



Universidad de las Ciencias Informáticas

Facultad 1

*Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas*

*Título: Sistema de Gestión de la Información de Recursos Humanos y Tecnológicos
para el Centro de Identificación y Seguridad Digital.*

Autores:

Yam Brayam Beltrán Rodríguez

Rocio Romero Alvarez

Tutores:

Ing. Yaicel Díaz Cordova

Ing. Hiraldo Manuel Anaya González

Co-Tutor: Ing. Denis Luis García Ruibal

La Habana, Junio de 2013

Declaramos ser los autores del trabajo titulado Sistema de Gestión de la Información de Recursos Humanos y Tecnológicos para el Centro de Identificación y Seguridad Digital y se le autoriza a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los ____ días del mes de ____ del año 2013.

Rocío Romero Alvarez

Yam Brayam Beltrán Rodríguez

Ing. Yaicel Díaz Cordova

Ing. Hiraldo Manuel Anaya González

Opinión de los tutores

Agradecimientos

Queremos agradecer a todas aquellas personas, que de una forma u otra contribuyeron con al desarrollo de este trabajo, aportando su granito de arena para lograr que nuestro sueño se hiciera realidad, en especial a los profesores Yadier Perdomo Cuevas, Jorge Landrian Garcia, Yendry Machado Garcia y a nuestro compañero de laboratorio Adrián Hernández Barrios. A nuestros tutores Yaicel Diaz Cordova y Hiraldo Manuel Anaya González por su disposición y ayuda incondicional cuando los hemos necesitado. A todos los involucrados de una forma u otra con nuestro proceso de formación durante estos cinco años de universidad.

Yam Brayam Beltrán Rodríguez

Agradezco por sobre todo a mi mamá por ese apoyo incondicional en estos cinco años, por haber luchado tanto para que yo estuviera donde hoy estoy, por ser ese angelito que nunca salió de mí cabeza diciéndome dale mí niño no te rindas yo sé que tú puedes. A mi papá, mi hermana, mis abuelas, mi sobrinito, y a toda la familia en generar por apoyarme y estar a mi lado en momentos difíciles de esta carrera dándome su apoyo y confianza en mí. A mis compañeros de aula, nunca olvidare los momentos que pasamos juntos buenos y malos me alegro de haber conocido amigos como ustedes. A Arlyn por el apoyo durante toda la carrera, por seguir ahí cuando nadie hubiera seguido, ayudándome y confiando en mí.

Rocío Romero Alvarez

Agradezco en primer lugar al ser que más quiero en este mundo, mi madre, por brindarme su apoyo en todo momento, en las buenas y en las malas, en las altas y en las bajas, por ser la luz que ha guiado mi camino por el bien, por ser madre y padre a la vez y por regalarme la vida que hoy tengo. Agradezco además a mi hermana, a mi hermano, a mi papá, a mis abuelitos y tías por regalarme la dicha de ser parte de esa familia tan linda que hoy somos. A todas aquellas personas que contribuyeron de una manera u otra al desarrollo de este trabajo de diploma.

Dedicatoria

Yam Brayam Beltrán Rodríguez

Quiero dedicarle este trabajo de diploma a todas las personas que me quieren mucho y que sin ellas este sueño un hubiese sido posible, en especial a mis padres, por ser los mejores del mundo, por luchar a mi lado para lograr juntos este sueño, por levantarme tantas veces cuando caía, a ti mami porque en estos cinco años viviste cada momento de alegría y sufrimiento a mí lado, porque desde el primer año ya para ti era ingeniero pese a las dificultades por eso y por mucho más te dedico este sueño que es más tuyo que mío. A mi hermana, por esos concejos en momentos de desesperación, por creer en mí cuando más lo necesitaba. A toda mi familia en generar a mis abuelas que ojalá siempre estén con migo, en fin a todos, todos dieron un granito de arena para lograr formar esta mejor persona que soy hoy.

Rocío Romero Alvarez

Le dedico este triunfo en primer lugar a mi madre, la cual fue el eslabón principal de apoyo en esta lucha por alcanzar el éxito de convertirme en una Ingeniera en Ciencias Informáticas, a mi hermano , mi hermana ,mi papá , mis abuelos y familia en general. A mis amigas y amigos durante estos cinco largos años, a mis profesores desde los inicios del círculo infantil hasta el quinto año de universidad y a la vida por darme esta grandiosa oportunidad.

Resumen

En la actualidad la gestión de la información de los recursos humanos y tecnológicos afronta nuevas exigencias, lo que hace que esté en constante cambio y transformación. A partir de este desarrollo, muchas organizaciones han enfocado sus esfuerzos en encontrar nuevas maneras de perfeccionar el control de los recursos humanos y tecnológicos, donde la elaboración de sistemas que contribuyan a llevar una detallada gestión del personal y de los recursos informáticos juegue un papel importante. Como parte de este proceso de perfeccionamiento, en el Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas (UCI), surge la necesidad de la implementación de un sistema que facilite la gestión de la información de los recursos humanos y tecnológicos con los que cuenta. Este sistema posee un aporte práctico, pues se contará con una herramienta capaz de gestionar y centralizar la información de los recursos tanto tecnológicos como humanos, lo que permite tener una mayor gestión en la organización y control de los recursos.

Palabras claves: información, recursos humanos, recursos tecnológicos

Índice

Introducción	1
Capítulo I: Fundamentación teórica	5
Introducción.....	5
1.1. Conceptos básicos	5
1.2. Estado del arte.....	7
1.3. Metodologías de desarrollo.....	12
1.3.1. Metodologías pesadas o robustas.....	12
1.3.2. Metodologías ágiles	14
1.3.3. Selección de la metodología	15
1.4. Selección de las herramientas	16
1.4.1. Marco de trabajo: django en su versión 1.4.2.....	16
1.4.2. Lenguaje de programación: Python en su versión 2.7.3.....	16
1.4.3. Entorno Integrado de Desarrollo: PyCharm en su versión 2.0.1	17
1.4.4. XHTML y CSS.....	17
1.4.5. Gestor de base de datos: PostgreSQL en su versión 9.1.7.1	17
1.4.6. Servidor web: Apache en su versión 2.2.20	18
1.4.7. Herramienta de modelado: Visual Paradigm en su versión 6.4	18
Conclusiones.....	18
Capítulo 2: Características del sistema	20
Introducción.....	20
2.1. Objeto de automatización	20
2.2. Información que se maneja.....	21
2.3. Propuesta del sistema	21
2.3.1. Modelo de dominio.....	21
2.4. Personas relacionadas con el sistema.....	23
2.5. Exploración.....	23
2.5.1. Lista de reserva del producto (LRP)	23
2.5.2. Funcionalidades del sistema	24
2.5.3. Historias de usuario.....	27
2.5.4. Arquitectura del sistema.....	29
2.5.5. Patrones del diseño.....	30
2.5.6. Diagrama de clases.....	34
2.5.7. Modelo de dato	36
2.6. Planificación	36
2.6.1. Estimación de esfuerzo por cada una de las historias de usuario identificadas	36
2.6.2. Plan de Iteraciones	37
2.6.3. Plan de entrega.....	38
2.7. Diseño	38
2.7.1. Tarjetas CRC	39
2.7.2. Tareas de ingeniería	40
Conclusiones.....	41
Capítulo III: Implementación y Pruebas.....	42
Introducción.....	42
3.1. Implementación	42
3.1.1. Estándares de codificación.....	42
3.1.2. Tareas de implementación para cada historia de usuario.....	45
3.1.3. Diagrama de componentes	46
3.1.4. Interfaces del sistema	47

3.1.5. Diagrama de despliegue	49
3.2. Pruebas	50
3.2.1. Pruebas de aceptación.....	50
3.2.2. Pruebas de aceptación para cada historia de usuario	50
Conclusiones.....	52
Conclusiones Generales.....	53
Recomendaciones	54
Bibliografía Referenciada.....	55
Bibliografía Consultada	58
Glosario de Términos.....	61
Anexos.....	62
Anexo 1. Entrevista.	62
Anexo 2. Historias de usuarios.	62
Anexo 3. Diagramas de clases.	65
Anexo 4. Modelo de dato.....	70
Anexo 5 Tarjetas CRC.....	76
Anexo 6. Tareas de ingeniería.....	78
Anexo 7. Tareas de implementación.	84
Anexo 8. Interfaces del sistema.....	90
Anexo 9. Casos de pruebas de aceptación.	92

Índice de Figuras

Figura 1 Fases e Iteraciones de la Metodología RUP 13

Figura 2 Diagrama del modelo de dominio 22

Figura 3 Patrón de diseño en las url. 31

Figura 4 Patrón de diseño utilizando la clase logging. 32

Figura 5 Patrón de diseño utilizando el __unicode__. 32

Figura 6 Patrón de diseño utilizando vista genérica. 33

Figura 7 Patrón de diseño en las plantillas..... 34

Figura 8 Patrón de diseño en las plantillas..... 34

Figura 14 Diagrama de clases AutenticarUsuario..... 35

Figura 15 Diagrama de clases MostrarReporte..... 36

Tabla 11 Estimación de esfuerzo de la historias de usuario. 37

Tabla 12 Plan de iteración..... 38

Tabla 13 Entrega del producto en cada iteración. 38

Figura 22 Codificador de comentario de implementación..... 43

Figura 23 Codificador de comentario de documentación. 43

Figura 24 Codificador de línea larga..... 44

Figura 25 Codificador de declaraciones..... 44

Figura 26 Codificador de declaraciones de clases de interfaces..... 45

Figura 27 Codificador de sentencia de retorno. 45

Figura 28 Diagrama de Componentes..... 47

Figura 29 Interfaz Autenticar usuario 48

Figura 30 Interfaz Mostrar reporte de estudiantes 49

Figura 33 Diagrama de Despliegue..... 49

Figura 9 Diagrama de clases GestionarEstudiante..... 65

Figura 10 Diagrama de clases GestionarMaquina 66

Figura 11 Diagrama de clases GestionarUsuario 67

Figura 12 Diagrama de clases GestionarTrabajador 68

Figura 13 Diagrama de clases GestionarProfesor 69

Figura 16 Modelo de dato parte 1 70

Figura 17 Modelo de dato parte 2 71

Figura 18 Modelo de dato parte 3 72

Figura 19 Modelo de dato parte 4 73

Figura 20 Modelo de dato parte 5 74

Figura 21 Modelo de dato parte 6 75

Figura 31 Interfaz Mostrar reporte departamento del profesor..... 90

Figura 32 Interfaz Mostrar reporte Departamento del profesor. 91

Índice de Tablas

Tabla 1 Personas relacionadas con el sistema.....	23
Tabla 2 Funcionalidades del Sistema.....	26
Tabla 3 Historia de usuario 1. Autenticar usuario.....	28
Tabla 4 Historia de usuario 2. Administrar usuario.....	28
Tabla 5 Historia de usuario 3. Gestionar trabajadores.....	29
Tabla 11 Estimación de esfuerzo de la historias de usuario.....	37
Tabla 12 Plan de iteración.....	38
Tabla 13 Entrega del producto en cada iteración.....	38
Tabla 14 Tarjeta CRC 1. TrabajadorListView.....	39
Tabla 10 Tarjeta CRC 2. Trabajador.....	39
Tabla 15 Tarjeta CRC 3. Persona.....	40
Tabla 22 Tarea de ingeniería 1. Autenticar usuario.....	40
Tabla 23 Tarea de ingeniería 2. Administrar usuario.....	41
Tabla 41 Tarea 1 de implementación. Autenticar usuario.....	46
Tabla 42 Tarea 2 de implementación Gestionar usuario.....	46
Tabla 43 Tarea 3 de implementación. Gestionar estudiantes.....	46
Tabla 60 Caso de Prueba de Aceptación 1. Autenticar usuario.....	51
Tabla 61 Caso de Prueba de Aceptación 2. Gestionar usuario.....	51
Tabla 62 Caso de Prueba de Aceptación 3. Gestionar trabajadores.....	51
Tabla 6 Historia de usuario 4. Gestionar máquina.....	63
Tabla 7 Historia de usuario 5. Mostrar reporte.....	63
Tabla 8 Historia de usuario 6. Gestionar estudiante.....	64
Tabla 9 Historia de usuario 7. Gestionar profesor.....	64
Tabla 15 Tarjeta CRC 2. PersonaListView.....	76
Tabla 16 Tarjeta CRC 3. EstudianteListView.....	76
Tabla 17 Tarjeta CRC 4. ProyectoProfesoresListView.....	76
Tabla 18 Tarjeta CRC 5. ListadoProfeDepartamentoListView.....	76
Tabla 19 Tarjeta CRC 6. ListadoProfeRolListView.....	77
Tabla 20 Tarjeta CRC 9. Maquina.....	77
Tabla 21 Tarjeta CRC 10. Estudiante.....	78
Tabla 24 Tarea de ingeniería 3. Gestionar estudiantes.....	78
Tabla 25 Tarea de ingeniería 4. Gestionar estudiantes.....	78
Tabla 26 Tarea de ingeniería 5. Gestionar estudiantes.....	79
Tabla 27 Tarea de ingeniería 6. Gestionar estudiantes.....	79
Tabla 28 Tarea de ingeniería 7. Gestionar trabajador.....	79
Tabla 29 Tarea de ingeniería 8. Gestionar trabajador.....	80
Tabla 30 Tarea de ingeniería 9. Gestionar trabajador.....	80
Tabla 31 Tarea de ingeniería 10. Gestionar trabajador.....	80
Tabla 32 Tarea de ingeniería 11. Gestionar máquina.....	81
Tabla 33 Tarea de ingeniería 12. Gestionar máquina.....	81
Tabla 34 Tarea de ingeniería 13. Gestionar máquina.....	81
Tabla 35 Tarea de ingeniería 14. Gestionar máquina.....	82
Tabla 36 Tarea de ingeniería 15. Mostrar reporte.....	82
Tabla 37 Tarea de ingeniería 16. Gestionar profesor.....	82
Tabla 38 Tarea de ingeniería 17. Gestionar profesor.....	83

Tabla 39 Tarea de ingeniería 18. Gestionar profesor.....	83
Tabla 40 Tarea de ingeniería 19. Gestionar profesor.....	83
Tabla 44 Tarea 4 de implementación. Gestionar estudiantes.....	84
Tabla 45 Tarea 5 de implementación. Gestionar estudiantes.....	84
Tabla 46 Tarea 6 de implementación. Gestionar estudiantes.....	85
Tabla 47 Tarea 7 de implementación. Gestionar trabajador.....	85
Tabla 48 Tarea 8 de implementación. Gestionar trabajador.....	85
Tabla 49 Tarea 9 de implementación. Gestionar trabajador.....	86
Tabla 50 Tarea 10 de implementación. Gestionar trabajador.....	86
Tabla 51 Tarea 11 de implementación. Gestionar máquina.....	86
Tabla 52 Tarea 12 de implementación. Gestionar máquina.....	87
Tabla 53 Tarea 13 de implementación. Gestionar máquina.....	87
Tabla 54 Tarea 14 de implementación. Gestionar máquina.....	87
Tabla 55 Tarea 15 de implementación. Mostrar reporte.....	88
Tabla 56 Tarea 16 de implementación. Gestionar profesor.....	88
Tabla 57 Tarea 17 de implementación. Gestionar profesor.....	88
Tabla 58 Tarea 18 de implementación. Gestionar profesor.....	89
Tabla 59 Tarea 19 de implementación. Gestionar profesor.....	89
Tabla 63 Caso de Prueba de Aceptación 4. Gestionar trabajadores.....	92
Tabla 64 Caso de Prueba de Aceptación 5. Gestionar trabajadores.....	93
Tabla 65 Caso de Prueba de Aceptación 6. Gestionar trabajadores.....	93
Tabla 66 Caso de Prueba de Aceptación 7. Gestionar máquina.....	93
Tabla 67 Caso de Prueba de Aceptación 8. Gestionar máquina.....	94
Tabla 68 Caso de Prueba de Aceptación 9. Gestionar máquina.....	94
Tabla 69 Caso de Prueba de Aceptación 10. Gestionar máquina.....	95
Tabla 70 Caso de Prueba de Aceptación 11. Mostrar reporte.....	95
Tabla 71 Caso de Prueba de Aceptación 12. Gestionar estudiante.....	96
Tabla 72 Caso de Prueba de Aceptación 13. Gestionar estudiante.....	96
Tabla 73 Caso de Prueba de Aceptación 14. Gestionar estudiante.....	97
Tabla 74 Caso de Prueba de Aceptación 15. Gestionar estudiante.....	97
Tabla 75 Caso de Prueba de Aceptación 16. Gestionar profesor.....	98
Tabla 76 Caso de Prueba de Aceptación 17. Gestionar profesor.....	98
Tabla 77 Caso de Prueba de Aceptación 18. Gestionar profesor.....	99
Tabla 78 Caso de Prueba de Aceptación 19. Gestionar profesor.....	99
Tabla 79 Caso de Prueba de Aceptación 20. Gestionar usuario.....	100
Tabla 80 Caso de Prueba de Aceptación 21. Gestionar usuario.....	100
Tabla 81 Caso de Prueba de Aceptación 22. Gestionar usuario.....	101

Introducción

En el mundo, la gestión del conocimiento por parte de las empresas adquiere nuevas características, determinadas por la calidad y la gestión de la información. Esta última, como ciencia, busca optimizar la manera en que colabora el personal laboral en aras de alcanzar los objetivos principales de su organización, tarea que exige estudio, dedicación y persistencia (1); implicando así un conjunto de procedimientos, acciones integradas y procesos normalizados que responden al ciclo de vida de la información. En las organizaciones más modernas cohabitan, indisolublemente ligadas, la gestión de la información, del conocimiento y de la calidad; donde la ética, la motivación y un correcto desempeño rinden incrementos constantes en sus resultados y en el reconocimiento de las organizaciones.

En Cuba esta realidad se impone cobrando fuerzas, enfrentándose al reto de adaptar las organizaciones al cambio, con una filosofía de mejoramiento continuo de los procesos (2). A partir del avance de las tecnologías de la información y las comunicaciones (TIC's), y el auge que ha tomado la informatización de las instituciones cubanas, es preciso que se desarrollen sistemas para la gestión de la información, haciéndose necesario que en los centros educacionales se apliquen esas nuevas técnicas para perfeccionar su funcionamiento, en cuanto a calidad y rapidez de procesamiento.

La UCI, es uno de los centros universitarios del país, que cuenta con una matrícula numerosa de estudiantes, profesores, trabajadores y equipos técnicos, vinculados a la producción de software de calidad. Como parte de su infraestructura productiva está el CISED, dedicado al desarrollo de soluciones integrales, productos y servicios en el campo de la identificación y la seguridad digital; que cuenta con varios departamentos, grupos, y un equipo de especialistas de alto nivel, con gran experiencia y reconocimiento a nivel nacional e internacional. Este centro posee cuantiosa suma de estudiantes, profesores y trabajadores, además de un conjunto de recursos informáticos para el desarrollo de las tareas productivas; donde el volumen de información que se maneja es cada vez mayor, lo que conlleva a que la búsqueda de ésta en determinado momento se torne un poco lenta.

Actualmente en el centro, la información que se desea gestionar se halla distribuida en diferentes fuentes, sin que los elementos que tienen en común cada una de ellas estén asociados entre sí; además es preciso la planificación y gestión de proyectos productivos y el trabajo docente investigativo, teniendo en cuenta la disponibilidad de los recursos humanos y tecnológicos. El acceso a la información almacenada en ocasiones provoca pérdida de tiempo, debido a que no se encuentra totalmente centralizada, igualmente existen informaciones que son enviadas mediante correo electrónico sin que quede constancia de la misma. La información que se maneja consiste en mayor medida en los datos de cada persona que radican en el centro, así como su ubicación dentro del mismo, además de las características de todos los

equipos tecnológicos existentes. Los subprocesos en los que puedan interactuar algunos de estos elementos también son considerados datos importantes en dicha gestión.

Luego de analizar la situación problemática descrita anteriormente se desprende el siguiente **problema a resolver**: ¿Cómo mejorar la gestión de la información de los recursos humanos y tecnológicos en las diferentes áreas del CISED?

Para ello se define como **objeto de estudio**: el proceso de gestión de la información de los recursos humanos y tecnológicos.

Para la solución del problema se formula como **objetivo general**: desarrollar un sistema informático que facilite la gestión de la información de los recursos humanos y tecnológicos del CISED que garantice la disponibilidad y centralización de sus principales procesos, el cual se desglosa en los siguientes **objetivos específicos**:

- Realizar un estudio teórico relacionado con diferentes sistemas de gestión de la información, orientado a identificar y seleccionar las mejores prácticas y soluciones existentes de este tipo.
- Identificar las necesidades de información relacionada con los recursos humanos y tecnológicos del centro.
- Implementar un sistema que gestione la información de los recursos humanos y tecnológicos en el CISED.
- Realizar pruebas al sistema para verificar su correcto funcionamiento y calidad del mismo.

Como **idea a defender se plantea**: si se realiza el análisis, diseño e implementación de un sistema de gestión de la información de recursos humanos y tecnológicos para el CISED, se garantizará una centralización y disponibilidad de la información como respaldo de los procesos principales de la entidad, así como una organización automática de la misma que permita al usuario un rápido acceso a ésta.

Para dar cumplimiento a los objetivos antes propuestos se definen las siguientes **tareas de investigación**:

- Análisis de las tendencias actuales relacionadas a los sistemas de gestión de la información.
- Definición de la metodología, tecnologías y herramientas a utilizar para el desarrollo de la propuesta de solución.
- Definición de la arquitectura del software y los patrones del diseño.
- Identificación y realización del levantamiento de la información a automatizar.
- Realización y descripción de los artefactos generados en la etapa del análisis.
- Realización del diagrama de clases y el modelo de dominio.
- Desarrollo y diseño de la interfaz gráfica del sistema de gestión, e implementación de las funcionalidades necesarias para cumplir con la seguridad del sistema.

- Implementación de las funcionalidades necesarias para mapear la información del sistema de gestión para el control de los recursos humanos y tecnológicos.
- Realización de pruebas funcionales al sistema de gestión para el control de los recursos humanos y tecnológicos.

Métodos científicos a utilizar:

Se considera método científico de investigación a una serie de pasos sistemáticos e instrumentos que conducen a un conocimiento científico.

Métodos teóricos

- **Analítico–sintético** facilitará el procesamiento de la información obtenida en la investigación realizada sobre el objeto de estudio. Se realizará un estudio profundo de los conceptos, metodologías y herramientas a utilizar en el desarrollo de la investigación basándose en la revisión de documentos, artículos e informes.

Métodos empíricos

- **Observación** proporcionará una visión realista y objetiva de los procesos, conceptos y definiciones presentes en el negocio.
- **Entrevista** permitirá obtener la información necesaria relacionada con los problemas presentes en los procesos de gestión de información de los recursos humanos y tecnológicos, mediante entrevistas realizadas a distintos directivos del CISED. **(ver Anexo 1)**

Justificación de la investigación.

Los sistemas de gestión de la información en la actualidad se han convertido para cualquier empresa u organización a nivel mundial en una herramienta necesaria para el desempeño eficiente de cada una de sus funciones. La relevancia de esta investigación se centra en la importancia que tiene en conocimiento y práctica la creación de un **Sistema de Gestión de la Información de los Recursos Humanos y Tecnológicos** para el centro CISED, ya que le brindará un mejor control y funcionamiento de toda la información relevante con la cual cuenta el centro en la actualidad, además de poder consultarla y modificarla de manera instantánea.

Aporte práctico de la investigación:

Con el desarrollo del sistema se contará con una herramienta capaz de gestionar y centralizar la información de los recursos tanto tecnológicos como humanos, lo que permitirá tener una factible y rápida gestión en la organización y control de los mismos.

La estructura de este documento consta de tres capítulos que abordarán todo el proceso de desarrollo del sistema:

Capítulo I: Fundamentación teórica: En este capítulo se abordarán los principales conceptos, que constituyen la base teórica de la presente investigación. Se realizará un estudio del estado del arte de diferentes sistemas homólogos existentes a nivel nacional e internacional, se especificará la metodología elegida para guiar el proceso de desarrollo del software, además de seleccionarse las herramientas y tecnologías que se van a utilizar.

Capítulo II: Características del sistema: En este capítulo además de realizarse una descripción general de la propuesta del sistema y su funcionamiento, se puntualizan los principales aspectos relacionados con el diseño del mismo. Se capturarán los requisitos funcionales y no funcionales. Se definirá la arquitectura que organice la lógica del sistema, además de especificarse los patrones del diseño que se van a aplicar.

Capítulo III: Implementación y prueba: En este capítulo se presentarán los aspectos relacionados con la implementación de la solución propuesta. Además de construirse los casos de prueba de aceptación en base a los requisitos definidos inicialmente para el desarrollo de la aplicación, se realizarán las principales pruebas a la aplicación para verificar que responda a un correcto funcionamiento.

Capítulo I: Fundamentación teórica

Introducción

En el presente capítulo se definen diferentes conceptos importantes relacionados estrechamente con la problemática a desarrollar, se realiza una breve comparación y valoración de algunos de los sistemas existentes en Cuba y en el mundo, similares al que se desea implementar, además de efectúa un estudio que enmarca la fundamentación de la utilización de las herramientas y lenguajes que serán utilizados en el análisis, diseño e implementación de la aplicación.

1.1. Conceptos básicos

A continuación se exponen diferentes conceptos que sirven sobre los cuales se fundamentan los aspectos teóricos del sistema que se desea implementar y que llevarán a una mayor comprensión de la investigación realizada.

- **Información:** Conjunto de datos acerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen un significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo. (3) Está constituida por un grupo de datos ya supervisados y ordenados, que sirven para construir un mensaje basado en un cierto fenómeno o ente. La misma permite resolver problemas y tomar decisiones, ya que su aprovechamiento racional es la base del conocimiento. Además es un recurso que otorga significado o sentido a la realidad, ya que mediante códigos y conjuntos de datos, da origen a los modelos de pensamiento humano. (4)
- **Gestión:** El concepto de gestión hace referencia a la acción y a la consecuencia de administrar o gestionar. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera. Administrar, por otra parte, abarca las ideas de gobernar, disponer, dirigir, ordenar u organizar una determinada cosa o situación. La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio. (5)
- **Gestión de la información:** Proceso mediatizado por un conjunto de actividades que permiten la obtención de información, lo más pertinente, relevante y económica posible, para ser usada en el desarrollo y el éxito de una organización. Es ir en busca de nuevos significados, análisis, aplicar el principio de que el todo, es más que la suma de las partes. Es producir un impacto en el ambiente de cualquier organización. Es un proceso que debe estar presente en cada uno de los pasos de la organización y requiere acción, decisión y evaluación.

Mediante la gestión de la información se proporcionan los recursos de información necesarios para una buena toma de decisiones, se desarrollan nuevos conocimientos que posibilitan calidad y eficiencia en los servicios y productos de las organizaciones. (6)

- **Recurso:** Es un medio de cualquier clase que permite satisfacer una necesidad o conseguir aquello que se pretende. Los recursos se pueden clasificar en tangibles (computadoras, personal) e intangibles (información). (7)
- **Recursos humanos:** Los recursos humanos (RR.HH.) son el conjunto de empleados y colaboradores que trabajan en una empresa u organización. Con este nombre también se denomina al departamento o la persona que se encarga de seleccionar, contratar, formar y retener a los trabajadores de una empresa entre otras funciones. La labor más básica sería la gestión administrativa tramitando altas y bajas de contratos laborales, nóminas y toda la documentación que genera la relación entre empresa y trabajador.

Una de las tareas del departamento de RR.HH. es la de utilizar de la mejor manera posible a los empleados bajo criterios de coste, buen clima de trabajo y desarrollo del potencial humano de cada trabajador. La intención es que el empleado se sienta satisfecho con su trabajo y que se involucre en alcanzar las metas de la empresa. También se encargarían de la planificación, reclutamiento y selección del personal. Definiendo las necesidades de plantilla de la empresa, estableciendo el perfil profesional para cada puesto de trabajo y reclutando y seleccionando a los candidatos para su ingreso en la organización. (8)

- **Recursos tecnológicos:** Un recurso tecnológico es un medio que se vale de la tecnología para cumplir con su propósito, y se pueden clasificar en tangibles (como una computadora, una impresora u otra máquina) o intangibles (un sistema, una aplicación virtual).

En la actualidad, los recursos tecnológicos son una parte importante de las empresas o de los hogares. La tecnología se ha convertido en un aliado clave para la realización de todo tipo de tareas. Los recursos tecnológicos ayudan a desarrollar las operaciones cotidianas de la empresa, desde la producción hasta la comercialización, pasando por las comunicaciones internas y externas o cualquier otra faceta. (9)

- **Sistema:** Un sistema es un módulo ordenado de elementos que se encuentran interrelacionados y que interactúan entre sí. (10)
- **Sistemas de información:** Conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de una organización. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general, los cuales procesan la información y la distribuyen de manera adecuada, buscando satisfacer las

necesidades de la organización. Su objetivo primordial es apoyar la toma de decisiones y controlar todo lo que en ella ocurre. Existen dos tipos de sistema de información, los formales y los informales; los primeros utilizan como medio para llevarse a cabo estructuras sólidas como ordenadores, los segundos son más artesanales y usan medios más antiguos como el papel y el lápiz o el boca a boca. Desde un punto de vista empresarial, los sistemas de información pueden clasificarse de diversas formas. Existen, por ejemplo, sistemas de información gerencial (con el fin de resolver conflictos en empresas), sistemas de procesamiento de transacciones (que se encargan de manejar la información en el contexto de los intercambios comerciales), sistemas de información ejecutiva (para los directivos), sistemas de soporte a decisiones (analizan los distintos factores que hacen al negocio para decidir qué rumbo tomar), sistemas de automatización de oficinas (aplicaciones que ayudan en el trabajo administrativo) y sistemas expertos (que emulan el comportamiento de un especialista en un dominio concreto). El concepto de sistema de información suele ser utilizado como sinónimo de sistema de información informático, aunque no son lo mismo. Este último pertenece al campo de estudio de la tecnología de la información y puede formar parte de un sistema de información como recurso material. (11)

- **Sistemas de gestión:** Un sistema de gestión es un conjunto de etapas unidas en un proceso continuo que permite trabajar ordenadamente una idea hasta lograr mejoras y su continuidad. Se establecen cuatro etapas en este proceso, que hacen de este sistema, un proceso circular virtuoso, pues en la medida que el ciclo se repita recurrente y recursivamente, se logrará en cada ciclo, obtener una mejora. Las cuatro etapas del sistema de gestión son: etapa de ideación, planeación, implementación y etapa de control. Es el proceso dentro del segmento de la gestión de información que sirve al interés corporativo. Se persigue asociar la información para beneficio de la organización en su totalidad mediante la explotación, desarrollo y optimización de los recursos de información; generalmente se manifiestan en las metas y objetivos corporativos. Por tanto, la gestión de recursos de información, es el vínculo gerencial que conecta los recursos de información corporativos con las metas y objetivos de la organización. (12)

1.2. Estado del arte

El presente epígrafe tiene como propósito realizar un análisis de aquellos sistemas o soluciones similares existentes en Cuba y en el resto del mundo, que podrían darle en cierta medida solución al problema planteado anteriormente.

- **Gv Sistemas**

La empresa Gv Sistemas de Argentina, fue creada en el año 1981 para atender los requerimientos de empresas en el área de RR.HH. y Liquidación de Haberes. Cuenta con un potente motor de cálculo (GVWIN) que le facilita a dicha área la realización de funciones menos operativas y más estratégicas para la toma de decisiones, gestionando operativa y administrativamente todos los procesos de recursos humanos de la empresa. Entre los servicios que brinda dicha institución, los cuales gestiona a través del motor de cálculo GVWIN, se encuentran la búsqueda y selección del personal, evaluaciones del personal a través de test psicológicos así como evaluaciones de desempeño profesional. (13)

- **Aplicaciones y Proyectos Informáticos (API)**

También en España existen empresas que gestionan y administran los recursos humanos de las entidades. En el caso de API, se creó en Barcelona en 1987 con la idea de dar una solución integrada en el ámbito informático, sin embargo con el tiempo se fue especializando en el área de gestión del personal y recursos humanos siendo en la actualidad una de las empresas líderes en ese sector. Dicha empresa, junto a IS3 (Soluciones Integrales de Ingeniería y Seguridad) y CTT (Centro de Transferencia de Tecnología) se han fusionado al grupo Conektia, aunando sus servicios y brindándolos a diversidad de clientes. De esta unión e intercambio de ideas y servicios surgió la plataforma Punto Ge, desarrollada con tecnologías de última generación por los profesionales que integran el grupo Conektia.

Punto Ge es una plataforma integrada para la gestión del personal y recursos humanos, tecnológicamente abierta, desarrollada en .Net, multisistema y modular, conformada junto a la aplicación por un set de servicios de consultoría, asesoría, gestión de proyectos y otros. Pensada para adaptarse a los cambios de las organizaciones, totalmente parametrizable a los requerimientos de las empresas, lo cual es de suma importancia porque constituye la oportunidad de ampliar el abanico de soluciones a las necesidades de los departamentos de los recursos humanos. (13)

- **Sistema de información documental diseñado para la gestión de correspondencia de la universidad de Antioquia**

El sistema de información documental, como organización, se caracteriza por controlar, hacer accesible la información y procesarla, además de derivar nuevas formas de información y maneras de representarla, será trabajo del documentalista, el desarrollo, la operatividad y la gestión del sistema de información, empleando el tipo de técnicas y tecnologías adecuadas para la transferencia de la misma.

Está desarrollado con un enfoque al servicio del usuario, en la universidad de Antioquia, tiene en cuenta el contexto institucional con la variedad de necesidades del usuario y la alta demanda de información, lo que hace necesario que la comunicación documental sea un continuo proceso fluyente, ágil y oportuno. Contiene un módulo de gestión de acuerdo a los objetivos estratégicos de la institución el cual integra los

elementos que hacen parte del contexto organizacional y de los recursos, la gestión de contenidos y de calidad, los soportes tecnológicos y los formatos de la información y de los conocimientos. (14)

- **Hum&Select**

HUM&SOFTWARE es una empresa especializada en el desarrollo de sistemas de información para recursos humanos. Su principal producto es el sistema Hum&Select, el cual es un paquete computacional que ha sido especialmente diseñado para satisfacer las necesidades del proceso de reclutamiento, selección y evaluación psicométrica del personal en las empresas así como para los recursos de salida dedicados a estas actividades. Su funcionamiento radica en los siguientes pasos

- **Captura de información:** El sistema cuenta con una interfaz sencilla, amigable e intuitiva la cual facilita el proceso de registro y uso de la información. Adicionalmente se cuenta con el modo de auto-captura (vía internet o en su red local) por medio del cual el mismo candidato registra su información sin la necesidad de requerir del tiempo del personal de la organización.
- **Almacenamiento de información:** Toda la información queda almacenada en los archivos electrónicos del sistema. Incluso las fotografías y documentos anexos (currículo en *word*, documentos escaneados) sin la necesidad de requerir el manejo complicado de archivos, el Hum&Select se encarga de todo.
- **Búsqueda de candidatos:** permite localizar fácil y rápidamente a los candidatos que cuenten con las características requeridas. Por medio de la definición de perfiles de búsqueda se especifican perfiles precisos, tan detallados como sea necesario (existen 18 criterios de selección disponibles) para identificar exactamente a aquellas personas que se apegan mejor al puesto requerido.
- **Requisiciones de personal:** El sistema permite controlar eficientemente cada requisición de personal y facilitar el seguimiento a detalle del proceso y actividades a realizar con cada candidato.
- **Flujo de información:** permite la consulta en línea acerca del avance del proceso de cada requisición y sus candidatos, actividades realizadas, planeadas y pendientes por terminar, de esta manera se agiliza considerablemente el flujo de información entre los consultores, el área de reclutamiento y selección, recursos humanos y el cliente final. Igualmente se puede enviar de manera automática desde el Hum&Select por medio de *e-mail* los reportes psicométricos y los expedientes de los candidatos, de esta manera se elimina el manejo de papelería impresa (ejemplo de: expediente, reporte psicométrico).
- **Ambiente multi-usuario y seguridad de acceso:** está diseñado para trabajar en su red local de manera multiusuario y permite configurar de una manera versátil las áreas del sistema e información que los diferentes tipos de usuarios pueden acceder.

Además incluye la bolsa de trabajo electrónica, la automatización de pruebas psicométricas y permite llevar un control total de las requisiciones de personal dando un seguimiento detallado al proceso de cada uno de los candidatos. (15)

- **Sistema Automatizado de RR.HH. de CIMEX**

En Cuba también se han desarrollado varios sistemas con el objetivo de llevar en sus empresas una buena gestión de los recursos humanos.

El Sistema Automatizado de RR.HH. de CIMEX es una aplicación web que se encuentra en su versión 2.2.4 y brinda servicios a todas las unidades de la empresa corporativa CIMEX. Este sistema está basado en la intranet de la misma, a él tienen acceso sólo los responsables de RR.HH. en cada unidad, a diferencia de otros sistemas a los que pueden acceder todos los usuarios de la empresa. Está estructurado de forma tal que a nivel central se puede chequear el trabajo en las unidades.

Dentro de los módulos que ofrece están:

- Diseño organizacional
- Reclutamiento y selección
- Gestión integral del personal
- Gestión integral de remuneraciones e incentivos
- Contabilidad
- Seguridad del trabajo
- Administración
- Reportes

Tiene como característica que es un sistema cuya interfaz es solamente visible dentro de esta empresa, y su solución es bastante puntual, adaptada a las necesidades de la misma, además de que contiene otros módulos que no son de ayuda para la implementación del SGIRHT. (16)

- **Sistema de Control de los recursos humanos y didácticos en los Centros de Auto-aprendizaje y Servicios de Idiomas Extranjeros (CASIE)**

Un CASIE es un centro de auto-aprendizaje y servicios de idiomas extranjeros creado en la UCI para promover la autonomía en el proceso de aprendizaje de los estudiantes del centro y facilitar materiales que apoyaran la formación académica en lenguas extranjeras, con el objetivo primario de desarrollar la comprensión de textos en inglés, considerada una habilidad rectora en la carrera de Ingeniería Informática.

Como parte de los servicios que se ofrecen, se aplica un registro de información sobre las actividades desarrolladas a diario en los mismos, el cual comprende:

- Registro de asistencia: proceso donde se guarda la información referente a los usuarios que visitan el CASIE.

- Reporte de asistencia: proceso donde se realiza el resumen general de la asistencia semanal al CASIE.
- Reporte de estado técnico: en este proceso se realiza una revisión del equipamiento técnico del CASIE y se genera un reporte plasmando el estado del mismo.
- Reservación de salón: en este proceso es donde se realiza la reservación de los distintos salones del CASIE.
- Planificación del horario de asesoría: proceso en el cual se lleva a cabo la planificación del horario de asesoría.
- Préstamo de recursos didácticos: en este proceso se registran los préstamos realizados de los recursos didácticos disponibles en los CASIES.
- Gestión de recursos didácticos: proceso en el cual se dan de alta y baja a los recursos didácticos en el CASIE. (17)

- **Sistema de Gestión de Información de los Recursos Humanos (SGIRH) de la Facultad 2**

Dicha aplicación web permite gestionar la información de estudiantes, profesores y proyectos productivos en la facultad 2 de la UCI, con el fin de lograr una mejor organización, que permita proveer a los clientes de información acerca de todo el quehacer productivo, docente e investigativo de estudiantes y profesores, sirviendo así como herramienta de ayuda en la toma de decisiones, la que se pone en práctica de manera intuitiva, teniendo en cuenta criterios subjetivos, lo que influye negativamente en la calidad y efectividad de la misma. La aplicación ofrece una amplia información referente a los estudiantes, profesores y proyectos productivos, garantizando la creación de un perfil más completo de estudiantes y profesores, propiciando la posibilidad de creación de una ficha técnica con mayor acabado de los proyectos productivos vinculados a la facultad, gestionando los cambios en cada perfil o ficha técnica si es producida alguna modificación o baja, además de permitir obtener reportes filtrados por los criterios que sean escogidos por el usuario.

El sistema cuenta con una interfaz web fácil para su uso permitiendo así que un elevado número de usuarios puedan disfrutar de sus prestaciones. Con la implementación de este sistema se le brinda la posibilidad a los directivos de la facultad el poder contar con una útil y rápida herramienta que ayuda en la toma de decisiones, mejorando así su trabajo al lograr centralizar la información que necesitan conocer de sus profesores, estudiantes y proyectos, gestionando la misma a través de la aplicación web. (18)

Justificación del estado del arte

Luego de un análisis exhaustivo de las principales características de los sistemas expuestos anteriormente, se ha llegado a la conclusión de la necesidad de implementar un sistema de gestión de recursos humanos y tecnológicos para el CISED, debido a que todos estos productos que existen

actualmente, fueron diseñados específicamente para ser utilizados por las entidades para las cuales fueron creadas, unos por ser privativos y otros porque solo son capaces de gestionar la información relacionada con el personal que manejan específicamente. Otro de los problemas identificados por los cuales se arribó a esta conclusión, fue la incapacidad de los productos existentes de gestionar no solo la información referente al personal sino también su ubicación dentro de la entidad correspondiente.

El sistema propuesto tendrá que ser capaz de gestionar la información de todos los estudiantes, trabajadores, profesores y equipos técnicos así como la ubicación de los mismos en los las diferentes área (departamentos, laboratorios, oficinas, proyectos) que integran el CISED. Se persigue además como parte de las funcionalidades del sistema, que en próximas iteraciones exista la posibilidad de asociación entre los principales conceptos del dominio y su respectiva ubicación espacial dentro del centro, pudiendo entonces agrupar la información por el área física en que están enmarcados.

1.3. Metodologías de desarrollo

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software. Mientras se desarrolla se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en la concepción de éstas, y qué papel deben jugar. Se detalla la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Las metodologías de desarrollo de software se pueden clasificar en dos grandes grupos:

- las metodologías orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán. Estas metodologías son llamadas metodologías pesadas o robustas.
- las metodologías orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del mismo al cliente, en cortos intervalos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se va desarrollando. Estas son llamadas metodologías ligeras o ágiles.

A continuación se muestra un análisis de dos metodologías punteras en el desarrollo de software actualmente:

1.3.1. Metodologías pesadas o robustas.

Estas metodologías, son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever

todo de antemano. Son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida. (19)

1.3.1.1. Metodología RUP (Rational Unified Process)

Una de las metodologías pesadas más conocidas y utilizadas es la metodología RUP cuyas principales características van dirigidas a lograr un software guiado por casos de uso, centrado en la arquitectura e iterativo e incremental. Divide el proceso de desarrollo en cuatro fases que definen su ciclo de vida:

- Inicio: El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- Elaboración: Etapa en la que se determina la arquitectura óptima del proyecto.
- Construcción: Se obtiene la capacidad operacional inicial.
- Transición: Obtener el producto acabado y definido.

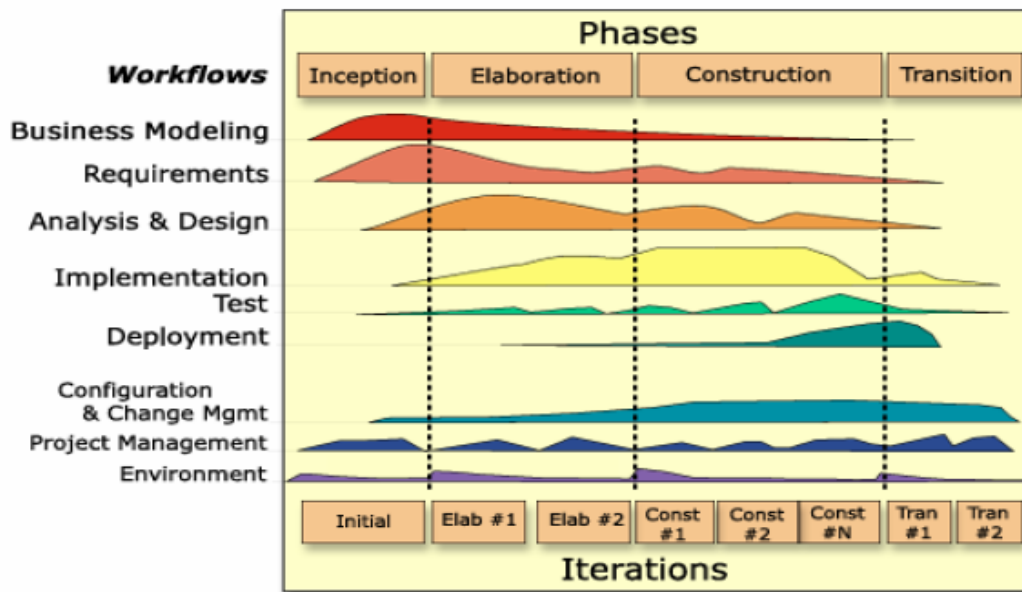


Figura 1 Fases e Iteraciones de la Metodología RUP

Esta metodología tiene como principios claves:

- Adaptación del proceso: el proceso debe adaptarse a las características de la organización para la que se está desarrollando el software.
- Balancear prioridades: debe encontrarse un balance que satisfaga a todos los inversores del proyecto.
- Colaboración entre equipos: debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes y resultados.

- Demostrar valor iterativamente: los proyectos se entregan, aunque sea de una forma interna, en etapas iteradas. En cada iteración se evaluará la calidad y estabilidad del producto, y se analizará la opinión y sugerencias de los inversores.
- Elevar el nivel de abstracción: motivar el uso de conceptos reutilizables.
- Enfocarse en la calidad: la calidad del producto debe verificarse en cada aspecto de la producción.

Ventajas de RUP:

- Evaluación en cada fase que permite cambios de objetivos.
- Funciona bien en proyectos de innovación.
- Es sencillo, ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.
- Seguimiento detallado en cada una de las fases. (20)

Desventajas de RUP

- La evaluación de riesgos es compleja.
- Excesiva flexibilidad para algunos proyectos.
- Se pone al cliente en una situación que puede ser muy incómoda para él, deberá ser capaz de describir y entender a un gran nivel de detalle el funcionamiento del proyecto para poder acordar correctamente el alcance del mismo. (21)

1.3.2. Metodologías ágiles

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término ágil aplicado al desarrollo de software. En ésta, participaron un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de las metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo de su creación. Tras esta reunión se creó *The Agile Alliance*, una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones a que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”. Este manifiesto contiene una serie de características como son: las personas constituyen el principal factor de éxito para un proyecto de software y la documentación del mismo debe ser corta y centrarse en lo fundamental. Además debe existir una interacción constante entre el cliente y el equipo de desarrollo y la planificación no debe ser estricta, sino flexible y abierta. (22)

1.3.2.1. Metodología Programación Extrema (XP):

XP constituye una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje

de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico.

El ciclo de vida de XP consta de 6 fases:

- Fase I (exploración): el equipo se familiariza con el proyecto y las tecnologías necesarias. Al mismo tiempo, el cliente desarrolla la mayor cantidad posible de historias de usuarios. La duración de esta fase depende del nivel de conocimientos del equipo sobre las tecnologías a utilizar.
- Fase II (planificación de entrega): el cliente prioriza las historias de usuarios y el equipo hace una estimación de duración para las mismas. Cada entrega debe realizarse en unos pocos meses.
- Fase III (iteraciones): cada iteración incluye historias de usuarios que no hayan sido incluidas en la iteración anterior, corrige defectos encontrados y asume tareas anteriormente sin terminar.
- Fase IV (producción): al sistema se le realiza una serie de pruebas rigurosas para ser trasladado al entorno del cliente.
- Fase V (mantenimiento): cada última versión se mantiene funcionando mientras el equipo desarrolla la siguiente. Esto requiere realizar tareas de soporte al cliente.
- Fase VI (muerte del proyecto): cuando el cliente no tiene más historias de usuarios que agregar y se comienza a refinar el rendimiento y otros aspectos de interés. Otra causa puede ser la cancelación del proyecto.

Algunas prácticas de desarrollo de XP son:

- Entregas pequeñas: se desarrollan versiones incompletas del producto, pero con funcionalidades totalmente operativas que añaden valor al mismo.
- Diseño simple: reduce al mínimo posible el código y la cantidad de funciones. Además trata de no duplicar lógica de negocio alguna.
- Refactorización: consiste en la revisión constante de código para reestructurarlo y hacerlo más entendible, eliminar duplicación de algoritmos y funciones.
- Programación en parejas: las principales ventajas de esta práctica son el poco tamaño del código, menor tasa de errores, solución rápida de los problemas y mejor diseño.
- Propiedad colectiva del código: cada programador puede cambiar cualquier fragmento de código en cualquier momento. Esto estimula la creatividad del equipo. (23)

1.3.3. Selección de la metodología

Una vez realizado el estudio sobre las posibles metodologías de desarrollo de software se escogió por parte del equipo de trabajo la metodología XP, debido a que el desarrollo del producto final debe

realizarse en un breve período de tiempo, el equipo de desarrollo está compuesto por cinco integrantes, entre los que se encuentra el cliente, y se centra casi en su totalidad en las tareas de desarrollo, preocupándose por hacer mejoras continuas al producto.

1.4. Selección de las herramientas

Un estudio detallado de las diferentes herramientas y tecnologías a utilizar, arrojó que la mayoría de las aplicaciones de gestión para la informatización de los procesos UCI, son desarrolladas en marcos de trabajo que usan como base el lenguaje php y el gestor de base de datos postgresql, con el interés de que sea consumida desde cualquier plataforma y sin necesidad de licenciamiento.

1.4.1. Marco de trabajo: django en su versión 1.4.2

Django es un marco de trabajo escrito en python, multiplataforma, de código abierto y que cumple en cierta medida el paradigma del controlador. Permite que los desarrolladores se dediquen a construir los objetos *Entity*, la lógica de presentación y control para ellos, además pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio no te repitas (DRY) el principio establece que, en un entorno informático, la información no debe repetirse, es decir, el conocimiento almacenado en un programa informático debe mantenerse en un sólo lugar.

Sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar estrictamente ningún paradigma particular, y en cambio prefieren hacer "lo que les parece correcto". Como resultado, por ejemplo, lo que se llamaría "controlador" en un "verdadero" entorno Modelo-Vista-Controlador (MVC) se llama en django "vista", y lo que se llamaría "vista" se llama "plantilla" y lo que es aún más importante proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. (24)

Todas estas particularidades son consideradas suficientes por los autores del presente trabajo para satisfacer las necesidades de la etapa de desarrollo, además fue escogido, por las necesidades de expansión de dicha solución mencionadas en secciones anteriores.

1.4.2. Lenguaje de programación: Python en su versión 2.7.3

Python es el lenguaje básico utilizado en django, no obstante, no se puede dejar de mencionar que es un software libre que posee una licencia de código abierto denominada *python software foundation license*, que es multiplataforma y multiparadigma, y que además la cantidad de librerías que contiene y los tipos de datos y funciones incorporadas en el propio lenguaje, ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero. (25)

1.4.3. Entorno Integrado de Desarrollo: PyCharm en su versión 2.0.1

PyCharm es un Entorno Integrado de Desarrollo (IDE de sus siglas en inglés) diseñado para trabajar con el lenguaje de programación python y las plantillas de django, marco de desarrollo web escrito en ese mismo lenguaje. Tiene comodidades como, formateo automático del código y completamiento del mismo para palabras reservadas, variables y clases. Posee también potencialidades como la navegación y la búsqueda a través del código y la refactorización del mismo en caso de cambios en cascada. Cuenta con intérprete para otros lenguajes que pudieran ser usados en el mismo proyecto como HTML, CSS y Javascript. (26)

1.4.4. XHTML y CSS

Django en la capa de presentación utiliza XHTML y CSS para lograr una visualización de los elementos y contenidos organizada y agradable a la vista del cliente.

- **XHTML**

HTML (Extensible Hypertext Markup Language o Lenguaje extensible de marcación de hipertexto), ofrece la norma clásica para crear páginas web. Surge a partir de la necesidad de un sistema estándar para añadir nuevas características a las páginas. Desde fuera, XHTML es muy similar a XML, sus etiquetas, atributos y valores; todas sus características le permiten que lo puedan entender todos los navegadores. Está diseñado para guiar el desarrollo de documentos bien formados y válidos.

- **CSS**

CSS (Cascade Style Sheet) especifica la forma del diseño de los documentos (tanto XHTML como HTML). Una misma página Web (un mismo documento XHTML, por ejemplo) puede ser vista de diferente forma en una estación de trabajo, gracias a diferentes hojas de estilo CSS.

Con CSS se logra aumentar la densidad de las palabras clave dentro de los contenidos, ya que muchas de las etiquetas ocuparán menos espacio. Esto también supone un menor peso para las páginas Web. CSS se puede lograr encerrar cada información dentro de la etiqueta que corresponda al significado semántico adecuado. El lenguaje de las Hojas de estilo, aunque es muy potente, es relativamente sencillo y fácil de aprender. (27)

1.4.5. Gestor de base de datos: PostgreSQL en su versión 9.1.7.1

El marco de trabajo escogido propone el uso de **PostgreSQL** como gestor de base de datos. Éste es un software libre, por lo que cualquiera puede disponer de su código fuente, modificarlo a voluntad y redistribuirlo libremente. Además, es gratuito y se puede descargar libremente de su página web para múltiples plataformas. Soporta operadores funcionales, métodos de acceso y tipos de datos definidos por

el usuario. Es capaz de manejar los registros, sin necesidad de que los usuarios tengan que esperar a que estén disponibles. Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. Es un magnífico gestor de bases de datos, capaz de competir con muchos gestores comerciales. (28) En la UCI existe un centro de desarrollo en el que uno de sus departamentos tiene como objeto social el desarrollo de funcionalidades de dicho gestor y además existe experiencia y documentación asociada a la instalación y configuración del mismo.

1.4.6. Servidor web: Apache en su versión 2.2.20

Django incluye un servidor web liviano para realizar pruebas y trabajar en la etapa de desarrollo, en la período de despliegue se recomienda **Apache** con *mod_python*. El mismo es un servidor altamente configurable de diseño modular que por su sencillez permite ampliar sus capacidades, de manera que cualquiera que posea experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

Es una tecnología gratuita con licencia de código abierto de tipo BSD (*Berkeley Software Distribution*). Funciona en Windows, en Linux y en otros sistemas de Unix. Presenta muchas otras características, entre ellas un elaborado índice de directorios, un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, gestión de recursos para procesos hijos, integración de imágenes del lado del servidor, reescritura de las URL, comprobación de la ortografía de las URL y manuales online. (29)

1.4.7. Herramienta de modelado: Visual Paradigm en su versión 6.4

Como herramienta para el modelado se propone el uso de **Visual Paradigm** en su versión 6.4 puesto que es muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue incluyendo aquellos estereotipos de la metodología seleccionada.

Esta herramienta tiene como ventaja que es de software libre y además incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros. (28)

Conclusiones

En este capítulo se elaboró un resumen de los principales conceptos a tratar durante el desarrollo de la aplicación y que conforman el objeto de estudio. Se realizó un análisis profundo de los diferentes sistemas con características similares que existen en Cuba y en el resto del mundo, demostrando la necesidad de la implementación de un software para la gestión de la información de los recursos humanos y

tecnológicos que le permita a los directivos del CISED optimizar su trabajo en cuanto a calidad y rapidez se refiere. Fueron definidas la metodología, tecnologías y herramientas que regirán todo el ciclo de desarrollo del software.

Capítulo 2: Características del sistema

Introducción

En este capítulo se describe la propuesta de solución teniendo en cuenta las dificultades que existen actualmente en la entidad, se muestran de manera detallada los procesos relacionados con el negocio mediante el modelo de dominio. Además se caracteriza el sistema a partir de la descripción del problema, se especifican los requerimientos funcionales y no funcionales del software y son definidos los patrones de diseño y la arquitectura a utilizar, así como los artefactos correspondientes a la etapa de planeación y diseño del software, propuestos por la metodología a utilizar.

2.1. Objeto de automatización

Para que se realice una correcta gestión de la información de los recursos humanos y tecnológicos en el CISED, es precisa la automatización de los siguientes procesos, los cuales ayudarán a solucionar los problemas que fueron identificados previamente.

- Gestionar trabajadores.
- Gestionar estudiante.
- Gestionar profesor.
- Gestionar máquina.
- Gestionar usuario.
- Autenticar usuario.
- Mostrar reportes:
 - Datos de las personas.
 - Listado de estudiantes.
 - Listado de trabajadores.
 - Dado el nombre del profesor decir su al departamento que pertenece.
 - Características de la pc.
 - Año de graduado de un profesor.
 - Proyecto del estudiante.
 - Cantidad de estudiantes por departamento.
 - Máquina en la que trabaja el profesor.
 - Máquina en la que trabaja el estudiante.
 - Categoría científica del profesor.
 - Cargo de las personas.
 - Categoría docente del profesor.
 - Proyecto del profesor.
 - Cantidad de estudiantes por laboratorio.
 - Tesis tutorada por un profesor.
 - Trabajador y su categoría.
 - Asignatura dado el profesor.
 - Profesores asignaturas y horas clases.
 - Personas y sus proyectos.
 - Profesores y su departamento.
 - Profesores dado una categoría científica.
 - Profesores dado una categoría docente.

- Profesores y cantidad de estudiantes que tutora.
- Trabajadores y sus eventos, postgrados y publicaciones.
- Trabajadores y los postgrados que cursa.
- Profesores y su rol.

2.2. Información que se maneja

La información que se maneja está dada en mayor medida por los datos de cada persona que radica en el centro, así como su ubicación dentro del mismo, además de las características relacionadas con las tecnologías que le están asignadas.

2.3. Propuesta del sistema

Se propone el desarrollo de un sistema web para el CISED que permita la obtención de reportes relacionados con la información del personal y las tecnologías existentes en el mismo. El proceso comenzará una vez que el usuario se autentique y según los permisos que tenga asignado logrará consultar la información que desee. El desarrollo de este software le tributará una serie de beneficios a los directivos del centro, permitiéndoles contar con una herramienta capaz de optimizar su trabajo.

2.3.1. Modelo de dominio

Un Modelo de Dominio es un artefacto, construido con las reglas del Lenguaje Unificado de Modelado (UML) durante la fase de concepción, compuesto por uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física. Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir. (30)

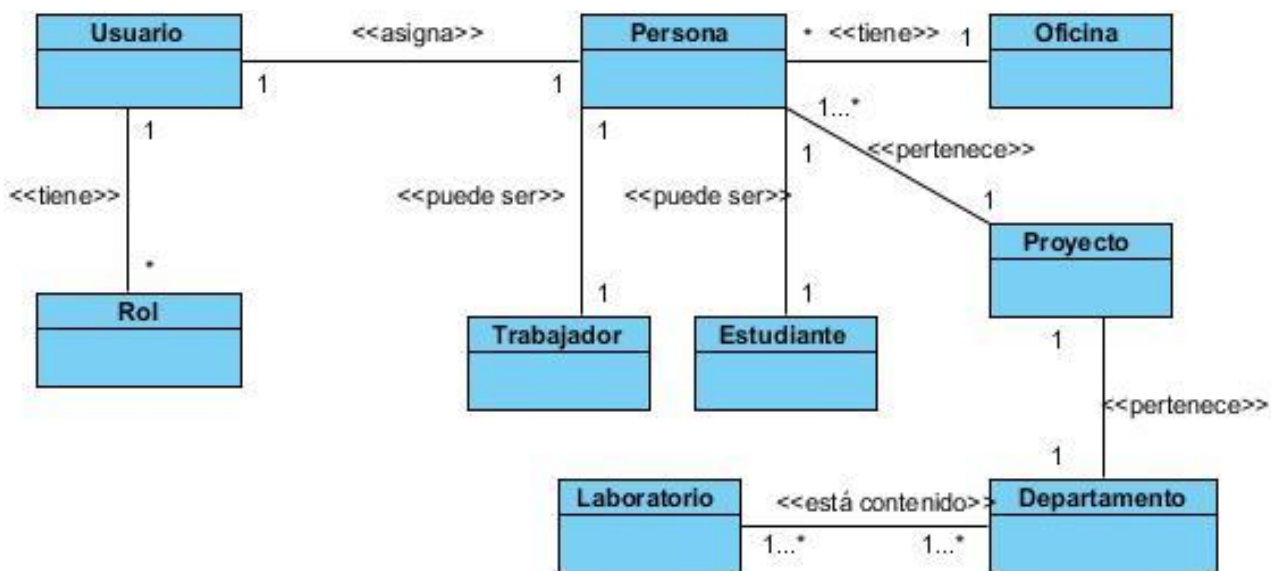


Figura 2 Diagrama del modelo de dominio

Definición de las entidades y conceptos fundamentales:

- **Persona:** concepto abstracto que representa a los estudiantes, trabajadores y asesores del CISED.
- **Usuario:** persona que tiene acceso al sistema con diferentes permisos que son obtenidos una vez que se registra en la solución mediante un identificador y una clave de acceso.
- **Rol:** diferentes permisos que se le otorgan a los usuarios del sistema con los cuales podrán tener acceso a las diferentes dimensiones del producto informático.
- **Trabajador:** persona que realiza algún tipo de trabajo o actividad a cambio de un salario en el CISED.
- **Estudiante:** son todas las personas que poseen la categoría de estudiante en el CISED.
- **Oficina:** aquellos locales del CISED en los que laboran directivos y asesores.
- **Laboratorio:** área destinada a la actividad docente productiva investigativa de los estudiantes, trabajadores y asesores del CISED.
- **Proyecto:** concepto que agrupa profesores, estudiantes y especialistas con el objetivo de lograr determinado conocimiento, producto o servicio.
- **Departamento:** constituye cada área administrativa en las cuales se divide el CISED.

2.4. Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Asesores	Son los encargados de controlar toda la información que se maneja.
Jefe de departamento	Es el encargado de controlar la información de su departamento y modificarla en determinado momento.
Subdirector del centro	Facultado para realizar todas las operaciones del sistema excepto eliminar una persona.
Jefe del centro	Facultado para realizar cualquier operación dentro del sistema.
Subdirector de tecnología	Persona con privilegios para la modificación de la información basada en las tecnologías.
Subdirector docente	Persona con privilegios para la modificación de la información basada en el personal.

Tabla 1 Personas relacionadas con el sistema.

2.5. Exploración

En esta fase, los clientes plantean a grandes rasgos los procesos que desean automatizar a través de las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán durante el desarrollo del producto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración consta de pocas semanas a pocos meses, dependiendo del conocimiento y familiaridad que tengan los programadores con la tecnología. (23)

2.5.1. Lista de reserva del producto (LRP)

Cuando un proyecto comienza es difícil tener claro todos los requisitos, el equipo de desarrollo se enfoca en las funcionalidades que por el momento suelen ser más importantes para el sistema. En el transcurso del tiempo esta lista puede crecer y modificarse debido a las constantes reuniones e intercambios entre el cliente y los desarrolladores donde adquieren mayor información sobre el producto. Al terminar de llenar la

lista el objetivo es obtener un producto competitivo, lo más correcto posible y que satisfaga las necesidades del cliente.

El uso de esta lista es de gran importancia pues pone en marcha una de las principales características de la metodología XP: ser recomendada para proyectos con requisitos imprecisos y muy variantes. Los programadores que utilizan esta metodología consideran que los cambios frecuentes de requisitos durante el desarrollo del software son un aspecto natural, inevitable y que poder adaptarse a éstos en cualquier parte del ciclo de vida del proyecto es una aproximación mejor y más realista. (31)

2.5.2. Funcionalidades del sistema

2.5.2.1. Requisitos funcionales del sistema

Requisitos funcionales	Prioridad
Gestionar trabajadores. <ul style="list-style-type: none"> • Insertar trabajador. • Modificar trabajador. • Eliminar trabajador. • Consultar datos de un trabajador. 	Alta
Gestionar estudiante. <ul style="list-style-type: none"> • Insertar estudiante. • Modificar estudiante. • Eliminar estudiante. • Consultar datos de un estudiante. 	Alta
Gestionar profesor. <ul style="list-style-type: none"> • Insertar profesor. • Modificar profesor. • Eliminar profesor. • Consultar datos de un profesor. 	Alta
Gestionar máquina. <ul style="list-style-type: none"> • Insertar pc. • Modificar pc. • Eliminar pc. • Consultar datos de una pc. 	Alta
Gestionar usuario. <ul style="list-style-type: none"> • Insertar usuario. 	Alta

<ul style="list-style-type: none"> • Modificar usuario. • Eliminar usuario. • Consultar datos de un usuario. 	
<p>Mostrar reportes.</p> <ul style="list-style-type: none"> • Datos de las personas. • Listado de estudiantes. • Listado de trabajadores. • Dado el nombre del profesor decir su al departamento que pertenece. • Características de la pc. • Año de graduado de un profesor. • Proyecto del estudiante. • Cantidad de estudiantes por departamento. • Máquina en la que trabaja el profesor. • Máquina en la que trabaja el estudiante. • Categoría científica del profesor. • Cargo de las personas. • Categoría docente del profesor. • Proyecto del profesor. • Cantidad de estudiantes por laboratorio. • Tesis tutorada por un profesor. • Trabajador y su categoría. • Asignatura dado el profesor. • Profesores asignaturas y horas clases. • Personas y sus proyectos. • Profesores y su departamento. • Profesores dado una categoría científica. • Profesores dado una categoría docente. 	<p>Muy alta</p>

<ul style="list-style-type: none"> • Profesores y cantidad de estudiantes que tutora. • Trabajadores y sus eventos, postgrados y publicaciones. • Trabajadores y los postgrados que cursa. • Profesores y su rol. 	
Autenticar usuario	Alta

Tabla 2 Funcionalidades del Sistema.

2.5.2.2. Requisitos no funcionales del sistema

- **Requisitos de Software**

PC cliente: debe contar al menos con las siguientes configuraciones:

- Sistema Operativo con interfaz gráfica.
- Un navegador web.

Servidor BD: Servidor de PostgreSQL 9.x

Servidor web: Apache en su versión 2.x habiéndole habilitado los módulos *python*, *postgres* y *ldap*.

- **Requisitos de Hardware**

PC cliente con 256 Mb o más de memoria RAM, microprocesador a 2.0 Ghz o superior y red a 100 Mbps.

Servidor: capacidad de almacenamiento de 40 GB o superior, memoria RAM de 1GB o superior, microprocesador a 2.2 Ghz y red a 100 Mbps.

- **Restricciones en el diseño y la implementación**

- Implementación del sistema con el lenguaje de programación Python 2.7.3.
- Gestor de base de datos PostgreSQL 9.1.7.1.
- IDE de desarrollo PyCharm 2.0.1.
- Visual Parading 6.4 como herramienta case para el desarrollo.
- Apache 2.0 como servidor web.

- **Requisitos de apariencia de interfaz externa**

Interfaz externa amigable y de navegación sencilla para el usuario.

- **Requisitos de seguridad**

Podrán acceder a la aplicación las personas autorizadas para el manejo de la información almacenada.

- **Requisitos de usabilidad**

El sistema puede ser usado por personas que tengan conocimientos básicos sobre el manejo de un ordenador.

2.5.3. Historias de usuario

En esta etapa los principales artefactos que se obtienen son las historias de usuario, XP las utiliza como una técnica para reemplazar el documento de requisitos. Siguen los principios básicos de requerimientos ágiles, y potencian la participación del equipo en la toma de decisiones. Se crean y evolucionan a medida que el proyecto avanza. Son peticiones concretas y pequeñas. Contiene la información imprescindible. Apoyan la cooperación, colaboración y conversación entre los miembros del equipo, lo que es fundamental. Se realiza una por cada característica principal del sistema. Difieren de los casos de uso porque son escritas por los clientes, no por los programadores. Las historias de usuario son la base de las pruebas de aceptación pues mediante estas se verifica si han sido implementadas correctamente y si cumplen las expectativas del cliente. Existen diferencias entre las historias de usuario y la habitual especificación de requisitos, divergiendo principalmente en el nivel de detalle. (31)

A continuación serán expuestas las principales historias de usuarios, para consultar el resto (**ver Anexo 2**):

Número: HU_1	Usuario: Hiraldo Manuel Anaya González		
Nombre historia: Autenticar usuario.			
Prioridad en negocio: Alta		Riesgo en desarrollo: Alto	
Puntos estimados: 2		Iteración asignada: 1	
Programador responsable: Rocio Romero Alvarez			
Descripción: La presente historia de usuario tiene como objetivo autenticar a una persona para así darle acceso a la aplicación y definir las actividades según el rol.			
Observaciones: En caso de que no exista el usuario se emitirá un aviso de usuario no válido.			

Requisitos no funcionales	Interfaz
Asuntos pendientes	Ninguno

Tabla 3 Historia de usuario 1. Autenticar usuario.

Número: HU_2	Usuario: Hiraldo Manuel Anaya González	
Nombre historia: Gestionar usuario.		
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo	
Puntos estimados: 3	Iteración asignada: 1	
Programador responsable: Yam Brayam Beltrán Rodríguez		
Descripción: La presente historia de usuario tiene como objetivo gestionar un usuario, al cual se le podrá hacer una serie de modificaciones.		
Observaciones: En caso de que el usuario no pertenezca al sistema, se emitirá un aviso de que el usuario no existe, además permite realizar varias funcionalidades como: <ul style="list-style-type: none"> • Adicionar: Define la forma en que se adicionará un usuario. • Eliminar: Define la forma en que se eliminará un usuario. • Modificar: Define la forma en que se modificará un usuario. • Consultar: Define la forma en que se consultará la información de un usuario en el sistema. 		
Requisitos no funcionales	Interfaz	
Asuntos pendientes	Ninguno	

Tabla 4 Historia de usuario 2. Administrar usuario.

Número: HU_3	Usuario: Hiraldo Manuel Anaya González	
Nombre historia: Gestionar trabajadores.		
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio	
Puntos estimados: 2	Iteración asignada: 1	
Programador responsable: Rocio Romero Alvarez		

Descripción: La presente historia de usuario tiene como objetivo gestionar un trabajador al cual se le podrá hacer una serie de modificaciones.	
Observaciones: Esta operación solo podrá ser realizada por la persona autorizada en el sistema para ello, la cual tendrá permisos para realizar los siguientes procedimientos: <ul style="list-style-type: none"> • Adicionar: Permitirá adicionar trabajadores al sistema. • Eliminar: Permitirá eliminar trabajadores del sistema. • Modificar: Permitirá modificar trabajadores en el sistema. • Consultar: Permitirá buscar un trabajador en el sistema y consultar su información. 	
Requisitos no funcionales	Interfaz
Asuntos pendientes	Ninguno

Tabla 5 Historia de usuario 3. Gestionar trabajadores.

2.5.4. Arquitectura del sistema

El **modelo vista controlador (MVC)** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (modelo, vista y controlador). El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Ventajas:

- Clara separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (TDD).
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables. (32)

Django sigue el patrón **MVC** tan al pie de la letra que puede ser llamado como un marco de trabajo MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

- M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.

- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el propio marco de trabajo siguiendo la URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la C es manejada por el mismo marco de trabajo y la parte más emocionante se produce en los modelos, las plantillas y las vistas, Django es conocido como un marco de trabajo MTV. En el patrón de diseño MTV:

- M significa “Model” (Modelo), la capa de acceso a la base de datos: contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tienen, y las relaciones entre ellos.
- T significa “Template” (Plantilla), la capa de presentación: contiene las decisiones relacionadas a la presentación: algunas cosas son mostradas sobre una página web u otro tipo de documento.
- V significa “View” (Vista), la capa de la lógica de negocios: contiene la lógica que accede al modelo y la delega a la plantilla apropiada: esto se puede concebir como un puente entre el modelo y las plantillas.

En la interpretación de Django de MVC, la vista describe los datos que son presentados al usuario; no necesariamente el cómo se mostrarán, pero si cuáles son presentados. (24)

2.5.5. Patrones del diseño

Los patrones de diseño son tácticas de media escala relacionados con la estructura y el comportamiento de entidades y sus relaciones. No influyen sobre toda la estructura del sistema, pero definen micro arquitecturas de subsistemas y componentes. Estos patrones solucionan problemas específicos del diseño y lo hacen orientado a objetos más flexibles, elegantes y por último reutilizables. Los mismos constituyen descripciones de las comunicaciones de objetos y clases que son personalizadas para resolver un problema general de diseño en un contexto particular, identifican las clases participantes y las instancias, sus papeles y colaboraciones, y la distribución de responsabilidades, pueden ser utilizados en cualquier lenguaje de programación orientado a objetos, adaptando los diseños generales a las características de la implementación particular. En general, un patrón tiene cuatro elementos esenciales:

- 1) El nombre del patrón se utiliza para describir un problema de diseño, su solución, y consecuencias en una o dos palabras. Nombrar un patrón incrementa inmediatamente el vocabulario de diseño. Esto permite diseños a un alto nivel de abstracción.
- 2) El problema describe cuando aplicar el patrón. Se explica el problema y su contexto. Esto podría describir problemas de diseño específicos tales como algoritmos como objetos. Podría describir

estructuras de clases u objetos que son sintomáticas de un diseño inflexible. Algunas veces el problema incluirá una lista de condiciones que deben cumplirse para poder aplicar el patrón.

- 3) La solución describe los elementos que forman el diseño, sus relaciones, responsabilidades y colaboraciones. La solución no describe un diseño particular o implementación, porque un patrón es como una plantilla que puede ser aplicada en diferentes situaciones. En cambio, los patrones proveen una descripción abstracta de un problema de diseño y como una disposición general de los elementos (clases y objetos en nuestro caso) lo soluciona.
- 4) Las consecuencias son los resultados de aplicar el patrón. Estas son muy importantes para la evaluación de diseños alternativos y para comprender los costes y beneficios de la aplicación del patrón. (33)

En django ocurren comúnmente una serie de patrones que hacen que la colaboración sea más fácil, más simple de codificar y/o más mantenible. Ninguno de ellos son patrones de diseño, en el sentido de patrones de diseño GoF, es decir, utiliza los mismos como le parece más cercano o conveniente en la solución del problema. Este marco de trabajo usa patrones a la hora de trabajar con las url, los modelos, los formularios, las vistas y las plantillas.

Para el trabajo con las url permite la definición de un fichero *url.py* donde estarán las url de la aplicación y el mismo puede estar repetido tanto a nivel de proyecto, como por cada una de las aplicaciones existentes. Unido a esto posibilita que cada url pueda ser nombrada con un término entendible. Ejemplo

```
urlpatterns = patterns('',

    # Uncomment the admin/doc line below to enable admin documentation:
    url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

),

url(r'^accounts/login/$', 'django.contrib.auth.views.login', {'template_name': 'Autenticacion.html',
name='login'}),
url(r'^accounts/logout/$', 'django.contrib.auth.views.logout_then_login', name='logout'),
# Uncomment the next line to enable the admin:
(r'^admin/', include(admin.site.urls)),

url(r'^Index/', 'GestionRecursosHumanosTecnologias.Gestion.views.Index'),
url(r'^Autenticacion/', 'GestionRecursosHumanosTecnologias.Gestion.views.Autenticacion'),
url(r'^Base_Reporte', 'GestionRecursosHumanosTecnologias.Gestion.views.Base_Reporte'),
```

Figura 3 Patrón de diseño en las url.

Existen diferentes formas de definir una aplicación sobre django en dependencia del modelo que se utilice, como por ejemplo: el uso de múltiples clases controladoras con la posibilidad de adicionarle nuevos métodos a la controladora por defecto, facilita realizar herencia entre relaciones.

Los modelos permiten además usar la clase *logging* permitiendo dejar una traza de cuando un objeto es creado, modificado o eliminado, también se cuenta en una solución de este tipo con el uso de buenas prácticas como el `__unicode__` posibilitando que aquellos elementos del modelo en que sus partes necesiten ser mostrados de forma agregada no sea necesario redefinirlos.

```
try:
    from logging.config import dictConfig
except ImportError:
    from django.utils.dictconfig import dictConfig

getLogger = logging.getLogger

logger = getLogger('django')
if not logger.handlers:
    logger.addHandler(NullHandler())
```

Figura 4 Patrón de diseño utilizando la clase *logging*.

```
class Estudiante(models.Model):
    personasolapin = models.ForeignKey(Persona, primary_key=True, db_column='personasolapin', verbose_name='persona')
    anno_acad = models.IntegerField(verbose_name='año académico')
    def __unicode__(self):
        return u'%s %s' % (self.personasolapin.nombre, self.personasolapin.apellidos)
    class Meta:
        db_table = u'estudiante'
```

Figura 5 Patrón de diseño utilizando el `__unicode__`.

Django brinda la posibilidad de utilizar vistas genéricas para aquellos casos en que pudieran repetirse el mismo tipo de información en varios escenarios, en este caso se implementa la vista con la llamada a la función: *render_to_response*.

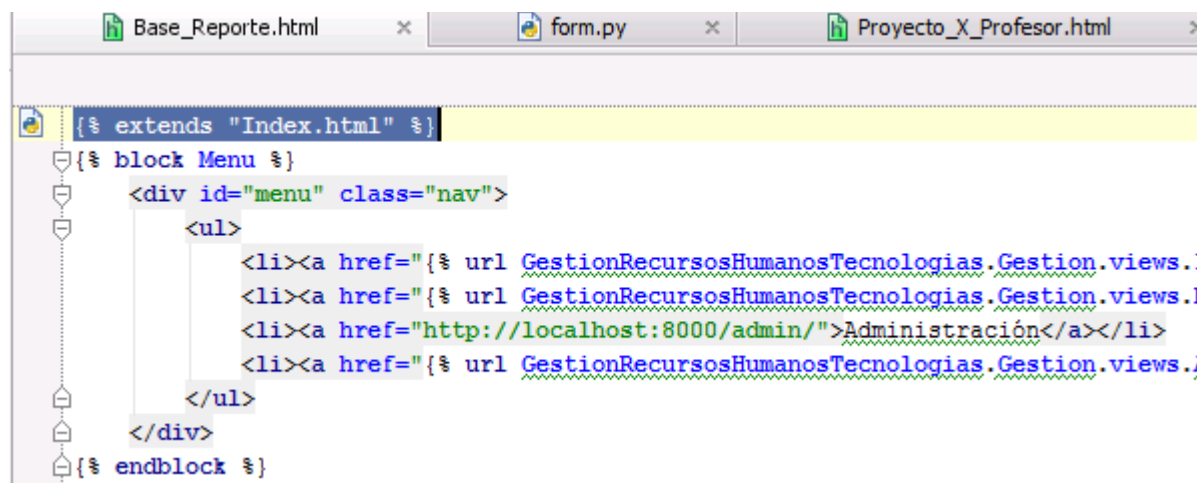
```
def Tesis_turoreada_por_el_profesor(request):
    errores = []
    if request.method == 'POST':
        form = Filtro_Profesor_tesis(request.POST)
        if form.is_valid():
            data = form.cleaned_data
            try:
                pers=Persona.objects.get(nombre= data['nombre'])
                persona=Persona.objects.get(apellidos= data['apellidos'])
                if(pers.solapin==persona.solapin):
                    if(Profesor.objects.filter(trabajadorpersonasolapin = pers.solapin).exists()):
                        prof=Profesor.objects.get(trabajadorpersonasolapin = pers.solapin)
                        if(ProfesorTesis.objects.filter(id_tutor=prof.trabajadorpersonasolapin
                            .personasolapin.solapin)):
                            tesis=ProfesorTesis.objects.filter(id_tutor=prof.trabajadorpersonasolapin
                                .personasolapin.solapin)
                            return render_to_response('Tesis_turoreada_por_el_profesor.html',{'form': form,
                                                                                                    'tesis': tesis
                                                                                                    },
                                context_instance = RequestContext(request))
            except Exception as e:
                errores.append('El profesor no existe ')
        else:
            form = Filtro_Profesor_tesis()
            list_tesis = Tesis.objects.all()
            return render_to_response('Tesis_turoreada_por_el_profesor.html',{'form': form,
                                                                                                    'list_tesis': list_tesis,
                                                                                                    'errores': errores },
                                context_instance = RequestContext(request))
```

Figura 6 Patrón de diseño utilizando vista genérica.

En el ejemplo anterior también se evidencia el uso de otra de las buenas prácticas propuestas por django como en este caso una única implementación para las llamadas de los métodos GET y POST.

Otro de los elementos que propone django como reutilización, es el uso de los formularios de tipo ModelForms permitiendo su asociación con clases del modelo de manera tal que no haya que especificar donde ni cómo persistir este objeto.

La capa de presentación que django propone es a través del uso de plantillas, dicha estructura se compone por un conjunto de páginas html, donde existe un fichero base.html general del cual extienden los demás ficheros de esta categoría. Ejemplo:



```
{% extends "Index.html" %}
{% block Menu %}
  <div id="menu" class="nav">
    <ul>
      <li><a href="{% url GestionRecursosHumanosTecnologias.Gestion.views.l
      <li><a href="{% url GestionRecursosHumanosTecnologias.Gestion.views.l
      <li><a href="http://localhost:8000/admin/">Administración</a></li>
      <li><a href="{% url GestionRecursosHumanosTecnologias.Gestion.views.l
    </ul>
  </div>
{% endblock %}
```

Figura 7 Patrón de diseño en las plantillas.



```
<body id="prof_proy">
  {% extends "Base Reporte.html" %}
  {% block WorkSpace %}
    <div class="art-post-inner art-article">
      <div class="art-postmetadataheader">
        <h2 class="art-postheader"></h2>
      </div>
    </div>
  {% endblock %}
</body>
```

Figura 8 Patrón de diseño en las plantillas.

2.5.6. Diagrama de clases

En este documento serán expuestos los diagramas de clases los cuales proporcionan una mayor comprensión de las funcionalidades que se desean implementar, a continuación se especifica el diagrama que correspondiente a la historia de usuario Autenticar usuario y Mostrar reporte, el resto se localizan en los anexos (ver Anexo 3):

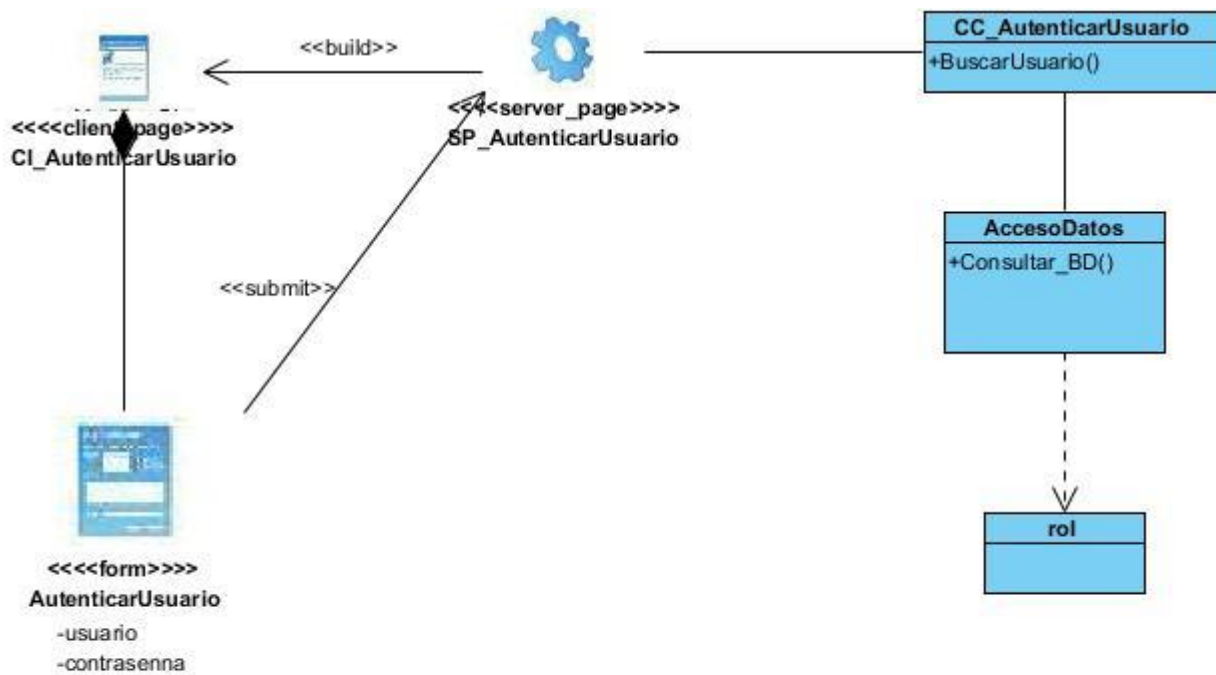


Figura 14 Diagrama de clases AutenticarUsuario

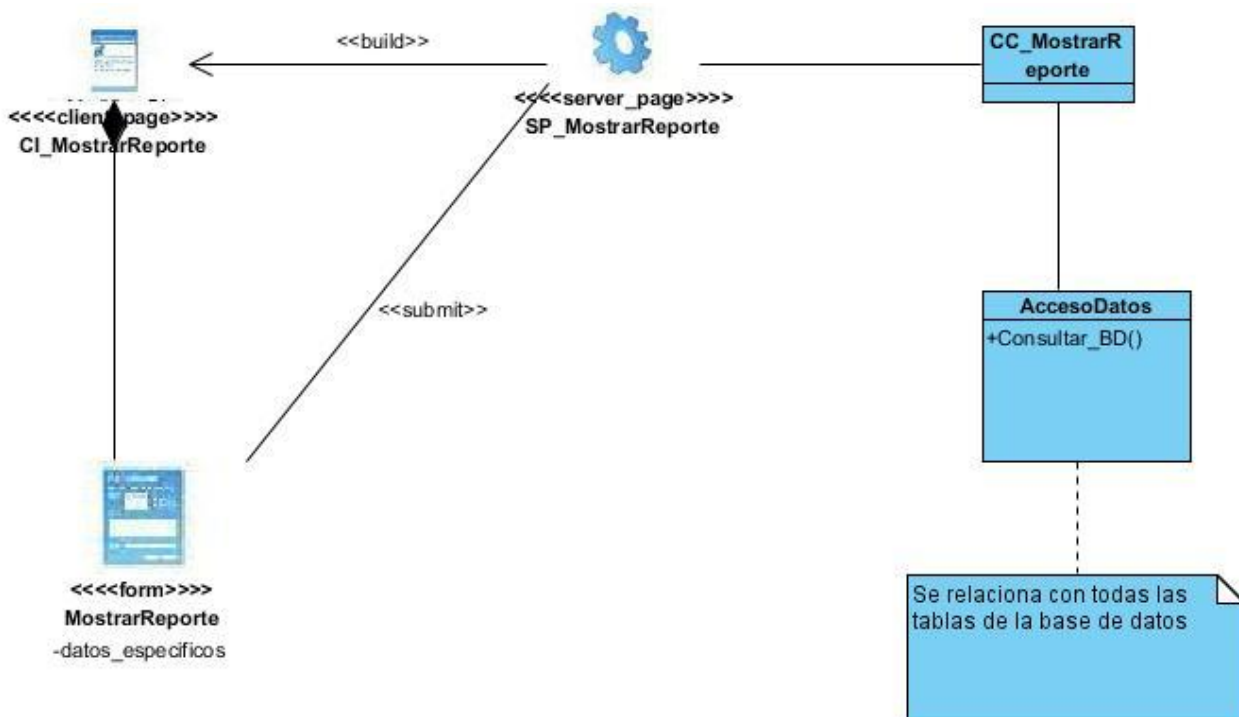


Figura 15 Diagrama de clases MostrarReporte

2.5.7. Modelo de dato

Es un conjunto de conceptos que nos permiten describir los datos, las relaciones que existen entre ellos, la semántica y las restricciones de consistencia. (34) **(ver Anexo 4)**

2.6. Planificación

En la fase de planificación se priorizan las historias de usuario, siendo el cliente quien determina esta prioridad. Por otra parte los programadores estiman cuánto esfuerzo requiere cada una, definiendo el cronograma y en conjunto con el cliente se toman los acuerdos de la primera entrega. La planificación se puede realizar basándose en el tiempo o el alcance y sólo toma uno o dos días aproximadamente. Una entrega debería obtenerse en no más de tres meses. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. (31)

2.6.1. Estimación de esfuerzo por cada una de las historias de usuario identificadas

La estimación del esfuerzo es determinada por la asignación de “peso” a un grupo de factores que inciden en el tiempo de realización de cada historia de usuario y que significará la cantidad de semanas que llevará la realización de cada una de ellas por separado. El rango de valores estará entre 1 y 3 respectivamente. A continuación se muestra la tabla con la información obtenida:

Historias de usuario	Puntos estimados
Autenticar usuario	3
Gestionar usuario	1
Gestionar estudiante	1
Gestionar trabajador	2
Gestionar máquinas	2
Mostrar reportes	3

Gestionar profesor	2
--------------------	---

Tabla 11 Estimación de esfuerzo de la historias de usuario.

2.6.2. Plan de Iteraciones

Una iteración es un período de tiempo en el que se realizan un conjunto de funcionalidades determinadas, que en el caso de la metodología aplicada corresponden a un grupo de historias de usuarios. Para determinar la cantidad de iteraciones se debe tener en cuenta la velocidad del proyecto, la cual es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de ellas. Para maximizar el valor del negocio de cada historia de usuario, el cliente es quien decide cuales se implementarán en cada iteración. (31)

Luego de una consulta previa con el cliente y de estimar los esfuerzos de cada historia de usuario se llegó a la siguiente distribución.

- **Iteración 1:** serán realizadas las historias de usuario que responden a las funcionalidades de menor complejidad como es el caso de:
 - Gestionar usuario.
- **Iteración 2:** se implementarán las historias de usuario que encierran las funcionalidades de mediana complejidad como es el caso de:
 - Gestionar trabajador.
 - Gestionar estudiante.
 - Gestionar profesor.
 - Gestionar máquinas.
- **Iteración 3:** se codificarán las historias de usuarios de mayor complejidad de implementación según las funcionalidades que representan como son el caso de:
 - Autenticar usuario.
 - Mostrar reporte.

Número de la iteración.	Historias de usuario según la prioridad	Duración en semanas.
1	<ul style="list-style-type: none"> • Gestionar usuario 	1

2	<ul style="list-style-type: none"> • Gestionar trabajador • Gestionar estudiante • Gestionar profesor • Gestionar máquinas 	7
3	<ul style="list-style-type: none"> • Autenticar usuario • Mostrar reporte 	6

Tabla 12 Plan de iteración.

2.6.3. Plan de entrega

Especifica exactamente qué historias de usuario serán implementadas en cada entrega del sistema y sus prioridades, además de detallar con exactitud cuáles serán implementadas en la próxima liberación. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo desarrollador durante las reuniones de planificación de entregas. (35)

La siguiente tabla muestra la fecha en que serán entregadas las diferentes versiones al cliente hasta obtener la versión 1.0 que constituye la versión final del producto, proceso que es permitido por la metodología utilizada, la cual posibilita la entrega de versiones del producto a corto plazo al cliente con el propósito de comprobar gradualmente su nivel de aceptación.

Entregable	Fin iteración 1 17 de marzo de 2013	Fin iteración 2 25 de abril de 2013	Fin iteración 3 16 de mayo de 2013
Cliente	0.1	0.2	1.0

Tabla 13 Entrega del producto en cada iteración.

2.7. Diseño

Para conseguir un diseño simple y sencillo XP se apoya en la construcción de las tarjetas CRC las cuales constituyen una alternativa de los UML y representan los principales objetos del sistema con sus responsabilidades y dependencias.

A continuación se muestran las principales tarjetas CRC, las restantes están definidas en los anexos (**ver Anexo 5**):

2.7.1. Tarjetas CRC

CLASE: TRABAJADORLISTVIEW	
Responsabilidades	Colaboradores
Listar datos de los trabajadores	<ul style="list-style-type: none"> • CargoOcupa • Proyecto • Profesor • AsignaturaGrupoDocenteProfesor • Asignatura • Persona • TrabajadorEvento • TrabajadorPublicaciones • ProfesorCursaPosgrado • ProfesorImpartePosgrado • Tesis • RolPersonaProyecto

Tabla 14 Tarjeta CRC 1. TrabajadorListView

CLASE: TRABAJADOR	
Responsabilidades	Colaboradores
Guardar la información referente a un trabajador	<ul style="list-style-type: none"> • Trabajador_Evento • Trabajador_Publicaciones • Persona • Profesor

Tabla 10 Tarjeta CRC 2. Trabajador

CLASE: PERSONA	
Responsabilidades	Colaboradores

<p>Guardar la información referente a una persona</p>	<ul style="list-style-type: none"> • Trabajador • Municipio • Estudiante • Cargo_Ocupa • Maquina • Oficina • Área_Física • Laboratorio • Departamento • Rol_Persona
---	---

Tabla 15 Tarjeta CRC 3. Persona

2.7.2. Tareas de ingeniería

Las tareas de la ingeniería componen el conjunto de acciones llevadas a cabo para darle cumplimiento a las distintas historias de usuario, las cuales son realizadas para la especificación de las operaciones ejecutadas por los programadores en cada una de ellas.

A continuación se muestran las principales tareas de ingeniería derivadas de cada historia de usuario, para consultar el resto (**ver Anexo 6**):

TAREA DE INGENIERÍA	
Número de tarea: HU1_T1	Historia de usuario: Autenticar usuario.
Nombre: Obtener datos del usuario y dar autorización según su rol.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: El sistema toma los datos del usuario y le permite el acceso según su rol.	

Tabla 22 Tarea de ingeniería 1. Autenticar usuario.

TAREA DE INGENIERÍA	
Número de tarea: HU2_T2	Historia de usuario: Administrar usuario.
Nombre: Se encarga de la administración de los usuarios.	
Tipo: Desarrollo.	Puntos Estimados: 1

Responsable: Yam Brayam Beltran Rodriguez.
Descripción: Se encarga de crear los usuarios y darle los permisos correspondientes según su rol.

Tabla 23 Tarea de ingeniería 2. Administrar usuario.

Conclusiones

Con el uso de la metodología XP, en el capítulo se describieron los principales requisitos funcionales y no funcionales como parte de la etapa del análisis permitiendo una mayor comprensión de las características del software a desarrollar. Unido a esto, se lograron describir cada una de las funcionalidades de la propuesta de solución a partir de la utilización de las historias de usuarios. La definición de la arquitectura y la selección de los patrones de diseño establecieron una guía que definió la estructura y componentes del producto final. Además con la elaboración de las tareas de la ingeniería y las tarjetas CRC se logró una mejor comprensión del funcionamiento integrado de todo el sistema.

Capítulo III: Implementación y Pruebas.

Introducción

En este capítulo es conformado el diagrama de despliegue el cual muestra las relaciones físicas entre los distintos nodos que componen el sistema una vez desplegado y el diagrama de componente donde se muestra la relación y dependencia entre sus diferentes componentes. Se realizan las pruebas de aceptación del cliente las cuales muestran el nivel de conformidad del mismo con el producto elaborado.

3.1. Implementación

La presencia del cliente en las diferentes fases de XP como parte del equipo de trabajo es indispensable. A la hora de codificar o implementar una historia de usuario es aún más necesaria su presencia pues son ellos los que crean las historias de usuario, negocian los tiempos en los que serán implementadas, especifican detalladamente lo que esta hará y verifican que la historia implementada cumpla la funcionalidad especificada cuando se realicen las pruebas. La codificación debe hacerse atendiendo a estándares de codificación ya creados. Los programadores lo han de seguir de tal manera que el código en el sistema se vea como si hubiera estado escrito por una sola persona. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. (31)

3.1.1. Estándares de codificación

Los estándares de codificación surgieron con el propósito de que un código fuera capaz de ser entendido por cualquier programador independientemente de su creador.

A continuación se muestran los distintos codificadores utilizados durante el desarrollo del producto.

3.1.1.1. Comentarios

Hay dos tipos de comentarios: los de implementación y los de documentación. Los comentarios de implementación describen el código, los de documentación una visión de más alto nivel para gente que no tiene el código a mano y que sólo quiere usarlo. La información de los comentarios no debe ser evidente, deben ser cosas que no están contenidas en el código. Los comentarios no deben estar en cajas dibujadas con asteriscos o similar ni deben incluir caracteres extraños. (36)

Ejemplo de utilización de comentarios de implementación y documentación respectivamente:

```
def Dpto_del_Est(request):
    errores=[]
    if request.method == 'POST':
        form = Filtro_Profesor_tesis(request.POST)
        if form.is_valid():
            data = form.cleaned_data
            try:
                persona=Persona.objects.get(nombre= data['nombre'])
                per=Persona.objects.get(apellidos = data['apellidos'])
                if(per.solapin == persona.solapin):
                    #comprobar si es estudiante
                    if(Estudiante.objects.filter( personasolapin = persona.solapin).exists()):
                        dpto= Departamento.objects.filter(id_dpto = persona.departamentoid_dpto.id_dpto)
                        return render_to_response('Dpto_del_Est.html',{'form': form,'dpto': dpto[0] , 'errores': errores },
                            context_instance = RequestContext(request))
            except Exception as e:
                errores.append('El estudiante no existe')
        else:
            form = Filtro_Profesor_tesis()
            list_est = Estudiante.objects.all()
            return render_to_response('Dpto_del_Est.html',{'form': form,'list_est': list_est , 'errores': errores },
                context_instance = RequestContext(request))
```

Figura 22 Codificador de comentario de implementación.

```
# Absolute filesystem path to the directory that will hold user-uploaded files.
# Example: "/home/media/media.lawrence.com/media/"
MEDIA_ROOT = ''

# URL that handles the media served from MEDIA_ROOT. Make sure to use a
# trailing slash.
# Examples: "http://media.lawrence.com/media/", "http://example.com/media/"
MEDIA_URL = ''

# Absolute path to the directory static files should be collected to.
# Don't put anything in this directory yourself; store your static files
# in apps' "static/" subdirectories and in STATICFILES_DIRS.
# Example: "/home/media/media.lawrence.com/static/"
STATIC_ROOT = os.path.join(os.path.dirname(__file__), 'Static')
```

Figura 23 Codificador de comentario de documentación.

3.1.1.2. Líneas largas

Cuando una línea no quepa en una única línea se debe fraccionar atendiendo a estos principios generales

Fraccionar después de una coma

Fraccionar después de un operador

Es mejor romper unidades de alto nivel que de bajo nivel (36) .Ejemplo:

```
from GestionRecursosHumanosTecnologias.Gestion.form import *
from models import Persona
from models import Estudiante, Profesor, ProfesorTesis, EstudianteAutorTesis, Evento, Publicaciones, Posgrado, TrabajadorEvento, \
    TrabajadorPublicaciones, ProfesorCursaPosgrado
from models import Tesis, Trabajador, ActivoFijo, AreaFisica, Laboratorio, Departamento, Proyecto, CargoOcup, Asignatura, Oficina, \
    ProfesorTesis, AsignaturaGrupoDocenteProfesor, ProfesorCursaPosgrado, ProfesorImpartePosgrado, Rol, RolPersonaProyecto
from models import Maquina
from django.views.generic import ListView
from django.core.exceptions import ObjectDoesNotExist
```

Figura 24 Codificador de línea larga.

3.1.1.3. Declaraciones

Se debe declarar cada variable en una línea distinta, no se deben declarar variables con el mismo nombre que otras en una misma función o método (36). Ejemplo:

```
def Listado_prof_dado_categoria_cientifica(request):
    errores=[]
    if request.method == 'POST':
        form =Cate_Cientifica(request.POST)
        if form.is_valid():
            data = form.cleaned_data
            try:
                prof = Profesor.objects.filter(categoria_cientifica = data['categoria_cientifica'])
                return render_to_response('Listado_Prof_Dado_Categoria_Cientifica.html',{'form': form, 'prof': prof, 'errores': errores},
                    context_instance = RequestContext(request))
            except Exception as e:
                errores.append('La categoria cientifica no existe')

    else:
        form =Cate_Cientifica()
        profesores=Profesor.objects.all()
        return render_to_response('Listado_Prof_Dado_Categoria_Cientifica.html',{'form': form, 'profesores': profesores, 'errores': errores},
            context_instance = RequestContext(request))
```

Figura 25 Codificador de declaraciones.

3.1.1.4. Declaraciones de clases e interfaces

No hay espacio entre el nombre del método, el paréntesis y la lista de parámetros (36). Ejemplo:

```
def Datos_categoria_docente(request):
    if request.method == 'POST':
        form = Formulario_Categoria_Docente(request.POST)
        if form.is_valid():
            data = form.cleaned_data
            profesor = Profesor.objects.filter(categoria_docente = data['categoria_docente'])
            return render_to_response('Listado_Prof_Dado_Categoria_Docente.html', {'form': form, 'profesor': profesor},
                context_instance = RequestContext(request))
        else:
            form = Formulario_Categoria_Docente()
            profesores = Profesor.objects.all()
            return render_to_response('Listado_Prof_Dado_Categoria_Docente.html', {'form': form, 'profesores': profesores},
                context_instance = RequestContext(request))
```

Figura 26 Codificador de declaraciones de clases de interfaces.

3.1.1.5. Sentencias de retorno

No se pondrá la expresión de retorno entre paréntesis a menos que con ello se gane en claridad (36).

Ejemplo:

```
def Listado_profesores_asignatura_posgrado(request):
    list_prof=Profesor.objects.all()
    list_curso_posgrado=ProfesorCursaPosgrado.objects.all()
    return render_to_response('Listado_Profesores_Asignatura_Posgrado.html',{'list_prof':list_prof,'list_curso_posgrado':list_curso_posgrado},
        context_instance = RequestContext(request))

def Listado_profesor_rol(request):
    list_prof=Profesor.objects.all()
    lista_rol=RolPersonaProyecto.objects.all()
    return render_to_response('Listado_Profesor_Rol.html',{'list_prof': list_prof, 'lista_rol': lista_rol},
        context_instance = RequestContext(request))
```

Figura 27 Codificador de sentencia de retorno.

3.1.2. Tareas de implementación para cada historia de usuario.

A continuación se muestran las principales tareas de implementación, para consultar el resto (ver Anexo 7):

TAREA	
Número de la tarea: 1	Historia de usuario: Autenticar usuario.
Nombre de la tarea: Obtener datos del usuario y dar autorización según su rol.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	

Descripción: El sistema muestra la opción para que el usuario pueda autenticarse.

Tabla 41 Tarea 1 de implementación. Autenticar usuario.

TAREA	
Número de la tarea: 2	Historia de usuario: Gestionar usuario.
Nombre de la tarea: Se encarga de la administración de los usuarios.	
Tipo de tarea: Desarrollo.	Puntos estimados: 3
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra la opción para que el administrador pueda crear un usuario, modificar un usuario y eliminar un usuario.	

Tabla 42 Tarea 2 de implementación Gestionar usuario.

TAREA	
Número de la tarea: 7	Historia de usuario: Gestionar estudiantes.
Nombre de la tarea: Modificar estudiante.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de un estudiante.	

Tabla 43 Tarea 3 de implementación. Gestionar estudiantes.

3.1.3. Diagrama de componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Debido a que los diagramas de componentes son más parecidos a los diagramas de casos de usos, éstos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema. (37)

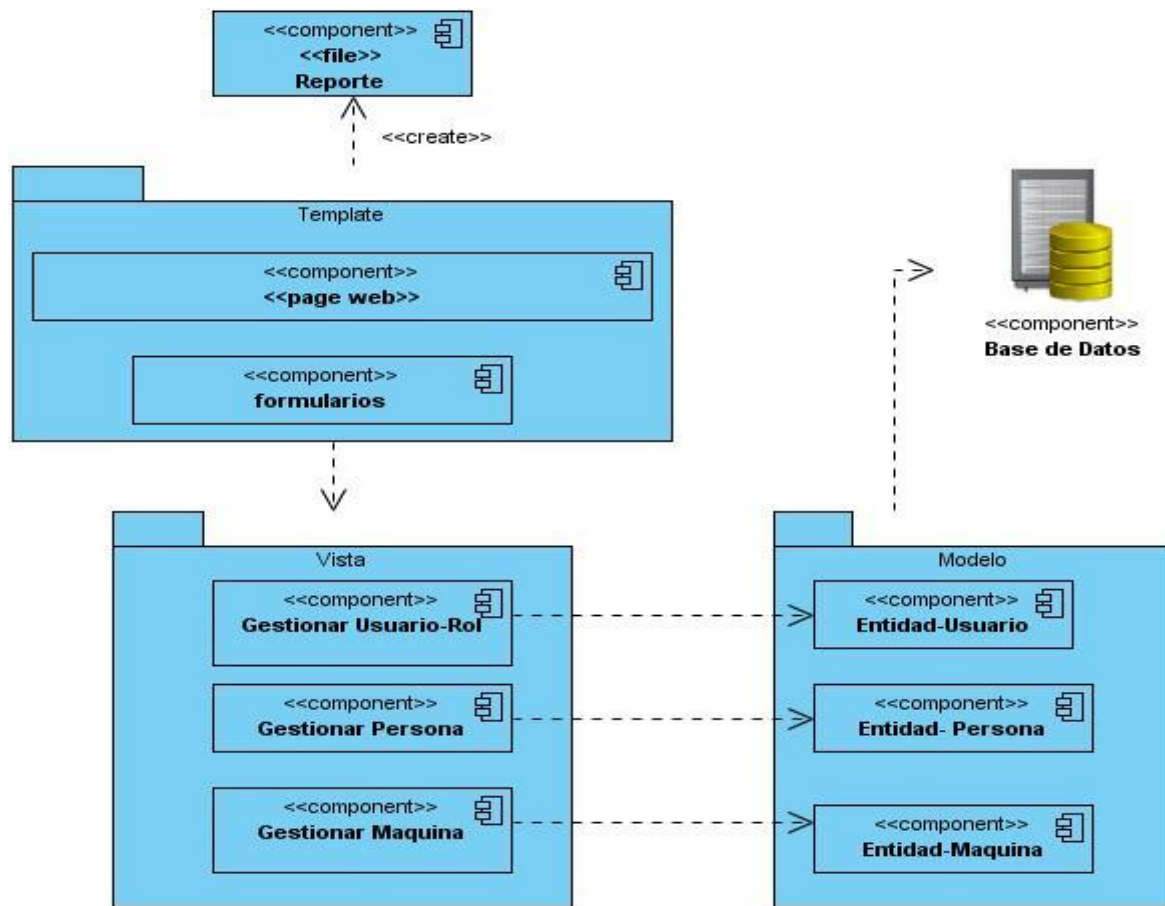


Figura 28 Diagrama de Componentes

3.1.4. Interfaces del sistema

El sistema desarrollado cuenta de dos secciones, una donde se realizan todas las operaciones de adicionar, eliminar y modificar la información relacionada con los estudiantes, trabajadores, profesores y equipos técnicos apoyada en la administración del sitio de django y la otra constituye un sitio web donde son mostrados los reportes deseados por el cliente y recogidos en la historia de usuario Mostrar reporte. A continuación se presentan las principales interfaces, para consultar el resto ([ver Anexo 8](#)):



Figura 29 Interfaz Autenticar usuario

Sistema de Gestión de Información de Recursos Humanos y Tecnológicos

Inicio **Reportes** Administración Cerrar Sesión

Reportes

- > Datos de las personas
- > Listado de estudiantes
- > Listado de trabajadores
- > Tesis tuoreada por un profesor
- > Características de la pc
- > Año graduado del profesor
- > Departamento del profesor
- > Proyecto del profesor
- > Trabajador y su categoría
- > PC del estudiante

Nombre	Apellidos	Solapín	Año académico
Rocio	Alvarez Gonzalez	EH-04403	2013
Yam Brayam	Beltran Rodriguez	eh04217	2013
Nilberto	Rodriguez Rojas	eh 786514	2013
Herber	Tamayo Valero	eh 04284	2013

« Anterior
Siguiete »

Figura 30 Interfaz Mostrar reporte de estudiantes

3.1.5. Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores, memoria. Los nodos se interconectan mediante soportes bidireccionales que pueden a su vez estereotiparse. Esta vista permite determinar las consecuencias de la distribución y la asignación de recursos. Las instancias de los nodos pueden contener instancias de ejecución, como instancias de componentes y objetos. (38)

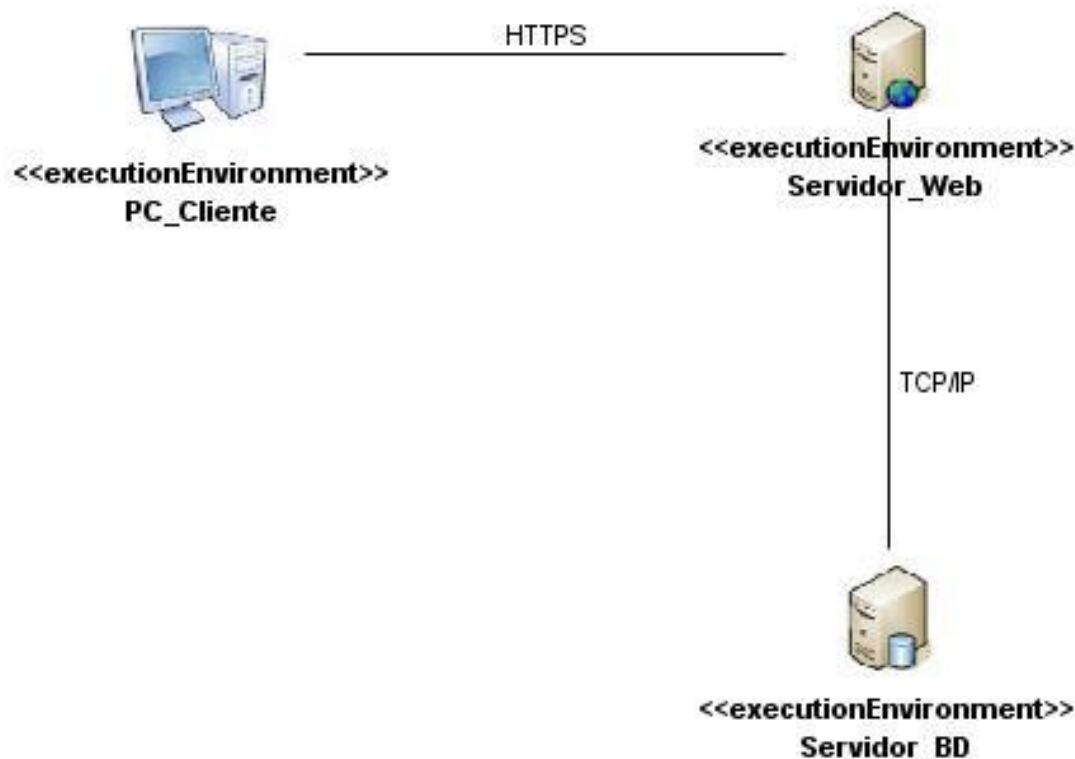


Figura 33 Diagrama de Despliegue

3.2. Pruebas

La metodología XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. (39)

3.2.1. Pruebas de aceptación

Las pruebas de aceptación XP, también llamadas pruebas del cliente son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptaciones derivan de las historias de los usuarios que se han implementado como parte de la liberación del software. (40)

A continuación se mencionan las pruebas de aceptación de las funcionalidades realizadas al sistema:

- Pruebas de aceptación para la historia de usuario Autenticar usuario.
- Pruebas de aceptación para la historia de usuario Administrar usuario.
- Pruebas de aceptación para la historia de usuario Gestionar trabajadores.
- Pruebas de aceptación para la historia de usuario Gestionar máquina.
- Pruebas de aceptación para la historia de usuario Mostrar reporte.
- Pruebas de aceptación para la historia de usuario Gestionar estudiante.
- Pruebas de aceptación para la historia de usuario Gestionar profesor.

3.2.2. Pruebas de aceptación para cada historia de usuario

A continuación se muestran las principales pruebas de aceptación para consultar el resto (**ver Anexo 9**):

Caso de Prueba de Aceptación.
Historia de usuario: Autenticar usuario.
Nombre: Autenticar usuario.
Descripción: Prueba para comprobar si el usuario está registrado.
Condiciones de Ejecución: La persona debe tener asignado previamente un usuario y una contraseña.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar usuario y contraseña.
Resultado Esperado: La autenticación ha sido realizada satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 60 Caso de Prueba de Aceptación 1. Autenticar usuario.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar usuario.
Nombre: Adicionar usuario.
Descripción: Prueba para comprobar si se ha añadido un usuario en el sistema.
Condiciones de Ejecución: Para adicionar un usuario se debe estar autenticado en el sistema como administrador.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la opción usuario. • Seleccionar la opción añadir usuario. • Llenar los campos del formulario. • Seleccionar la opción guardar.
Resultado Esperado: Se adicionó con éxito el usuario "nombre del usuario añadido".
Evaluación de la Prueba: Satisfactoria.

Tabla 61 Caso de Prueba de Aceptación 2. Gestionar usuario.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar trabajadores.
Nombre: Adicionar trabajadores.
Descripción: Prueba para comprobar si se adiciona un trabajador en el sistema.
Condiciones de Ejecución: Para gestionar un trabajador se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la opción trabajador. • Seleccionar la opción añadir trabajador. • Llenar los campos del formulario. • Seleccionar la opción guardar.
Resultado Esperado: Se añadió con éxito el trabajador "nombre del trabajador".
Evaluación de la Prueba: Satisfactoria.

Tabla 62 Caso de Prueba de Aceptación 3. Gestionar trabajadores.

Conclusiones

La confección de los diagramas de componentes y de despliegue arrojaron una detallada descripción de la estructura de la solución una vez desplegada. Fueron definidos los estándares de codificación, los cuales proponen las mejores prácticas de programación, ayudando aún mejor entendimiento del código. Además se elaboraron los casos de pruebas por cada una de las funcionalidades que debe cumplir el software obtenido, las cuales demostraron una gran satisfacción por parte del cliente.

Conclusiones Generales

Luego de la investigación realizada y con el objetivo de apoyar el proceso de gestión de información de Recursos Humanos y Tecnológicos para el CISED, se arribó a las siguientes conclusiones:

Se realizó un estudio teórico donde pudo constatar que ninguna de las aplicaciones existentes resuelven el problema o pueden ser utilizadas para resolver la situación existente en el CISED.

Fue identificada toda la información relacionada con los recursos humanos y tecnológicos del CISED, la cual proporcionó una mejor comprensión de todos los procesos relacionados con la gestión de información en dicho centro.

Se logró implementar un sistema de gestión de información que facilita al personal del CISED la gestión de los recursos humanos y tecnológicos.

Se realizaron las pruebas al sistema las cuales permitieron corroborar que este cumple con las funcionalidades previstas.

Recomendaciones

Una vez cumplido el objetivo general propuesto se recomienda:

- ✓ Poner en práctica la propuesta realizada y hacer extensible el sistema a otras áreas.
- ✓ Agregar nuevas funcionalidades según las necesidades que vayan surgiendo en el transcurso del tiempo.

Bibliografía Referenciada

1. **Santos, Armando Cuesta.** *TECNOLOGÍA DE GESTIÓN DE RECURSOS HUMANOS.* 2010.
2. **Dueñas Prendes , Lourdes M. .** *Caracterización de un Sistema de Gestión de Información Científico Tecnológica con enfoque a procesos: garantía para la mejora continua. Estudio de caso.* Ciudad de la Habana : s.n., 2004.
3. Promonegocios. [En línea] [Citado el: 3 de abril de 2013.]
<http://www.promonegocios.net/mercadotecnia/que-es-informacion.html>.
4. Definición.DE. [En línea] [Citado el: 2 de abril de 2013.] definicion.de/informacion/.
5. Definición.DE. [En línea] [Citado el: 22 de febrero de 2013.] definicion.de/gestion/.
6. **Ponjuán Dante, Gloria .** *Gestioón de información en las organizaciones principios, conceptos y aplicaciones.* Rosario : Ediciones Nuevo Paradigma, 2004.
7. THE FREE DICTIONARY. [En línea] [Citado el: 12 de diciembre de 2013.]
<http://es.thefreedictionary.com/recurso..>
8. ASINOM. [En línea] [Citado el: 10 de diciembre de 2012.] <http://www.asinom.com/que-son-recursos-humanos.php>.
9. Definición.DE. [En línea] [Citado el: 10 de diciembre de 2012.] definicion.de/recursos-tecnologicos/.
10. Definición.DE. [En línea] [Citado el: enero de 7 de 2013.] <http://definicion.de/sistema/>.
11. Definición.DE. [En línea] [Citado el: 13 de enero de 2013.] <http://definicion.de/sistema-de-informacion/>.
12. Mejora tu gestión. [En línea] [Citado el: 8 de abril de 2013.] <http://mejoratugestion.com/mejora-tu-gestion/que-es-un-sistema-de-gestion/>.
13. **Dieguez, Leyanis Bernal y Peña Rodríguez, Yilian .** *Sistema de Gestión de Información de los Recursos Humanos de la Facultad 2.* Ciudad de la Habana : s.n., 2009.
14. **Martínez Toro , Carlos Mario .** *Sistema de información documental diseñado para la gestión de correspondencia de la universidad de ANTIOQUIA.* 2012 .
15. Hum&Software S.A. de C.V. Monterrey N.L. México. [En línea] [Citado el: 10 de febrero de 2013.]
<http://www.humsoftware.com/>.
16. **Reyes Vargas, Lillianne y Quesada Guerrero, Diego.** *Sistema de Gestión de Recursos Humanos de la Unión de Empresas de Recuperación de Materias Primas.* La Habana : s.n., junio 2010.
17. **Santander Yhanes, Renier y Rosales Ramírez , Alberto Esteban .** *Sistema de Control de los recursos humanos y didácticos en los Centros de Auto--aprendizaje y Servicios de Idiomas Extranjeros ((CASIE)).* . Ciudad de la Habana : s.n., julio del 2010 .

18. **Montoya Rodríguez, Elda.** *Tema: Sistema de Gestión de Información de los Recursos Humanos (SGIRH) de la Facultad 2. Módulo de Estudiantes Módulo de Profesores Módulo de Proyectos Productivos.* Ciudad de la Habana : s.n., 2009.
19. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aureliano David.** *METODOLOGIA DE DESARROLLO DEL SOFTWARE.* 2008.
20. **Reyes González, Ing. Yunia, González Bravo, Ing. Lisbet M. y Ruiz Constanten, Msc. Yadira .** *ESTUDIO COMPARATIVO SOBRE LAS PRINCIPALES METODOLOGÍAS PESADAS Y ORIENTADAS A OBJETO EN EL DESARROLLO DE SOFTWARE.* 2008.
21. Zoho Docs. [En línea] [Citado el: 2 de febrero de 2013.]
https://export.writer.zoho.com/public/ana_karen/metodologia-rup/fullpage.
22. **Letelier, Patricio y Penadés, M^a Carmen.** *Métodologías ágiles para el desarrollo de software.* s.l. : Universidad politécnica de Valencia.
23. Ciencia y Técnica Administrativa. [En línea] 2008. [Citado el: 2 de febrero de 2013.]
<http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
24. **Holovaty, Adrian y Kaplan-Moss, Jacob .** *El libro de Django.* 2008.
25. Competencias en Tic. [En línea] [Citado el: 6 de febrero de 2013.]
http://competenciastic.educ.ar/pdf/lenguajes_de_programacion_4.pdf.
26. PyCharm. [En línea] 2013. [Citado el: 5 de febrero de 2013.]
www.jetbrains.com/pycharm/features/index.html.
27. **Nodarse García, Fernando.** *SIB: Arquitectura e integración de subsistemas.* Ciudad de la Habana : s.n., junio de 2011.
28. **Bernal Dieguez, Leyanis y Pena Rodriguez, Yilian.** *Sistema de Gestion de Informacion de los Recursos Humanos de la Facultad 2 : modulos: Investigacion, Postgrado, Actividades, Administracion, Reportes.* La Habana : s.n., 2009.
29. **Montoya Rodríguez, Elda.** *Sistema de Gestion de Informacion de los Recursos Humanos (SGIRH) de la Facultad 2 : Modulo de Estudiantes : Modulo de Profesores : Modulo de Proyectos Productivos.* Ciudad de la Habana : s.n., 2009.
30. TECNOLOGÍA Y SYNERGIX. [En línea] 2010. [Citado el: 18 de febrero de 2013.]
nergix.wordpress.com/2008/07/10/modelo-de-dominio/.
31. **Rodríguez Ramírez, Idalmis y Hernández Armas, Alejandro Luis.** *Sistema de gestión para el Departamento de Recursos Humanos de la Dirección de Producción Audiovisual del centro FORTES.* Ciudad de La Habana : s.n., 2011.
32. Scribd. [En línea] [Citado el: 22 de febrero de 2013.] <http://es.scribd.com/doc/105808242/Investigacion-Modelo-Vista-Controlador>.

33. **MARTÍNEZ JUAN, FRANCISCO JAVIER.** *Guía de Construcción de Software en java con Patrones de Diseño.* OVIEDO : s.n.
34. **Fernández Rivera, Javier.** aurea. [En línea] [Citado el: 23 de marzo de 2013.] <http://aurea.es/wp-content/uploads/modelodedatos.pdf>.
35. Documeto de Informática. [En línea] [Citado el: 20 de marzo de 2013.] http://www.inf.utfsm.cl/~visconti/xp/Plan_Entregas_2.doc.
36. **Arias Calleja, Manuel .** Centro de Investigación sobre Sistemas Inteligentes de Ayuda a la Decisión Universidad Nacional de Educación a Distancia (UNED). [En línea] [Citado el: 3 de marzo de 2013.] <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>.
37. Lenguaje Unificado de Modelado UML. [En línea] [Citado el: 12 de marzo de 2013.] <http://umlhugomartinez.blogspot.com/p/diagramas-de-estructura.html>.
38. **Marca Huallpara, Hugo Michael y Quisbert Limachi , Nancy Susana.** Universidad Sale Siana Bolivar. [En línea] [Citado el: 20 de marzo de 2013.] virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc.
39. **J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres.** Lenguajes y Sistemas Informáticos. [En línea] [Citado el: 21 de marzo de 2013.] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf.
40. Site. [En línea] [Citado el: 21 de marzo de 2013.] <https://sites.google.com/site/xpmetodologia/marco-teorico/funcionamiento>.

Bibliografía Consultada

1. Que es un sistema, sus tipos y su teoria? [En línea] [Citado el: 8 de abril de 2013.] <http://informatica-colegiom.foroactivos.net/t12-que-es-un-sistema-sus-tipos-y-su-teoria>.
2. ¿Qué es un sistema de información? [En línea] [Citado el: 23 de enero de 2013.] <http://dsp.mx/blog/sistemas-de-informacion/53-que-es-un-sistema-de-informacion>.
3. Gvsistemas. [En línea] [Citado el: 2 de diciembre de 2012.] <http://www.gvsistemas.com.ar/>.
4. Sistema de información documental diseñado para la gestión de correspondencia de la universidad de ANTIOQUIA. [En línea] [Citado el: 2 de diciembre de 2012.] tesis.udea.edu.co/jspui/bitstream/10495/.../MONOGRAFIA%20%20E.I.B.p...
5. Metodologías de desarrollo de software. [En línea] [Citado el: 2 de diciembre de 2012.] www.ecured.cu/index.php/Metodologías_de_desarrollo_de_software.
6. Que es un framework de desarrollo web. [En línea] diciembre de 2010. [Citado el: 10 de diciembre de 2012.] <http://kokiing.blogspot.com/2010/12/que-es-un-framework-de-desarrollo-web.html>.
7. Django. [En línea] [Citado el: 8 de abril de 2013.] <http://www.djangoproject.com/>.
8. Qué es un entorno de desarrollo integrado, IDE. [En línea] [Citado el: 10 de abril de 2013.] <http://programaciondesarrollo.es/que-es-un-entorno-de-desarrollo-integrado-ide/>.
9. PostgreSQL-es. [En línea] [Citado el: 14 de enero de 2013.] http://www.postgresql.org.es/sobre_postgresql.
10. **Leyva, Armando Alexander Rosales y Márquez Figueroa , Angel.** *Análisis y Diseño del módulo Desarrollo del personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3.* Ciudad de la Habana : s.n., 2009.
11. MODELO DEL DOMINIO. [En línea] 2003. [Citado el: 25 de febrero de 2013.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.
12. Diseño. [En línea] [Citado el: 10 de diciembre de 2012.] <http://tomotesis.googlecode.com/svn/trunk/tomo/apendiceXP.tex>.
13. **Santana, Yaumi Ruiz y Arufe Vega, Dayami .** *Sistema de Gestión de los Recursos Humanos en el Politécnico de Informática "América Labadí Arce".* Ciudad de la Habana : s.n., 2007.
14. **Ochoa, Manuel Ramón Almaguer.** *Sistema de Recursos Humanos.* Ciudad de la Habana : s.n., 2007.
15. **Andino, Tania Sachyn González y Ramos Oliva, Renier .** *Sistema de Gestión de Información en el Proceso de Formación de Recursos Humanos en la Salud. Componente de Gestión Académica.* Ciudad de la Habana : s.n., 2008.

16. **Mendieta, Yisell De la Peña y Vega Rivero, Aliesky** . *Marco de Procesos Primarios para la Gestión de Recursos Humanos en proyectos de software de la Facultad 3*. Ciudad de la Habana : s.n., 2008.
17. **Rosa, Aniuska Vivas de la y Vivas de la Rosa, Aliuska**. *Proceso de pruebas de software para el Sistema Integral para la Gestión de Recursos Humanos (Ceres) de la Aduana General de la República (AGR)*. Ciudad de la Habana : s.n., 2008.
18. **Mejías, Yensy Yoan Guerrero y Rill Arencibia, Edgar**. *Sistema informático para la administración de los Recursos Humanos. Módulo de selección de candidatos*. Ciudad de la Habana : s.n., 2009.
19. **Wilson, Gleisy Nordet**. *Estrategia de Gestión de Recursos Humanos para el proyecto del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas*. Ciudad de la Habana : s.n., 2010.
20. **Rosales., José Antonio León y Roblejo Rondón., Jesús Manuel** . *Sistema Informático de Gestión de Información de los Recursos Humanos y las Actividades de las Coordinaciones Regionales de Prevención del Delito*. Ciudad de la Habana : s.n., 2010.
21. **González., Gustavo Ponce y Lantigua Ramírez., Pedro Luis**. *Sistema de gestión de los recursos humanos (RRHH) del Departamento Productivo Señales Digitales de la facultad 9*. Ciudad de la Habana : s.n., 2010.
22. **Hernández, Angel Luis Morales**. *Sistema de Gestión de la información del conocimiento de los Recursos Humanos para los Polos Productivos de la facultad 9*. Ciudad de la Habana : s.n., 2010.
23. **Marichal., Sandy Soler y Torralbas Garcés., Anthony** . *Procedimiento para la gestión del desempeño de los Recursos Humanos del Centro de Desarrollo de Informática Industrial de la UCI basado en el Área Temática Motivación y Gestión del Desempeño de People CMM*. Ciudad de la Habana : s.n., 2010.
24. **Metodología XP**. Metodología XP. [En línea] [Citado el: 5 de 5 de 2013.] <https://sites.google.com/site/xpmetodologia/marco-teorico/funcionamiento>.
25. monografias.com. [En línea] [Citado el: 6 de enero de 2013.] <http://www.monografias.com/trabajos93/sistemas-presupuestarios/sistemas-presupuestarios.shtml>.
26. slideshare. [En línea] [Citado el: 16 de enero de 2013.] <http://www.slideshare.net/rodrigoluffy/esistemas-de-informacion-rodrigo>.
27. slideshare. [En línea] [Citado el: 2 de febrero de 2013.] <http://www.slideshare.net/edgarespinoza/programacion-extrema>.
28. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. s.l. : Addison Wesley, 2000. ISBN-84-7829-036-2.
29. **Figueroa, Roberth F. , Solís, Camilo J. y Cabrera, Armando A**. *Metodologías tradicionales vs metodologías ágiles*. s.l. : Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

30. **Letelier , Patricio y Penadés, M^a Carmen .** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España : Universidad Politécnica de Valencia.
31. **Calderón , Amaro y Valverde Rebaza, Sarah Dámaris.** *Metodologías Ágiles*. Trujillo-Perú : Universidad Nacional de Trujillo, 2007.
32. **Rumbaugh, James y Jacobson, Ivar.** *El lenguaje Unificado de Modelado, Manual de Referencia*. 2000. ISBN: 8478290370.
33. **Billy Reynoso, Carlos .** *Introducción a la Arquitectura de Software*. s.l. : UNIVERSIDAD DE BUENOS AIRES.
34. **Reynoso , Carlos y Kicillof, Nicolás .** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. s.l. : UNIVERSIDAD DE BUENOS AIRES, 2004.
35. **GAMMA, E., HELM, R. y JOHNSON, R. y VLISSIDES, J.** *Patrones de diseño*. 2000.
36. **Larman, Craig.** *UML y patrones*. 1999.
37. Didáctica y divulgación de la programación. [En línea] [Citado el: 14 de enero de 2013.] http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193.
38. Mis respuestas.com. [En línea] [Citado el: 14 de enero de 2013.] <http://www.misrespuestas.com/que-es-un-servidor-web.html>.
39. **Leyva, Armando Alexander Rosales y Márquez Figueroa, Miguel Angel.** *Análisis y Diseño del módulo Desarrollo del personal del Sistema de Gestión de Recursos Humanos por competencias en los proyectos productivos de la facultad 3*. Ciudad de la Habana : s.n., 2009.
40. **Arias, Henry González y Hernández Muñoz, Dayron .** *MaGISTral. Sistema Integral de Gestión de Recursos para el proyecto GENESIG*. Ciudad de la Habana : s.n., 2010.
41. **Vážan, Robert.** SourceAFIS. [En línea] <http://www.sourceafis.org/blog/datasheet/>.
42. **Figueroa, Roberth G. , Solís, Camilo J. y Cabrera , Armando A. .** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. Ecuador Loja : s.n.
43. **McClure, Carma .** *The CASE Experience*. s.l. : BYTE, Abril 1989.
44. Instituto Tecnológico de Sonora. [En línea] [Citado el: 7 de enero de 2013.] http://biblioteca.itson.mx/oa/dip_ago/introduccion_sistemas/p3.htm.
45. **Letelier, Patricio y Penadés, M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Valencia : s.n.

Glosario de Términos.

1. **HTTP:** Hipertext Transfer Protocol. [Protocolo de transferencia de hipertexto]. Conjunto de estándares que permite a los usuarios de la Web intercambiar información. Método que se utiliza para transferir documentos desde el sistema donde se almacenan las páginas hasta los usuarios individuales.
2. **IDE:** Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.
3. **UCI:** Universidad de las Ciencias Informáticas.
4. **CISED:** Centro de Identificación y Seguridad Digital.
5. **R.R.H.H.:** Recursos Humanos.

Anexos

Anexo 1. Entrevista.

1 ¿Diga cuáles son los principales procesos de su área?

1.1 ¿Cuáles son los conceptos que participan en los procesos?

1.1.2 ¿Cuáles son los tipos de interacciones que ocurren entre estos conceptos?

1.2 ¿Cuál es la información que se maneja de cada proceso?

2 ¿Diga cuáles son los informes comunes de su área?

2.1 ¿Cuáles son los listados de datos que se manejan en los informes de su área?

2.2 ¿Cuáles son los datos más importantes utilizados en esos informes?

2.3 ¿Nivel de conformidad de cada informe?

3 ¿Cuál es el impacto esperado de la solución?

Anexo 2. Historias de usuarios.

Número: HU_4	Usuario: Hiraldo Manuel Anaya González		
Nombre historia: Gestionar máquina.			
Prioridad en negocio: Alta		Riesgo en desarrollo: Medio	
Puntos estimados: 2		Iteración asignada: 1	
Programador responsable: Rocio Romero Alvarez			
Descripción: La presente historia de usuario tiene como objetivo gestionar una máquina, a la cual se le podrá hacer una serie de operaciones.			
Observaciones: Esta operación solo podrá ser realizada por la persona asignada para ello la cual podrá:			
Adicionar: Permitirá adicionar una máquina al sistema.			
Eliminar: Permitirá eliminar una máquina del sistema.			
Mostrar: Permitirá mostrar la información de una máquina existente en el sistema.			
Consultar: Permitirá buscar una máquina existente en el sistema y consultar su información.			

Requisitos no funcionales	Interfaz
Asuntos pendientes	Ninguno

Tabla 6 Historia de usuario 4. Gestionar máquina.

Número: HU_5	Usuario: Hiraldo Manuel Anaya González	
Nombre historia: Mostrar reporte.		
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto	
Puntos estimados: 3	Iteración asignada: 2	
Programador responsable: Yam Brayam Beltrán Rodríguez		
Descripción: La presente historia de usuario tiene como objetivo mostrar los reportes de los trabajadores, estudiantes y máquinas existentes en la entidad.		
Observaciones: Esta operación solo podrá ser realizada por las personas autorizadas para ello.		
Requisitos no funcionales	Interfaz	
Asuntos pendientes	Ninguno	

Tabla 7 Historia de usuario 5. Mostrar reporte.

Número: HU_6	Usuario: Hiraldo Manuel Anaya González	
Nombre historia: Gestionar estudiante.		
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio	
Puntos estimados: 2	Iteración asignada: 1	
Programador responsable: Rocio Romero Alvarez		
Descripción: La presente historia de usuario tiene como objetivo gestionar un estudiante, al		

se le podrá hacer una serie de cambios.	
Observaciones: Esta operación solo podrá ser realizada por personas autorizadas para:	
Adicionar: Permitirá adicionar un estudiante al sistema.	
Eliminar: Permitirá eliminar un estudiante del sistema.	
Mostrar: Permitirá mostrar la información de un estudiante existente en el sistema.	
Consultar: Permitirá buscar un estudiante existente en el sistema y consultar su información.	
Requisitos no funcionales	Interfaz
Asuntos pendientes	Ninguno

Tabla 8 Historia de usuario 6. Gestionar estudiante.

Número: HU_7	Usuario: Hiraldo Manuel Anaya González	
Nombre historia: Gestionar profesor.		
Prioridad en negocio: Alta		Riesgo en desarrollo: Alto
Puntos estimados: 2		Iteración asignada: 1
Programador responsable: Yam Brayam Beltrán Rodríguez		
Descripción: La presente historia de usuario tiene como objetivo gestionar un profesor.		
Observaciones: Esta operación solo podrá ser realizada por el personal calificado para:		
Adicionar: Permitirá adicionar un profesor al sistema.		
Eliminar: Permitirá eliminar un profesor del sistema.		
Mostrar: Permitirá mostrar la información de un profesor existente en el sistema.		
Consultar: Permitirá buscar un profesor existente en el sistema y consultar su información.		
Requisitos no funcionales	Interfaz	
Asuntos pendientes	Ninguno	

Tabla 9 Historia de usuario 7. Gestionar profesor.

Anexo 3. Diagramas de clases.

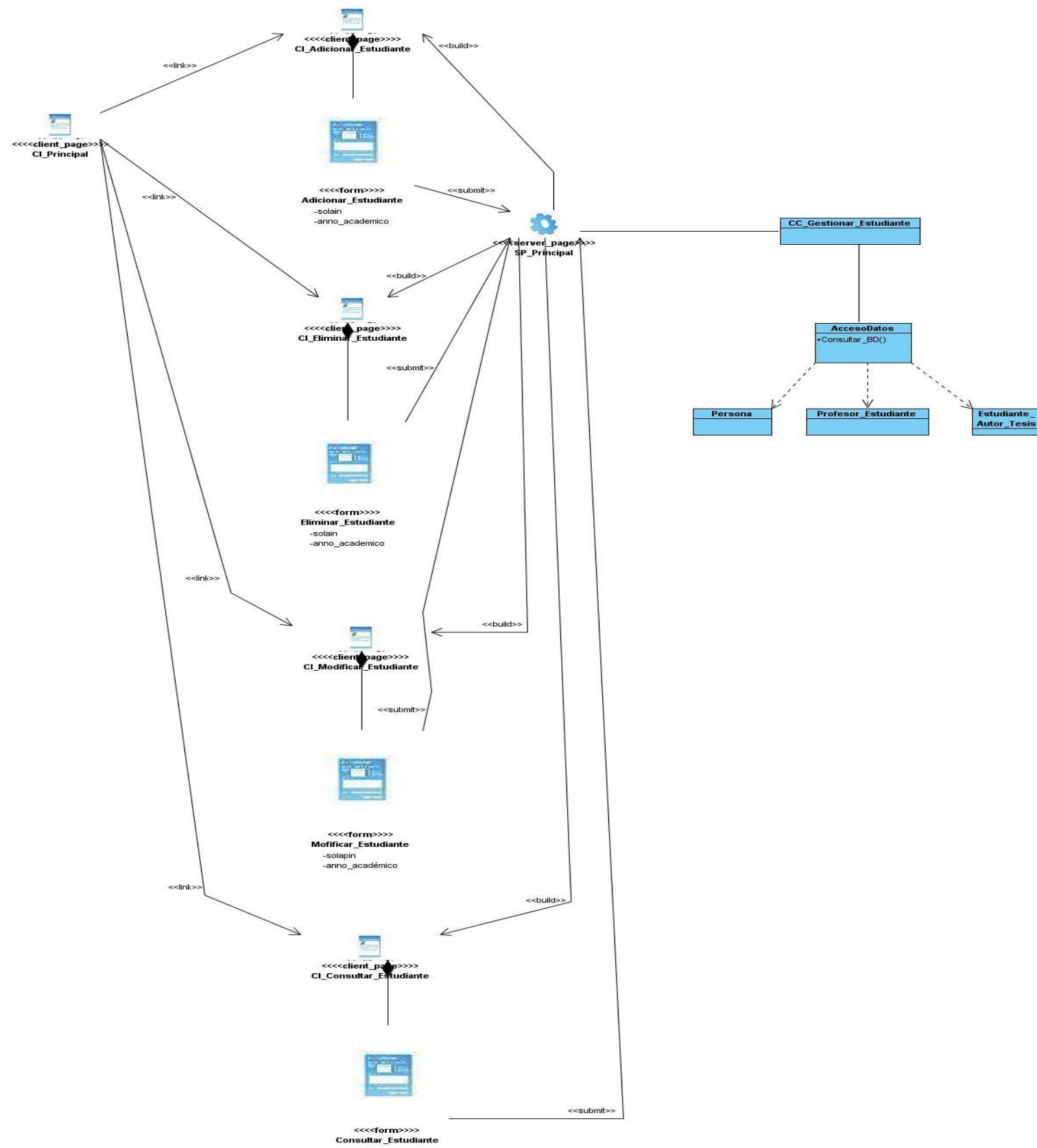


Figura 9 Diagrama de clases GestionarEstudiante

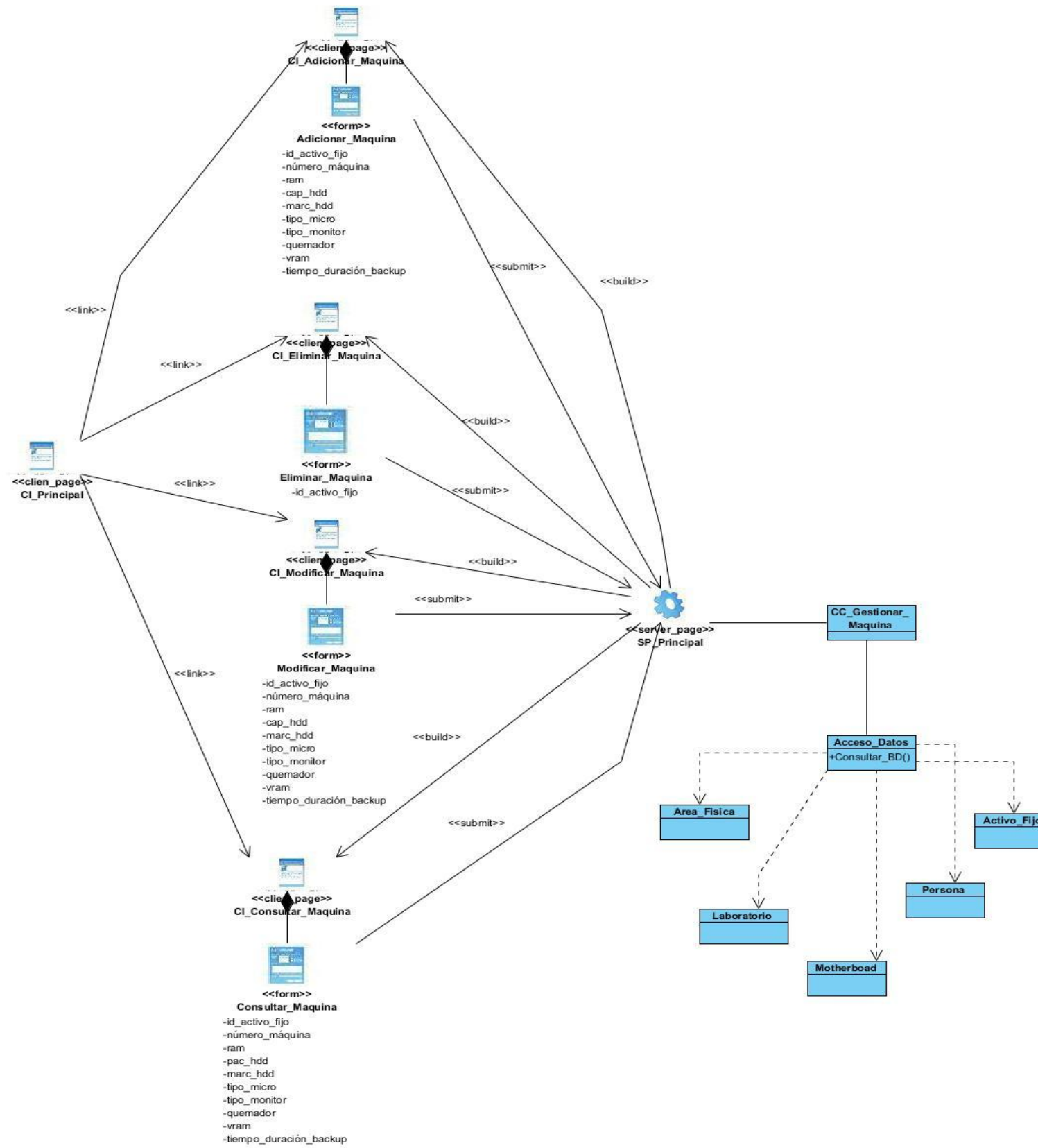


Figura 10 Diagrama de clases GestionarMaquina

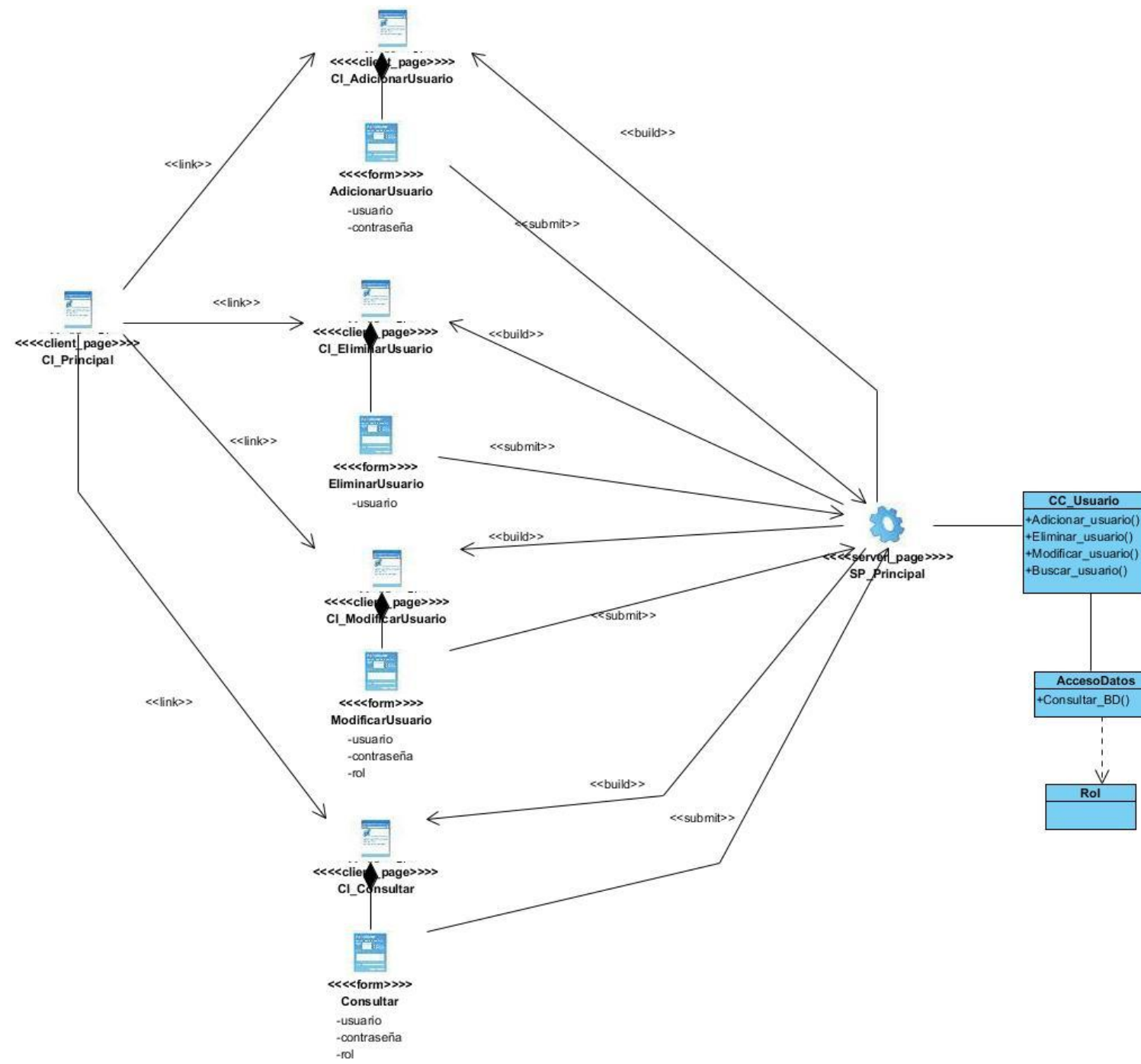


Figura 11 Diagrama de clases GestionarUsuario

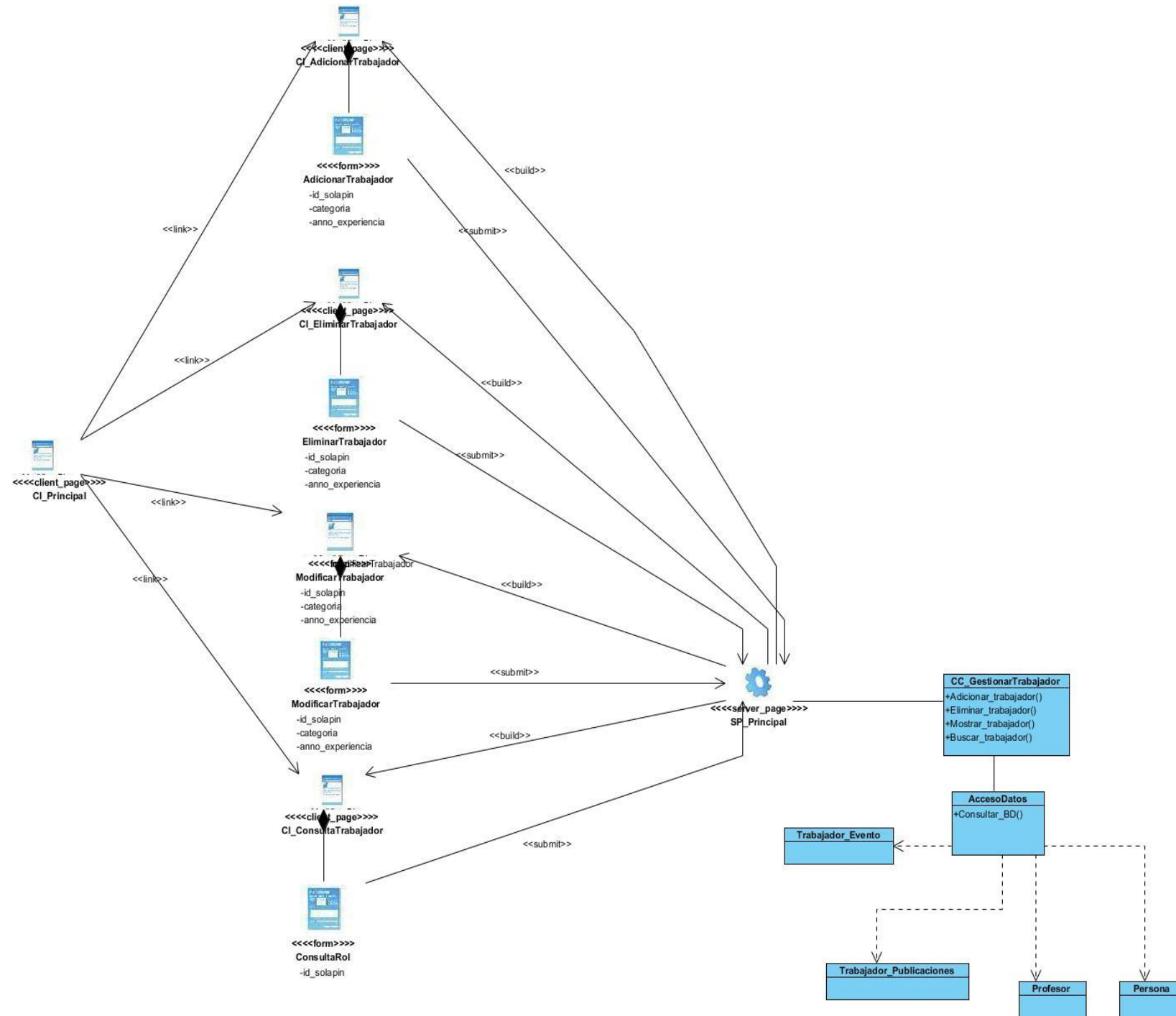


Figura 12 Diagrama de clases GestionarTrabajador

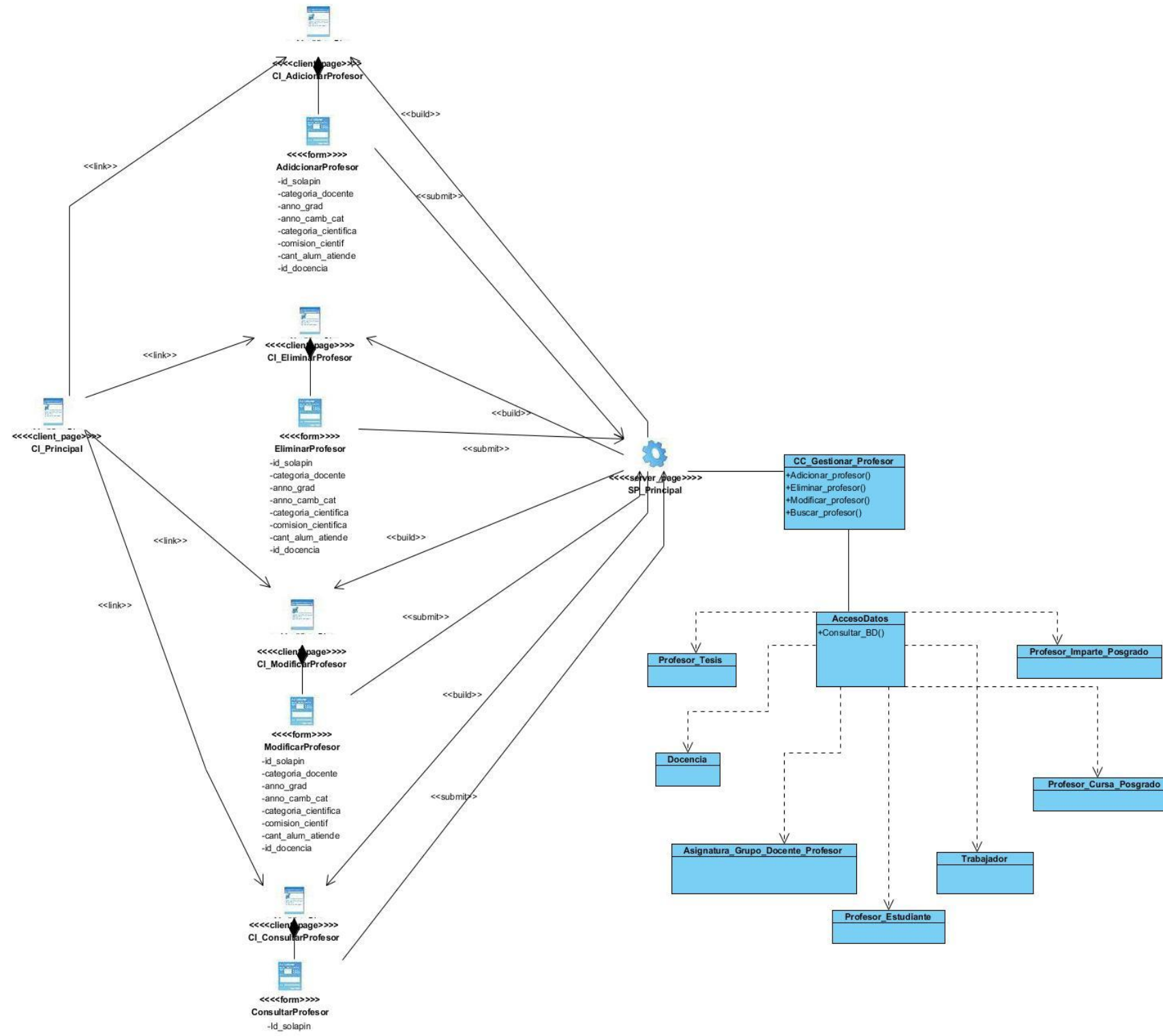


Figura 13 Diagrama de clases GestionarProfesor

Anexo 4. Modelo de dato.

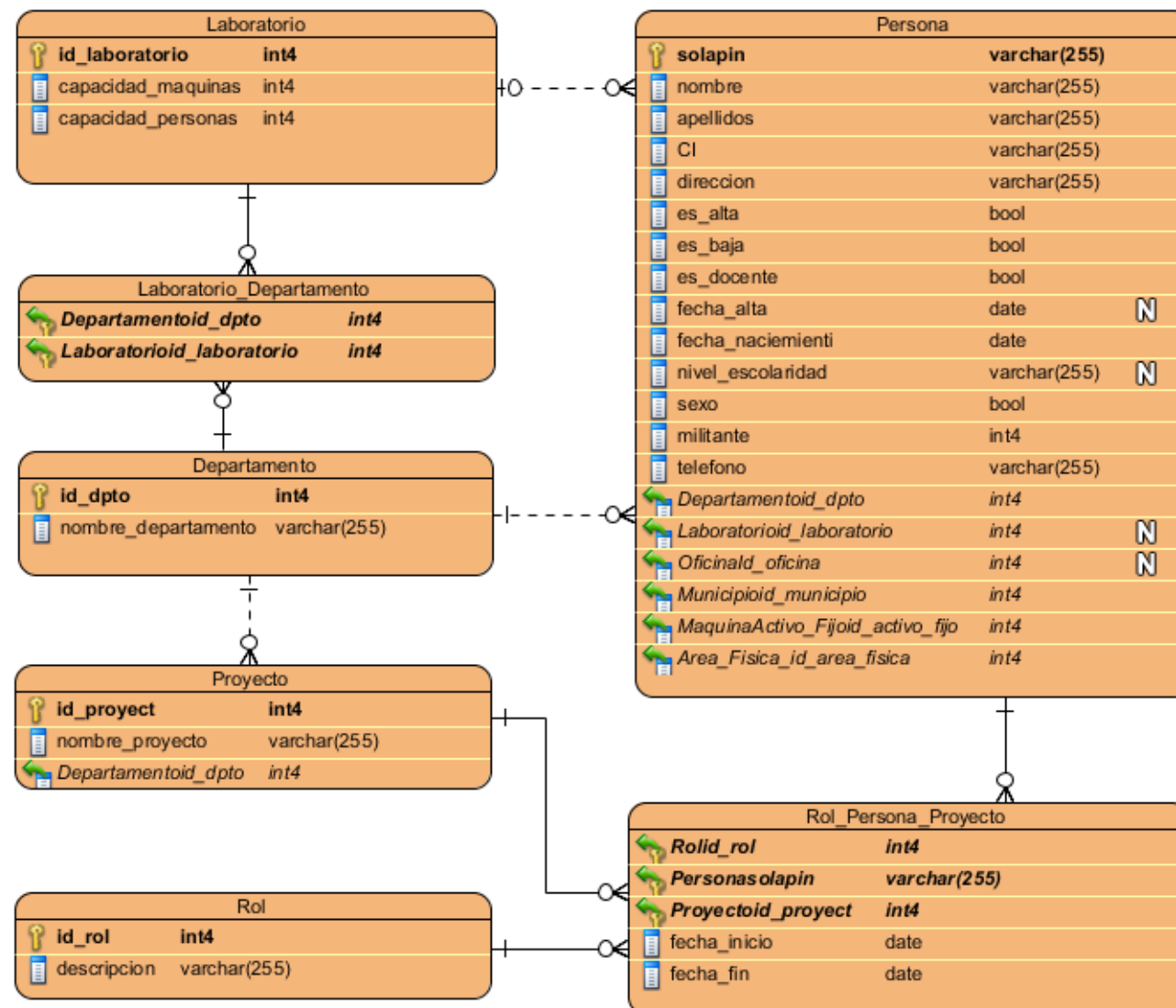


Figura 16 Modelo de dato parte 1

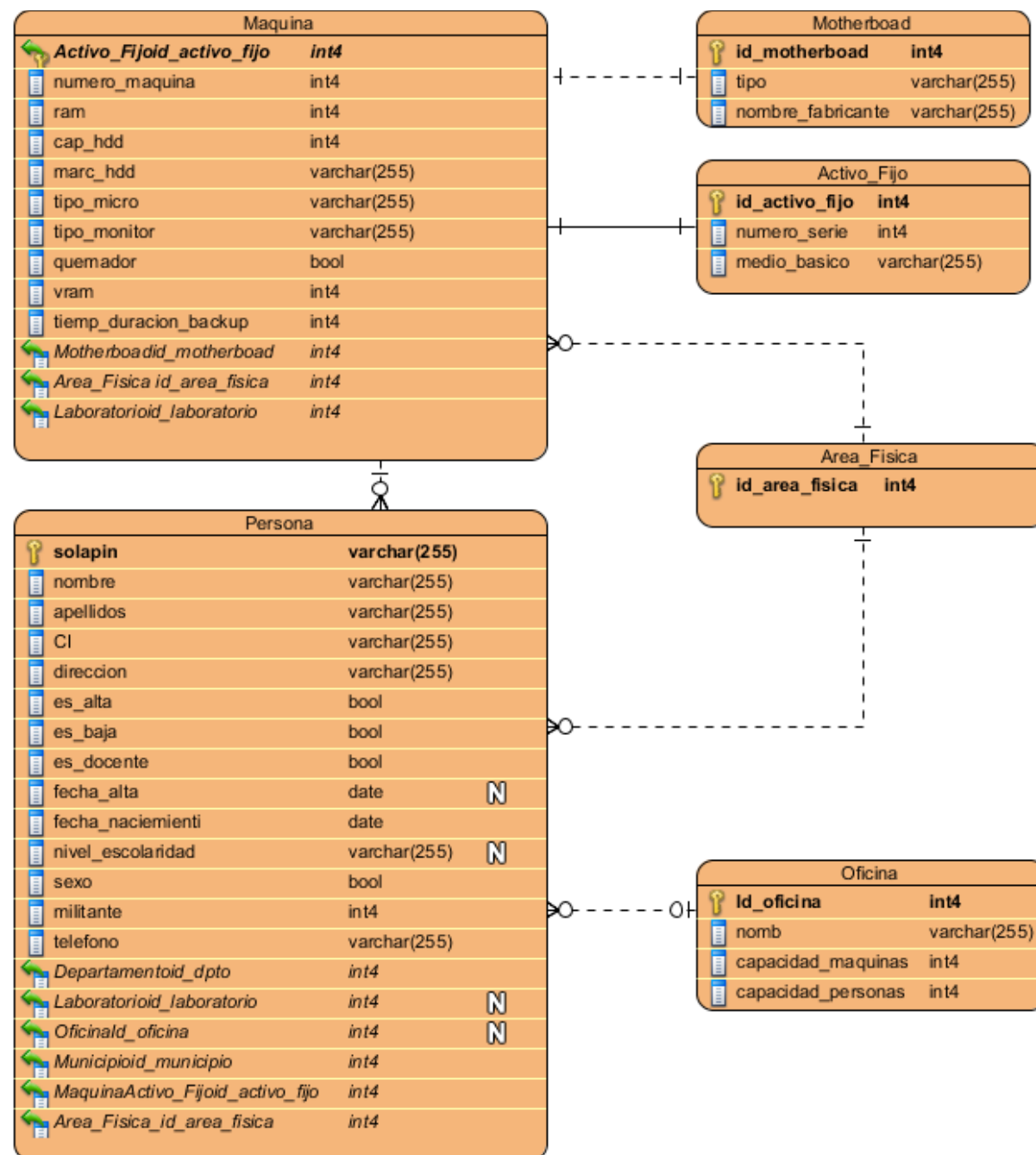


Figura 17 Modelo de dato parte 2



Figura 18 Modelo de dato parte 3

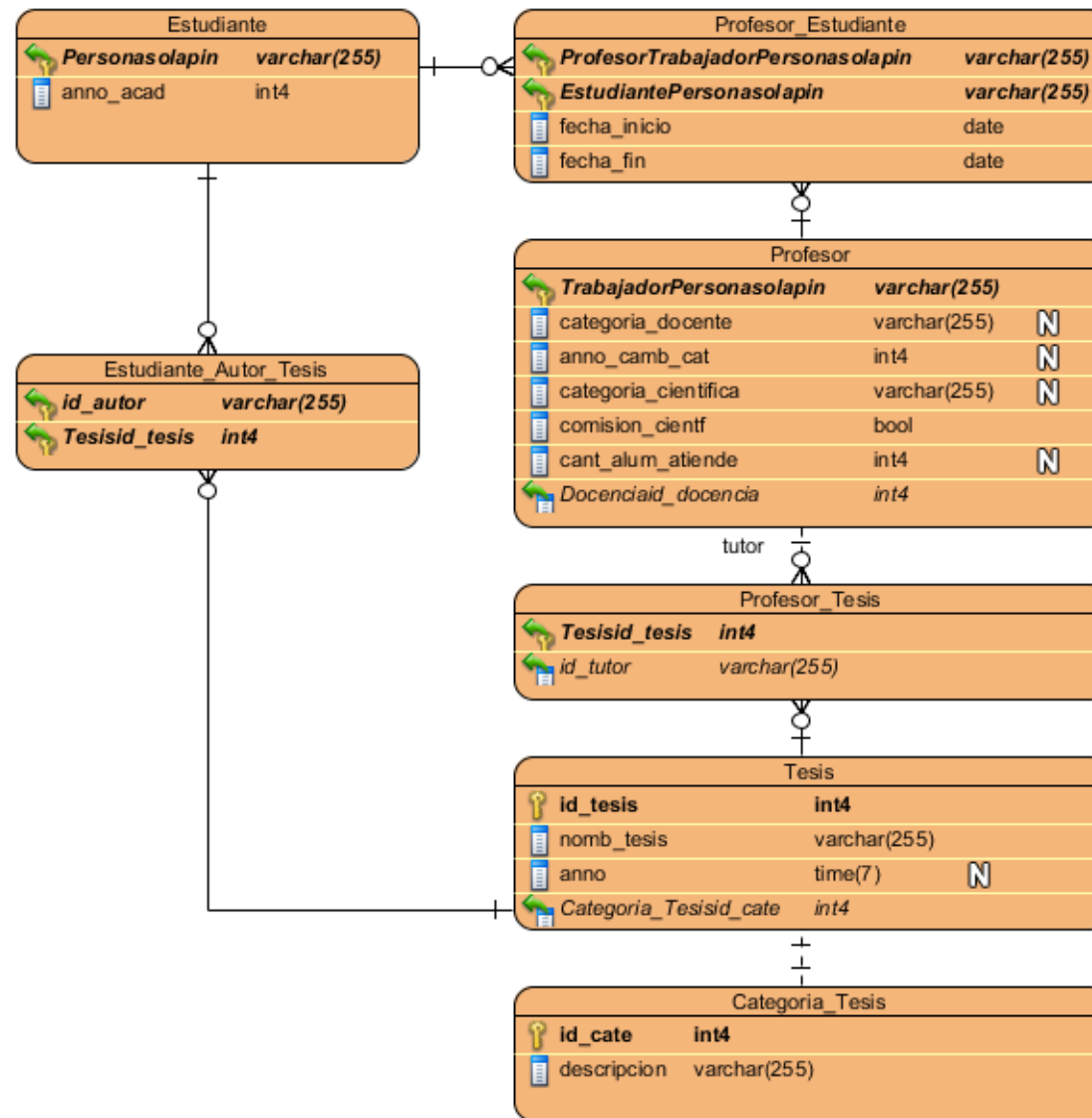


Figura 19 Modelo de dato parte 4

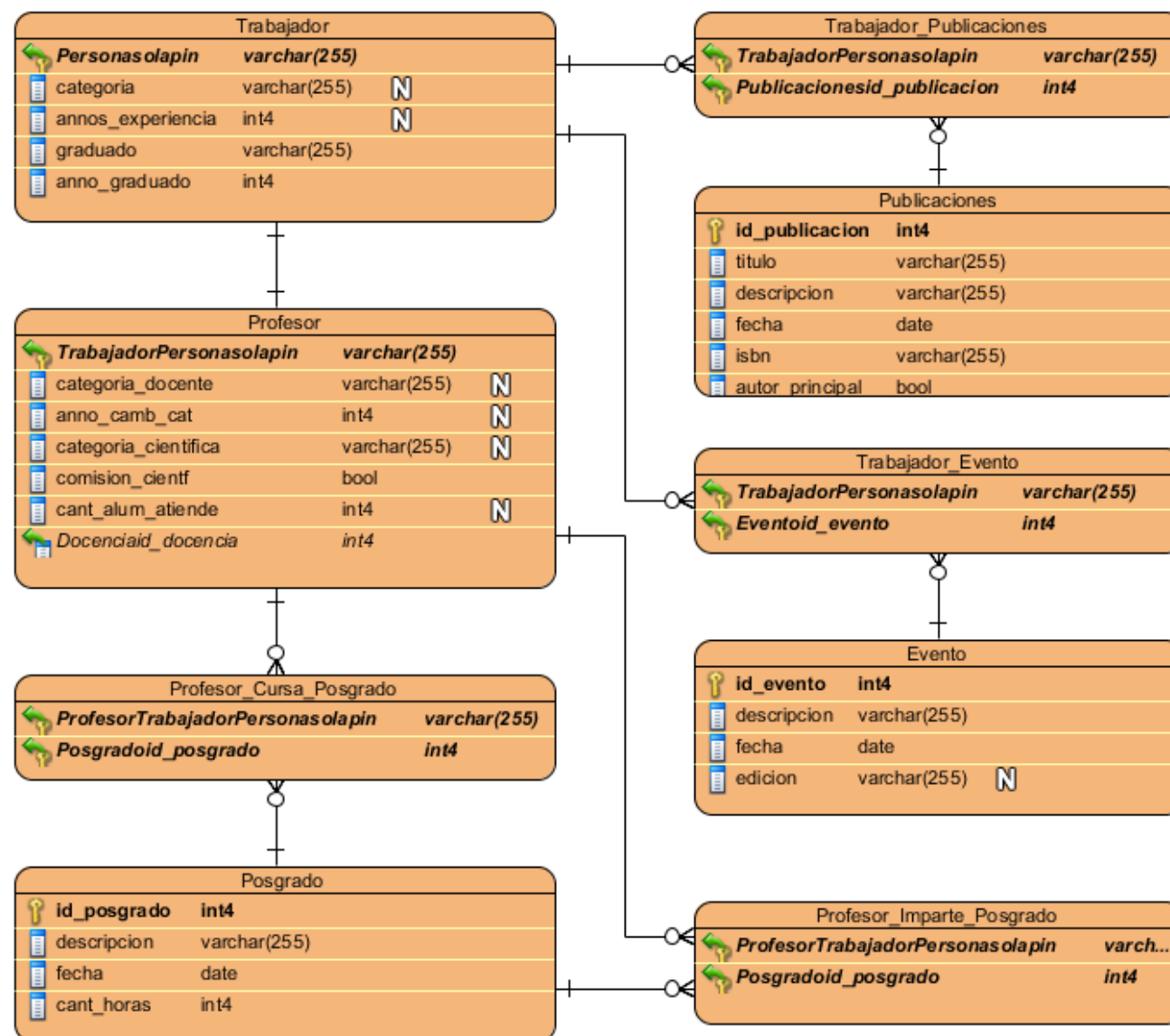


Figura 20 Modelo de dato parte 5

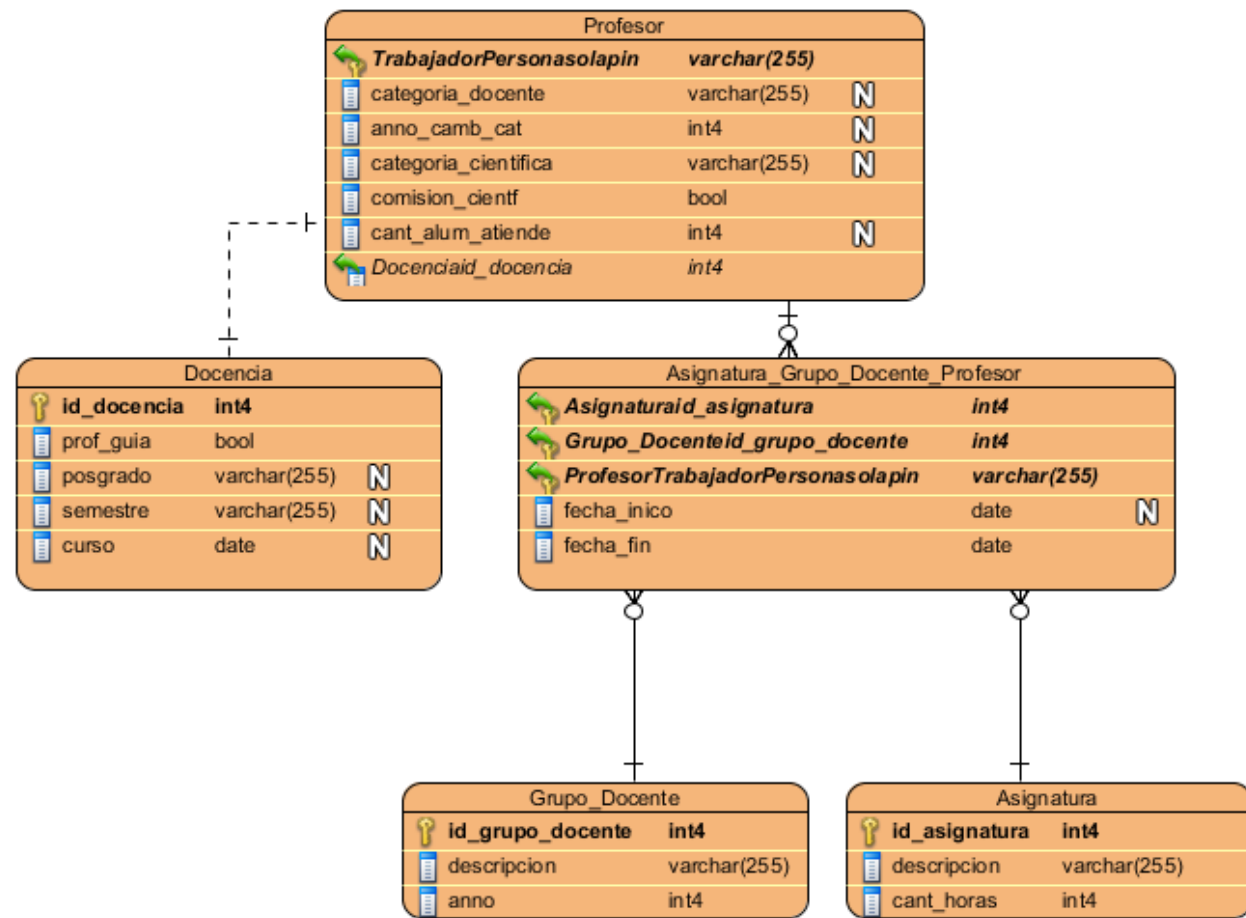


Figura 21 Modelo de dato parte 6

Anexo 5 Tarjetas CRC.

Clase: PersonaListView	
Responsabilidades	Colaboradores
Listar datos de las personas	<ul style="list-style-type: none"> • Persona

Tabla 15 Tarjeta CRC 2. PersonaListView

Clase: EstudianteListView	
Responsabilidades	Colaboradores
Listar datos de los estudiantes	<ul style="list-style-type: none"> • Estudiantes

Tabla 16 Tarjeta CRC 3. EstudianteListView

Clase: ProyectoProfesoresListView	
Responsabilidades	Colaboradores
Listar nombre y apellidos de los profesores y al proyecto al que pertenece.	<ul style="list-style-type: none"> • Profesor • RolPersonaProyecto

Tabla 17 Tarjeta CRC 4. ProyectoProfesoresListView

Clase: ListadoProfeDepartamentoListView	
Responsabilidades	Colaboradores
Listar nombre y apellidos de los profesores y al departamento al que pertenece.	<ul style="list-style-type: none"> • Profesor • Departamento

Tabla 18 Tarjeta CRC 5. ListadoProfeDepartamentoListView

Clase: ListadoProfeRolListView	
Responsabilidades	Colaboradores
Listar nombre y apellidos de los profesores y el rol que desempeña.	<ul style="list-style-type: none"> • Profesor • RolPersonaProyecto

Tabla 19 Tarjeta CRC 6. ListadoProfeRolListView

Clase: Maquina	
Responsabilidades	Colaboradores
Guardar la información referente a una máquina	<ul style="list-style-type: none"> • Área_Física • Persona • Laboratorio • Motherboard • Activo_Fijo

Tabla 20 Tarjeta CRC 9. Maquina

Clase: Estudiante	
Responsabilidades	Colaboradores
Guardar la información referente a un estudiante	<ul style="list-style-type: none"> • Estudiante_Autor_Tesis • Persona • Profesor_Estudiante

Tabla 21 Tarjeta CRC 10. Estudiante

Anexo 6. Tareas de ingeniería.

Tarea de ingeniería	
Número de tarea: HU4_T7	Historia de usuario: Gestionar estudiantes.
Nombre: Modificar estudiante.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite hacer cambios a los estudiantes en el sistema.	

Tabla 24 Tarea de ingeniería 3. Gestionar estudiantes.

Tarea de ingeniería	
Número de tarea: HU4_T8	Historia de usuario: Gestionar estudiantes.
Nombre: Eliminar estudiante.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite eliminar un estudiante.	

Tabla 25 Tarea de ingeniería 4. Gestionar estudiantes.

Tarea de ingeniería	
Número de tarea: HU4_T9	Historia de usuario: Gestionar estudiantes.
Nombre: Insertar estudiante.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	

Descripción: Permite insertar un nuevo estudiante.

Tabla 26 Tarea de ingeniería 5. Gestionar estudiantes.

Tarea de ingeniería	
Número de tarea: HU4_T10	Historia de usuario: Gestionar estudiantes.
Nombre: Mostrar estudiante.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Muestra un estudiante.	

Tabla 27 Tarea de ingeniería 6. Gestionar estudiantes.

Tarea de ingeniería	
Número de tarea: HU5_T11	Historia de usuario: Gestionar trabajador.
Nombre: Modificar trabajador.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite modificar un trabajador.	

Tabla 28 Tarea de ingeniería 7. Gestionar trabajador.

Tarea de ingeniería	
Número de tarea: HU5_T12	Historia de usuario: Gestionar trabajador.
Nombre: Eliminar trabajador.	
Tipo: Desarrollo.	Puntos Estimados: 1

Responsable: Rocio Romero Alvarez.
Descripción: Permite eliminar un trabajador.

Tabla 29 Tarea de ingeniería 8. Gestionar trabajador.

Tarea de ingeniería	
Número de tarea: HU5_T13	Historia de usuario: Gestionar trabajador.
Nombre: Insertar trabajador.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite insertar un trabajador.	

Tabla 30 Tarea de ingeniería 9. Gestionar trabajador.

Tarea de ingeniería	
Número de tarea: HU5_T14	Historia de usuario: Gestionar trabajador.
Nombre: Mostrar trabajador.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Muestra un trabajador.	

Tabla 31 Tarea de ingeniería 10. Gestionar trabajador.

Tarea de ingeniería	
Número de tarea: HU6_T15	Historia de usuario: Gestionar máquina.
Nombre: Modificar máquina.	

Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite modificar una máquina.	

Tabla 32 Tarea de ingeniería 11. Gestionar máquina.

Tarea de ingeniería	
Número de tarea: HU6_T16	Historia de usuario: Gestionar máquina.
Nombre: Eliminar máquina.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite eliminar una máquina.	

Tabla 33 Tarea de ingeniería 12. Gestionar máquina.

Tarea de ingeniería	
Número de tarea: HU6_T17	Historia de usuario: Gestionar máquina.
Nombre: Insertar máquina.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Permite insertar una máquina.	

Tabla 34 Tarea de ingeniería 13. Gestionar máquina.

Tarea de ingeniería	
Número de tarea: HU6_T18	Historia de usuario: Gestionar máquina.

Nombre: Mostrar máquina.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Rocio Romero Alvarez.	
Descripción: Muestra una máquina.	

Tabla 35 Tarea de ingeniería 14. Gestionar máquina.

Tarea de ingeniería	
Número de tarea: HU7_T19	Historia de usuario: Mostrar reporte.
Nombre: Muestra un reporte con la información solicitada.	
Tipo: Desarrollo.	Puntos Estimados: 2
Responsable: Yam Brayam Beltran Rodriguez.	
Descripción: Muestra un reporte de lo deseado por el usuario.	

Tabla 36 Tarea de ingeniería 15. Mostrar reporte.

Tarea de ingeniería	
Número de tarea: HU8_T20	Historia de usuario: Gestionar profesor.
Nombre: -Modificar profesor.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Yam Brayam Beltran Rodriguez.	
Descripción: Permite modificar un profesor.	

Tabla 37 Tarea de ingeniería 16. Gestionar profesor.

Tarea de ingeniería

Número de tarea: HU8_T21	Historia de usuario: Gestionar profesor.
Nombre: Eliminar profesor.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Yam Brayam Beltran Rodriguez.	
Descripción: Permite eliminar un profesor.	

Tabla 38 Tarea de ingeniería 17. Gestionar profesor.

Tarea de ingeniería	
Número de tarea: HU8_T22	Historia de usuario: Gestionar profesor.
Nombre: Insertar profesor.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Yam Brayam Beltran Rodriguez.	
Descripción: Permite insertar un profesor.	

Tabla 39 Tarea de ingeniería 18. Gestionar profesor.

Tarea de ingeniería	
Número de tarea: HU8_T23	Historia de usuario: Gestionar profesor.
Nombre: Mostrar profesor.	
Tipo: Desarrollo.	Puntos Estimados: 1
Responsable: Yam Brayam Beltran Rodriguez.	
Descripción: Muestra un profesor.	

Tabla 40 Tarea de ingeniería 19. Gestionar profesor.

Anexo 7. Tareas de implementación.

Tarea	
Número de la tarea: 8	Historia de usuario: Gestionar estudiantes.
Nombre de la tarea: Eliminar estudiante.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda eliminar los datos de un estudiante.	

Tabla 44 Tarea 4 de implementación. Gestionar estudiantes.

Tarea	
Número de la tarea: 9	Historia de usuario: Gestionar estudiantes.
Nombre de la tarea: Insertar estudiante.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda insertar los datos de un estudiante.	

Tabla 45 Tarea 5 de implementación. Gestionar estudiantes.

Tarea	
Número de la tarea: 10	Historia de usuario: Gestionar estudiantes.
Nombre de la tarea: Mostrar estudiante.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	

Descripción: El sistema muestra los datos de un estudiante.

Tabla 46 Tarea 6 de implementación. Gestionar estudiantes.

Tarea	
Número de la tarea: 11	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Modificar trabajador.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de un trabajador.	

Tabla 47 Tarea 7 de implementación. Gestionar trabajador.

Tarea	
Número de la tarea: 12	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Eliminar trabajador.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda eliminar los datos de un trabajador.	

Tabla 48 Tarea 8 de implementación. Gestionar trabajador.

Tarea	
Número de la tarea: 13	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Insertar trabajador.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2

Programador responsable: Rocio Romero Alvarez.
Descripción: El sistema muestra la opción para que el administrador pueda insertar los datos de un trabajador.

Tabla 49 Tarea 9 de implementación. Gestionar trabajador.

Tarea	
Número de la tarea: 14	Historia de usuario: Gestionar trabajador.
Nombre de la tarea: Mostrar trabajador.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra los datos de un trabajador.	

Tabla 50 Tarea 10 de implementación. Gestionar trabajador.

Tarea	
Número de la tarea: 15	Historia de usuario: Gestionar máquina.
Nombre de la tarea: Modificar máquina.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de una máquina.	

Tabla 51 Tarea 11 de implementación. Gestionar máquina.

Tarea	
Número de la tarea: 16	Historia de usuario: Gestionar máquina.
Nombre de la tarea: Eliminar máquina.	

Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda eliminar los datos de una máquina.	

Tabla 52 Tarea 12 de implementación. Gestionar máquina.

Tarea	
Número de la tarea: 17	Historia de usuario: Gestionar máquina.
Nombre de la tarea: Insertar máquina.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra la opción para que el administrador pueda insertar los datos de una máquina.	

Tabla 53 Tarea 13 de implementación. Gestionar máquina.

Tarea	
Número de la tarea: 18	Historia de usuario: Gestionar máquina.
Nombre de la tarea: Mostrar máquina.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Rocio Romero Alvarez.	
Descripción: El sistema muestra los datos de una máquina.	

Tabla 54 Tarea 14 de implementación. Gestionar máquina.

Tarea	
Número de la tarea: 19	Historia de usuario: Mostrar reporte.

Nombre de la tarea: Muestra un reporte con la información solicitada.	
Tipo de tarea: Desarrollo.	Puntos estimados: 3
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra un reporte con la petición deseada.	

Tabla 55 Tarea 15 de implementación. Mostrar reporte.

Tarea	
Número de la tarea: 20	Historia de usuario: Gestionar profesor.
Nombre de la tarea: Modificar profesor.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra la opción para que el administrador pueda modificar los datos de un profesor.	

Tabla 56 Tarea 16 de implementación. Gestionar profesor.

Tarea	
Número de la tarea: 21	Historia de usuario: Gestionar profesor.
Nombre de la tarea: Eliminar profesor.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra la opción para que el administrador pueda eliminar los datos de un profesor.	

Tabla 57 Tarea 17 de implementación. Gestionar profesor.

Tarea

Número de la tarea: 22	Historia de usuario: Gestionar profesor.
Nombre de la tarea: Insertar profesor.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra la opción para que el administrador pueda insertar los datos de un profesor.	

Tabla 58 Tarea 18 de implementación. Gestionar profesor.

Tarea	
Número de la tarea: 23	Historia de usuario: Gestionar profesor.
Nombre de la tarea: Mostrar profesor.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Programador responsable: Yam Brayam Beltran Rodriguez.	
Descripción: El sistema muestra los datos de un profesor.	

Tabla 59 Tarea 19 de implementación. Gestionar profesor.

Anexo 8. Interfaces del sistema.

The screenshot displays the main interface of the 'Sistema de Gestión de Información de Recursos Humanos y Tecnológicos'. At the top, a blue banner contains the system's title. Below this is a navigation bar with buttons for 'Inicio', 'Reportes', 'Administración', and 'Cerrar Sesión'. A left-hand sidebar is titled 'Reportes' and lists several menu items with right-pointing chevrons: 'Datos de las personas', 'Listado de estudiantes', 'Listado de trabajadores', 'Tesis tuoreada por un profesor', 'Características de la pc', 'Año graduado del profesor', 'Departamento del profesor', 'Proyecto del profesor', 'Trabajador y su categoría', and 'PC del estudiante'. The main content area shows a search form with two input fields: 'Nombre:' containing 'Aliana' and 'Apellidos:' containing 'Dias Perez'. To the right of these fields is a blue 'Aceptar' button. Below the search fields is a dropdown menu for 'Nombre del departamento', which is currently expanded to show the option 'Targetas Inteligentes'.

Figura 31 Interfaz Mostrar reporte departamento del profesor.

Sistema de Gestión de Información de Recursos Humanos y Tecnológicos

Inicio

Reportes

Administración

Cerrar Sesión

Reportes

- › Datos de las personas
- › Listado de estudiantes
- › Listado de trabajadores
- › Tesis turoreada por un profesor
- › Características de la pc
- › Año graduado del profesor
- › Departamento del profesor
- › Proyecto del profesor
- › Trabajador y su categoría
- › PC del estudiante

Nombre:

Apellidos:

Aceptar

El profesor no existe

Nombre del departamento

Figura 32 Interfaz Mostrar reporte Departamento del profesor.

Anexo 9. Casos de pruebas de aceptación.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar trabajadores.
Nombre: Eliminar trabajadores.
Descripción: Prueba para comprobar si se ha eliminado uno o varios trabajadores en el sistema.
Condiciones de Ejecución: Para eliminar un trabajador se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la opción trabajador. • Seleccionar un trabajador. • Seleccionar la opción eliminar trabajador.
Resultado Esperado: Eliminado/s (número de eliminados) trabajador/s satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 63 Caso de Prueba de Aceptación 4. Gestionar trabajadores.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar trabajadores.
Nombre: Modificar trabajadores.
Descripción: Prueba para comprobar si se ha modificado un trabajador en el sistema.
Condiciones de Ejecución: Para modificar un trabajador se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla trabajador. • Seleccionar un trabajador. • Modificar los campos deseados. • Seleccionar la opción guardar.
Resultado Esperado: Se modificó con éxito el trabajador “nombre del trabajador modificado”.

Evaluación de la Prueba: Satisfactoria.

Tabla 64 Caso de Prueba de Aceptación 5. Gestionar trabajadores.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar trabajadores.
Nombre: Consultar trabajadores.
Descripción: Prueba para comprobar si se ha consultado un trabajador en el sistema.
Condiciones de Ejecución: Para consultar un trabajador se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar los reportes relacionados con trabajador.
Resultado Esperado: Consulta satisfactoria.
Evaluación de la Prueba: Satisfactoria.

Tabla 65 Caso de Prueba de Aceptación 6. Gestionar trabajadores.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar máquina.
Nombre: Adicionar máquina.
Descripción: Prueba para comprobar si se ha adicionado una máquina en el sistema.
Condiciones de Ejecución: Para adicionar una máquina se debe estar autenticado en el sistema como director o como subdirector de tecnología.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla máquina. • Seleccionar añadir máquina. • Llenar los campos del formulario. • Seleccionar la opción guardar.
Resultado Esperado: Se añadió con éxito la máquina "número de la máquina + número del laboratorio".
Evaluación de la Prueba: Satisfactoria.

Tabla 66 Caso de Prueba de Aceptación 7. Gestionar máquina.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar máquina.
Nombre: Eliminar máquina.
Descripción: Prueba para comprobar si se ha eliminado una máquina en el sistema.
Condiciones de Ejecución: Para eliminar una o varias máquinas se debe estar autenticado en el sistema como director o como subdirector de tecnología.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla máquina. • Seleccionar una máquina. • Seleccionar la opción eliminar máquina.
Resultado Esperado: Eliminado/s (número de eliminadas) máquina/s satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 67 Caso de Prueba de Aceptación 8. Gestionar máquina.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar máquina.
Nombre: Modificar máquina.
Descripción: Prueba para comprobar si se ha modificado una máquina en el sistema.
Condiciones de Ejecución: Para modificar una máquina se debe estar autenticado en el sistema como director o como subdirector de tecnología.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla máquina. • Seleccionar una máquina. • Modificar los campos deseados. • Seleccionar la opción guardar.
Resultado Esperado: Se modificó con éxito la máquina "número de la máquina + número del laboratorio".
Evaluación de la Prueba: Satisfactoria.

Tabla 68 Caso de Prueba de Aceptación 9. Gestionar máquina.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar máquina.
Nombre: Consultar máquina.
Descripción: Prueba para comprobar si se ha consultado una máquina en el sistema.
Condiciones de Ejecución: Para consultar una máquina se debe estar autenticado en el sistema como director o como subdirector de tecnología.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar los reportes relacionados con máquina.
Resultado Esperado: Consulta satisfactoria.
Evaluación de la Prueba: Satisfactoria.

Tabla 69 Caso de Prueba de Aceptación 10. Gestionar máquina.

Caso de Prueba de Aceptación.
Historia de usuario: Mostrar reporte.
Nombre: Mostrar reporte.
Descripción: Prueba para comprobar.
Condiciones de Ejecución: Para mostrar un reporte se debe estar autenticado en el sistema como director, o como subdirector de tecnología, o como subdirector docente, o como subdirector de investigación y posgrado o como invitado.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar reporte.
Resultado Esperado: Resultado de la consulta, o sea, los datos resultantes de la base de datos.
Evaluación de la Prueba: Satisfactoria.

Tabla 70 Caso de Prueba de Aceptación 11. Mostrar reporte.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar estudiante.
Nombre: Adicionar estudiante.
Descripción: Prueba para comprobar si se ha adicionado un estudiante en el sistema.

Condiciones de Ejecución: Para adicionar un estudiante se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla estudiante. • Seleccionar la opción añadir estudiante. • Llenar los campos del formulario. • Seleccionar la opción guardar.
Resultado Esperado: Se añadió con éxito el estudiante "nombre del estudiante".
Evaluación de la Prueba: Satisfactoria.

Tabla 71 Caso de Prueba de Aceptación 12. Gestionar estudiante.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar estudiante.
Nombre: Eliminar estudiante.
Descripción: Prueba para comprobar si se ha eliminado un estudiante del sistema.
Condiciones de Ejecución: Para eliminar un estudiante se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla estudiante. • Seleccionar un estudiante. • Seleccionar la opción eliminar estudiante.
Resultado Esperado: Eliminado/s (número de estudiantes) estudiante/s satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 72 Caso de Prueba de Aceptación 13. Gestionar estudiante.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar estudiante.
Nombre: Modificar estudiante.
Descripción: Prueba para comprobar si se ha modificado un estudiante en el sistema.

Condiciones de Ejecución: Para modificar un estudiante se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla estudiante. • Seleccionar un estudiante. • Modificar los campos deseados. • Seleccionar la opción guardar.
Resultado Esperado: Se modificó con éxito el estudiante "nombre del estudiante".
Evaluación de la Prueba: Satisfactoria.

Tabla 73 Caso de Prueba de Aceptación 14. Gestionar estudiante.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar estudiante.
Nombre: Consultar estudiante.
Descripción: Prueba para comprobar si se ha consultado un estudiante en el sistema.
Condiciones de Ejecución: Para consultar un estudiante se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar los reportes relacionados con estudiante.
Resultado Esperado: Consulta satisfactoria.
Evaluación de la Prueba: Satisfactoria.

Tabla 74 Caso de Prueba de Aceptación 15. Gestionar estudiante.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar profesor.
Nombre: Adicionar profesor.
Descripción: Prueba para comprobar si se ha adicionado un profesor en el sistema.
Condiciones de Ejecución: Para adicionar un profesor se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución:

<ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla profesor. • Seleccionar la opción añadir profesor. • Llenar los campos del formulario. • Seleccionar la opción guardar.
Resultado Esperado: Se añadió con éxito el profesor "nombre del profesor".
Evaluación de la Prueba: Satisfactoria.

Tabla 75 Caso de Prueba de Aceptación 16. Gestionar profesor.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar profesor.
Nombre: Eliminar profesor.
Descripción: Prueba para comprobar si se ha eliminado un profesor en el sistema.
Condiciones de Ejecución: Para eliminar un profesor se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla profesor. • Seleccionar un profesor. • Seleccionar la opción eliminar profesor.
Resultado Esperado: Eliminado/s (número de profesores) profesor/s satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 76 Caso de Prueba de Aceptación 17. Gestionar profesor.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar profesor.
Nombre: Modificar profesor.
Descripción: Prueba para comprobar si se ha modificado un profesor en el sistema.
Condiciones de Ejecución: Para modificar un profesor se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución:

<ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla profesor. • Seleccionar un profesor. • Modificar los campos deseados. • Seleccionar la opción guardar.
Resultado Esperado: Se modificó con éxito el profesor "nombre del profesor".
Evaluación de la Prueba: Satisfactoria.

Tabla 77 Caso de Prueba de Aceptación 18. Gestionar profesor.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar profesor.
Nombre: Consultar profesor.
Descripción: Prueba para comprobar si se ha consultado un profesor en el sistema.
Condiciones de Ejecución: Para consultar un profesor se debe estar autenticado en el sistema como director o como subdirector docente.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Seleccionar los reportes relacionados con profesor.
Resultado Esperado: Consulta satisfactoria.
Evaluación de la Prueba: Satisfactoria.

Tabla 78 Caso de Prueba de Aceptación 19. Gestionar profesor.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar usuario.
Nombre: Eliminar usuario.
Descripción: Prueba para comprobar si se ha eliminado un usuario del sistema.
Condiciones de Ejecución: Para eliminar un usuario se debe estar autenticado en el sistema como administrador.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla usuario. • Seleccionar un usuario. • Seleccionar la opción eliminar usuario.

Resultado Esperado: Eliminado/s (número de usuarios a eliminar) usuario/s satisfactoriamente.
Evaluación de la Prueba: Satisfactoria.

Tabla 79 Caso de Prueba de Aceptación 20. Gestionar usuario.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar usuario.
Nombre: Modificar usuario.
Descripción: Prueba para comprobar si se ha modificado un usuario en el sistema.
Condiciones de Ejecución: Para modificar un usuario se debe estar autenticado en el sistema como administrador.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla usuario. • Seleccionar un usuario. • Modificar los campos deseados. • Seleccionar la opción guardar.
Resultado Esperado: Se modificó con éxito el usuario "nombre del usuario modificado".
Evaluación de la Prueba: Satisfactoria.

Tabla 80 Caso de Prueba de Aceptación 21. Gestionar usuario.

Caso de Prueba de Aceptación.
Historia de usuario: Gestionar usuario.
Nombre: Consultar usuario.
Descripción: Prueba para comprobar si se ha consultado un usuario en el sistema.
Condiciones de Ejecución: Para consultar un usuario se debe estar autenticado en el sistema como administrador.
Entrada / Pasos de ejecución: <ul style="list-style-type: none"> • Entrar a la administración del sitio. • Seleccionar la tabla usuario. • Seleccionar consultar usuario.
Resultado Esperado: Consulta satisfactoria.

Evaluación de la Prueba: Satisfactoria.

Tabla 81 Caso de Prueba de Aceptación 22. Gestionar usuario.