Modelos de análisis y diseño

Gestión de personal ucm

Miguel Pascual Domínguez, Javier Pellejero Ortega, Isabel Pérez Pereda, Iván Prada Cazalla, Jesús Recio Herranz, Álvaro Rodríguez García

Gestor personal UCM

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Proyecto Ingeniería del Software |  | Email portavoz: alvarr11@ucm.es |
|  |  |  |

Contenido

[Tabla modificaciones ii](#_Toc452008974)

[Modelo de Análisis 1](#_Toc452008975)

[Modelo de Diseño 5](#_Toc452008976)

[Pruebas 6](#_Toc452008977)

[Apartado 4 7](#_Toc452008978)

[Apartado 5 8](#_Toc452008979)

# Tabla modificaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Autor | Fecha | Versión | Descripción |
| Grupo completo | 25/3/2016 | 1.0 | Creación del documento |
| Iván Prada Cazalla | 8/4/2016 | 1.1 | División en trozos del apartado de analisis y comienzo de la tarea |
|  | 15/4/2016 | 1.2 | Completar análisis |
|  | 22/4/2016 | 1.3 | Inicio del apartado de diseño. Estructuración |
|  | 29/4/2016 | 2.0 | Añadidos diagramas |
|  | 13/5/2016 | 2.1 | Añadidos diagramas |

# Modelo de Análisis

## Análisis del sistema de software

En este apartado nos vamos a dedicar a hacer un analisis pormenorizado del software. Se trata de realizar un sistema que se encargará de simular el funcinamiento de una aplicación encargada de llevar el sistema de gestión del personal de la universidad complutense.

Los objetivos, requisitos y restricciones del sistema ya quedáron expuestos en el modelo de dominio y en los documentos de requisitos.

Vamos a hacer una división en una serie de niveles contextuales que nos mostrarán una mejor visión del sistema.

### Análisis de la arquitectura(paquetes, clases, requisitos)

En este apartado procederemos a identificar los paquetes, clases evidentes y requisitos especiales comunes que formarán inicialmente parte de nuestro sistema.

Para la identificación de paquetes, agruparemos por funcionalidad común(para los dos grandes subsistemas), y organizaremos los paquetes por capas, para dar una cierta estructuración a nuestra aplicación(ya que tenemos que utilizar un MVC y una arquitectura multicapa).

Tendremos los siguientes paquetes agrupados por funcionalidad común(subsistemas):

* Empleado

El subsistema de empleado será el encargado de realizar todas las operaciones correspondientes a la inserción, busqueda, visualización y vista de los empleados que se demanden.

* Usuario

El subsistema de usuario será el encargado de realizar todas las operaciones relacionadas con la creación y la eliminación de los usuarios que pueden acceder al sistema, así como sus permisos.

Estos a su vez estarán divididos en una serie de paquetes, que agruparán una funcionalidad común, y que estarán en cada uno de los subsistemas, además de las clases principales que tendrian:

* Vistas y controlador

Paquetes encargados de gestionar las interfaces gráficas

Clases iniciales para empleado y usuario: vista genérica.

* Lógica y reglas

Encargados de la seguridad y la lógica de nuestro programa

Clases iniciales para empleado: Empleados, AreaDeTrabajo, Contrato, EmpleadoPAS, EmpleadoPDI, TipoBaja, TipoDocente

Clases iniciales para usuario: Usuario, TIpoFacultad, TIpoPermiso.

#Añadir diagrama clases puestas vacias

La agrupación por capas sería la siguiente(aunque esto se considera diseño):

#### Capa de integración

Contendra a todas las clases y paquetes que realicen una comunicación con la base de datos.

#### Capa de negocio

En ella se situarán todas las clases y paquetes encargados de llevar la lógica de negocio.

#### Capa de presentación

En esta capa irán todas las clases y paquetes destinados a la visualización de los datos solicitados por el actor y procesados por el sistema.

### Descripción del dominio de la información

En este apartado haremos una descripción del dominio de la información, es decir, de los tipos de datos que manejará nuestra aplicación.

Se manejaran los distintos tipos de datos:

-Tipo empleado: representa a una persona dada de alta en nuestro sistema, y que vendrá descrita por :

* Identificador: es un numero unico y designado por el sistema, con el que un empleado quedará identificado.
* Nombre: nombre del empleado que se introducirá en el sistema.
* Apellido1: primer apellido del empleado.
* Apellido2: segundo apellido del empleado.
* Path\_foto: foto adjunta al empleado que quedará registrada en el sistema.
* Dirección: ubicación en la que vive el empleado.
* Idiomas: conjunto de idiomas que conoce el empleado.
* HistorialPath: historial referente al empleado.
* NominaPath: nomina del empleado.
* Contrato: contrato del empleado.
* CurriculumPath: curriculum del empleado.
* TipoBaja: en caso de haber una baja de un empleado cursada, quedará aquí registrada.
* Facultad: facultad en la que se encuentra el empleado.

Si es PDI:

* Departamento: departamento al que pertenece el empleado.
* Especialidad: materia en la que el empleado está especializado.
* Despacho: despacho destinado al empleado.
* TipoDocente: a seleccionar entre ayudante, asociado, titular, catedratico.

Si es PAS:

* ÁreaTrabajo: asignación de tareas que desempeña el empleado.
* Categoría: categoria a la que pertenece el empleado.

-Tipo contrato: información almacenada referente a un contrato. Vendrá descrito por:

* Cuenta bancaria: número de cuenta del banco en la que se hará el ingreso al empleado.
* Horas\_de\_trabajo: número de horas de trabajo designadas a ese empleado.
* Fecha\_Fin: fecha del fin de contrato, en caso de que sea temporal.
* esTemporal: nos indicará si un empleado es temporal.

-Tipo Usuario: representará a un usuario del sistema, y vendrá definido por:;

* Nombre: nombre del usuario, con el que accederá a su sesión.
* Contraseña: clave del usuario para acceder al sistema.
* TipoFacultad: facultad a la que pertenece (matematicas, fisica, quimica, biologia, informatica, medicina, ninguna…).
* TipoPermiso: permiso que restringe la consulta de acceso a empleados(superusuario, administrador\_rectorado, administrador\_facultad, secretario\_pas, secretario\_pdi).

# Modelo de Diseño

En este apartado pasamos a diseñar el sistema a partir de los puntos estudiados en la parte de requisitos y análisis.

En primera instancia haremos un esquema general de los subsistemaas ya mencionados en la sección de diseño, y pasaremos a describir las partes que lo componen. Para ello utilizaremos diagramas de clases, de secuencia, y de actividades.

## Subsistema Empleado

## Subsistema Usuario

# Pruebas

En este apartado trataremos 5 pruebas para 5 casos de uso distintos: “Login”, “Añadir Empleado”, “Añadir Usuario”, “Ver Ficha de Empleado” y “Realizar una búqueda”.

Para cada caso de prueba seguiremos este esquema:

1. Introducción
2. Validación de datos
   1. Descripción
   2. Condiciones de ejecución
   3. Entrada
   4. Validaciones de datos
   5. Resultado esperado

Caso de Prueba: Añadir Usuario

1. Introducción

Este artefacto cubre el conjunto de pruebas realizadas sobre el Caso de Uso: Añadir Usuario.

Las pruebas realizadas a este caso de uso son:

-Validación de usuario a añadir no existe

-Comprobación de permisos adecuados

1. Validación de datos
   1. Descripción

La aplicación comprueba que el usuario que se quiere añadir a la base de datos no existe así como que el usuario logueado tiene los permisos necesarios para añadir usuarios de ese tipo (administrador de rectorado, administrador de facultad, secretario de tipo PAS o de tipo PDI).

* 1. Condiciones de ejecución
  2. Entrada