Documento de Arquitectura de

Software IEEE-1471-2000



## Control del documento

**Proyecto *Sistema de Configuración de software***

**Titulo Arquitectura del Sistema V 2.0**

**Generado por:**

**José Ibáñez**

**Javier Paredes**

**Emigdio Franco**

**Adam Duarte**

**Rodolfo Aranda**

## Introducción

### Propósito

Este documento proporciona una descripción comprensiva arquitectónica del sistema, usando un número finito de vistas diferentes para representar los distintos aspectos que se requieren para capturar y transportar las decisiones significativas que han sido hechas sobre el sistema.

### Alcance

El presente documento contiene el diseño elaborado para el proyecto **Sistema de Configuración de Software*,*** 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.

El documento está organizado alrededor de tres módulos principales.

1. Módulo de Administración
2. Módulo de Gestión de configuración
3. Módulo de Desarrollo

Al contrario de muchas otras actividades técnicas, el desarrollo de sistemas intensivos en software dedica la mayoría de sus esfuerzos a la especificación y modelado.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.

Los modelos son presentados por vistas o diagramas, generalmente utilizando notaciones gráficas como el UML.

Por otro lado, los programas de computadora son construidos por medio del uso de herramientas de traducción automáticas llamados compiladores, para los cuales es construida la forma lineal y más detallada del software del sistema: el código fuente.

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.

Este documento ha sido generado directamente del análisis del **Sistema de Configuración de Software** La mayoría de las secciones ha sido extraída del Modelo de ieee 1471(Institute of Electrical and Electronics Engineers ).

* 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 web de **Sistema de Configuración de Software** , y sirve como base para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje python en su versión 3.3 con el framework Django en su versión 2.2.

* 1. Recomendaciones de conformidad con esta práctica.

N/A.

### Referencias

Las referencias aplicables a este documento son:

- *IEEE 830-1998 ST*

*- IEEE 1471-2000*

- *Metodologías Ágiles(Scrum)*

1. **Definiciones, acrónimos y abreviaciones. DAS:** Documento de Arquitectura de Software.

**CONFIGURACIÓN DE SOFTWARE:** Sistema que se especializa en la gestión de configuración de las actividades en el sector del desarrollo de software.

**STAKEHOLDER**: Individuo, equipo u organización con intereses relativos al sistema **UML**: Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

**SCRUM**: Es un marco de trabajo para desarrollo ágil de software.

**MÓDULO DE ADMINISTRACIÓN:** Módulo que permite la administración de usuarios y roles, además de configuraciones del sistema.

**MÓDULO DE GESTIÓN DE CONFIGURACIÓN**: Módulo que permite la generación de líneas bases. Deberá permitir el control de las líneas bases generadas por producto y la administración de los cambios de los elementos que se encuentren en las líneas bases. **MÓDULO DE DESARROLLO:** Módulo que administra todos los elementos de los productos.

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

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

**COMPONENTES:** es una unidad modular de un programa software con interfaces y dependencias bien definidas que permiten ofertar o solicitar un conjunto de servicios o funcionales.

**FRAMEWORK:** Es un framework diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web

### Framework Conceptual

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

Django es un framework de aplicaciones web gratuito y de código abierto (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

En el patrón de diseño es MTV, basado en el MVC(Modelo-Vista-Controlador)

Debido a que la "C" es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un *Framework MTV*

* + - ***M significa "Model" (Modelo)***, la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
    - ***T significa "Template" (Plantilla),*** la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.
    - ***V significa "View" (Vista),*** la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelos y las plantillas.
  1. **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 Negocio.

### 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 sistema de software de Sistema de Gestión de Configuración del Software.

### Descripciones prácticas de arquitectura

N/A.

### Documentación de la arquitectura

N/A.

### Identificación de los Stakeholders y sus responsabilidades

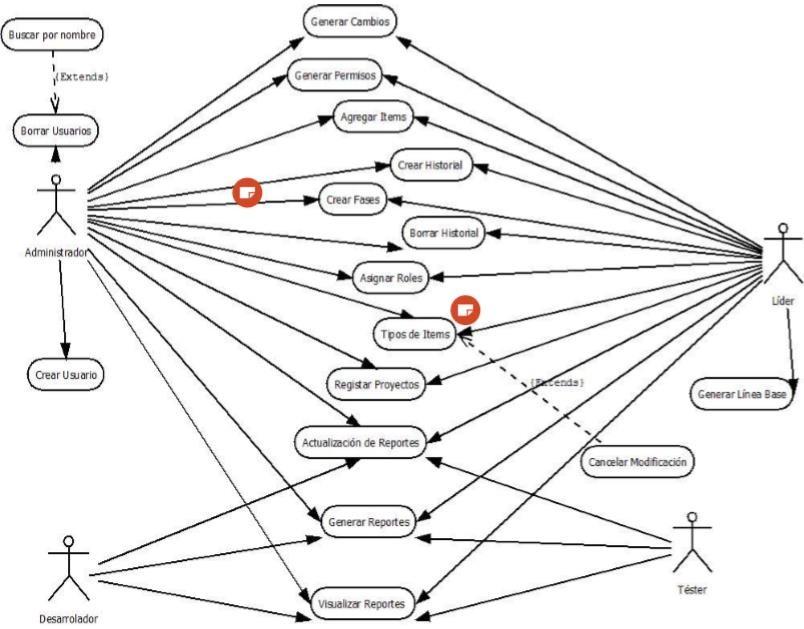
|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **descripción** | **escenario** | **Vistas** |
| Product Owner | Es el actor principal o primordial por ser quien interactúa con más casos de uso, otorga los permisos y tiene acceso a todas las acciones.  . | Escenario de negocios  Escenario de diseño | * Generar cambios * Generar Permisos * Agregar Item * Crear Historial * Borrar Historial * Asignar Roles * Registrar Proyectos * Actualización de Reportes * Generar Reportes * Visualizar Reporte * Borrar Usuarios * Crear Usuarios * Tipos de Ítem * Crear Fase |
| Scrum Master | Encargado de gestionar correctamente donde van las informaciones del sistema. | * Escenario de negocios   Escenario de diseño | * Generar cambios * Generar Permisos * Agregar Item * Crear Historial * Borrar Historial * Asignar Roles * Registrar Proyectos * Actualización de Reportes * Generar Reportes * Visualizar Reporte * Tipos de Ítem * Crear Fase * Generar Línea Base |
| Desarrollador | Es el actor encargado de hacer las implementaciones correctamente las especificaciones del sistema | Escenario de diseño | * Actualización de Reportes * Generar Reportes * Visualizar Reporte |

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

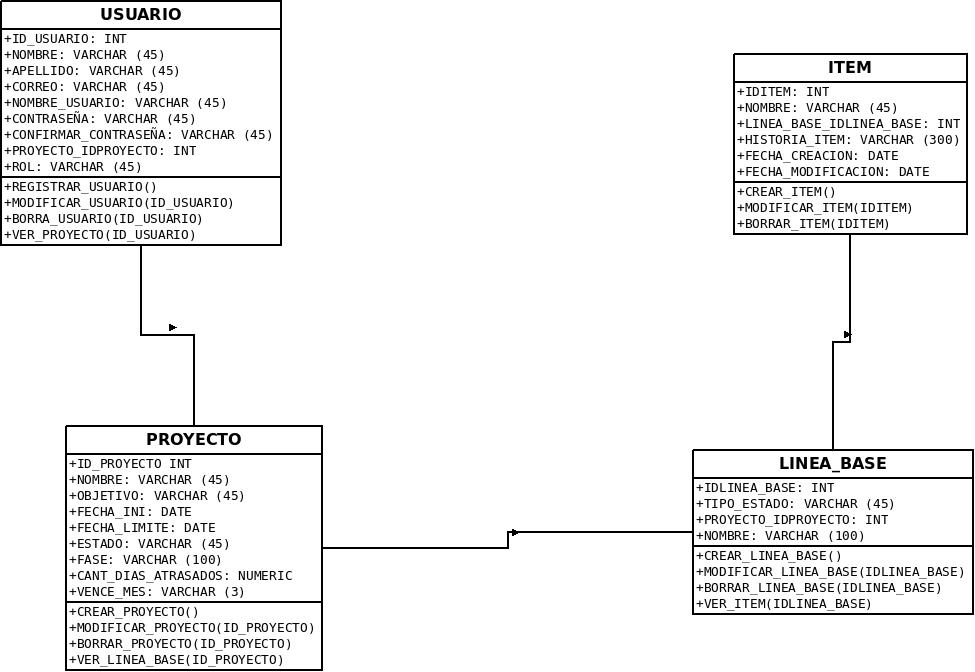
|  |  |
| --- | --- |
| **Vistas** | **UML** |
| Escenarios | Casos de uso |
| Lógica | Clases |
| Procesos | Secuencia |

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

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

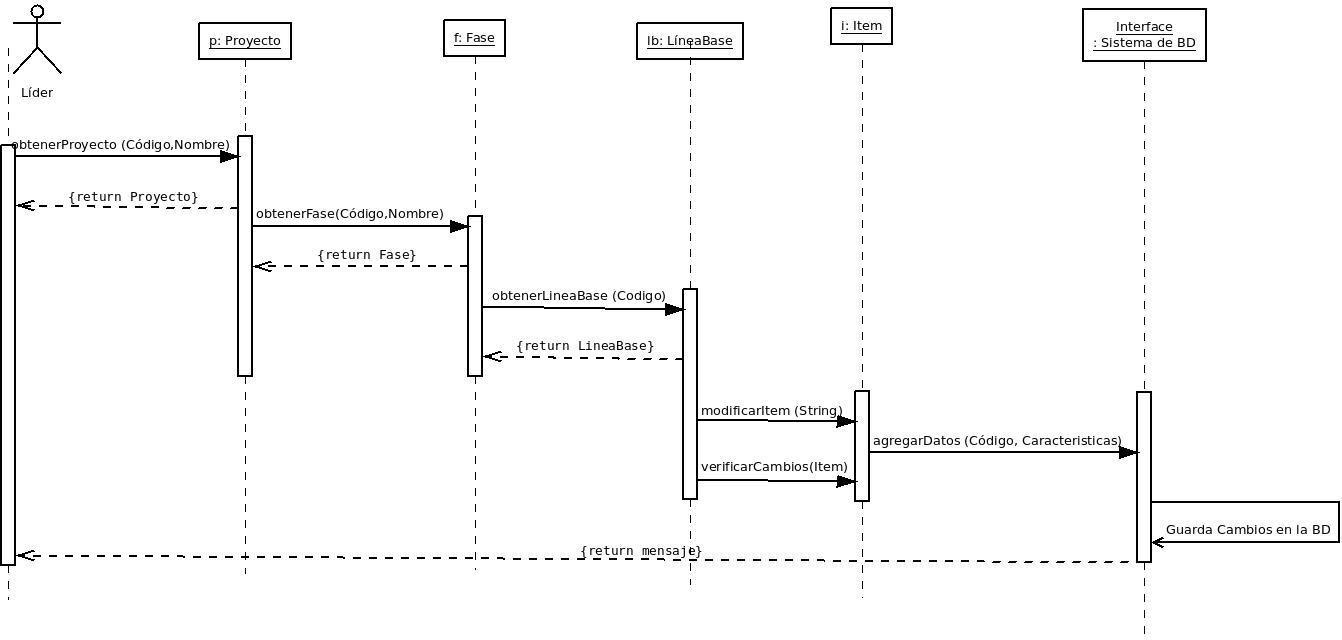


**Vista.- Lógica Diagramas.- Clases**

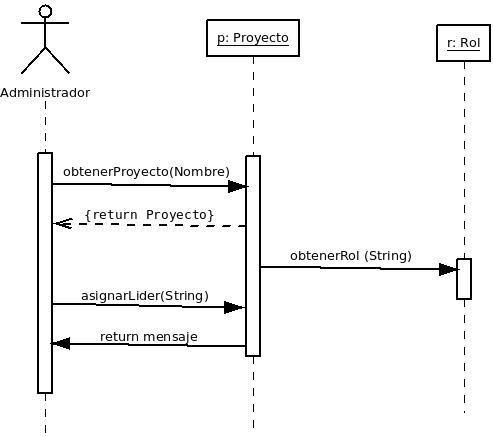


**Vista.- Procesos**

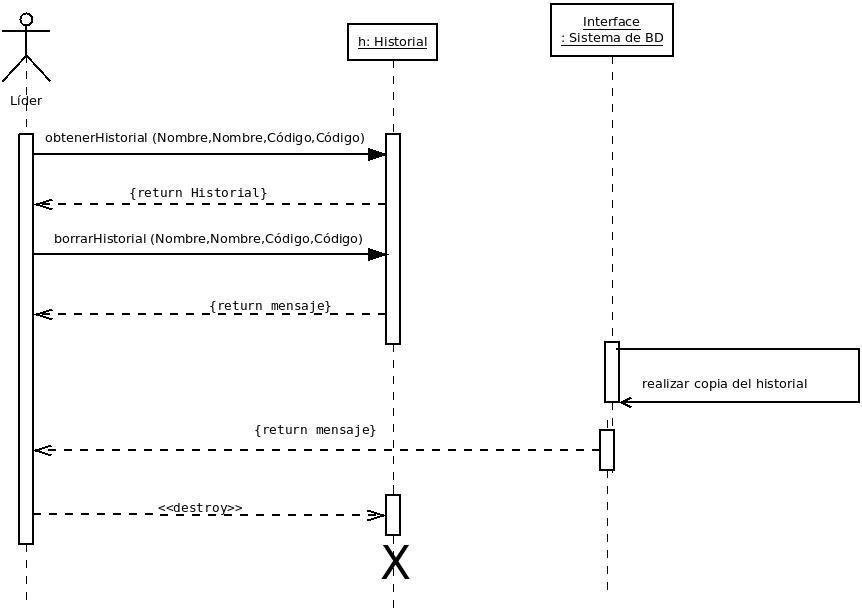
**Diagrama.- Secuencia (Cambio de Ítem)**



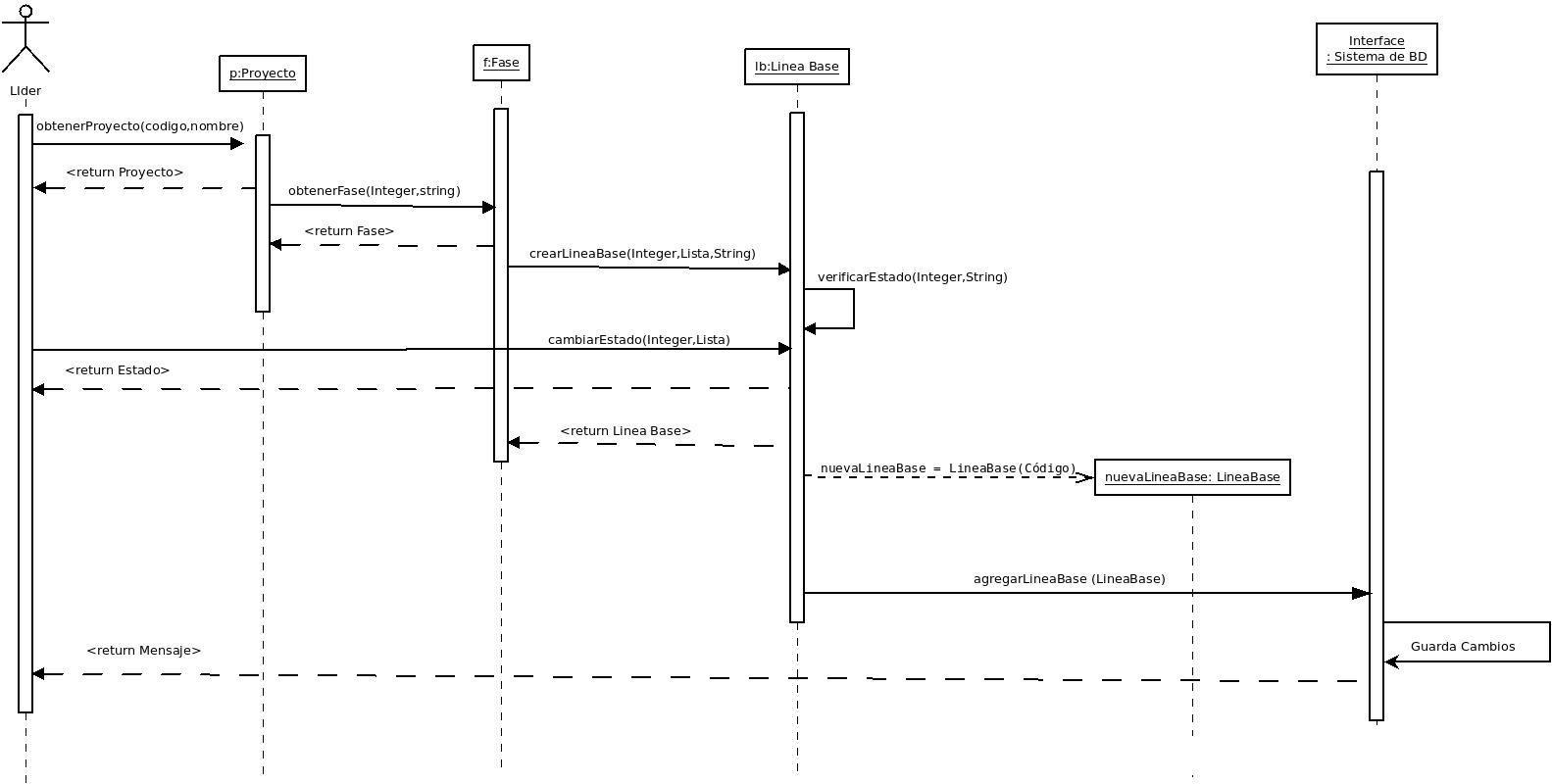
**Diagrama de secuencia (Asignar Líder a Proyecto)**



### Diagrama de secuencia (Borrar Historial)



**Diagrama de secuencia (Generación de Línea Base)**



* 1. **Consistencia en la cantidad de vistas de la arquitectura.**

**DESCRIPCION DE MODULOS**

|  |  |  |
| --- | --- | --- |
| **Nombre del módulo** | **descripción** | **Componentes incluidos** |
| Administración | Módulo que permite la administración de usuarios y roles, además de configuraciones del sistema | * Añadir Usuario * Crear Roles * Asignación de roles |
| Gestión de Configuración | Módulo que permite la generación de líneas bases. Deberá permitir el control de las líneas bases generadas por producto y la administración de los cambios de los elementos que se encuentren en las líneas bases. | * Crear Línea Base * Modificar Línea Base |
| Desarrollo | Módulo que administra todos los elementos de los productos. | * Crear item * Modificar item * Modificar fase |

**DESCRIPCIÓN DE COMPONENTES**

|  |  |  |
| --- | --- | --- |
| **Nombre del componente** | **descripción** | **Componentes relacionados** |
| Ítem | Contiene la lógica para: Crear ítem,  Modificar ítem,  Borrar ítem. | Línea Base |
| Proyecto | Contiene la lógica para: Crear Proyecto,  Modificar Proyecto,  Borrar Proyecto. | Usuario  Línea Base |
| Usuario | Contiene la lógica para:  Crear usuario,  Modificar usuario,  Borrar usuario. | Proyecto |
| Línea Base | Contiene la lógica para: Crear Línea Base, cambiar estado, modificar línea base, borrar línea base. | Proyecto  Ítem |

### Arquitectura lógica.

**Performances**

N/A

**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 **Sistema de Configuración de Software** será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal.

### 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 máquina. 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 alto nivel que será efectuada con lenguajes que es de conocimiento de los integrantes del grupo, lenguaje de programación python, framework Django y una Base de datos PostgreSQL.

Y eso va a depender directamente de las características de los desarrolladores, capacidad de aprendizaje, y en muchos casos opciones propias del proyecto para la cual se efectúa el diseño.