

Modelo de Diseño

TEMPUS





*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 5](#_Toc496188288)

[Propósito 5](#_Toc496188289)

[Alcance 5](#_Toc496188290)

[Definiciones, siglas y abreviaturas. 6](#_Toc496188291)

[Referencias 6](#_Toc496188292)

[Visión general 6](#_Toc496188293)

[Diseño de Casos de Uso 6](#_Toc496188294)

[Diseño del CU01 – Ingresar al sistema 6](#_Toc496188295)

[Diagrama de paquetes 6](#_Toc496188296)

[Diagrama de Interacción 6](#_Toc496188297)

[Diseño de Flujo de eventos 6](#_Toc496188298)

[Requerimientos especiales o de implementación 7](#_Toc496188299)

[Diseño del CU02 – Importar horarios de cursada 8](#_Toc496188300)

[Diagrama de paquetes 8](#_Toc496188301)

[Diagrama de Interacción 8](#_Toc496188302)

[Diseño de Flujo de eventos 9](#_Toc496188303)

[Requerimientos especiales o de implementación 9](#_Toc496188304)

[Diseño del CU03 – Importar mesas de examen 10](#_Toc496188305)

[Diagrama de paquetes 10](#_Toc496188306)

[Diagrama de interacción 10](#_Toc496188307)

[Diseño de flujo de eventos 11](#_Toc496188308)

[Requerimientos especiales o de implementación 11](#_Toc496188309)

[Diseño del CU04 – Buscar horarios de cursada 12](#_Toc496188310)

[Diagrama de paquetes 12](#_Toc496188311)

[Diagrama de interacción 12](#_Toc496188312)

[Diseño de flujo de eventos 13](#_Toc496188313)

[Requerimientos especiales o de implementación 13](#_Toc496188314)

[Diseño del CU05 – Buscar mesas de examen 13](#_Toc496188315)

[Diagrama de paquetes 13](#_Toc496188316)

[Diagrama de interacción 14](#_Toc496188317)

[Diseño de flujo de eventos 14](#_Toc496188318)

[Requerimientos especiales o de implementación 14](#_Toc496188319)

[Diseño de Objetos 14](#_Toc496188320)

[Aula 14](#_Toc496188321)

[Descripción 14](#_Toc496188322)

[Atributos 14](#_Toc496188323)

[Métodos 15](#_Toc496188324)

[Aulas 15](#_Toc496188325)

[Descripción 15](#_Toc496188326)

[Atributos 15](#_Toc496188327)

[Métodos 15](#_Toc496188328)

[Asignatura 16](#_Toc496188329)

[Descripción 16](#_Toc496188330)

[Atributos 16](#_Toc496188331)

[Métodos 16](#_Toc496188332)

[Carrera 17](#_Toc496188333)

[Descripción 17](#_Toc496188334)

[Atributos 17](#_Toc496188335)

[Métodos 17](#_Toc496188336)

[Clase 18](#_Toc496188337)

[Descripción 18](#_Toc496188338)

[Atributos 18](#_Toc496188339)

[Métodos 18](#_Toc496188340)

[Cursada 19](#_Toc496188341)

[Descripción 19](#_Toc496188342)

[Atributos 19](#_Toc496188343)

[Métodos 19](#_Toc496188344)

[Docente 19](#_Toc496188345)

[Descripción 19](#_Toc496188346)

[Atributos 19](#_Toc496188347)

[Métodos 20](#_Toc496188348)

[Usuario 20](#_Toc496188349)

[Descripción 20](#_Toc496188350)

[Atributos 20](#_Toc496188351)

[Métodos 21](#_Toc496188352)

[Rol 21](#_Toc496188353)

[Descripción 21](#_Toc496188354)

[Atributos 21](#_Toc496188355)

[Métodos 21](#_Toc496188356)

[Permiso 21](#_Toc496188357)

[Descripción 21](#_Toc496188358)

[Diseño de Subsistemas 22](#_Toc496188359)

[Subsistema de Aulas 22](#_Toc496188360)

[Propósito 22](#_Toc496188361)

[Función 22](#_Toc496188362)

[Subordinados 22](#_Toc496188363)

[Dependencias 22](#_Toc496188364)

[Recursos 23](#_Toc496188365)

[Subsistema de Carreras 23](#_Toc496188366)

[Propósito 23](#_Toc496188367)

[Función 23](#_Toc496188368)

[Subordinados 23](#_Toc496188369)

[Subsistema de Cursadas 24](#_Toc496188370)

[Propósito 24](#_Toc496188371)

[Función 24](#_Toc496188372)

[Subordinados 24](#_Toc496188373)

[Subsistema de Mesas 24](#_Toc496188374)

[Propósito 24](#_Toc496188375)

[Función 25](#_Toc496188376)

[Subordinados 25](#_Toc496188377)

[Subsistema de Usuarios 26](#_Toc496188378)

[Propósito 26](#_Toc496188379)

[Función 26](#_Toc496188380)

[Subordinados 26](#_Toc496188381)

[Diagramas 28](#_Toc496188382)

[Diagrama de componentes 28](#_Toc496188383)

[Diagrama de Clases 28](#_Toc496188384)

[Diagrama de Secuencia 29](#_Toc496188385)

[Diagramas de Paquetes 30](#_Toc496188386)

[Diagrama de Colaboración 31](#_Toc496188387)

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

[Esta sección debe indicar el propósito del documento Modelo de Diseño y la audiencia esperada para este documento.]

Alcance

[Esta sección es una breve descripción de a que se aplica el documento Modelo de Diseño, que es afectado o influenciado por este documento.]

Definiciones, siglas y abreviaturas.

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

Referencias

[Esta sección debe proporcionar una lista completa de todos los documentos a los que se hace referencia en el documento Modelo de Diseño. Cada documento debe identificarse por el título, número del informe (si se aplica), fecha, y organización que lo publica. Especifique las fuentes de las que pueden obtenerse las referencias. Esta información puede proporcionarse por la referencia a un apéndice o a otro documento.]

Visión general

[Esta sección describe que contiene el resto del documento Modelo de Diseño y explica como se organiza este documento.]

Diseño de Casos de Uso

Diseño del CU01 – Ingresar al sistema

Diagrama de paquetes

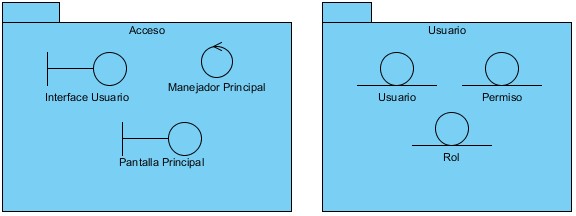


Diagrama de Interacción

[Se sugiere realizar un diagrama de interacción para representar la realización de cada Caso de Uso.]

Diseño de Flujo de eventos

[Descripción escrita en términos de objetos y subsistemas de diseño, que explica y complementa el diagrama de interacción y sus niveles.]

Requerimientos especiales o de implementación

[Descripción que recoge los requerimientos (no funcionales) en la realización de un caso de uso.]

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



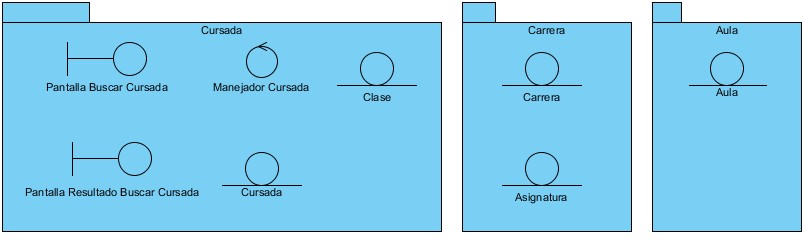
### Requerimientos especiales o de implementación

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.

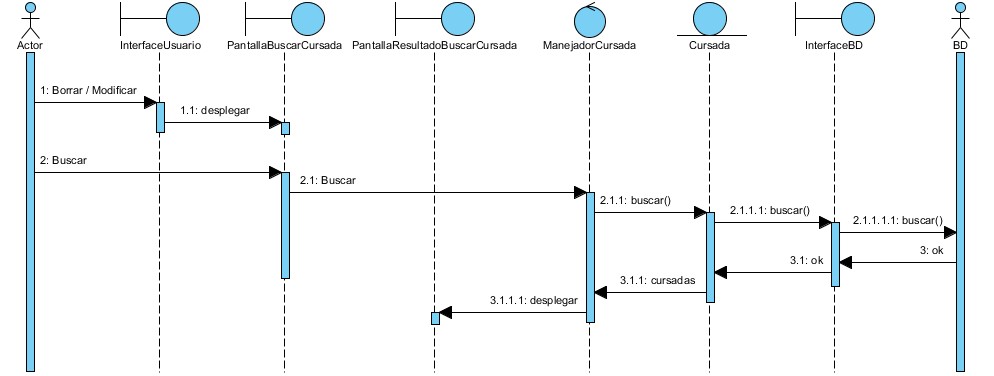
Con este objetivo, se divide la importación en dos partes: Una para leer, detectar y mostrar errores en el archivo, y otra parte donde se lleva a cabo la carga en la base de datos.

## Diseño del CU04 – Buscar horarios de cursada

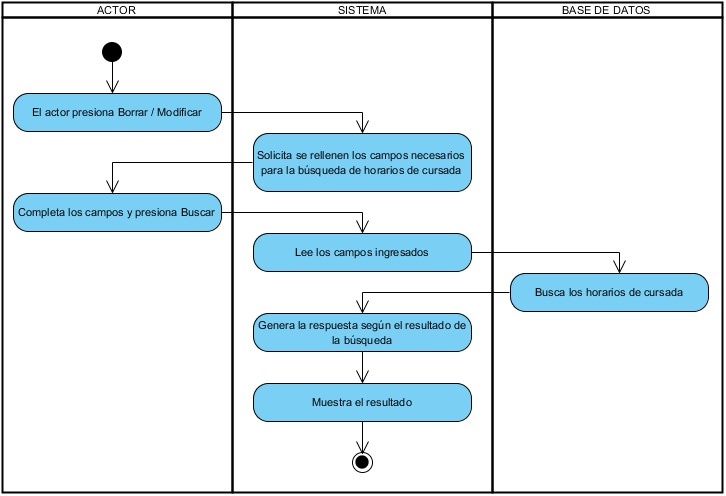
### Diagrama de paquetes



### Diagrama de interacción



### Diseño de flujo de eventos



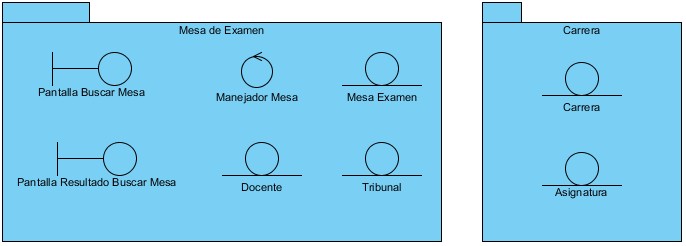
### Requerimientos especiales o de implementación

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.

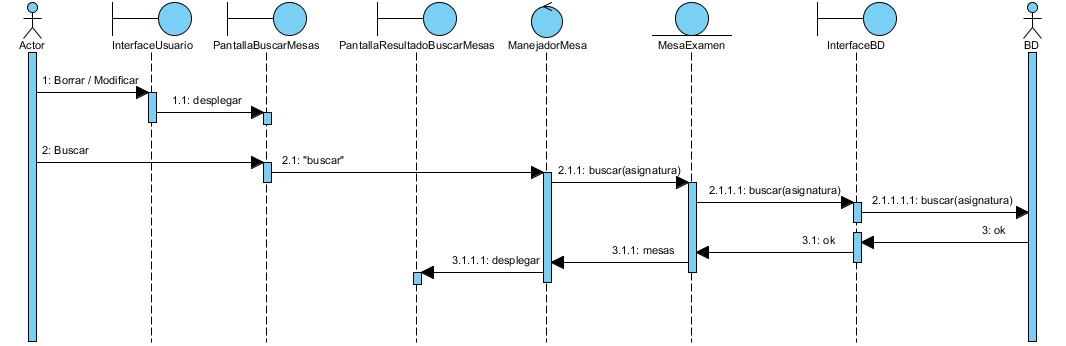
En el caso de la aplicación móvil, se debe buscar una forma de representar tablas grandes en consideración del tamaño de las pantallas en los dispositivos móviles.

## 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. |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Constructor | Un constructor que no contiene ningún parámetro. Sirve para crear un aula sin información asociada. |
| Constructor | Un constructor que utiliza el identificador de un aula. Sirve para obtener la información del aula desde la base de datos. |
| Obtener (get) | Todos los atributos de clase contienen los métodos get. |
| Modificar (set) | Todos los atributos de clase contienen los métodos set. |
| 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. |
| Obtener | Se obtienen los datos de una determinada aula. |
| Modificar | Se modifica la información correspondiente al aula en la base de datos. |
| 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. |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Construct | Constructor que no contiene parámetros de entrada. Sirve para crear y utilizar el objeto. |
| Get aulas | Se obtiene el conjunto de aulas que es controlado por la clase. |
| Listar disponibles | Consulta en la base de datos todas las aulas que se encuentren disponibles. Se obtienen para todos los sectores y horarios. |
| Listar disponibles | Consulta en la base de datos todas las aulas que se encuentren disponibles en un determinado sector. |
|  |  |

## 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. |
| Nombre | Nombre de la asignatura. Es un atributo privado que debe ser accedido a través de los métodos públicos. |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Constructor | Contiene el constructor de la clase asignatura. |
| Obtener (get) | Todos los atributos de clase contienen los métodos get. |
| Modificar (set) | Todos los atributos de clase contienen los métodos set. |
| Crear | Se crea una nueva aignatura 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. |
| Obtener | Se obtienen los datos de una determinada asignatura. |
| Modificar | Se modifica la información correspondiente a la asignatura en la base de datos. |
| Obtener horarios | Se utiliza para obtener los horarios de clase que contiene el aula especificada. |

## 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. |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Constructor | Constructor de clase. |
| Obtener (get) | Todos los atributos de clase contienen los métodos get. |
| Modificar (set) | Todos los atributos de clase contienen los métodos set. |
| Crear | Se crea una nueva carrera y se almacena en la base de datos. Se requiere el código y nombre de la carrera. |
| Borrar | Se elimina una determinada carrera de la base de datos. |
| Obtener | Se obtienen los datos de una determinada carrera. |
| Modificar | Se modifica la información correspondiente a la carrera en la base de datos. |

## 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 | Corresponde al día de la semana en el que se dicta la clase. Es un atributo numérico que puede ir desde 1 hasta 6. |
| Desde | Horario desde que comienza la clase. Tiene el formato HH:MM |
| Hasta | Horario a la que finaliza la clase. Tiene el formato HH:MM |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Construct | Constructor del objeto. |
| Obtener (get) | Cada uno de los atributos de clase contiene los métodos get. |
| Modificar (set) | Cada uno de los atributos de clase dispone de los métodos set. |
| 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. |
| 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 saber el identificador de la misma. |
| 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** |
|  |  |
|  |  |
|  |  |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Construct | Constructor de cursada. |
| Obtener horarios de asignatura | Se obtienen los horarios de cursada (clases) de una determinada asignatura. Se requiere el nombre de la asignatura. |
| Obtener horarios de carrera | Se obtienen los horarios de cursada (clases) para una carrera. Se requiere el código de la carrera. |

## Docente

### Descripción

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

### Atributos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Identificador | Campo 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. |

### Métodos

|  |  |
| --- | --- |
| **Nombre** | **Descripción** |
| Construct | Constructor del objeto. |
| Obtener (get) | Cada uno de los atributos de clase contiene los métodos get. |
| Modificar (set) | Cada uno de los atributos de clase dispone de los métodos set. |
| Crear | Crea un nuevo docente y lo almacena en la base de datos. |
| Borrar | Se borra un determinado docente de la base de datos. |
| Buscar | Se busca un docente en la base de datos. Para ello se debe saber el identificador de la misma. |
| Modificar | Se modifica un docente en la base de datos. Se pueden modificar todos los campos a excepción del identificador. |

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. |
| Metodo Login | 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 |  |
| 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. |

Rol

### Descripción

### Atributos

### Métodos

Diagrama de clases

Permiso

### Descripción

Diseño de Subsistemas

Subsistema de Aulas

Propósito

[Descripción de por qué el subsistema existe. Este atributo debe dar la razón de la creación del subsistema. Como ser la funcionalidad específica y los requerimientos de performance por los cuales fue creado. También describe requerimientos especiales que se deben lograr con él que no están incluidos en la especificación de requerimientos del software.]

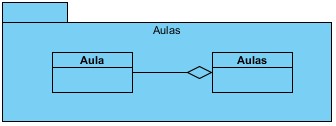
Función

[Expresa qué realiza el subsistema. Establece la transformación aplicada a las entradas del subsistema para producir la salida deseada.]

Subordinados

[Se identifican los objetos de diseño y subsistemas de diseño que componen el subsistema que se describe. Se propone representar esta información con un diagrama de paquetes.]

* **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:



Dependencias

[Descripción de la relación de este subsistema con otros subsistemas de diseño. Describir la naturaleza de cada interacción incluyendo características como tiempo y condiciones de la interacción. Estas, pueden involucrar la iniciación, orden de ejecución, datos compartidos, creación, duplicación, uso o almacenamiento.

Se propone representar esta información con una tabla de dependencias.]

|  |  |  |
| --- | --- | --- |
| **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

[Se deben Identificar y describir todos los recursos externos al diseño que necesita el subsistema para realizar su función. Especificar las reglas de interacción y métodos para usar el recurso.

Este atributo brinda información sobre elementos como dispositivos físicos (impresoras, particiones de disco, bancos de memoria), servicios de software (librerías, servicios del sistema operativo), y recursos de procesamiento (ciclos de CPU, ubicación de memoria, buffers).

Se deben describir características de uso como el tiempo de proceso al cual se debe adquirir el recurso e incluir la cantidad de tiempo de uso. Debe incluir también la identificación de capacidad potencial y facilidades de manejo del recurso.]

## Subsistema de Carreras

### Propósito

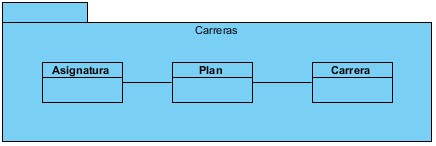
### Función

### 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:



## Subsistema de Cursadas

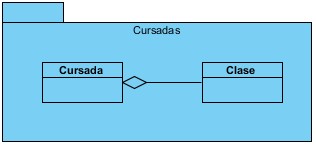
### Propósito

### Función

### 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:



## 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 complemente 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.

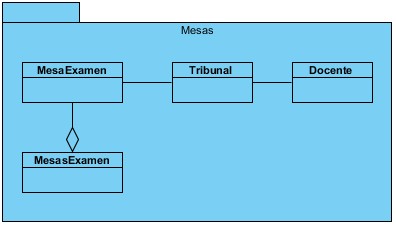
Se permite la creación de una mesa de examen en forma individual en caso que se requiera por alguna razón.

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. Las vistas del subsistema son las que se observan en la siguiente imagen.

## 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.

### 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.

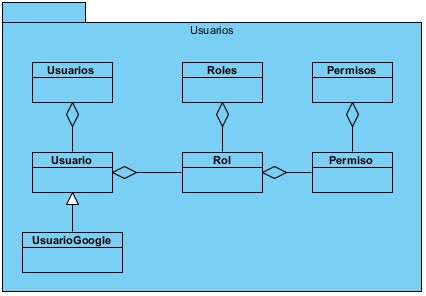
Con esto se puede establecer las opciones que tiene disponible un usuario y presentarlas según corresponda. Se brinda un acceso al sistema para aquellas opciones habilitadas que tenga el usuario.

### 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.

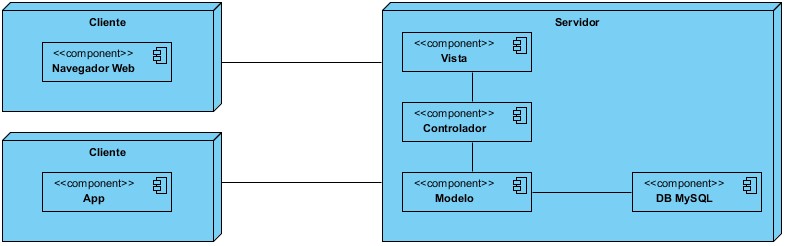
A continuación se adjunta un diagrama de clases que muestra los modelos que componen al subsistema:



# 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.

Los diagramas de clases tienen las siguientes características:

* Las clases define el ámbito de definición de un conjunto de objetos.
* Cada objeto pertenece a una clase.
* Los objetos se crean por instanciación de las clases.

En su representación gráfica contamos con:

* Nombre de la Clase.
* Atributos de la Clase.
* Operaciones con las Clases.]
* 

Diagrama de Secuencia

[En el diagrama de secuencia se muestra la interacción de los objetos que componen un sistema de forma temporal.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase]



Diagramas de Paquetes

[Los diagramas de Paquetes se usan para reflejar la organización de paquetes y sus elementos. Los usos más comunes de para los diagrama de paquete son para organizar diagramas de casos de uso y diagramas de clases, estos paquetes son como grandes contenedores de clases.

Los elementos contenidos en un paquete comparten el mismo espacio de nombres, esto significa que los elementos contenidos en un mismo espaciode nombres específico deben tener nombres únicos.Como otra característica de estos diagramas, cada paquete se debeidentificar con un nombre único y opcionalmente mostrar todos loselementos dentro del mismo.]



Diagrama de Colaboración

[Un diagrama de colaboración, se puede decir que es una formaalternativa al diagrama de secuencias a la hora de mostrar un escenario.

Este tipo de diagrama muestra las interacciones que ocurren entre losobjetos que participan en una situación determinada.A diferencia del diagrama de secuencia, el diagrama de colaboración seenfoca en la relación entre los objetos y su topología de comunicación.En estos diagramas los mensajes enviados de un objeto a otro se representamediante flechas, acompañado del nombre del mensaje, los parámetros yla secuencia del mensaje.

Estos diagramas están indicados para mostrar una situación o flujo deprograma específico y son considerados uno de los mejores diagramaspara mostrar o explicar rápidamente un proceso dentro de la lógica delprograma.]

