ERS (Especificación de Requerimientos)

A continuación, se indicarán los requerimientos que se desarrollarán en la aplicación.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R001 | **Estado** | Pendiente – Desarrollo – **Finalizado** |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Base de Datos | |
| En este requerimiento se establecen la creación de tablas que lleva el proyecto. | | | |
| **Especificación** | | | |
| Para este proyecto se establecieron las siguientes tablas:  Museo (**8**)  Salas (**5**)  Colecciones (**5**)  Precios (**4**)  Temática (**5**)  Valoración (**5**)  Visitante (**7**)  Áreas (**2**)  ComisiónTarjeta (**3**)  Entrada (**7**)  Especie (**9**) | | | |
| **Justificación** | | | |
| Ejemplo: Este requerimiento es necesario para el buen funcionamiento del proyecto (nombre).  **Museo:** Esta tabla es fundamental porque almacena la información básica y principal de cada museo, como su nombre, ubicación y tipo. Sin esta tabla, no se podría gestionar ni identificar los museos dentro del sistema.  **Salas:** Permite clasificar las diferentes áreas o salas dentro de cada museo.  **Colecciones:** Guarda los datos de las colecciones que se exhiben en las salas. Es vital para la gestión del inventario cultural y para ofrecer información detallada sobre las piezas exhibidas.  **Precios:** Registra los precios de entrada a las salas, diferenciando días de la semana. Es indispensable para la gestión comercial y la venta de entradas.  **Temática:** Define las características y épocas de las colecciones o salas, ayudando a clasificar y contextualizar las exhibiciones.  **Valoración:** Permite registrar la opinión y calificación de los visitantes sobre las salas.  **Visitante:** Contiene la información de los visitantes registrados, necesaria para la gestión de entrada.  **Áreas:** Define las distintas áreas funcionales o administrativas del museo, facilitando la organización interna y la asignación de recursos.  **ComisiónTarjeta:** Administra las comisiones aplicadas a diferentes tipos de tarjetas para el pago de entradas, importante para la correcta facturación y control financiero.  **Entrada:**Registra cada entrada vendida o utilizada por los visitantes, con detalles como fecha, precio y método de pago. Es esencial para el control de acceso y la gestión económica.  **Especie:** Detalla las especies o piezas específicas dentro de las colecciones, incluyendo datos científicos y características. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R002 | **Estado** | Pendiente – Desarrollo – **Finalizado** |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Conexion | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar el proceso de conexión del servidor de base de datos Oracle, a javaFx | | | |
| **Especificación** | | | |
| Se desarrolló el proceso de conexión entre el servidor de base de datos Oracle y la aplicación JavaFX siguiendo un video tutorial. La conexión se configuró inicialmente en NetBeans, utilizando su apartado **Databases** para establecer la comunicación mediante JDBC. Posteriormente, en el proyecto JavaFX se creó una carpeta específica para manejar la conexión y la descarga de la información de las tablas, permitiendo que la aplicación acceda y utilice los datos de forma eficiente. | | | |
| **Justificación** | | | |
| Ejemplo: Esta conexión es necesaria para que la aplicación JavaFX pueda interactuar directamente con la base de datos Oracle, lo que permite consultar, actualizar y gestionar la información en tiempo real. Sin esta integración, la aplicación no podría acceder a los datos esenciales para su funcionamiento, limitando su utilidad y capacidad de respuesta. Además, organizar la conexión y manejo de datos en una carpeta específica facilita el mantenimiento y escalabilidad del proyecto. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R003 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant.Salas | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de salas(está dentro de mantenimiento) | | | |
| **Especificación** | | | |
| La tabla Salas estructura el espacio museístico en unidades manejables, proporcionando la base para relacionar colecciones, temáticas, precios y valoraciones específicas de cada sala dentro del museo. Se debe integrar con otras tablas, como la de museo, para saber donde enlazar información.  Se necesitan botones, para manejas con mayor facilidad el añadir o eliminar elementos.  Cuadros de texto para un mejor orden  Utilizar clases DAO o servicios para interactuar con la base de datos. | | | |
| **Justificación** | | | |
| Esta tabla es fundamental para representar las diferentes salas o espacios dentro de cada museo, vinculándolas con el museo correspondiente mediante el campo id\_museo. Permite almacenar el nombre y una descripción de cada sala, facilitando la organización y gestión de las áreas físicas donde se exhiben colecciones o se desarrollan actividades. Además, el campo nombremuseo puede usarse para mostrar el nombre del museo asociado, facilitando consultas y reportes. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R004 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant.Colecciones | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de collecciones | | | |
| **Especificación** | | | |
| directamente en un archivo FXML, definiendo los componentes necesarios como tablas, etiquetas, botones y campos para mostrar y gestionar las colecciones.  El controlador Java asociado manejará la lógica de negocio: cargar datos desde la base de datos (tablas Colecciones y relacionadas), actualizar la vista, responder a eventos de usuario (selección, búsqueda, edición).  Utilizar clases DAO o servicios para interactuar con la base de datos. | | | |
| **Justificación** | | | |
| La gestión de colecciones es fundamental porque permite organizar, conservar y proteger las piezas del museo, que suelen tener un valor histórico, cultural y económico incalculable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R005 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant.Especies | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de especies | | | |
| **Especificación** | | | |
| Se creará un archivo FXML que defina la estructura visual de la pantalla de especies, utilizando contenedores como VBox, HBox o BorderPane para organizar elementos como tablas, etiquetas, botones y campos de texto.  La pantalla permitirá al usuario consultar, filtrar y posiblemente editar o agregar registros de especies, mejorando la experiencia mediante una interfaz clara y responsiva.  Este desarrollo garantiza una pantalla de especies bien estructurada, mantenible y alineada  Utilizar clases DAO o servicios para interactuar con la base de datos. | | | |
| **Justificación** | | | |
| La pantalla de especies es fundamental porque permite gestionar de manera eficiente y organizada la información relacionada con las diferentes especies que forman parte de la colección del museo o institución. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R006 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant.Tematica | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de tematicas | | | |
| **Especificación** | | | |
| Se creará un archivo FXML que defina la estructura visual, usando contenedores como VBox, HBox o BorderPane  Declarar variables @FXML para los componentes UI (Tabla, botones, campos).  Implementar el método initialize() para cargar los datos iniciales y configurar la tabla.  Crear métodos @FXML para manejar eventos (ej. agregarTematica(), editarTematica(), eliminarTematica()).  Utilizar clases DAO o servicios para interactuar con la base de datos. | | | |
| **Justificación** | | | |
| Facilita la comprensión y contextualización del contenido exhibido, permitiendo que los visitantes y el personal accedan a información clara y ordenada que enriquece la experiencia educativa y cultural | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R007 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant.Precios | |
| Este requerimiento describe la forma a desarrollar de la pantalla de precios | | | |
| **Especificación** | | | |
| Utilizar clases DAO o servicios para interactuar con la base de datos.  Algunas funciones que podría implemengta son;   * **cargarPrecios** * **agregarPrecio** * **modificarPrecio** * **eliminarPrecio** | | | |
| **Justificación** | | | |
| importancia de gestionar de manera eficiente los ingresos derivados de la venta de entradas, servicios y productos. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R008 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant. Comisiones | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de comisiones | | | |
| **Especificación** | | | |
| Una tabla (TableView) para listar las comisiones con columnas relevantes (por ejemplo, nombre, porcentaje, descripción).  Cargará los datos de comisiones desde la base de datos a colecciones observables para mostrarlos en la tabla.  Gestionará eventos de usuario para realizar operaciones CRUD sobre las comisiones  Validará las entradas para asegurar que los datos sean consistentes y correctos.  Actualizará la interfaz en tiempo real según las acciones realizadas. | | | |
| **Justificación** | | | |
| Se fundamenta en la necesidad de optimizar la gestión administrativa y financiera del museo, | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R009 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Mant. Museo | |
| (Descripción general del requerimiento)  Este requerimiento describe la forma a desarrollar de la pantalla de museos | | | |
| **Especificación** | | | |
| Se crearán clases modelo que representen la entidad Museo, con atributos como id, nombre, dirección, teléfono, horarios y descripción, facilitando la manipulación y transferencia de datos.  El controlador utilizará la conexión JDBC para consultar, insertar, actualizar y eliminar registros de museos, garantizando la persistencia y consistencia de la información. | | | |
| **Justificación** | | | |
| es clave para mejorar la experiencia del visitante y la administración interna, como se observa en museos modernos que integran tecnologías interactivas y sistemas de gestión documenta | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R010 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Entradas | |
| Este requerimiento describe la forma a desarrollar la carpeta entradas | | | |
| **Especificación** | | | |
| Esta estructura modular permitirá actualizar, agregar o modificar recursos relacionados con las entradas sin afectar otras partes del sistema, facilitando la escalabilidad y el mantenimiento.  El código incluirá funcionalidades para crear nuevas entradas, asignarles datos personalizados y generar documentos o archivos (como PDF o tickets imprimibles) basados en plantillas almacenadas en la carpeta entradas.  El código incorporará validaciones para asegurar que los datos ingresados sean correctos y que las entradas generadas sean válidas y únicas, evitando errores o fraudes. | | | |
| **Justificación** | | | |
| La generación automática de entradas personalizadas (con datos como fecha, tipo de boleto y código QR) proporciona mayor comodidad y seguridad tanto para los visitantes como para el personal del museo. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R011 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | valoración | |
| Este requerimiento describe la forma a desarrollar la carpeta valoracion | | | |
| **Especificación** | | | |
| La carpeta debe mantener la integridad de los documentos y respetar su procedencia, asegurando que los archivos no se alteren ni se mezclen con otros que no correspondan  Debe tener espacios de texto donde especifique a que museo y detalles de mas se esta refiriendo, para dar su reseña. | | | |
| **Justificación** | | | |
| Se fundamenta en la importancia de contar con una estructura organizada y sistemática que permita gestionar, conservar y acceder eficientemente a la información relacionada con la evaluación y valoración de las colecciones del museo. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R012 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Reportes | |
| Este requerimiento describe la forma a desarrollar la carpeta reportes | | | |
| **Especificación** | | | |
| Permitan crear nuevas valoraciones, asociando los documentos y datos correspondientes.  Consulten valoraciones existentes, mostrando detalles y enlaces a los archivos.  Actualicen o eliminen valoraciones según sea necesario.  Asegurar que solo usuarios autorizados puedan modificar o acceder a documentos sensibles, implementando controles de acceso y validaciones en la carga y descarga de archivos. | | | |
| **Justificación** | | | |
| Es un componente clave para organizar y preservar la documentación relacionada con la evaluación museográfica, y el código asociado debe contemplar la gestión eficiente, segura y ordenada de estos archivos y sus metadatos, integrándose con la base de datos y la interfaz de usuario para ofrecer una solución completa y funcional. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R013 | **Estado** | Pendiente – **Desarrollo** – Finalizado |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | Controladores | |
| Este requerimiento describe la forma a desarrollar las funcionalidades de la carpeta de los controladores de cada tabla que lo necesite | | | |
| **Especificación** | | | |
| El controlador se declara en el archivo FXML mediante el atributo fx:controller="paquete.ClaseControlador", lo que permite que JavaFX asocie la interfaz visual con la clase que manejará sus eventos y datos.  En el controlador se implementan métodos que responden a acciones del usuario, como clics en botones o cambios en campos de texto, permitiendo que la interfaz sea interactiva. | | | |
| **Justificación** | | | |
| Los controladores en JavaFX son esenciales porque separan la interfaz visual (FXML) de la lógica del programa.Permiten gestionar eventos y acciones del usuario, haciendo la aplicación interactiva y dinámica. Además, favorecen la reutilización de código y la integración con herramientas de diseño, lo que acelera el desarrollo y mejora la experiencia del usuario. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Requerimiento** | R014 | **Estado** | Pendiente – Desarrollo – **Finalizado** |
| **Solicitante** | Hairol Romero | **Puesto** | Profesor |
| **Descripción** | **Módulo:** | (Nombre modulo) | |
| Este requerimiento describe la forma a desarrollar la carpeta fxml | | | |
| **Especificación** | | | |
| En el proyecto se crea una carpeta específica llamada **fxml** dentro del directorio **src/main/resources** o en una ubicación estándar de recursos, para que los archivos FXML sean correctamente reconocidos y cargados por la aplicación.  Un contenedor principal (como (VBox),(AnchorPane), (BorderPane) etc.) que engloba todos los elementos visuales.  Controles y nodos gráficos como botones (Button), etiquetas (Label), campos de texto (TextField)  Propiedades de los elementos, por ejemplo, texto, tamaño, alineación, eventos (como onAction)) | | | |
| **Justificación** | | | |
| FXML permite mantener la estructura y apariencia de la interfaz separadas del código Java que controla el comportamiento, lo que mejora la organización y legibilidad del proyecto. | | | |