# Proyecto:

Documento de Arquitectura del Sistema  
(SAD)

**Software Architecture Document**

**IEEE-1471-2000**

Equipo de Trabajo:

Aguirre Olvera Miguel Angel

|  |
| --- |
| TABLA DE CONTENIDO |

Contenido

[Proyecto: 1](#_Toc38841578)

[LISTADO DE FIGURAS 3](#_Toc38841579)

[LISTADO DE TABLAS 4](#_Toc38841580)

[SECCIÓN 1: DESCRIPCIÓN DEL DOCUMENTO 5](#_Toc38841581)

[IEEE-1471-2000 5](#_Toc38841582)

[SECCIÓN 2: CONCEPTUAL FRAMEWORK 5](#_Toc38841583)

[2.1 Architectural description in context 6](#_Toc38841584)

[2.2 Stakeholders and their roles 6](#_Toc38841585)

[2.3 Architectural activities in the life cycle 6](#_Toc38841586)

[2.4 Stakeholders 6](#_Toc38841587)

[SECCIÓN 3: ARCHITECTURAL DESCRIPTION PRACTICES 7](#_Toc38841588)

[3.1 Architectural documentation 7](#_Toc38841589)

[3.2 Identification of stakeholders and concerns 7](#_Toc38841590)

[3.3 Selection of architectural viewpoints 7](#_Toc38841591)

[3.4 Architectural views 7](#_Toc38841592)

[3.5 Consistency among architectural views 13](#_Toc38841593)

[descripción de módulos 13](#_Toc38841594)

[SECCIÓN 4: REQUERIMIENTOS DE CALIDAD o ARCHITECTURAL RATIONALE 14](#_Toc38841595)

[4.1 Árbol de Utilidad 14](#_Toc38841596)

[4.2 Escenarios de Calidad Priorizados 16](#_Toc38841597)

[SECCIÓN 5: RELACIONES ENTRE LAS VIEWPOINTS 17](#_Toc38841598)

[Arquitectura Lógica 17](#_Toc38841599)

[SECCIÓN 6: SIMBOLOGÍA 18](#_Toc38841600)

|  |
| --- |
| LISTADO DE FIGURAS |

Figura 1 caso de uso

Figura 2 diagrama de clases

Figura 3 diagrama de componentes

Figura 4 diagrama de despliegue

Figura 5 diagrama de secuencia login

Figura 6 diagrama de secuencia registrar

Figura 7 diagrama de secuencia actualizar

Figura 8 diagrama de secuencia borrar

Figura 9 diagrama de secuencia reporte

|  |
| --- |
| LISTADO DE TABLAS |

Tabla 1 listado de Stakeholders

Tabla 2 Stakeholders y expectativas

Tabla 3 identificación de Stakeholder y escenario

Tabla 4 vistas

Tabla 5 descripción de módulos

Tabla 6 árbol de calidad

Tabla 7 escenarios priorizados

|  |
| --- |
| SECCIÓN 1: DESCRIPCIÓN DEL DOCUMENTO |

|  |
| --- |
| 1.1 Propósito y Alcance |

El siguiente documento está dirigido a los usuarios de este sistema para ir aumentando su capacidad y su funcionalidad está dirigido también para un proyecto de riesgos de la ciudad de México y su propósito es informar a la población el posible peligro que pueda haber en su localidad y las medidas precautorias que pueden tomar

|  |
| --- |
| 1.2 Organización del Documento |

Este documento está dividido en 6 secciones donde la primera es una introducción del documento y un contexto del porque se está realizando el proyecto

En la segunda sección damos información de los interesados en el proyecto

En la tercera parte empezamos a detallar la arquitectura que vamos a utilizar para este proyecto

En la cuarta parte contamos con la calidad del proyecto y los puntos que se atienden

En la quinta parte del documento tenemos son las relaciones que existen entre los puntos de vista con los atributos de calidad

Y la sexta parte es la simbología utilizada en este documento en forma de lista y detallado

|  |
| --- |
| 1.3 Convenciones |

N/A

|  |
| --- |
| 1.4 Terminología y Definiciones |
|  |

Administrador: persona encargada de gestionar la información almacenada en la base de datos

Cliente: todo aquel que necesite información proveniente del sistema

|  |
| --- |
| 1.5 Documentos Relevantes |

# IEEE-1471-2000

|  |
| --- |
| SECCIÓN 2: CONCEPTUAL FRAMEWORK |

|  |
| --- |
| 2.1 Architectural description in context |

Este documento presenta la arquitectura como una serie de vistas basadas en la arquitectura de software del modelo 4+1 de Kruchten. Estas vistas son: la vista de escenarios, la vista lógica., la vista de desarrollos, la vista física, la vista de procesos. No hay ninguna vista separada de una misma implementación, descrita en este documento. Estas vistas están hechas sobre Lenguaje de modelo unificado (UML)

|  |
| --- |
| 2.2 Stakeholders and their roles |

| Stakeholder | Roles | Expectativas de Calidad descritas |
| --- | --- | --- |
| Alfonso Gregorio Rivero Duarte | Profesor | Que el alumno cumpla con los requerimientos del proyecto final |

|  |
| --- |
| 2.3 Architectural activities in the life cycle |

* Identificar y especificar los requerimientos guía. Éstos incluyen a los atributos de calidad, los requerimientos funcionales primarios y las restricciones del sistema. La arquitectura se puede diseñar alrededor de estos.
* Diseñar la arquitectura e, idealmente, implementar una “arquitectura ejecutable” que permita materializar el diseño de la arquitectura.
* Documentar los aspectos fundamentales del diseño, teniendo especial cuidado en capturar las decisiones de diseño, para comunicarlas al equipo y que sirvan de guía.
* Realizar una evaluación poco después de que se ha terminado el diseño y antes de proceder a la construcción del sistema.

|  |
| --- |
| 2.4 Stakeholders |

Tabla 1: Listado de los Stakeholders

| Stakeholder | Descripción |
| --- | --- |
| Alfonso Gregorio Rivero Duarte | Profesor de la universidad iberoamericana materia impartida Arquitectura de Software |

Tabla 2: Stakeholders y Expectativas

| Stakeholder | Expectativas según 9126 |
| --- | --- |
| Alfonso Rivero Duarte | Cumplir con la funcionalidad del proyecto y los patrones implementados |

|  |
| --- |
| SECCIÓN 3: ARCHITECTURAL DESCRIPTION PRACTICES |

|  |
| --- |
| 3.1 Architectural documentation |

se usarán diagramas de Clases para la parte lógica

para el despliegue un diagrama de componentes

para el proceso un diagrama de actividad

para la parte física un diagrama de despliegue

para la parte vista de escenario se usará casos de uso

|  |
| --- |
| 3.2 Identification of stakeholders and concerns |

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Descripción** | **Escenario** | **Vistas** |
| **Alfonso Rivero** | **Profesor evaluador** | **Escenario complete** | **todo** |

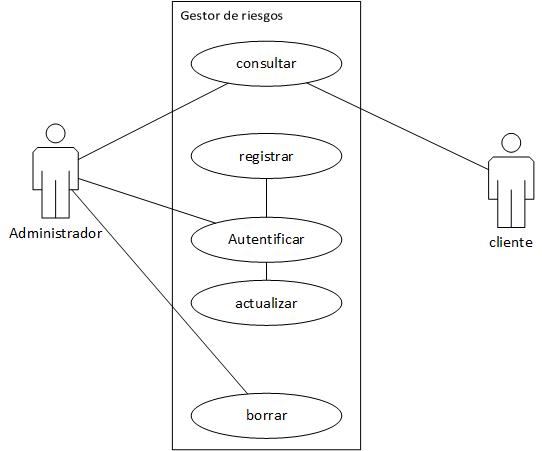
|  |
| --- |
| 3.3 Selection of architectural viewpoints |

|  |  |
| --- | --- |
| Vistas | UML |
| Escenarios | Casos de uso |
| Lógica | Clases |
| Desarrollo | Componentes |
| Física | Despliegue |
| Procesos | Secuencia |

|  |
| --- |
| 3.4 Architectural views |

Vista Escenario

Diagrama de caso de uso



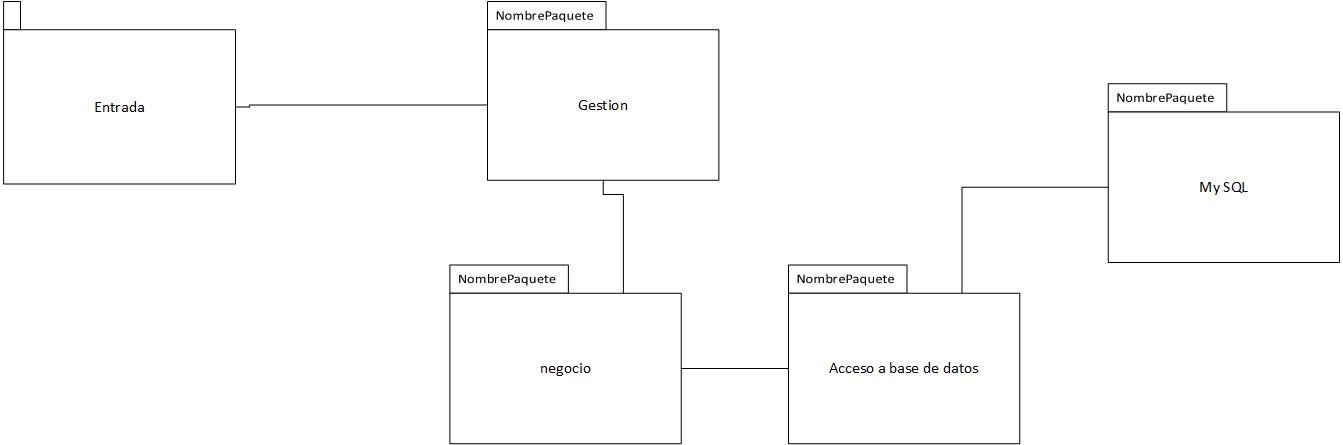
Vista lógica

Diagrama de Clases



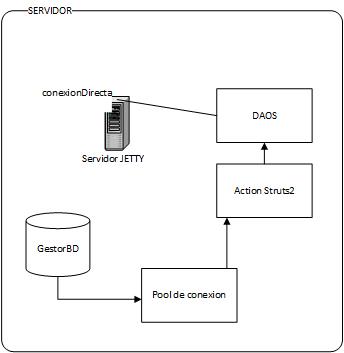
Desarrollo

Diagrama de componentes general



Vista Física

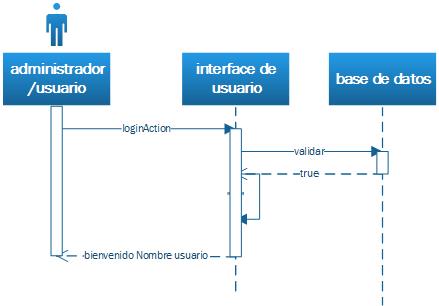
Diagrama de despliegue



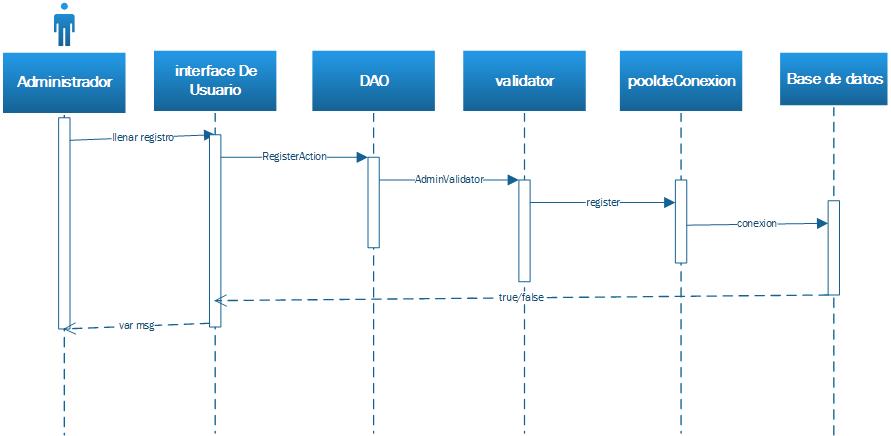
Vista-Procesos

Diagrama de secuencia

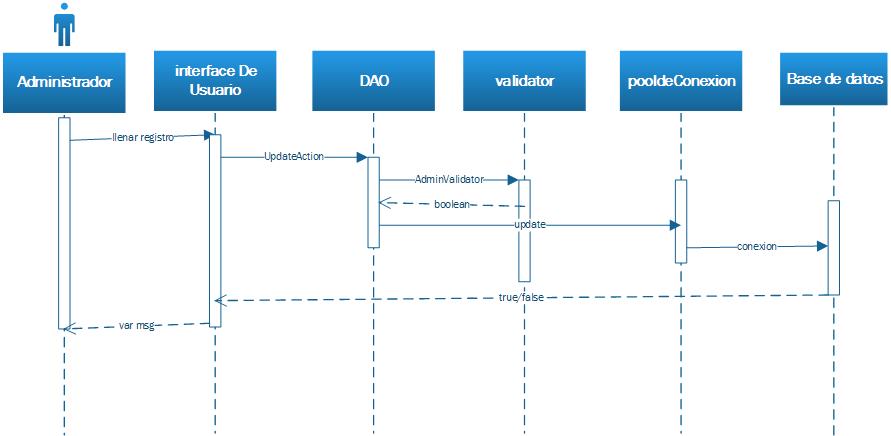
Login



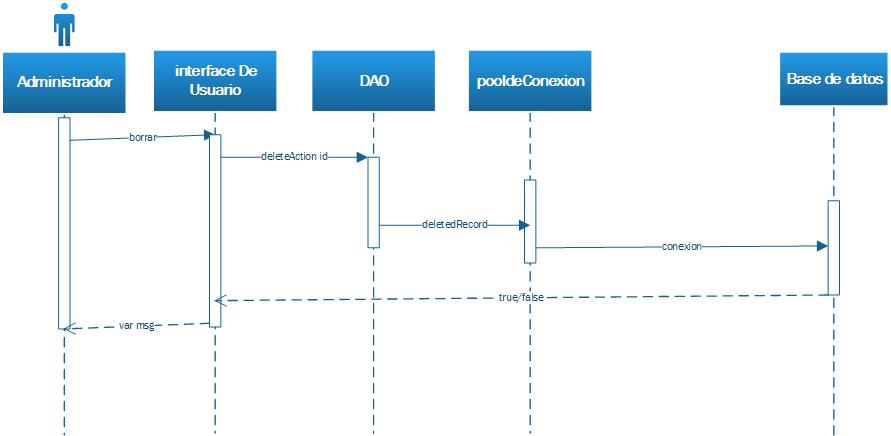
Registro



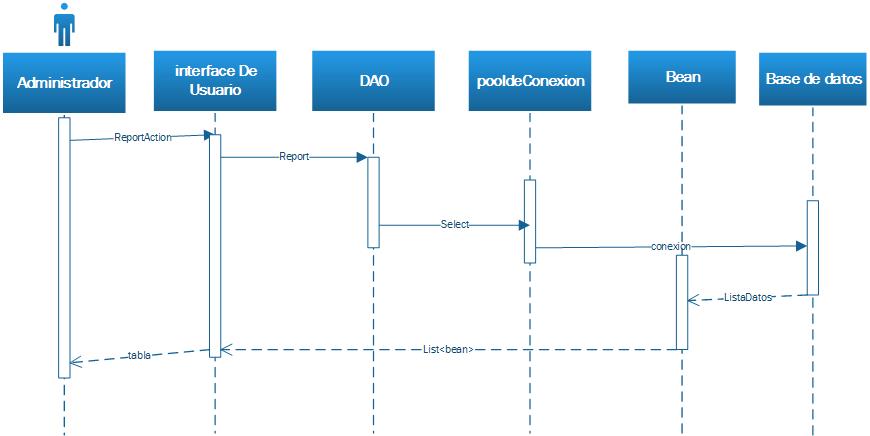
Actualizar



Borrar



Reporte



|  |
| --- |
| 3.5 Consistency among architectural views |

## descripción de módulos

|  |  |  |
| --- | --- | --- |
| Nombre de modulo | descripción | Componentes inclusos |
| gestión | Modulo que agrupa las clases encargadas de hacer las funciones de la aplicación | gestión de usuarios  Gestión de reportes |
| Negocio | Modulo encargado de agrupar todas las acciones del frame y los patrones | Action  ActionRiesgo  ActionLogin  ActionReport  ActionRegistro  ActionUpdate  ActionDelete |
| Acceso de datos | Modulo encargado de la conexión a la base de datos y la comunicación a la vista | Pool de conexión  Validator  Daos |

|  |
| --- |
| SECCIÓN 4: REQUERIMIENTOS DE CALIDAD o ARCHITECTURAL RATIONALE |

|  |
| --- |
| 4.1 Árbol de Utilidad |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Desempeño | |
| Latencia | ID | Descripción | Prioridad |
| Pool de conexión | 1 | El sistema debe de tener mayor capacidad para responder a la conexión | 2 |
| Escalabilidad |  |  |  |
| Pool de conexión | 2 | Con el pool podemos conectarnos a mas servidores de base de datos | 1 |
| Información de usuario | 3 | Puede crecer la información para ser más completa | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Fiabilidad | |
| Tolerancia a Fallas | ID | Descripción | Prioridad |
| Funciones | 4 | El usuario no puede borrar la información | 3 |
| Crud | 5 | Conexión con las demás tablas | 3 |
| Recuperabilidad |  |  |  |
| Vista | 6 | Reporte completo con la información que la complementa | 1 |
| Disponibilidad |  |  |  |
| Pool de conexión | 7 | Mas conexiones en el pool | 1 |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Mantenimiento | |
| Modificación | ID | Descripción | Prioridad |
| Framework Struts2 | 8 | El sistema si es mantenible, pero cambiar de frame para más seguridad | 1 |
|  |  |  |  |
| Facilidad de Pruebas |  |  |  |
| Test action | 9 | Se puede tener test de los action del frame, pero no podemos tener métodos completos ya que se ejecuta todo cuando se llama | 2 |
| Test pool | 10 | Se puede hacer test en el pool para un motor diferente de base de datos | 1 |
| Estabilidad |  |  |  |
| Jsp | 11 | Los métodos del frame se ejecutan antes de hacer cualquier cosa en la vista | 1 |
| Flexibilidad |  |  |  |
| Framework | 12 | Struts 2 no permite mucha estabilidad para el cambio | 2 |
| Escalabilidad |  |  |  |
| Diseño del proyecto | 13 | El proyecto es tan cerrado que solo tiene una sola función en general y no se puede hacer mucho | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo de Calidad: | | Seguridad | |
| Integridad | ID | Descripción | Prioridad |
| Login | 14 | Mejorar el login para que solo sea una sesión | 3 |
| Borrar | 15 | Que el usuario no pueda borrar ni modificar | 2 |
| Confidencialidad |  |  |  |
| url | 16 | El id de lo que se va a modificar se muestra en las url | 1 |
| Contraseñas de usuario | 17 | Ser mas robustos con las contraseñas | 1 |
| Disponibilidad |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| 4.2 Escenarios de Calidad Priorizados |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Escenario de Calidad #** | | **14** | **Stakeholder:** | **Prioridad:3** |
| **Atributo de Calidad** | Seguridad | | | |
| **Justificación** | Se necesita un loggin para que solo el administrador entre | | | |
| **Fuente** | Usuario | | | |
| **Estímulo** | Modificación para entrar directamente como administrador o usuario | | | |
| **Artefacto** | Login LoginDAO | | | |
| **Ambiente** | seguridad | | | |
| **Respuesta** | Que se permita un tipo de usuario con prioridades | | | |
| **Medida de la Respuesta** | Modificar el método login y buscar la manera de que se mantenga en el sistema durante una sesión | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Escenario de Calidad #** | | **8** | **Stakeholder:** | **Prioridad:2** |
| **Atributo de Calidad** | mantenimiento | | | |
| **Justificación** | Cambio de frame para buscar una seguridad | | | |
| **Fuente** | Usuario | | | |
| **Estímulo** | Difícil diseñar y manejar sus principales desventajas es la seguridad | | | |
| **Artefacto** |  | | | |
| **Ambiente** | Mantenimiento | | | |
| **Respuesta** | Buscar un frame que tenga los requisitos suficientes para cumplir con la seguridad que no cuenta este frame | | | |
| **Medida de la Respuesta** | Manera de migrar el sistema a un ambiente más seguro | | | |

|  |
| --- |
| SECCIÓN 5: RELACIONES ENTRE LAS VIEWPOINTS |

## Arquitectura Lógica

Performance

La arquitectura de software apoya al requerimiento de la aplicación

1. El sistema soporta hasta ahora 10 usuarios conectados gracias al pool de conexión
2. El sistema es capas de modificar la base de datos siempre y cuando se haya validado todo

Calidad

1. El sistema entrega la información requerida
2. El sistema cumple su propósito

Ejemplo de uso

1. Se puede usar para visualizar la información y prevenir algún problema local

Detalles de implementación

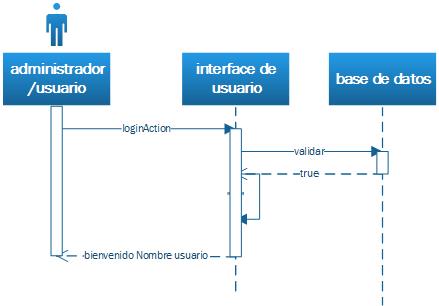
Se necesita el código para implementar todo lo que viene en este documento y la modificación probable que se puede hacer mas adelante sin problema alguno el código esta diseñado de manera monolítica ya que se encuentra la vista, el controlador y el modelo en un solo servidor

Lenguaje y plataforma

El lenguaje utilizado es Java con un Framework llamado Struts2 la vista está hecho en JSP y con html, freemarker y Bootstrap para los detalles estéticos

Se puede implementar con cualquier lenguaje orientado a objeto, pero la parte de las acciones de Struts se tendrá que modificar por parte de los desarrolladores

|  |
| --- |
| SECCIÓN 6: SIMBOLOGÍA |



Usuario / administrador

Objeto inicializado

## 

Vida del objeto

Mensaje

Caso de uso

clase

Base de datos

Paquete