DOCUMENTO DE ARQUITECTURA

Tabla de contenido

1. Introducción

1.1. Propósito

El presente documento tiene por fin, dar a conocer la especificación de las historias de usuario que se han dado paulatinamente en el desarrollo de la plataforma Dock Management en donde se irán exponiendo los diferentes requerimientos tanto funcionales como no funcionales tenidos en cuenta, para la implementación del sistema en cuestión, el cual va dirigido a uno de los tres actores mencionados.

2. Modelo Entidad Relación

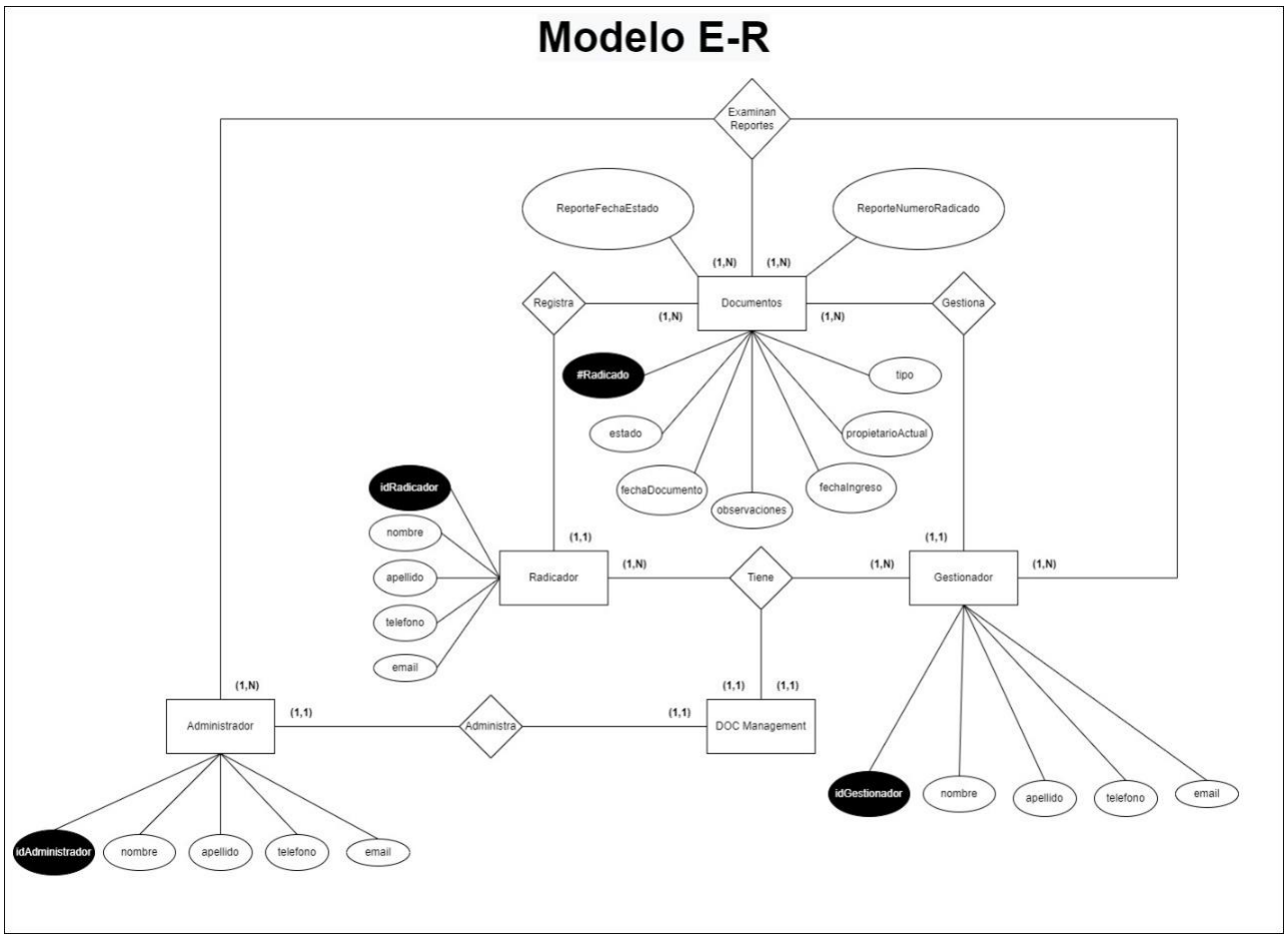

Diagrama de Entidad - Relación

3. Algoritmos implementados

- Algoritmos de búsqueda

  Los algoritmos de búsqueda están diseñados para comprobar o recuperar un elemento de cualquier estructura de datos en la que se almacene dicho elemento.

- Algoritmos de ordenación

  Son métodos para reorganizar un gran número de elementos en un orden específico, como de mayor a menor, o viceversa, o incluso en algún orden alfabético.

- Algoritmo Iterativo

  Ejecuta los pasos en iteraciones. Su objetivo es encontrar aproximaciones sucesivas en secuencia para llegar a una solución.

- Algoritmo recursivo

  Un algoritmo recursivo se llama a sí mismo con valores de entrada más pequeños y devuelve el resultado para la entrada actual.

- Algoritmo voraz

  Un algoritmo codicioso es un enfoque para resolver un problema seleccionando la mejor opción disponible en ese momento.

4. Buenas prácticas de desarrollo

4.1 Funcionales

- **Mantenga el código simple y consistente:**
  La idea es reducir la complejidad innecesaria. El código simple, a veces denominado código limpio, es más fácil de leer y de gestionar. La simplicidad y un estilo de código coherente también facilitarán el trabajo del equipo de desarrollo cuando llegue el momento de la actualización, sobre todo si los desarrolladores originales ya no están en el equipo.

- **Pruebas continuas:**
  Las pruebas continuas de extremo a extremo le darán más confianza en la calidad del código. También se asegurará de que el código se alinea con necesidades del usuario y que todos los componentes funcionan juntos como se espera. Las pruebas continuas también aumentan la cobertura del código.

- **Utilice varios recursos para comprobar el código:**
  Antes de enviar el código al equipo de pruebas, hacer que los desarrolladores comprueben su código con otro desarrollador. Además de identificar los errores en una fase más temprana del proceso de desarrollo, esto ayuda a los desarrolladores a aprender unos de otros para mejorar sus habilidades de codificación.

- **Establezca estimaciones de tiempo:**
  Los plazos cortos generan estrés. Del mismo modo, un tiempo excesivo puede hacer que los desarrolladores dejen las cosas para más tarde. Intentar encontrar ese punto óptimo que permita a los desarrolladores avanzar de forma rentable, pero sin exigirles demasiado, de modo que la calidad del código se vea afectada.

- **Utilizar el control de versiones**
  El control de versiones se refiere a un marco de ingeniería de software que rastrea todos los cambios y los sincroniza con un archivo maestro almacenado en un servidor remoto. Un sistema eficaz de control de versiones es un aspecto clave a la hora de escribir código de producción. El uso del control de versiones en un proyecto profesional le proporcionará los siguientes beneficios:

  1. En caso de que el sistema se caiga, se hace una copia de seguridad de todo el código en el servidor.
  2. Todos los miembros del equipo de ingeniería de software pueden sincronizar sus cambios regularmente descargándolos del servidor.
  3. Numerosos desarrolladores pueden trabajar de forma independiente para añadir/eliminar características o realizar cambios en un mismo proyecto, sin que ello afecte al trabajo de otros miembros.
  4. Si se producen errores graves, siempre se puede volver a una versión anterior y estable del código base que se sabe que funciona.

4.2 No Funcionales

- **Preparación del entorno**
  Esta etapa tiene como principal objetivo preparar las condiciones necesarias que permitan la ejecución exitosa del resto de las etapas definidas. Para lograr este objetivo se ejecutan las siguientes acciones: se identifican los interesados relevantes, es decir todos los que de una forma u otra esperan algo o intervienen en el proceso de desarrollo y deben involucrarse en las actividades de desarrollo de RNF (arquitecto, analista, desarrollador, probador, jefe de proyecto, clientes). Posteriormente se identifican las fuentes de conocimiento, para lo cual se propone la realización de una auditoría del conocimiento cuyos resultados deberán ser representados en un mapa de conocimiento, de esta manera es posible identificar las competencias y conocimiento existente, así como quien puede transferir o mejorar su nivel de apropiación de un conocimiento determinado, lo cual permitirá definir acciones que refuercen el conocimiento disponible.

- **Identificación y análisis de RNF**

  Esta etapa tiene como objetivo fundamental identificar los requisitos no funcionales aplicando métodos y técnicas que facilitan estas actividades. En primer lugar, se deberán identificar los requisitos de calidad de los clientes para lo cual se propone realizar entrevistas a los proveedores de requisitos, a partir de una guía de preguntas preconcebidas que indaguen en los requisitos de calidad, cuyos resultados deben quedar reflejados en el producto de trabajo Necesidades del cliente. Se deberá establecer un peso o importancia (Alto, Medio, Bajo, No Aplica) relativa de las características de la calidad, teniendo en cuenta las características del sistema a desarrollar, propósito y uso previsto.

- **Especificación de RNF**

  Esta etapa tiene como propósito documentar formalmente los RNF identificados. Para lo cual ejecutan las siguientes acciones: se describen los RNF identificados teniendo en cuenta un conjunto de recomendaciones para su redacción y una plantilla definida que permite organizar los elementos necesarios para su comprensión y posterior implementación. Se clasifican acorde al modelo de calidad definido por la NC ISO/IEC 25010, asociados a las características de calidad que define el modelo.

- **Verificación y validación de RNF**

  Esta etapa tiene como objetivo garantizar el cumplimiento de los atributos de calidad de la especificación de RNF así como su correspondencia con las necesidades de los interesados. Se ejecutan las actividades de verificación y validación según los procesos definidos en la institución (IPP-2016 Proceso para las actividades de calidad (PPQA, VER & VAL)). Durante esta etapa se ejecuta el proceso: Ejecutar Revisiones Técnicas Formales (RTF) de requisitos (IPP-2014 Ejecutar Revisiones Técnicas Formales a nivel de proyecto). Los expertos ejecutan las revisiones a partir de las listas de chequeo, donde se comprueban mediante un grupo de indicadores, los atributos de calidad de las especificaciones.

5. Arquitectura del Producto/Sistema

5.1 Vista Funcional

5.1.1 Modelo de Análisis

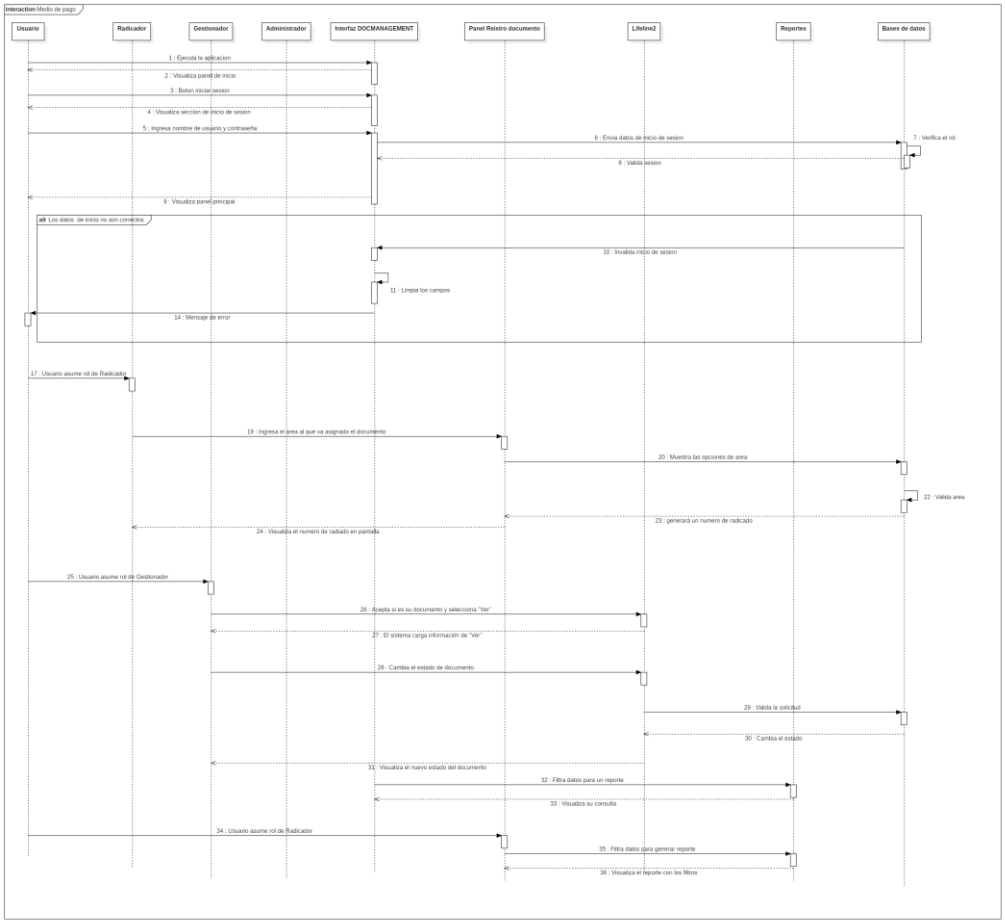

Diagrama de Secuencia

5.1.2 Vista de Despliegue - Ambiente Físico

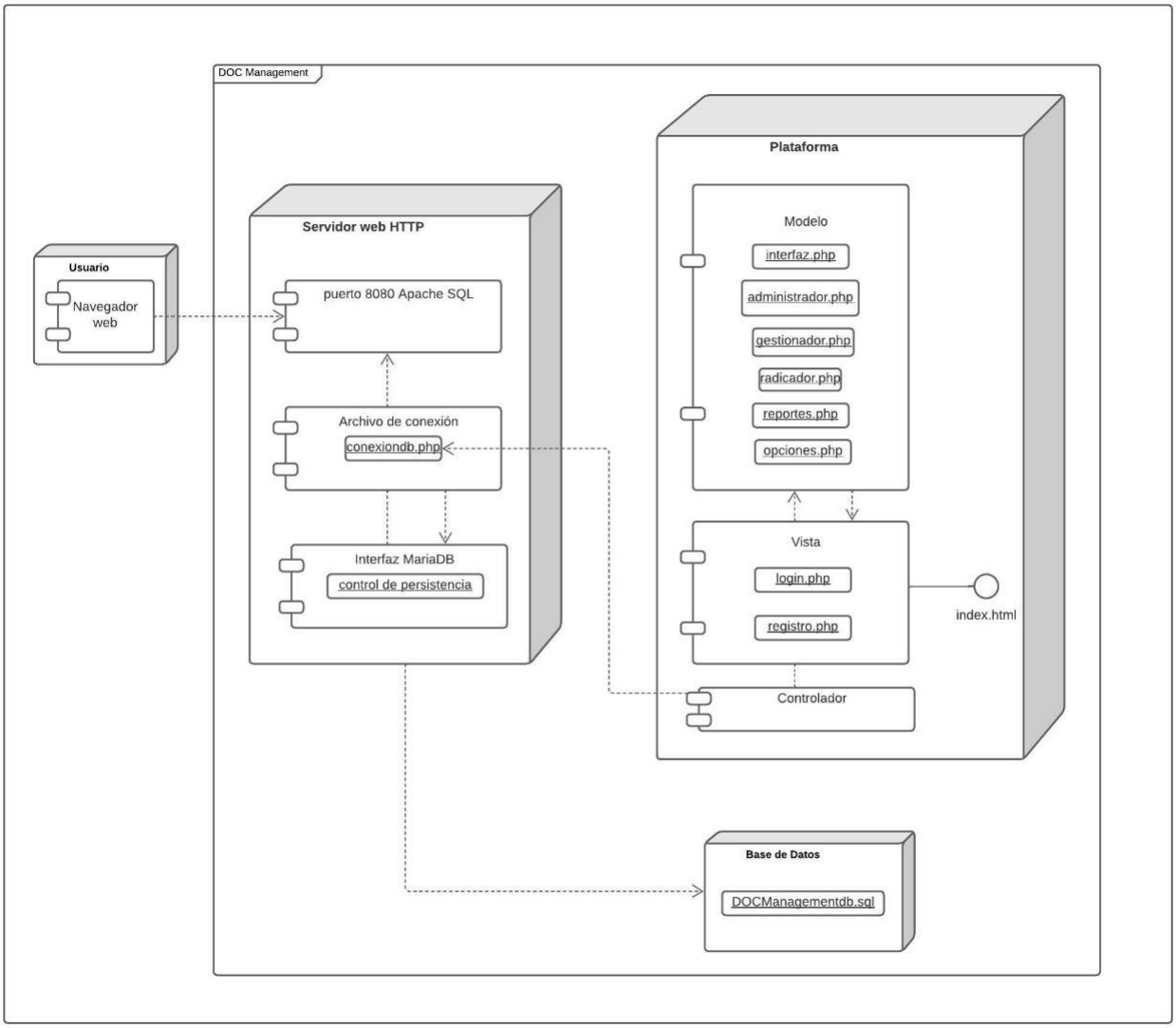

Diagrama de despliegue