

Universidad de las Ciencias Informáticas

Facultad 3



**“Desarrollo del componente Capacitación y
Desarrollo del subsistema Capital humano del
Sistema Integral de Gestión Cedrux.”**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autora: Yudisleydi González Tirado.

Tutor(es): Ing. Yunet Suárez Abrante.

Co-tutor: Ing. Analina Benítez Guzmán.

La Habana

Junio 2013

A close-up photograph of Fidel Castro Ruz, wearing a green military cap with a gold star on the front. He is looking slightly to the left of the camera with a serious expression. The background is blurred, showing a red flag on the left and a blue sky on the right.

“Capital humano implica no sólo conocimientos, sino también y muy esencialmente conciencia, ética, solidaridad, sentimientos verdaderamente humanos, espíritu de sacrificio, heroísmo, y la capacidad de hacer mucho con muy poco.”

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yudisleydi González Tirado

Autor

Ing. Yunet Suárez Abrante

Tutor

Ing. Analina Benítez Guzmán

Co-tutor

DATOS DE CONTACTO

Datos de los tutores

Síntesis del tutor: **Ing. Yunet Suarez Abrante**: Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2008. Actualmente se desempeña como analista del proyecto Capital Humano del Departamento de Aduana, CEIGE, Facultad 3.

Síntesis del Co-tutor: **Ing. Analina Benítez Guzmán**: Ingeniero en Ciencias Informáticas. Graduado en la Universidad de las Ciencias Informáticas en el 2012. Se desempeñó como arquitecto del proyecto Capital Humano del Departamento de Aduana del Centro CEIGE. Actualmente se desempeña como Analista del proyecto Fiscalía, CEGEL, Facultad 3.

Primero que nada agradecerle a mi familia por todo el apoyo y cariño que me han brindado, en especial a mis padres por darme la oportunidad de ser una mejor persona y enseñarme a luchar por las cosas que quiero, a mi hermana por estar siempre cuando más la necesite y apoyarme en los momentos más difíciles. A mis tías, tíos y primos por todo lo que me han enseñado a lo largo de estos años. A Raquel y Oneysi que fueron mis primeras amigas en esta universidad y juntas pasamos muchas cosas buenas y malas, y más que amigas son como dos hermanas para mí. A mis tutoras Yunet y Analina por la ayuda en la realización de este trabajo y estar cuando más las necesitaba sin importar la hora.

A mis amigas Yuri, Lily y mi negrita Yiset por aguantarme todas mis malcriadeces y estar presentes en los momentos que más me hacía falta una mano amiga. Gracias por sus consejos y regaños cuando hacían falta. A mis amistades Aliuska, Alfredo, Yandy, Jiubel, Pompa, Rofy, Carlos por los buenos momentos que pasamos juntos desde primer año. A mis compañeros de aula desde 3er año, los cuales han sido más que compañeros amigos en estos últimos años en la universidad, siempre voy a recordar los buenos momentos, las fiestas que compartimos juntos y la ayuda de todos cuando más la necesite. Agradecerle también a mis compañeros del proyecto por todo lo que me enseñaron. Gracias a Alejandro por toda su ayuda y estar cuando más lo necesitaba. A Luis por los momentos lindos que pasamos juntos y los consejos, gracias por estar presente en los momentos más difíciles y soportar mis malcriadeces.

A mi mamá por todo el amor que me ha brindado en estos años y ser mi guía.

A mi papá por todo el apoyo y amor que me ha dado.

A mi hermana por ser un ejemplo para mí y apoyarme siempre en mis decisiones

*A mi abuelito Adolfo por el amor que me brindo desde pequeña y aunque ya no este entre nosotros sé
que desde el cielo está orgulloso de mi por este logro.*

RESUMEN

En la actualidad el uso de las tecnologías en la automatización de los procesos en las empresas o entidades presupuestadas es muy demandado por la gran productividad que este provoca, haciendo más fácil el manejo de la información de las empresas.

El proceso de Capacitación y desarrollo del personal en las entidades empresariales cubanas se efectúa en hojas de cálculo o procesadores de texto. Esto provoca demora en los servicios, así como pérdida de información dado por la manipulación de los datos por una o varias personas, lo que crea además un escenario propicio a ilegalidades dificultando al mismo tiempo el control de esta información, influyendo negativamente en la eficiencia y rapidez de la toma de decisiones en cuanto a la preparación y formación del personal.

En el presente trabajo se propone desarrollar el componente Capacitación y desarrollo del subsistema Capital Humano del sistema integral de gestión Cedrux, como parte del proceso de informatización de los procesos empresariales en las entidades laborales en Cuba. Para el desarrollo del componente se hace uso del lenguaje de programación del lado del servidor PHP en su versión 5.2.6 y del lado del cliente Java Script empleándose ambos mediante el IDE Netbeans en su versión 7.0.1. La construcción y documentación de la solución se realizó basándose en el Modelo de Desarrollo del centro CEIGE¹, obteniendo como resultado una solución capaz de gestionar la información de manera correcta para el proceso de Capacitación y desarrollo con las particularidades de la economía cubana.

Palabras claves: Capital humano, Cedrux, Gestión de Capital humano, Capacitación y desarrollo.

¹ CEIGE: Centro de Informatización de la Gestión de Entidades.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN 1

1. FUNDAMENTACIÓN TEÓRICA 5

 1.1 Introducción.....5

 1.2 Principales Conceptos5

1.2.1 Concepto de capital humano.....5

 1.3 Sistemas que incluyen el proceso Capacitación y Desarrollo.6

 1.3.1 SAP ERP6

 1.3.2 ERP GNU Enterprise8

 1.3.3 Sage ABRA HRMS9

 1.3.4 EXACT.....10

 1.4 Modelo de desarrollo de software.....11

 1.5 Tecnologías y herramientas para el desarrollo13

 1.6 Marco de trabajo13

 1.7 Herramienta para el modelado.....14

 1.8 Herramienta para el desarrollo colaborativo14

 1.9 Lenguajes de programación.....15

 1.10 Notaciones para el modelado.....15

 1.11 Gestor de Base de Datos16

 1.12 Servidor Web16

 1.13 Navegador Web16

 1.14 Conclusiones parciales17

2. PROPUESTA DE SOLUCIÓN 18

 2.1 Introducción.....18

 2.2 Descripción del negocio18

 2.3 Modelado del negocio20

 2.3.1 Mapa de procesos20

 2.4 Procesos de Negocio21

 2.5 Modelo conceptual22

 2.6 Técnicas y métodos empleados en la captura de requisitos23

2.6.1 Patrones de reutilización de requisitos24

 2.7 Especificación de los Requisitos del Sistema.....25

 2.7.1 Requisitos Funcionales25

 2.8 Requisitos no funcionales.....28

 2.9 Validación de los requisitos29

2.9.1 Criterios para la verificación de los requisitos.....	30
2.10 Conclusiones parciales.....	31
3. ANÁLISIS Y DISEÑO.....	33
3.1 Introducción.....	33
3.2 Patrones de diseño.....	33
3.2.1. Patrones GOF.....	33
3.2.2. Patrones GRASP.....	34
3.3 Patrón arquitectónico.....	35
3.4 Diagramas de clases del diseño.....	37
3.5 Diagrama de secuencia.....	40
3.6 Diagrama de entidad relación.....	40
3.7 Diagrama de componentes.....	41
3.8 Métricas para la validación del diseño propuesto.....	42
3.8.1. <i>Tamaño operacional de clases</i>	43
3.8.2. <i>Métrica Relaciones entre Clases</i>	45
3.9 Conclusiones parciales.....	48
4. IMPLEMENTACIÓN Y PRUEBA.....	49
4.1. Introducción.....	49
4.2. Implementación del sistema.....	49
4.2.1. PascalCasing.....	49
4.2.2. CamelCasing.....	50
4.4. Funcionalidades implementadas.....	50
4.6. Pruebas de software.....	52
4.6.1 <i>Prueba de Caja Blanca</i>	52
4.6.1.1 <i>Cálculo de la complejidad ciclomática a partir de un segmento de código.</i>	54
4.6.1.2 <i>Casos de prueba para el camino básico # 1</i>	55
4.6.1.3 Caso de prueba para el camino básico # 2.....	55
4.6.1.4 Caso de prueba para el camino básico # 3.....	55
4.6.2 Pruebas de Caja Negra	56
4.7 Conclusiones parciales.....	62
CONCLUSIONES.....	63
RECOMENDACIONES.....	64
REFERENCIAS.....	65

GLOSARIO 70

Índice de Figuras

Figura 1. Mapa de procesos	20
Figura 2. Diagrama del proceso Determinación de las necesidades individuales de capacitación.....	21
Figura 3. Modelo conceptual	22
Figura 4. Ejemplo de aplicación del patrón MVC en los diagramas de clase	36
Figura 5. Gestionar necesidades de capacitación.....	38
Figura 6. Diagrama de secuencia Adicionar acciones de capacitación	40
Figura 7. Diagrama Entidad-Relación	41
Figura 8. Diagrama de componentes	42
Figura 9. Resultados de la aplicación de la métrica TOC para cada clase del sistema.....	44
Figura 10. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad	44
Figura 11. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación	45
Figura 12. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización	45
Figura 13. Métrica Relaciones entre clases	46
Figura 14. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC	46
Figura 15. Resultados de la aplicación de la métrica RC para cada clase del sistema.	47
Figura 16. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento	47
Figura 17. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento	47
Figura 18. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización	47
Figura 19. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas	48
Figura 20. Interfaz Gestionar Plan individual	51
Figura 21. Grafo de flujo asociado al método	53

Índice de Tablas

Tabla 1. Tabla de sistemas.....	11
Tabla 2. Descripción del diagrama de clases del diseño. Gestionar acciones de capacitación.....	38
Tabla 3. Tamaño operacional de clases (TOC).....	43
Tabla 4. Rango de valores para la evaluación de la métrica TOC	43

Índice de Anexos

Anexo 1. Diagrama del proceso Elaborar el Plan de Capacitación y desarrollo de la entidad	67
Anexo 2. Diagrama del proceso Control de la ejecución del Plan de Capacitación y desarrollo	68
Anexo 3. Diagrama de clases del diseño Gestionar acciones de capacitación	69

INTRODUCCIÓN

El Recurso Humano, también conocido como Potencial Humano o Activo Humano, hace referencia al conjunto de trabajadores, empleados o personal que conforma un negocio o empresa.(34) En toda empresa existe un grupo de personas encargadas de la administración de los Recursos Humanos, facultadas para seleccionar, contratar, formar, emplear y retener a los colaboradores de la organización.

El valor intangible de los recursos humanos (RRHH) recibe el nombre de Capital humano, este constituye el recurso más importante dentro de una organización, su unión a instalaciones y tecnología de punta contribuyen al logro de los objetivos institucionales de la organización a la que pertenecen. Es por ello que desarrollar en el personal los conocimientos, habilidades y/o competencias necesarias es uno de los retos más importantes en los que se encuentran inmersas todas las empresas.

Toda empresa u organización tiene destinado su presupuesto para diversas actividades enfocadas en alcanzar sus metas y objetivos, sin embargo obvian un punto esencial: la capacitación y desarrollo de sus recursos humanos. La capacitación de los trabajadores se debe considerar no como un lujo o un gasto, sino como una inversión que contribuye a la formación del capital humano de la empresa que a su vez proporciona beneficios para la misma. Además esta capacitación ayuda al desarrollo personal y profesional de los mismos y posibilita que las empresas cuenten con trabajadores capaces en sus puestos de trabajo.

El Desarrollo es el proceso simultáneo a la capacitación dirigido a alcanzar multihabilidades, destrezas y valores en los trabajadores que les permiten desempeñar puestos de trabajo de perfil amplio, asegurando la formación del trabajador durante su vida laboral.(18)

La tendencia en estos últimos años muestra que un porcentaje muy alto de las compañías a nivel mundial, invierten entre el cinco y el ocho por ciento de su presupuesto anual en programas de Capacitación y Desarrollo para sus trabajadores.(39)

Los ERP (Aplicaciones o sistemas de planificación de recursos empresariales) son una de las herramientas más utilizadas por las empresas para digitalizar la documentación y procesos que se llevan a cabo en las mismas. Estos incluyen entre sus funcionalidades algunas encargadas de la gestión del Capital Humano y de manera específica su capacitación y desarrollo.

Las nuevas proyecciones de Cuba incluyen el desarrollo de un sistema ERP para informatizar los procesos de las empresas y como parte de esta proyección, en la Universidad de las Ciencias Informáticas (UCI) más específicamente en el “Centro de Informatización Gestión de Entidades (CEIGE)” se lleva a cabo la realización del Sistema Integral de Gestión de Entidades “**Cedrux**”. Dicho software cuenta con un subsistema encargado de las tareas que se llevan a cabo en las áreas de RRHH llamado Capital Humano, en el cual se realizan procesos como el cálculo de la nómina, reclutamiento, la evaluación del desempeño, la estimulación moral y material, entre otros. Sin embargo en estos momentos no cuenta con un componente que garantice el adecuado manejo de la capacitación y desarrollo de los trabajadores.

La administración de recursos humanos de cada organización con el objetivo de mejorar las competencias y habilidades de sus empleados lleva a cabo el proceso de capacitación y desarrollo de los mismos, proceso que en la actualidad en las empresas cubanas se realiza de forma manual o con hojas de cálculo y procesadores de texto, trayendo como consecuencia el aumento del margen de error dado por la manipulación de los datos de una o varias personas, lo que crea además un escenario propicio a ilegalidades dificultando al mismo tiempo el control de esta información. Influyendo en la eficiencia de la toma de decisiones en cuanto a la preparación y adiestramiento de los trabajadores. Además obstaculiza la determinación y elaboración del plan de capacitación, así como la preparación de inversiones destinadas a la capacitación del capital humano de la empresa. Las formas actuales de llevar a cabo este proceso, no cumplen con las necesidades de mantener actualizado todo el registro de la labor de capacitación para cada empleado de la organización, restan tiempo a las personas encargadas de los procesos relacionados y no son eficientes en la realización de esta tarea que posee un nivel de importancia elevado para cada institución o empresa; por lo que es imprescindible su inclusión en Cedrux, para facilitar el flujo de información haciendo su transmisión y tramitación más rápida y eficaz, con el menor costo posible en tiempo y recursos.

Por lo anterior se plantea como **problema a resolver**:

¿Cómo contribuir a mejorar la gestión del proceso de Capacitación y Desarrollo del capital humano en las empresas cubanas?

Se define como **objeto de estudio**: El proceso de Capacitación y Desarrollo del Capital Humano.

Como **campo de acción**: La informatización del proceso de Capacitación y Desarrollo del capital humano en las empresas cubanas.

Para darle solución al problema planteado se tiene como **objetivo general**: Desarrollar el componente Capacitación y Desarrollo, de manera que contribuya a la mejora de la gestión del Proceso Capacitación y Desarrollo del capital humano en las empresas cubanas.

Para cumplir con el objetivo y lograr una solución adecuada a la problemática, se plantean los siguientes **objetivos específicos**:

- Elaborar el marco teórico-referencial de la investigación a partir del estudio de las particularidades de los módulos de Capacitación y Desarrollo existentes.
- Realizar la modelación del negocio.
- Capturar y describir los requerimientos necesarios para el proceso de Capacitación y Desarrollo.
- Realizar el diseño del sistema a partir de los requisitos del proceso de Capacitación y Desarrollo.
- Implementar las funcionalidades necesarias para la gestión del proceso de Capacitación y Desarrollo.
- Validar el resultado obtenido mediante pruebas.

Se propone la siguiente **idea a defender**: Si se realiza el desarrollo del componente Capacitación y Desarrollo en el subsistema Capital Humano de Cedrux se contribuye a la mejora de la gestión del Proceso Capacitación y desarrollo.

Las **tareas a cumplir** son las siguientes:

1. Elaboración del modelo conceptual.
2. Elaboración del diagrama de procesos.
3. Análisis de los requisitos funcionales de los procesos.
4. Elaboración del modelo de datos.
5. Diseño de los nuevos componentes.
6. Realización de los diagramas de clases de los nuevos componentes.
7. Diseño de los diagramas de clases.
8. Realización de los Diagramas de interacción de cada una de las funcionalidades nuevas.

9. Implementación de la configuración de capacitación y desarrollo.
10. Implementación de las funcionalidades de capacitación y desarrollo.
11. Validación del resultado obtenido mediante pruebas de caja blanca y de caja negra.

En el presente trabajo de diploma la estructura en capítulos será la siguiente:

Capítulo 1: Fundamentación teórica. Describe el estudio del estado del arte de los sistemas que gestionan Capacitación y Desarrollo, así como la fundamentación de los conceptos, metodologías, lenguajes y herramientas a utilizar.

Capítulo 2: Propuesta de solución. Se describen las características del sistema proporcionando un punto de partida para realizar el modelado de negocio, donde se describen todos los procesos y requisito funcionales y se obtienen todos los artefactos para esta etapa en específico definidos por el Modelo de Desarrollo.

Capítulo 3: Análisis y Diseño del sistema. Se realiza el análisis y diseño del componente. Se describirán los patrones de diseño empleados.

Capítulo 4: Implementación y prueba. Se describe todo lo relacionado con la implementación del componente, así como la validación de la solución propuesta.

1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se abarcan un grupo de conceptos y teorías relacionadas con la formación de los trabajadores en las empresas. Se realiza una investigación a sistemas que gestionan el capital humano e incluyen el proceso de Capacitación y Desarrollo y se realiza una descripción de las herramientas y tecnologías a utilizar en el desarrollo del componente.

1.2 Principales Conceptos

1.2.1 Concepto de capital humano

El concepto y valor de capital humano como factor económico primario se conoció en 1964 con la publicación del libro **Capital Humano** de Gary S. Becker² donde señala que “la inversión en dar conocimientos, formación e información a las personas; esta inversión permite a la gente dar un mayor rendimiento y productividad en la economía moderna y aprovechar el talento de las personas”.(32) En la actualidad el capital humano está reconocido ampliamente como la fuente principal para la creación de la riqueza en esta "Era del Conocimiento". Al pasar de los años se han dado diversas definiciones acerca del Capital Humano como por ejemplo:

“Capital humano es el conjunto de conocimientos, habilidades y actitudes tanto presentes como potenciales de los empleados en una determinada organización a partir de los cuales desarrollará su propuesta de empresa y alcanzará los objetivos propuestos.” (17)

Luego surge otro concepto relacionado con el capital humano al que se le denomina **Administración del Capital Humano** (ARH), este término surge debido al desarrollo y los avances científicos - técnicos; abarca todas las funciones y responsabilidades dirigidas a atraer, contratar, desarrollar y retener los recursos de las personas, aspectos esenciales para el éxito de una compañía. Desde la perspectiva de la ARH *"En cualquier institución importante, sea empresa, organismo estatal o cualquier otra, la capacitación y desarrollo de su potencial humano es una tarea a la cual los mejores*

²Gary Stanley Becker (Pottsville, Pennsylvania, 2 de diciembre de 1930) es un economista estadounidense y profesor de la Universidad de Chicago. Recibió el Premio Nobel de Economía en 1992 por ampliar el dominio del análisis microeconómico a un mayor rango de comportamientos humanos fuera del mercado. Entre sus obras destacan: "Economía de la discriminación" (1957), "El Capital Humano" (1964) y "Tratado sobre la familia" (1981).

dirigentes han de dedicar enorme cantidad de tiempo y atención"(32); básicamente la capacitación impacta en los niveles de productividad, motivación e imagen organizacional.

La Capacitación es el conjunto de acciones de preparación que desarrollan las entidades laborales dirigidas a mejorar las competencias, calificaciones y recalificaciones para cumplir con calidad las funciones del puesto de trabajo y alcanzar los máximos resultados productivos o de servicios. Este conjunto de acciones permite crear, mantener y elevar los conocimientos, habilidades y actitudes de los trabajadores para asegurar su desempeño exitoso. (18)

El desarrollo del capital humano es el aumento en la capacidad de la producción del trabajo alcanzada con mejoras en las capacidades de los trabajadores. Estas capacidades se adquieren con el entrenamiento, la educación y la experiencia y se refiere al conocimiento práctico. (35)

La **acción de capacitación** expresa en forma descriptiva el objetivo final que se requiere lograr con los trabajadores que participan en la misma, a partir de las necesidades identificadas en el diagnóstico o determinación de necesidades de capacitación realizadas.(18)

El **modo de formación** define la vía o método a utilizar, mediante el cual se lleva a cabo la acción de capacitación. (18)

El **capacitador** es la persona especializada en la rama en que se va a dar la capacitación y es el encargado de enseñar al empleado las nuevas técnicas a utilizar dentro de la empresa. (10)

Luego de haber estudiado el criterio de los autores se puede decir que la **Capacitación** "es la adquisición sistemática de actitudes y conceptos, conocimientos, roles o habilidades con que se obtiene una mejoría del desempeño en el trabajo"(22) y el **Desarrollo** *es una actividad organizada, estructurada y en continuo proceso, cuyo objetivo es enriquecer y hacer a los trabajadores más capaces.*(35)

1.3 Sistemas que incluyen el proceso Capacitación y Desarrollo.

1.3.1 SAP ERP

Este software ayuda a optimizar la utilización, asignación y control de los recursos empresariales. Entre las múltiples funcionalidades que SAP ERP ofrece se pueden encontrar: Contabilidad financiera e interna, Análisis empresarial, Gestión de operaciones, Gestión de servicios corporativos,

Autoservicios y Gestión del capital humano. La solución SAP ERP Human Capital Management (SAP ERP HCM) tiene como ventajas que:

- Optimiza la contribución de cada empleado coordinando aptitudes, actividades e incentivos de los empleados con los objetivos empresariales y las estrategias necesarias para alcanzarlos.
- Proporciona herramientas para gestionar, medir y recompensar las contribuciones individuales y colectivas.
- Proporciona funcionalidades integradas que optimizan los procesos de Gestión del Capital Humano (HCM) y los integran a la perfección en todas las operaciones globales.
- Proporcionan acceso en tiempo real a la información que acelera la toma de decisiones por parte del personal.
- Le permiten asignar las personas correctas a los proyectos adecuados y en el momento oportuno.
- Dan soporte tanto a los empleados como a los directivos a lo largo de todo el ciclo de vida del empleado y capacitan a los empleados para gestionar procesos en un entorno colaborativo.(12)

Los principales subcomponentes son los siguientes:

- **Gestión del personal (PA)**
- **Gestión de tiempos (PT)**
- **Desarrollo del personal (PD)**

El componente **Desarrollo del personal** es utilizado para maximizar la utilidad de los empleados de la empresa, en este se pueden planificar e implementar medidas específicas de personal y formación para promover el desarrollo profesional de los empleados. Las necesidades de desarrollo del personal se determinan comparando los requisitos de trabajo actuales o futuros con las cualificaciones de los empleados, sus preferencias y sus aspiraciones. Utilizan planes para planificar y gestionar las medidas de desarrollo del personal a corto o largo plazo. Mediante la medición y el

análisis de los programas de recursos humanos, como los de la sucesión y el aprendizaje, se recibe información valiosa que le permite afinar sus estrategias de gestión del talento y el plan para el futuro. Realiza un perfil contra los puestos de trabajo, para determinar la habilidad y los conocimientos, que a su vez puede vincularse directamente a los planes de formación, para hacer frente a la necesidad de desarrollo individual. Posee un sistema de aprendizaje empresarial basado en la Web, clases virtuales y cursos on-line. (12)

Para su correcto funcionamiento necesita de otros módulos del propio software los cuales son : Gestión de organización, el componente Cualificaciones/Requisitos para fijar, estructurar y gestionar el catálogo de cualificaciones, Sistemas de cualificación para planificar, realizar y evaluar calificaciones, Planificación de carreras y sucesión para trabajar en los escenarios de planificación de carreras y sucesión, y Planes de desarrollo para elaborar planes globales e individuales de desarrollo del personal en el sistema.

SAP ERP a pesar de estar certificado para su posible utilización en Cuba es desarrollado para plataforma de software propietario, por lo que no es una solución factible, debido a que el país no cuenta con los recursos financieros necesarios para adquirirlo y sus licencias tienen un alto nivel monetario en el mercado internacional.

1.3.2 ERP GNU Enterprise

El ERP GNU Enterprise es un proyecto desarrollado por la comunidad de software libre, destinado al sector empresarial. Un conjunto de paquetes desarrollados utilizando las herramientas, para implementar un completo sistema ERP de recursos humanos, contabilidad, relaciones con los clientes y gestión de proyectos, hasta cadenas de suministros o comercio electrónico.

GNUe también es un conjunto de paquetes que utilizan herramientas para aplicar un recurso completo de planificación institucional (ERP). El módulo de Recursos Humanos está concebido para, entre otras funcionalidades, calcular nómina de sueldo, pago de retenciones, acumulación de vacaciones, gestionar el reclutamiento de sus empleados, y su formación. En este último mantiene una lista de instructores, clases, instalaciones, alumnos y cursos. Permite gestionar el proceso de autorizar la formación, la subcontratación de la misma y la evaluación del entrenamiento o formación recibida, mantiene un historial de los cursos y eventos de aprendizaje recibidos por los trabajadores.

Proporciona un plan de desarrollo de formación para cada empleado a partir de la comparación de las habilidades requeridas para cada puesto de trabajo y las habilidades de los trabajadores. (13)

GNUe a pesar de cumplir con la línea del país sobre la migración hacia aplicaciones de código abierto no puede ser utilizado como solución al problema que se expone puesto que no satisface las políticas establecidas en Cuba para la formación de los trabajadores en las empresas.

1.3.3 Sage ABRA HRMS

Es una herramienta para la gestión de la información del empleado. Sirve a empresas de cualquier tamaño, ayuda a administrar los beneficios de la empresa, con herramientas completas de administración de beneficios. (14)

- Mantiene registros de formación de los empleados donde refleja el estado actual de formación.
- Define las necesidades de formación en el momento de contratar.
- Permite definir las necesidades específicas de capacitación, preparación de informes de cumplimiento, y asegurar que las certificaciones se cumplan a tiempo.
- Ayuda a definir los requisitos de formación para los nuevos empleados en función de un conjunto de habilidades actuales.
- Sigue los requisitos de formación recurrentes para mantener las certificaciones de los empleados.
- Administra los costos de formación de manera eficaz.
- Permite tomar decisiones viendo la historia de los gastos del programa de formación.
- Comprueba automáticamente las capacidades y los requisitos previos del curso.
- Asignación automática de los empleados a la formación basada en los requisitos de trabajo.

Sage ABRA HRMS sin embargo no es una propuesta viable para la economía del país, se incurren en costosos gastos por los conceptos de licencia y mantenimiento de software, debido a que las tecnologías que se utilizan para el desarrollo y sustento del sistema son propietarios.

1.3.4 EXACT

Exact es una solución web que permite integrar, administrar, controlar y mejorar la forma de trabajo de todo el personal, así como añadir valor a través del negocio. (20)

- Utiliza los perfiles de competencia para identificar las necesidades de desarrollo.
- Permite la confección del plan de capacitación a partir de las competencias necesarias de cada cargo y las competencias deseadas por cada empleado, llevando un control del resultado de la formación del mismo en los distintos cursos.
- Permite a los trabajadores inscribirse en línea a los cursos y manda las solicitudes de aprobación a los jefes de forma automática.
- Proporciona prospectos de los cursos y gestiona el contenido de los mismos; registra y da seguimiento a los certificados.

En el estudio de estas soluciones informáticas se hizo uso de la técnica **Análisis de soluciones existentes**, donde se observaron las interfaces de los sistemas ERP existentes, obteniendo así una idea de posibles requisitos para la solución informática a desarrollar.

- **Análisis de soluciones existentes:** Esta técnica consiste en analizar diferentes sistemas ya desarrollados que posean características congruentes con el sistema a ser construido. Se analizan las interfaces de usuarios que estos presentan para comprender el tipo de datos que estos manejan y como lo realizan.(36)

En los sistemas anteriormente estudiados se observan algunas de las limitantes que estos presentan para su utilización en Cuba, en algunos existe la necesidad de comprar el sistema en su totalidad

para su correcto funcionamiento debido a que necesita integración con el resto de los subsistemas, siendo poco útil adquirir solamente algunos módulos. A continuación en la Tabla 1 se muestran las limitaciones que presentan los sistemas analizados. Se puede apreciar como los software están contruidos para ser utilizados en plataformas privativas, característica que difiere con la política de migración a software libre de nuestro país.

Características	Sistemas			
	Sage ABRA HRMS	SAP ERP HCM	ERP GNU Enterprise	EXACT
Cumple con la Resolución Cubana No. 29/06.	NO	NO	NO	NO
Cumple con el paradigma de independencia tecnológica del país.	NO	NO	SI	NO
Dependencia entre módulos.	NO	SI	NO	NO

Tabla 1. Tabla de sistemas

Luego de estudiar y analizar los sistemas anteriores es posible decidir que ninguno cumple con todas las características para una adecuada solución al problema que se expone, por lo que se cree necesario el desarrollo de un sistema que responda a las particularidades de la economía cubana, que incluya las características positivas de los sistemas mencionados anteriormente. En lo que a Capacitación y desarrollo se refiere, en la mayoría de los casos los sistemas poseen funcionalidades requeridas por el Sistema de Gestión Integrada de Capital Humano en Cuba, pero no recogen todos los requisitos de la Resolución Cubana No. 29/06.

1.4 Modelo de desarrollo de software

El modelo de desarrollo propuesto por CEIGE describe la secuencia de actividades para la construcción y desarrollo de soluciones informáticas. Para el ciclo de vida de los proyectos del centro se tienen en cuenta la secuencia de actividades y disciplinas por áreas de procesos planteadas en el nivel dos de CMMI (*Capability Maturity Model Integration*) establecido en la UCI en julio del 2011.

A continuación se detallan las disciplinas y los artefactos a generar por cada una de ellas, teniendo en cuenta la estructura del presente trabajo.

Modelado del negocio: Es la disciplina destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito. (3)

- Modelo Conceptual
- Mapa de procesos de Negocio
- Descripción de procesos de negocio

Requisitos: El esfuerzo principal en la disciplina de Requisitos es desarrollar un modelo del sistema que se va a construir. Incluye un conjunto de artefactos que describen todas las interacciones que tendrán los usuarios con el software y que responden a los requisitos funcionales del sistema. Se especifican los requisitos funcionales y no funcionales. (3)

- Especificación de requisitos de software
- Descripción de requisitos
- Documento de salidas del sistema (actualizado)

Análisis y diseño: Durante esta disciplina es modelado el sistema para que soporte todos los requisitos. Esto contribuye a una arquitectura sólida y estable que se convierte en un plano para la próxima fase. Los artefactos generados en esta etapa son más formales y específicos de una implementación. En caso de llevarse a cabo la reutilización de componentes software ya desarrollados, durante esta fase se ajusta el modelado existente a los requisitos actuales. (3)

- Diagrama de clases del diseño por componentes o por módulos
- Diagrama de secuencia
- Diagrama de componentes
- Modelo de datos

Implementación: A partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares. Al reutilizar componentes software ya implementados se lleva a cabo el desarrollo necesario para ajustar a los requisitos actuales y posteriormente realizar la integración de los componentes. (3)

- Implementación del componente Capacitación y Desarrollo

Pruebas internas: Durante esta disciplina se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple. (3)

- No conformidades
- Acta de liberación

1.5 Tecnologías y herramientas para el desarrollo

El Departamento de Tecnología del CEIGE cuenta con un marco de trabajo denominado Sauxe, el cual se utiliza para el desarrollo de los proyectos del centro. Sauxe cuenta con una arquitectura en capas que a su vez presenta en su capa superior un Modelo Vista Controlador³. Está compuesto por un conjunto de componentes reutilizables que provee la estructura genérica, con soporte para entornos multientidad, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo, alejando a los programadores de los detalles arquitectónicos. (16)

1.6 Marco de trabajo

El marco de trabajo utiliza tecnologías libres como:

- **Zend_Framework 1.7:** Es un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es de código abierto y trabaja con PHP 5. Está formado por una serie de métodos estáticos y varios componentes que se pueden reutilizar. (16)
- **Doctrine 1.2.1:** Potente y completo sistema de mapeo relacional de objetos (por sus siglas en inglés ORM) para PHP 5.2 con una capa de abstracción de bases de datos incorporado. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos. (16)

³Modelo Vista Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- **Extjs 2.2:** Es una biblioteca o conjunto de librerías de JavaScript para el desarrollo de aplicaciones web interactivas, usa tecnologías AJAX⁴, DHTML⁵ y DOM⁶. Permite realizar completas interfaces de usuario, fáciles de usar, muy parecidas a las conocidas aplicaciones de escritorio, posibilitando a los desarrolladores concentrarse en la funcionalidad de las aplicaciones en vez de en las advertencias técnicas. (16)

1.7 Herramienta para el modelado

- **Visual Paradigm 6.4:** Se decidió utilizar el Visual Paradigm para visualizar y diseñar los elementos de software debido a que es multiplataforma y por las facilidades que brinda para el diseño de los diagramas necesarios y su documentación. Soporta los principales estándares como UML⁷, SysML⁸, BPMN⁹ y XML¹⁰. Ofrece un completo conjunto de herramientas a los equipos de desarrollo de software, necesarios para la captura de requisitos, la planificación de software, la planificación de pruebas, el modelado de clases y el modelado de datos. (16)

1.8 Herramienta para el desarrollo colaborativo

- **SVN (Subversión) 1.6.6:** Es un software libre de sistema de control de versiones, bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversión es que los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo. (16)

⁴ Asynchronous JavaScript And XML

⁵ HTML Dinámico

⁶ Document Object Model

⁷ Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

⁸ (Systems Modeling Language) es un lenguaje de especificación de sistemas.

⁹ Business Process Modeling Notation, estándar de modelado de procesos de negocio.

¹⁰ eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C).

1.9 Lenguajes de programación

- **JavaScript:** Es un lenguaje de programación orientado a objetos, basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Este lenguaje está basado en acciones que posee menos restricciones, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas, entre otros. Con JavaScript se puede crear diferentes efectos e interactuar con nuestros usuarios, es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros. (16)
- **PHP 5.2.6:** Lenguaje de programación generalmente usado en el desarrollo de sitios web dinámicos y actualmente es casi el más utilizado en todo el mundo, solo se necesita un navegador web para ejecutarlo, es usado mayormente del lado del servidor, por lo que los script se ejecutan remotamente y el resultado aparece en la máquina cliente (local), tiene soporte para muchos tipos de bases de datos, entre las principales están MySQL, PostgreSQL, SQLite, entre otras. PHP es una solución escalable de alto rendimiento, con todas las herramientas que se pueden esperar de un lenguaje de programación, como son los bucles y las operaciones matemáticas. Puede manejar variables, enviar cabeceras HTTP, crear gráficos en tiempo real, instalar cookies, administrar la autenticación, redireccionar usuarios y mucho más. (16)

1.10 Notaciones para el modelado

- **UML:** Lenguaje unificado de modelado o Unified Modeling Language según sus siglas en inglés. Constituye un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. Concebido como un lenguaje de modelado de propósito general, por lo que puede emplearse para la especificación de la mayoría de los sistemas orientados a objetos o solo para componentes. Igualmente útil para el modelado de aplicaciones de disímiles dominios de aplicación y plataformas de objetos, proporcionando una gran flexibilidad y expresividad a la hora de modelar sistemas. (16)

- **BPMN** (Business Process Modeling Notation): Es el nuevo estándar para el modelado de procesos de negocio y servicios web. Es una notación a través de la cual se expresan los procesos de negocio en un diagrama de procesos de negocio (BPD). Este estándar agrupa la planificación y gestión del flujo de trabajo, así como el modelado y la arquitectura. Proporciona un lenguaje gráfico común, con el fin de facilitar su comprensión a los usuarios de negocios. Integra las funciones empresariales. Utiliza una Arquitectura Orientada por Servicios (SOA), con el objetivo de adaptarse rápidamente a los cambios y oportunidades del negocio. Combina las capacidades del software y la experiencia de negocio para optimizar los procesos y facilitar la innovación del negocio.(28)

1.11 Gestor de Base de Datos

- **PostgreSQL 8.3:** Como gestor de bases de datos se decidió utilizar este ya que incluye características como son la herencia, tipos de datos, funciones, restricciones, disparadores, además de reglas e integridad transaccional, es multiplataforma, existe suficiente documentación pública y libre. Utiliza un modelo cliente/servidor además de usar también multiprocesos por lo que un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. (16)

1.12 Servidor Web

- **Apache 2.2.9:** Se decidió utilizar este servidor de aplicaciones ya que es un servidor gratuito de código fuente abierto y potente que ofrece un servicio estable y sencillo para su mantenimiento y configuración; es multiplataforma, permite personalizar la respuesta ante posibles errores que el servidor pueda tener, es altamente configurable en la creación y gestión de logs. (16)

1.13 Navegador Web

- **Mozilla Firefox 2.17:** Presenta una forma rápida y eficiente de navegar por la web, contiene un PluginFirebug que se utiliza para ver los errores del código, con la integración de este Plugin a Firefox se puede inspeccionar y editar HTML lo cual hace que sea fácil de encontrar

elementos HTML enterrados profundamente en la página. Permite monitorear las actividades de la red y saber por qué las páginas tardan en cargar. Incluye un potente depurador de JavaScript que le permite poner en pausa la ejecución en cualquier momento, brinda la posibilidad de saber inmediatamente y proporcionar información detallada y útil sobre errores de JavaScript, CSS¹¹ y XML¹². Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL.

(16)

1.14 Conclusiones parciales

El desarrollo de software para la gestión del Capital Humano en las empresas en los últimos años ha tenido una evolución considerable. En este capítulo se realizó un estudio de los sistemas más relevantes que incluyen el proceso de Capacitación y Desarrollo. Se conoció su inadecuada adaptación a las necesidades con el paradigma de independencia tecnológica por el cual apuesta el país, por ello la necesidad existente en el desarrollo del componente en el subsistema Capital Humano del Sistema Cedrux, que responda a las características específicas de las entidades laborales de nuestro país. Se describen las tecnologías, herramientas y el modelo de desarrollo demostrando por sus características que son las adecuadas para el desarrollo de la solución. Todos estos elementos antes mencionados conforman la fundamentación teórica que sustenta el desarrollo del presente trabajo.

¹¹ Cascading Style Sheets

¹² eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C).

2. PROPUESTA DE SOLUCIÓN

2.1 Introducción

En el presente capítulo se brinda una descripción del proceso de Capacitación y desarrollo, se describen las técnicas utilizadas para la captura de requisitos, así como la descripción de los requisitos funcionales para un posterior modelado del diseño. Se obtendrán varios artefactos que serán de ayuda para las posteriores etapas del desarrollo como son: la descripción del proceso, los mapas de procesos y el modelo conceptual.

2.2 Descripción del negocio

El proceso de Capacitación y Desarrollo en las empresas nace del equilibrio necesario entre los conocimientos y habilidades actuales y futuras de las personas, y las necesidades presentes y futuras de la organización en función de su entorno, su misión y su estrategia. El sistema de formación o entrenamiento apoya a los miembros de la organización para desempeñar su trabajo actual con la calidad requerida, colaborando en el desarrollo de la persona para cumplir futuras responsabilidades.

Las acciones de Capacitación y Desarrollo de los Recursos Humanos que realiza la dirección de la entidad laboral, deben estar relacionadas con los procesos de producción o servicios que estas realizan y con los conocimientos, habilidades y actitudes que deben poseer los recursos humanos para llevarlos a cabo. La dirección de la entidad laboral es la encargada de definir los objetivos, el alcance, las metas y los resultados que se proponen lograr, en materia de capacitación y desarrollo de su personal, el Director es el máximo responsable de que se ejecute la planificación, organización, ejecución y control de los resultados de la capacitación y desarrollo de los recursos humanos.

Para asegurar este proceso en los recursos humanos se debe:

- Organizar y ejecutar el diagnóstico o determinación de las necesidades de capacitación de los trabajadores en correspondencia con los puestos de trabajo que desempeñan, los objetivos y las proyecciones de desarrollo de la entidad.
- Discutir y acordar con los trabajadores y las organizaciones sindicales los planes de capacitación y desarrollo elaborados.

- Elaborar y cumplir el plan anual de capacitación y desarrollo de los recursos humanos para dar respuesta a las necesidades identificadas.
- Organizar y coordinar las acciones de capacitación necesarias para resolver las necesidades de capacitación de los trabajadores que hayan sido identificadas.
- Asegurar los recursos humanos, materiales y financieros necesarios para esta actividad, que le permita contar con la base material de estudio, los medios técnicos, las instalaciones, los instructores y otros recursos que se requieran.
- Propiciar la incorporación de los trabajadores que lo necesiten a los distintos niveles del sistema de educación. (18)

La gestión de la Capacitación y desarrollo cuenta con tres subprocesos fundamentales:

- **Determinación de las necesidades individuales de capacitación:** Este es el elemento fundamental que debe realizar la dirección de la entidad para elaborar luego el plan individual de capacitación y desarrollo de cada trabajador, es la base de la planificación de su organización y ejecución en la entidad, por lo que debe realizarse a nivel de las diferentes áreas y categorías ocupacionales. Para realizar este proceso se utilizan diferentes técnicas y herramientas que permiten identificar la brecha de conocimientos, habilidades y actitudes que presenta cada trabajador y los requerimientos que se exigen para el cargo que desempeña.
- **Elaborar el Plan de capacitación y desarrollo de la entidad:** La dirección de la entidad, en correspondencia con sus objetivos globales y específicos y los resultados del diagnóstico o determinación de necesidades de capacitación, determina la estrategia a seguir para llevar a cabo la capacitación y desarrollo de sus recursos humanos y para ello elaboran y aprueban, en el último trimestre del año en curso, el plan de capacitación y desarrollo para el próximo año.
- **Control de la ejecución del Plan de Capacitación y desarrollo:** Trimestralmente se evalúa por la dirección de la entidad laboral, el cumplimiento del plan de capacitación aprobado, teniendo en cuenta el cumplimiento del inicio y fin de las acciones de capacitación previstas y la participación de los trabajadores previstos a matricular en cada acción.

2.3 Modelado del negocio

2.3.1 Mapa de procesos

El mapa de procesos ofrece una visión general del sistema, es útil para representar gráficamente la interacción entre los mismos. Con el fin de lograr una mayor visión en cuanto a las interrelaciones con los procesos dentro del negocio de Capacitación y Desarrollo, en la Figura 1 y **¡Error! No se encuentra el origen de la referencia.** se muestran los mapas de procesos de las relaciones internas y externas respectivamente.

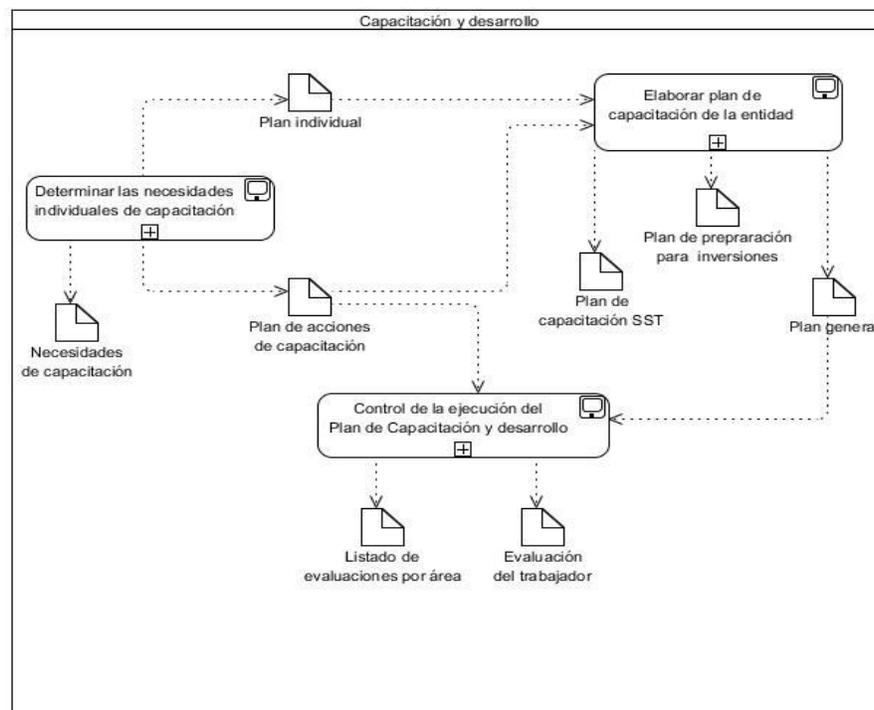


Figura 1. Mapa de procesos

El mapa de procesos antes presentado ofrece una visión gráfica de las entradas y las salidas en los diferentes momentos que conforman el proceso de Capacitación y Desarrollo, además del flujo de información entre los mismos.

2.4 Procesos de Negocio

Un **proceso de negocio** es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene entradas, funciones y salidas. Puede ser parte de un proceso mayor que lo abarque o bien puede incluir otros procesos de negocio que deban ser incluidos en su función. (11)

El proceso **Determinación de las necesidades individuales de capacitación** tiene como objetivo identificar las necesidades de capacitación por área en la empresa, así como la elaboración del plan individual de cada trabajador. El resto de los procesos y sus descripciones detalladas se pueden encontrar en los anexos.

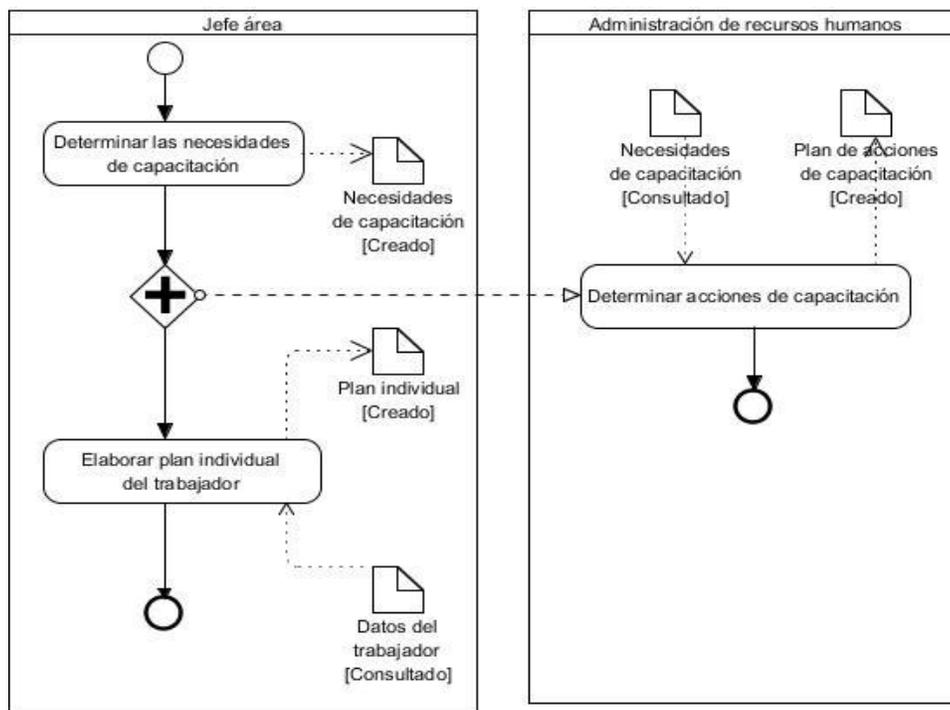


Figura 2. Diagrama del proceso Determinación de las necesidades individuales de capacitación

2.5 Modelo conceptual

El modelo conceptual se concibe como un mapa de conceptos y sus relaciones incluyendo suposiciones acerca de la naturaleza tanto de los fenómenos que esos conceptos representan como sus relaciones e incluyen un alto nivel de abstracción.(29)

En la Figura 3 se muestra el modelo conceptual que se generó para el proceso de Capacitación y Desarrollo.

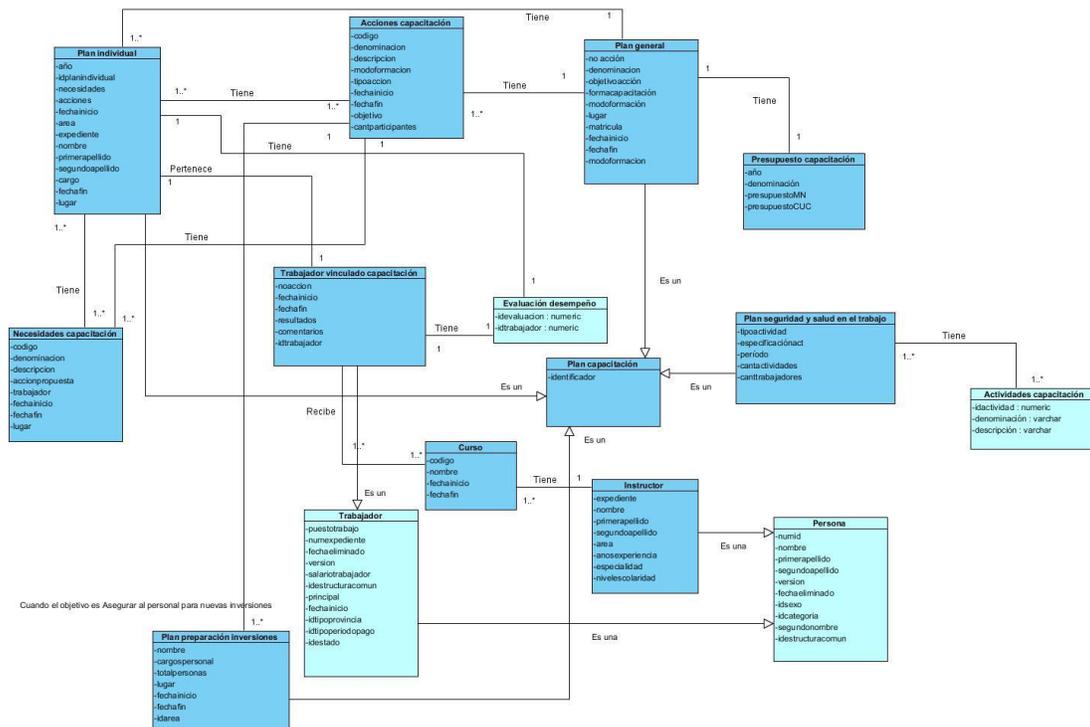


Figura 3. Modelo conceptual

A continuación se describen cada uno de los conceptos presentes en la ilustración anterior:

- **Acciones capacitación:** Documento que contiene las acciones de capacitación necesarias para resolver las necesidades de capacitación de los trabajadores que hayan sido identificadas.
- **Necesidades capacitación:** Documento que contiene todas las necesidades de capacitación encontradas en cada área de la empresa.

- **Plan individual:** Concepto que contiene todos los planes individuales de capacitación realizados a cada trabajador de la empresa que necesite capacitación.
- **Trabajador vinculado capacitación:** Entidad que contiene los datos de los trabajadores que se encuentren vinculados a capacitación.
- **Curso:** Entidad que contiene los datos de los cursos de capacitación activos en la empresa, así como el registro de los trabajadores que se encuentran recibiendo un curso.
- **Instructor:** Entidad que contiene los datos de los instructores de los cursos de capacitación. El instructor del curso puede ser un trabajador interno de la empresa o no que esté capacitado para impartir del curso.
- **Plan preparación para inversiones:** Almacena los datos de los planes de preparación para inversiones. Estos planes se realizan cuando el objetivo de la acción de capacitación sea “Asegurar al personal para nuevas inversiones”.
- **Plan general:** En esta entidad se guardan los datos del plan general de capacitación de la entidad laboral. Se elaboran y se aprueban en el último trimestre del año en curso, el plan de capacitación y desarrollo del próximo año.
- **Presupuesto capacitación:** Entidad en la cual se guarda el presupuesto aprobado por la dirección de la entidad laboral para el plan de capacitación y desarrollo.
- **Plan seguridad y salud en el trabajo:** En esta entidad se guardan los datos del plan de capacitación relacionado con seguridad y salud en el trabajo.

2.6 Técnicas y métodos empleados en la captura de requisitos

Durante la disciplina de Requisitos los analistas se valen de diferentes técnicas de capturas de requisitos que es “*la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema*”. (30) Algunas de las técnicas utilizadas en este trabajo son:

- **Entrevistas:** El uso de las entrevistas le permite al analista obtener conocimiento del problema y comprender los objetivos de la solución, mediante esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen disímiles tipos de entrevista y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización. En este caso se han concebido tres fases para realizarla: preparación de la

entrevista (fijar el objetivo que se pretende alcanzar y definir previamente las preguntas), realización de la entrevista y análisis (una vez culminada la entrevista es necesario leer las notas tomadas para confirmar la información). Las entrevistas fueron realizadas a Yoneida Cazaña Rodríguez Especialista General en el área de Capital humano y los temas fueron relacionados con la definición de las funciones para el proceso de Capacitación y desarrollo y la elaboración del plan general de capacitación de la entidad.

- **Estudio de la documentación:** El uso de esta técnica depende de la información que se tiene almacenada acerca del software que se desee desarrollar. Las entidades laborales guardan información relacionada con los procesos que realizan a diario, los modelos o informes necesarios para su desarrollo. El papel de los analistas con esta técnica se basa en determinar posibles requerimientos sobre la base de analizar la documentación utilizada por la empresa. En este caso se estudiaron documentos tales como:

Leyes y resoluciones constitucionales:

- RESOLUCIÓN No. 29/06 ORG. Y CONTROL Y DESA. CAPACITACION.
- RESOLUCIÓN No. 21/99.
- Norma cubana 3001. Sistema de Gestión Integrada de Capital Humano-Requisitos.

Otros documentos creados por funcionales, especialistas y clientes para el uso interno del proyecto:

- CH-CD-02, Capacitación y Desarrollo, Procesos del Negocio.

Otra de las técnicas utilizadas fue el Análisis de soluciones existentes mencionada en el capítulo anterior en el epígrafe del estudio de los sistemas existentes.

2.6.1 Patrones de reutilización de requisitos

Los requisitos funcionales obtenidos en este trabajo utilizan los patrones de reutilización: alta, baja, modificación y consulta (CRUD, Create, Read, Update y Delete en inglés). Este patrón permite realizar agrupaciones de requisitos donde se modelan todas las operaciones que pueden ser realizadas sobre la aplicación.

2.7 Especificación de los Requisitos del Sistema

Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar.(23) Para la elaboración del proceso de Capacitación y Desarrollo se obtuvieron 10 agrupaciones de requisitos para un total de 45 requisitos funcionales. A continuación se muestra un listado de los mismos.

2.7.1 Requisitos Funcionales

1 RF: Gestionar acciones capacitación:

- Adicionar acciones capacitación
- Modificar acciones capacitación
- Eliminar acciones capacitación
- Listar acciones capacitación

2 RF: Gestionar necesidades capacitación:

- Adicionar necesidades capacitación
- Modificar necesidades capacitación
- Eliminar necesidades capacitación
- Listar necesidades de capacitación

3 RF: Gestionar plan individual:

- Adicionar plan individual capacitación
- Modificar plan individual capacitación
- Eliminar plan individual capacitación
- Listar necesidades capacitación por plan individual
- Listar plan individual

4 RF: Gestionar plan general capacitación:

- Mostrar plan general de capacitación
- Listar acciones de capacitación asociadas al plan general
- Listar plan individual de cada trabajador

5 RF: Gestionar plan de capacitación de seguridad y salud en el trabajo (SST)

- Adicionar plan de capacitación SST

Modificar plan de capacitación SST
Eliminar plan de capacitación SST
Listar actividades de capacitación SST
Listar plan de capacitación SST

6 RF: Gestionar instructor:

Adicionar instructor
Modificar instructor
Eliminar instructor
Listar curso de capacitación por instructor
Listar instructores

7 RF: Gestionar cursos:

Adicionar curso
Modificar curso
Eliminar curso
Emitir evaluación final
Listar cursos de capacitación

8 RF: Gestionar trabajadores vinculados a capacitación:

Adicionar trabajador
Modificar trabajador
Eliminar trabajador
Listar curso de capacitación por trabajador
Listar trabajadores
Inscribirse en curso de capacitación

9 RF: Gestionar plan de preparación para inversiones:

Adicionar plan de inversiones
Modificar plan de inversiones
Eliminar plan de inversiones
Listar plan de inversiones

10 RF: Registrar presupuesto anual de capacitación:

Adicionar presupuesto anual

Modificar presupuesto anual

Eliminar presupuesto anual

Listar presupuesto anual

Especificación del requisito Gestionar acciones de capacitación

Descripción textual del requisito Adicionar acción de capacitación

Precondiciones	El usuario este autenticado en el sistema.
Flujo de eventos	
Flujo básico	
1	Se introducen los datos de la acción de capacitación: Código. Denominación. Descripción. Objetivo. Modo formación. Tipo acción. Fecha inicio. Fecha fin.
2	El sistema valida los datos introducidos(Ver validaciones 1)
3	Si los datos son correctos el sistema los registra.
4	El sistema confirma el registro de los datos.
5	Concluye el requisito.
Pos-condiciones	
1	Se registró en el sistema una nueva acción de capacitación.
Flujos alternativos	
Flujo alternativo 2.a Información incompleta	
1	El sistema señala los datos vacíos y permite llenarlos.
2	El usuario llena los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 2.b Información errónea	
1	El sistema señala los datos erróneos y permite corregirlos.
2	El usuario corrige los datos.
3	Volver al paso 2 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 2.c Adicionar plan inversiones	

1	Se introducen los datos de la acción de capacitación.	
2	El sistema valida los datos introducidos(Ver validaciones 1)	
3	Si el usuario selecciona el objetivo Aseguramiento del personal para planes de inversiones.	
4	Se solicita confirmación para adicionar el plan de inversiones.	
5	Si el usuario confirma se adiciona el plan de inversiones.	
6	El sistema confirma el registro del plan.	
Pos-condiciones		
1	Se adiciona el plan de preparación para inversiones	
Flujo alternativo *.aEl usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se registran los datos.	
Validaciones		
1	Se validan los datos según lo establecido por el modelo conceptual Capacitación y desarrollo..	
Relaciones	Requisitos Incluidos	Listado de los requisitos incluidos por el requisito. N/A
	Extensiones	N/A
Conceptos	Acciones de capacitación	Visibles en la interfaz: Código. Denominación. Descripción. Objetivo. Tipo acción. Utilizados internamente: N/A

2.8 Requisitos no funcionales

Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación del sistema. Estos surgen de las necesidades del cliente (políticas de seguridad de la organización, fiabilidad, accesibilidad) y son aplicados al sistema en su totalidad. (25)

De acuerdo con los requerimientos suplementarios del CEIGE algunos de los requisitos con los que contará el sistema son:

Seguridad

- Todo uso de las funcionalidades del sistema requiere la autenticación de los usuarios.
 - El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema.

- El sistema manejará mecanismos de encriptación para las contraseñas de los usuarios.
- El sistema registrará las trazas de operaciones realizadas por cada usuario en todo momento.

Confiabilidad

Madurez

- El sistema no permitirá la entrada de datos incorrectos.
- El sistema impondrá campos obligatorios para garantizar la integridad de la información que se introduce por el usuario.
- Ninguna información que se haya ingresado en el sistema y se haya asociado a alguna operación será eliminada físicamente de la base de datos.
- El sistema contendrá un mecanismo de alertas y avisos sobre cambios de estados y/o realización de operaciones.

Tolerancia ante fallos

- Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que no dependen de esta deberán seguir funcionando.
- Ante el fallo de una funcionalidad del sistema, el resto de las funcionalidades que dependen de esta deberán notificar su imposibilidad de realización debido a la causa original.
- El sistema permite detectar fallos internos y notificar al usuario de la ocurrencia de estos.

2.9 Validación de los requisitos

La validación de requisitos tiene como misión demostrar que la captura de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (40). El proceso de validación de requisitos comprende actividades que generalmente se realizan una vez obtenida una primera versión de la documentación de requisitos. Para realizar la aprobación de los requisitos es posible apoyarse de las siguientes técnicas

- **Prototipos de interfaz de usuarios:** Mediante esta técnica se logra una versión inicial del sistema demostrando cómo se van a disponer posteriormente los conceptos que intervienen en el mismo. Los prototipos son la manera en que el cliente pueda apreciar una primera muestra del producto, y dar ideas o recomendaciones que guíen las necesidades que la aplicación deba cumplir.

- **Revisiones:** Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.(36)

2.9.1 Criterios para la verificación de los requisitos

Luego de haberse realizado la validación de los requisitos al proceso de Capacitación y Desarrollo se desean verificar dichos requisitos como parte del trabajo de investigación y a fin de obtener numéricamente una evaluación y aceptación de los mismos como parte de la solución del sistema. Para ello se utilizaron los criterios de correctitud, completitud y consistencia del “modelo de McCall”¹³ por ser criterios de verificación reconocidos, fáciles de aplicar y arrojar resultados cuantitativos con los que se puede demostrar la calidad.

Correctitud: La correctitud (C) es aceptable si el valor que se obtiene es menor estricto que 0.10 y se obtiene dividiendo el total de observaciones de negocio realizadas mediante la validación (TO) entre el total de requisitos revisados (T).

Mediante la validación de los requisitos se detectó que en la agrupación de requisitos funcionales Gestionar necesidades de capacitación la falta del requisito Listar necesidades por área, siendo esta la única observación realizada en el negocio.

$$C = TO/T$$

Teniéndose un total de cuatro especificidades de negocio realizadas mediante la validación (TO) de un total de 44 requisitos que han sido generados en este trabajo y pasaron el proceso de revisión (T). Teniéndose entonces que:

$$C = 4/44$$

$$C = 0.01$$

El resultado obtenido de correctitud es de un 0.01 < 0.10 por tanto, las especificaciones de requisitos son aceptadas.

¹³ El modelo de McCall fue el primer modelo presentado para la verificación de la calidad del software en 1977, y se originó motivado por US Air Force y DoD.

Compleitud: La completitud (CO) es aceptable si el valor obtenido es mayor que 0.90 y se obtiene restándole 1 a la división entre el total de observaciones de formato y el total de secciones del documento (TS).

$$CO = 1 - OF/TS.$$

Teniéndose en cuenta que no hubo secciones de formato vacías para un total de seis secciones obligatorias por cada documento de especificación de requisitos se obtuvieron cero observaciones en el registro de formato:

$$CO = 1 - 0/6.$$

$$CO = 1.$$

El resultado obtenido de completitud es de $1 > 0.90$ por tanto, las especificaciones de requisitos se consideran completas.

Consistencia: La Consistencia (CI) es aceptable si el valor obtenido es menor que 0.20 y se obtiene de la división entre el total de observaciones de consistencia (O) y el total de requisitos revisados (T).

$$CI = O/T$$

Teniéndose cero observaciones en el registro de consistencias ya que no se detectaron observaciones de inconsistencias durante la etapa de validación para un total de 44 especificaciones de requisitos revisados:

$$CI = 0/44$$

$$CI = 0$$

El resultado obtenido de consistencia es de $0 < 0.20$ por tanto, las especificaciones de requisitos se consideran consistentes.

2.10 Conclusiones parciales

En este capítulo se evidencia la etapa inicial y el desarrollo de la fase de Modelado de negocio propuesta por el modelo de desarrollo del centro CEIGE. Mediante la descripción y modelación de los procesos del negocio, la elaboración del modelo conceptual, y con el uso de diferentes técnicas de captura de requisitos, así como las técnicas para una correcta validación de requisitos, se obtuvieron las especificaciones de requisitos funcionales que debe tener el sistema. De manera general en esta etapa se describieron 10 agrupaciones de requisitos para un total de 45 requisitos funcionales que responden a los procesos descritos y se obtuvo el modelo conceptual de Capacitación y Desarrollo,

conformando así un conjunto inicial de evidencias para el posterior diseño e implementación de la aplicación.

3. ANÁLISIS Y DISEÑO

3.1 Introducción

En el presente capítulo se especifican los patrones de arquitectura y del diseño del sistema, así como una representación gráfica de diagramas del diseño correspondiente al sistema, entre los que está el diagrama de clase del diseño y los diagramas de secuencia, realizándose una breve descripción de los mismos.

3.2 Patrones de diseño

3.2.1. Patrones GOF

Los patrones de diseño son un conjunto de estrategias, o buenas prácticas, por lo que su aplicación en el diseño del sistema puede facilitar el trabajo en muchas situaciones.

Uno de los patrones en que está basado el diseño de este trabajo es en GoF acrónimo de Gang of Four (Banda de los cuatro) por sus siglas en inglés. Dichos patrones permiten minimizar las dependencias y reducir la complejidad. Favorece a un Bajo Acoplamiento entre los clientes y los subsistemas, respondiendo a uno de los patrones GRASP, permiten variar las clases internas, de manera transparente a los clientes que las usan.(23)

Los patrones GOF están divididos en tres grandes grupos: estructurales, creacionales y de comportamiento. De ellos son utilizados en la modelación del diseño del proceso de Capacitación y Desarrollo los siguientes:

- **Fachada:** Es un patrón estructural que proporciona una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan. (9) La utilización de este patrón está reflejada en la creación y empleo de la clase IOC ya que esta funciona como intermediario en la comunicación entre componentes.

Cadena de responsabilidad: Es un patrón de comportamiento que evita acoplar el emisor de una petición a su receptor, al dar a más de un objeto la posibilidad de responder a la petición. Crea una cadena con los objetos receptores y pasa la petición a través de la cadena hasta que esta sea tratada por algún objeto. En este trabajo su uso en el diseño es enfocado principalmente para validar la producción de errores al insertar elementos en la base de datos, el cual es captado por las capas

superiores, reenviando la excepción hasta la capa de aplicación donde dicho error se traduce al lenguaje del usuario. Este patrón se refleja en las clases controladoras y las del modelo donde se establece la una cadena a seguir para las peticiones de la clase controladora al modelo y del modelo a la entidad. Ejemplos: GestionarNecesidadescapController y DatNecesidadescapModel.

- **Instancia única:** Es un patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.(24) Se evidencia en las clases del dominio las cuales poseen métodos estáticos a los cuales se puede acceder sin necesidad de crear nuevas instancias de dichas clases.

3.2.2. Patrones GRASP

Otro de los patrones en los que se basa el diseño de este trabajo son los patrones GRASP (General Responsibility Assignment Software Patterns, Patrones Generales de Software para Asignación de Responsabilidades), describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. Existen nueve patrones Grasp de los cuales se aplican: Experto, Creador, Controlador, Alta cohesión, Bajo acoplamiento.

- **Experto:** El patrón experto asignar responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.(24) Este patrón fue utilizado en la implementación de varias clases en la aplicación, ejemplo de algunas son las clases controladoras y las clases del modelo, pues son las que cuentan con la información necesaria para cumplir sus responsabilidades. Ejemplo: DatNecesidadescapModel, GestionarNecesidadescapController.
- **Creador:** Permite crear instancias de otras clases según la responsabilidad de la misma. Esto posibilita un bajo acoplamiento y mejores oportunidades de reutilización.(24) Este patrón se utilizó en la implementación de las clases del paquete Domain, quienes son las encargadas de crear los objetos de tipo Doctrine_Query, para permitir el acceso a la información almacenada a nivel de datos, ejemplo de algunas son: DatAccionescapModel, DatInstructorModel y DatPlanindividualModel.

- **Controlador:** La aplicación del patrón Controlador consiste en asignar la responsabilidad de administrar un mensaje de eventos del sistema a una clase que represente: el negocio, la organización global, o el sistema global.(37)El patrón controlador se utiliza en este caso debido a que la aplicación es muy extensa, brindando la posibilidad de tener en vez de un controlador, varios controladores (Controllers) más pequeños especializados en las funcionalidades propias de cada agrupación de requisitos impidiendo la sobresaturación.
- **Alta cohesión:** La cohesión (o más exactamente la cohesión funcional) es una medida de cuan relacionadas y enfocadas están las responsabilidades de un clase. La utilización de este patrón permite que las clases contengan las responsabilidades estrechamente relacionadas, sin tener que realizar un trabajo excesivo, posibilitando la claridad y facilidad con que se entiende el diseño, simplificando el mantenimiento y las mejoras en funcionalidad, logrando en ocasiones un bajo acoplamiento (37). Se emplea en las clases controladoras y del modelo ya que fueron asignadas responsabilidades a las clases de forma tal que la cohesión siguiera siendo alta, o sea, cada clase se encargará de realizar solamente las funciones que estén en correspondencia con la responsabilidad que posea. Ejemplo: DatNecesidadescapModel.
- **Bajo acoplamiento:** El bajo acoplamiento es un principio que asigna la responsabilidad de controlar el flujo de eventos del sistema a clases específicas, esto permite la centralización de actividades. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (37). Se emplea en la parte del negocio, ya que solo existen las relaciones necesarias entre las clases para mantener un bajo acoplamiento. Ejemplo: GestionaraccionescapController y DatAccionescapModel.

3.3 Patrón arquitectónico

El principal patrón arquitectónico usado es el Modelo Vista Controlador (MVC), este separa en tres capas diferentes a los datos de una aplicación, la interfaz de usuario y la lógica de control, lo que permite construir y probar el modelo, independientemente de la representación visual. 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 el controlador representa la Lógica de negocio.(38)

- ✓ **Modelo:** Esta capa administra el comportamiento y los datos del dominio de la aplicación, responde a requerimientos de información sobre su estado y a instrucciones de cambiar el estado.
- ✓ **Vista:** Esta capa maneja la visualización de la información, es decir que presenta el modelo en un formato adecuado para interactuar, que usualmente es la interfaz de usuario.
- ✓ **Controlador:** Esta capa controla el flujo de datos entre la vista y el modelo; es decir que responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases.

A continuación se muestra un ejemplo de la aplicación de este patrón arquitectónico en el componente desarrollado.

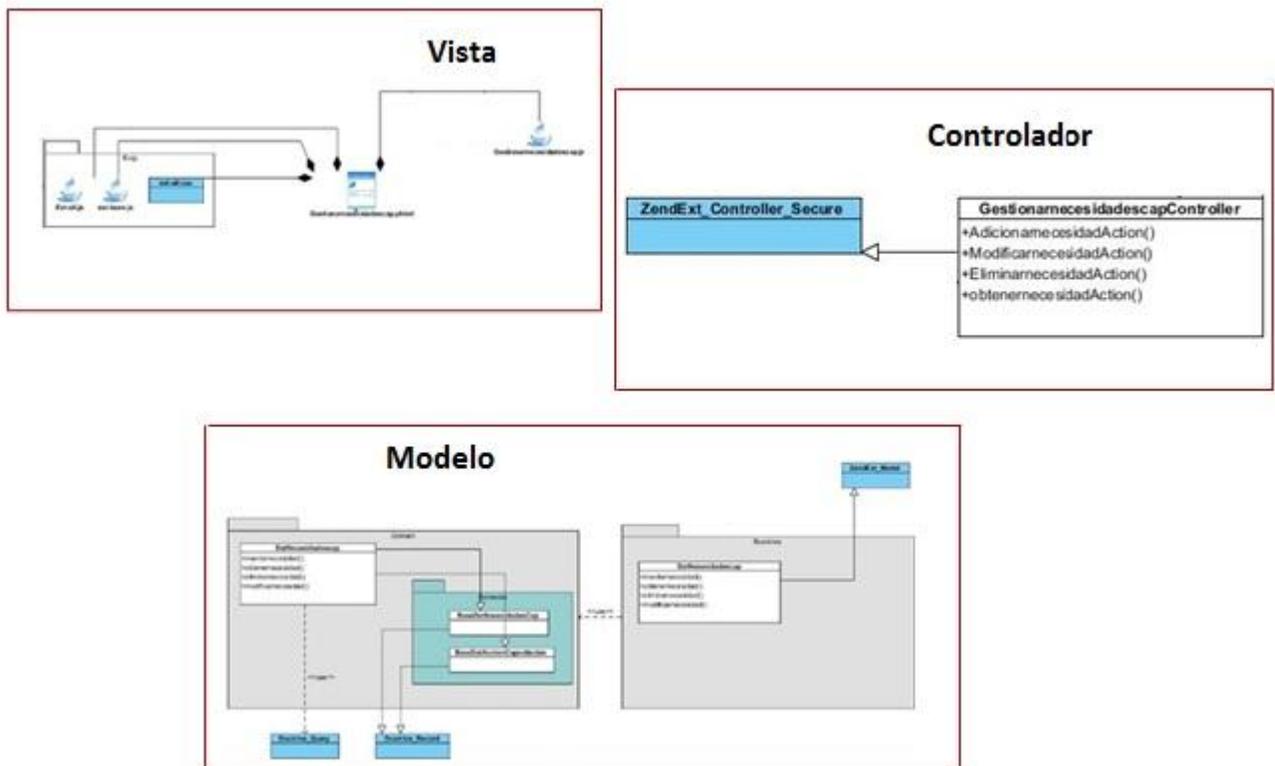


Figura 4. Ejemplo de aplicación del patrón MVC en los diagramas de clase

3.4 Diagramas de clases del diseño

Un diagrama de clases del diseño para UML describe gráficamente las especificaciones de las clases del software y de las interfaces de una aplicación, representa las definiciones de las entidades del software conteniendo de manera general las clases, asociaciones, métodos, dependencias y navegabilidad. (27)

En el diagrama de clases del diseño que se muestra a continuación cada clase cumple una función específica: En las clases “controller” es donde se realiza la lógica del negocio, llama a los métodos de las clases “model” o puede acceder directamente a la “domain”, las clases “model” son las intermediarias entre las clases “controller” y las “domain”, y estas últimas representan las consultas de la base de datos.

Para modelar el diseño del proceso de Capacitación y Desarrollo se generó un total de 11 diagramas de clases del diseño de los cuales se muestra un ejemplo a continuación (Figura 5):

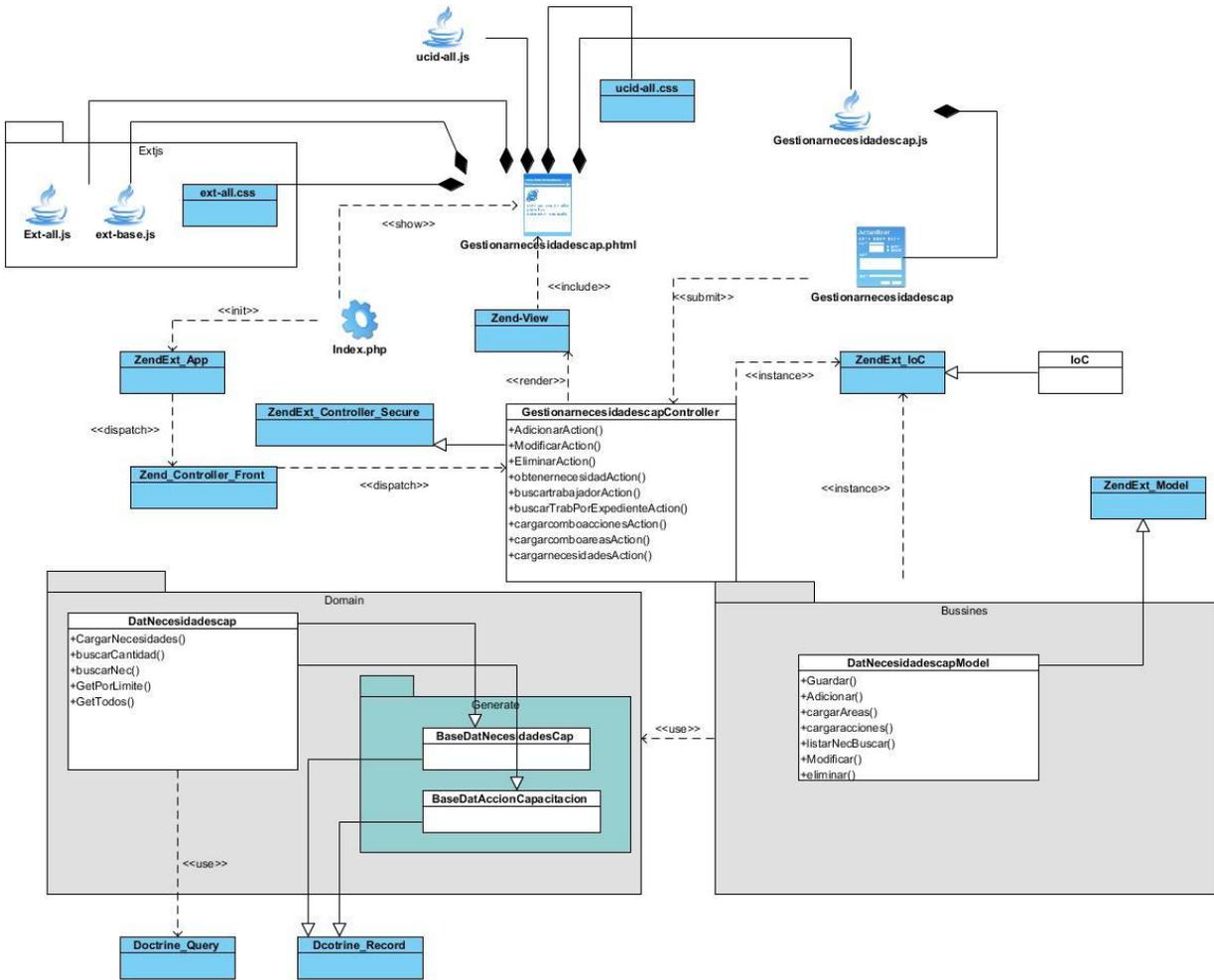


Figura 5. Gestionar necesidades de capacitación

Tabla 2. Descripción del diagrama de clases del diseño. Gestionar acciones de capacitación

Tipo de clase: Nombre de clase	Nombre del método	Descripción del método
Controladora: GestionaraccionescapController	adicionarAccionCapacitacionAction() modificarAccionCapacitacionAction() eliminarAccionCapacitacionAction()	Captura los datos necesarios para luego ejecutar las acciones y se los envía a la clase "model".

	cargarAccionesCapAction()	Encargado de enviar a la clase "domain" una petición para cargar el grid de las acciones de capacitación.
Modelo: GestionaraccionescapModel	adicionar() modificar() eliminar()	Los métodos encargados de llamar a los métodos adicionar (objeto), modificar (objeto), eliminar (objeto) respectivamente en la clase "domain" pasándole el objeto en cuestión.
Negocio: Gestionaraccionescap	eliminar()	Resecciona los datos y modifica la base de datos eliminando la acción de capacitación seleccionada.
	adicionar() modificar()	Resepciona los datos, los clasifica y los guarda en la base de datos, espera la confirmación de que hayan persistido correctamente los mismos y envía en proceso inverso la respuesta de la operación realizada.
	buscarPorIdAccion()	Recepciona y busca los datos de una acción de capacitación por su identificador.

3.5 Diagrama de secuencia

Un diagrama de secuencia muestra una interacción, que representa la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. Este tipo de diagramas es usado para modelar interacción entre objetos en un sistema según UML.

A continuación se muestra el diagrama de secuencia para el requisito Gestionar acciones de capacitación:

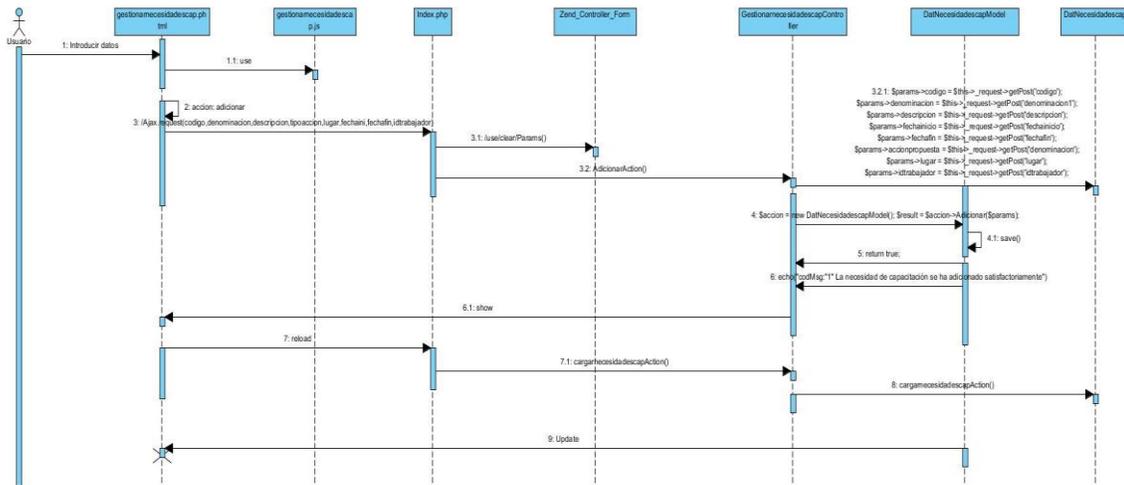


Figura 6. Diagrama de secuencia Adicionar acciones de capacitación

3.6 Diagrama de entidad relación

Un **diagrama o modelo entidad-relación** (E-R "Entity relationship", o, DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades.(21)

En el diagrama entidad-relación del proceso de Capacitación y Desarrollo se obtuvieron un total de veintiuna (21) tablas de las cuales diecisiete (17) son propias del proceso y cuatro (4) son relaciones externas. A continuación se muestra el DER obtenido:

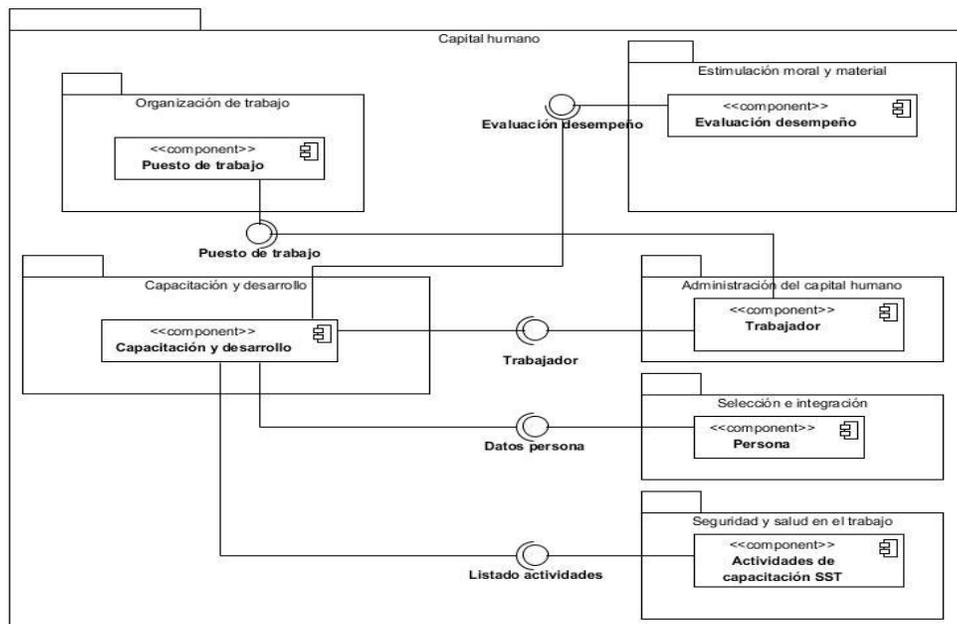


Figura 8. Diagrama de componentes

De manera general el componente Capacitación y Desarrollo necesita de diferentes componentes dentro del subsistema Capital Humano para obtener los datos necesarios para la capacitación y desarrollo de los trabajadores de la empresa.

3.8 Métricas para la validación del diseño propuesto

Las métricas de software son una medida cuantitativa del grado de calidad de los atributos internos del software. Permiten investigar cuán bien están definidas las clases y el sistema, lo cual tiene un impacto directo en el mantenimiento del mismo, tienen como propósito entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo realizado. (33)

Una vez concluido el diseño del componente Capacitación y desarrollo, se llevó a cabo su validación basada en las técnicas de Tamaño operacional de las clases (TOC) de Lorenz y Kidd y Relaciones entre clases (RC) de Chidamber y Kemerer.

A continuación en el próximo epígrafe se desarrollarán las métricas propuestas.

3.8.1. Tamaño operacional de clases

El tamaño operacional de las clases está basado en el número de métodos asignados a una clase. Mediante el cual se calcula el nivel de responsabilidad de los métodos, la complejidad de implementación de los mismos y su reutilización a fin de inspeccionar la efectividad del diseño. A continuación se muestran los resultados obtenidos luego de haber aplicado la métrica TOC.

Atributos que afecta	Modo en que lo afecta
Responsabilidad	El aumento del TOC provoca un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	El aumento del TOC provoca un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC provoca una disminución en el grado de reutilización de la clase.

Tabla 3. Tamaño operacional de clases (TOC)

	Categoría	Criterio
Responsabilidad	Baja	\leq Prom
	Media	Entre Prom. y 2^* Prom
	Alta	$> 2^*$ Prom.
Complejidad implementación	Baja	\leq Prom
	Media	Entre Prom. y 2^* Prom
	Alta	$> 2^*$ Prom.
Reutilización	Baja	\leq Prom
	Media	Entre Prom. y 2^* Prom
	Alta	$> 2^*$ Prom.

Tabla 4. Rango de valores para la evaluación de la métrica TOC

Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
7	Baja	Baja	Alta
9	Media	Media	Media
9	Media	Media	Media
4	Baja	Baja	Alta
10	Media	Media	Media
8	Media	Media	Media
10	Media	Media	Media
9	Media	Media	Media
6	Baja	Baja	Alta
8	Media	Media	Media
9	Media	Media	Media
7	Baja	Baja	Alta
2	Baja	Baja	Alta
5	Baja	Baja	Alta
5	Baja	Baja	Alta
10	Media	Media	Media
11	Media	Media	Media
7	Baja	Baja	Alta
6	Baja	Baja	Alta
8	Media	Media	Media
5	Baja	Baja	Alta
5	Baja	Baja	Alta
5	Baja	Baja	Alta
5	Baja	Baja	Alta
10	Media	Media	Media
12	Media	Media	Media
11	Media	Media	Media
5	Baja	Baja	Alta
10	Media	Media	Media
10	Media	Media	Media

Figura 9. Resultados de la aplicación de la métrica TOC para cada clase del sistema

Las gráficas a continuación muestran los resultados obtenidos para cada uno de los atributos medidos.

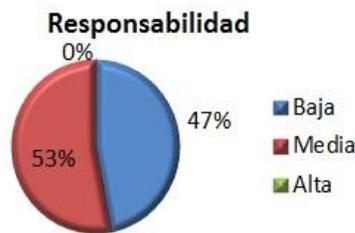


Figura 10. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad



Figura 11. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de implementación



Figura 12. Representación en % de los resultados de la evaluación de la métrica TOC en el atributo Reutilización

Al realizar un análisis de los resultados obtenidos en la aplicación de la métrica TOC en las clases controladoras y del modelo se demuestra que la mayoría de estas clases se encuentran dentro de las categorías media y baja, quedando demostrado de esa forma la elevada reutilización, baja complejidad y responsabilidad que presenta el diseño propuesto, permitiendo una sencilla implementación y una amplia realización de pruebas con mayor facilidad, por lo que los resultados obtenidos según esta métrica son positivos.

3.8.2. Métrica Relaciones entre Clases

Las relaciones entre las clases están dadas por el número de relaciones de uso que tenga una clase con otras. Mediante la cual se calcula el Acoplamiento, la Complejidad, la Reutilización y la Cantidad de pruebas a fin de inspeccionar la efectividad del diseño.

Atributos que afecta:	Modo en que lo afecta:
Acoplamiento	El aumento del RC provoca un aumento del acoplamiento de la clase.
Complejidad del mantenimiento	El aumento del RC provoca un aumento de la complejidad del mantenimiento de la clase.
Reutilización	El aumento del RC provoca una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	El aumento del RC provoca un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Figura 13. Métrica Relaciones entre clases

	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad Mant.	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.
Reutilización	Baja	> 2* Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	<= Prom.
Cantidad de Pruebas	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.

Figura 14. Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC

La imagen a continuación ofrece una selección de las clases del sistema a las que se le aplicó la métrica y los resultados obtenidos para cada atributo evaluado.

Clase	Cantidad de Relaciones de U	Acoplamiento	Complejidad Man	Reutilización	Cantidad de Pruebas
GestionaraccionescapController	3	Alto	Media	Media	Media
GestionarcursoController	2	Medio	Baja	Alta	Baja
GestionarnecesidadescapController	2	Medio	Baja	Alta	Baja
GestionarinstructorController	2	Medio	Baja	Alta	Baja
GestionarplangeneralController	3	Alto	Media	Media	Media
GestionarplancapController	6	Alto	Alta	Baja	Alta
GestionarplanpreinversionesContro	2	Medio	Baja	Alta	Baja
GestionarplansstController	2	Medio	Baja	Alta	Baja
GestionartrabincorporadosController	4	Alto	Media	Media	Media
RegistrarpresupuestoController	1	Bajo	Baja	Alta	Baja

Figura 15. Resultados de la aplicación de la métrica RC para cada clase del sistema.



Figura 16. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento



Figura 17. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Complejidad de mantenimiento

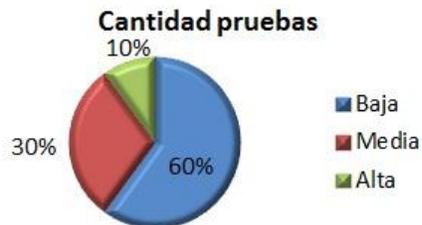


Figura 18. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Reutilización



Figura 19. Representación en % de los resultados de la evaluación de la métrica RC en el atributo Cantidad de pruebas

Luego de aplicar la métrica de diseño RC y obtenidos los resultados, se puede concluir que el diseño propuesto tiene una calidad aceptable teniendo en cuenta que la mayoría de las clases empleadas poseen menos de 3 dependencias de otras clases, lo que lleva a evaluaciones positivas de los atributos de calidad involucrados (acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización). Favoreciendo de esta manera la reutilización de las clases así como la modificación e implantación del diseño.

3.9 Conclusiones parciales

La validación del diseño propuesto arrojó resultados favorables, demostrándose así la calidad del mismo, permitiendo identificar las principales clases de la solución, las relaciones existentes entre ellas y sus métodos asociados, sentándose así las bases para la fase de implementación del componente Capacitación y Desarrollo, haciendo más fácil de esta manera la reutilización y modificación de las clases.

4. IMPLEMENTACIÓN Y PRUEBA

4.1. Introducción

En el presente capítulo se describe la implementación del componente Capacitación y desarrollo, además se especifican las pruebas de software que son usadas para la validación y comprobación de la calidad del sistema desarrollado.

4.2. Implementación del sistema

El marco del proyecto Cedrux tiene definidos estándares de codificación que permiten una mejor integración de las líneas de producción y establecen pautas que conllevan a lograr que el código sea legible y reutilizable, permitiendo un mejor entendimiento por parte del equipo de desarrollo y soporte. (30)

Dentro de los estándares utilizados se encuentran:

4.2.1. PascalCasing

El estándar PascalCasing establece que los nombres de clases están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula. Poner algo de la nomenclatura utilizada

Nomenclatura de clases según su tipo:

- **Controllers:** clases controladoras del negocio.

El nombre de la clase controladora debe estar estructurado por el nombre propio de la misma en mayúsculas seguido por la palabra Controller y heredar siempre de la súper clase del frameworkZendExt_Controller_Secure. Ejemplo: GestionarnecesidadescapController.

Clases modelos

- **Business:** Clases modelo del negocio.

Las clases modelo tendrán por identificador el nombre de la tabla en la que trabajan seguido por la palabra Model y heredarán de la clase del frameworkZend_Ext llamada ZendExt_Model, pueden incluir los prefijos Dat o Nom para diferenciar entre sus usos. Ejemplo: DatNecesidadescapModel.

- **Domain:** clases entidades del dominio.

Los archivos situados en el domain tienen el mismo nombre de las tablas que representan, definiéndolas como clases php, pueden incluir los prefijos Dat o Nom para diferenciar entre sus usos. Ejemplo: DatNecesidadescap.

- **Generated:** Clases bases del dominio

Las clases que se encuentran dentro de Generated comienza su nombre con la palabra: “Base”, seguido del nombre de la tabla en la Base de Datos. Ejemplo: BaseDatNecesidadescap.

- **Services:** Clases que ofrecen los servicios de los componentes.

Estas clases, de acuerdo con las operaciones que realizan y prestaciones que brindan, se definen con calificativos sugerentes, agregándoles la terminación Service. Ejemplo: BuscarPersonaService.

4.2.2. CamelCasing

El estándar CamelCasing es parecido al PascalCasing con la particularidad de que la letra inicial del identificador no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones.

Nomenclatura de las funciones

El identificativo a emplear para las funciones o métodos se escribe con la primera palabra en minúscula utilizando la notación CamelCasing y nombres que deduzcan su propósito. Ejemplo: adicionar.

Los denominadores de las acciones de las clases controladoras tienen la peculiaridad de ir seguidos por la palabra “Action”. Ejemplo: adicionarNecesidadesCapacitacionAction.

4.4. Funcionalidades implementadas

En la **Figura 21** se muestra la interfaz Gestionar Plan individual, que ofrece las opciones de Adicionar, Modificar, Eliminar el plan individual de cada trabajador, así como Listar las necesidades reflejadas en cada plan, para ver el resto de las interfaces puede consultar los anexos.

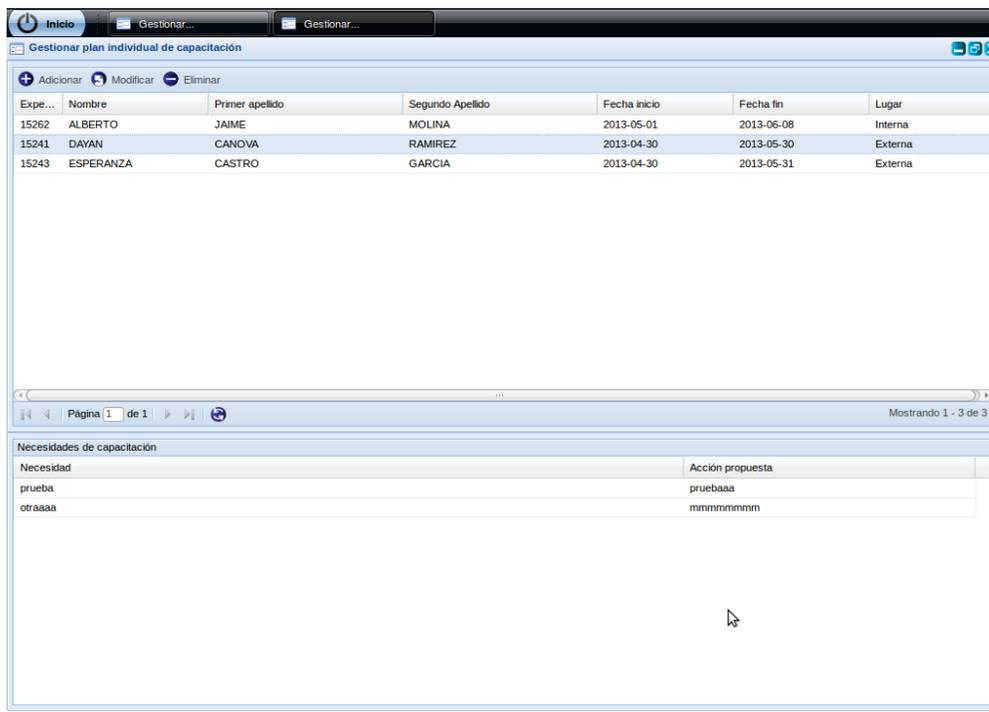


Figura 20. Interfaz Gestionar Plan individual

4.5. Integración entre componentes

La integración entre los diferentes módulos o componentes se basa en el patrón Inversión de Control (IoC) y se realiza a través de un componente incluido en el Zend_Framework, que permite operar sobre distintos esquemas en la base de datos realizando las transacciones adecuadas. En este se incluye el fichero ioc.xml que contiene la ubicación de cada uno de los componentes y los servicios que ofrecen las clases services correspondientes.

Servicios que utiliza el componente como parte de la integración:

- **BuscarTrabajador:** Servicio que brinda el componente Trabajador del subsistema Capital Humano que le permite al componente Capacitación y desarrollo obtener los datos del trabajador mediante su número de expediente, nombre, primer y segundo apellido.

- DameAreas: Servicio que brinda el subsistema Estructura y Composición que le permite al componente Capacitación y desarrollo obtener los datos del área mediante el identificador.
- ListarActividadesCapacitacion: Servicio que brinda el componente Seguridad y salud en el trabajo que le permite al componente Capacitación y desarrollo obtener los datos de las actividades de capacitación relacionadas con seguridad y salud en el trabajo para elaborar el plan de capacitación de seguridad y salud en el trabajo.

4.6. Pruebas de software

Las pruebas de software son una tarea en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y se confecciona una evaluación de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (33)

4.6.1 Prueba de Caja Blanca

La prueba de caja blanca también conocida como prueba de caja de cristal, se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Para la solución desarrollada la prueba de Caja Blanca aplicada fue la del camino básico. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico de caminos de ejecución. (19)

Para realizar la prueba del camino básico es necesario dibujar a partir del diseño o del código fuente en grafo de flujo asociado y calcular su complejidad ciclomática. A continuación se muestra se muestra el código del método AdicionarAction encargado de adicionar las necesidades de capacitación de cada trabajador.

```
function adicionarAccionCapacitacionAction() {
    $accion = new DatNecesidadescapModel();1
    $params = new DatNecesidadescap();1

    $params->codigo = $this->_request->getPost('codigo');1
    $params->denominacion1 = $this->_request->getPost('denominacion1');1
    $params->descripcion = $this->_request->getPost('descripcion');1
    $params->accionpropuesta = $this->_request->getPost('denominacion');1
    $params->fechainicio = $this->_request->getPost('fechainicio');1
    $params->fechafin = $this->_request->getPost('fechafin');1
    $params->lugar = $this->_request->getPost('lugar');1
    $params->idtrabajador = $this->_request->getPost('idtrabajador');1

    $existe= $accion->buscarCodigo($params->codigo);2
    $existe2= $accion->buscarDemon($params->denominacion1);2
    if($existe)3
        echo 3;4
    elseif($existe2)5
        echo 4;6
    else
    $result = $accion->Adicionar($params);7
    echo $result;8
}
```

Para el cálculo de la complejidad ciclomática es necesario representar el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones, quedando como se muestra en la figura.

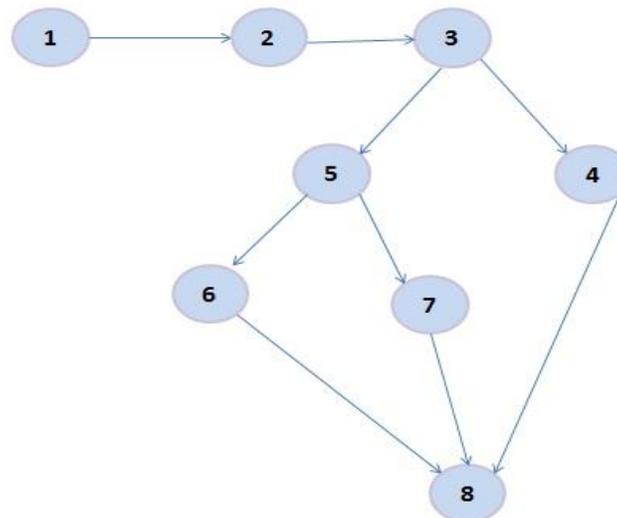


Figura 21. Grafo de flujo asociado al método

4.6.1.1 Cálculo de la complejidad ciclomática a partir de un segmento de código.

La complejidad ciclomática es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y propone un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

La complejidad ciclomática de un código se puede calcular de tres maneras diferentes.

Para calcular la complejidad ciclomática del método `listarplanAction` fueron utilizadas estas tres formas para lograr una amplia verificación de los resultados.

Las fórmulas para realizar dicho cálculo son:

$$V(G) = (A - N) + 2$$

$$V(G) = (9-8) + 2$$

$$V(G) = 3$$

Siendo A la cantidad total de aristas del grafo y N la cantidad de nodos.

1. $V(G) = P + 1$

$$V(G) = 2+1$$

$$V(G) = 3$$

Siendo P la cantidad de nodos predicado (son aquellos de los cuales parten dos o más aristas)

2. $V(G) = R$

$$V(G) = 3$$

Siendo R la cantidad de regiones que posee el grafo.

En cada una de las fórmulas $V(G)$ representa el valor del cálculo. Según los resultados obtenidos en cada uno de estos cálculos se puede concluir que la complejidad ciclomática del código analizado es 3, determinándose a su vez que existen tres caminos posibles por donde puede

circular el flujo y que esta misma cantidad representa el límite superior de casos de prueba que se le pueden aplicar a dicho código.

A continuación se muestran los caminos básicos por donde puede circular el flujo.

Camino básico # 1: 1-2-3-4-8

Camino básico # 2: 1-2-3-5-6-8

Camino básico # 3: 1-2-3-5-7-8

4.6.1.2 Casos de prueba para el camino básico # 1

Camino básico # 1: 1-2-3-5-6-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$existe es true, el código insertado ya existe, por lo que no se puede adicionar la necesidad de capacitación.

Entrada: \$params->codigo

Resultados esperados: Se lanza el mensaje “El código insertado ya existe. Por favor elija otro.”.

4.6.1.3 Caso de prueba para el camino básico # 2

Camino básico # 1: 1-2-3-4-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro \$existe2 es true, la denominación insertada ya existe, por lo que no se puede adicionar la necesidad de capacitación.

Entrada: \$params->denominacion1

Resultados esperados: Se lanza el mensaje “La denominación insertada ya existe. Por favor elija otra.”.

4.6.1.4 Caso de prueba para el camino básico # 3

Camino básico # 1: 1-2-3-4-8

Descripción: El dato de entrada cumplirá con el siguiente requisito:

Los parámetros \$existe, \$existe2 son false, se puede realizar la operación de Adicionar la necesidad de capacitación.

Entrada: \$params=[\$params[0]->codigo=>12,\$params[0]->denominacion1=>Formar nuevos trabajadores para ocupar plazas vacantes, \$params[0]->descripcion=>Preparar a los trabajadores para ocupar alguna plaza vacante dentro de la empresa , \$params[0]->accionpropuesta=>Preparar trabajador para ocupar un nuevo puesto de trabajo, \$params[0]->fechainicio=>2013-06-05, \$params[0]->fechafin=>2013-06-30,\$params[0]->lugar=>Externa,\$params[0]->idtrabajador=>900001238]

Resultados esperados: Se lanza el mensaje “La necesidad de capacitación se ha adicionado satisfactoriamente.”.

4.6.2 Pruebas de Caja Negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software y se centra principalmente en los requisitos funcionales del software. (19)

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

De las técnicas antes mencionadas se seleccionó la partición de equivalencia la cual permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Para la aplicación de esta técnica se realizan los diseños de casos de prueba los cuales se basan en una evaluación de las clases de equivalencia para una condición de entrada.

Para validar el componente atendiendo a la técnica partición de equivalencia Se realizaron 45 diseños de casos de prueba. Se decidió realizar tres iteraciones donde los resultados obtenidos en la primera iteración fueron:

Iteración	Cantidad de NC	NC Significativas
1	20	2

A continuación se especifica el caso de prueba para el requisito “Adicionar acción de capacitación”.

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el subsistema Capital Humano /Capacitación y desarrollo/ Gestionar acciones capacitación.

Requisitos a probar

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar acción	El sistema debe permitir adicionar una acción.	EP 1.1: Adicionar acción introduciendo datos válidos.	<ul style="list-style-type: none"> ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos de la acción correctamente. ➤ Se presiona el botón Aceptar. ➤ Se muestra un mensaje de información. ➤ Se presiona el botón Aceptar.
		EP 1.2: Adicionar la acción introduciendo datos válidos presionando el botón Aplicar .	<ul style="list-style-type: none"> ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos de la acción correctamente. ➤ Se presiona el botón Aplicar. ➤ Se muestra un mensaje de información. ➤ Se presiona el botón Aceptar. ➤ Se limpian los campos. ➤ Se mantiene abierta la interfaz.
		EP 1.3: Adicionar acción introduciendo datos inválidos.	<ul style="list-style-type: none"> ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos inválidos de la acción. ➤ Se presiona el botón Aceptar. ➤ Se muestra un mensaje informando del error.

EP Adicionar acción dejando campos vacíos.	1.4:	<ul style="list-style-type: none"> ➤ Se presiona el botón Adicionar. ➤ Se introducen los datos dejando algún campo en blanco. ➤ Se presiona el botón Aceptar. ➤ Se muestra un mensaje informando del error.
EP Cancelar.	1.5:	<ul style="list-style-type: none"> ✓ Se presiona el botón Adicionar. ✓ Se introducen o no los datos de la acción. ✓ Se presiona el botón Cancelar.

Descripción de variable

No	Nombre de campo	Tipo	Válido	Inválido	Inválido	Inválido	Inválido
1	Código	Campo texto (3 caracteres).	de Números	Letras, caracteres especiales.	Vacío.	NA	NA
2	Denominación	Campo texto.	de Letras.	Números, caracteres especiales.	Vacío	NA	NA
3	Descripción.	Campo texto.	de Letras.	Números, caracteres especiales.	Vacío	NA	NA
4	Objetivo	Campo texto.	de Letras.	Números, caracteres	Vacío	NA	NA
5	Modo formación.	Elegir editable).	(No NA	NA	NA	NA	NA
6	Tipo acción	Elegir editable).	(No NA	NA	NA	NA	NA
7	Fecha inicio	Campo fecha(No editable)	de NA	NA	NA	NA	NA
8	Fecha fin	Campo fecha(No editable)	de NA	NA	NA	NA	NA

Juegos de datos a probar

Id del escenario	Escenario	Código	Denominación	Descripción	Objetivo	Modo formación	Tipo acción	Fecha inicio	Fecha fin	Respuesta del sistema	Resultado de la prueba
------------------	-----------	--------	--------------	-------------	----------	----------------	-------------	--------------	-----------	-----------------------	------------------------

EP 1.1	Adicionar acción introduciendo datos válidos.	V(123)	V(Entrenamiento en el puesto de trabajo)	V(Entrenar al trabajador para su puesto de trabajo)	V(Preparación de Cuadros)	V(Cursos de capacitación)	V(Planificada)	V(2013-06-5)	V(2013-06-30)	Se adiciona la acción de capacitación. El sistema muestra el siguiente mensaje: "La acción de capacitación se ha adicionado satisfactoriamente."	NA
EP 1.2	Adicionar la acción introduciendo datos válidos presionando el botón Aplicar .	V(56)	V(Preparación para inversiones)	V(Inversión)	V(Aseguramiento del personal para planes de inversión)	V(Cursos de perfeccionamiento o promoción)	V(No planificada)	V(2013-10-05)	V(2013-10-05)	Se adiciona la acción de capacitación. El sistema muestra el siguiente mensaje: "La acción de capacitación se ha adicionado satisfactoriamente."	NA
EP:1.3	Adicionar acción introduciendo datos inválidos.	I(Letras,/,*,@,#,%^!&())	V(Curso perfeccionamiento)	V(Perfeccionamiento)	V(Preparación política)	V(Cursos de perfeccionamiento o promoción)	V(Planificada)	V(2013-08-09)	V(2013-08-09)	El sistema muestra el siguiente mensaje: "Por favor verifique"	NA

V(89)	I(1/*@#\$\$&(!))	V(Perfeccionamiento)	V(Preparación política)	V(Cursos de perfeccionamiento o promoción)	V(Planificada)	V(2013-08-09)	V(2013-08-09)	V(2013-08-09)	nuevamente que hay campo(s) con valor(es) incorrecto(s)". Se muestra marcado en color rojo el campo requerido.
V(89)	V(Curso perfeccionamiento)	I(1/*@#\$\$&(!))	V(Preparación política)	V(Cursos de perfeccionamiento o promoción)	V(Planificada)	V(2013-08-09)	V(2013-08-09)	V(2013-08-09)	
V(89)	V(Curso perfeccionamiento)	V(Perfeccionamiento)	I(1/*@#\$\$&(!))	V(Cursos de perfeccionamiento o promoción)	V(Planificada)	V(2013-08-09)	V(2013-08-09)	V(2013-08-09)	
V(89)	V(Curso perfeccionamiento)	V(Perfeccionamiento)	V(Preparación política)	I(1/*@#\$\$&(!))	V(Planificada)	V(2013-08-09)	V(2013-08-09)	V(2013-08-09)	

EP: 1.4	Ad ici on ar la ac ció n de ja nd o ca m po s va	I(Va cí o)	V(Curso perfeccion amiento)	V(Perfec cionamie nto)	V(Prepara ción política)	V(Cursos de perfeccion amiento o promoció n)	V(Plani ficada)	V(201 3-08- 09)	V(2013- 0815)	El sistem a muestr a el siguien te mensa je: "Por favor verifiq ue nueva mente que hay	NA
		V(8 8)	I(Vacío)	V(Perfecci onamiento)	V(Preparaci ón política)	V(Curs os de perfecc ionami ento o promoc ión)	V(Pla nifica da)	V(2 013 -08- 09)	V(20 13- 0815)		
		V(8 8)	V(Curso perfecciona miento)	I(Vacío)	V(Prepa ración política)	V(Cursos de perfeccion amiento o promoción)	V(Planifi cada)	V(2 013 -08- 09)	V(2 013- 081 5)		
		V(88)	V(Curso perfecciona miento)	V(Perfeccion amiento)	I(Vacío)	V(Cursos de perfeccion amiento o promoción)	V(Planif icada)	V(20 13- 08- 09)	V(201 3-08- 15)		
		V(88)	V(Curso perfecciona miento)	V(Perfeccion amiento)	V(Prepa ración política)	I(Vacío)	V(Planif icada)	V(20 13- 08- 09)	V(201 3-08- 15)		

EP: 1.5	Canc elar. NA	NA	Se cancel a la operaci ón y se cierra la ventan a.	N A							
------------	---------------------	----	----	----	----	----	----	----	----	--	--------

4.7 Conclusiones parciales

Se implementaron todas las clases diseñadas logrando un módulo de Capacitación y Desarrollo funcional y que se adecua a las normas de Cedrux. Se especificaron los estándares de codificación propuestos por el CEIGE para la implementación del sistema. Se describieron y aplicaron las pruebas de caja blanca y caja negra, para evaluar el componente desarrollado, las cuales mostraron que el mismo contaba con un adecuado funcionamiento, demostrando así el cumplimiento de las necesidades del cliente y la calidad requerida en el mismo.

CONCLUSIONES

- Se elaboró el marco teórico de la investigación donde se realizó un estudio de varios sistemas de gestión que incluyen el proceso de Capacitación y desarrollo de los trabajadores en las empresas analizando sus funcionalidades, lo cual arrojó que era más factible desarrollar un solución que cumpliera con las normas establecidas para la formación del personal en las entidades laborales en Cuba, así como con el paradigma de independencia tecnológica por la que aboga el país. También se describieron las herramientas y tecnologías a usar definidas por el centro CEIGE y se hizo un estudio que posibilitó entender y avalar la selección de las mismas.
- Se realizó el análisis y diseño de la solución propuesta, donde se generaron varios artefactos a partir de la captura y especificación de los requisitos, lo cual permitió obtener los diagramas de clase del diseño para cada escenario, los diagramas de secuencias, así como los diseños de casos de prueba por cada requisito para un total de 45 diseños y el modelo conceptual con un total de 15 tablas.
- Se realizó la validación del diseño aplicando las métricas Tamaño operacional de las clases (TOC) de Lorenz y Kidd y Relaciones entre clases (RC) de Chidamber y Kemerer, donde los resultados arrojados demuestran la presencia de valores positivos en los indicadores de calidad medidos.
- Se desarrolló el componente Capacitación y Desarrollo para el Sistema Integral de Gestión Cedrux a partir del diseño propuesto, el cual permite la gestión de este proceso como establecen las normas vigentes en el país para las empresas cubanas.
- Se realizaron las pruebas de caja blanca y caja negra con el objetivo de minimizar la ocurrencia de errores del sistema antes de su implantación en las entidades nacionales, las cuales mostraron que el sistema contaba con un correcto funcionamiento, manifestando así el cumplimiento de las necesidades del cliente y la calidad requerida en el mismo.

RECOMENDACIONES

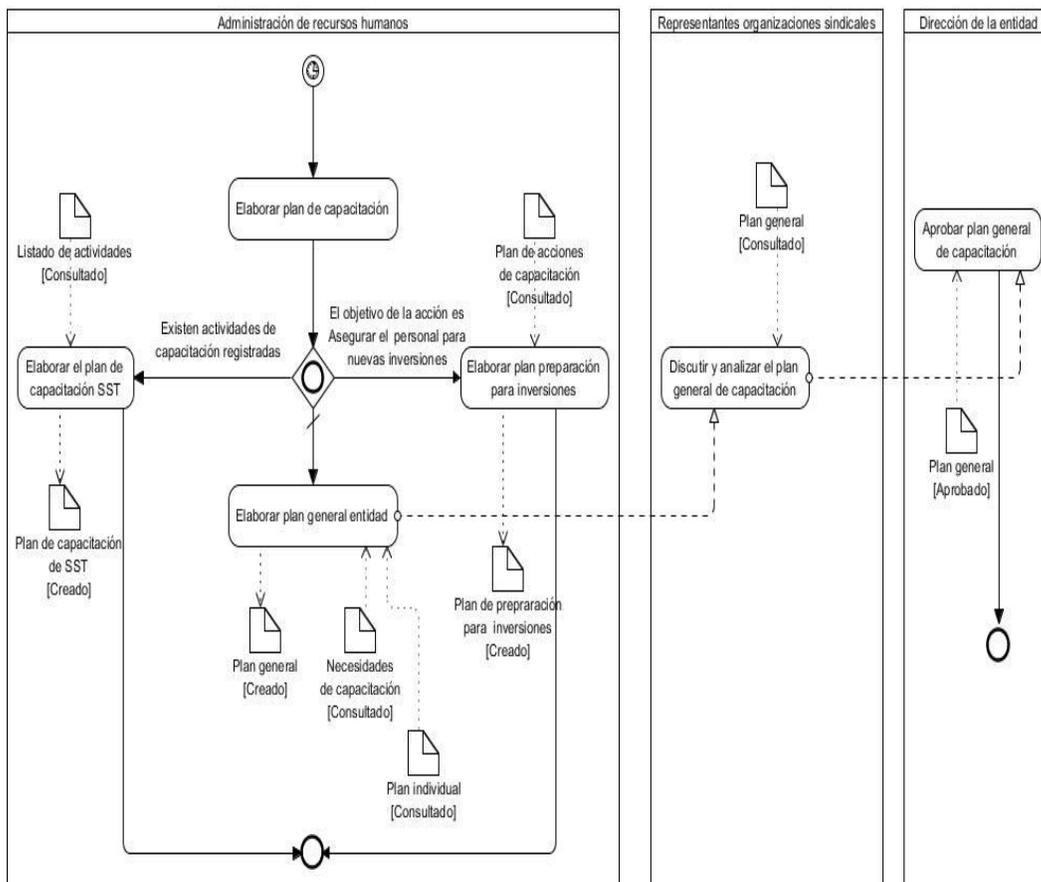
- Se recomienda que se lleven a cabo futuras versiones del componente Capacitación y Desarrollo donde se agregue una funcionalidad para evaluar el impacto logrado en el proceso de producción y servicios de la entidad.
- Se recomienda que se incluya en el proceso de Determinación de las necesidades individuales de capacitación la definición de las necesidades incluyendo las competencias laborales del trabajador.
- Realizar el despliegue del componente Capacitación y Desarrollo como parte del subsistema de Capital humano del sistema Cedrux.

REFERENCIAS

1. **G. Martínez, Larissa** . *Administración de recursos humanos*. 2004. págs. 8-9.
2. **Morales, Aylín, Caballero, Juan Alberto y Abrante, Yunet Suárez**. *COMPONENTE DE SOFTWARE PARA GESTIONAR COMPETENCIAS LABORALES MEDIANTE CEDRUX*. 2011.
3. **CEIGE**. *CEIGE-Modelo de Desarrollo de Software*. La Habana : s.n., 2012.
4. **Foundation, The Apache Software**. *www.apache.org*. [En línea] 2008. [Citado el: 27 de Noviembre de 2012.] <http://projects.apache.org/indexes/quick.html>.
5. **López, Alejandro Cadavid**. Mozilla Firefox, el navegador web del momento. [En línea] 2009. [Citado el: 27 de Noviembre de 2012.] <http://www.maestrosdelweb.com/editorial/firefox/>.
6. **Tedeschi, Nicolás**. ¿Qué es un Patrón de Diseño? [En línea] 2006. [Citado el: 27 de Noviembre de 2012.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
7. <http://elvex.ugr.es>. <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>. [En línea] [Citado el: 6 de 3 de 2013.]
8. **Entidades, Centro de Informatización de Gestión de**. *Modelo de desarrollo de software*. 2012.
9. **Javier Nieto Diego, Pablo Ramos Fernández**. *El Patrón Fachada*. s.l. : Departamento de Informática y Automática.
10. **Gutiérrez, José Luis Delgado**. *La importancia de la capacitación en los recursos humanos los recursos humanos. Hitos de Ciencias Económico* . 2000.
11. **Ponce, Ma. Eugenia López**. “DEFINICIÓN Y COMPRENSIÓN DE PROCESO“. Agosto de 2012.
12. (SAP Global. [cited 2010 Febrero]; Available from: sap.com).
13. (GNUEnterprise.[cited 2010 Febrero]; Available from: <http://www.gnuenterprise.org/packages/hr.php>).
14. (The Short List for Business Software. [cited 2010 Febrero]; Available from: 2020software.com).
15. **Aylin Morales Ortiz, Yunet Abrante**. *COMPONENTE DE SOFTWARE PARA GESTIONAR COMPETENCIAS LABORALES MEDIANTE CEDRUX*.
16. **Entidades, Centro de Informatización de Gestión de**. *Modelo de desarrollo de software* . 2012.
17. **Contreras, Rita del Carmen Castillo**. *Desarrollo del capital humano en las organizaciones*.
18. *RESOLUCION 29 ORG. Y CONTROL Y DESA. CAPACITACION* . Habana : s.n., 2006.
19. **Eva.uci.cu**. *Material de caja blanca y caja negra.pdf*.
20. *Exact Synergy*. 2009.
21. **Pressman**. *Ingeniería del Software: Un enfoque práctico* . . 2005.
22. **Muchinsky, P.**(2002). *Psicología aplicada al trabajador*. Sexta edición.Mexico: Thomson Learning.
23. **Larman, Craig**. 1999. *UML y patrones, Introducción al análisis y diseño orientado a objeto*. 1999.

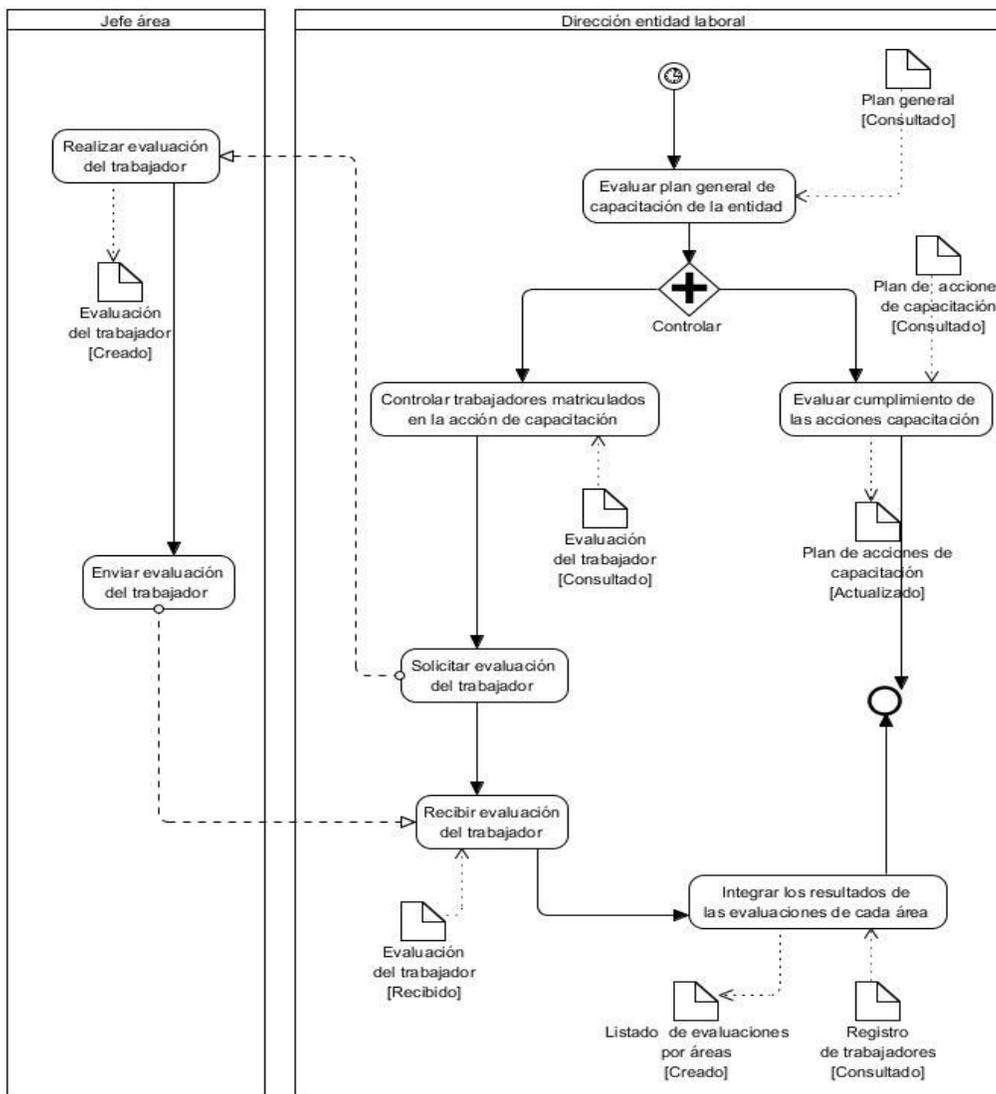
24. **Schmidt, Douglas.***PATTERN-ORIENTED SOFTWARE ARCHITECTURE VOLUME 2: Patterns for Concurrent and Networked Objects.* 2000.
25. **Gil, Manuel Torres.** Portal de la Universidad de Almería. [En línea] [Citado el: 2 de marzo de 2013.] <http://indalog.ual.es/mtorres/LP/Prueba.pdf...>
26. **Ivar Jacobson, Grady Booch, James Rumbaugh.** 2000. *Proceso Unificado de Desarrollo de Software.* 2000.
27. **Visconti, Marcello y Astudillo, Hernán.** 2010. *Fundamentos de Ingeniería de Software.* Departamento de Informática, Universidad Técnica Federico de Santa María : s.n., 2010.
28. **Analítica.** *Manual de diagramación de procesos bajo estándar BPMN.*
29. Herramientas CASE. [En línea] [Citado el: 8 de diciembre de 2012.]
30. **Díez, A.** 2001. *IRqA y el desarrollo de proyectos: Experiencias Prácticas. I Jornadas de Ingeniería de Requisitos.* Sevilla.España : s.n., 2001.
31. Proceso de Desarrollo y Gestión de Proyectos de Software. [En línea] [Citado el: 10 de diciembre de 2012.]
32. **Becker, Gary Stanley.** 2002. *La inversión en talento como valor de futuro.* EEUU : Rev. Capital Humano, 2002, Vol. 153.
33. **Rearte, Ing. Emilio.** Base de Datos y UML. 2002
34. Buenas tareas [En línea] [Citado el: 6 de junio de 2013.]
<http://www.buenastareas.com/ensayos/Area-De-Recursos-Humanos/4970874.html>
35. Buenas tareas[En línea] [Citado el: 6 de junio de 2013.]
<http://www.buenastareas.com/ensayos/Desarrollo-Del-Capital-Humano/968643.html>
36. **Nora Koch, María José Escalona.**Ingeniería de Requisitos en Aplicaciones para la Web. Sevilla : s.n., 2002.
37. **Tedeschi, Nicolás.**¿Qué es un Patrón de Diseño? 2006.
38. **Ochoa, Prof. Sergio.**Diseño y Arquitectura. 2005.
39. **Acha, Gustavo** [En línea] [Citado el: 10 de junio de 2013.]
<http://www.materiabiz.com/mbz/capitalhumano/nota.vsp?nid=28252>
40. Ecured [En línea] [Citado el: 10 de mayo de 2013.]
www.ecured.cu/index.php/Validación_de_Requisitos

ANEXOS



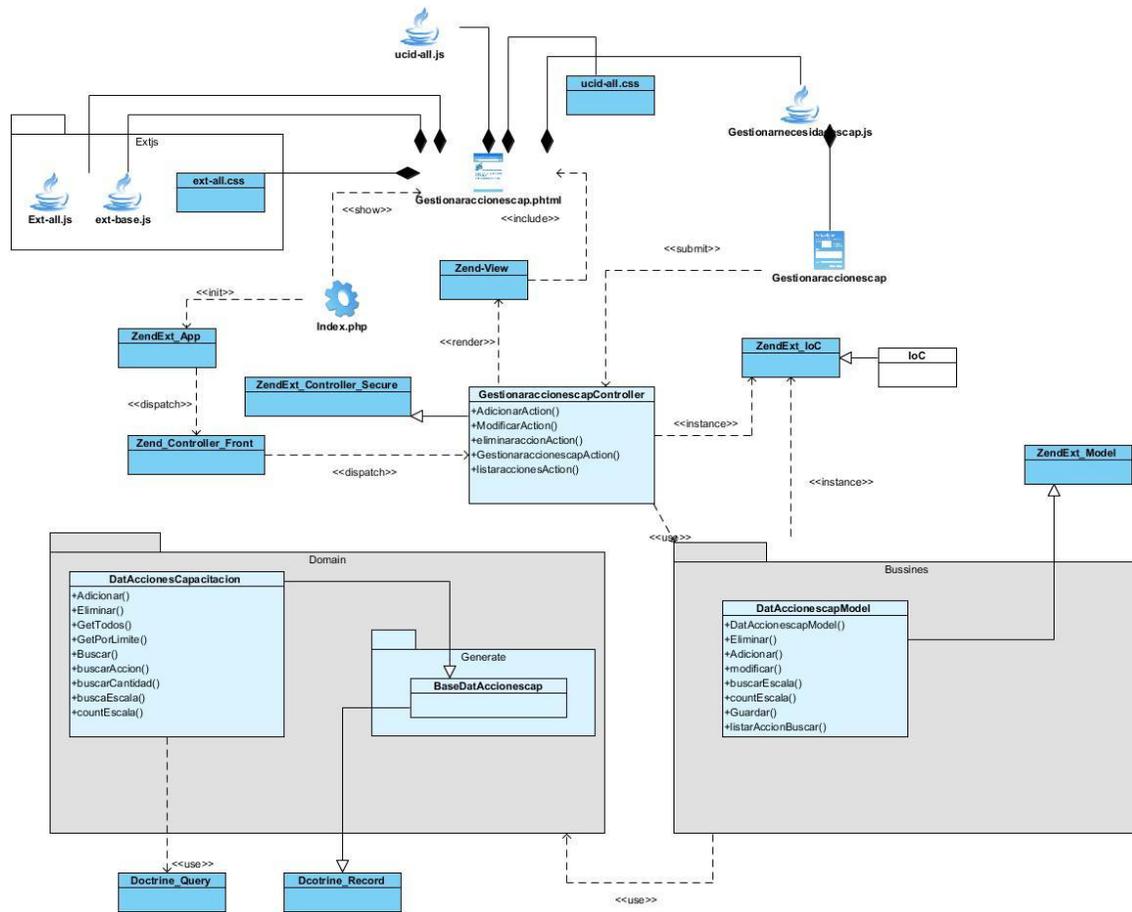
Anexo 1. Diagrama del proceso Elaborar el Plan de Capacitación y desarrollo de la entidad

Mediante el proceso Plan de Capacitación y desarrollo de la entidad se garantiza la elaboración del plan general de la entidad, el plan de adiestramiento laboral a recién graduados, plan de preparación para inversiones y los planes de capacitación relacionados con seguridad y salud en el trabajo.



Anexo 2. Diagrama del proceso Control de la ejecución del Plan de Capacitación y desarrollo

El proceso de **Control de la ejecución del Plan de Capacitación y Desarrollo** es llevado por la dirección de la entidad laboral con el propósito de verificar el cumplimiento de las acciones de capacitación propuestas luego de haber obtenido las necesidades de capacitación existentes dentro de la empresa, evaluar el impacto del plan de capacitación aprobado y tomar decisiones y acciones para mejorar el proceso de capacitación y desarrollo de los recursos humanos y asegurar el retorno de la inversión que realizan en esta actividad.



Anexo 3. Diagrama de clases del diseño Gestionar acciones de capacitación

GLOSARIO

Capital Humano: Cuantificación y valoración de los recursos humanos. Valor de las habilidades, capacidades, experiencias y conocimientos de las personas que integran una organización.

Componente: Elemento de un sistema software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas.

Entidad: Empresa, unidad presupuestada u otro tipo de organización similar con una gestión económica, financiera, organizativa, técnica, productiva, comercial, laboral y contractual, con autonomía controlada, en cumplimiento de lo establecido por el Gobierno.

Gestión: Es el conjunto de diligencias que se realizan para desarrollar un proceso o para lograr un producto determinado. Es también la dirección o administración de una empresa o de un negocio.

Proceso: Según lo establecido en el apartado 3.4.1 de la NC ISO 9000:2005, el proceso es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados.

Subsistema: Cada uno de los componentes principales de un sistema que este dividido en componentes. Cada subsistema abarca aspectos del sistema que comparten alguna propiedad común.

Trabajador: Es aquella persona que ocupa un puesto de trabajo en la entidad y desempeña una labor determinada por la cual recibe haberes o salarios.