

Tipo de artículo: Artículo original
Temática: soluciones informáticas
Recibido: 11/01/2019 | Aceptado: 22/02/2019 | Publicado: 20/03/2019

Diseño de un Sistema de Gestión de Información de Recursos Humanos *Design of a Human Resources Information Management System*

Dayana Hardy Demestre¹, Yulina Hernández González²

¹ Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas, dhardy@estudiantes.uci.cu

² Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas, yulina@uci.cu

* Autor para correspondencia: dhardy@estudiantes.uci.cu

Resumen

El Vicedecanato Administrativo es un departamento de la facultad de ciencias y tecnologías computacionales (CITEC) de la Universidad de Ciencias Informáticas (UCI), este se encarga de las tareas de administración y economía de la facultad, dicho departamento solicita a la autora del presente proyecto la elaboración de un sistema informático que ayude a mejorar la gestión de la información de los recursos humanos de dicha facultad. El trabajo de diploma describe el proceso de desarrollo del Sistema web para el departamento de vicedecanato administrativo; en él se muestran los aciertos y problemas presentados durante el proceso de desarrollo, así como la descripción de la metodología empleada. Para el desarrollo de esta investigación se utilizaron los métodos empíricos entrevista y análisis de documentos mientras que se empleó la modelación y el analítico-sintético como métodos teóricos. La metodología de desarrollo de software AUP-UCI guió el desarrollo de la solución propuesta, utilizando como marcos de trabajo Django en su versión 1.10 y el lenguaje Python en su versión 3.5, PostgreSQL 9.1 como gestor de base de datos y PyCharm Community 2018.3.2 como entorno de desarrollo. Se obtuvo como resultado un sistema con una arquitectura sólida diseñada sobre la base del patrón GRASP y validada con diferentes técnicas de pruebas de software.

Palabras claves: recursos humanos, gestión de información, sistema informático.

Abstract

The Administrative Vice Dean is a department of the faculty of computer science and technologies (CITEC) of the University of Computer Science (UCI), this is responsible for the administration and economics of the faculty, said department requests the author of the present project the elaboration of a computer system that helps to improve the management of the information of the human resources of said faculty. The diploma work describes the process of developing the web system for the department of administrative vice-dean; in the other they show the successes and problems presented during the development process, as well as the description of the methodology used. For the development of this research, empirical methods of interviewing and analysis of documents were used, while

modeling and analytical-synthetic methods were used as theoretical methods. The AUP-UCI software development methodology guided the development of the proposed solution, using as frameworks Django in its version 1.10 and the Python language in its version 3.5, PostgreSQL 9.1 as the database manager and PyCharm Community 2018.3.2 as development environment. The result was a system with a solid architecture designed on the basis of the GRASP standard and validated with different software testing techniques.

Keywords: *human resources, information management, computer system.*

Introducción

La Universidad de las Ciencias Informáticas (UCI) es una de las universidades con mayor capacidad tecnológica instalada en el país. Esta como institución desarrolladora de software, se enfoca en automatizar la mayor cantidad de procesos. En la Facultad de Ciencias y Tecnologías Computacionales CITEC, el Vicedecanato Administrativo, se encarga de las actividades de administración y economía de la Facultad CITEC; este es un ente fundamental en los procesos que se desarrollan en la misma. Este departamento cuenta con un Vicedecano Administrativo, un técnico de recursos humanos y una asistente de dirección. El técnico de recursos humanos es la persona encargada de gestionar toda la información referente a los recursos humanos de la Facultad CITEC, entendida como recursos humanos a los trabajadores de la facultad (FERNÁNDEZ 2006), (MAR, OMAR *et al.* 2015a), (MAR, OMAR *et al.* 2015b).

Actualmente la actividad desempeñada por el técnico de recursos humanos se realiza de forma manual, los datos laborales y personales de los trabajadores de la facultad se encuentran descentralizados, lo que evita que se pueda acceder a ellos en cualquier momento, esto incide notablemente en demoras a la hora de reunir información. En ocasiones la información se pierde por deterioro de documentos, debido a su uso manual. El registro y consulta de los datos resulta complejo por el gran volumen de información a almacenar; de igual manera pueden cometerse errores a la hora de registrar los mismos. Además por el cúmulo de información es posible la dualidad de esta; lo que trae consigo la redundancia en los datos y quita exactitud en el momento de generar los reportes. Todo lo antes mencionado hace que se vuelva compleja la gestión de los recursos humanos de la facultad, lo que influye negativamente, imposibilitando así el logro de las metas y objetivos (MONDY 2005), (BEER 1990).

Materiales y métodos o Metodología computacional

En la presente sesión se realiza la descripción de una serie de elementos para la comprensión de la propuesta de solución de la investigación. Con la construcción del modelo de dominio se obtendrá un acercamiento mayor del

problema a resolver y con la identificación de los requisitos funcionales y no funcionales serán mostradas las cualidades con las cuales deberá de cumplir el sistema. Además se muestran en el mismo un conjunto de diagramas desarrollados durante el análisis y diseño del sistema según las características propuestas por la metodología seleccionada, AUP-UCI (MAR, O *et al.* 2015c), (SANTOS 2012).

Modelo Conceptual

Se le denomina modelo conceptual aquello que proporciona la ventaja de subrayar una concentración en los conceptos del dominio, y no en las entidades del software (LARMAN 2005). El modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos o mediante la entrevista con los expertos del dominio.

Este modelo de dominio constituye el punto de partida para lograr realizar un diseño acertado del producto a desarrollar, para definir a partir del mismo un conjunto de clases conceptuales u objetos existentes del mundo real en el dominio del problema planteado (GARZA 2012), (SALAS 2010), (RODRÍGUEZ 2010). Como resultado se determinó que realizar un modelo de dominio constituiría una forma óptima de manejar los cambios que se producen en el mismo.

Para una mejor comprensión de los conceptos asociados al funcionamiento de la plataforma se mostrará a continuación el glosario de términos y el modelo de dominio.

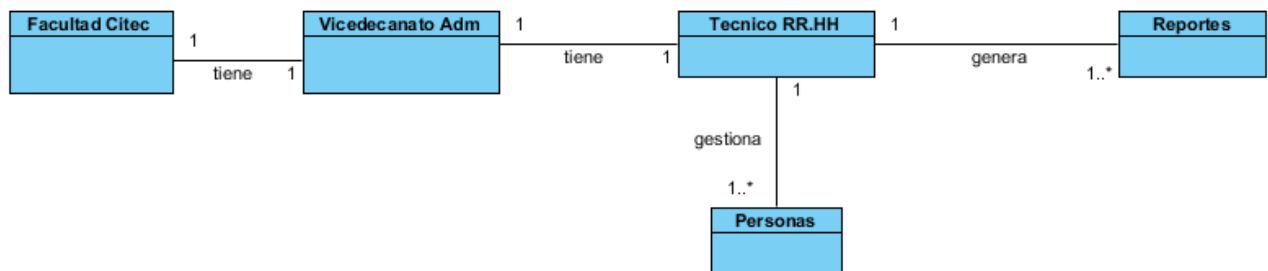


Figura 1: Modelo Conceptual.

Glosario de términos

Facultad CITEC: representa la facultad de ciencias y tecnologías computacionales CITEC donde se encuentra el Vicedecanato Administrativo de la Universidad de Ciencias Informáticas.

Vicedecanato Administrativo: Departamento encargado de Administración y Economía dentro de la facultad CITEC de la UCI.

Técnico de Recursos Humanos: Persona encargada de ejecutar todos los procesos referentes a recursos humano.

Personas: constituye los recursos humanos de la facultad CITEC.

Reportes: representa los archivos generados por el sistema a partir de la información obtenida de la base de datos.

Descripción de la propuesta de solución

El sistema desarrollado tiene como objetivo fundamental gestionar la información de los Recursos Humanos de la facultad CITEC adscripta al Vicedecanato Administrativo. Para ello el sistema debe permitir al administrador gestionar diferentes usuarios y al técnico de RRHH insertar, modificar, eliminar y listar diferentes personas; además de generar reportes de acuerdo a las categorías docentes, residencia, cargos, expedientes, u organizaciones.

Especificación de requisitos

En el proceso de ingeniería de software, los requisitos se utilizan como datos de entrada en la etapa de diseño del producto que se desea lograr. Estos representan qué debe hacer el sistema, pero no cómo hacerlo, además, incorporan las cualidades o propiedades que el mismo debe poseer.

Los requisitos ayudan al ingeniero del software a entender mejor el problema en cuya solución trabaja. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software (PRESSMAN 2010). En la especificación de requisitos del software se definen requisitos funcionales los cuales describen los servicios o funciones que se esperan del sistema y los requisitos no funcionales que constituyen restricciones sobre los requisitos funcionales.

Para el levantamiento de los requisitos se utilizó la entrevista, que es una técnica de recopilación de información mediante una conversación profesional, con la que además de adquirirse información acerca de lo que se investiga, tiene importancia educativa; y depende en gran medida del nivel de comunicación entre el investigador y los participantes en la misma.

Requisitos funcionales

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos. Los requerimientos funcionales del sistema describen con detalle la función de éste, sus entradas, salidas y excepciones. Los requerimientos funcionales para un sistema software se pueden expresar de diferentes formas.

Tabla 1: Requisitos Funcionales

Requisitos Funcionales	
RF1:Autenticar Usuario	
RF2: Insertar Persona	RF7: Generar reportes por categoria docente
RF3:Modificar Persona	RF8:Generar reporte por área.
RF4:Eliminar Persona	RF9:Generar reporte por expediente
RF5:Listar Persona	RF10:Generar reporte por cargo.
RF12: Generar reporte por o organización	RF11: Generar reporte por residencia

Requisitos no funcionales

Los requisitos no funcionales son las restricciones de los servicios o funciones ofrecidas por el sistema, restringen el espacio de posibles soluciones. Estos requisitos se basan en las características del sistema y en la carga que se espera tenga que soportar el mismo. Son características generales con las que debe cumplir el producto. A continuación se describen los requisitos no funcionales del sistema.

RNF1: Interfaz externa: Deberá contar con una interfaz sencilla que les muestre a los usuarios el producto.

RNF2: Requisito de software: El sistema operativo que soporta la solución es Ubuntu/Linux 11.04 o superior.

RNF3:Restricciones del diseño. El software deberá ser desarrollado en su totalidad con tecnologías y componentes de código abierto.

RNF4: Requisito de Seguridad: Cada usuario de acuerdo a su jerarquía en el sistema tendrá acceso a las funcionalidades permitidas.

RNF5: Requisito de seguridad: Garantizar la edición de la información solo al personal que tiene permiso a realizar esta actividad.

RNF6: Requisitos de Portabilidad:El sistema debe ser multiplataforma.

RNF7: Requisitos de hardware: El producto necesitará un servidor con procesador Dual-Core Xeon 1.60 GHZ, una RAM de 1 Gb como mínimo, un disco duro de 80 Gb de capacidad como mínimo y una tarjeta de red Ethernet Gigabit.

Resultados y discusiones

Los actores del negocio son roles que juegan los software, personas, grupos u organizaciones que interactúan con un negocio determinado, inicializa los casos de uso, en este caso el negocio se desarrolla en el Vicedecanato Administrativo de la Facultad de ciencias y tecnologías computacionales de la Universidad de las ciencias informáticas y en esta organización los actores del negocio se identifican a continuación

Tabla 2: Actores del negocio.

Actores	Descripción
Técnico de RRHH	Encargado de ejecutar todos los procesos referentes a recursos humano.

Un modelo de datos es un conjunto de conceptos y reglas que permiten representar las propiedades estáticas y dinámicas del mundo real. Es una descripción de algo conocido como contenedor de datos, así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos (Ríos, 2011). A continuación se representa el modelo de datos que se utiliza para el desarrollo de la automatización de la gestión de la información de los recursos humanos de la facultad CITEC. MAL

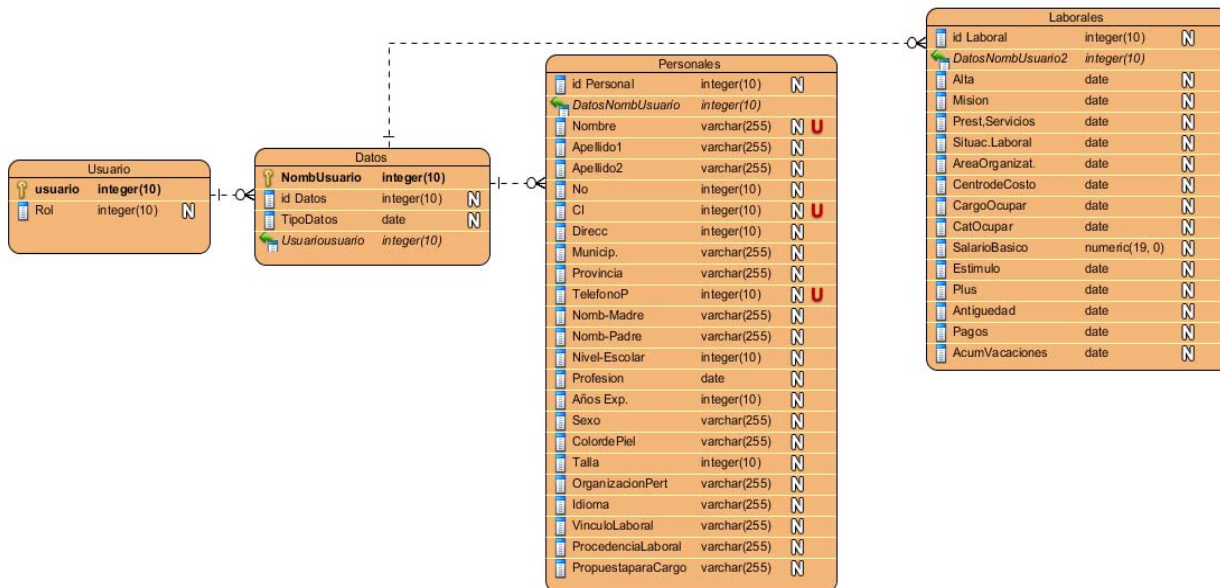


Figura 2: Modelo de datos.

Arquitectura de software

La arquitectura de software es la estructura de un sistema o bien la forma en que va a estar organizado este; incluye los elementos que lo conforman, sus propiedades visibles desde el exterior y las relaciones que existen entre ellos.

Patrones arquitectónicos.

Para la solución propuesta se decidió usar el framework de desarrollo Django, lo que implica una serie de decisiones de diseño. Django sigue una arquitectura Modelo-Vista-Controlador, solo que hace una adaptación de esta a Modelo-Vista-Plantilla (a partir de ahora MTV por sus siglas en inglés, Model Template View). Por tanto, el sistema propuesto hereda una arquitectura MTV.



Figura 3: Funcionamiento del MTV de Django. (Infante, 2012)

Modelo: Contiene toda la información sobre los datos. Cada una de las entidades de la base de datos se encuentra en el modelo en forma de clases de Python, y sus atributos se almacenan en variables con ciertos parámetros. También estos archivos poseen métodos, lo que permite indicar y controlar el comportamiento de los datos.

Vista: Es la capa de la lógica de negocios, contiene la lógica que accede al modelo y la delega a la plantilla apropiada. Esta capa sirve de “puente” entre el modelo y la plantilla, se presenta en forma de funciones de Python y su función principal es determinar qué datos serán visualizados en las plantillas.

Plantilla: Define la interfaz de las páginas web, o sea, cómo se van a mostrar los datos al usuario. También posee algunas etiquetas propias del framework Django.

URLConf: Posee todas las configuraciones de las URLs del sistema, a través de este archivo se puede controlar el despliegue de las vistas, acción conocida como mapeo de URLs.

Patrones de diseño

Patrones GRASP: Los patrones GRASP (por sus siglas en inglés, General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

Los patrones GRASP utilizados en la solución propuesta fueron:

Creador: este patrón es quien guía el proceso de asignación de responsabilidades relacionadas con la creación de objetos. Tiene el objetivo de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. Este patrón se pone de manifiesto en la clase

Bajo acoplamiento: Plantea que se debe poder reutilizar las funcionalidades de las distintas clases, con un nivel de dependencia mínima. Este patrón se evidencia en todas las aplicaciones web que funcionen sobre Django, pues cada pieza de las aplicaciones tiene un propósito clave, que puede modificarse sin afectar otras piezas. Por ejemplo, se puede cambiar la URL de cierta parte de la aplicación sin tener que afectar la implementación subyacente o se puede modificar el HTML de una página sin tener que tocar el código Python que la renderiza.

Alta cohesión: Asigna responsabilidades de manera tal que la cohesión siga siendo alta, o sea que las funcionalidades de las clases estén altamente relacionadas de forma tal que exista una colaboración entre ellas para compartir el esfuerzo y no caiga todo el peso sobre una única clase. Usar este patrón simplifica el mantenimiento y favorece el bajo acoplamiento. Este patrón se utiliza en la clase CrearPersona.

Controlador: Permite manejar todos los eventos del sistema, al servir de intermediario entre las interfaces y el algoritmo que las implementa. Este patrón se evidencia en la clase CrearPersona.

Patrones GOF : Patrones publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML. Existen 23 patrones GoF de los cuales 15 se utilizan con frecuencia.

Los patrones de diseño del grupo GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. A continuación, se describe el patrón GoF utilizado en la solución propuesta:

Decorador: Patrón estructural que extiende la funcionalidad de un objeto dinámicamente de manera tal que es transparente a sus clientes, utiliza una instancia de una subclase de la clase original que delega las operaciones al objeto original. Este patrón se evidencia en la clase Login require (decorador que trae django por defecto, para acceder a una clase)

Conclusiones

Una vez culminadas las actividades de planificación y diseño de la propuesta de solución se obtuvo una visión más clara y detallada del sistema que se desea desarrollar. Los requerimientos funcionales y no funcionales obtenidos a partir del proceso de identificación de requisitos, constituyeron elementos claves en la construcción de la propuesta de solución.

La utilización de los patrones de diseño permitió identificar aspectos importantes de la estructura del diseño del sistema web propuesto, lo que garantizó una mayor organización e hizo el código más legible. Los artefactos generados constituyeron no solo la base del sistema, sino que también permitieron una mejor comprensión del mismo

Referencias

- BEER, M. Y. C. Dirección de Recursos Humanos. Técnicas de Harvard Business School *CECSA, México*, 1990.
- FERNÁNDEZ, L. J. Gestión por competencias. Un modelo estratégico para la dirección de Recursos Humanos. *Madrid*., 2006, Prentice Hall, Pearson.
- GARZA, R. Concepción de un procedimiento utilizando herramientas cuantitativas para mejorar el desempeño empresarial *RII*, 2012, Vol.33(No.3).
- LARMAN, C. UML y Patrones *California: Prentice Hall*, 2005.
- MAR, O.; M. Y. L. VÁZQUEZ, *et al.* Modelo multicriterio multiexperto utilizando Mapa Cognitivo Difuso para la evaluación de competencias *Ciencias de la Información*, 2015a, 46(2): 17-22.
- MAR, O.; Y. Z. VÉLIZ, *et al.* Motor de inferencia decisional en sistema informático para la evaluación del desempeño *Revista Cubana de Ciencias Informáticas*, 2015b, 9(4): 16-29.
- MAR, O.; Y. ZULUETA, *et al.* Motor de inferencia decisional en sistema informático para la evaluación del desempeño *RCCI*, 2015c, Vol.9(No.4): 16-29.
- MONDY, R. Administración de los recursos humanos *México: Editorial Pearson Prentice-Hi*, 2005.
- PRESSMAN, R. S. Ingeniería de Software: Un enfoque práctico 7ma edición. , 2010.
- RODRÍGUEZ, J. F. O. Evaluación del desempeño del docente de la unidad educativa prof. fernando ramírez *REVISTA ICONO*, 2010, No14, Vol. 2, pp. 58-70. ISSN 1697-8293.
- SALAS, R. S. Propuesta de estrategia para la evaluación del desempeño laboral de los médicos en Cuba *Educ Med Super*, 2010, v.24 n.3 jul.-sep. ISSN 1561-2902.
- SANTOS, A. C. Evaluando desempeños: alineamiento estratégico y productividad. *FORUM Empresarial*, 2012, Vol. 17, Núm. 1. pp. 1-30. ISSN 1541-8561.