

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 4](#_Toc494308586)

[Propósito 4](#_Toc494308587)

[Alcance 4](#_Toc494308588)

[Definiciones, siglas y abreviaturas. 4](#_Toc494308589)

[Referencias 4](#_Toc494308590)

[Visión general 5](#_Toc494308591)

[Diseño de Casos de Uso 5](#_Toc494308592)

[Diseño del CU01 – Ingresar al sistema 5](#_Toc494308593)

[Diagrama de paquetes 5](#_Toc494308594)

[Diagrama de Interacción 5](#_Toc494308595)

[Diseño de Flujo de eventos 5](#_Toc494308596)

[Requerimientos especiales o de implementación 5](#_Toc494308597)

[Diseño del CU02 – Importar horarios de cursada 6](#_Toc494308598)

[Diagrama de paquetes 6](#_Toc494308599)

[Diagrama de Interacción 6](#_Toc494308600)

[Diseño de Flujo de eventos 6](#_Toc494308601)

[Requerimientos especiales o de implementación 6](#_Toc494308602)

[Diseño del CU03 – Importar mesas de examen 6](#_Toc494308603)

[Diagrama de paquetes 6](#_Toc494308604)

[Diseño de Objetos 6](#_Toc494308605)

[Usuario 6](#_Toc494308606)

[Descripción 6](#_Toc494308607)

[Atributos 6](#_Toc494308608)

[Métodos 7](#_Toc494308609)

[Diagrama de clases 7](#_Toc494308610)

[Rol 8](#_Toc494308611)

[Permiso 8](#_Toc494308612)

[Diseño de Subsistemas 8](#_Toc494308613)

[Subsistemas Específicos 8](#_Toc494308614)

[[Nombre del Subsistema Específico 1] 8](#_Toc494308615)

[Propósito 8](#_Toc494308616)

[Función 8](#_Toc494308617)

[Subordinados 8](#_Toc494308618)

[Dependencias 8](#_Toc494308619)

[Recursos 9](#_Toc494308620)

[Diagramas 10](#_Toc494308621)

[Diagrama de componentes 10](#_Toc494308622)

[Diagrama de Clases 10](#_Toc494308623)

[Diagrama de Secuencia 11](#_Toc494308624)

[Diagramas de Paquetes 12](#_Toc494308625)

[Diagrama de Colaboración 13](#_Toc494308626)

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

[En esta sección se especifica el comportamiento de los casos de uso, mediante subsistemas u objetos de diseño que interactúan, y se determinan las operaciones e interfaces de los distintos subsistemas u objetos de diseño.

Un objeto de diseño es una abstracción de un objeto X o varios objetos X en la implementación del sistema. Se define en base a parámetros, reglas, operaciones que realiza, requerimientos de implementación, interfaz de usuario (si corresponde) y relaciones con otros objetos.

Esta especificación se realiza mediante Diagrama de Paquetes, Diseño de flujo de eventos, Diagramas de interacción y requerimientos especiales o de implementación.]

Diseño del CU01 – Ingresar al sistema

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

[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

[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 CU03 – Importar mesas de examen

### Diagrama de paquetes

Diseño de Objetos

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:

* Identificador del usuario. Corresponde al identificador del usuario en la base de datos.
* Email. Corresponde al correo institucional que tiene el usuario.
* Nombre. Es el nombre del usuario.
* Método login. Es la forma en la que accede el usuario. Puede ser un usuario de tipo manual o un usuario google.
* Estado. El usuario se puede dar de baja pero no se elimina de la base de datos. El estado indica cuando un usuario se encuentra activo o inactivo.
* Datos.
* Roles. Un arreglo con los roles que cumple el usuario.

### Métodos

Los métodos que contiene el usuario son los siguientes:

* Obtener (Get) para cada uno de los atributos.
* Modificar (Set) para cada uno de los atributos.
* Posee rol. Devuelve verdadero o falso si el usuario tiene un rol determinado.
* Agregar rol. Agrega un nuevo rol al usuario.

Interacciones:

Diagrama de estado

Interfaces que implementa

Rol

### Descripción

### Atributos

### Métodos

Diagrama de clases

Permiso

### Descripción

Diseño de Subsistemas

Subsistemas Específicos

[Nombre del Subsistema Específico 1]

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.]

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.]

# Diagramas

Diagrama de componentes

[Lo que distingue el diagrama de componentes de otro tipo de diagramases sin duda su contenido. Normalmente contiene componentes, interfacesy relaciones entre ellos.

Los componentes pertenecen a un mundo físico, es decir, representan a unbloque de construcción al modelar aspectos físicos de un sistema.

Cada componente debe tener un nombre que lo distinga de los demás. Al igual que las clases los componentes pueden enriquecerse con compartimientos adicionales que muestran sus detalles.]



Diagrama de Clases

[En UML el diagrama de clases es uno de los tipos de diagramas o símboloestático y tiene como fin describir la estructura de un sistema mostrando susclases, atributos y relaciones entre ellos.Estos diagramas son utilizados durante el proceso de análisis y diseño de lossistemas informáticos, en donde se intentan conformar el diagramaconceptual 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 lenguajede 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 depaquetes y sus elementos. Los usos más comunes de para los diagrama depaquete son para organizar diagramas de casos de uso y diagramas declases, estos paquetes son como grandes contenedores de clases.

Los elementos contenidos en un paquete comparten el mismo espacio denombres, 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.]

