

ADSISTEM

Manual Técnico

Versión: 0100

Fecha: 17/09/2016

CONTENIDO

[1. DOCUMENTO ARQUITECTURA DE SOFTWARE 3](#_Toc462062631)

[1.1 INTRODUCCIÓN 3](#_Toc462062632)

[2. IDENTIFICACIÓN DE STAKEHOLDERS Y SUS RESPONSABILIDADES 4](#_Toc462062633)

[2.1 STAKEHOLDERS (CLIENTES, PROVEEDORES, VENDEDORES) 4](#_Toc462062634)

[2.2 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 4](#_Toc462062635)

[2.3 REUNIÓN DE REVISIÓN DEL SPRINT 4](#_Toc462062636)

[3. VISTAS DE LA ARQUITECTURA 5](#_Toc462062637)

[4. DIAGRAMA CASOS DE SUSO 6](#_Toc462062638)

[5. DIAGRAMAS DE CLASES 7](#_Toc462062639)

[6. DIAGRAMAS DE SECUENCIA 8](#_Toc462062640)

[7. DIAGRAMA DE PAQUETES 8](#_Toc462062641)

[8. DIAGRAMA DE COMPONENTES 8](#_Toc462062642)

[9. PLANTEAMIENTO DEL PROBLEMA 8](#_Toc462062643)

[10. ARQUITECTURA 8](#_Toc462062644)

[11. MODELO ENTIDAD RELACIÓN 8](#_Toc462062645)

# DOCUMENTO ARQUITECTURA DE SOFTWARE

# 1.1 INTRODUCCIÓN

En el siguiente manual se evidencia todas las herramientas y metodologías utilizadas en el desarrollo del proyecto ADSISTEM.

# IDENTIFICACIÓN DE STAKEHOLDERS Y SUS RESPONSABILIDADES

# STAKEHOLDERS (CLIENTES, PROVEEDORES, VENDEDORES)

Son las personas que hacen posible el proyecto y para quienes el proyecto producirán el beneficio acordado que justifica su desarrollo. Sólo participan directamente durante las revisiones del "sprint".

# REUNIÓN DE PLANIFICACIÓN DEL SPRINT

Se realiza una reunión con el dueño del producto en la cual se definen que historias de usuario se irán desarrollando en cada Sprint definiendo la fecha de inicio y la de fecha del cierre del Sprint.

# REUNIÓN DE REVISIÓN DEL SPRINT

Se realiza una reunión con el dueño del producto en la fecha definida del cierre el Sprint, en la cual el dueño hace revisión general de que las historias de usuario se desarrollaron correctamente si a lo largo de esta revisión se presenta posibles mejoras el dueño asignara peticiones indicando que se debe volver a desarrollar.

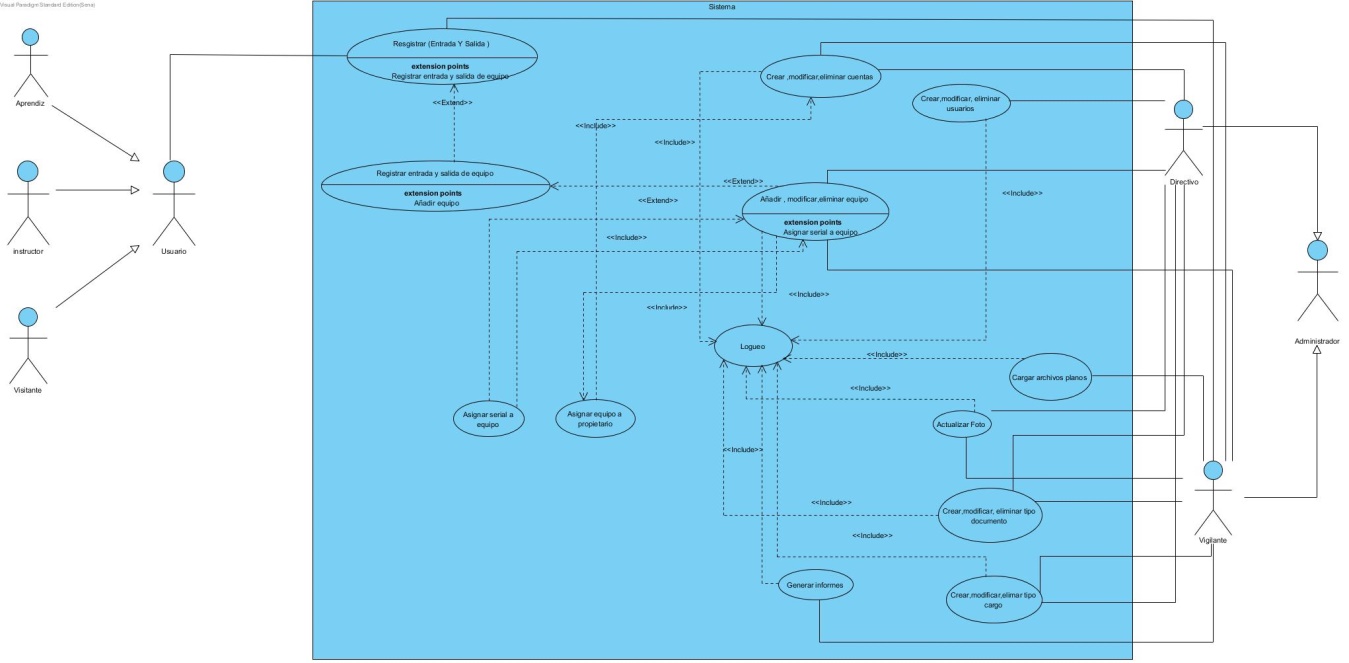
# VISTAS DE LA ARQUITECTURA

# VISTA DE INTEGRACION En esta vista se encuentra todo la parte de integración es decir las clases asociadas a la base de datos ADSISTEM.

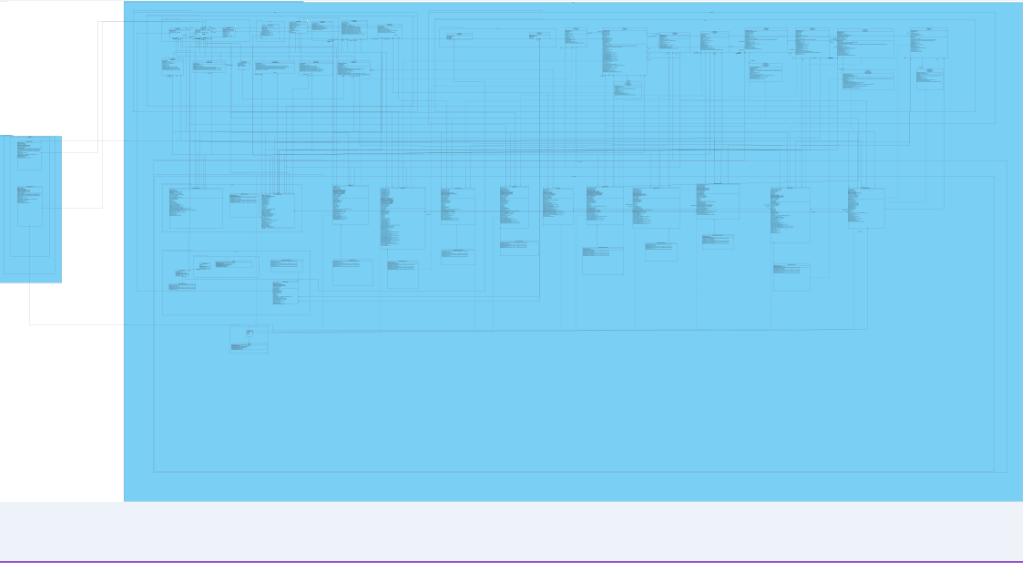
# VISTA DE NEGOCIO] En esta vista se encuentra toda la parte lógica del negocio es decir métodos que permiten el funcionamiento del Software.

# VISTA DE PRESENTACIÓN En esta vista se encuentra toda la parte de presentación es decir que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

# DIAGRAMA CASOS DE SUSO



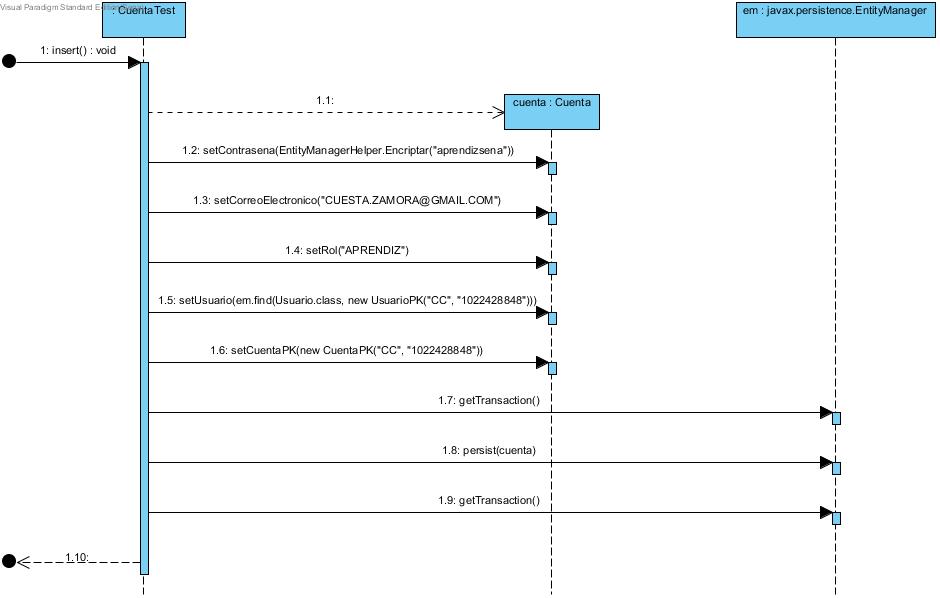
# DIAGRAMAS DE CLASES



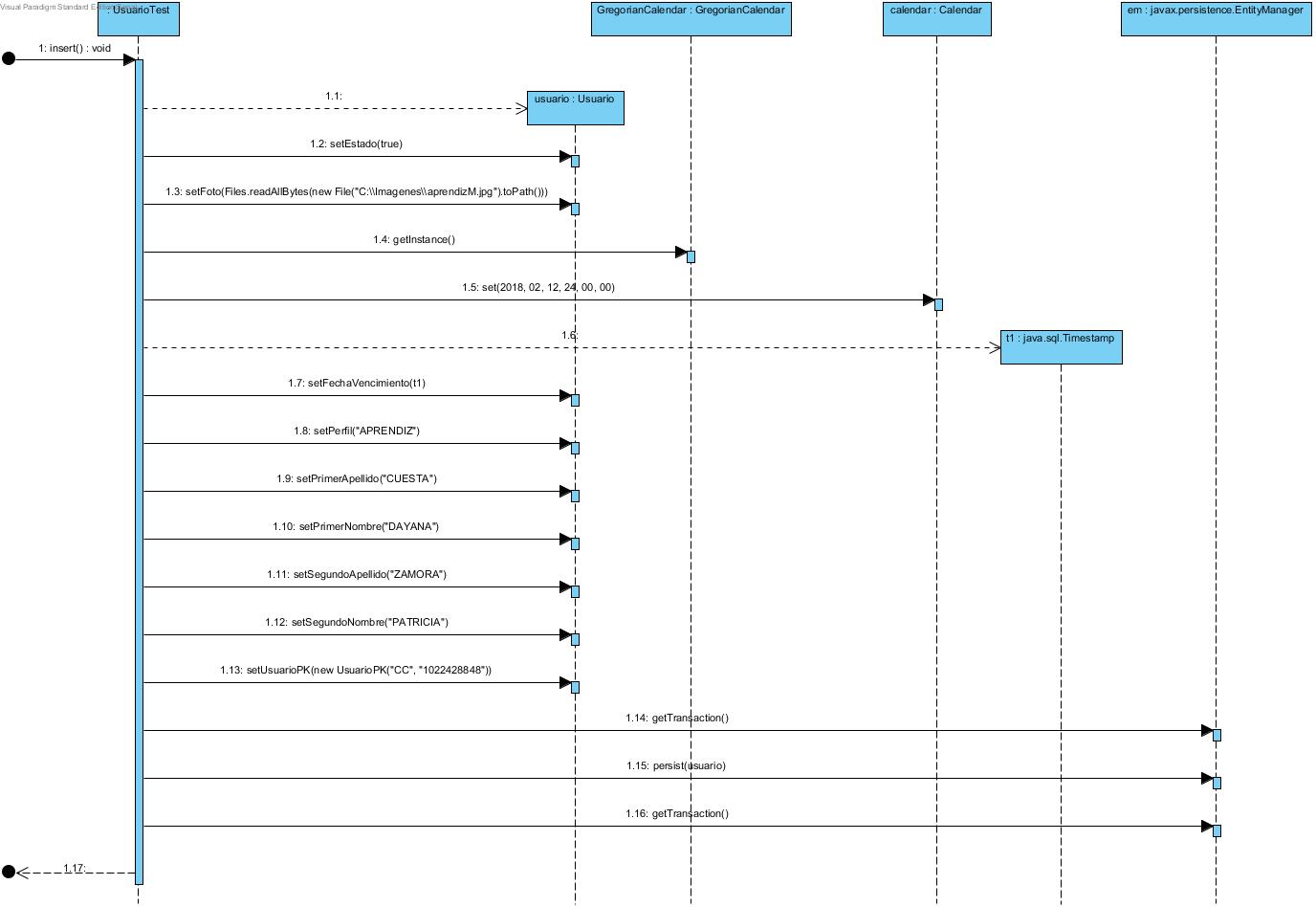
# DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia se manejaron en la versión 3 del proyecto con las clases:

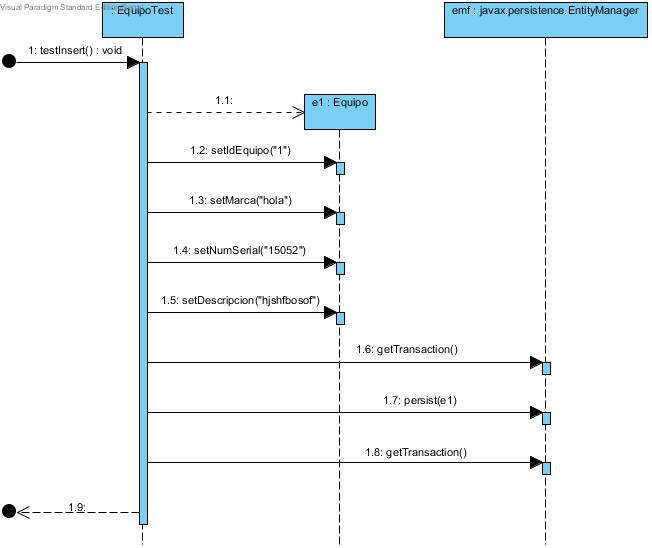
Cuenta



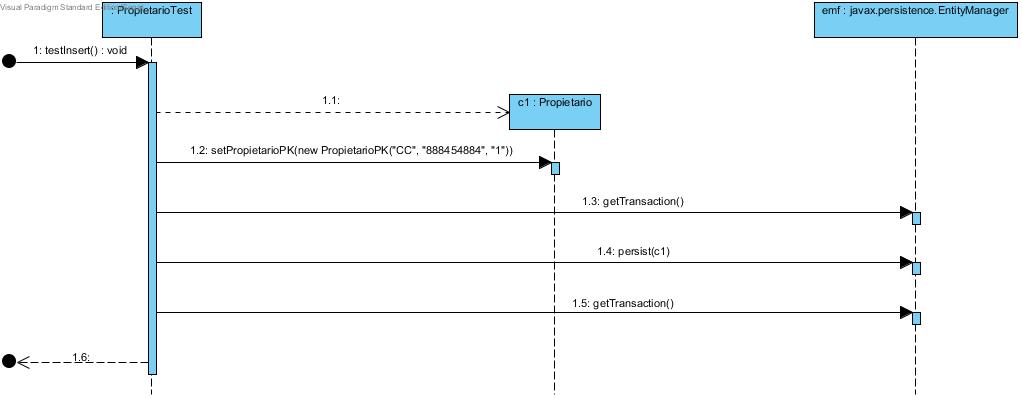
Usuario



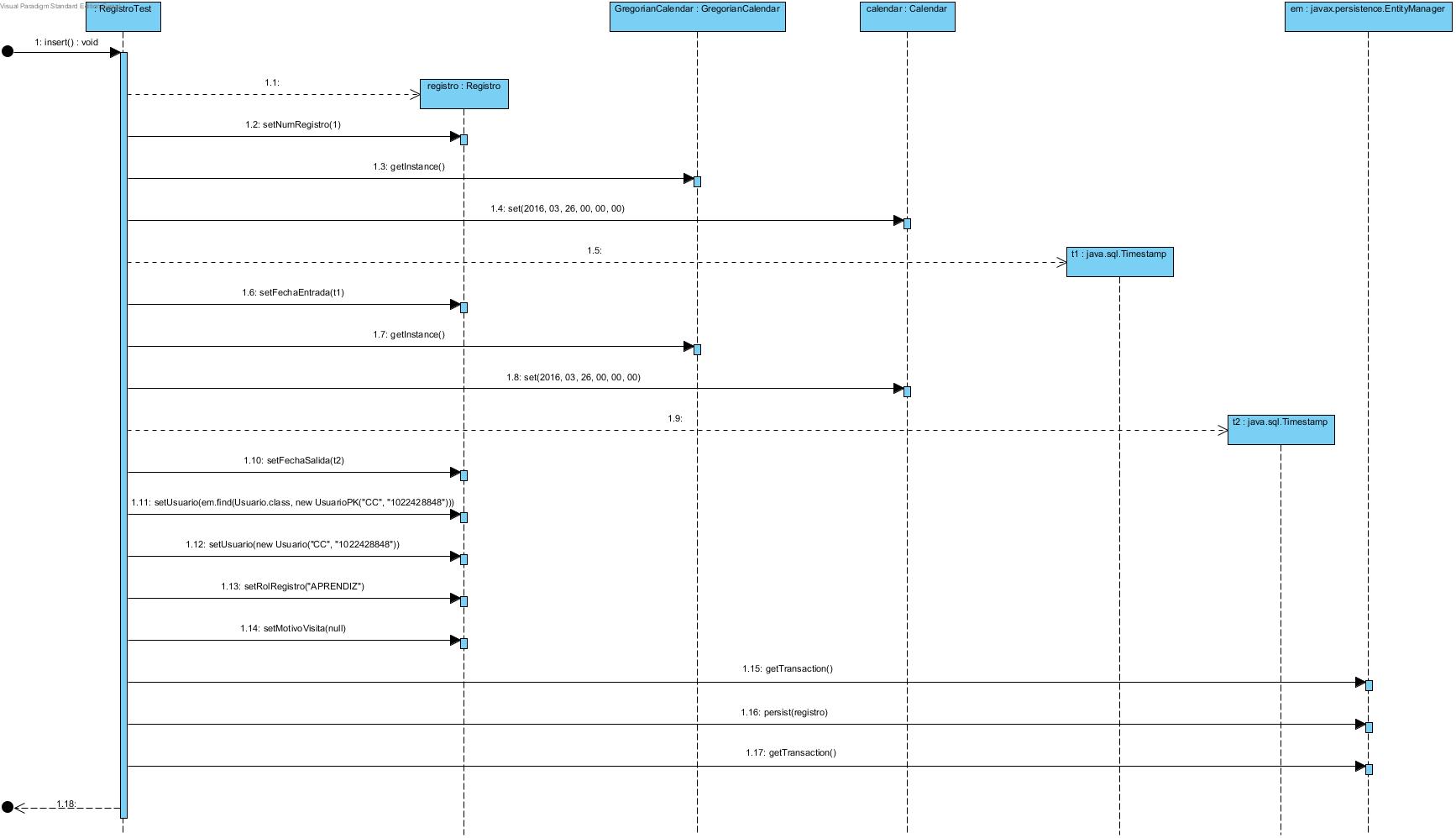
Equipo



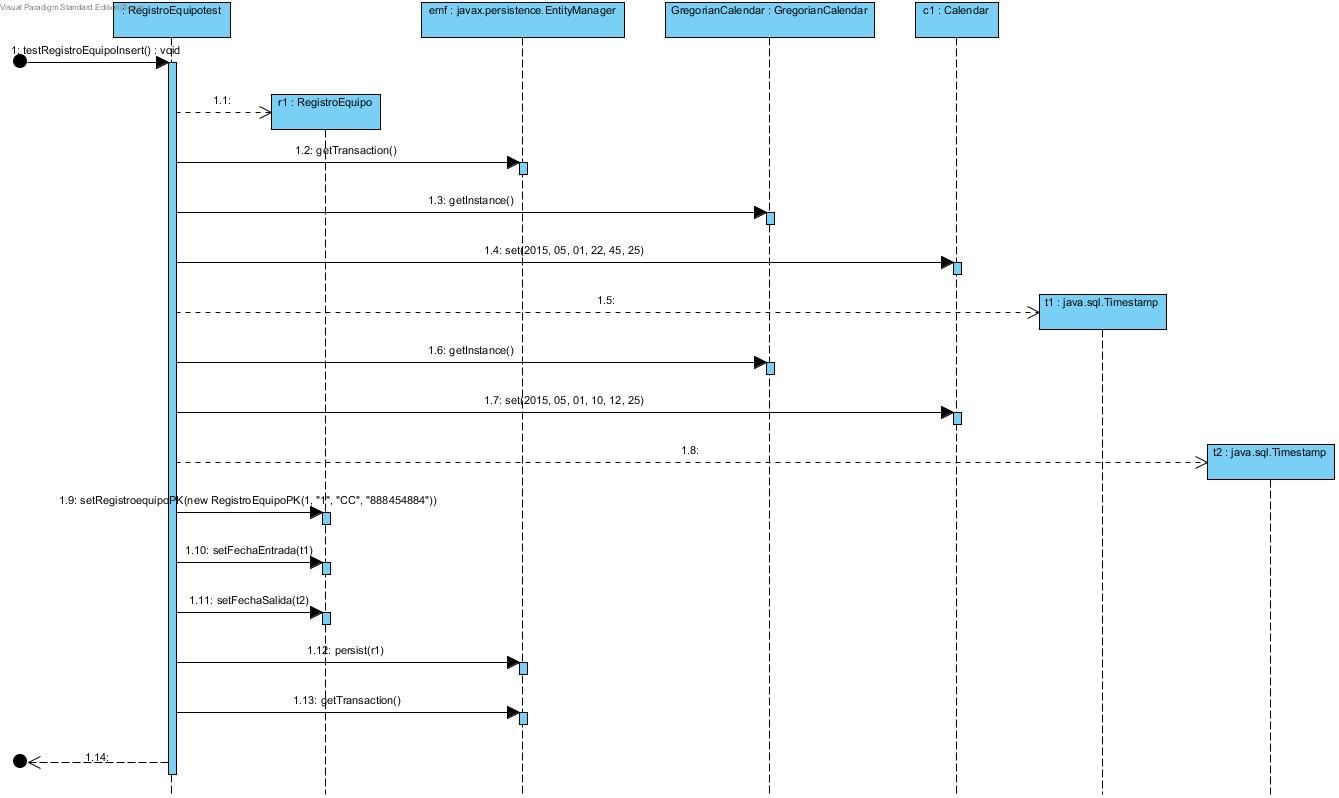
Propietario



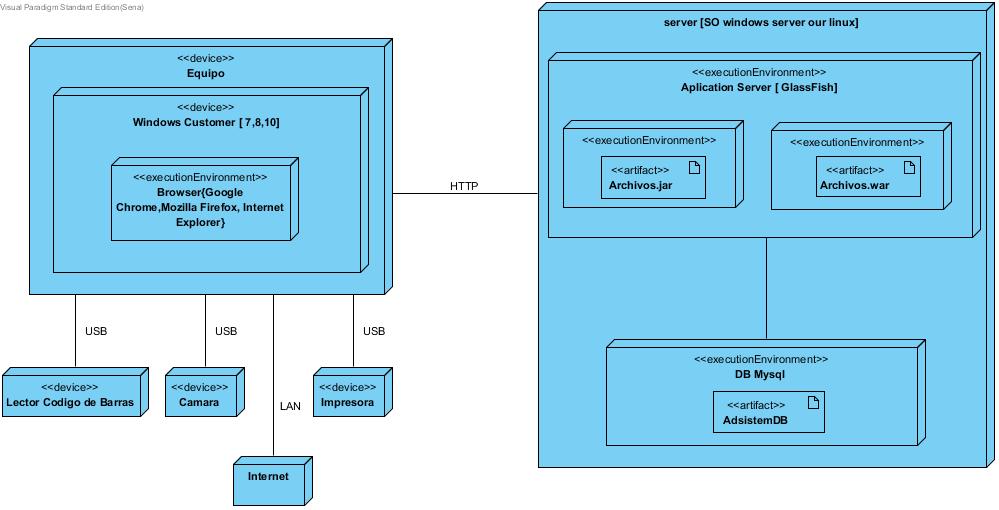
Registro



Registro equipo



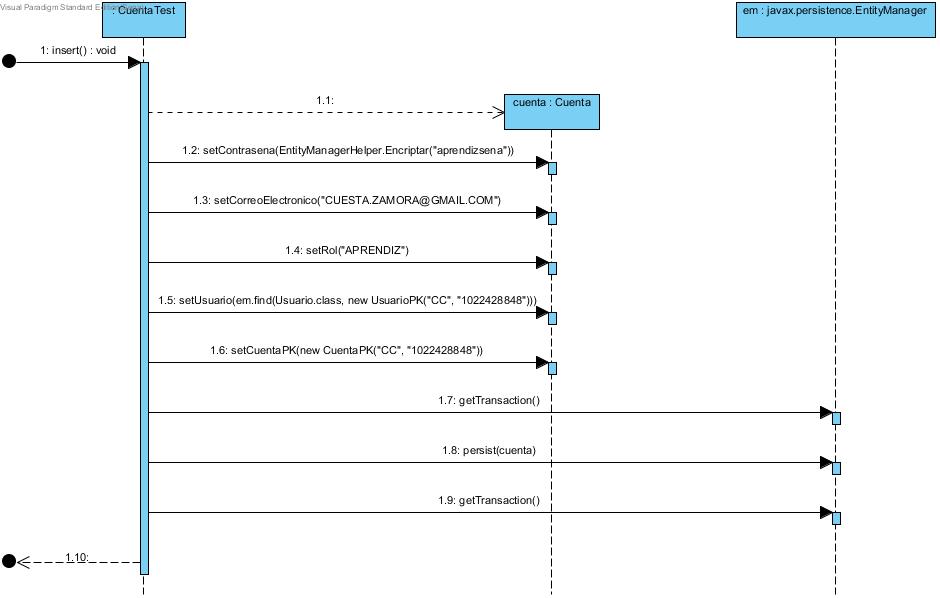
# DIAGRAMA DE DESPLIEGUE



# PLANTEAMIENTO DEL PROBLEMA

En la sede Restrepo se requiere un sistema de información que agilice el registro de la entrada y salida de Usuarios quienes cuentan con datos tales como Tipo de Documento, Numero de Documento, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Tipo de Cargo (con el cual se podrá diferenciar al usuario), fecha de finalización de su formación, un Estado donde se evidenciara si se encuentra activo o inactivo en el sistema y foto del usuario. Los administradores accederán al sistema por medio de un Usuario que contara con un Tipo de Documento, Numero de Documento, Rol (Vigilante, Directivo), Contraseña y un Correo Electrónico (en caso de que olvide su contraseña y puede recuperar el usuario por medio del correo electrónico), si un usuario es Propietario de un Equipo lo podrá realizar el Registro del Equipo por medio número de serial, marca y una breve descripción del mismo.

# ARQUITECTURA

Arquitectura de n capas.

Es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado

# MODELO ENTIDAD RELACIÓN

