones de forma no intrusiva.
* Seguridad: Como un requerimiento no funcional implementado como un aspecto (AOP) a través del framwork Acegi.
* Testing: Provee un package de testing específico para componentes del framework e integrado con JUnit.
* Hibernate: Gestión integrada de transacciones y recursos.

**SISTEMA: PRODUCTO 1**

Módulos por requerimiento



En el cuadro seleccionado se muestran los módulos necesarios con las operaciones básicas de agregar, modificar y eliminar.



En la imagen se aprecia el diseño de la base de datos para una mejor integración del sistema para las funcionalidades