Arquitectura del Sistema

Vesta Risk Manager

T-Code

Agustín Collareda, Cintia Hernandez, Hugo Frey



La Arquitectura del software, comprende el conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución.

Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto. Este documento describe en términos generales, las ideas principales detrás de la arquitectura escogida para el mismo.



Tabla de contenido

[Introducción 5](#_Toc179728747)

[Propósito 5](#_Toc179728748)

[Alcance 5](#_Toc179728749)

[Definiciones, Acrónimos, y Abreviaturas 5](#_Toc179728750)

[Referencias 6](#_Toc179728751)

[Representación Arquitectónica 6](#_Toc179728752)

[Representación 6](#_Toc179728753)

[Objetivos Arquitectónicos y Restricciones 7](#_Toc179728754)

[Objetivos Generales 7](#_Toc179728755)

[Objetivos Específicos 7](#_Toc179728756)

[Descripción de Procesos 7](#_Toc179728757)

[Vista de Caso de Uso 8](#_Toc179728758)

[Descripción de los Actores 8](#_Toc179728759)

[Contexto del sistema 8](#_Toc179728760)

[Vista Lógica 8](#_Toc179728761)

[Perspectiva General 8](#_Toc179728762)

[Paquetes de Diseño importantes arquitectónicamente 8](#_Toc179728763)

[Vista de Procesos 9](#_Toc179728764)

[Vista de Liberación 9](#_Toc179728765)

[Vista de Implementación 9](#_Toc179728766)

[Perspectiva General 9](#_Toc179728767)

[Capas 9](#_Toc179728768)

[Vista de Datos (opcional) 9](#_Toc179728769)

[Tamaño y Rendimiento 10](#_Toc179728770)

[Calidad 10](#_Toc179728771)

[Diagramas 11](#_Toc179728772)

[Diagramas de Despliegue 11](#_Toc179728773)

[Diagrama de Objetos 11](#_Toc179728774)

[Diagramas de Paquetes 12](#_Toc179728775)

Arquitectura del Sistema

Introducción

Propósito

Este documento tiene como propósito brindar una visión comprensible de la arquitectura general, utilizando diferentes vistas de la arquitectura para ilustrar diferentes aspectos del sistema. Captura las decisiones más importantes en lo que respecta a la arquitectura del sistema que fueron tomadas en el proyecto.

Desde el punto de vista de un desarrollador, este documento le brindará, con certeza, una buena razón para darle a la arquitectura de software la importancia que tiene en todo proyecto de desarrollo.

Desde el punto de vista de un cliente, este documento les permitirá comprender cómo la arquitectura del sistema responde a sus necesidades y requisitos, y les dará la confianza de que las decisiones técnicas tomadas son las adecuadas para alcanzar los objetivos del proyecto. Además, servirá como base para evaluar el progreso del desarrollo y la calidad del producto final, facilitando la comunicación y el intercambio de ideas entre todos los involucrados para asegurar el éxito del proyecto.

Alcance

El alcance de este documento abarca la descripción de la arquitectura general del sistema, incluyendo sus componentes principales, sus interacciones, y las decisiones de diseño relevantes que se tomaron durante el desarrollo del proyecto. Se incluirán diagramas y descripciones para proporcionar una comprensión clara de la estructura del sistema. Además, se abordarán aspectos relacionados con la escalabilidad, seguridad, rendimiento y mantenibilidad del sistema, considerando tanto los requisitos actuales como posibles necesidades futuras. El documento no cubrirá detalles de implementación específicos o codificación a nivel bajo, sino que se enfocará en los aspectos estructurales y decisiones arquitectónicas clave.

Definiciones, Acrónimos, y Abreviaturas

[Esta sección provee las definiciones de todos los términos, acrónimos y abreviaturas requeridas para interpretar apropiadamente el Documento de Arquitectura de Software. Esta información puede referenciar al Glosario del Proyecto.]

Referencias

[Esta sección provee una lista completa de documentos referenciados en cualquier lugar del Documento de Arquitectura de Software. Identifica cada documento por su título, número de reporte (si aplica), fecha, organización que publica, puede reverenciarse a un apéndice o a otro documento.]

Representación Arquitectónica

Las características inherentes al sistema hacen que la vista de casos de uso y la vista lógica sean las más relevantes y por ello serán las más extensas.

La arquitectura está representada por diferentes vistas utilizando notación UML de forma que permitan visualizar, entender y razonar sobre los elementos significativos de la arquitectura e identificar las áreas de riesgo que requieren mayor detalle de elaboración. Este documento es una forma de comunicar el modelo del subsistema, presentando la información y discusiones estructuradamente.

La arquitectura del subsistema se descompone en las siguientes dimensiones:

* Requerimientos: Requerimientos funcionales y no-funcionales del sistema.
* Elaboración: Representación lógica del sistema y representación de tiempo de ejecución.
* Implementación: Vista de módulos implementados, potenciales escenarios de infraestructura y el deployment de los módulos.

La siguiente sección detalla las vistas de la arquitectura que serán utilizadas para cubrir las dimensiones mencionadas.

Representación

La arquitectura del sistema está representada siguiendo las recomendaciones de RUP. Las vistas necesarias para especificar dicho subsistema se enumeran a continuación:

**Vista de Casos de Uso**: Describe el proceso de negocio más significativo y el modelo del dominio. Presenta los actores y los casos de uso para el sistema.

**Vista de Restricciones:** Describe restricciones tecnológicas, normativas, uso de estándares, entre otros, las cuales deben ser respetadas tanto por el proceso de desarrollo como por el producto desarrollado.

**Vista QoS:** Incluye aspectos de calidad, y describe los requerimientos no-funcionales del sistema.

**Vista Lógica:** Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias.

**Vista de Procesos:** Describe los procesos concurrentes del sistema.

**Vista de Implementación:** Describe los componentes de deployment construidos y sus dependencias.

**Vista de Datos:** Presenta los modelos de datos, los servicios de persistencia y los servicios de transaccionalidad utilizados.

**Vista de Deployment:** Presenta aspectos físicos como topología, infraestructura informática, e instalación de ejecutables. Incluye además plataformas y software de base.

Estas vistas son las generales para cualquier sistema que se intente desarrollar, para nuestro caso, solo nos explayaremos en las que son significativas para este desarrollo.

La implementación todas estas, están sujetas a las demandas propias del desarrollo.

Las vistas presentadas forman en su conjunto una especificación completa del sistema, la cual se delinea en el siguiente diagrama. Las dependencias entre las vistas indican dependencias entre las vistas tanto a nivel de arquitectura como a nivel de diseño.

Objetivos Arquitectónicos y Restricciones

[Esta sección describe los requerimientos y objetivos que tienen algún impacto significante en la arquitectura; por ejemplo, seguridad, garantía, privacidad, portabilidad, distribución, reuso. Captura restricciones especiales que puedan aplicar estrategias de diseño e implementación, herramientas de desarrollo, estructura del equipo, programación, código legado, etc.]

Objetivos Generales

[Esta sección describe de manera general los objetivos del proyecto, con el ánimo de comprender que motiva al grupo de desarrollo emprender el desarrollo]

Objetivos Específicos

[Esta sección describe los objetivos que tienen algún impacto significante en la arquitectura. Intenta describir detalladamente que es lo que se pretende obtener con el desarrollo del proyecto.]

Descripción de Procesos

[Esta sección describe los procesos que son significativos a la arquitectura del sistema se debe tener en cuenta el flujo de información y que personas interactúan con la misma]

Vista de Caso de Uso

Esta vista presenta la percepción que tiene el usuario de las funcionalidades del sistema. Se presenta el proceso de negocio más importante y los casos de uso críticos que se derivan de éste. Este capítulo provee el contexto y determina el alcance del resto del documento. Primeramente, se describe el Negocio, luego se presenta el modelo del dominio para el Sistema, se identifican actores y se detallan los casos de uso significativos.

Descripción de los Actores

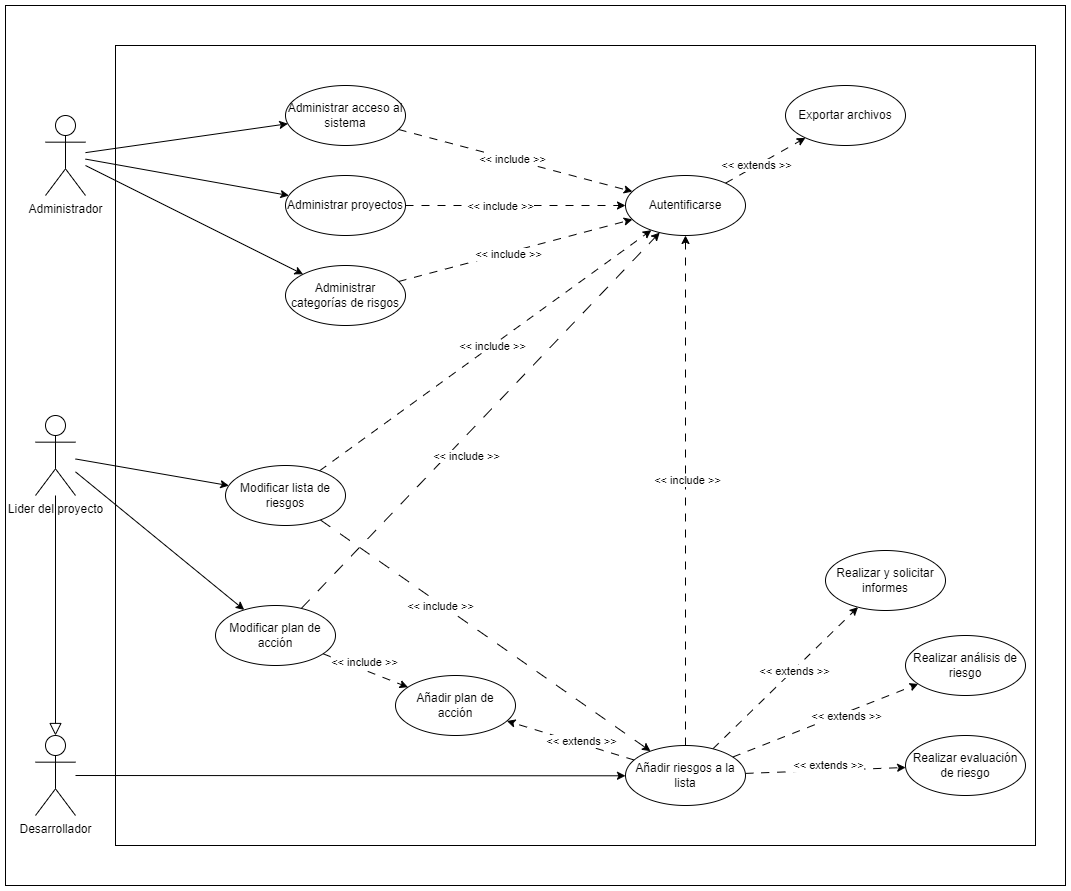
|  |  |
| --- | --- |
| Nombre del Actor | Administrador |
| Definición | Es el responsable principal de la gestión y configuración de la plataforma. Sus funciones clave incluyen la creación y modificación de proyectos, así como la gestión de usuarios, lo que implica crear cuentas y asignar personal a distintos proyectos. Este rol es esencial para mantener la estructura y organización del sistema. |

|  |  |
| --- | --- |
| Nombre del Actor | Líder del proyecto |
| Definición | Es el responsable de supervisar a los desarrolladores. Este rol tiene las mismas capacidades que un desarrollador, pero con funciones adicionales. Puede modificar los riesgos identificados, ajustar las clasificaciones, actualizar los planes de acción y programar los tiempos de ejecución. Su papel es crucial para mantener un control efectivo sobre el proceso de gestión de riesgos en los proyectos. |

|  |  |
| --- | --- |
| Nombre del Actor | Desarrollador |
| Definición | Representa a los participantes más comunes asignados a un proyecto específico. Sus capacidades incluyen añadir nuevos riesgos a la lista del proyecto, proponer planes de acción, llevar a cabo evaluaciones, exportar archivos, y generar informes y análisis de riesgo. Todas estas acciones las realizan a través de un sistema de autenticación que verifica su identidad y permisos dentro del proyecto. |

Contexto del sistema

Los casos de uso críticos para el proceso se describen en esta sección. Se comienza presentando las relaciones entre los casos de uso identificados, seguido por una versión expandida de los mismos. El diagrama a continuación muestra cómo los diferentes casos de uso interactúan entre sí y cómo está compuesto el contexto general del sistema.



Vista Lógica

[En esta sección se describen las partes significantes de arquitectura del modelo de diseño, tal como su composición dentro del sistema y paquetes. Y para cada paquete importante, su descomposición dentro de estructuras (clases) y utilidades de las mismas. Podría introducir estructuras arquitectónicas importantes y describir sus responsabilidades, así como relaciones, operaciones y atributos importantes.]

Perspectiva General

[Esta sección describe la composición general de modelo de diseño en términos de jerarquía y capas.]

Paquetes de Diseño importantes arquitectónicamente

[Para cada paquete significante, incluye una sección con su nombre, una descripción breve y un diagrama con estructuras significantes y paquetes contenidos dentro del paquete.

Para cada estructura significante en el paquete, incluye su nombre, una breve descripción, y, opcionalmente, una descripción de algunas de sus principales responsabilidades, operaciones y atributos.]

Vista de Procesos

[Esta sección describe la descomposición del sistema dentro del proceso ligeros (simple hilos de control) y procesos pesados (grupos de procesos ligeros). Organice la sección por grupos de procesos que se comunican e interactúan. Describe el principal modo de comunicación entre procesos, tal como mensajes, interrupciones.]

El proceso comienza cuando un equipo inicia un proyecto de desarrollo de software (o de otro tipo); durante el desarrollo del proyecto, antes de iniciar cada nueva iteración, se identifican los posibles riesgos; posteriormente se evalúan los riesgos identificados, otorgándoles una puntuación del 1 al 10 para su impacto y probabilidad; finalmente se calcula su factor de riesgo como el producto del impacto y la probabilidad del riesgo.

Si el factor de riesgo es igual o menor a 9, el riesgo se ignora; si es igual o menor a 36 se realiza una nueva evaluación antes de la próxima iteración; si es igual o menor a 64 se debe realizar un plan de riesgos para ese riesgo, aunque esto puede ser pospuesto hasta la siguiente iteración en el caso de que haya muchos riesgos para gestionar; si el riesgo es mayor a 64 se debe realizar un plan de riesgo bajo cualquier circunstancia.

Según el diseño del sistema, se encuentran las siguientes funcionalidades principales:

|  |
| --- |
| **Acceso al sistema** |
| Descripción: Permitirá al usuario ingresar al sistema con su cuenta de Google, la cual debe estar previamente registrada como Administrador, usuario estándar o espectador. |
| **Administración de acceso** |
| Descripción: Permitirá a los administradores registrar el email de los usuarios del sistema y asignarles un perfil de usuario (administrador, usuario estándar o espectador). |
| **Administración de proyectos** |
| Descripción: Permitirá a los administradores crear proyectos y editar sus atributos, como nombre, descripción, participantes, estado e iteraciones. Los participantes de un proyecto serán usuarios estándar del sistema, y podrán participar del proyecto con distintos roles (ej: líder del proyecto o desarrollador). Las iteraciones contaran con nombre, fecha de inicio y fecha de finalización. |
| **Administración de riesgos** |
| Descripción: Permitirá a los participantes de un proyecto agregar, eliminar y modificar los riesgos del proyecto. |
| **Administración de categorías de riesgo** |
| Descripción: Permitirá a los administradores agregar, eliminar o modificar las categorías de riesgo que se utilizaran en los proyectos. Estos cambios no se verán reflejados en proyectos que ya se encuentren en desarrollo. |
| **Administración de evaluaciones** |
| Descripción: Permitirá a los participantes de un proyecto realizar evaluaciones sobre los riesgos identificados. |
| **Administración de planes de acción** |
| Descripción: Permitirá a los participantes de un proyecto agregar, eliminar y modificar planes de acción para riesgos con factor de riesgo mayor a 36. También Permitirá agregar tareas a los planes de riesgo generados. No se Permitirá eliminar o modificar planes de acción que ya se encuentren en ejecución. |
| **Generación de informes** |
| Descripción: Permitirá a los participantes de un proyecto realizar informes o solicitarlos al sistema. |
| **Exportar archivos** |
| Descripción: Permitirá a los participantes de un proyecto exportar informes o graficos del proyecto. |

Vista de Liberación

[Esta sección describe una o más configuraciones físicas de la red (hardware) en las cuales el software se libera y funciona. Es una vista del modelo de liberación. En un mínimo para cada configuración debe indicar los nodos físicos (computadoras, CPUs) que ejecutan el software y sus interconexiones (bus, LAN, punto al punto, etcétera.) también incluye un mapa de los procesos de la vista de procesos sobre los nodos físicos.]

Vista de Implementación

[Esta sección describe la estructura total del modelo de la puesta en marcha, la descomposición del software en capas y subsistemas en el modelo de implementación y cualquier componente arquitectónico significativo.]

Perspectiva General

[Esta subdivisión nombra y define varias capas y su contenido, las reglas que gobiernan la inclusión a una capa dada, y los límites entre las capas. Incluir un diagrama componente que demuestre las relaciones entre las capas.]

Capas

[Para cada capa, se debe incluir una subdivisión con su nombre, una enumeración de los subsistemas situados en la capa, y un diagrama de componentes.]

Vista de Datos (opcional)

[Se debe colocar una descripción del almacenamiento de datos persistentes del sistema. Esta sección es opcional si hay o no datos persistentes, o si la traducción entre el modelo del diseño y el modelo de los datos es trivial.]

Tamaño y Rendimiento

[Colocar una descripción con las características principales de la dimensión del software que afectan la arquitectura, así como las restricciones de desempeño del objetivo.]

Calidad

[Describir cómo la arquitectura del software contribuye a la capacidad general (con excepción de funcionalidad) del sistema: extensibilidad, confiabilidad, portabilidad, etcétera. Si estas características tienen significación especial, por ejemplo seguridad, garantía o implicaciones de privacidad, deben ser claramente delineadas.]

Diagramas

Diagramas de Despliegue

[Básicamente este tipo de diagrama se utiliza para modelar el Hardware utilizado en la implementación del sistema y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos, componentes y asociaciones. En el UML 2.0 los componentes ya no están dentro de nodos, en cambio puede haber artefactos (archivo, un programa, una biblioteca o Base de datos) u otros nodos dentro de nodos.

Además los Diagramas de Despliegue muestran la configuración en funcionamiento del sistema incluyendo su software y su hardware. Para cada componente de un diagrama es necesario que se deba documentar las características técnicas requeridas, el tráfico de red, el tiempo de respuesta, etc.]

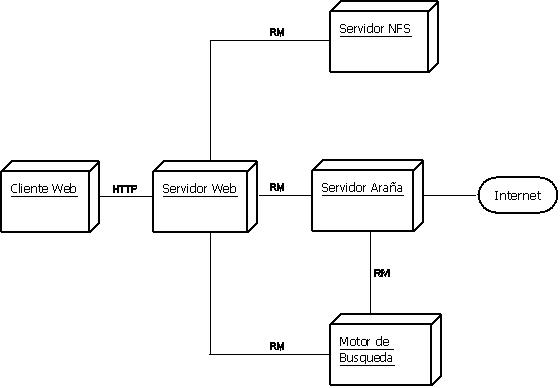


Diagrama de Objetos

[Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de la clase del Diagrama de Clases, cabe aclarar que el mismo cuenta con objetos y enlaces. En estos diagramas también es posible encontrar las clases para tomar como referencia su instanciación. En otras palabras el Diagrama de Objetos muestra un conjunto de objetos y sus relaciones en un momento concreto. Los Diagramas de Objetos son realmente útiles para modelar estructuras de datos complejas. ]



Diagramas de Paquetes

