

Modelo de Diseño

TEMPUS



TEMPUS

Gestión de Horarios de Cursada y Mesas de Examen

El propósito del Modelo de Diseño es empezar a realizar los casos de usodesarrollados durante las etapas anteriores en especial la etapa de captura de requerimientos. Es decir, tomar el Modelo deCasos 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	6
<i>Propósito</i>	<i>6</i>
<i>Alcance</i>	<i>6</i>
<i>Definiciones, siglas y abreviaturas.</i>	<i>6</i>
Referencias.....	6
<i>Visión general.....</i>	<i>7</i>
Diseño de Casos de Uso	7
<i>Diseño del CU01 –Ingresar al sistema</i>	<i>7</i>
Diagrama de paquetes.....	7
Diagrama de Interacción.....	7
Diseño de Flujo de eventos.....	7
Requerimientos especiales o de implementación	7
<i>Diseño del CU02 – Importar horarios de cursada.....</i>	<i>8</i>
Diagrama de paquetes.....	8
Diagrama de Interacción.....	8
Diseño de Flujo de eventos.....	9
Requerimientos especiales o de implementación	9
<i>Diseño del CU03 – Importar mesas de examen.....</i>	<i>10</i>
Diagrama de paquetes.....	10
Diagrama de interacción.....	10
Diseño de flujo de eventos	11
Requerimientos especiales o de implementación	11
<i>Diseño del CU04 – Buscar horarios de cursada</i>	<i>12</i>
Diagrama de paquetes.....	12
Diagrama de interacción.....	12
Diseño de flujo de eventos	13
Requerimientos especiales o de implementación	13
<i>Diseño del CU05 – Buscar mesas de examen</i>	<i>13</i>
Diagrama de paquetes.....	13
Diagrama de interacción.....	14
Diseño de flujo de eventos	14
Requerimientos especiales o de implementación	14
Diseño de Objetos	14
<i>Aula</i>	<i>14</i>
Descripción	14
Atributos	14
Métodos.....	15

<i>Aulas</i>	15
Descripción	15
Atributos	15
Métodos.....	15
<i>Asignatura</i>	16
Descripción	16
Atributos	16
Métodos.....	16
<i>Carrera</i>	17
Descripción	17
Atributos	17
Métodos.....	17
<i>Clase</i>	18
Descripción	18
Atributos	18
Métodos.....	18
<i>Cursada</i>	19
Descripción	19
Atributos	19
Métodos.....	19
<i>Cursadas</i>	20
Descripción	20
Atributos	20
Métodos.....	20
<i>Docente</i>	20
Descripción	20
Atributos	21
Métodos.....	21
<i>Llamado</i>	21
Descripción	21
Atributos	22
Métodos.....	22
<i>Mesa de Examen</i>	22
Descripción	22
Atributos	23
Métodos.....	23
<i>Mesas</i>	24
Descripción	24
Atributos	24
Métodos.....	24
<i>Plan25</i>	
Descripción	25
Atributos	25
Métodos.....	25

<i>Usuario</i>	<i>26</i>
Descripción	26
Atributos	26
Métodos.....	26
Diseño de Subsistemas	27
<i>Subsistema de Aulas.....</i>	<i>27</i>
Propósito.....	27
Función	27
Subordinados	27
Dependencias	28
Recursos.....	28
<i>Subsistema de Carreras.....</i>	<i>29</i>
Propósito.....	29
Función	29
Subordinados	29
Recursos.....	30
<i>Subsistema de Cursadas.....</i>	<i>30</i>
Propósito.....	30
Función	30
Subordinados	31
Recursos.....	31
<i>Subsistema de Mesas</i>	<i>32</i>
Propósito.....	32
Función	32
Subordinados	32
Recursos.....	33
<i>Subsistema de Usuarios</i>	<i>33</i>
Propósito.....	33
Función	33
Subordinados	34
Diagramas	36
<i>Diagrama de componentes.....</i>	<i>36</i>
<i>Diagrama de Clases.....</i>	<i>36</i>
<i>Diagramas de Paquetes</i>	<i>38</i>

Modelo de Diseño

Introducción

El Modelo de diseño es un modelo de objetos que va describir la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tiene impacto en el sistema.

Propósito

El propósito del Modelo de Diseño es manifestar el diseño del sistema. Se puede decir que presenta una “programación visual” que debe ser mantenido durante todo el ciclo de vida del software.

Alcance

En concreto, los propósitos del diseño son:

- Adquirir una comprensión en profundidad de los aspectos relacionados con los requerimientos funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables, sistemas operativos, tecnologías de distribución y concurrencia, tecnologías de interfaz de usuario, etc.
- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes capturando los requerimientos o subsistemas individuales, interfaces y clases.
- Ser capaz de descomponer los trabajos de implementación en partes más manejables que puedan ser llevadas a cabo por diferentes partes del equipo.

Definiciones, siglas y abreviaturas.

Las definiciones, siglas y abreviaturas se pueden observar detalladamente en el Glosario.

Referencias

El Modelo de Diseño se relaciona con:

- Especificación de requerimientos de software.
- Arquitectura del sistema.
- Modelo de Datos.
- Especificaciones de Casos de Uso.

Visión general

En principio el documento inicia con el diseño de los casos de uso que se han especificado en los documentos Especificación de Caso de Uso.

En un segundo punto se realiza el diseño de Objetos y finalmente se realiza el diseño de los subsistemas identificados.

En este documento se presenta el diagrama de clases obtenido.

Diseño de Casos de Uso

Diseño del CU01 -Ingresar al sistema

Diagrama de paquetes

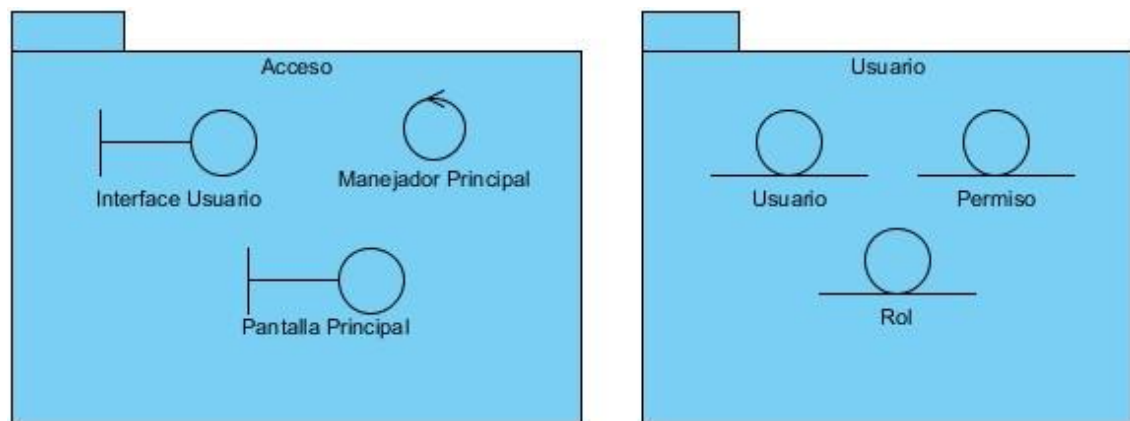


Diagrama de Interacción

No aplica

Diseño de Flujo de eventos

No aplica

Requerimientos especiales o de implementación

No aplica

Diseño del CU02 – Importar horarios de cursada

Diagrama de paquetes

En esta sección se identifican los objetos y subsistemas de diseño que intervienen en el caso de uso y sus relaciones.

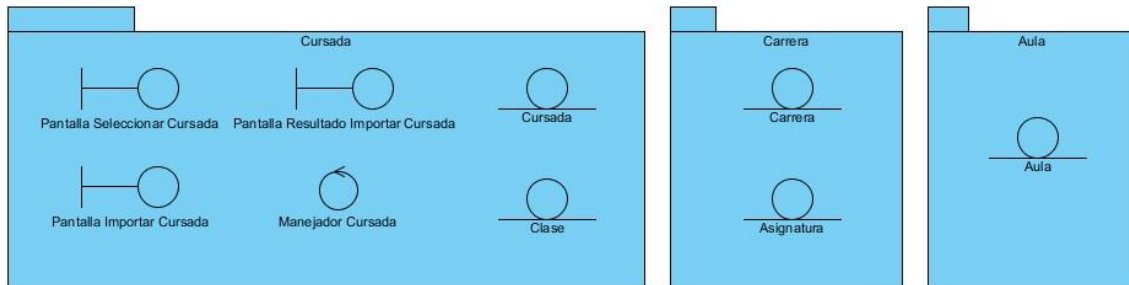
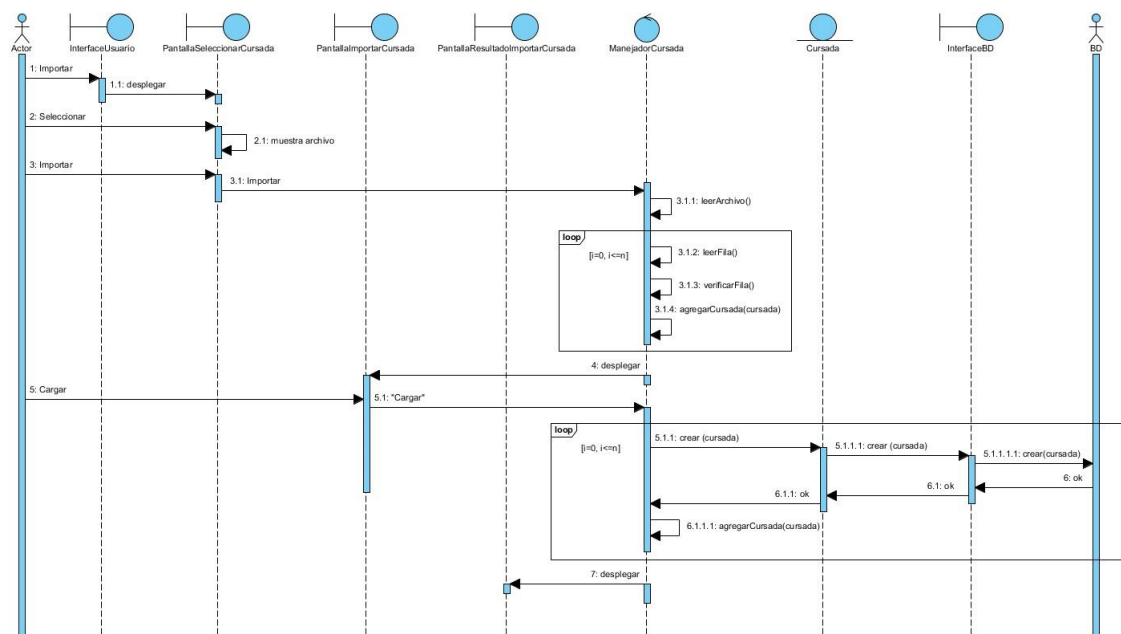
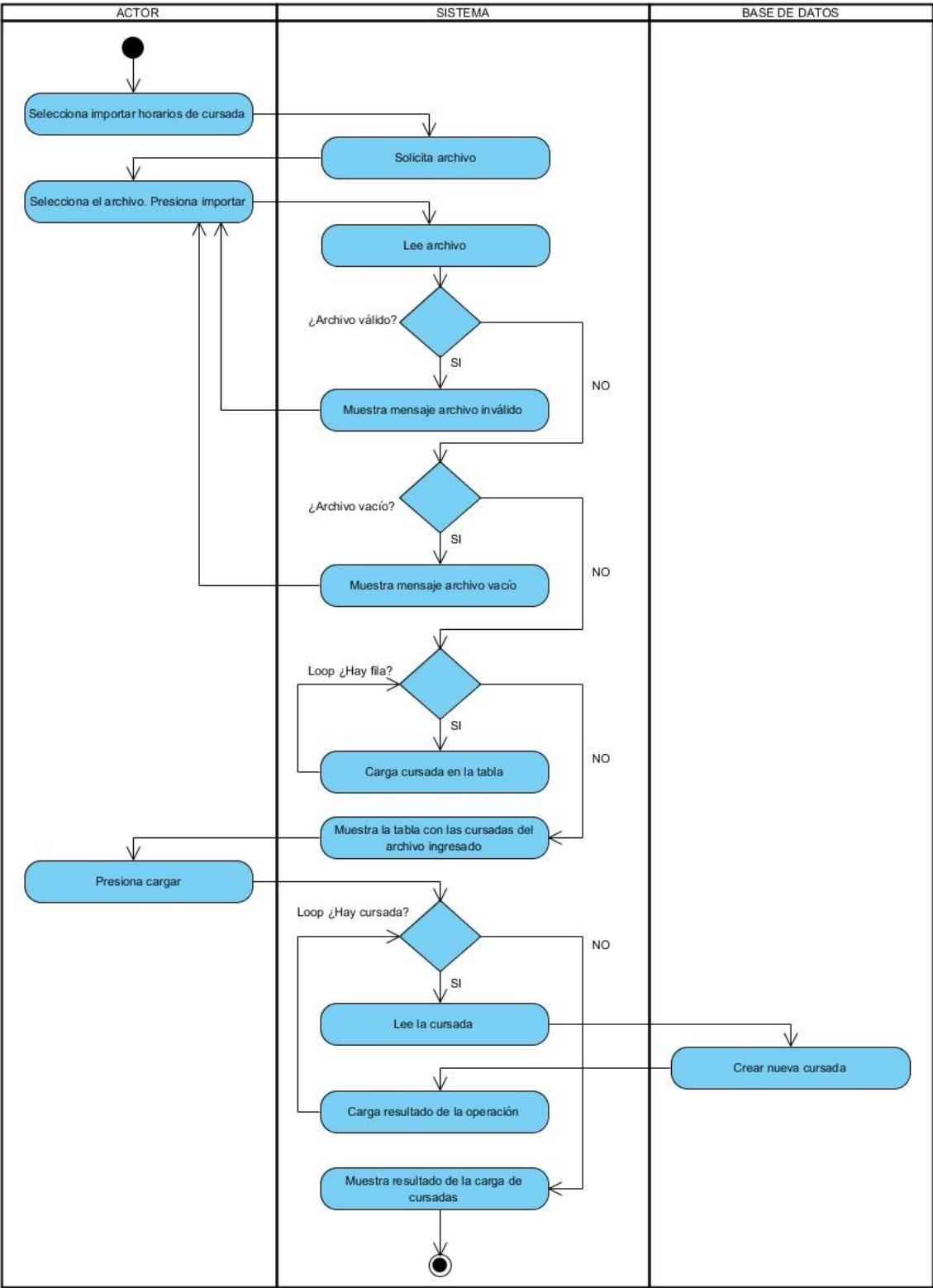


Diagrama de Interacción

En esta sección se utilizan los objetos detectados para indicar su interacción. El detalle de cada uno de los pasos que se realiza en el siguiente diagrama de secuencia se encuentra en el documento Especificación CU02 - Importar horarios de cursada. Lo que se visualiza en el diagrama corresponde al flujo principal del caso de uso.



Diseño de Flujo de eventos



Requerimientos especiales o de implementación

Este caso de uso representa una de las funcionalidades más importantes del sistema. Por lo tanto el procesamiento del archivo con los horarios de cursada debe cumplir con

requerimientos de carga y de rendimiento. Debe ser capaz de realizar la carga de grandes volúmenes de información y hacer el procesamiento en un tiempo razonable.

Con este objetivo, se divide la importación en dos partes: Una para leer, detectar y mostrar errores en el archivo, y otra parte donde se lleva a cabo la carga en la base de datos.

Diseño del CU03 – Importar mesas de examen

Diagrama de paquetes

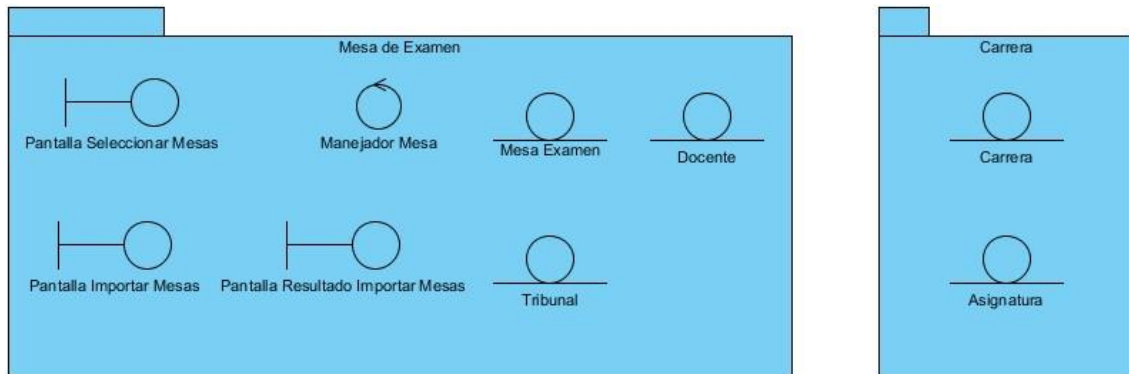
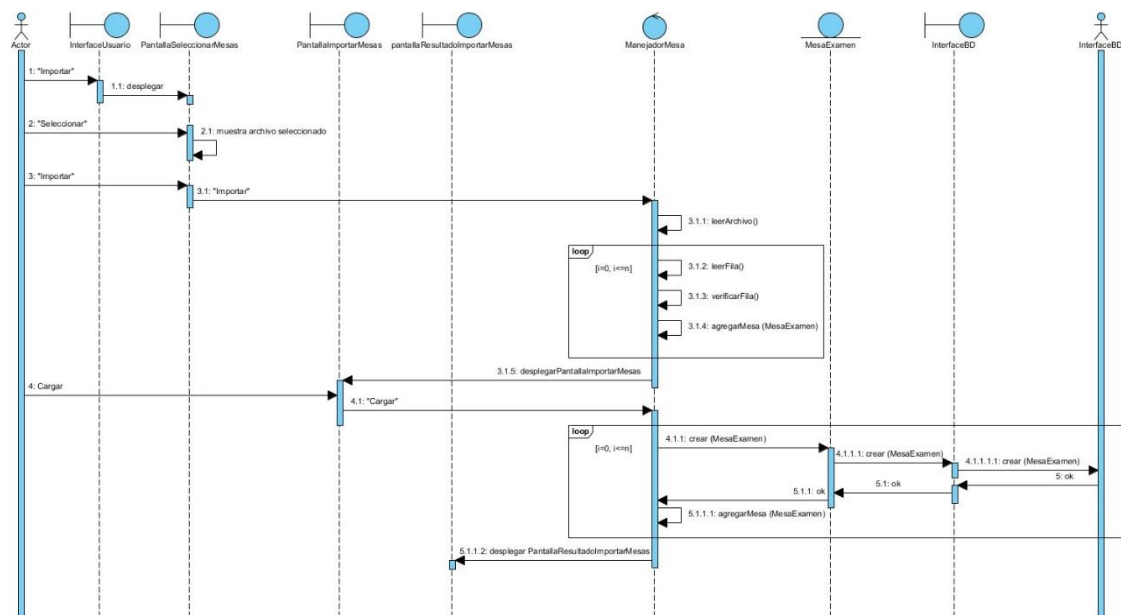
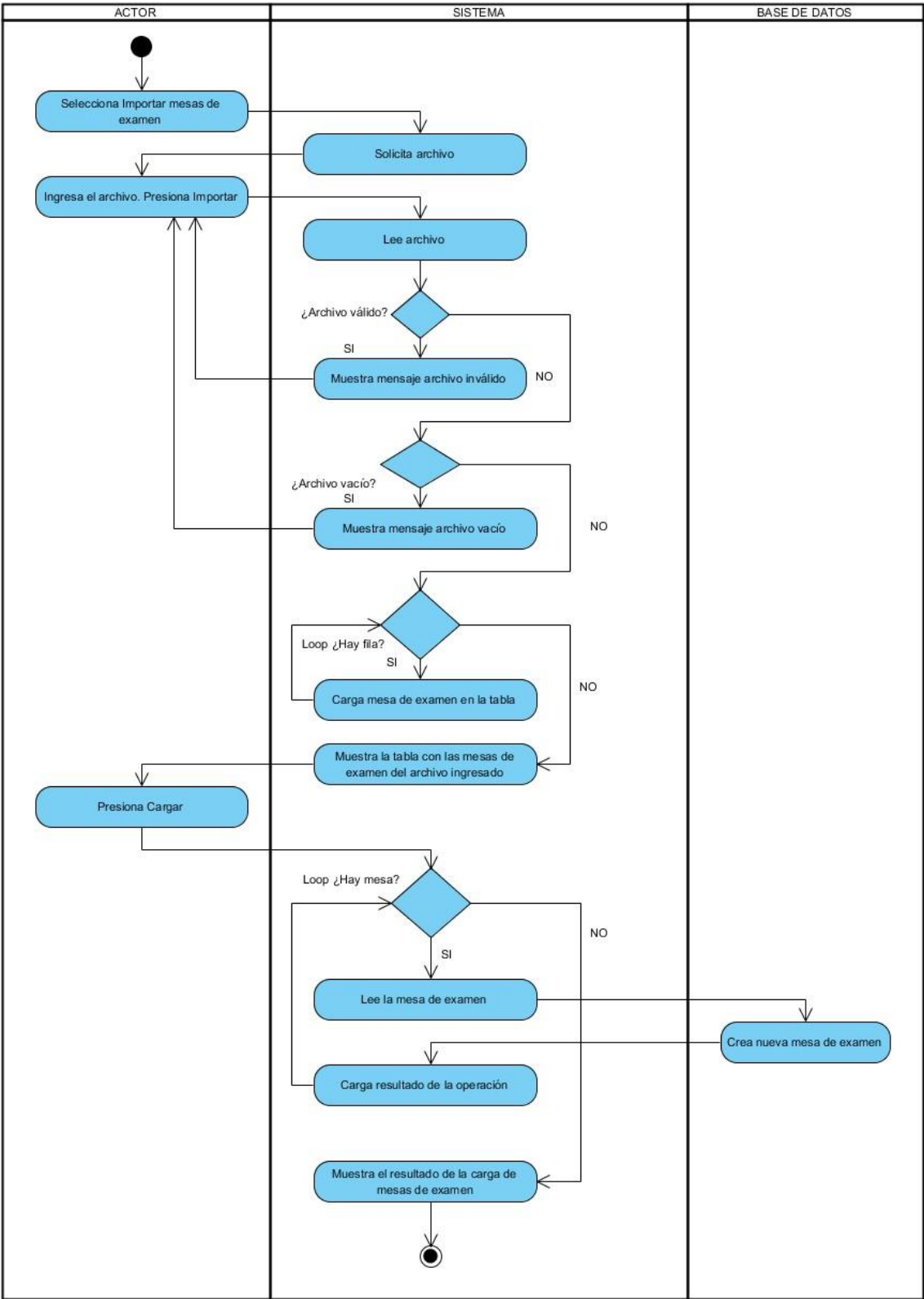


Diagrama de interacción

El detalle de cada uno de los pasos que se realiza en el siguiente diagrama de secuencia se encuentra en el documento Especificación CU03 - Importar mesas de examen. Lo que se visualiza en el diagrama corresponde al flujo principal del caso de uso.



Diseño de flujo de eventos



Requerimientos especiales o de implementación

Este caso de uso representa una de las funcionalidades más importantes del sistema junto con Importar horarios de cursada. Por lo tanto el procesamiento del archivo con las mesas de

examen debe cumplir con requerimientos de carga y de rendimiento. Debe ser capaz de realizar la carga de grandes volúmenes de información y hacer el procesamiento en un tiempo razonable.

Con este objetivo, se divide la importación en dos partes: Una para leer, detectar y mostrar errores en el archivo, y otra parte donde se lleva a cabo la carga en la base de datos.

Diseño del CU04 – Buscar horarios de cursada

Diagrama de paquetes

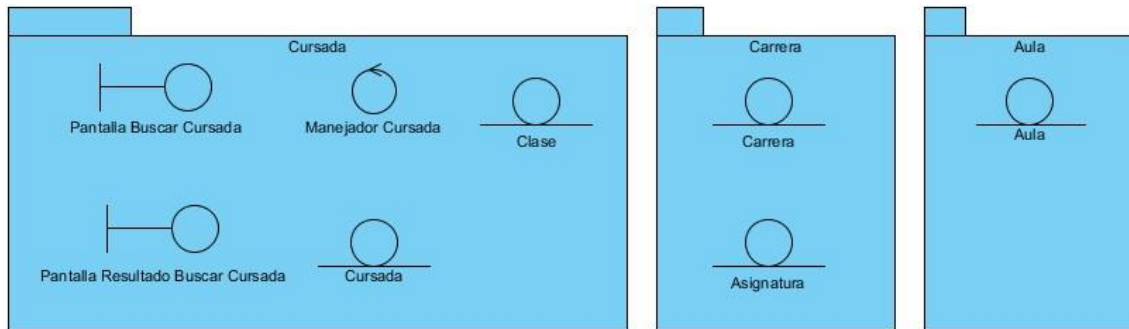
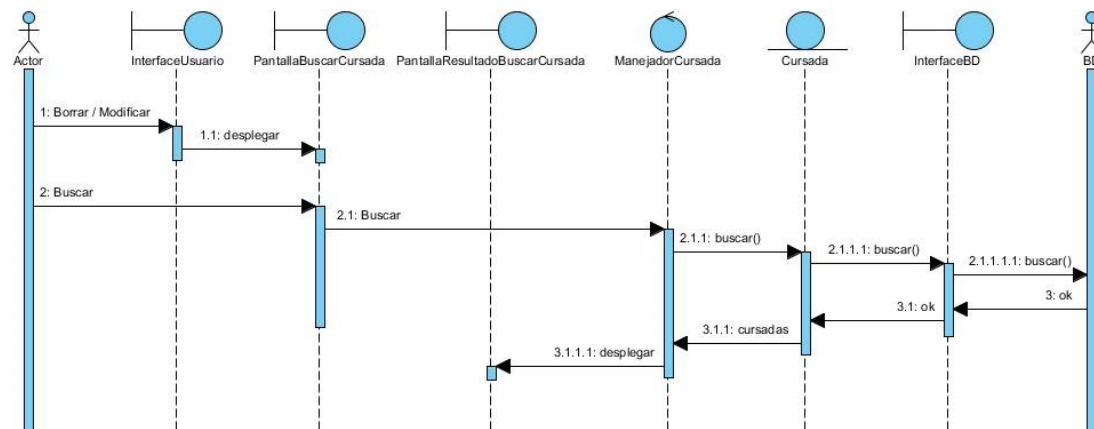
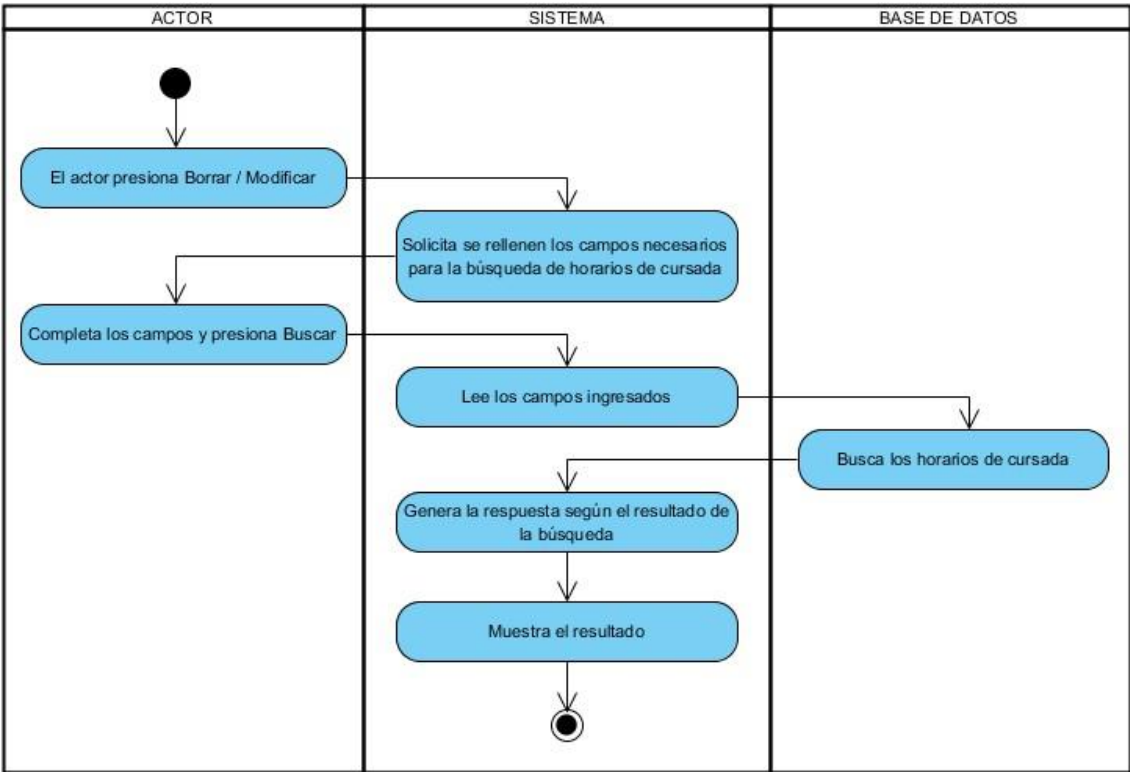


Diagrama de interacción



Diseño de flujo de eventos



Requerimientos especiales o de implementación

Este caso de uso es accedido desde una aplicación móvil y desde una página web. En el caso de la aplicación móvil, el acceso es público mientras que el acceso desde la página web debe ser a través del inicio de sesión.

En el caso de la aplicación móvil, se debe buscar una forma de representar tablas grandes en consideración del tamaño de las pantallas en los dispositivos móviles.

Diseño del CU05 – Buscar mesas de examen

Diagrama de paquetes

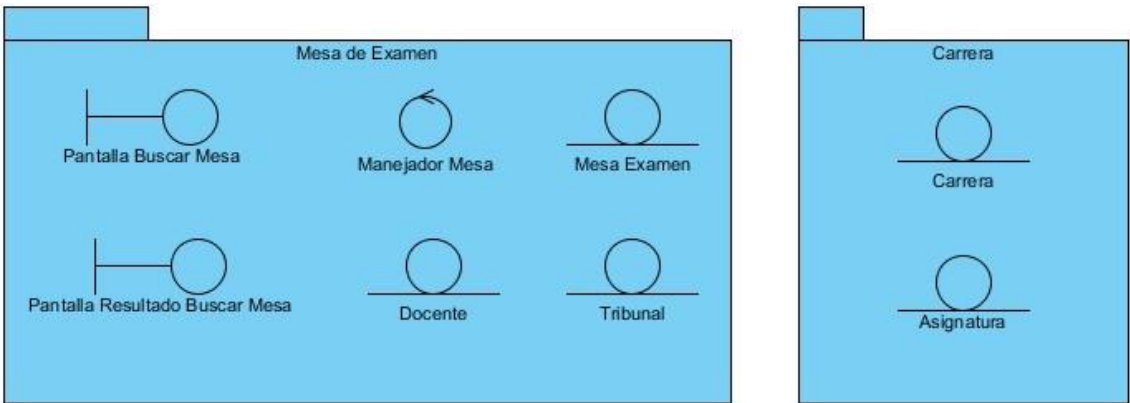
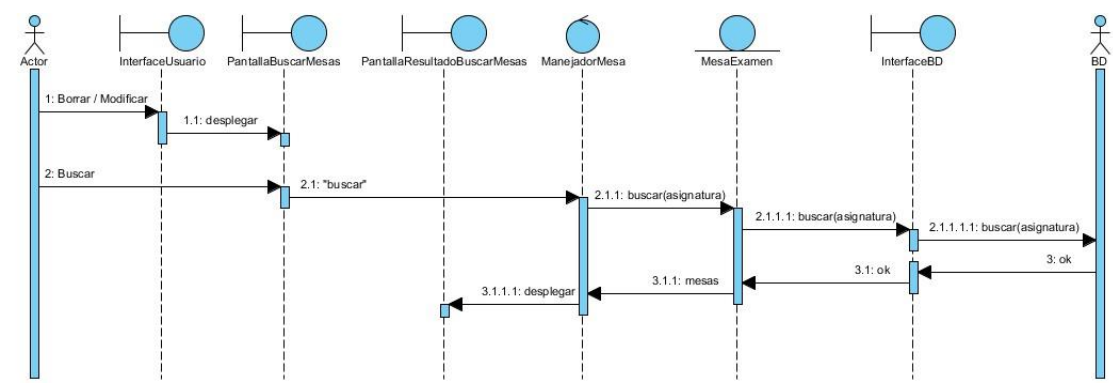


Diagrama de interacción



Diseño de flujo de eventos

Requerimientos especiales o de implementación

Este caso de uso es accedido desde una aplicación móvil y desde una página web. En el caso de la aplicación móvil, el acceso es público mientras que el acceso desde la página web debe ser a través del inicio de sesión.

En el caso de la aplicación móvil, se debe buscar una forma de representar tablas grandes en consideración del tamaño de las pantallas en los dispositivos móviles.

Diseño de Objetos

Aula

Descripción

Este objeto representa a un aula donde se dictan clases y se desarrollan las mesas de examen. Este objeto se corresponde con la tabla aula de la base de datos. Contiene los mismos atributos.

Atributos

Nombre	Descripción
Identificador	Corresponde a la clave primaria de la tabla “Aula” en la base de datos del sistema. Es un atributo numérico privado.
Nombre	Nombre de aula. Es un atributo privado de texto.
Sector	Sector donde se ubica el aula. Es un atributo privado de texto.
Datos	Este atributo se utiliza para obtener los resultados que se realizan con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene un Aula. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Constructor	Método constructor de clase. Se utiliza para crear un nuevo objeto. Este método puede recibir el identificador del aula para obtener su información de la base de datos. En caso contrario, se crea con los atributos nulos.
Crear	Se crea una nueva aula y se almacena en la base de datos. Se requiere el sector y numero de la nueva aula. Para ello se debe verificar que la misma no exista.
Borrar	Se elimina una determinada aula de la base de datos. Para ello se requiere el identificador de la misma.
Buscar	Se busca la información de un aula. En caso que exista se actualizan los atributos del objeto con la información correspondiente. En caso contrario se dejan los atributos en nulo.
Modificar	Se modifica la información correspondiente al aula en la base de datos. Se debe recibir el identificador del aula y la nueva información.
Obtener horarios	Se utiliza para obtener los horarios de clase que contiene el aula especificada.

Aulas

Descripción

Este se utiliza para manejar un conjunto de aulas. Se pueden realizar operaciones sobre un grupo de aulas, es por ello que se diferencia de aula.

Atributos

Nombre	Descripción
Aulas	Un arreglo de una o más aulas. Este atributo es privado.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene aulas. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor que no contiene parámetros de entrada. Sirve para crear y utilizar el objeto.
Listar disponibles	Consulta en la base de datos todas las aulas que se encuentren disponibles. Se puede indicar el sector y la franja horaria para consultar (desde-hasta).

Asignatura

Descripción

Este objeto representa a una asignatura. Se corresponde a la asignatura de la base de datos.

Atributos

Nombre	Descripción
Identificador	Identificador de la clase. Corresponde con la clave primaria de la tabla "Asignatura" en la base de datos. Es un atributo numérico y privado.
Nombre	Nombre de la asignatura. Es un atributo de texto privado que debe ser accedido a través de los métodos públicos.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene una Asignatura. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Constructor	Contiene el constructor de la clase asignatura. Se puede indicar el identificador de asignatura para obtener la información desde la base de datos. En caso que no se indique, se crea el objeto con los atributos nulos.
Crear	Se crea una nueva asignatura y se almacena en la base de datos. Se requiere el nombre de la nueva asignatura. Para ello se debe verificar que la misma no exista.
Borrar	Se elimina una determinada asignatura de la base de datos. Para ello se debe recibir el identificador de la asignatura.
Buscar	Se busca y obtienen los datos de una determinada asignatura. Cuando se encuentra la información se actualizan los atributos del objeto.

	En caso contrario se colocan los atributos nulos.
Modificar	Se modifica la información correspondiente a la asignatura en la base de datos. Se requiere del identificador de asignatura y la nueva información.

Carrera

Descripción

Este objeto representa a una carrera. Una carrera contiene una o muchas asignaturas dependiendo el año. Este objeto contendrá la lógica necesaria para operar con la base de datos según los métodos correspondientes.

Atributos

Nombre	Descripción
Código	Código de la carrera. Es un atributo numérico privado que debe ser accedido a través de los métodos públicos.
Nombre	Nombre de la carrera. Es un atributo de texto privado que debe ser accedido a través de los métodos públicos.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene una Carrera. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Constructor	Constructor de clase. Se puede recibir el identificador de carrera para obtener la información desde la base de datos. En caso que no se indique, se crea el objeto con los atributos nulos.
Crear	Se crea una nueva carrera y se almacena en la base de datos. Se requiere el código y nombre de la carrera nueva. Para realizar esto, se realiza una verificación para establecer que la carrera no exista.
Borrar	Se elimina una determinada carrera de la base de datos. Para ello se requiere saber el código de la carrera a eliminar.
Buscar	Se busca y obtienen los datos de una determinada carrera. Para realizar esta búsqueda se utiliza el nombre de carrera. En todo caso, siempre se devuelve un solo objeto.
Modificar	Se modifica la información correspondiente a la carrera en la base de datos. Se requiere saber el código de la carrera y su nueva información (Nombre).

Clase

Descripción

Una clase se relaciona con asignaturas y carreras. Una clase corresponde a una asignatura dentro de una determinada carrera. Este objeto contiene los atributos correspondientes a la tabla Clase de la base de datos. Además, tiene los métodos necesarios para crear, borrar, modificar y buscar una determinada clase.

Atributos

Nombre	Descripción
Identificador	Identificador de la clase. Se corresponde con la clave primaria en la base de datos. Es un atributo numérico.
Día	Atributo numérico privado que corresponde al día de la semana en el que se dicta la clase. Los valores posibles son 1, 2, 3, 4, 5 o 6.
Desde	Atributo de texto privado que representa el horario desde que comienza la clase. Tiene el formato HH:MM
Hasta	Atributo de texto privado que representa el horario a la que finaliza la clase. Tiene el formato HH:MM
Aula	Un atributo de tipo Aula.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene una Clase. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Se puede recibir un identificador de clase con el cual se obtiene la información desde la base de datos. En caso que no se indique, los atributos serán nulos cuando se cree el objeto.
Crear	Crea una nueva clase y la almacena en la base de datos. Se requiere saber el día, hora de inicio y hora de fin de la nueva clase. Además se debe establecer el aula, la cual debe existir en la base de datos.
Borrar	Se borra una determinada clase de la base de datos. Para ello se requiere saber el identificador de la clase que se va a eliminar.
Buscar	Se busca una clase en la base de datos. Para ello se debe recibir el día, hora de inicio, hora de fin y el aula. Si se encuentra un resultado, se actualizan los atributos. En caso contrario, serán nulos.
Modificar	Se modifica una clase en la base de datos. Se pueden modificar todos los campos a excepción del identificador.

Cursada

Descripción

Una cursada contiene las clases de una determinada asignatura para una carrera. Se pueden obtener todos los horarios de cursada para una asignatura o para una carrera completa. Este objeto permite establecer una comunicación con la base de datos y obtener toda la información necesaria a través de sus métodos.

Atributos

Nombre	Descripción
Plan	Atributo privado de tipo Plan. Este atributo es necesario para conocer a que asignatura, carrera y en qué año se dictan las clases que contiene una determinada cursada.
Clases	Un arreglo que contiene un conjunto de clases. Son el conjunto de clases que contiene una asignatura en una carrera para un determinado año.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene una Cursada. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor de cursada. Se crea el objeto con sus atributos nulos.
Crear	Método público que realiza la creación del conjunto de clases en la base de datos. Se debe recibir el plan al que pertenecen las clases. Es decir que debe estar cargado en la base de datos la asignatura y carrera.
Crear relación	Método privado que realiza la creación de un nuevo registro en la tabla cursada de la base de datos. Recibe el identificador de asignatura, carrera y clase.
Buscar	Método público que realiza la búsqueda del conjunto de clases perteneciente a una asignatura y/o carrera. Se debe recibir el identificador de la asignatura y/o carrera.
Buscar relación	Método privado que realiza la búsqueda de un registro en la tabla cursada de la base de datos. Esto se realiza para obtener la relación entre asignatura, carrera y clase.
Obtener horarios	Se obtienen todas las clases que contiene una asignatura en una carrera. Cuando se obtienen resultado se carga el arreglo de clases asociados al objeto.

Cursadas

Descripción

Este objeto contiene un conjunto de Cursadas con el cual trabajar. Se utiliza para realizar la creación, búsqueda y eliminación de múltiples cursadas.

Atributos

Nombre	Descripción
Cursadas	Atributo privado. Es un arreglo que contiene un conjunto de objetos de tipo Cursada.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

Nombre	Descripción
Construct	Constructor de cursada. Se crea el objeto con sus atributos nulos.
Crear	Método público que realiza la creación del conjunto de cursadas en la base de datos. Se recibe un arreglo con las cursadas que se van a crear.
Buscar	Método público que realiza la búsqueda del conjunto de cursadas a partir del nombre de una asignatura. Cuando se indica un nombre, se realiza la búsqueda de todas las cursadas similares al campo ingresado. En caso que no se indique, se buscan todas las cursadas cargadas.
Borrar	Método público que realiza la eliminación de todos los registros vinculados a la cursada. Por ello se hace la eliminación de los registros en la tabla clase y cursada de la base de datos. Cuando se realiza correctamente devuelve true, caso contrario devuelve false.

Docente

Descripción

Contiene la información de un docente. Este objeto se relaciona con tribunal para una determinada mesa de examen.

Atributos

Nombre	Descripción
Identificador	Atributo privado y numérico que se corresponde con la clave primaria de la tabla "Docente" en la base de datos.
Nombre	Nombre completo del docente. Es un atributo de texto privado.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene un Docente. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Se puede recibir un identificador de docente con el cual se obtiene la información desde la base de datos. En caso que no se indique, los atributos serán nulos cuando se cree el objeto.
Crear	Crea un nuevo docente y lo almacena en la base de datos. Para ello se debe recibir el nombre del docente.
Borrar	Se borra un determinado docente de la base de datos. Para ello se debe recibir el identificador del docente.
Buscar	Se busca un docente en la base de datos. Para ello se debe recibir el nombre del docente. Cuando se encuentra un resultado se actualizan los atributos, en caso contrario serán nulos. En todos los casos, siempre se obtiene un único resultado porque la búsqueda no es por aproximación (LIKE).
Modificar	Se modifica un docente en la base de datos. El único atributo modificable del docente es el nombre, por lo cual se debe enviar el identificador y el nuevo nombre.

Llamado

Descripción

Un Llamado tiene la información necesaria para saber la hora, día y lugar en el que se dicta una mesa de examen. Se puede realizar la creación, búsqueda, modificación y eliminación de registros en la tabla Llamado de la base de datos con este objeto.

Atributos

Nombre	Descripción
Identificador	Atributo privado y numérico que se corresponde con la clave primaria de la tabla "Docente" en la base de datos.
Fecha	Atributo de texto privado que corresponde a la fecha en que se dicta el llamado. Cuando se almacena en la base de datos el formato de fecha es AAAA/MM/DD.
Hora	Atributo de texto privado que corresponde a la hora en que se dicta el llamado. El formato es HH:MM.
Aula	Atributo privado de tipo Aula. Relaciona el llamado con el lugar donde se dicta la mesa de examen.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene un Llamado. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Se puede recibir un identificador de llamado con el cual se obtiene la información desde la base de datos. En caso que no se indique, los atributos serán nulos cuando se cree el objeto.
Crear	Crea un nuevo llamado y lo almacena en la base de datos. Para ello se debe recibir la fecha, hora y aula. El aula puede ser nula.
Borrar	Se borra un determinado llamado de la base de datos. Para ello se debe recibir el identificador del llamado.
Buscar	Se busca un llamado en la base de datos. Para ello se debe recibir la fecha, la hora y el aula. Cuando se encuentra un resultado, se actualizan los atributos del objeto. En caso contrario, serán todos nulos.
Modificar	Se modifica un llamado en la base de datos. Para ello se debe recibir el identificador del llamado a modificar junto con la nueva información asociada.

Mesa de Examen

Descripción

Una mesa de examen se relaciona con el objeto Plan, Tribunal y Llamado. Representa a la mesa de examen de una asignatura para una determinada carrera. La mesa de examen puede contener uno o dos llamados. Con este objeto se puede realizar la creación,

búsqueda, eliminación y modificación de registros en la tabla Mesa Examen de la base de datos.

Se debe destacar que la mesa de examen nunca puede contener ambos llamados en nulo cuando se crea en la base de datos. Es decir, alguno puede ser nulo pero no ambos a la vez.

Atributos

Nombre	Descripción
Identificador	Atributo privado y numérico que se corresponde con la clave primaria de la tabla "Mesa Examen" en la base de datos.
Plan	Atributo privado de tipo Plan. Se utiliza para relacionar la mesa de examen con una asignatura y carrera.
Tribunal	Atributo privado de tipo Tribunal. Se utiliza para obtener y operar sobre el tribunal asociado a la mesa de examen.
Primero	Atributo privado de tipo Llamado. Se utiliza para conocer y operar sobre la información del primer llamado asociado a la mesa de examen.
Segundo	Atributo privado de tipo Llamado. Se utiliza para conocer y operar sobre la información del segundo llamado asociado a la mesa de examen.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene una Mesa de Examen. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Se puede recibir un identificador de mesa con el cual se obtiene la información desde la base de datos. En caso que no se indique, los atributos serán nulos cuando se cree el objeto.
Crear	Crea una nueva mesa de examen y lo almacena en la base de datos. Para ello se debe recibir el plan, tribunal, primer y segundo llamado. Antes de realizar la creación de la mesa de examen se realiza una búsqueda para controlar que no exista previamente.
Borrar	Se borra una determinada mesa de examen de la base de datos. Para ello se debe recibir el identificador de la mesa de examen a eliminar.
Buscar	Se busca una mesa de examen en la base de datos. Para ello se debe recibir el plan al que pertenecería la mesa. Cuando se obtiene un resultado, se actualizan los atributos del objeto, en caso contrario los mismos serán nulos.
Modificar	Se modifica una mesa de examen en la base de datos. Para ello se debe recibir el identificador de la mesa a modificar junto con la nueva información asociada.

Mesas

Descripción

Este objeto contiene un conjunto de Mesas de Examen con el cual trabajar. Se utiliza para realizar la creación, búsqueda y eliminación de múltiples mesas de examen.

Atributos

Nombre	Descripción
Mesas	Atributo privado. Contiene un arreglo de Mesas Examen.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene el objeto Mesas. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Realiza la creación del objeto para comenzar a trabajar con un conjunto de mesas de examen.
Crear	Crea un conjunto de mesas de examen en la base de datos. Recibe un arreglo con el conjunto de mesas de examen que se han de crear. Con este método se realiza la eliminación de todas las mesas que existan, para posteriormente recorrer el arreglo recibido y crear cada una de las mesas de examen.
Borrar	Se borra toda la información de mesas de examen que exista en la base de datos. Para ello se elimina la información que contiene la tabla docente, tribunal, llamado y mesa de examen. Cuando la eliminación es correcta se devuelve true. En caso contrario se devuelve false.
Buscar	Se busca un conjunto de mesas de examen a partir del nombre de una asignatura. Cuando se indica un nombre de asignatura se realiza la búsqueda por aproximación (similares). En caso que no se indique un nombre de asignatura, se realiza la búsqueda de todas las mesas cargadas en la base de datos.
Cantidad llamados	Este método se utiliza para conocer la cantidad de llamados que tienen las mesas de examen. Es necesario para saber la cantidad de columnas que se deben mostrar en una tabla durante la búsqueda. Dado que no se conoce el número de llamados de un turno, este método cuenta la cantidad de mesas que tienen dos llamados. Devuelve el número de mesas que tiene dos llamados.

Plan

Descripción

Este objeto representa la relación entre asignatura y carrera. Además, contiene el atributo año necesario para establecer en que año de la carrera se encuentra la asignatura. Este objeto se relaciona con las cursadas y mesas de examen.

Atributos

Nombre	Descripción
Asignatura	Atributo privado de tipo Asignatura.
Carrera	Atributo privado de tipo Carrera.
Año	Atributo numérico privado que representa el año en que se dicta la asignatura dentro de la carrera.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.

Métodos

A continuación se establecen el conjunto de métodos que contiene el objeto Plan. Se debe aclarar que para cada atributo se establece el correspondiente Getter y Setter.

Nombre	Descripción
Construct	Constructor del objeto. Realiza la creación del objeto. Se puede recibir el identificador de asignatura y código de carrera, con los cuales se obtiene la información correspondiente desde la base de datos. En caso que no se indiquen, se crea el objeto con los atributos nulos.
Crear	Crea un nuevo registro en la tabla asignatura carrera de la base de datos. Antes de la creación se realiza una búsqueda para evitar que se inserte nuevamente. Se debe recibir el identificador de asignatura y el código de carrera.
Borrar	Se borra un registro a partir del identificador de la asignatura y el código de la carrera.
Buscar	Se busca un registro en la tabla. Para ello se debe indicar el identificador de asignatura y código de carrera. Cuando se encuentra un registro, se actualizan los atributos del objeto con la información obtenida. En caso contrario, los atributos serán nulos.

Usuario

Descripción

Este objeto representa a un usuario del sistema. Se utiliza para controlar si un usuario está registrado en el sistema y si tiene los permisos necesarios para acceder a cierta funcionalidad. Como tal, un usuario cumple uno varios roles y son estos los que contienen los permisos a las funcionalidades disponibles en el sistema.

Atributos

Los atributos del usuario tienen relación con la correspondiente tabla de la base de datos y las relaciones que esta tiene. Los atributos que contiene un usuario son:

Nombre	Descripción
Identificador	Campo privado y numérico que se corresponde con la clave primaria de la tabla "Usuario" en la base de datos.
Email	Campo de texto privado que representa el correo institucional del usuario.
Nombre	Campo de texto privado que representa el nombre de usuario.
MetodoLogin	Campo privado que sirve para establecer la forma en que se logea el usuario. Puede ser un usuario manual o google.
Estado	Campo privado que sirve para desactivar el estado de un usuario. El estado indica si el usuario está activo o inactivo.
Datos	Un atributo privado que se utiliza para obtener el resultado de las consultas con la base de datos. Este atributo es utilizado solamente en el ámbito de esta clase, por lo que no requiere métodos para obtenerlo o modificarlo.
Roles	Conjunto de uno o más roles que puede cumplir el usuario.

Métodos

Los métodos que contiene el usuario son los siguientes:

Nombre	Descripción
Construct	Método constructor.
Obtener (Get)	Para cada uno de los atributos se presenta un método get.
Modificar (Set)	Para cada uno de los atributos se presenta un método set.
Posee rol	Devuelve verdadero o falso si el usuario tiene un rol determinado.
Agregar rol	Agrega un nuevo rol al usuario.

Diseño de Subsistemas

Los subsistemas de diseño y clases de diseño representan abstracciones del subsistema y componentes de la implementación del sistema. Estas abstracciones son directas, y representan una sencilla correspondencia entre el diseño y la implementación.

Los subsistemas de diseño son una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Un subsistema puede constar de clases de diseño, interfaces y otros subsistemas. Por otro lado, un subsistema puede proporcionar interfaces que representan la funcionalidad que exporta en términos de operaciones.

Subsistema de Aulas

Propósito

Este subsistema ha sido creado para agrupar los componentes de diseño que se encuentra relacionados con las Aulas. Los componentes constan de las vistas, modelos y manejadores que se han identificado. Además, surge de la fuerte cohesión que existe entre dichos componentes.

Este subsistema permite representar una separación de ciertos aspectos de diseño. Esto permite que se puedan desarrollar los componentes por separado, e incluso en forma simultánea con otros subsistemas.

Se realiza la creación de este subsistema con el objetivo de permitir la modularidad, mantenimiento y el crecimiento del sistema. El agrupamiento de las clases que se relacionan entre sí, permite el diseño de este subsistema.

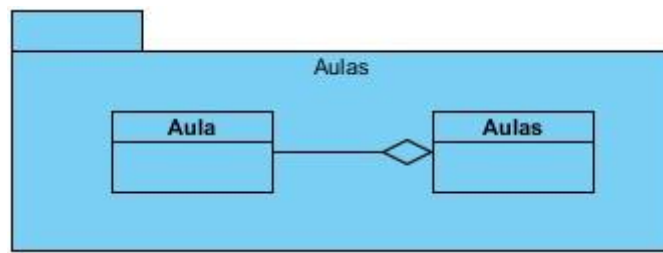
Función

El subsistema se encarga de realizar todas las operaciones que se describen los casos de uso relacionados con las aulas. Esto quiere decir que las funciones que cumple este subsistema están relacionados con los siguientes casos de uso:

- Crear aula.
- Buscar aula.
- Borrar aula.
- Modificar aula.
- Generar horario de aula.

Subordinados

- **Modelos.** Los modelos que integran este subsistema son Aula y Aulas. La descripción de estos objetos de diseño se ha realizado en el apartado anterior. En el siguiente diagrama de clases se observa la relación entre estos modelos:



- **Vistas.** Las vistas son las interfaces con las que interactúa el usuario. Estos son:
 - **Borrar.** Presenta el resultado de una eliminación de aula.
 - **Buscar.** Presente el formulario para realizar una búsqueda de aulas.
 - **Modificar.** Presenta el formulario para modificar un aula.
 - **Resultado buscar.** Presenta el resultado de una búsqueda.
 - **Resultado informe.** Presenta el resultado de un informe.
 - **Resultado modificar.** Presenta el resultado de una modificación.
 - **Informe.** Presenta el formulario para realizar un informe.
- **Controladores.** Este subsistema tendrá un controlador denominado Manejador Aulas. El mismo será encargado de intermediar entre las vistas y los modelos.

Dependencias

No aplica

Subsistema del que depende	Naturaleza de interacción	Características
[Identificación del subsistema del que depende]	[Condiciones para que se realice la interacción]	[Características de la interacción, como ser, pasaje de parámetros, mensajes, datos compartidos, etc.]

Recursos

No se han identificado recursos externos al diseño necesarios para realizar la función.

Subsistema de Carreras

Propósito

Este subsistema ha sido creado para agrupar los componentes de diseño que se encuentra relacionados con las Carreras y Asignaturas. Los componentes constan de las vistas, modelos y manejadores que se han identificado. Además, surge de la fuerte cohesión que existe entre dichos componentes.

Este subsistema permite representar una separación de ciertos aspectos de diseño. Esto permite que se puedan desarrollar los componentes por separado, e incluso en forma simultánea con otros subsistemas.

Se realiza la creación de este subsistema con el objetivo de permitir la modularidad, mantenimiento y el crecimiento del sistema. El agrupamiento de las clases que se relacionan entre sí, permite el diseño de este subsistema.

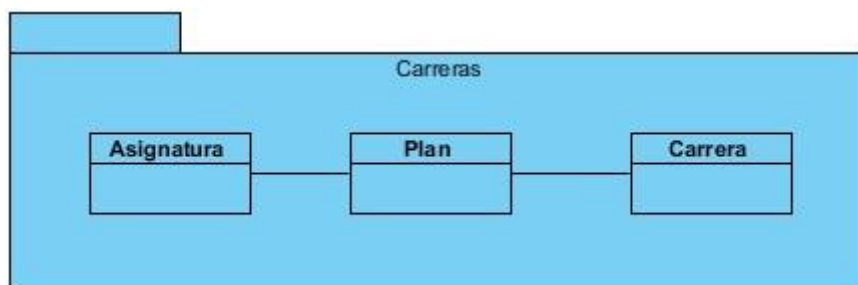
Función

El subsistema se encarga de realizar todas las operaciones relacionadas con las asignaturas y carreras. Los principales casos de uso del sistema requieren del uso de carreras, asignaturas y sus relaciones. Este subsistema será encargado de cubrir estos aspectos. Es decir, realizar operaciones de creación, búsqueda, borrado y modificación sobre asignaturas y carreras.

Subordinados

Como se ha mencionado, este subsistema está integrado por los objetos de diseño que se relacionan directamente con una mesa de examen. Dado que se sigue un patrón de diseño MVC, este subsistema estará compuesto por Modelos, Vistas y Controladores.

- **Modelos.** Los modelos que componen este subsistema son: Asignatura, Carrera y Plan. La descripción de estos objetos se ha realizado en el apartado de Diseño de Objetos. A continuación se adjunta un diagrama de clases que muestra los modelos que componen al subsistema:



- **Vistas.** No hay vistas que compongan este subsistema. La información que se utiliza en este subsistema es tomada de las vistas pertenecientes a los subsistemas de cursadas y mesas.
- **Controladores.** No hay controlador para este subsistema.

Recursos

No se han identificado recursos externos al diseño necesarios para realizar la función.

Subsistema de Cursadas

Propósito

Este subsistema ha sido creado para agrupar los componentes de diseño que se encuentra relacionados con las Cursadas. Los componentes constan de las vistas, modelos y manejadores que se han identificado. Además, surge de la fuerte cohesión que existe entre dichos componentes.

Este subsistema permite representar una separación de ciertos aspectos de diseño. Esto permite que se puedan desarrollar los componentes por separado, e incluso en forma simultánea con otros subsistemas.

Se realiza la creación de este subsistema con el objetivo de permitir la modularidad, mantenimiento y el crecimiento del sistema. El agrupamiento de las clases que se relacionan entre sí, permite el diseño de este subsistema.

Función

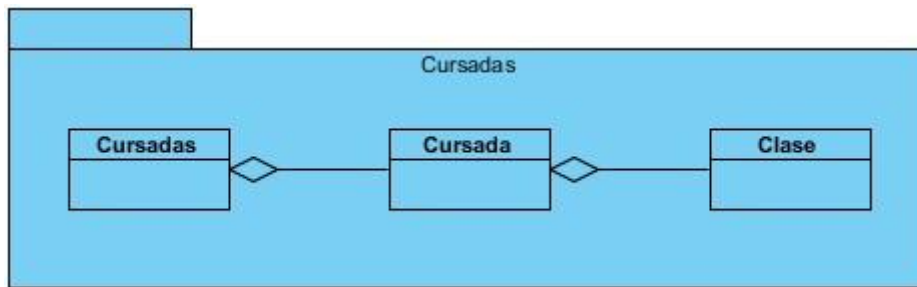
El subsistema se encarga de realizar todas las operaciones que se describen los casos de uso relacionados con las Cursadas. Esto quiere decir que las funciones que cumple este subsistema están relacionados con los siguientes casos de uso:

- Importar horarios de cursada.
- Buscar horarios de cursada.
- Modificar horario de cursada.
- Crear horario de cursada.
- Informe horario de cursada.
- Borrar horario de cursada.

Subordinados

Como se ha mencionado, este subsistema está integrado por los objetos de diseño que se relacionan directamente con una mesa de examen. Dado que se sigue un patrón de diseño MVC, este subsistema estará compuesto por Modelos, Vistas y Controladores.

- **Modelos.** Los modelos que componen el subsistema son Cursada y Clase. La descripción de estos objetos de diseño se ha realizado en el apartado correspondiente. La relación entre estos se observa en el siguiente diagrama:



- **Vistas.** Las vistas que componen el presente subsistemas son las que se observan a continuación.
 - **Borrar.** Presenta el resultado de una eliminación de cursada.
 - **Buscar.** Presente el formulario para realizar una búsqueda de cursadas.
 - **Crear.** Presenta el formulario para crear una cursada.
 - **Importar.** Presenta la tabla con los datos de un archivo csv.
 - **Informe.** Presenta el formulario para realizar un informe.
 - **Modificar.** Presenta el formulario para modificar una cursada.
 - **Resultado buscar.** Presenta el resultado de una búsqueda.
 - **Resultado importar.** Presenta el resultado de la importación de un archivo.
 - **Resultado informe.** Presenta el resultado del informe.
 - **Resultado modificar.** Presenta el resultado de la modificación de cursada.
 - **Seleccionar.** Presenta el formulario para seleccionar un archivo csv.
- **Controladores.** Se ha identificado al Manejador Cursada. Este controlador es encargado de comunicar las vistas con los modelos correspondientes para el desarrollo de las actividades.

Recursos

No se han identificado recursos externos al diseño necesarios para realizar la función. Cabe destacar que este subsistema será accedido desde clientes representados por navegadores web y dispositivos móviles.

Subsistema de Mesas

Propósito

Este subsistema tiene el propósito de brindar la funcionalidad necesaria para administrar las mesas de examen. Está integrado por las clases correspondientes a Docente, Tribunal y Mesa de Examen.

Se realiza la creación de este subsistema con el objetivo de permitir la modularidad, mantenimiento y el crecimiento del sistema. El agrupamiento de las clases que se relacionan entre sí, permite el diseño de este subsistema.

Función

Las funciones que cumple este subsistema se relacionan con los siguientes casos de uso:

- Importar mesas de examen.
- Crea mesa de examen.
- Buscar mesas de examen.
- Borrar mesa de examen.
- Modificar mesa de examen.
- Informe de mesa de examen.

Por lo tanto, podemos decir que mediante el uso de este subsistema se podrá realizar la importación de un archivo que contenga mesas de examen para poder ser almacenado complementemente en la base de datos. A partir de esta información almacenada, se permite realizar la búsqueda de mesas de examen junto con la posibilidad de realizar una eliminación o modificación.

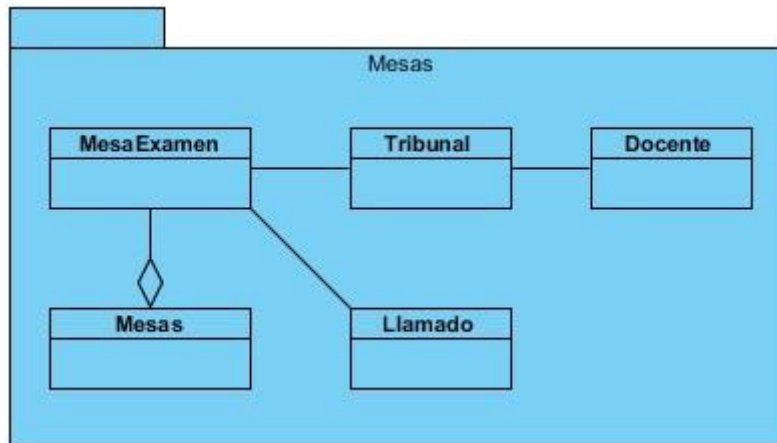
Se permite la creación de una mesa de examen en forma individual en caso que se requiera por alguna razón.

Se permite la generación de informes sobre las mesas de examen como obtener la disponibilidad de aulas para que sean asignadas a dicha mesa de examen.

Subordinados

Como se ha mencionado, este subsistema está integrado por los objetos de diseño que se relacionan directamente con una mesa de examen. Dado que se sigue un patrón de diseño MVC, este subsistema estará compuesto por Modelos, Vistas y Controladores.

- **Modelos.** Los modelos que componen este subsistema son: Mesa de Examen, Mesas de Examen, Tribunal y Docente. La descripción de estos objetos se ha realizado en el apartado de Diseño de Objetos. A continuación se adjunta un diagrama de clases que muestra los modelos que componen al subsistema:



- **Vistas.** Un usuario del sistema interactúa a través de las vistas.
 - **Borrar.** Presenta el resultado de una eliminación de mesa.
 - **Buscar.** Presente el formulario para realizar una búsqueda de mesas.
 - **Crear.** Presenta el formulario para crear una mesa de examen.
 - **Importar.** Presenta la tabla con los datos de un archivo csv.
 - **Informe.** Presenta el formulario para realizar un informe.
 - **Modificar.** Presenta el formulario para modificar una mesa de examen.
 - **Resultado buscar.** Presenta el resultado de una búsqueda.
 - **Resultado importar.** Presenta el resultado de la importación de un archivo.
 - **Resultado informe.** Presenta el resultado del informe solicitado.
 - **Resultado modificar.** Presenta el resultado de la modificación de mesa.
 - **Seleccionar.** Presenta el formulario para seleccionar un archivo csv.
- **Controladores.** Se ha identificado al Manejador Mesa. Este controlador es encargado de comunicar las vistas con los modelos correspondientes para el desarrollo de las actividades.

Recursos

No se han identificado recursos externos al diseño necesarios para realizar la función. Cabe destacar que este subsistema será accedido desde clientes representados por navegadores web y dispositivos móviles.

Subsistema de Usuarios

Propósito

Este subsistema tiene el propósito de brindar la funcionalidad necesaria para administrar los usuarios del sistema. Este subsistema está integrado por las clases que corresponden a Usuarios, Roles y Permisos.

El presente subsistema es reutilizado a partir del proyecto UargFlow.

Función

Las funciones que cumple este subsistema se relacionan con los siguientes casos de uso:

- Crear usuario.
- Buscar usuario.
- Borrar usuario.
- Modificar usuario.
- Crear rol.
- Buscar rol.
- Borrar rol.
- Modificar rol
- Crear permiso.
- Buscar permiso.
- Borrar permiso.

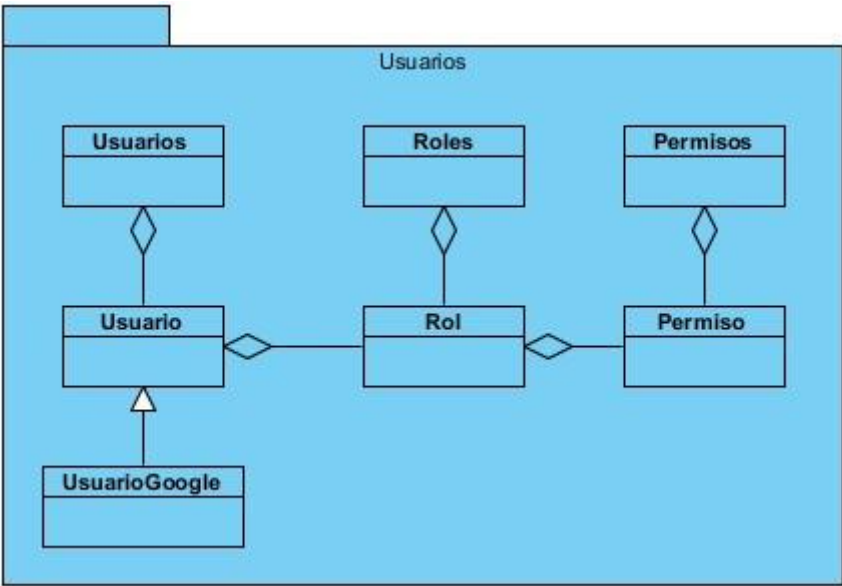
Por lo tanto, podemos decir que mediante el uso de este subsistema se podrá realizar la creación de usuarios, roles y permisos. Gestionar los roles que cumple un determinado usuario, junto con los permisos asociados a dicho rol.

Con esto se puede establecer las opciones que tiene disponible un usuario y presentarlas según corresponda. Se brinda un acceso al sistema para aquellas opciones habilitadas que tenga el usuario.

Subordinados

Como se ha mencionado, este subsistema está integrado por los objetos de diseño que se relacionan directamente con una mesa de examen. Dado que se sigue un patrón de diseño MVC, este subsistema estará compuesto por Modelos, Vistas y Controladores.

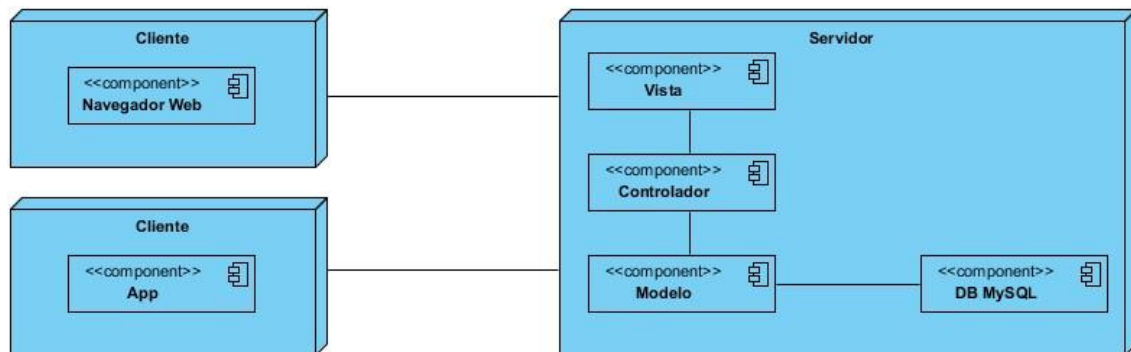
- **Modelos.** Los modelos que componen este subsistema son: Usuario, Usuario Google, Usuarios, Rol, Roles, Permiso y Permisos. La descripción de estos objetos se ha realizado en el apartado de Diseño de Objetos.
A continuación se adjunta un diagrama de clases que muestra los modelos que componen al subsistema:



Diagramas

Diagrama de componentes

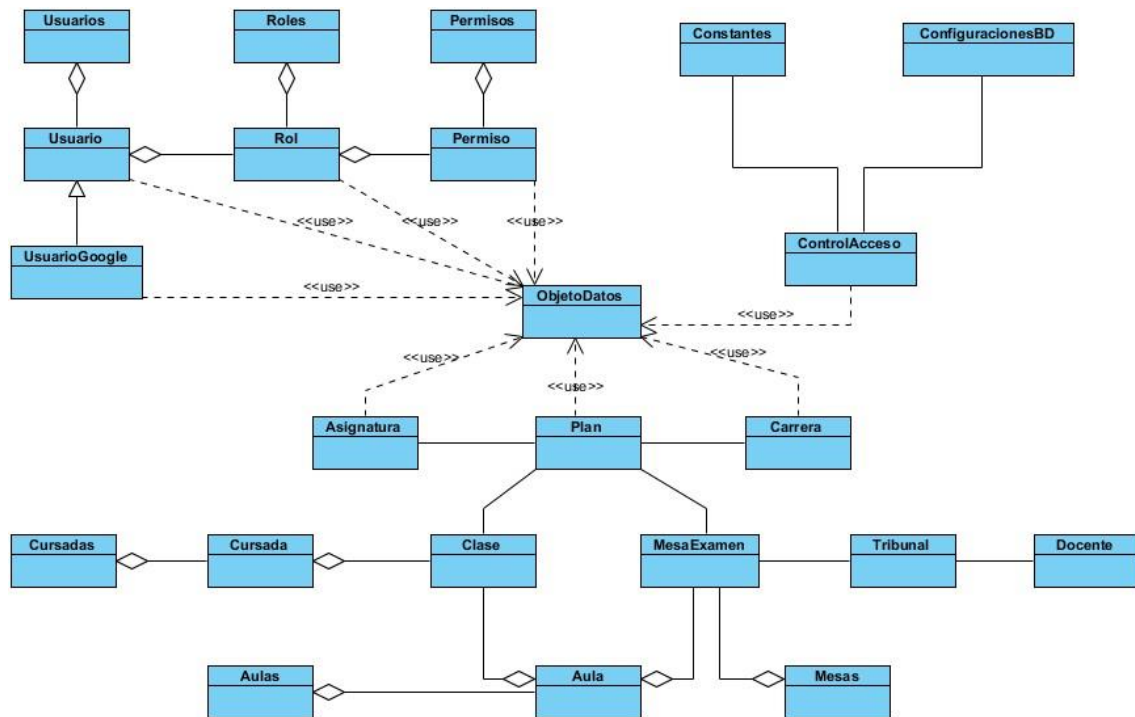
En el servidor se realiza una división de los componentes según: modelos, vistas y controladores. Un cliente se conecta con la vista del sistema y esta se comunica con el controlador correspondiente. El controlador se encarga de comunicar y relacionar las vistas y modelos según las funcionalidades que deban realizarse. Son los modelos los que generan las sentencias SQL que se ejecutan en la base de datos.



Se debe mencionar que tanto las vistas, modelos como controladores se dividen según el subsistema al que pertenezcan.

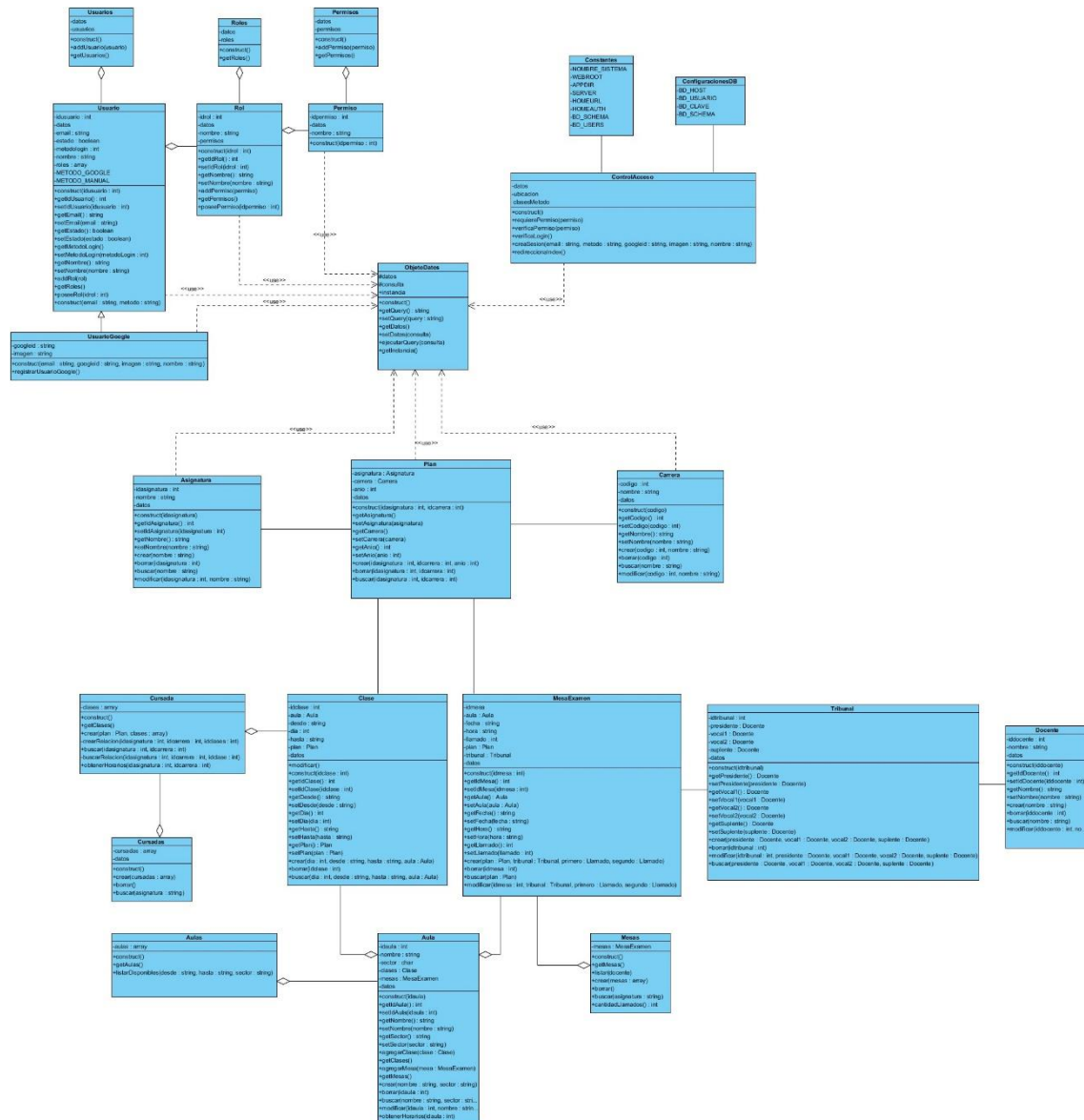
Diagrama de Clases

En UML el diagrama de clases es uno de los tipos de diagramas o símbolo estático y tiene como fin describir la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, en donde se intentan conformar el diagrama conceptual de la información que se manejará en el sistema. Como ya sabemos UML es un modelado de sistema Orientados a Objetos, por ende los conceptos de este paradigma se incorporan a este lenguaje de modelado.



A continuación se observa el mismo diagrama de clases con toda la información correspondiente a cada caso. Se pueden observar los atributos y los métodos correspondientes.

Modelo de Diseño TEMPUS



Diagramas de Paquetes

No aplica

