

# Documento de Arquitectura de Software

## IEEE-1471-2000

### Control del documento

---

#### Proyecto

Gestión de Proyectos

#### Título

*Arquitectura del Sistema – [v1.0 al 30 de Agosto del 2019]*

#### Generado por

Estudiante de Informática - [Luz Marina Espinola Ozuna]

Estudiante de Informática - [Viviana Sánchez Britez]

Estudiante de Informática - [Iván Fretes]

Estudiante de Informática - [Martín Flecha]

Estudiante de Informática - [Mónica González]

#### Aprobado por

# 1. Introducción

## 1.1 Propósito

La documentación de este proyecto tiene el propósito de exponer los requerimientos del Sistema de Gestión de Configuración del Software, en él se podrá visualizar requerimientos funcionales, no funcionales en detalle del proyecto, además de las expectativas o necesidades del cliente. Estas especificaciones de requerimientos se construyeron en base a planteamientos entre los desarrolladores y los usuarios del sistema.

## 1.2 Alcance

El documento de Especificación de Requerimientos de Software (ERS), es de gran importancia debido a que el mismo será utilizado como asesoramiento o guía tanto para los usuarios, como para los clientes que utilizaran el sistema, a su vez facilitará la comprensión entre los miembros desarrolladores para la implementación y utilización del sistema.

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 GESTIÓN DE PROYECTOS y el modelo de diseño puesto e implementado en Rational Rose Versión

7.0. La mayoría de las secciones ha sido extraída del Modelo de Rational Rose Version 7.0 y la utilización de plantillas de referencia de ATAM (*Architecture Tradeoff Analysis Method*) y del modelo 4+1 de Kruchten.

### **1.3 Usuarios Interesados**

Administrador del Sistema

Administración de Roles y Usuarios, configuración del sistema.

Lider de Proyectos

Administra los proyectos que se le asigna, crea fases, crea y gestiona líneas base de cada fase, cambio de estado de fases y líneas base.

Desarrollador

Crea items/ requerimientos en las líneas base y modifica los mismos si es requerido previa aprobación del líder

Tester

Verificación de los proyectos y aprobación de los mismos

### **1.4 Recomendaciones de conformidad con esta práctica.**

N/A.

## **2. Referencias**

Las referencias aplicables a este documento son:

Archivo de Documento

IEEE. 1471

Título del Documento

Especificación de Requisitos según el estándar de IEEE 1471

Organización que lo Publica

ANSI/IEEE

### 3. Definiciones, acrónimos y abreviaciones.

DAS: Documento de Arquitectura de Software

ERS: Especificación de Requisitos Software (ERS, SRS): Documento formal de los Requisitos del Sistema

- RF: Requerimiento Funcional.
- RNF: Requerimiento No Funcional.
- Usuario: aquellas personas que accederán con un nombre de usuario una contraseña o clave que le permitirá el ingreso al sistema . En consecuencia de acuerdo al permiso que este posea, gozará de los privilegios que el sistema le será otorgado.
- Rol: Es la función o papel que desempeña una persona dentro del proyecto.

Un rol es un conjunto de permisos definidos en el sistema.

1. Líder del Proyecto

2. Analista Desarrollador

3. Administrador del Sistema

4. Tester

- Estados: momento temporal a la que cada fase está sujeta debido a que la misma puede presentar cambios conforme a lo que el proyecto va avanzando. Es utilizada para dar un informe sobre que se desarrolla durante el proceso del proyecto.
- Línea base: en ellas se realizarán los requerimientos para tener el control de cambios que serán efectuados durante las fase.
- Ítems: son los atributos de definición.
- Fases: las mismas se conformarán de 4 estados.
  - Pendiente: cuando es creada la fase.
  - Activa: mientras se está trabajando sobre cada ítems
  - Cerrada: cuando los ítems están finalizados.

- Bloqueado: cuando el proceso queda pausa o más bien bloqueado debido a algún cambio realizado en otro estado.

Las fases no son simultáneas, pueden haber varias líneas Bases en una base, puede haber solo un requerimiento y este puede no sufrir modificaciones.

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.

Usuarios: 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.

## **4. Framework Conceptual**

### **4.1 Descripción de la arquitectura en contexto**

### **4.2 Stakeholders y sus roles**

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

### **4.3 Actividades de arquitectura en el ciclo de vida**

N/A.

#### 4.4 Usos de las descripciones de arquitectura

Las descripciones de arquitectura de este documento se usarán para referenciar el diseño del sistema de software del Proyecto.




#### 5. Descripciones prácticas de arquitectura

N/A.

##### 5.1 Documentación de la arquitectura

N/A.

##### 5.2 Identificación de los Usuarios y sus responsabilidades

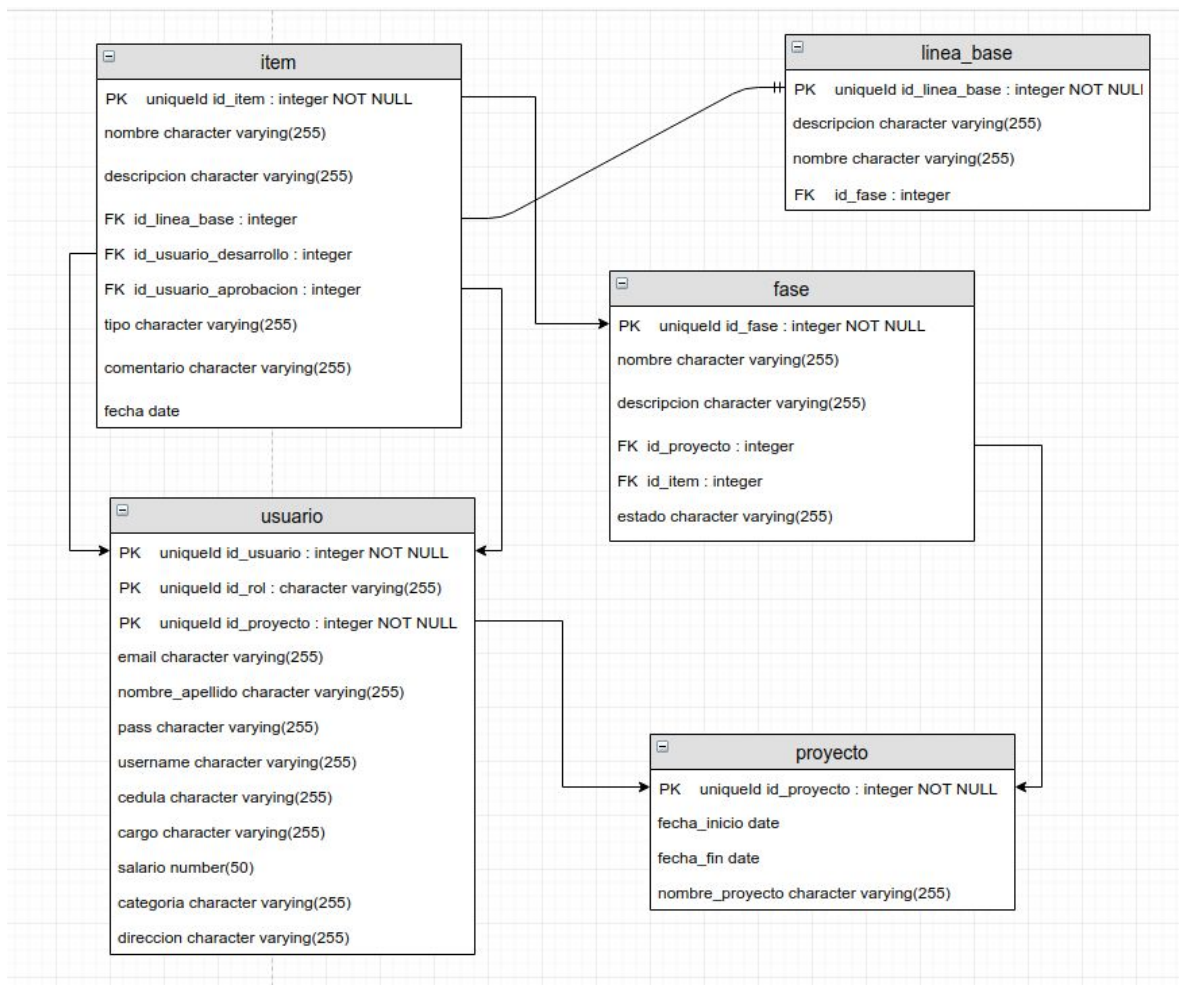
Stakeholder	descripción	escenario	Vistas
 Administrador	Es el usuario encargado de administrar los roles y usuarios.	Administración de Roles.  Administración de Usuarios.	Crear usuario Loguearse al sistema Modificar contraseña  Delegar funciones por rol Asignar un rol al usuario creado
 Líder de Proyecto	Es la persona que administra los proyectos que le son asignados	crear , gestionar y cambiar de estado las fases y líneas de base e Items.	Crear proyectos Definir los roles del proyecto Otorgar estados a cada proyecto Emitir reportes Crear y Cambiar los estados del ítem
 Desarrollador	Es la persona encargada de la modificación de la descripción del ítem	modificar y cambiar estados de ítems mientras el estado del ítem esté en pendiente	Modificar la descripción del ítem Revertir Item Solicitar desbloqueo de línea base Consultar historial

### 5.3 Selección de puntos de vista de la arquitectura.

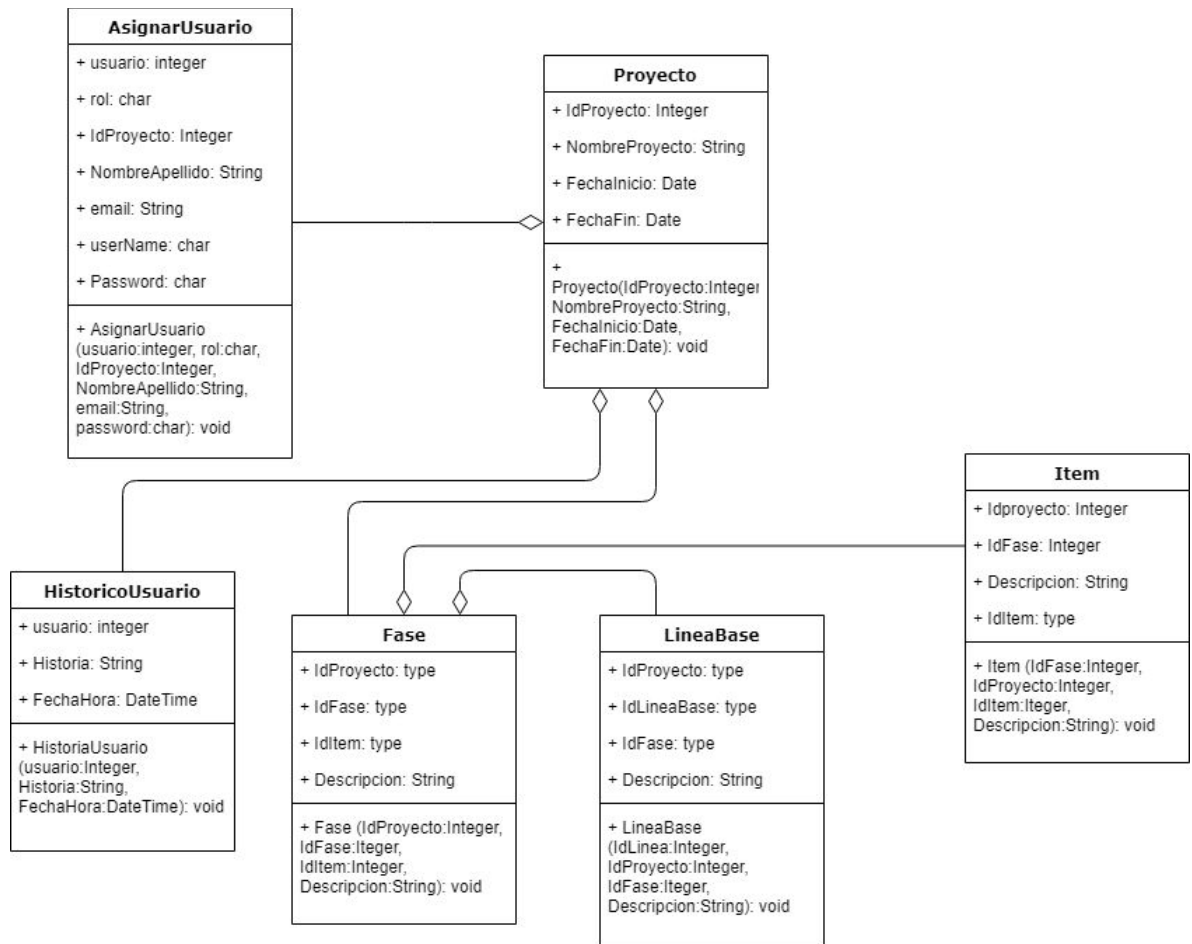
Vistas	UML
Lógica	DER
Escenarios	Casos de uso
Procesos	Secuencia
Desarrollo	Componentes

### 5.4 Vistas de la arquitectura

#### - Diagrama de Entidad Relación

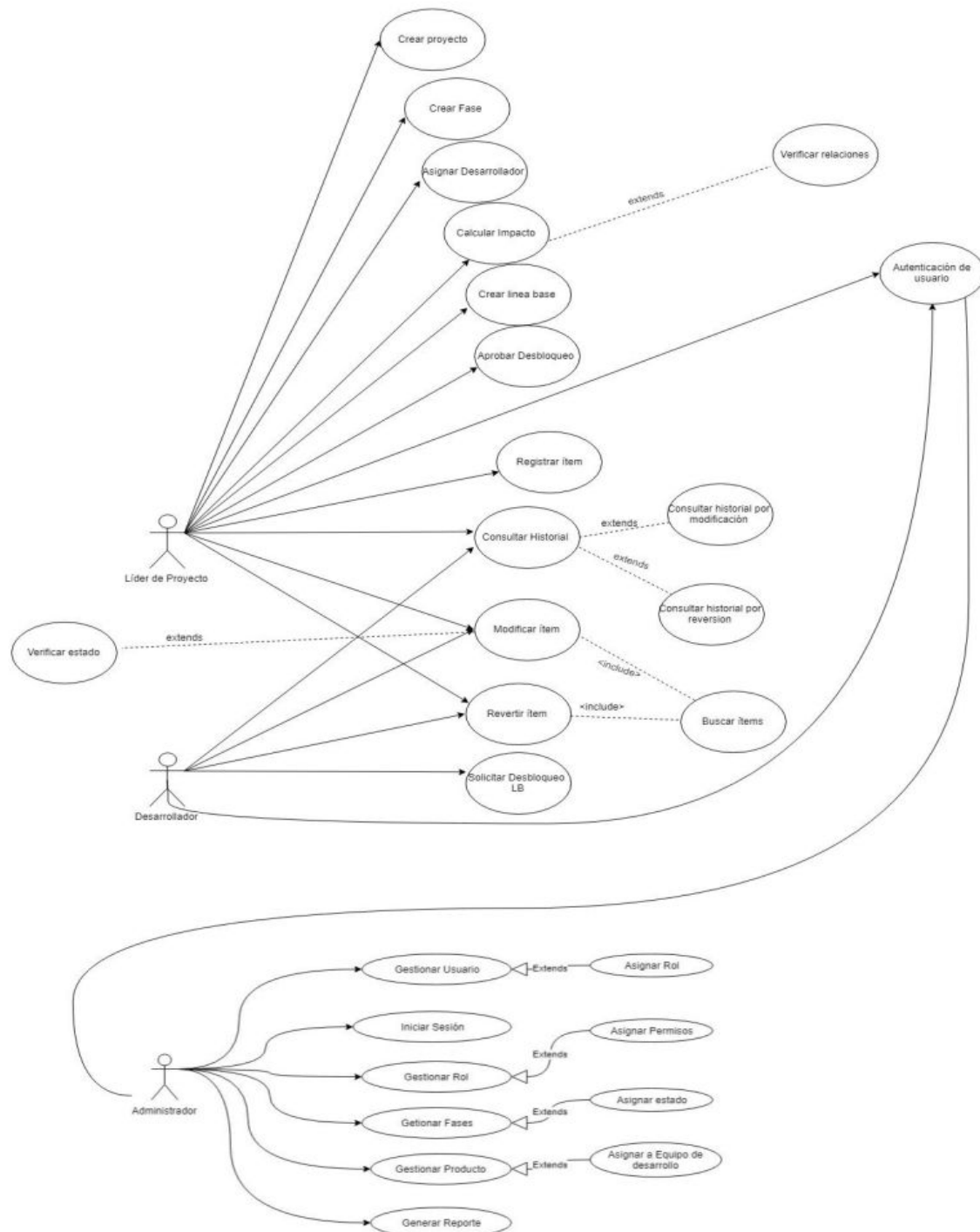


## - Diagrama de classes



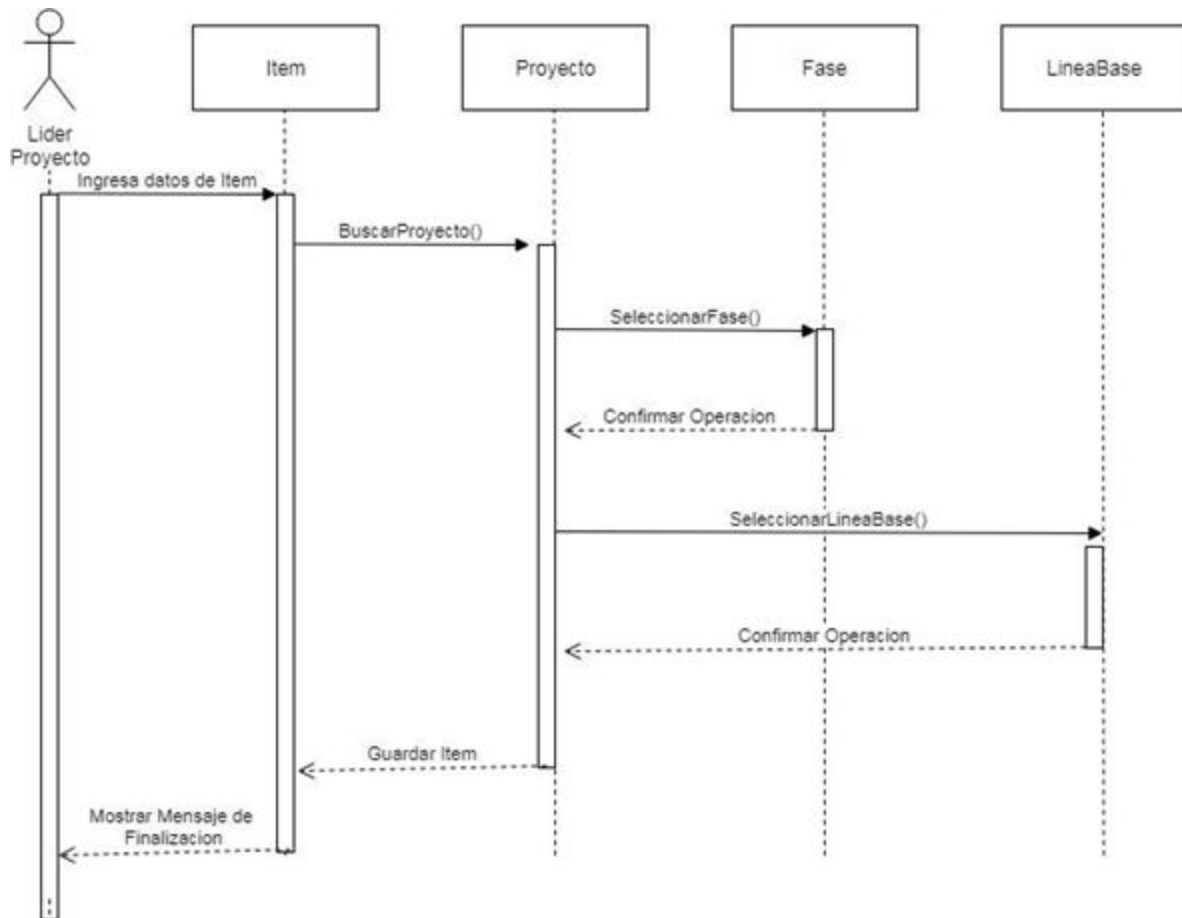


## - Caso de uso del negocio



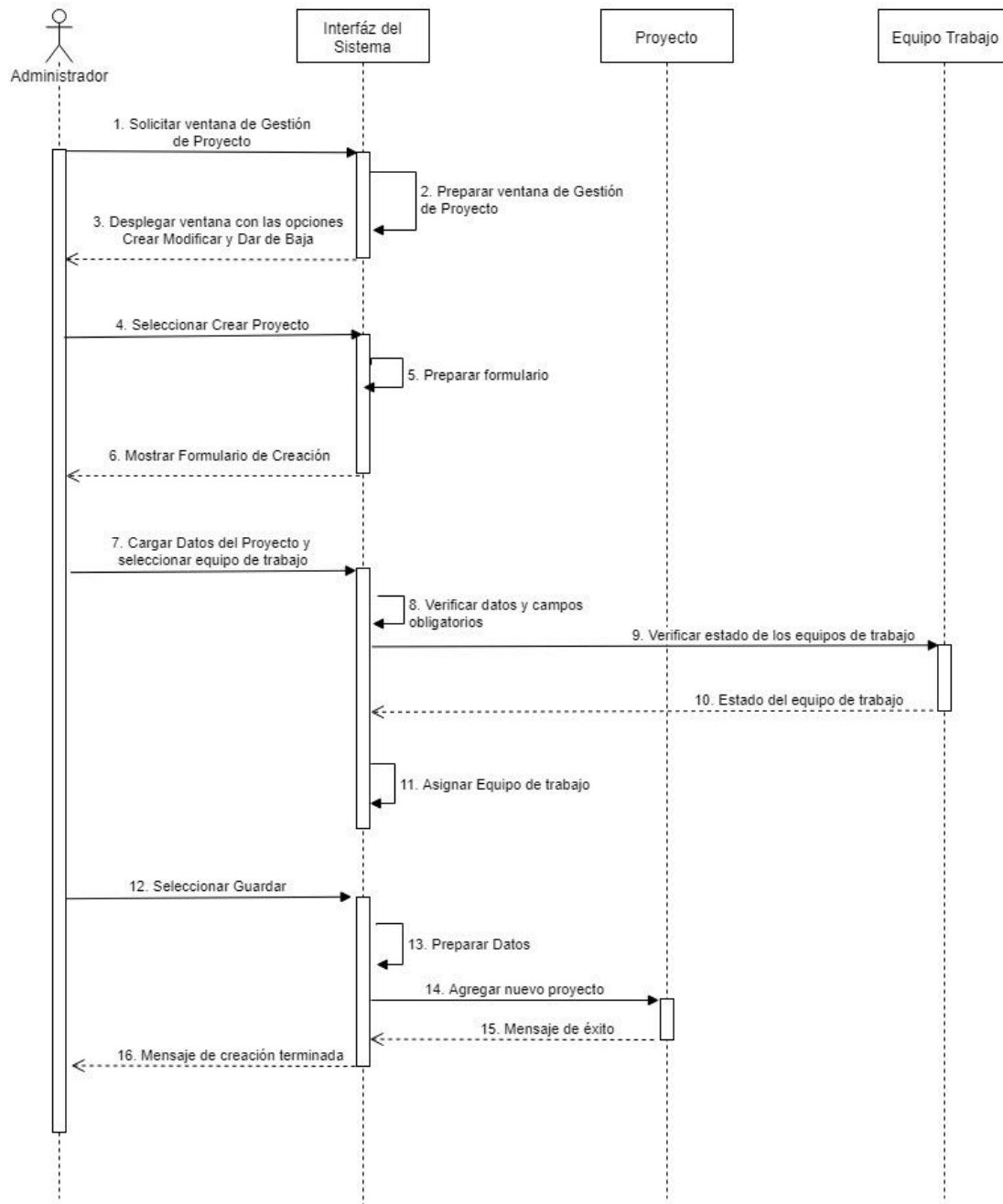
## Diagrama de secuencia

**CREAR ITEM** (Ingresar Datos: Id (Automático), Fecha-Hora de carga (Automático), Usuario que carga (Automático), Nombre de ítems-requerimiento, Descripción del ítem-requerimiento, Id Línea Base relacionado, Estado Pendiente por defecto).



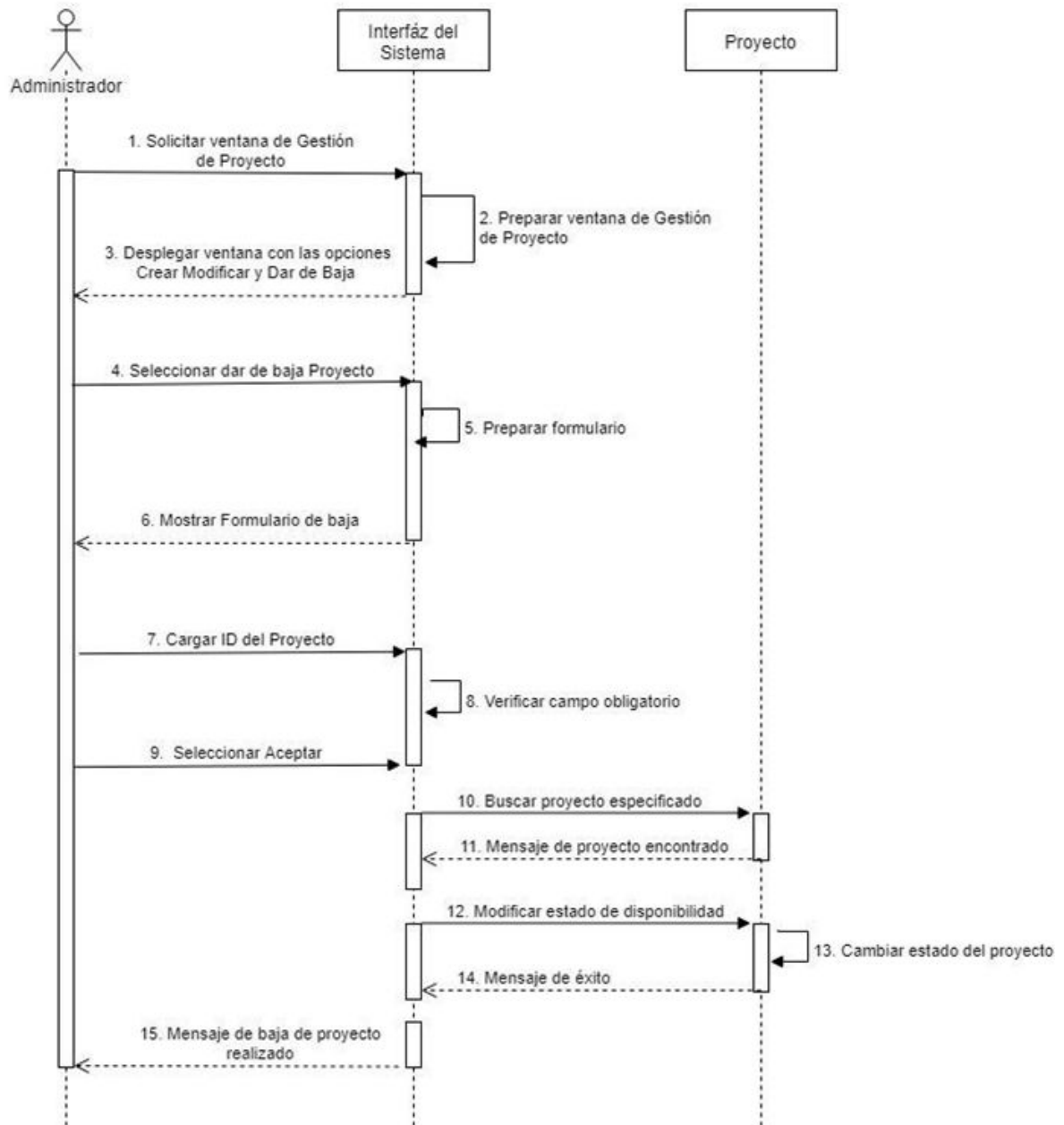
## CREAR PROYECTO

Gestionar Proyecto

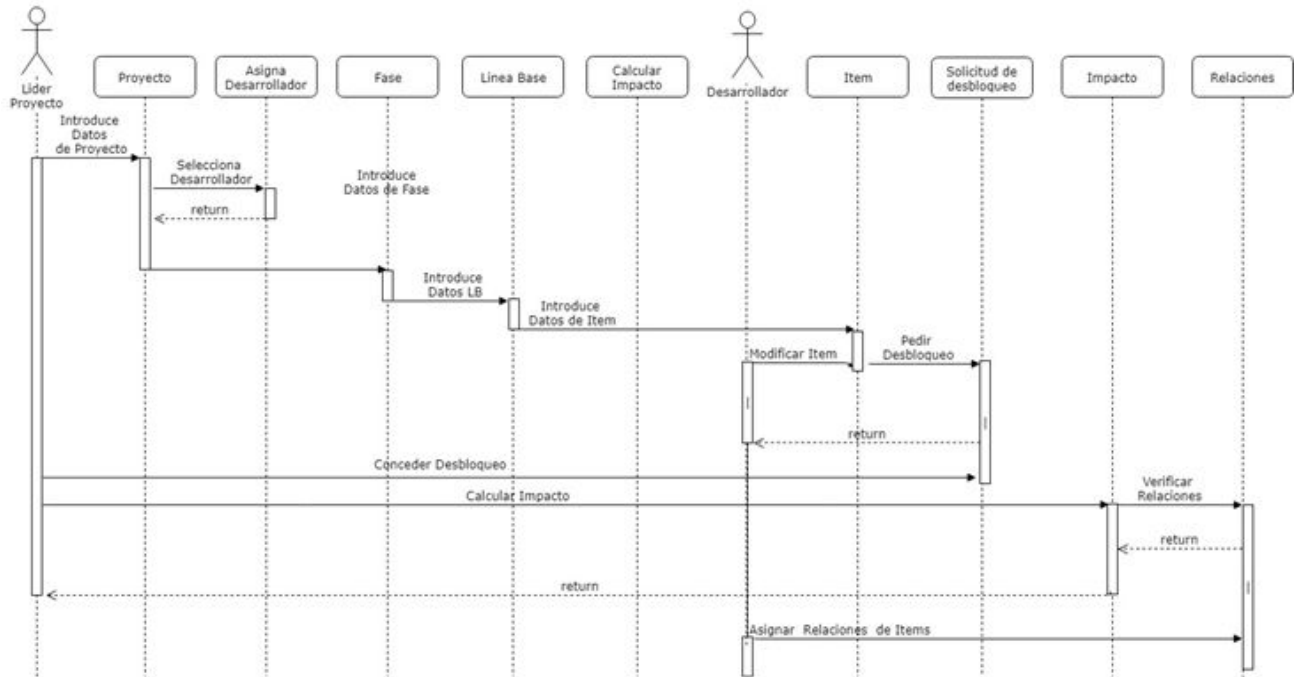


## ELIMINAR PROYECTO

Gestionar Proyecto



GESTION DE CONFIGURACION



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

### DESCRIPCION DE MODULOS

Nombre del módulo			descripción	Componentes incluidos
			Módulo que administra los roles y usuarios..	<ul style="list-style-type: none"> <li>● CrearUsuario</li> <li>● AutenticarUsuario</li> <li>● ModificaContraseña</li> <li>● AsignarRol</li> <li>● ValidarUsuario</li> <li>● ValidarAcceso</li> <li>● HabilitarFase</li> <li>● HabilitarRol</li> <li>● DelegarFunciones</li> </ul>
	Administrador			
			Modulo que gestiona los proyectos asignados (estados, fases, lineas base).	<ul style="list-style-type: none"> <li>● CrearProyecto</li> <li>● GestionarProyecto</li> <li>● CambiarEstadoFase</li> <li>● CambiarEstadoLinea</li> <li>● DefinirRoles</li> </ul>
	Gestión			
			Contiene los informes generados en base a los proyectos y tareas en proceso y finalizados.	<ul style="list-style-type: none"> <li>● EmitirReporte</li> <li>● FiltroBusquedaReporte</li> <li>● DetalleReporte</li> <li>● EditarReporte</li> </ul>
	Reportes			

## DESCRIPCIÓN DE CONECTORES

### 5.6 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 usuarios simultáneos contra la base de datos central en cualquier tiempo 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 Framework que utilizaremos pertenece al modelo de Vista Controlador.

### 5.7 Ejemplo de uso.

N/A.

### 5.8 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**

#### Herramientas

- Servidor de Aplicaciones
  - nginx
- Servidor de Base de datos
  - PostgreSQL
- Backend (Servidor)
  - Lenguaje de programación (python)
  - Framework (flask / django)
- Frontend (Cliente)
  - HTML
  - CSS3
  - Javascript
  - Librerías
    - Bootstrap 4
    - jQuery