Elaboración y Gestión de Actas

Entregable 1 – Desarrollo de Software Basado en Componentes

**Andrés Orlando López Henao**

**Jhon Harvey Medina Acevedo**

**Jhonnatan Alexander Martínez Rojas**

**Rafael Posada Fernández**

Universidad EAFIT, 2018

# Modelo Verbal

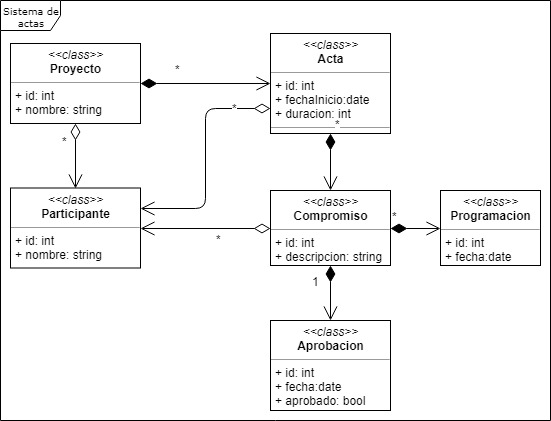
Aplicación para la elaboración y gestión de actas de las reuniones llevadas a cabo dentro de un proyecto, con su correspondiente seguimiento de compromisos, reprogramaciones, informes de estado, aprobaciones y versiones con ajustes (trazabilidad).

# Requisitos Funcionales

Se adjunta el Product Backlog como archivo de Excel con las historias de usuario y sus respectivos criterios de aceptación, las dos primeras historias conforman el M.P.V. (mínimo producto viable).

# Diagrama de Clases

## Diagrama



## Descripción

**Clases de la aplicación:**

**Proyecto (Clase principal):** Clase para administrar los diferentes proyectos que se gestionaran dentro de la app.

**Participante:** Clase para gestionar los miembros de los diferentes proyectos

**Acta:** Clase para almacenar el/las acta(s) de reuniones por proyecto

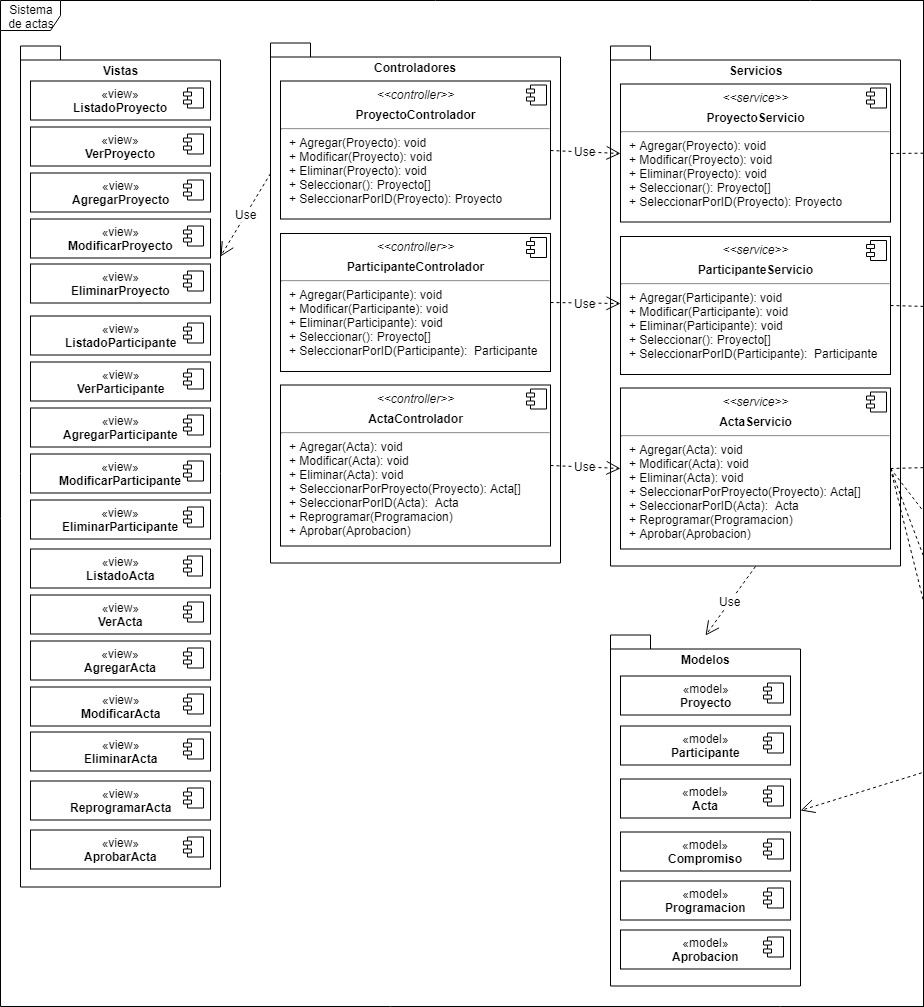
**Compromiso:** Clase para almacenar los compromisos que se establecen en las actas del proyecto.

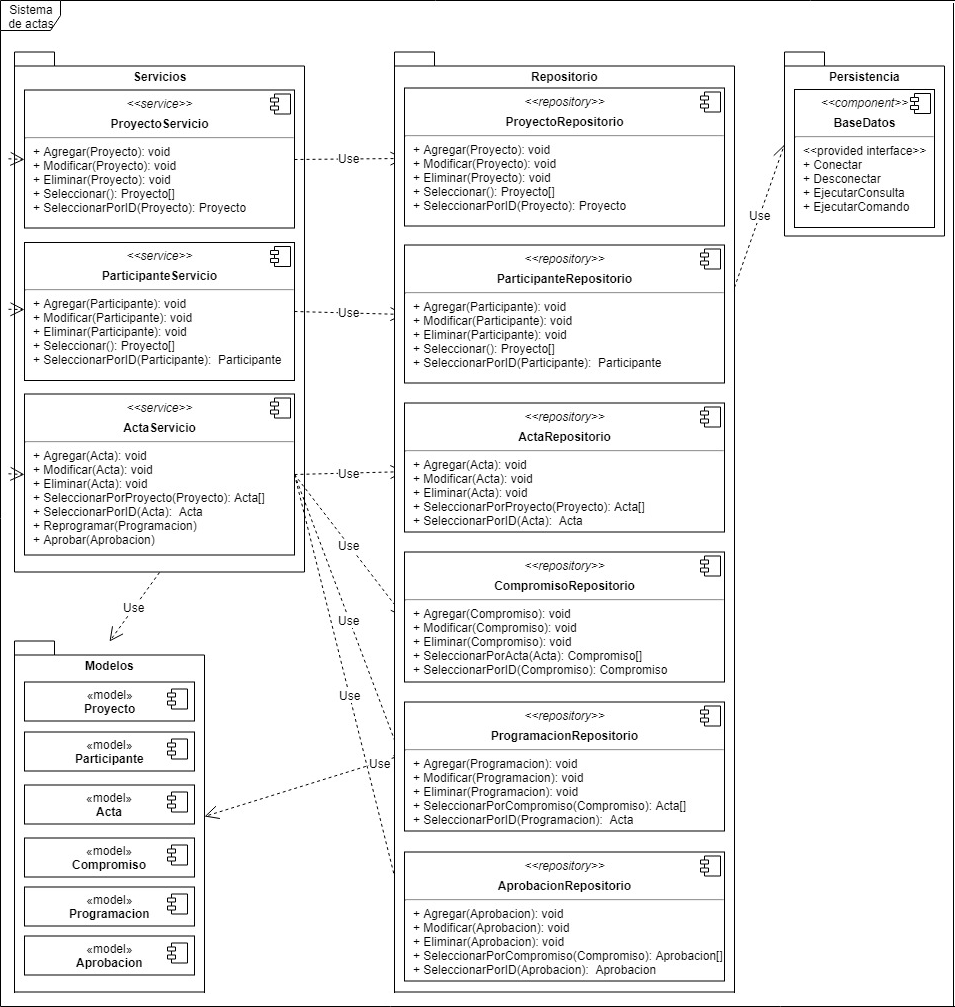
**Aprobación:** Clase para almacenar el consentimiento de los compromisos.

**Programación:** Clase para almacenar las diferentes programaciones que se realizan a los compromisos establecidos.

# Diagrama de Componentes UML

## Diagrama





## Descripción

**Listado Proyecto**: Vista que permite visualizar todos los proyectos disponibles en el sistema para facilitar su administración.

**Ver Proyecto:** Permite la visualización individual de cada proyecto para ver a detalle los elementos que lo componen.

**Agregar Proyecto:** Vista que permite la creación de un nuevo proyecto que posterior mente será gestionada.

**Modificar Proyecto:** Vista que permite la edición de un proyecto en específico.

**Eliminar Proyecto:** Vista que permite eliminar un proyecto para que no sea posible su manipulación desde el sistema.

**Listado Participante:** Vista que permite ver la visualización de todos los participantes de cada proyecto.

**Ver Participante:** Vista que permite ver a detalle los participantes.

**Agregar Participante:** Vista que permite agregar nuevos participantes.

**Modificar Participante:** Vista que permite modificar los participantes de forma individual.

**Eliminar Participante:** vista que permite eliminar a un participante.

**Listado Acta:** Vista que permite ver la visualización de todas las actas de cada proyecto.

**Ver Acta:** Vista que permite ver a detalle las Actas.

**Agregar Acta:** Vista que permite agregar nuevas actas.

**Modificar Actas:** Vista que permite modificar las actas de forma individual.

**Eliminar Acta:** vista que permite eliminar un acta.

**Reprogramar Acta:** Vista que permite la reprogramación de las actas.

**Aprobar Acta:** Vista que permite la aprobación de las actas.

**Proyecto Controlador:** Controller que contiene la lógica (métodos y funciones) de las vistas relacionadas con el Proyecto para su correcto despliegue.

**Proyecto Participante:** Controller que contiene la lógica (métodos y funciones) de las vistas relacionadas con el Participante para su correcto despliegue.

**Proyecto Acta:** Controller que contiene la lógica (métodos y funciones) de las vistas relacionadas con el Acta para su correcto despliegue.

**Proyecto Servicio:** Aplica la lógica del negocio correspondiente a los proyectos, permite el control del flujo de información y la comunicación entre la base de datos y la clase controladora, permitiendo realizar acciones como persistencia de la entidad Proyecto entre otras (Crud).

**Acta Servicio:** Aplica la lógica del negocio correspondiente a las actas, permite el control del flujo de información y la comunicación entre la base de datos y la clase controladora, permitiendo realizar acciones como persistencia de la entidad Acta entro otras (Crud).

**Participante Servicio:** Aplica la lógica del negocio correspondiente a los participantes, permite el control del flujo de información y la comunicación entre la base de datos y la clase controladora, permitiendo realizar acciones como persistencia de la entidad Participante entro otras (Crud).

**Proyecto:** Modelo que permite la implementación de los atributos y métodos que contiene Proyecto (Pojo) para su manejo dentro de las reglas del negocio del sistema.

**Acta:** Modelo que permite la implementación de los atributos y métodos que contiene Acta (Pojo) para su manejo dentro de las reglas del negocio del sistema.

**Participante:** Modelo que permite la implementación de los atributos y métodos que contiene Participante (Pojo) para su manejo dentro de las reglas del negocio del sistema.

**Compromiso:** Modelo que permite la implementación de los atributos y métodos que contiene Compromiso (Pojo) para su manejo dentro de las reglas del negocio del sistema

**Programación:** Modelo que permite la implementación de los atributos y métodos que contiene Programación (Pojo) para su manejo dentro de las reglas del negocio del sistema

**Aprobación:** Modelo que permite la implementación de los atributos y métodos que contiene Aprobación (Pojo) para su manejo dentro de las reglas del negocio del sistema

**Proyecto Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Acta Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Participante Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Compromiso Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Programación Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Aprobación Repositorio:** Contiene los Scripts de persistencia y consulta a la base de datos, permitiendo facilidad y velocidad en la interacción a la hora de realizar procedimientos como la persistencia entre otros.

**Persistencia:** Es el motor de base de datos que se encara de la gestión de la información, de entregar y gestionar las conexiones necesarias para salvar los datos y de la entrega de los mismos.

# Descripción del Componente Externo

La aplicación web será desarrollada en Visual Studio c# asp .net.

* Usando mvc el framework se generan los paquetes Vistas y Controladores.
* Los paquetes Modelos, Repositorios y Persistencia se generarán cada uno como biblioteca dll.
* El acceso a datos en el componente Persistencia se hará a través del componente EntityFramework de .net
* El Paquete Servicios expondrá los servicios web usando SOAP con la intención de implementar el paradigma SOA.