



**Sistema de Gestión de Programas Analíticos de Estudio
(SIGPAE)
Documento de la Arquitectura del Software**

Versión 1.7

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Historia de Revisión

Fecha	Versión	Descripción	Autor
20/10/2012	1.0	Versión Inicial	Guava Solutions
06/11/2012	1.1	Segunda Versión	Guava Solutions
16/11/2012	1.1	No hubo cambios	Guava Solutions
15/02/2013	1.2	Tercera versión	Guava Solutions
28/02/2013	1.3	Cuarta versión	Guava Solutions
08/03/2013	1.4	Quinta versión	Guava Solutions
02/04/2013	1.5	Sexta versión	Guava Solutions
17/05/2013	1.6	Séptima versión	Guava Solutions
20/06/2013	1.7	Octava versión	Guava Solutions

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Tabla de Contenidos

1.	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Siglas, y Abreviaciones	4
1.4	Referencias	5
1.5	Vista Global	5
2.	Representación Arquitectónica	5
3.	Metas y Restricciones Arquitectónicas	6
4.	Vista de Casos de Uso	7
5.	Vista Lógica	8
5.1	Visión general	8
5.2	Paquetes de Diseño Significativos Arquitectónicamente	9
5.3	Realizaciones de los casos de uso	10
6.	Vista de Procesos	16
7.	Vista de Implantación	16
8.	Vista de Implementación	16
9.	Vista de Datos	20
	9.1 Diagrama ERE	20
	9.2 Diccionario de Datos	22
10.	Tamaño y Desempeño	29
11.	Calidad	29

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Documento de la Arquitectura del Software

1. Introducción

El Sistema de Gestión de Programas Analíticos de Estudios de la USB tiene como meta cumplir con todos los requerimientos y expectativas planteadas por DACE que engloban a los programas analíticos de estudios. El sistema posee ciertas funcionalidades que serán descritas a través de una serie de diagramas para mejorar su entendimiento y ayudar a implementarlas con una visión única.

1.1 Propósito

Este documento provee una visión general de la arquitectura del Sistema de Gestión de Programas Analíticos de Estudios, usando distintas vistas arquitectónicas para representar distintos aspectos del sistema. Reúne y comunica las decisiones arquitectónicas importantes que se han hecho sobre el sistema, brindando al lector una visión global y comprensible del diseño general del mismo.

El modelado de las vistas permitirá crear un modelo del sistema mucho más completo y una base sólida para su desarrollo. En la versión inicial, incluye la vista de casos de Uso con el diagrama de casos de uso, la vista lógica con el modelo conceptual y el diagrama de clases y la vista de datos con el diagrama ER y el diccionario de datos.

1.2 Alcance

El documento provee una visión general de la arquitectura del Sistema de Gestión de Programas Analíticos de Estudios (SIGPAE). Se omiten las vistas de procesos, implementación e implantación ya que no nos conciernen en esta versión.

El Documento de la Arquitectura del Software (DAS) es una de las referencias centrales dentro de la documentación: los diagramas de Casos de Uso, Documento Visión, y prácticamente todas las herramientas de diseño de software, se ven afectados por la arquitectura o estructura del sistema (asimismo todos los demás documentos serán, de manera indirecta, derivados de éste).

1.3 Definiciones, Siglas, y Abreviaciones

Se presentan las siglas a emplear a continuación, con su significado y definiciones:

- USB: Universidad Simón Bolívar.
- DII: Dirección de Ingeniería de Información de la Universidad Simón Bolívar.
- DSI: Departamento de Seguridad Integral de la USB
- GPA: Gestor de Programas Analíticos.
- DST: Dirección de Servicios Telemáticos de la USB.
- LDAP: Lightweight Directory Access Protocol.
- CAS: Central Authentication Service
- RUP: *Rational Unified Process*. Se trata de una metodología para describir el proceso de desarrollo de software.
- UML: *Unified Modeling Language*. Es el lenguaje de modelado visual de sistemas de software.
- MVC: Patrón de arquitectura orientada a capas Modelo-Vista-Controlador, llamado así por las siglas en inglés de Model-View-Controller.

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

- ERE: Entidad Relación Extendido
- Usuario: Personas que tendrán acceso al sistema y que podrán, dependiendo del grado o nivel de privilegio establecido, realizar acciones dentro del mismo.
- Cliente: para efectos del Sistema, se refiere a la “Comisión de Convenios” de la USB.
- Stakeholders: interesados o involucrados en cierto problema que necesitan una solución óptima. Desde el punto de vista del desarrollo de sistemas, son personas o entidades que están interesada en la realización de un proyecto, auspiciando el mismo ya sea mediante su poder de decisión o de financiamiento.☐

1.4 Referencias

- Documento Visión 1.2 – Septiembre-Diciembre 2012 – Guava Solutions
- Documento ERS Versión 1.9 – Abril-Julio 2013 – Guava Solutions
- Diagramas de casos de uso – Abril-Julio 2013 – Guava Solutions
- Diagrama de vista lógica: de clases y de dominio – Abril-Julio – Guava Solutions
- Modelo ERE – Abril-Julio 2013 – Guava Solutions

1.5 Vista Global

El Documento de Arquitectura de Software consta de 11 secciones debidamente definidas. Se desarrollará detalladamente la representación arquitectónica para establecer lo que se quiere funcional y estructuralmente del sistema, además con ayuda de las vistas se podrá explicar de la mejor manera el desempeño del mismo.

Con exactitud diremos que se describe con precisión lo que es la arquitectura del software, además de los requerimientos y objetivos del mismo que pueden impactar relevantemente en la arquitectura. Ésta será descrita con exactitud con las visiones que se desarrollan a lo largo de varias secciones, y que permitirán analizar el sistema por partes para poder especificar las totalidades de sus funcionalidades y como éstas cumplen los requerimientos del cliente.

2. Representación Arquitectónica

La Arquitectura de Software básicamente es considerada como una vista del sistema donde se incluyen sus componentes principales y la interacción y coordinación entre ellos, necesarios para cumplir con los requerimientos específicos del sistema.

En este documento se utiliza el Modelo 4+1 Vistas de Kruchten, el cual organiza la descripción de la arquitectura de software en cinco vistas donde se exponen las decisiones de diseño y una quinta vista para ilustrar y validar dichas decisiones. Estas vistas se componen por la de casos de uso, lógica, procesos, implementación e implantación. Es importante destacar que éstas son complementadas con elementos de UML que proporcionen más información a nivel gráfico. Las vistas son las siguientes:

- La vista de casos de uso, también llamada escenarios, contiene los casos de uso críticos del sistema, representados mediante el diagrama de casos de uso. Es de gran importancia ya que a través de esta vista se puede entender mejor la funcionalidad del sistema y funciona como un indicador que ayuda al diseñador a descubrir los elementos de la arquitectura durante su diseño.

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

- La vista lógica comprende las abstracciones fundamentales del sistema a partir del dominio del problema, trata principalmente los requerimientos funcionales del sistema y forma la estructura del mismo, incluye el modelo conceptual y el diagrama de clases.
- La vista de procesos describe el diseño de concurrencia y aspectos de sincronización y muestra algunos de los requisitos no funcionales, como son desempeño, rendimiento, seguridad, confiabilidad, entre otros.
- La vista de desarrollo o implantación expone las características y requisitos que debe cumplir el sistema a nivel técnico.
- La vista de implementación engloba la distribución de paquetes de la aplicación y de manera general de la distribución y funcionamiento de las distintas capas del sistema. Describe el mapeo del software en el hardware.

Finalmente, aunque el modelo no cuenta con una vista de datos, en el documento se incluye, ya que permite englobar la capa en donde se almacenan todos los datos que se ingresan en el sistema. Para esta vista nos apoyaremos en un diagrama ERE, su diccionario de datos y restricciones consideradas. Así mismo, en esta versión del documento solo se presentará la vista lógica, la vista de casos de uso y la vista de datos.

3. Metas y Restricciones Arquitectónicas

La principal meta en el desarrollo del proyecto, es lograr un sistema de alta calidad, eficiente, que cumpla con los requerimientos establecidos y sea sencillo, confiable y amigable de forma tal que facilite la interacción con el usuario. Básicamente se pide que el sistema se desarrolle de forma modular, con la idea de facilitar su mantenimiento, al igual que se requiere que sea multiplataforma para proveer de mayor libertad al cliente y no limitarlo a una sola tecnología. Se desea garantizar a las coordinaciones, departamentos, profesores, DACE y estudiantes una manera sencilla y cómoda de gestionar los programas de estudios de la USB.

En cuanto a las restricciones, la DII establece una serie de lineamientos que deben ser respetados, como los que se presentan a continuación:

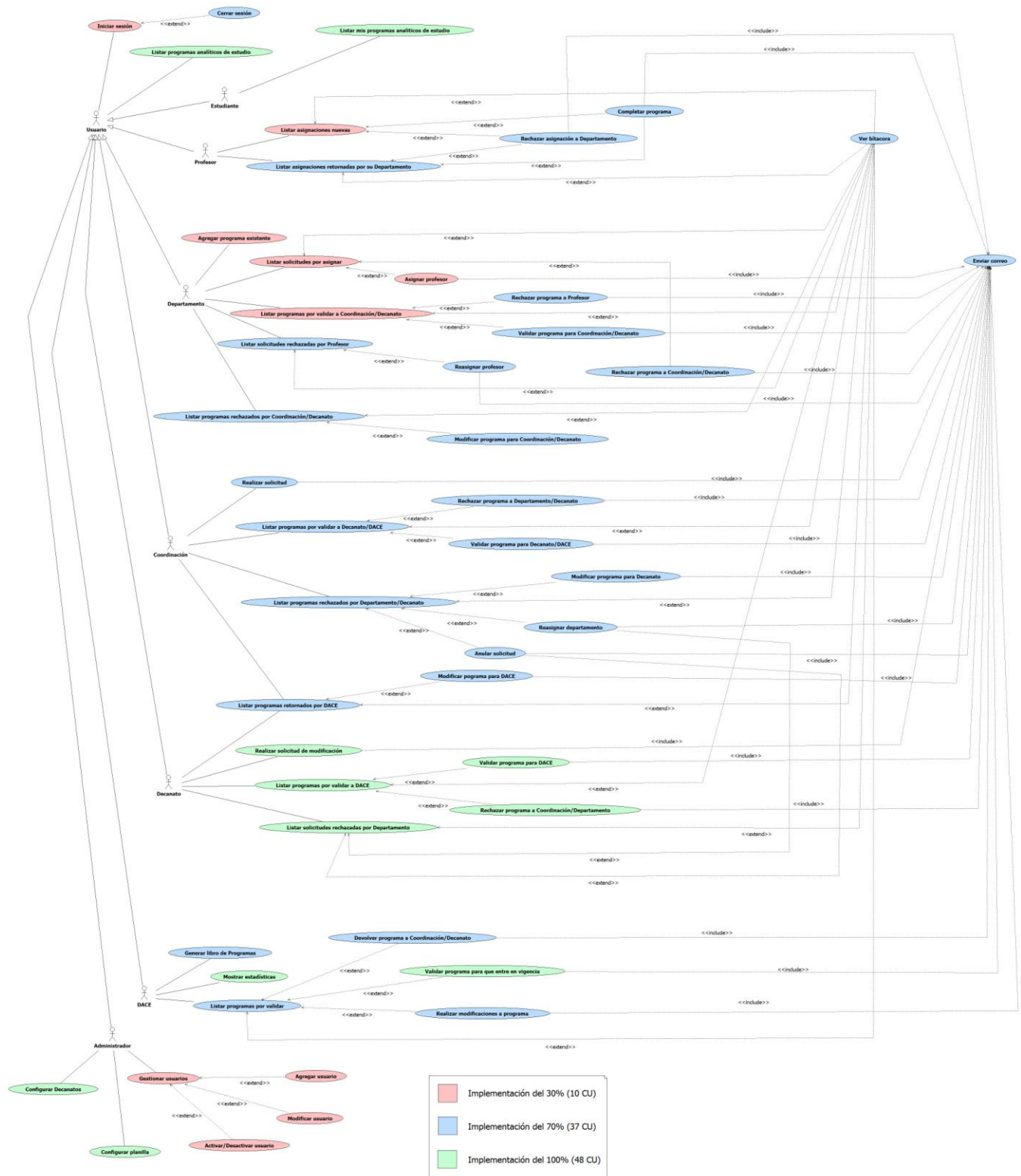
- Uso del lenguaje de programación Java
- Uso del DBMS PostgreSQL.
- Uso del servidor de aplicaciones JBoss versión 4.2.3.GA.
- Implementación mediante el Framework Struts 1.x
- Utilizar la arquitectura MVC (modelo, vista, controlador) como base para dividir la funcionalidad de los objetos que participan en el mantenimiento y presentación de datos.

Una de las más grandes restricciones que se tiene es de tiempo, dado que hay una fecha fijada de entrega de este sistema. Así mismo, para el cumplimiento del ejercicio práctico al que corresponde el desarrollo del sistema, es obligatorio desarrollar el sistema bajo la metodología RUP y utilizar TRAC para gestionar el proyecto. Finalmente, no es necesario ajustarse a ninguna arquitectura implementada con anterioridad.

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

4. Vista de Casos de Uso

A continuación se presenta el modelo de casos de uso refinado:



Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

5. Vista Lógica

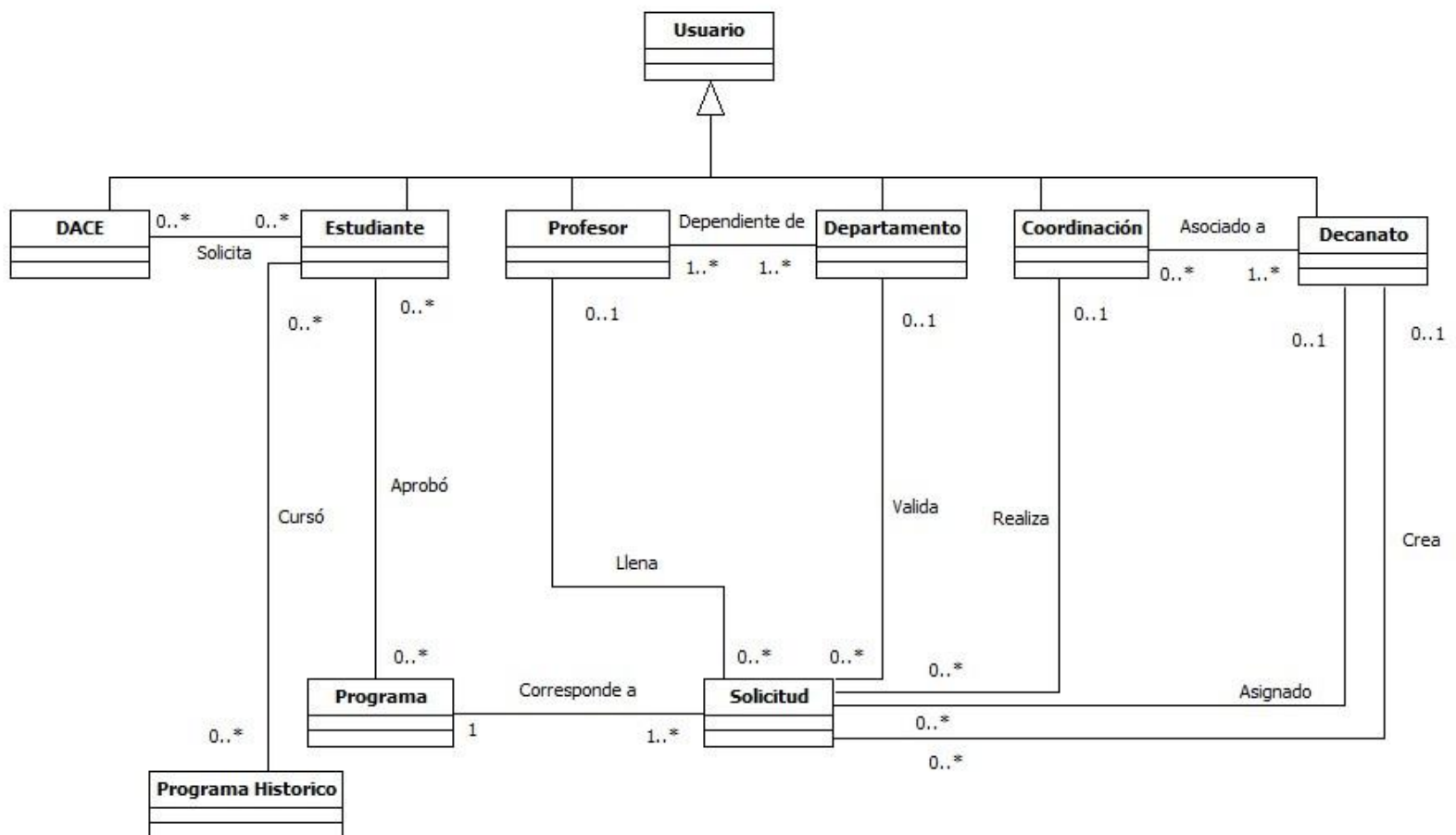
Se describen las partes del modelo de diseño que son significativas arquitectónicamente, tales como, su descomposición dentro de subsistemas y paquetes de servicios; y para cada paquete, su descomposición dentro de clases y utilidades de clases. Introducir las clases relevantes arquitectónicamente y describir sus responsabilidades, así como sus relaciones, operaciones y atributos.

5.1 Visión general

El modelo de diseño del Sistema de Gestión de Programas Analíticos de DACE, tiene sus bases más importantes en dos divisiones, por un lado tiene que ser capaz de poder permitirle a los departamentos una subida de los programas analíticos históricos, de manera sencilla, para que se pueda generar una base de datos coherente y eficiente, que posea de manera concreta toda la información, ya sea vieja o la nueva por introducir, dentro del sistema.

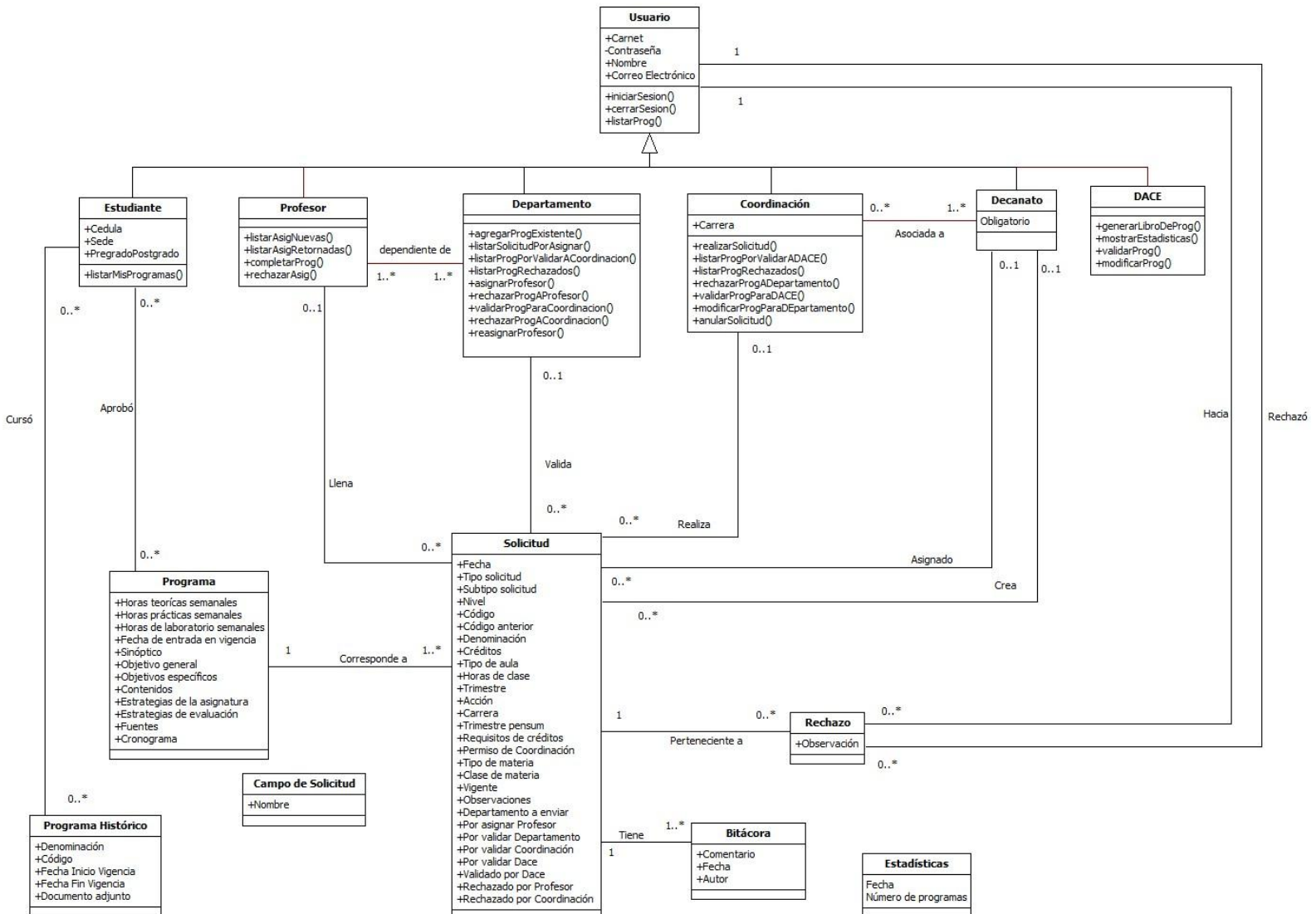
Por el otro lado, es necesario garantizar la comunicación y envío de solicitudes entre las diferentes coordinaciones y departamentos, para que estos verifiquen la información, y en caso de estar correcta, la validen. En el siguiente modelo se aprecia las clases a ser utilizadas, con sus respectivas operaciones, para resolver la problemática explicada anteriormente.

A continuación se presenta el modelo de dominio:



5.2 Paquetes de Diseño Significativos Arquitectónicamente

A continuación se presenta el diagrama de clases:

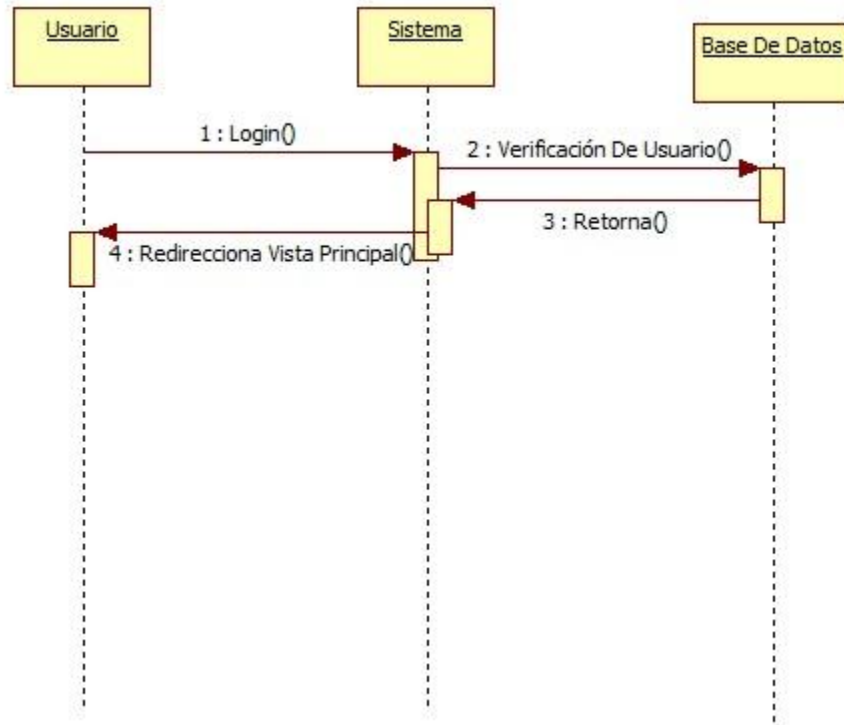


Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

5.3 Realizaciones de los casos de uso

A continuación se presentan los diagramas de secuencia de los casos de uso críticos:

Iniciar sesión:



1: El usuario ingresa USBID y contraseña

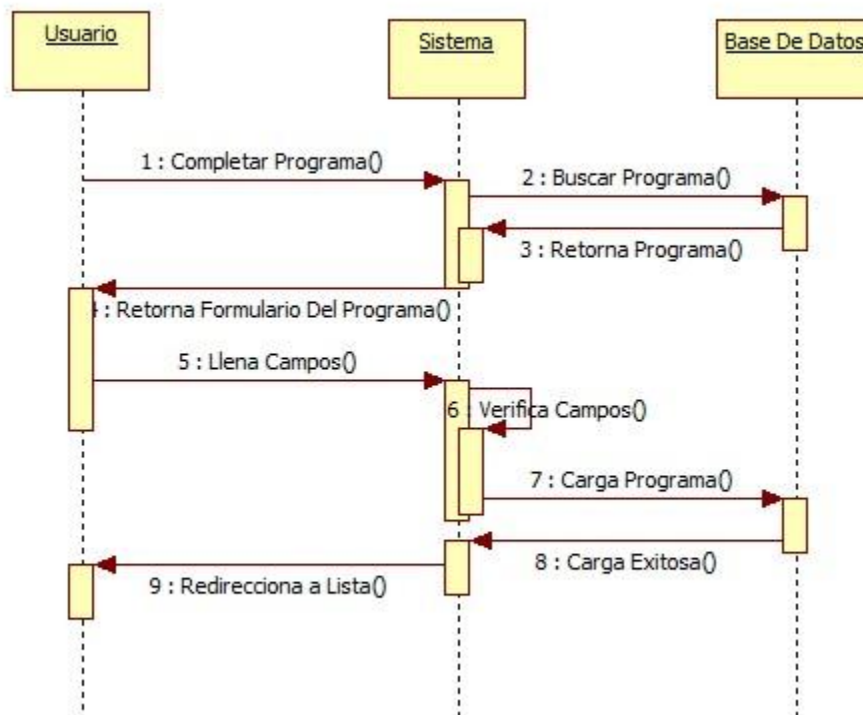
2: El sistema verifica los datos con la base de datos

3: La base de datos retorna la verificación de los datos

4: El sistema re direcciona al usuario a la vista principal según su rol

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

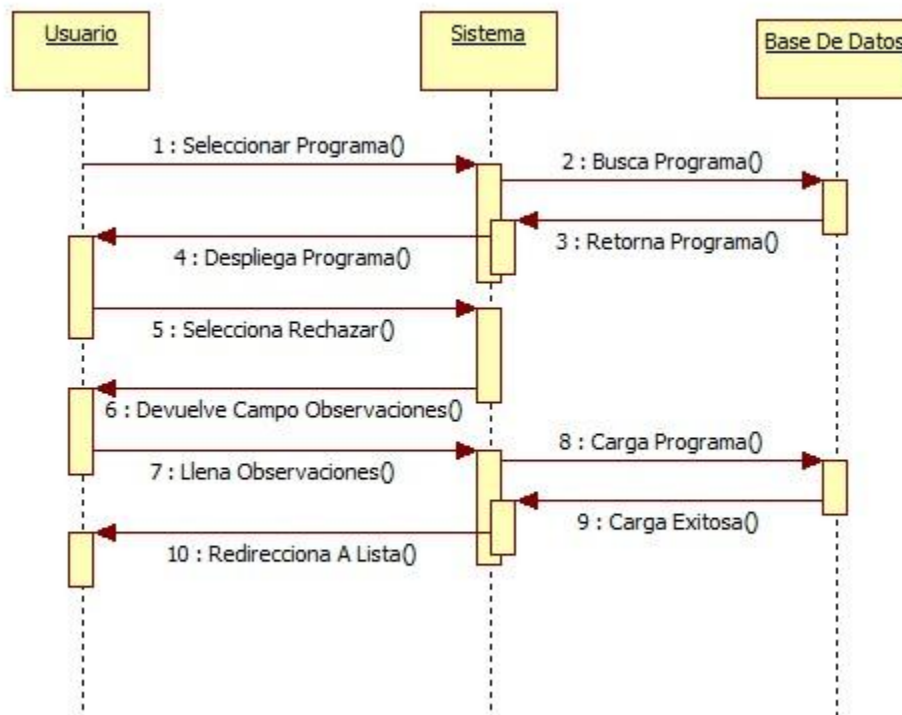
Completar programa:



- 1: El usuario selecciona la opción de completar programa en la lista de asignaciones que posee
- 2: El sistema busca el programa en la base de datos
- 3: La base de datos retorna el programa una vez que ha sido encontrado
- 4: El sistema muestra al usuario el formulario del programa
- 5: El usuario llena los campos del programa
- 6: El sistema verifica los campos
- 7: El sistema carga el programa en la base de datos
- 8: La base de datos carga exitosamente el programa
- 9: El sistema re direcciona al usuario a la lista de sus asignaciones

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

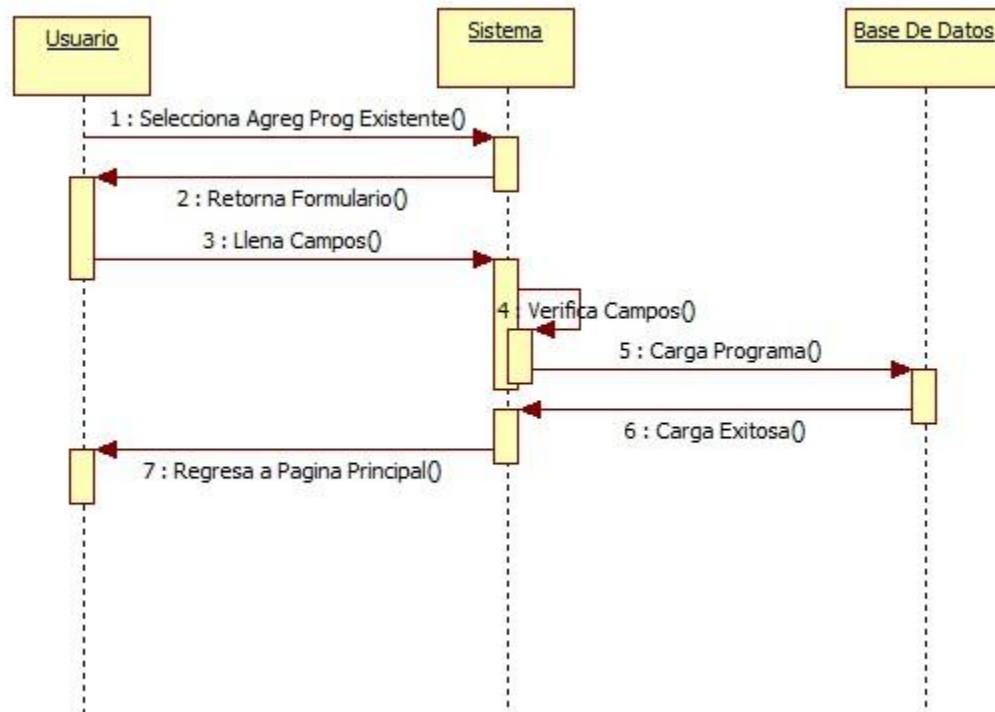
Rechazar programa a Departamento:



- 1: El usuario selecciona el programa que desea rechazar en la lista de asignaciones que posee
- 2: El sistema busca el programa en la base de datos
- 3: La base de datos retorna el programa una vez que ha sido encontrado
- 4: El sistema muestra al usuario la información del programa
- 5: El usuario selecciona la opción de rechazar
- 6: El sistema muestra al usuario el campo de observaciones
- 7: El usuario llena las observaciones de porque va a rechazar el programa
- 8: El sistema carga el programa como rechazado en la base de datos
- 9: La base de datos carga exitosamente el programa rechazado
- 10: El sistema re direcciona al usuario a la lista de sus asignaciones

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

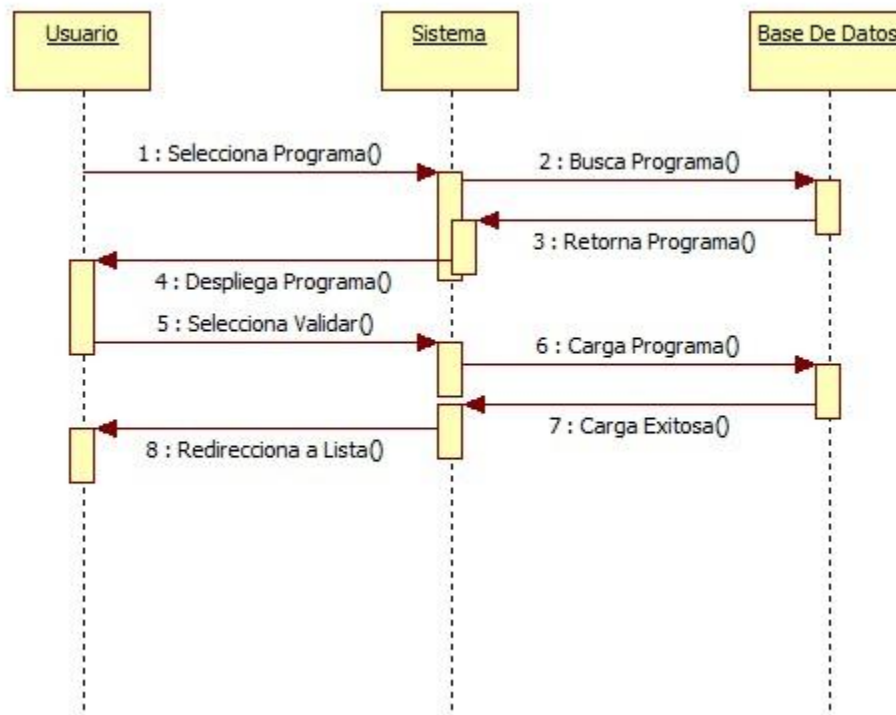
Agregar programa existente:



- 1: El usuario selecciona la opción de agregar programa existente
- 2: El sistema muestra formulario correspondiente al usuario
- 3: El usuario llena campos
- 4: El sistema verifica los campos
- 5: El sistema carga el programa en la base de datos
- 6: La base de datos carga exitosamente el programa
- 7: El sistema re direcciona al usuario a la página principal de Departamento

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Validar programa para Coordinación:



1: El usuario selecciona el programa que desea validar en la lista de asignaciones que posee

2: El sistema busca el programa en la base de datos

3: La base de datos retorna el programa una vez que ha sido encontrado

4: El sistema muestra al usuario la información del programa

5: El usuario selecciona la opción de validar

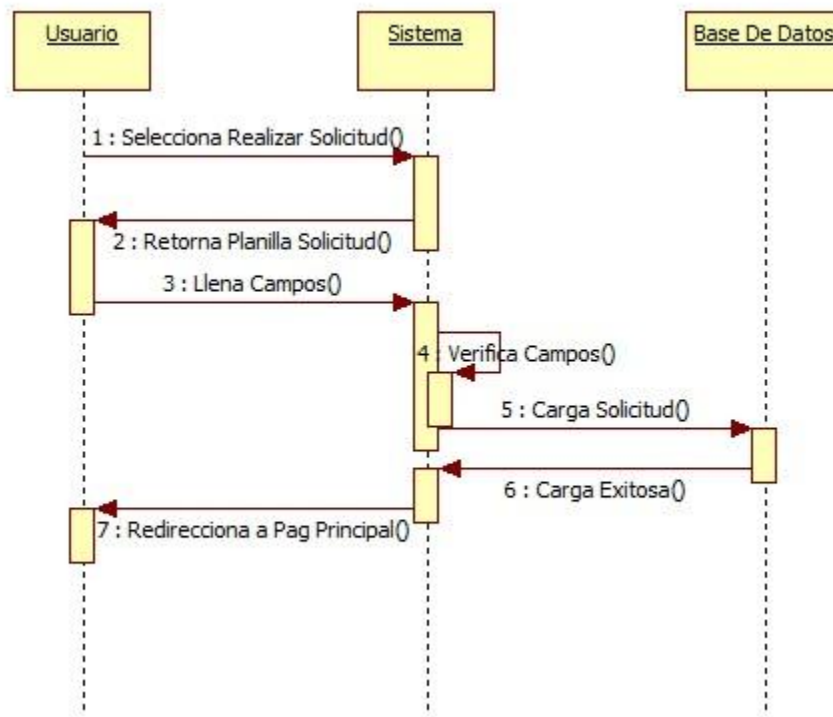
6: El sistema carga el programa como validado en la base de datos

9: La base de datos carga exitosamente el programa validado

10: El sistema re direcciona al usuario a la lista de sus programas por validar

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Realizar solicitud:



- 1: El usuario selecciona la opción de realizar solicitud
- 2: El sistema muestra la planilla de solicitud
- 3: El usuario llena los campos correspondientes
- 4: El sistema verifica los campos
- 5: El sistema carga la solicitud en la base de datos
- 6: La base de datos carga exitosamente la solicitud
- 7: El sistema re direcciona al usuario a la página principal de Coordinación

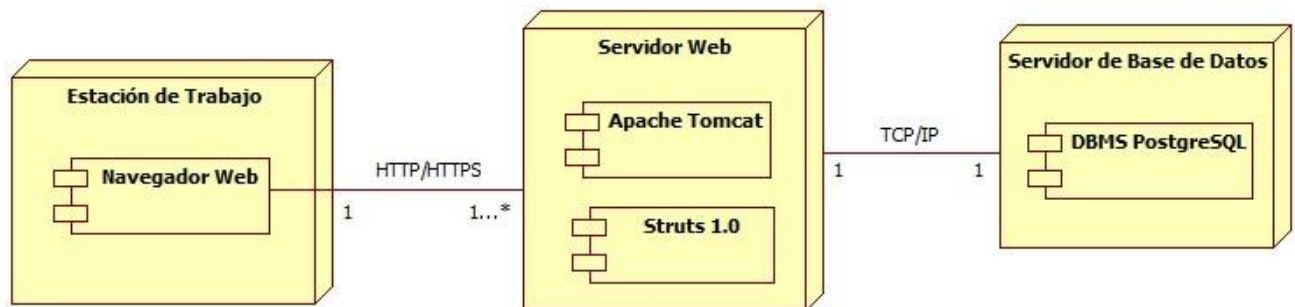
Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

6. Vista de Procesos

La vista de procesos no se va a desarrollar por el momento.

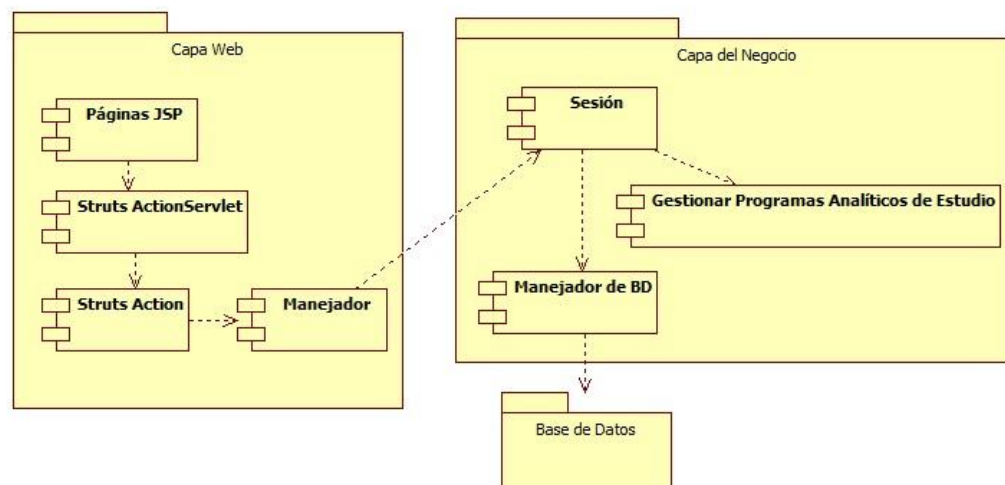
7. Vista de Implantación

A continuación se presenta el diagrama de implantación como un diagrama de despliegue, el cual describe la topología del sistema la estructura de los elementos de software y hardware empleados (los cuales son los nodos) y las relaciones entre ellos (en forma de enlaces de red y conexiones TCP/IP):



8. Vista de Implementación

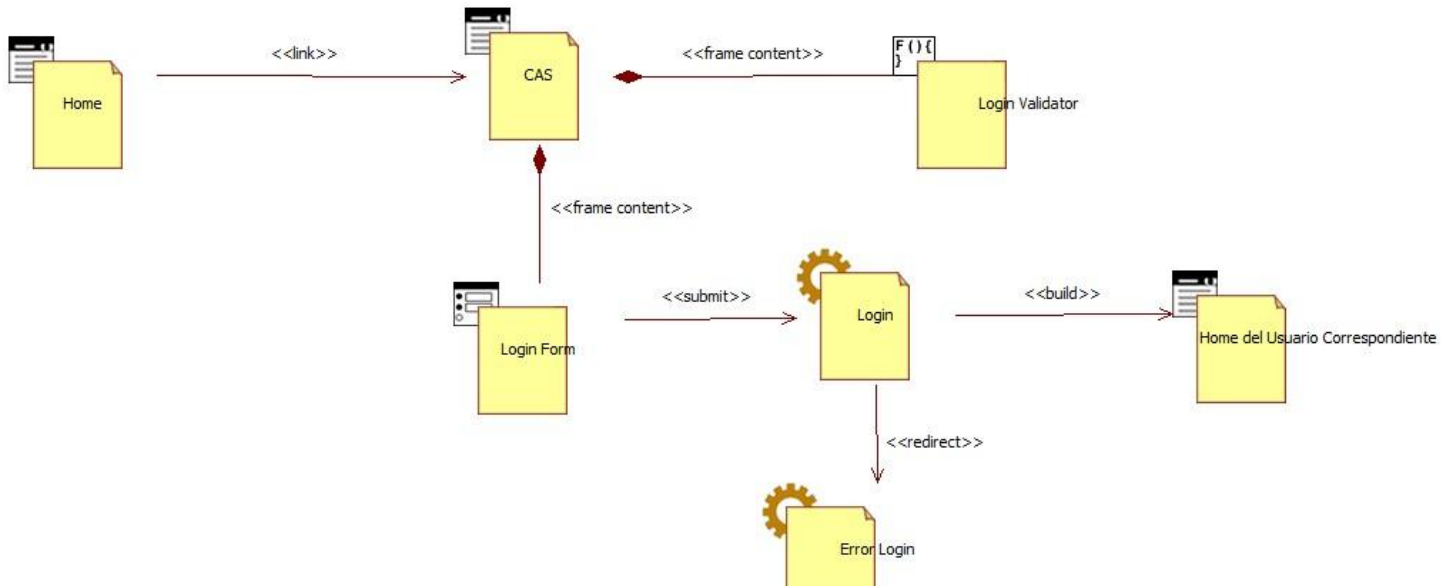
A continuación se presenta el diagrama de implementación como un diagrama de componentes, el cual representa como nuestro sistema se divide en componentes y las dependencias entre ellos:



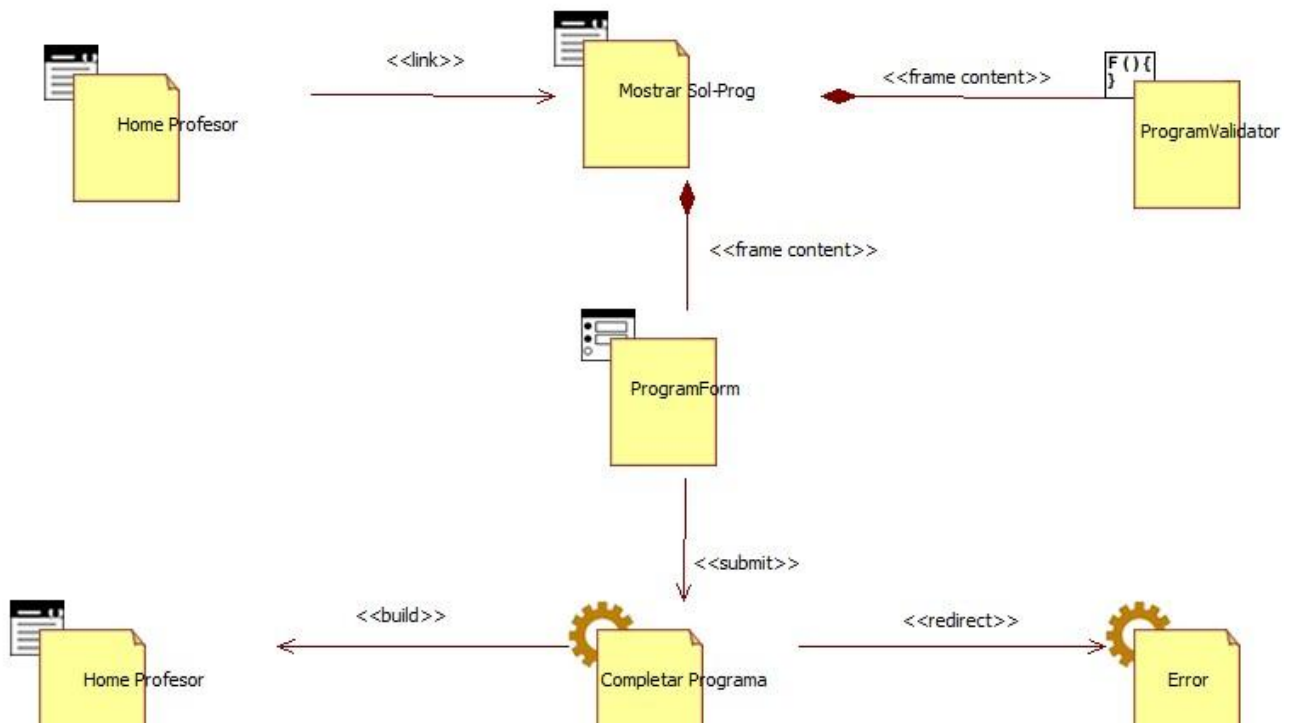
Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

A continuación se presentan los diagramas WAE:

Iniciar sesión:

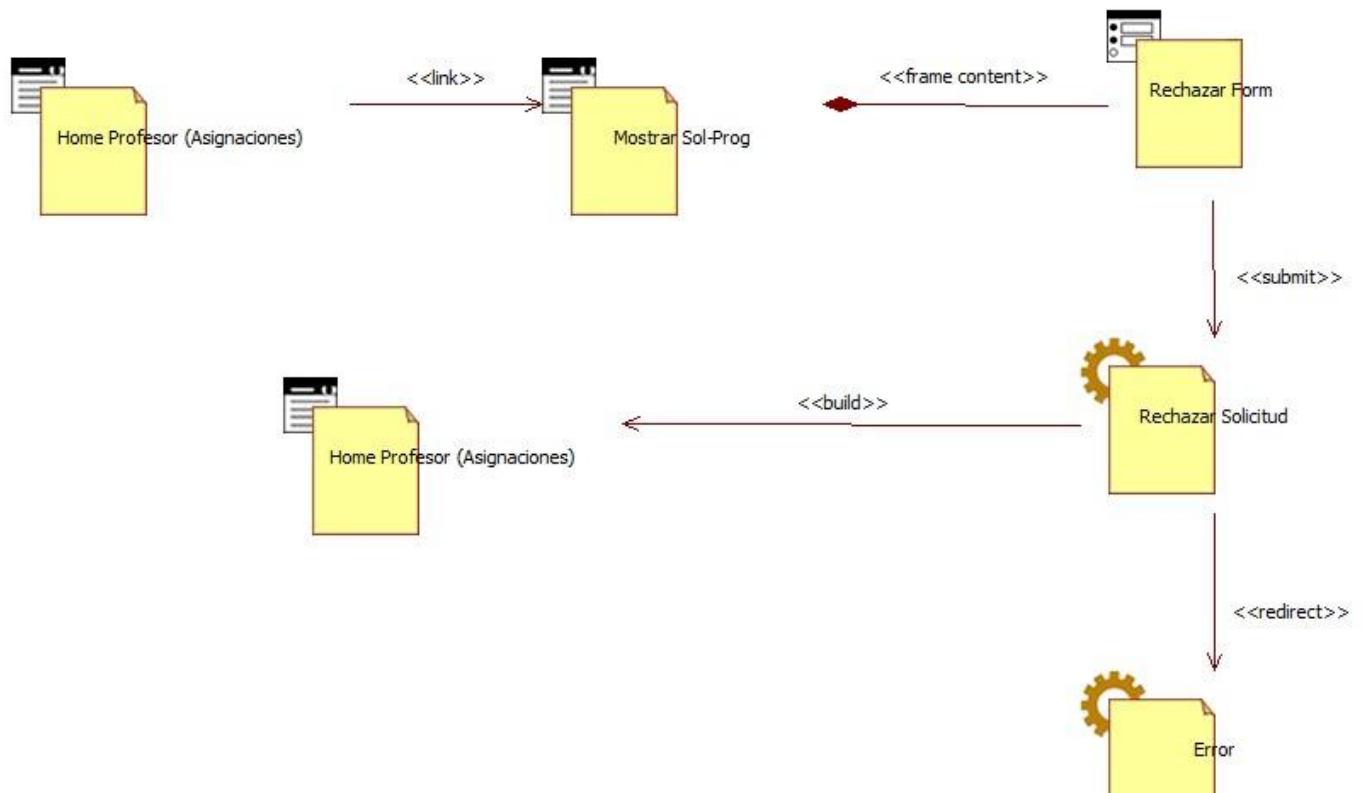


Completar programa:

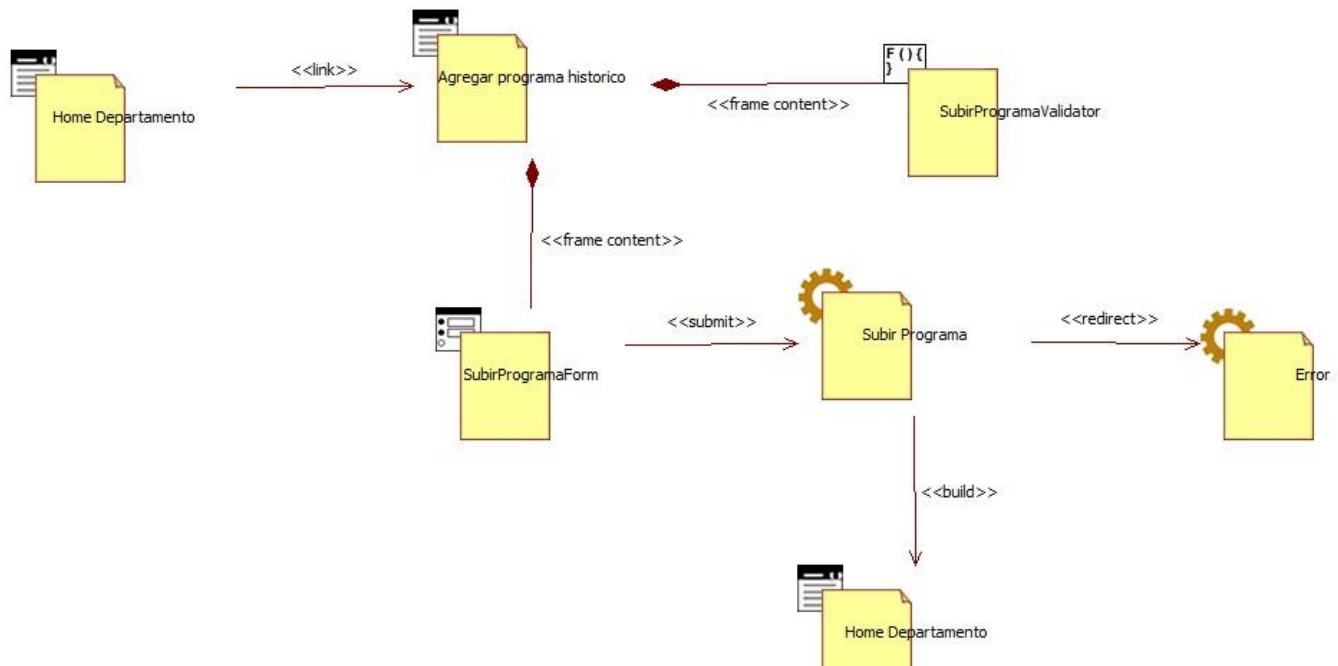


Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Rechazar programa a Departamento:

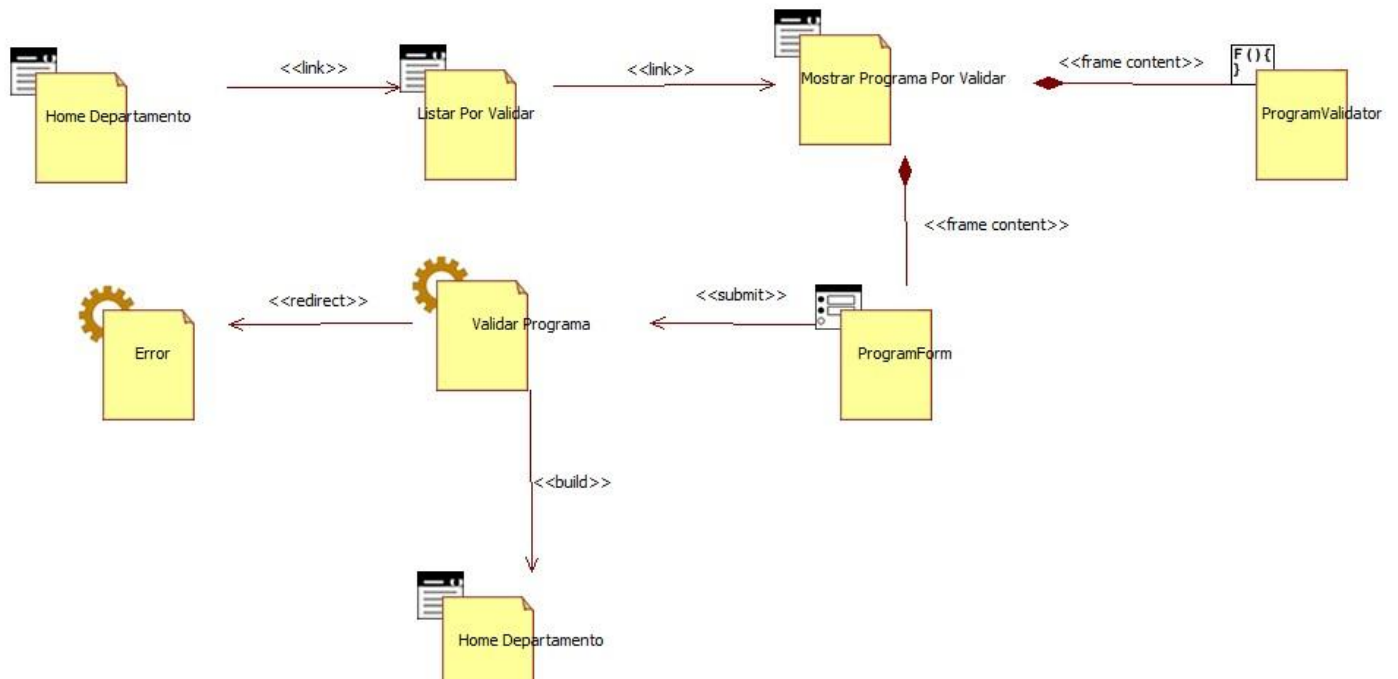


Agregar programa existente:



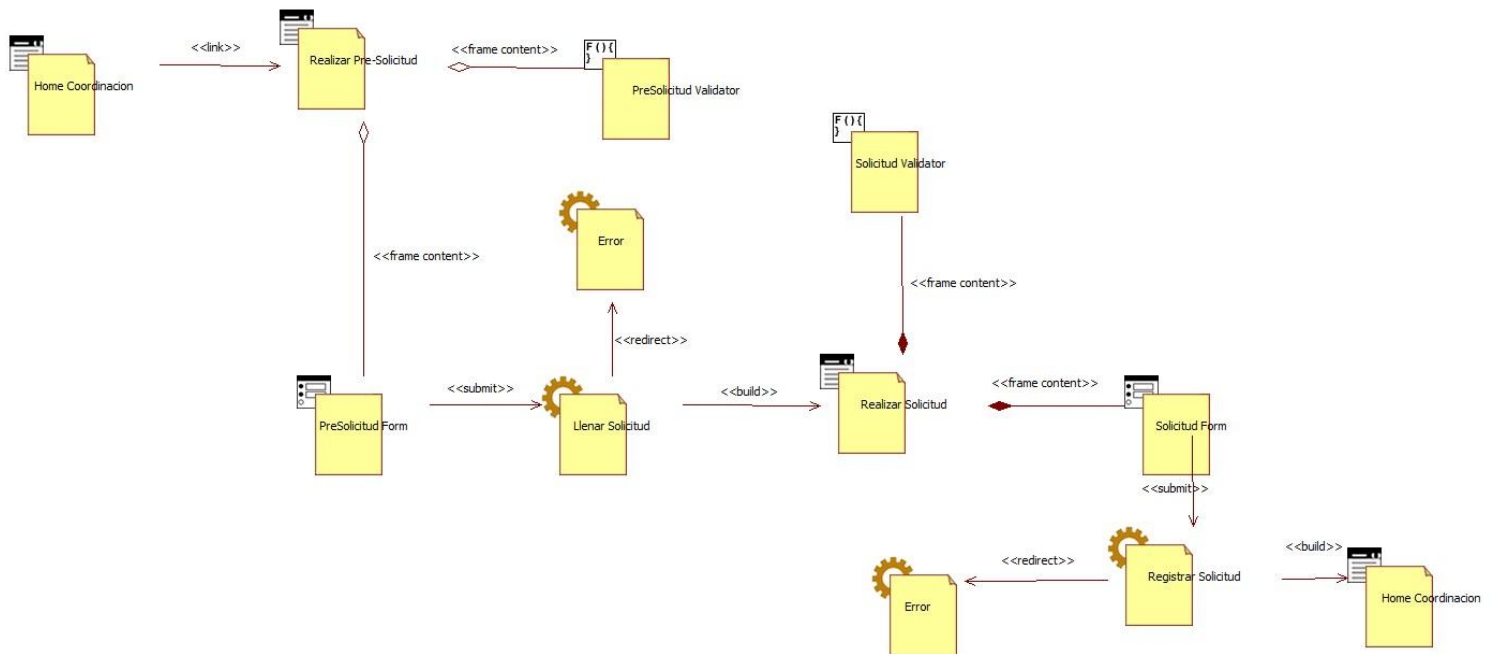
Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Validar programa para Coordinación:



Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

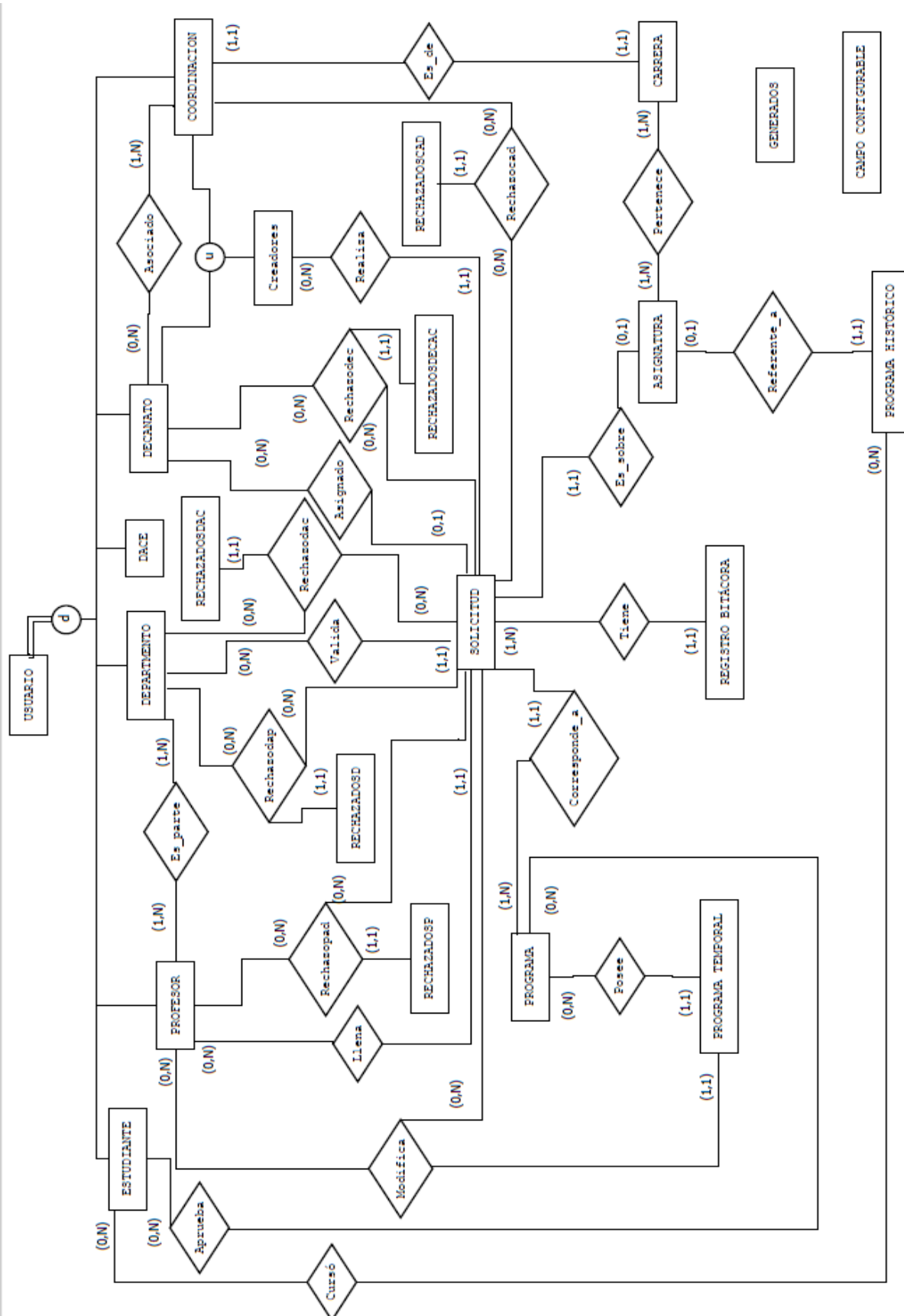
Realizar solicitud:



9. Vista de Datos

9.1 Diagrama ERE

A continuación se muestra el diagrama ERE que muestra la perspectiva del almacenamiento de los datos en el sistema. En el diagrama no se muestran todos los atributos con la finalidad de mostrar un modelo claro a la vista, sin embargo cada uno de ellos será especificado en el Diccionario de Datos (Sección 9.2)



Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

9.2 Diccionario de Datos

A continuación se presenta la Tabla 9.2.1 donde se encontrara una descripción de cada una de las entidades definidas, la semántica de las mismas y de sus atributos.

Entidad	Semántica	Atributos	Semántica del atributo	Dominio
Usuario	Usuario del sistema	Usbid	Identificador del usuario dentro de la USB	Secuencia de caracteres
		Clave	Clave del usuario	Secuencia de caracteres
		Nombre	Nombre del usuario	Secuencia de caracteres
		Email	Email del usuario	Secuencia de caracteres
		Tipo	Rol del usuario	Secuencia de caracteres
Estudiante	Estudiante de la USB	Hereda los atributos de la Superclase Usuario	-	-
		Cédula	Cedula del estudiante	Secuencia de caracteres
		Tipo	Indica si el estudiante es de Pregrado o Postgrado y si pertenece a la sede del Litoral o Sartenejas	Número entero entre 1 y 4
Solicitud	Solicitud de control de asignatura	Id	Identificador de la solicitud	Numero Real positivo distinto de cero
		NomCoord	Nombre del creador de la solicitud	Secuencia de caracteres
		PorAsignar	Indica si la solicitud se encuentra en el estado de por asignar profesor	Booleano
		PorValidarD	Indica si la solicitud se encuentra en el estado de por validar por departamento	Booleano
		PorRevisarP	Indica si la solicitud se encuentra en el estado de por completar por profesor	Booleano
		RechazadoC	Indica si la solicitud fue rechazada por la Coordinación	Booleano
		ValidadoC	Indica si la solicitud fue validada por la Coordinación	Booleano
		EnviadoD	Indica si el Departamento envió la solicitud a la Coordinación	Booleano

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

		DevueltoDace	Indica si Dace retorno el programa a la Coordinación o Decanato	
		FechaElab	Fecha de elaboración	DateTime
		Tipo_Solicitud	Tipo de solicitud (Creación, Modificación y Eliminación)	Secuencia de caracteres
		SubTipo_Solicitud	Subtipo de la solicitud (Inclusión, Modificación de planilla y Modificación de programa)	Secuencia de caracteres
		Nivel	Indica el nivel de la asignatura por ejemplo: Pregrado o Postgrado	Secuencia de caracteres
		Cod	Código de la asignatura	Secuencia de caracteres
		Cod_Anterior	Código anterior, para cuando se está modificando el código	Secuencia de caracteres
		Denominación	Nombre de la asignatura	Secuencia de caracteres
		Creditos	Créditos de la asignatura	Numero Real positivo distinto de cero
		Tipo_Aula	Indica el tipo de aula que requiere la asignatura	Secuencia de caracteres
		HTeoría	Indica si la asignatura tiene horas de teoría o no	Booleano
		HPractica	Indica si la asignatura tiene horas de practica o no	Booleano
		Hlab	Indica si la asignatura tiene horas de laboratorio o no	Booleano
		Trime	Trimestre en que se dicta la asignatura	Secuencia de caracteres
		Accion	Acción de la solicitud ("I", "M" o "E")	Secuencia de caracteres
		Afecta_Carrera	Nombre de la carrera afectada	Secuencia de caracteres
		TrimestreP	Indica si la asignatura se dicta en trimestre pensum o no	Booleano
		Requisito_cre	Indica si la asignatura tiene requisitos de cantidad de créditos	Booleano
		Permiso_coord	Indica si la asignatura necesita permiso de coordinación o no	Booleano

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

		Tipo_Materia	Indica el tipo de la asignatura como por ejemplo: Electiva o extraplan.	Secuencia de caracteres
		Clase_Materia	Clase de la materia	Secuencia de caracteres
		Observaciones	Observaciones	Secuencia de caracteres
		Vigente	Indica si la solicitud está vigente	Booleano
		ValidadoDace	Indica si la solicitud fue validada por Dace	Booleano
		Especial	Indica si la solicitud corresponde a una asignatura especial	Booleano
		Impartición	Indica la forma de impartir la asignatura como por ejemplo presencial o a distancia	Secuencia de caracteres
		Obsanul	Indica la observación en caso de que la solicitud sea anulado	Secuencia de caracteres
Programa	Programa Analítico de asignatura de la USB	Id	Identificador del programa	Serial
		H_teoría	Número de horas de teoría	Número Real positivo
		H_prac	Número de horas de práctica	Número Real positivo
		H_lab	Número de horas del laboratorio	Número Real positivo
		Fecha_vigTrim	Trimestre de entrada en vigencia del programa	Secuencia de caracteres
		Fecha_vigAño	Año de entrada en vigencia del programa	Número Real positivo distinto de cero
		Obj_g	Objetivo general que plantea el alcance de los aprendizajes esperados al final de la asignatura	Secuencia de caracteres
		Obj_e	Objetivos específicos	Secuencia de caracteres
		Contenidos	Contenido o ejes temáticos	Secuencia de caracteres
		Estrategias	Estrategias metodológicas que se desarrollaran para alcanzar los aprendizajes	Secuencia de caracteres
		Estrat_eval	Estrategias de evaluación	Secuencia de caracteres
		Fuentes	Fuentes de información	Secuencia de caracteres
		Cronograma	Cronograma de actividades	Secuencia de caracteres

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

		Sinóptico	Síntesis de lo que se desea para el programa analítico	Secuencia de caracteres
Programa Temporal	Programa Temporal que representa el programa de una asignatura que se está modificando.	Id	Identificador del programa temporal	Serial
		H_teoria	Número de horas de teoría	Número Real positivo distinto de cero
		H_prac	Número de horas de práctica	Número Real positivo distinto de cero
		H_lab	Número de horas del laboratorio	Número Real positivo distinto de cero
		Fecha_vigTrim	Trimestre de entrada en vigencia del programa modificado	Secuencia de caracteres
		Fecha_vigAño	Año de entrada en vigencia del programa modificado	Número Real positivo distinto de cero
		Obj_g	Objetivo general que plantea el alcance de los aprendizajes esperados al final de la asignatura	Secuencia de caracteres
		Obj_e	Objetivos específicos	Secuencia de caracteres
		Contenidos	Contenido o ejes temáticos	Secuencia de caracteres
		Estrategias	Estrategias metodológicas que se desarrollaran para alcanzar los aprendizajes	Secuencia de caracteres
		Estrat_eval	Estrategias de evaluación	Secuencia de caracteres
		Fuentes	Fuentes de información	Secuencia de caracteres
		Cronograma	Cronograma de actividades	Secuencia de caracteres
		Sinóptico	Síntesis de lo que se desea para el programa analítico	Secuencia de caracteres
Departamento	Departamento de la USB	Hereda los atributos de la Superclase Usuario	-	-
Asignatura	Asignatura dictada en la USB	CodAsig	Código de la asignatura	Secuencia de caracteres
		Nombre	Nombre de la asignatura	Secuencia de caracteres
		Tipo	Indica si la asignatura es de tipo Especial o Normal	Secuencia de caracteres
Profesor	Profesor de la USB	Hereda los atributos de la Superclase	-	-

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

		Usuario		
Coordinación	Coordinación de una carrera de la USB	Hereda los atributos de la Superclase Usuario	-	-
Carrera	Carrera dictada en la USB	Código	Código de la carrera	Secuencia de caracteres
		Nombre	Nombre de la carrera	Secuencia de caracteres
Decanato	Decanato de la USB	Hereda los atributos de la Superclase Usuario	-	-
DACE	Usuarios representantes de DACE	Hereda los atributos de la Superclase Usuario	-	-
Programa Histórico	Programa no vigente de una asignatura de la USB	Id	Identificador del programa histórico	Serial
		Trimestrei	Trimestre de entrada en vigencia	Secuencia de caracteres
		Anoi	Año de entrada en vigencia	Número Real positivo distinto de cero
		Trimestref	Trimestre en que dejó de ser vigente	Secuencia de caracteres
		Anof	Año en que dejó de ser vigente	Número Real positivo distinto de cero
		Ruta	Ruta del archivo del programa	Secuencia de caracteres
RechazadosP	Rechazo de un profesor a la asignación de su Departamento	Usbidd	Usbid del departamento al que se rechaza la solicitud	Secuencia de caracteres
		Observacionesrd	Descripción del motivo del rechazo	Secuencia de caracteres
		Num	Identificador del rechazo	Serial
RechazadosD	Rechazo de un Departamento a un profesor	Usbidp	Usbid del profesor al que se rechaza el programa	Secuencia de caracteres
		Observacionesrd	Descripción del motivo del rechazo	Secuencia de caracteres
		Num	Identificador del rechazo	Serial
RechazadosDaC	Rechazo de un Departamento a una Coordinación	Usbidc	Usbid de la Coordinación a la que se rechaza la solicitud	Secuencia de caracteres
		Observacionesrd	Descripción del motivo del rechazo	Secuencia de caracteres
		Num	Identificador del rechazo	Serial
RechazadosCaD	Rechazo de una Coordinación a un Departamento	Usbidd	Usbid del Departamento al que se rechaza la solicitud	Secuencia de caracteres
		Observacionesrd	Descripción del motivo del rechazo	Secuencia de caracteres

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

		Num	Identificador del rechazo	Serial
RechazadosDEC aC	Rechazo de un Decanato a una Coordinación	Usbidcoor	Usbid de la Coordinación a la que se rechaza la solicitud	Secuencia de caracteres
		Observacionesrd	Descripción del motivo del rechazo	Secuencia de caracteres
		Num	Identificador del rechazo	Serial
Registro bitácora	Entrada de la bitácora de una solicitud	Usbid	Identificador del usuario que realizó la acción a registrar en la bitácora	Secuencia de caracteres
		Id	Identificador de cada entrada de la de bitácora de una propuesta	Serial
		Comentario	Contenido del comentario que explica la acción realizada	Secuencia de caracteres
		Fecha	Fecha y hora en que se realizó la acción a registrar	DateTime
Generados	Información sobre la generación de un libro de programas por parte de DACE	Id	Identificador del libro generado	Serial
		Fecha	Fecha en que se genero el libro de programas	DateTime
		Nro_pag	Número de páginas del libro de programas	Secuencia de caracteres
		Nro_prog	Número de programas incluidos en el libro	Secuencia de caracteres
Creadores	Posibles creadores de una solicitud, incluye Coordinaciones y Decanatos (Categoría)	-	-	-

A continuación se presenta la tabla 9.2.2 donde se encontrara una descripción de cada una de las interrelaciones definidas, semántica de las mismas y de sus atributos.

Interrelación	Semántica	Atributos	Semántica del atributo	Dominio
Corresponde_a(p,s)	El programa p corresponde a la solicitud s	-	-	-
Es_sobre(s,a)	La solicitud s es sobre la asignatura a	-	-	-
Referente_a(ph,a)	El programa histórico ph es referente a la asignatura a	-	-	-
Realiza(c,s)	Un creador realiza la solicitud s	-	-	-
Valida(d,s)	El departamento d valida la solicitud s	-	-	-

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Es_parte(p,d)	El profesor p es parte del departamento d	-	-	-
Llena(p,s)	El profesor p llena el programa de la solicitud s	-	-	-
Asignado(d,s)	El decanato d está asignado a la solicitud s			
Es_de(c,ca)	La Coordinación c es de la carrera c	-	-	-
Pertenece(a,c)	La asignatura a pertenece a la carrera c	-	-	-
Posee(p,pt)	El programa p posee el programa temporal pt	-	-	-
Modifica(pf,pt,s)	El profesor pf modifica el programa temporal pt de la solicitud s	-	-	-
Rechazopad(s,p,r)	El profesor p rechaza la solicitud s con el rechazo r a un Departamento	-	-	-
Rechazodap(s,d,r)	El departamento d rechaza la solicitud s con el rechazo r a un profesor	-	-	-
Rechazodac(s,d,r)	El departamento d rechaza la solicitud s con el rechazo r a una Coordinación	-	-	-
Rechazocad(s,c,r)	La Coordinación c rechaza la solicitud s con el rechazo r a un Departamento	-	-	-
Rechazodec(s,dc,r)	El decanato d rechaza la solicitud s con el rechazo r a una Coordinación	-	-	-
Aprueba(e,p)	Un estudiante e aprueba el contenido de un programa p	-	-	-
Cursó(e,ph)	Un estudiante e cursó el contenido de un programa histórico ph	-	-	-
Tiene(s,b)	La solicitud s tiene el registro de bitácora b	-	-	-

Sistema de Gestión de Programas Analíticos de Estudio (SIGPAE)	Versión: 1.7
Documento de la Arquitectura de Software	Fecha: 20/06/2012

Asociado(d,c)	El Decanato d está asociado con la Coordinación c	-	-	-
---------------	---	---	---	---

10. Tamaño y Desempeño

- El sistema a desarrollar estará basado en tecnología Web, el tiempo estimado de respuesta no será demasiado alto.
- El recurso necesario en el usuario para la utilización del sistema es un explorador o navegador de internet.
- La tecnología Web utilizada para el desarrollo de este sistema facilita el uso, la comprensión y el manejo por parte del usuario.
- El sistema debe estar diseñado para ser usado por varias personas al mismo tiempo, se estima que un promedio de 10 a 20. En el caso más extremo todos los departamentos, coordinaciones y DACE están haciendo uso del sistema simultáneamente.
- La base de datos debería ser de un tamaño considerable, dado que no se tiene certeza de cuantos programas analíticos podrían llegar a ser registrados en el futuro.

11. Calidad

- En cuanto a la usabilidad, el sistema permitirá un manejo sencillo y cómodo para los usuarios, gracias a la sencillez de su interfaz grafica.
- La arquitectura MVC (Modelo-Vista-Controlador) separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo del sistema, por ejemplo, si el sistema debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para el dispositivo móvil, sin necesidad de modificar el controlador y el modelo.
- El sistema es desarrollado con una plataforma de programación que puede funcionar en diversas plataformas lo cual permite que la aplicación pueda funcionar tanto en sistema operativo Windows como en Linux garantizando una alta portabilidad.
- A nivel de seguridad el sistema tiene acceso protegido por el CAS.
- Se depende de factores externos como la estabilidad del servidor donde se instale el sistema, por lo que no es posible garantizar con seguridad la disponibilidad
- El ciclo de vida del proyecto de desarrollo del sistema está basado en fases, divididas en iteraciones. Al concluir cada iteración se verifica si se cumplieron los objetivos para avanzar a la siguiente fase, asegurando así la obtención de un software de buena calidad y bien documentando garantizando su mantenimiento en el tiempo.
- La extensibilidad es la capacidad de agregar o modificar el comportamiento de un objeto
- El desarrollo del sistema basado en módulos permite ser escalable ya que permite la incorporación nuevas características para ampliar sus capacidades de manera fácil