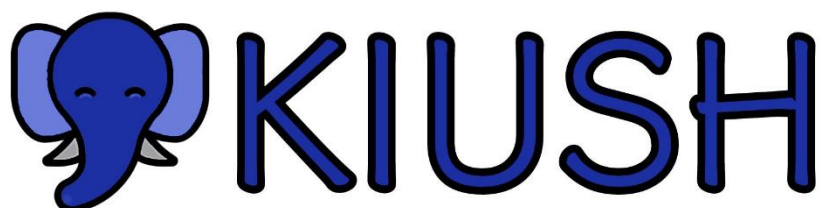


Modelo de Diseño

KIUSH

Bahamonde Yohana, Chuchuy José Martín, Gleadell Carla
Yield Yields



[Este documento es la plantilla base para elaborar el documento Modelo de Diseño.

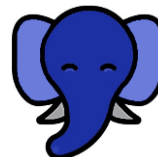
Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección, los cuales se encuentran con estilo “PSI – Comentario”. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda en estilo “Normal”.

Para actualizar la tabla de Contenido, haga clic con el botón derecho del ratón sobre cualquier línea del contenido de la misma y seleccione Actualizar campos, en el cuadro que aparece seleccione Actualizar toda la tabla y haga clic en el botón Aceptar.]

El propósito del Modelo de Diseño es empezar a realizar los casos de uso desarrollados durante las etapas anteriores en especial la etapa de captura de requerimientos. Es decir, tomar el Modelo de Casos de Uso y las Especificaciones Suplementarias creadas con anterioridad entre otros insumos y generar un modelo de diseño que pueda ser usado por los desarrolladores durante la etapa de implementación.

Tabla de contenido

Introducción	5
<i>Propósito</i>	<i>5</i>
<i>Alcance</i>	<i>5</i>
<i>Definiciones, siglas y abreviaturas.</i>	<i>6</i>
Referencias.....	6
<i>Visión general.....</i>	<i>6</i>
Diseño de Casos de Uso	7
<i>Diseño del Caso de Uso: CU9 Modificar curso.....</i>	<i>7</i>
Diagrama de paquetes.....	7
Diagrama de Interacción.....	8
Diseño de Flujo de eventos.....	8
Requerimientos especiales o de implementación	9
<i>Diseño del Caso de Uso: CU10 Asignar Docente</i>	<i>9</i>
Diagrama de paquetes.....	9
Diagrama de Interacción.....	9
Diseño de Flujo de eventos.....	9
Requerimientos especiales o de implementación	10
<i>Diseño del Caso de Uso: CU16 Modificar Preinscripto</i>	<i>10</i>
Diagrama de paquetes.....	10
Diagrama de Interacción.....	11
Diseño de Flujo de eventos.....	11
Requerimientos especiales o de implementación	12
<i>Diseño de Objetos</i>	<i>12</i>
<i>Curso</i>	<i>12</i>
<i>Preinscripto</i>	<i>12</i>
Diseño de Subsistemas	13
<i>Subsistemas Específicos</i>	<i>13</i>
UARGFlow	13
Propósito.....	13
Función	13
Subordinados.....	13
Dependencias	13
Recursos.....	14
Diagramas	15
<i>Diagrama de componentes.....</i>	<i>15</i>



<i>Diagrama de Clases.....</i>	<i>15</i>
<i>Diagrama de Secuencia.....</i>	<i>15</i>
<i>Diagramas de Paquetes</i>	<i>16</i>
<i>Diagrama de Colaboración.....</i>	<i>16</i>

Modelo de Diseño

Introducción

El modelo de diseño es un refinamiento y formalización, donde se toman en cuenta las consecuencias del ambiente de implementación. El resultado son especificación muy detalladas de todos los objetos, incluyendo sus operaciones y atributos. El modelo de diseño se basa en el diseño por responsabilidades.

Propósito

Se requiere un modelo de diseño, ya que el modelo de análisis no es lo suficientemente formal para alcanzar el código fuente. Por tal motivo se refinan los objetos, incluyendo las operaciones y atributos. También otro objetivo, es validar los resultados de los modelos de requisitos y análisis. Durante el diseño, se ve si los resultados anteriores son apropiados para la implementación. Si se descubren aspectos que no están claros en alguno de los modelos anteriores, estos se aclaran, posiblemente regresando a etapas anteriores.

Como el modelo de análisis define la arquitectura general del sistema, se busca obtener la arquitectura detallada como resultado del modelo de diseño, de manera que haya una continuidad de refinamiento entre los 2 modelos

Las estructuras con las cuales se trabaja en el modelo de diseño son básicamente las mismas que en el modelo de análisis. Sin embargo, el punto de vista cambia, ya que se toma un paso hacia la implementación. El modelo de análisis debe verse como un modelo conceptual y lógico del sistema, en tanto el modelo de diseño debe acercarse al código fuente. Esto significa que se cambia el punto de abstracción del código fuente final. Por lo tanto, el modelo de diseño debe ser una descripción de cómo debe estructurarse, administrarse y escribirse el código fuente.

Alcance

En este documento se establecen límites en el proyecto KIUSH y se definen con precisión los objetos, plazos y entregables del proyecto. Este documento influye en los entregables ya que especifica los resultados tangibles que se producirán y entregarán durante el proyecto. También establece los plazos de tiempo que se espera que dure el proyecto o cuando se completaran ciertas tareas específicas. Además, este documento será utilizado por todos los participantes del proyecto para asegurar que todos partan desde la misma base o idea, y para evitar cualquier malentendido o confusión.

Definiciones, siglas y abreviaturas.

[Esta sección debe proporcionar las definiciones de todos los términos, las siglas, y abreviaturas requeridas para interpretar apropiadamente el documento Modelo de Diseño. Esta información puede proporcionarse por la referencia al Glosario del proyecto.]

Referencias

Especificaciones de requerimientos – 22/9/2023 – Yield Yelders

Modelo de Casos de Uso – 22/9/2023 – Yield Yelders

Visión general

La visión general del documento modelo de diseño proporciona una descripción del contenido y la organización del documento. Este documento contiene detalles sobre el diseño, la arquitectura, los componentes y las funcionalidades del sistema o producto que se está diseñando. También abarca los objetivos del diseño, las restricciones y los criterios de éxito. Este documento influye en las decisiones de diseño y desarrollo, y puede afectar a los desarrolladores, los gestores de proyectos, los stakeholders y otros miembros del equipo del proyecto.

Diseño de Casos de Uso

Diseño del Caso de Uso: CU9 Modificar curso

Diagrama de paquetes

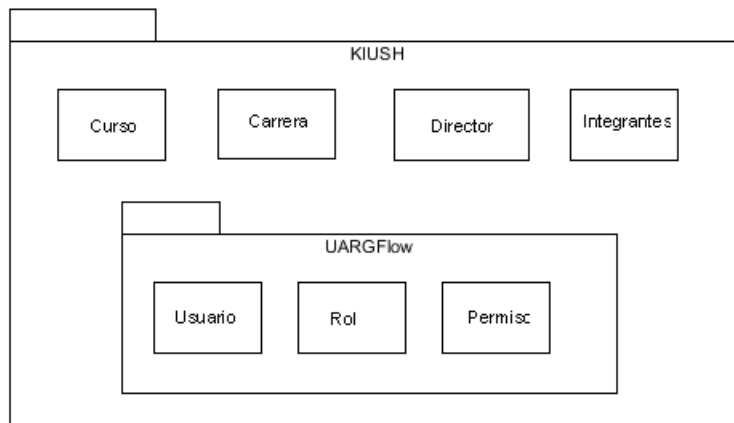
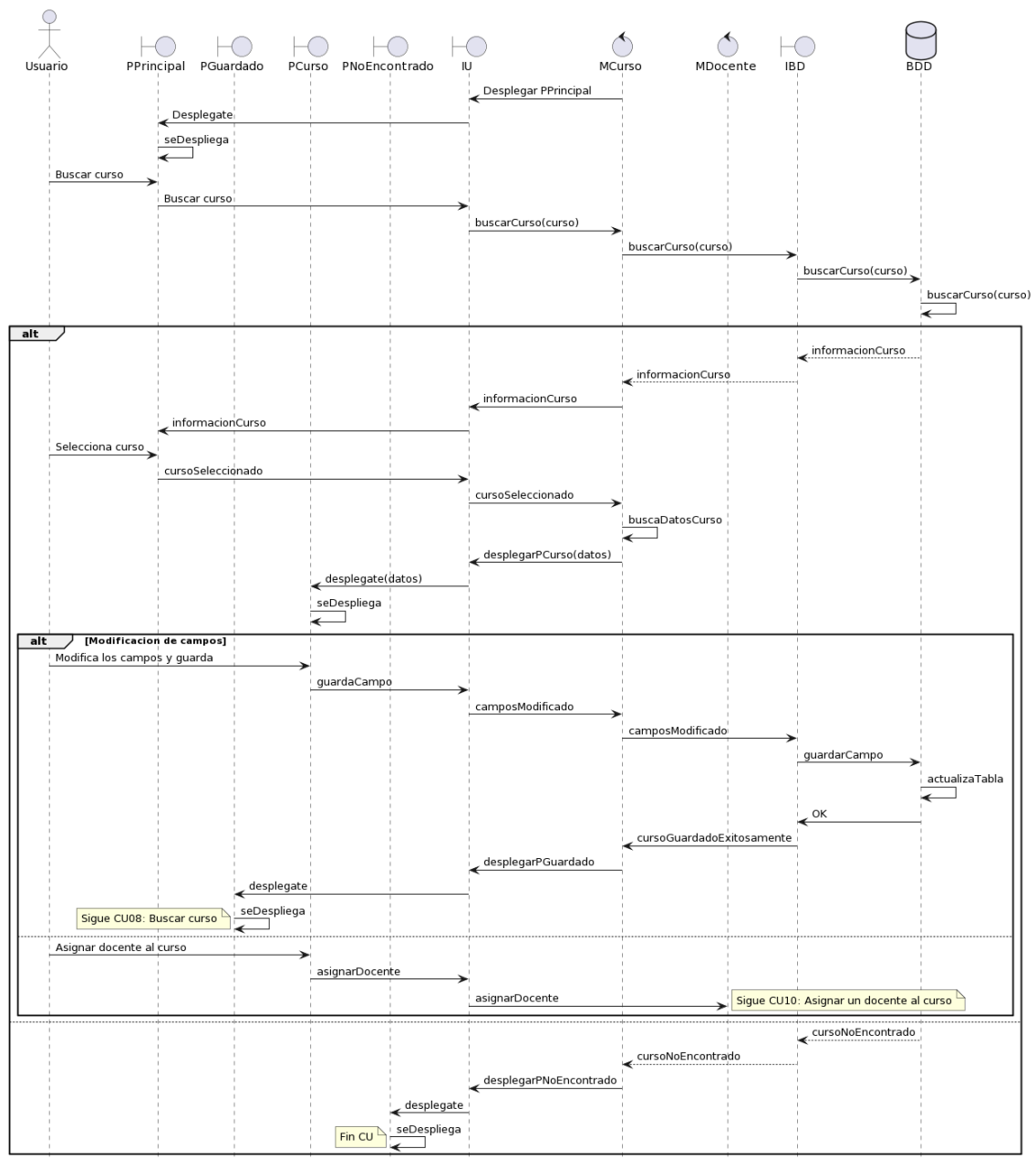


Diagrama de Interacción



Diseño de Flujo de eventos

1. El usuario ingresa en el buscador el curso.
2. El sistema busca si existe el curso ingresado, en la BD.
3. La BD válida si existe el curso
4. La BD encuentra el curso, le devuelve la información al sistema.
5. El sistema le muestra al usuario los cursos a modificar.
6. El usuario selecciona el curso a modificar.
7. El sistema le muestra los campos con los datos que desea modificar.
8. El usuario modifica el campo que desea cambiar.
9. El usuario selecciona la acción guardar
10. El sistema actualiza los cambios en la BD
11. La BD actualiza la tabla de cursos.

12. La BD le avisa que guardó el curso modificado al sistema
13. El sistema despliega una mensaje que se guardó exitosamente

Requerimientos especiales o de implementación

Tener el curso cargado

Tener el rol de gestionar cursos

Diseño del Caso de Uso: CU10 Asignar Docente

Diagrama de paquetes

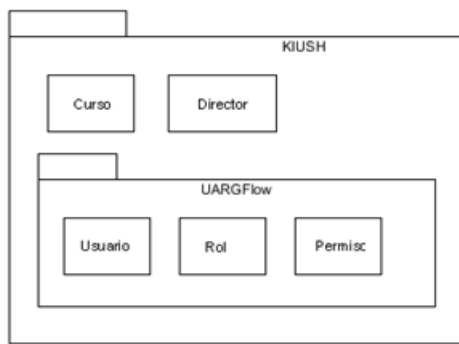
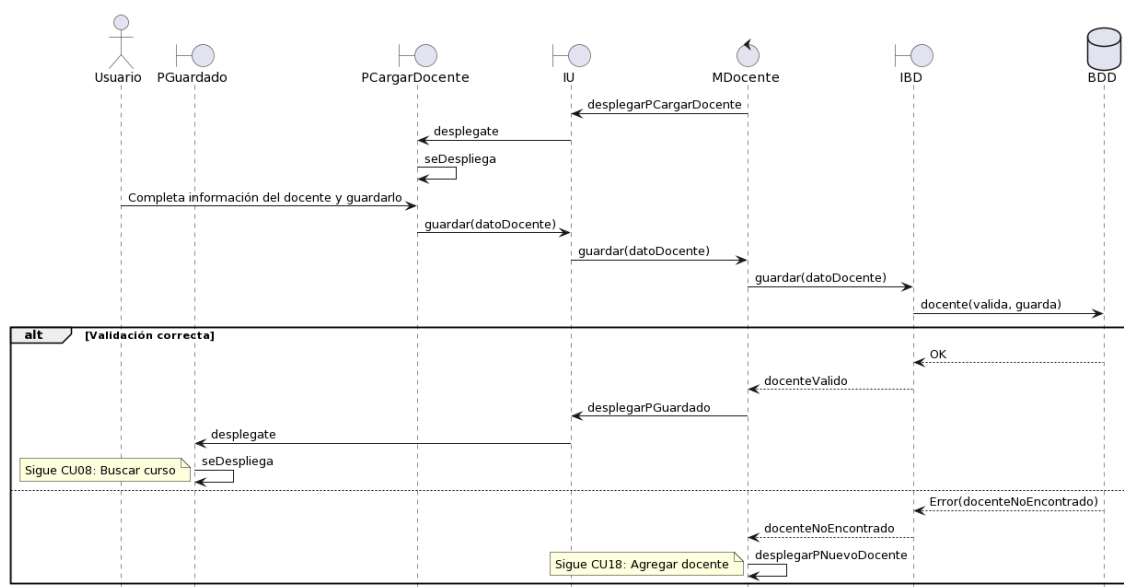


Diagrama de Interacción



Diseño de Flujo de eventos

1. El sistema muestra al usuario una pantalla donde cargar datos del docente

2. El usuario completa el nombre del docente a cargo del curso
3. El usuario selecciona guardar los cambios del curso
4. El sistema guarda los cambios en la BD.
5. La BD válida y guarda al docente.
6. El sistema despliega un mensaje de guardado los cambios

Requerimientos especiales o de implementación

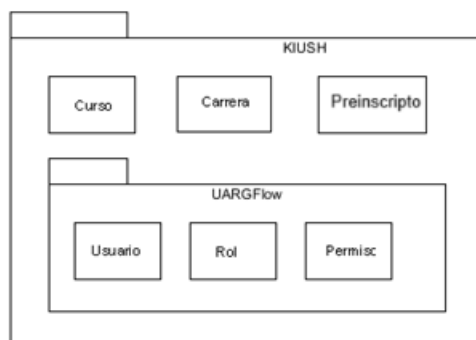
Tener el curso cargado

Tener el rol de gestionar cursos

Luego de crear un curso o modificarlo, se le puede asignar un docente o profesional encargado de dictar el curso, este puede ser uno o varios, ya que se registra todo el equipo de desarrollo.

Diseño del Caso de Uso: CU16 Modificar Preinscripto

Diagrama de paquetes



9. La PantallaCargarPreinscripto envía los datos a la InterfazUsuario
10. La InterfazUsuario envía los datos al ManejadorPreinscripto
11. El ManejadorPreinscripto solicita actualizar los datos a la BDKIUSH
12. La BDKIUSH actualiza los datos exitosamente
13. El ManejadorPreinscripto despliega el mensaje “guardado exitosamente” en la InterfazUsuario
14. La InterfazUsuario despliega el mensaje “guardado exitosamente” en la PantallaCargarPreinscripto.

Requerimientos especiales o de implementación

Precondición 1: el usuario administrativo debe iniciar sesión con un correo de Gmail valido, y que se encuentre en la base de datos del sistema con el rol y permisos necesarios para la modificación de un preinscripto.

Precondición 2: Tener al menos un curso que esté o haya estado habilitado para poder cargar preinscriptos

Precondición 3: Buscar el curso al cual se encuentra preinscripto el interesado y que se encuentre en la base de datos.

Precondición 4: una vez que se encuentra el curso, buscar al inscripto del cual se desea modificar sus datos con su número de DNI o nombre y apellido y que se encuentre en la base de datos.

Diseño de Objetos

Curso

- Descripción: poseer las características de un curso, como el nombre, la fecha, la descripción del curso, etc.
- Objetivo: poder registrar diferentes cursos con las mismas propiedades
- Atributos: nombre, fechas, lugar, estado, descripción
- Métodos: `getNombre()`, `setNombre(nombre)`, `getDescripcion()`, `setDescripcion(descripcion)`, `getEstado()`, `cambiarEstado (estado)`, `getFecha()`, `setFecha(fecha)`, `getLugar()`, `setLugar(sitio)`
- Interacciones: clase usuario, clase director, clase preinscripto

Preinscripto

- Descripción:
- Objetivo:
- Atributos: nombre, apellido, email, dni, estado, nota, asistencia
- Métodos: `getNombre()`, `setNombre(nombre)`, `getApellido()`, `setApellido(nombre)`, `getEmail()`, `setEmail(email)`, `getDni()`, `setDni()`, `getEstado()`, `cambiarEstado (estado)`, `setNota(nota)`, `getNota()`, `setAsistencia()`, `getAsistencia()`.
- Interacciones: clase carrera, clase tipo, clase curso

Diseño de Subsistemas

Subsistemas Específicos

UARGFlow

Propósito

Es brindar una plataforma web intuitiva como base para poder ser aplicada en diferentes temáticas asignadas por la asignatura Laboratorio de Proyectos de Software. Además de poder ser expandible, dando más funcionalidades al nuevo sistema y la definición del modelo de datos.

Función

Poder lograr registrar y darle los permisos correspondientes a los usuarios que accederán a la plataforma web, con sus correspondientes vistas y acciones para poder llevar a cabo la realización de sus tareas diarias.

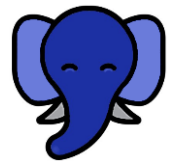
Subordinados

Los subsistemas que se lograron a detectar son la utilización de una API de Google, para poder ingresar con una cuenta de Gmail, cuando es correcta la validación, se logra devolver el nombre de la cuenta y el Gmail.

Dependencias

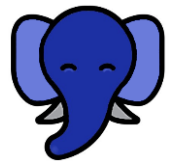
Se relaciona por medio de una cuenta de Gmail, para poder ingresar el sistema UARGFlow como requisito fundamental, y una vez obtenida la cuenta, se logra definir el rol dentro del sistema de la Base de Datos de UARGFlow.

Subsistema del que depende	Naturaleza de interacción	Características
Api de Google	Para poder iniciar sesión con Gmail	Una vez validado, se logra obtener el nombre y el Gmail, para validar los permisos en la base de datos UARGFlow



Recursos

La cantidad de tiempo de uso es momentáneo, porque una vez obtenido la validación del Gmail, se logra obtener el nombre del usuario y su cuenta. Que se utilizara para poder verificar si en la Base de Datos de UARGFlow está registrado con un rol distintivo, para poder brindar las acciones que desempeñara en su trabajo de día a día.



Diagramas

Diagrama de componentes

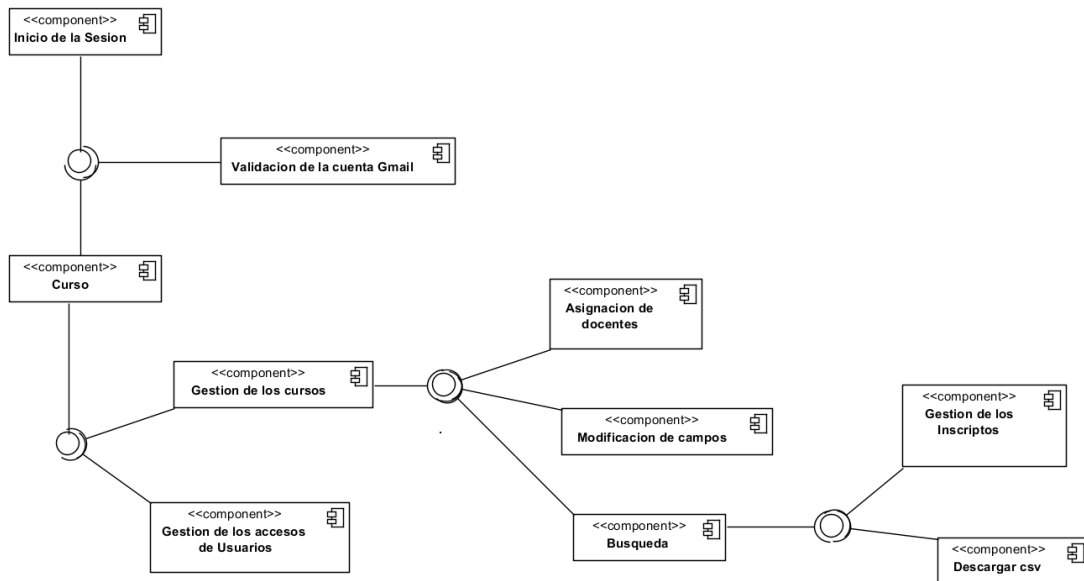


Diagrama de Clases

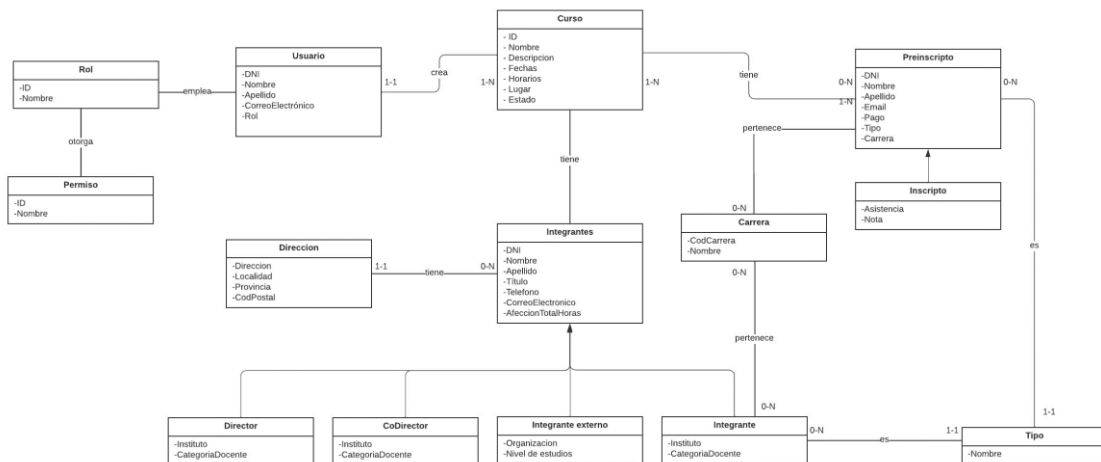


Diagrama de Secuencia

Diagramas de Paquetes

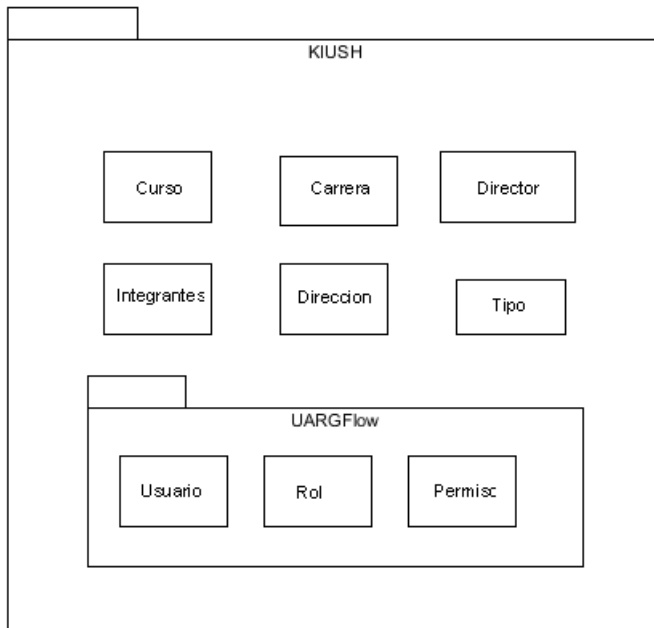


Diagrama de Colaboración

