**Documento de Arquitectura de Software IEEE-1471-2000**

# Control del documento

**Proyecto**

***Sistema de Configuración de Software***

# Título

***Arquitectura del Sistema – [v1.0 al 23 de Agosto de 2019]***

# Generado por

***Universitario* - *[Aldo Aliende]***

***Universitario - [Aracely Cristaldo]***

***Universitario - [Paolo López]***

***Universitario - [José Servián]***

***Universitario - [Mateo Vera]***

# Aprobado por

**Prof. Lic. Lilian Demattei**

**Prof. Lic. Lilian Riveros**

# 1. Introducción

## 1.1 Propósito

El presente documento tiene como propósito definir de forma clara y precisa los requerimientos, funcionalidades, condiciones y acuerdos que deberá cumplir el sistema, así como servir de guía oficial para el proceso de diseño, creacion e implementacion del sistema.

## 1.2 Alcance

El presente documento contiene el diseño elaborado para el proyecto ***Sistema de Gestion de Configuración de Software para la Implementación y Diseño de un Proyecto,*** el cual es producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios, además abarca la definición de los requerimientos funcionales, las especificaciones de los roles.

El documento está organizado alrededor de tres ideas principales.

1. Las características generales del diseño

2. Los requisitos atendidos por el diseño

3. Los modelos y vistas que lo detallan

La última sección del documento indica la forma en que se puede obtener el código fuente del proyecto así como las instrucciones de compilación necesarias para lograr la ejecución de los componentes que este código detalla.

* 1. Usuarios Interesados

Este documento de Arquitectura de Software (DAS), puede ser usado por todos aquellos usuarios que deseen comprender el diseño y construcción de la aplicación de ***Sistema de Gestión de Configuración de Software para la Implementación y Diseño de un Proyecto,*** y sirve como base para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje que más les acomode.

* 1. Recomendaciones de conformidad con esta práctica.

N/A.

## Referencias

Las referencias aplicables a este documento son:

- IEEE-1471-2000

* + IEEE 830 - 1998
  + The 4+1 View .Kruchten - 1009

## Definiciones, acrónimos y abreviaciones.

SGCS: Sistema de Gestión de Configuración de Software.

DAS: Documento de Arquitectura de Software

HTTP: Protocolo de Transferencia de Hipertexto.

TCP: Protocolo de control de transmisión.

ARQUITECTURA DE SOFTWARE: 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.

DESCRIPCION DE ARQUITECTURA: colección de productos de documentación.

VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

TIPOS DE VISTAS: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

STAKEHOLDER: Individuo, equipo u organización con intereses relativos al sistema.

ESCENARIO: especifica el comportamiento y limita el interés de un área específica del sistema para uno o varios stakeholders.

MODULO O COMPONENTE: cualquier elemento estructural abstracto, visible, externo, de alto nivel, analizable, que pueda constituir una funcionalidad de la solución del sistema.

ATRIBUTOS DE CALIDAD: un atributo de calidad, es una cualidad deseable de la solución, que pueda manifestarse en forma de requerimiento no funcional, que pueda ser medible, testeable y finalmente evaluable.

Usuario: Persona que usará el sistema y al cual podrá acceder a través de un

nombre de usuario y una contraseña.

Permiso: derechos o autorizaciones que se establecen en el sistema para poder

realizar ciertas operaciones.

Rol: Es un conjunto de permisos que pueden ser creados y modificados en el

sistema para asignar a los usuarios.

Sprint o línea base: Intervalo de tiempo predefinido en el que se asignar un conjunto definido de tareas a ser realizadas de manera a entregar un avance del producto.

User Stories (US): Es la representación de un requisito utilizando el lenguaje

común del usuario

Estado:​ Parámetro que indica avance, aplica para proyectos, user stories, sprint o líneas base.

Backlog:​ Conjunto de User Stories.

Product Backlog: Conjunto de User Stories correspondientes a un proyecto

específico.

Item: Es la representación de un requisito utilizando el lenguaje común del usuario

## Framework Conceptual

* 1. **Descripción de la arquitectura en contexto**

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) en su versión 2.0 desarrolladas usando Dia 0.97.2 y Sql Power Architect 1.0.8 ,python 3.7.4 y Django 2.2.4 (paolo) (aldo)

Los estilos arquitectónicos serán referenciados en este documento de arquitectura, según las recomendaciones de la Arquitectura de software del modelo 4+1 de Kruchten.

## Stakeholders y sus roles

Este documento representa la identificación de Stakeholders y sus roles a partir de la interpretación de los casos de uso del Proyecto.

## Actividades de arquitectura en el ciclo de vida

N/A.

## Usos de las descripciones de arquitectura

Las descripciones de arquitectura de este documento se usarán para referenciar el diseño del SGCS.

## Descripciones prácticas de arquitectura

N/A.

## Documentación de la arquitectura

N/A.

## Identificación de los Stakeholders y sus responsabilidades

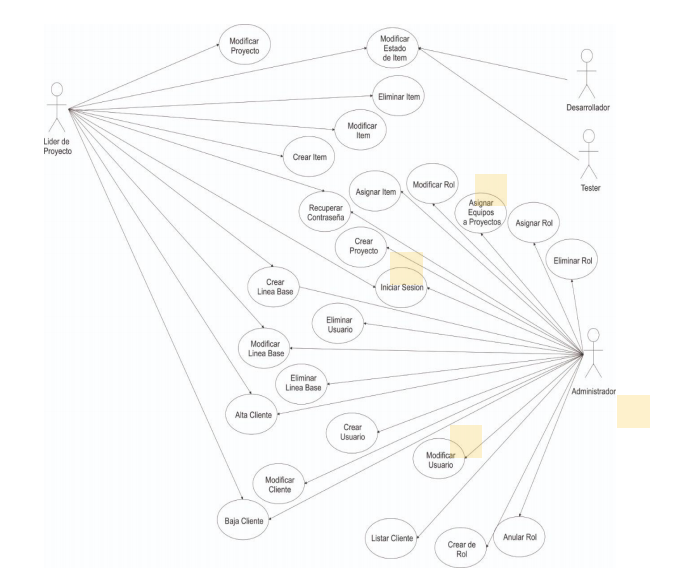
|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Descripción** | **Escenario** | **Vistas** |
| Líder o Scrum master | Es el usuario dueño de dar las tareas al team scrum. | Escenario crear línea base  Escenario de listar líneas base  Modificar línea base  Eliminar línea base  Asignar team al proyecto  Crear los items  Editar items  Eliminar items | CU Crear línea base  CU Listar línea base  CU Editar línea base  CU Eliminar línea base  CU Asignar team al proyecto  CU Crear los ítems  CU Editar ítems  CU Eliminar ítems |
| Administrador | Es la persona que interactúa con el sistema para crear roles y usuario cuando se crea un proyecto. | Escenario de creación de usuarios  Escenario modificar usuario.  Escenario eliminar usuario | CU Crear Rol  CU Modificar Rol  CU Eliminar Rol  CU Consultar Roles del Proyecto |
| Product Owner | Es la persona responsable de tener una visión de lo que desea construir, y transmitir esa visión a todo el equipo. | Escenario a un usuario un proyecto  Escenario de creación del proyecto  Escenario de editar del proyecto  Escenario de Listar los proyectos  Escenario de eliminar los proyectos | CU Asignar a un usuario un proyecto  CU Crear proyecto  CU Listar proyectos  CU Editar proyectos  CU Eliminar proyectos |
| Desarrollador | Persona encargada de realizar las tareas disponibles en una línea base. | Escenario modificar estado del ítem. | CU modificar estado. |

* 1. **Selección de puntos de vista de la arquitectura.**

|  |  |
| --- | --- |
| **Vistas** | **UML** |
| Escenarios | Casos de uso |
| Lógica | Clases |
| Procesos | Secuencia |

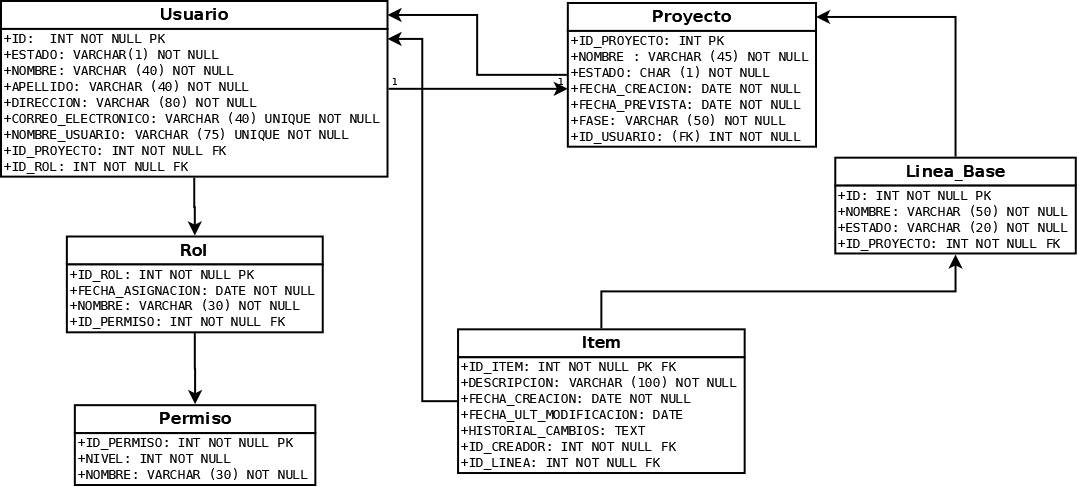
* 1. **Vistas de la arquitectura Vistas.- escenarios**

**Diagrama.- Caso de uso del negocio – Caso de uso de diseño**

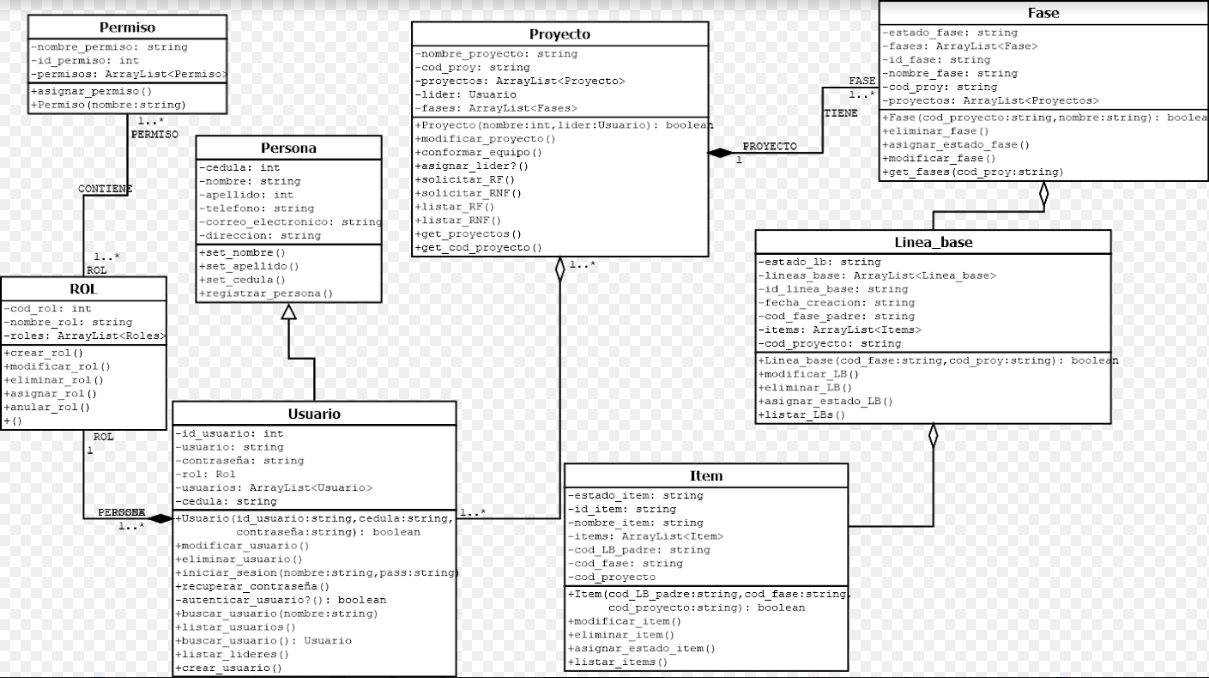
****

## Vista.- Lógica

## DER

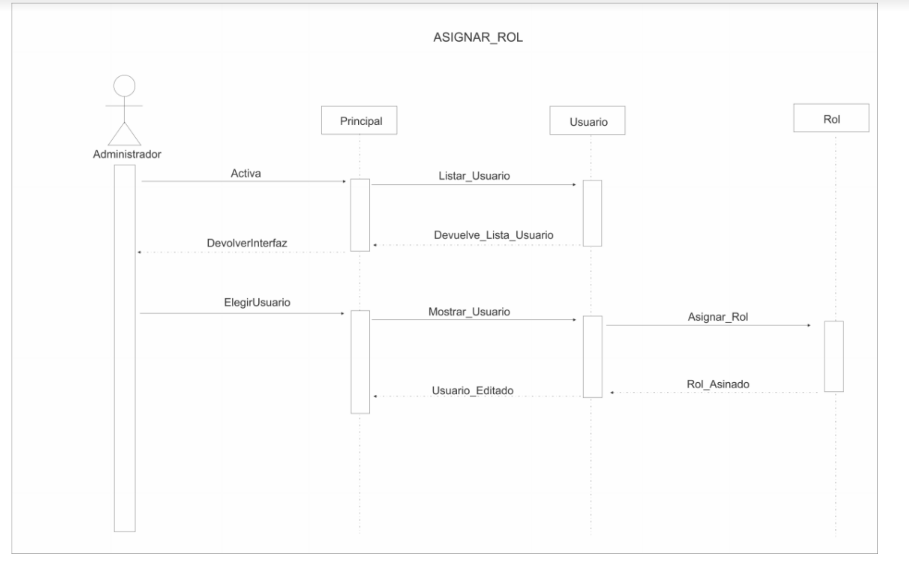


## Diagramas.- Clases

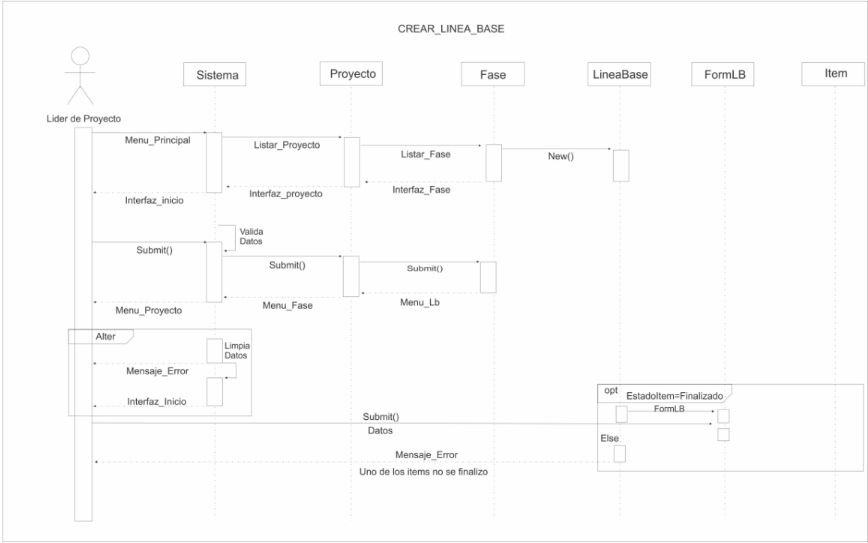


**Vista.- Procesos**

**Diagrama de secuencia (Asignar rol)**

****

## Diagrama de secuencia (Crear línea base)

****

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Diagrama de secuencia (Crear proyecto)

## Consistencia en la cantidad de vistas de la arquitectura.

**DESCRIPCION DE MODULOS**

|  |  |  |
| --- | --- | --- |
| **Nombre del módulo** | **descripción** | **Componentes inclusos** |
|  | Módulo que agrupa las clases e interfaces encargadas de orquestar las clases y agrupan las funcionalidades | * Administración   De permisos.   * Administración de roles. * Administrador de usuarios |
|  | Módulo que agrupan todas las clases que componen un proyecto , cada clase contiene su propia interfaz para exponer la funcionalidad de las otras capas | * Proyecto. * Fase. * Línea Base * Ítem. |
|  | Módulo que hace la persistencia a la base de datos. | * Proyecto\_d. * Fase\_d * Línea\_Base\_d * Ítem\_d |

**DESCRIPCIÓN DE COMPONENTES**

|  |  |  |
| --- | --- | --- |
| **Nombre del componente** | **descripción** | **Componentes relacionados** |
| P | Contiene la lógica para nuevos proyectos ,eliminar y actualizar | * Usuario * Fase |
|  | Contiene la lógica para nuevas líneas base eliminar actualizar | * Fase * Ítem |

**DESCRIPCIÓN DE CONECTORES**

* 1. **Arquitectura lógica.**

**Performances**

La arquitectura de software escogida apoya a los requerimientos no funcionales y requerimientos de arquitectura de sistemas descritos en los anexos de este documento.

1. El sistema apoyará hasta 2000 usuarios simultáneos contra la base de datos central en cualquier tiempo dado, y hasta 500 usuarios simultáneos contra los servidores locales en un momento dado.
2. El sistema proporcionará el acceso a la base de datos de catálogo de curso de herencia sin más que una 10 segunda latencia.
3. El sistema debe ser capaz de completar el 80 % de todas las transacciones dentro de 2 minutos.
4. La parte de cliente requerirá el espacio de disco de menos de 20 MB y la RAM de 32 MB.

## Calidad

La arquitectura de software apoya las exigencias de calidad, como estipulado en la especificación anexa a este documento.

1. El interfaz de usuario será WEB.
2. El interfaz de usuario del Sistema ProjectManager será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal establecidas por ISO 9126.
3. Cada despliegue de opciones de pantalla, tendrá la ayuda en línea para el usuario. La ayuda En línea incluirá paso a paso instrucciones en la utilización del Sistema. La ayuda En línea incluirá definiciones para términos y acrónimos.

## Ejemplo de uso.

N/A.

## Detalles de la implementación

La especificación de un sistema intensivo en software tiene como última representación al código fuente de los componentes. Este código indica los más finos detalles del software, por medio de un lenguaje preciso, capaz de ser traducido automáticamente a instrucciones de la maquina. Acompaña al código, las llamadas *previsiones de compilación*, constituidos por todos los elementos de soporte necesarios para realizar la construcción de los componentes a partir del conjunto de códigos. Esta

sección detalla la obtención y uso del paquete de código fuente para el proyecto. De manera de facilitar el uso de este, para las futuras ampliaciones o correcciones del sistema.

## 5.8.1 Lenguajes y plataformas

La lógica de diseño arquitectónico aplicada en este documento, abre la posibilidad de que la implementación de bajo nivel sea efectuada con lenguajes que solamente cumpla con la característica de Orientación a Objetos (Punto NET, Java, SmallTalk,Python etc.). Y eso va a depender directamente de las características de los desarrolladores, capacidad de aprendizaje, y en muchos casos opciones propias de la empresa para la cual se efectúa el diseño. Si la implementación se desea desarrollar bajo lenguajes que no cumplan las características mencionadas, se deberá confeccionar una nueva vista que cumpla con los requerimientos funcionales y no funcionales de los stakeholders que lo solicitan.