

Universidad de Ciencias Informáticas

Facultad 1



Vista de la arquitectura de sistema y datos del subsistema
Capital Humano del proyecto ERP-Cuba.

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

Autor:

Jorge Luis Naranjo Hurtado

Tutor:

Ing. Ariel Torres Gálvez

Cotutor:

Ing. Alain Fernández Deronceré

Ciudad de La Habana, Cuba, Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo de diploma y reconozco 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 _____.

Jorge Luis Naranjo Hurtado

Ing. Ariel Torres Gálvez

Firma del Autor

Firma del Tutor

Pensamiento

Si tu intención es describir la verdad, hazlo con sencillez y la elegancia déjasela al sastre.

Albert Einstein

Agradecimientos:

Agradecerle ante todo a mis queridos padres: Jorge y Elizabeth por creer siempre en mí, por todo el apoyo y amor que me han brindado en toda mi vida y por haber sabido guiarme por el camino correcto y ser mis ejemplos.

A mi hermano Geobel por formar parte de mí en todo momento y ayudarme en cualquier situación.

A mi novia Migdailis por haberme apoyado mucho este año en la tesis y por el amor y cariño que me brinda.

A su familia que se ha preocupado tanto por mí: a Luis Enrique su papá, Magalis su mamá, sus abuelos Melva y Juan, a su tío German, a Migdalia su tía y su abuela Juana.

A mis vecinos Isabel, Manuela, Pelao y Pablo.

A Alexaide que pregunta por mí con la botella en la mano.

A Titin y Norelvis por apoyarme tanto en mi carrera y esperar este triunfo con anhelo.

A mis tíos Rogelio y Michelo por preocuparse tanto por mí.

A mi tía Rubi que me quiere mucho y me aconseja por el bien a pesar de estar tan distante de mí.

A mi amigo Luisito y su mamá Deisis que puedo contar con ellos para lo que los necesite.

A mi oponente Rosendo por la ayuda tan grande que me brindó con el trabajo.

Al arquitecto de datos de Capital Humano Arnolis por todo el tiempo que le interrumpí.

Al tutor Ariel al cotutor Alain y al tribunal por la ayuda brindada.

A los barberos Kiki, Rolo, Jabier, Elder, Arguelles porque ya cumplimos 5 años aquí y nos mantenemos.

A mis amigos Javier, Kerli, Alberto y Ramón por compartir conmigo estos 5 años y por todas las cosas que hemos pasado juntos y me han soportado.

A la familia de Ramón: su papá Ramón, Edita su mamá y su hermano Carlos ya que forman parte de mi familia aquí en la Habana.

A mi hermano Maceo que hemos compartido momentos malos pero lo que pasa conviene como dijimos una vez y ya somos ingenieros.

A los chicos del 11 203: Pedro, Alexander, Amed y Gustavo que compartieron conmigo un curso entero y la pasamos super...

A las chicas fieles a la causa: Ané, Ivis, Yailín, Adriana por compartir conmigo estos cinco años.

Dedicatoria

Dedicarle este triunfo antes que nada a mi papá, mi mamá y mi hermano por el amor tan grande que siento por ellos y porque sé que siempre estarán para cualquier situación que los necesite.

A mi abuelita Eulalia y a mi hermano Lesdier, que no están físicamente, pero existen siempre en mi pensamiento y sé que les hubiese gustado haberme visto realizado un ingeniero.

*Y finalmente a toda mi familia porque ya hay un **“Ingeniero”** en ella.*

Resumen

En la actualidad las instituciones y empresas cubanas han incrementado el uso de aplicaciones informáticas, con el objetivo de elevar el nivel económico y tener un control eficiente sobre los recursos que manejan. Entre las aplicaciones informáticas puestas en práctica se encuentran los sistemas de gestión, los cuales son softwares con el objetivo de integrar y automatizar todos los procesos que se llevan a cabo en las empresas. Estos sistemas deben contar con una arquitectura de software adecuada y con una descripción de la misma que permita facilitar su entendimiento y apoye el desarrollo del sistema. Actualmente la Universidad de las Ciencias Informáticas (UCI) se encuentra inmersa en la realización del sistema integral de gestión: Cedrux, donde se inserta el propósito de este trabajo que persigue actualizar la descripción de la arquitectura de datos y de sistema, construyendo las vistas correspondientes del subsistema Capital Humano de Cedrux.

Para lograrlo se dividió el trabajo en tres partes fundamentales. Primeramente se realiza la fundamentación teórica, efectuando un estudio del por qué es importante describir las arquitecturas que soportan los sistemas informáticos, consultando estándares de cómo describir de una forma adecuada y estudiando la estructura que posee la arquitectura de Capital Humano. Posteriormente se actualizará la documentación de la vista de la arquitectura de sistema, generando cada uno de los artefactos que la conforman como: Línea Base del Subsistema, Especificación de Componente, Criticidad y Complejidad de Componentes y la Matriz de Integración. Finalmente se realiza la actualización a la documentación de la vista de la arquitectura de datos, donde también se describen una serie de artefactos que la conforman como: Modelo de Datos, Diccionario de Datos, Matriz de integración y las Pruebas de Conceptos del Diseño.

PALABRAS CLAVES

ERP, Capital Humano, vista de la arquitectura de datos, vista de la arquitectura de sistema.

Índice

Introducción	1
Capítulo I Fundamentación Teórica	3
Introducción	3
1.1. Arquitectura de software	3
1.2. Vistas de la arquitectura	4
1.3. Descripción de la arquitectura de software	4
1.4. Importancia de describir la arquitectura	8
1.5. Reglas para realizar una buena descripción	9
1.6. Estructura de los expedientes tecnológicos del ERP	11
1.7. Descripción de la arquitectura de subsistema Capital Humano	12
1.8. Herramientas a utilizar	13
Conclusiones	15
Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano	16
Introducción	16
2.1 Expediente de sistema del subsistema Capital Humano	16
2.2 Línea Base del Subsistema	17
2.2.1 Especificación de Componentes	17
2.2.2 Análisis de la Complejidad y Criticidad	19
2.2.3 Priorización de los componentes	21
2.2.4 Mapa de componentes	22
2.2.5 Diseño de clases	24
2.2.6 Matriz de Integración	26
Conclusiones	28
Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano	29
Introducción	29
3.1 Expediente de datos del subsistema Capital Humano	29
3.2 Estándares	30
3.2.1 Normas de comentariado	30

3.2.2 Estándares de nomenclatura	30
3.3 Descripción de las clases persistentes de Capital Humano	33
3.4 Modelo de Datos	38
3.5 Diccionario de Datos	38
3.6 Matriz de Trazabilidad.....	47
3.7 Pruebas de Conceptos	49
Conclusiones	50
Conclusiones generales.....	52
Recomendaciones	53
Referencias bibliográficas.....	54
Bibliografía	55
Anexos.....	57
Glosario de términos.....	65

Índice de Figuras

Figura 1.1 Modelo 4+1 Vistas Arquitectónicas.....	6
Figura 1.2 Expediente tecnológico de la arquitectura de Cedrux	12
Figura 2.1 Estructura del expediente de sistema del subsistema Capital Humano	16
Figura 2.2 Especificación de los componentes Configuración y Persona de Capital Humano.....	18
Figura 2.3 Complejidad y criticidad del componente persona.	21
Figura 2.4 Mapa de componentes del subsistema Capital Humano	23
Figura 2.5 Diseño de clases del componente Régimen de trabajo y descanso.	24
Figura 2.6 Diseño de clases del componente Pagos Adicionales.	25
Figura 2.7 Matriz de Integración.	27
Figura 3.1 Estructura del expediente de datos del subsistema Capital Humano	29
Figura 4.1 Diseño de clases del componente Puesto de Trabajo.	58
Figura 4.2 Diseño de clases del componente Trabajador.	59
Figura 4.3 Diseño de clases del componente Trabajador.	60
Figura 4.4 Diseño de clases del componente Nómina.	61
Figura 4.5 Diseño de clases del componente Submayores.....	62
Figura 4.6 Diseño de clases del componente Cierre.	63
Figura 4.7 Diseño de clases del componente Incidencias.....	63

Índice de Tablas

Tabla 2.1 Componentes priorizados	22
Tabla 3.1 Descripción de las clases persistentes de Capital Humano.	33
Tabla 3.2 Descripción de disparadores	39
Tabla 3.3 Descripción de la función f_listado_regimentrabajodescanso	40
Tabla 3.4 Descripción de la tabla dat_persona.....	42
Tabla 3.5 Descripción de atributos de la tabla dat_persona	43
Tabla 3.6 Descripción de relación entre la tabla dat_persona y nom_sexo	44
Tabla 3.7 Descripción de relación entre la tabla dat_persona y nom_categoriapersona	44
Tabla 3.8 Descripción de la tabla dat_trabajador	45
Tabla 3.9 Descripción de atributos en la tabla dat_trabajador.....	45
Tabla 3.10 Descripción de relación entre tabla dat_trabajador y dat_submayorvac	46
Tabla 3.11 Descripción de relación entre tabla dat_trabajador y nom_contrato.....	46
Tabla 3.12 Matriz de Trazabilidad del subsistema Capital Humano	47

Índice de Anexos

Anexo 1: Complejidad y Criticidad	57
Anexo 2: Especificación de Componentes	57
Anexo 3: Clases del Diseño	57
Anexo 4: Diccionario de Datos.....	64
Anexo 5: Matriz de Trazabilidad	64
Anexo 6: Prueba de Concepto.....	64

Introducción

La gestión del Capital Humano es un aspecto que cada día cobra mayor importancia en el mundo. Esto se debe fundamentalmente a que repercute en el éxito y desarrollo de cualquier empresa, al influir de una forma directa en sus niveles de calidad de servicios y en los de productividad del trabajo.

En la actualidad el país posee la necesidad de gestionar el Capital Humano de una forma eficiente, debido a la mala gestión de entidades presupuestadas y empresariales que afecta directamente la economía del país. A partir de esto la Universidad de las Ciencias Informáticas en colaboración con la Unidad de Compatibilización, Integración y Desarrollo de software para la defensa (UCID), se enfrasca en la tarea de construir un sistema de Planificación de Recursos Empresariales (ERP) que sea aplicable a cualquier empresa. Lo anterior exige poner especial atención y cuidado al diseño de la arquitectura bajo la cual estará soportado el funcionamiento del sistema.

El desarrollo de la arquitectura de software es una etapa fundamental en el desarrollo del mismo, esta disciplina establece los fundamentos para que analistas, diseñadores y programadores, trabajen en una línea común con una comunicación adecuada mediante la documentación de la arquitectura, para lograr los objetivos del sistema.

La correcta documentación de la arquitectura de software en el proyecto ERP-Cuba resulta de vital importancia por tener un alcance tan amplio y estar distribuido el desarrollo del software en varias líneas que posteriormente se integrarán en un solo producto. Por ello mantener la documentación actualizada resulta un aspecto de importancia en el desarrollo del proyecto. Debido a esto se hace necesario actualizar la documentación existente en el proyecto, pues ha generado una serie de problemas como: incidencias en costos, afectación de los atributos de calidad del producto, que las soluciones no resuelvan todos los problemas a nivel de sistema, que se vea afectado el uso de la reutilización del código y sea afectada la línea de vida del proyecto. Además, no existen elementos estructurados que sirvan como guía central en la integración de las proyecciones arquitectónicas: Procesos-Sistemas-Datos, Tecnología-Seguridad-Sistema, Sistema-Diseño-Seguridad y Tecnología-Legal.

En particular en el subsistema Capital Humano se han identificado una serie de cambios de gran magnitud que sugieren una actualización de la documentación, lo que ha posibilitado el surgimiento de algunos problemas con la integración de los componentes y el uso de los servicios que brinda cada uno de ellos. Además, existen servicios duplicados, desconocimiento del estado de integración del subsistema con otros subsistemas y la afectación de la reutilización del código por parte del equipo de trabajo.

Debido a la situación mencionada se plantea como **problema científico**: La falta de actualización de la descripción arquitectónica de sistema y datos del subsistema Capital Humano afecta la capacidad de entendimiento de los involucrados y por consiguiente el soporte y desarrollo de nuevas versiones del sistema.

Para la solución del problema planteado el **objeto de estudio** trazado lo constituye la arquitectura de sistema y datos, quedando enmarcado el **campo de acción** en la Arquitectura de sistema y datos del subsistema Capital Humano.

Para la realización de este trabajo se planteó como **objetivo general**: Actualizar las vistas de arquitectura de sistema y datos del subsistema Capital Humano para facilitar su entendimiento por los involucrados y apoyar el soporte y desarrollo de nuevas versiones del sistema. Para dar cumplimiento a dicho objetivo se trazan los siguientes **objetivos específicos**:

- Realizar el marco teórico de la investigación.
- Actualizar la vista de arquitectura de sistema del subsistema Capital Humano.
- Actualizar la vista de arquitectura de datos del subsistema Capital Humano.

Posibles resultados:

Vistas de la arquitectura de sistema y de datos del subsistema Capital Humano del proyecto ERP Cuba.

Idea a defender:

Si se actualizan las vistas de arquitectura de sistema y datos del subsistema Capital Humano se facilitará su entendimiento por parte de los involucrados y apoyará el soporte y desarrollo de las nuevas versiones del sistema.

El contenido de este trabajo está estructurado en tres capítulos. En el capítulo I se abordará lo referente a la fundamentación teórica con que se debe contar para el desarrollo de este trabajo. En el capítulo II se desarrollarán los artefactos de la vista de sistema del subsistema Capital Humano y en el capítulo III se desarrollarán los artefactos de la vista de datos del subsistema Capital Humano.

Capítulo I Fundamentación Teórica

Introducción

El capítulo trata los aspectos más importantes de la arquitectura del software, donde se retoma su definición, refiriéndose además a las vistas de la misma, las cuales ocupan un lugar importante en el desarrollo de un software. Se realiza un estudio de los métodos de descripción arquitectónica que existen, específicamente el modelo que propone RUP de las 4+1 vistas arquitectónicas. Además, se expone la importancia de documentar las arquitecturas de los proyectos y las reglas que existen para realizar una buena documentación. Concluyendo con una visión de la estructura que poseen los expedientes tecnológicos contenedores de las vistas en Cedrux y una descripción de la arquitectura de Capital Humano. Es importante tener en cuenta que en la actualidad en el mundo no existe una abundante documentación en la que se pueda apoyar el trabajo; debido a que la documentación de la arquitectura es un aspecto algo pobre a pesar de su importancia y que a diario aumenta su crecimiento y desarrollo.

1.1. Arquitectura de software

“La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.”

(1) La misma está compuesta por la estructura de los elementos de un programa o sistema, sus interrelaciones y los principios o reglas que gobiernan su diseño y evolución a lo largo del tiempo. Establece los fundamentos para que el equipo de desarrollo logre trabajar en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. (2)

La arquitectura a grandes rasgos, es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se les percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. Además por el carácter tan amplio que posee la misma y la responsabilidad que conlleva corregir lo que está realizado o simplemente desarrollarlo, toda arquitectura que responda a un software responsable y con prestigio, debe estar estructurada adecuadamente y contar con los artefactos necesarios que sirvan de apoyo al equipo de programadores. De tal manera que permita un buen entendimiento por parte de los

involucrados en su desarrollo. Por ello es de suma importancia tener en cuenta y analizar cada una de las vistas de la arquitectura donde se describe todo lo referente al producto.

1.2. Vistas de la arquitectura

Las vistas permiten visualizar, entender y razonar los elementos más significativos de la arquitectura y a su vez identificar las áreas de riesgo que requieren mayor detalle de elaboración. Toda arquitectura de software debe describir diversos aspectos. Generalmente cada uno de estos aspectos se describe de la manera más comprensible si se utilizan distintos modelos o vistas. Es importante destacar que cada uno de ellos constituye una descripción parcial de una misma arquitectura y es favorable que exista cierto solapamiento entre ellos, es decir, la descripción de una arquitectura de un software debe estar distribuida en la descripción de las vistas por la cual está compuesta. “Cada paradigma de desarrollo exige diferentes números y tipos de vistas o modelos para describir una arquitectura”. (3) No obstante, existen al menos tres vistas fundamentales que por lo general están presentes en muchas arquitecturas:

- La vista **estática**: describe qué componentes tiene la arquitectura.
- La vista **funcional**: describe qué hace cada componente.
- La vista **dinámica**: describe cómo se comportan los componentes a lo largo del tiempo y cómo interactúan entre sí.

Las vistas usadas para una arquitectura dependen de las necesidades y el uso que se le dará a la misma, por ello para cada arquitectura independientemente de cualquier metodología existente, se puede contar con tantas vistas como sean necesarias mientras su estructura y diseño lo exijan. Es por ello que las vistas definidas en Cedrux no desarrollan una metodología en específico, sino que se basan en el estudio y análisis del modelo que propone RUP para su definición, como se mostrará posteriormente.

1.3. Descripción de la arquitectura de software

Documentar la arquitectura de software es cuestión de documentar las vistas relevantes. Como plantea Clements en el libro “Documentación de la Arquitectura de Software”:

“El principio básico de documentar una arquitectura como un conjunto de vistas aporta una ventaja de divide y vencerás a la tarea de documentación, pero si las vistas son

irrevocablemente diferentes, sin relación la una con la otra, nadie podría entender el sistema como un todo.”

Un sistema pobremente documentado carece de valor aunque haya funcionado bien en alguna ocasión. En el caso de programas pequeños y poco importantes que sólo se utilizan durante un corto período de tiempo, unos cuantos comentarios en el código podrían ser suficientes. No obstante, la mayoría de los programas cuya única documentación es el código, se quedan obsoletos rápidamente y es imposible mantenerlos. A menos que se sea infalible y se viva en un mundo en el que nada cambia, se tendrá que volver a consultar el código que ya estaba escrito y se pondrán en duda decisiones que se tomaron durante el desarrollo del mismo. Si no se documentan las decisiones se estará siempre cometiendo los mismos errores y tratando de comprender lo que se pudo haber descrito fácilmente en una ocasión. La falta de documentación no sólo genera trabajo adicional, sino también tiende a dañar la calidad del código.

(4) Documentar la arquitectura de software es una tarea complicada y exige un criterio de ingeniería maduro. Documentar de forma concisa es un error habitual, pero si se escriben documentaciones extensas, estas atosigarán al lector y constituirán una carga en el momento de conservarlas. Es esencial documentar sólo los asuntos correctos. La documentación no sirve de ayuda para ningún involucrado si su extensión desalienta a las personas cuando la consultan.

Muchas veces los arquitectos no documentan los proyectos y no acompañan a los desarrolladores en el proceso de documentación. En ocasiones no se involucran de la forma adecuada en cada proyecto y sólo definen de vez en cuando un aspecto general de la arquitectura. A partir de esto, muchas empresas dedicadas al desarrollo de software, actualmente describen su arquitectura mediante un documento de arquitectura de software (4).

En el proyecto ERP-Cuba existen problemas con la documentación como se pudo observar en la introducción a este trabajo, por lo que es vital entender y analizar de manera efectiva los métodos de descripción existentes para poder dar solución a los objetivos planteados. En la actualidad se han desarrollado varios métodos para la descripción de la arquitectura como son: el método que propone RUP de las 4+1 vistas arquitectónicas, el método Perspectivas Arquitectónicas que son conceptos que se basan en la idea de establecer un punto de vista arquitectónico, la ISO 42010 /IEEE (Instituto de Ingenieros Electricistas y Electrónicos) 1471 que es la práctica recomendada para la Descripción de la Arquitectura de Software de Sistemas Intensivos y por último el método desarrollado por el SEI (Instituto de Ingeniería de Software): Vistas y Perspectivas que realiza la documentación de la arquitectura a partir de la identificación de las vistas relevantes de la arquitectura.

➤ Método que propone RUP de las 4+1 vistas arquitectónicas

Las 4+1 vistas arquitectónicas propuestas por la metodología RUP, facilitan la comprensión de la arquitectura propuesta y aportan argumentos para la identificación de los elementos claves del sistema en desarrollo, proporcionando información al cliente sobre las decisiones arquitectónicas en la construcción del producto (5). Esta metodología establece los siguientes propósitos fundamentales:

- Proporcionar una comprensión arquitectónica global del sistema a los stakeholders (revisores técnicos) mediante el uso de vistas arquitectónicas, de modo que se muestren los aspectos más significativos del sistema.
- Proporcionar soporte para la toma de decisiones arquitectónicamente significativas que deban ser tomadas en el desarrollo del sistema.
- Organizar el desarrollo del sistema.
- Fomentar la reutilización.
- Hacer evolucionar el sistema.

El modelo 4+1 propone una descripción de la arquitectura del software usando cinco vistas concurrentes como lo muestra la figura 1.1. Cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema.

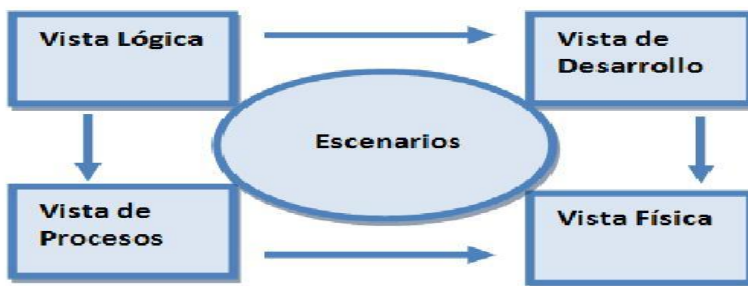


Figura 1.1 Modelo 4+1 Vistas Arquitectónicas

- **La vista lógica**

La vista lógica describe principalmente los requisitos funcionales, lo que el sistema debe brindar en términos de servicios a sus usuarios. El sistema se descompone en una serie de abstracciones claves tomadas principalmente del dominio del problema en la forma de objetos o clases de objetos. Aquí se aplican los principios de abstracción, encapsulamiento y herencia. Esta descomposición no sólo se realiza

para potenciar el análisis funcional, sino también para identificar mecanismos y elementos de diseño comunes en diversas partes del sistema. Un diagrama de clases muestra un conjunto de clases y sus relaciones lógicas: asociaciones, uso, composición, herencia y similares. Además grupos de clases relacionadas pueden agruparse en categorías de clases.

- **La vista de procesos**

La arquitectura de procesos toma en cuenta algunos requisitos no funcionales tales como el rendimiento y la disponibilidad. Se enfoca en asuntos de concurrencia y distribución, integridad del sistema y de tolerancia a fallas. La vista de procesos también especifica en cuál hilo de control se ejecuta efectivamente una operación de una clase identificada en la vista lógica. La arquitectura de procesos se describe en varios niveles de abstracción, donde cada nivel se refiere a distintos intereses. El nivel más alto de la arquitectura de procesos puede verse como un conjunto de redes lógicas de programas comunicantes (llamados “procesos”) ejecutándose en formas independientes y distribuidas a lo largo de un conjunto de recursos de hardware conectados mediante una red. Múltiples redes lógicas pueden usarse para apoyar la coexistencia de versiones de software de simulación o de prueba. Un proceso es una agrupación de tareas que forman una unidad ejecutable. Los procesos representan el nivel al que la arquitectura de procesos puede ser controlada tácticamente. Además los procesos pueden replicarse para aumentar la distribución de la carga de procesamiento o para mejorar la disponibilidad (5).

- **La vista física**

Describe el mapeo del software en el hardware y refleja los aspectos de distribución.

- **Vista de desarrollo**

Describe la organización estática del software en su ambiente de desarrollo.

Los diseñadores construyen la arquitectura usando varios elementos arquitectónicos elegidos apropiadamente. Estos elementos satisfacen la mayor parte de los requisitos de funcionalidad y el rendimiento del sistema, así como también otros requisitos no funcionales tales como: confiabilidad, escalabilidad, portabilidad y disponibilidad del sistema (5).

El equipo de arquitectura del proyecto ERP–Cuba después de un estudio previo realizado y tomando como partida el estudio del método que propone RUP de las 4+1 vistas arquitectónicas, estableció las siguientes vistas para el flujo de trabajo de la arquitectura:

- Vista de la Arquitectura de Sistema.
- Vista de la Arquitectura de Datos.
- Vista de la Arquitectura de Integración.
- Vista de la Arquitectura de Seguridad.
- Vista de la Arquitectura de Presentación.
- Vista de la Arquitectura de Tecnología.
- Vista de la Arquitectura de Infraestructura.

Para la realización de este trabajo se centrará la atención solamente en la descripción de la vista de la arquitectura de sistema y la vista de la arquitectura de datos.

1.4. Importancia de describir la arquitectura

Un propósito primario de la arquitectura de software es servir de vehículo de comunicación entre stakeholders. La documentación de la arquitectura debe ser lo suficientemente abstracta para ser rápidamente comprendida por nuevos stakeholders, pero a la vez suficientemente detallada como para servir de plan detallado en un análisis posterior (6). Los diferentes stakeholders tienen diferentes necesidades, niveles y tratamientos de la información, por tanto, no se debe esperar producir un documento arquitectónico para que cada lector lo interprete de la misma forma, más bien se debe producir una documentación que ayude a estructurar bien la información.

Comprendiendo quiénes son los stakeholders y cómo usarán la documentación ayudará a organizarla y ponerla al alcance de estos de forma utilizable. La documentación de la arquitectura debe ser redactada de forma tal que pueda servir tanto a los stakeholders experimentados como a los nuevos que se vayan incorporando.

Por todo lo mencionado anteriormente se concluye que la documentación de la arquitectura tiene tres usos fundamentales:

1. Sirve como medio de educación. El uso educacional consiste en iniciar a las personas en el sistema. Estas personas pueden ser nuevos miembros de equipo de desarrollo, analistas externos o incluso nuevos arquitectos.
2. Sirve como vehículo primario para la comunicación entre los stakeholders. El uso preciso de una arquitectura como el vehículo de comunicación depende de qué stakeholder está haciendo la comunicación.
3. Se utiliza como base para el análisis de sistemas. Para soportar el análisis, la documentación de la arquitectura debe contener la información necesaria para realizar un análisis particular (6).

1.5. Reglas para realizar una buena descripción

En el mundo a pesar de que la descripción de la arquitectura es un aspecto algo pobre aún, constituye un factor de desarrollo debido al margen tan amplio que posee, existen reglas que dominan la forma de describir, reglas que definen al documentador un conjunto de estándares sumamente importantes. (7) Entre tantas existen diez que encabezan la lista, estas son:

Regla 1: Escribir la documentación desde el punto de vista del lector

No usar vocabulario específico y tener en cuenta que el documento se leerá muchas más veces de las que fue escrito.

Regla 2: Evitar la repetición innecesaria

Cuidar no decir las mismas palabras más de una vez a menos que sea necesario. Intentar referir a otras partes del documento u otros documentos. Aunque a veces promueve una profunda comprensión o trasmite énfasis, la repetición con variaciones puede ser confusa.

Regla 3: Evitar la ambigüedad

Utilizar las mismas palabras para referirse a los mismos conceptos, no buscar dar matices para mejorar una exposición, elegir los aspectos a desarrollar y concentrarse en ellos.

Regla 4: Usar una organización estándar

Mantener una organización estándar de la documentación ayudará a los lectores a encontrar lo que buscan y les servirá para asegurarse de que está completa y correcta. Organizar los diagramas de dependencia en un directorio estándar del modelo, que puede corresponder a la vista de dependencias integradas, facilita el acceso a los modelos y ayuda a crearlos estableciendo para ellos un lugar específico predefinido.

Regla 5: Registrar las razones

Dejar constancia de las razones que motivaron las decisiones tomadas y de las alternativas descartadas. Para registrar las razones de cada decisión es difícil utilizar diagramas, aunque pueden agregarse notas aclaratorias.

Regla 6: Mantener la documentación actualizada

Mantener la documentación actualizada, de lo contrario no será consultada. Evitar colocar demasiados detalles en la documentación, para no redundar información entre el software y la documentación.

Regla 7: Revisar si la documentación cumple su propósito

Lograr que los lectores revisen y den su aprobación a la documentación. Como toda herramienta de comunicación, la vista integrada de dependencias debe ser revisada por lectores para confirmar su capacidad de transmitir la información que almacenan.

Regla 8: Publicar eficientemente la documentación

Es importante lograr que la información esté disponible para todos los lectores en forma rápida y a un bajo costo de búsqueda para fomentar su uso.

Regla 9: Utilizar herramientas

Tanto el formato como las herramientas de edición seleccionadas, deben promover la comodidad y permitir el acceso de todos los lectores. Elegir herramientas que en lo posible permitan fácilmente vincular documentos entre sí y buscar en el repositorio de información.

Regla 10: Tener en cuenta los objetivos del proyecto

Al redactar la documentación, el documentador deberá tener en mente los objetivos estratégicos del proyecto. Una vista integrada de dependencias ayuda a realizar el análisis de impacto de los cambios y mejora la flexibilidad del sistema. No es recomendable para sistemas con ciclo de vida muy acortado y baja probabilidad de cambios, ya que el esfuerzo de desarrollar esta documentación puede no traer beneficios significativos. Por otro lado, en metodologías ágiles puede ser apropiado porque puede mantenerse cerca del código y automatizarse el control del sistema en base a la documentación. La aplicación práctica de los modelos integrados de dependencias facilita documentar información, para la cual no se tiene un espacio definido. Esa característica aporta un valor especial pues facilita la construcción de la documentación ofreciendo un espacio para ubicar la información y recíprocamente aporta una organización definida de la documentación que permita buscar y acceder con facilidad a la misma. En las metodologías orientadas al plan y los documentos, este modelo se organiza directamente como una sección de la documentación de arquitectura y diseño, en particular en los modelos orientados a vistas como los que se describen, se pueden agregar como una vista más o como un apéndice.

Estas reglas permiten una mejor relación entre lectores y artefactos, debido a que no es solamente escribir, sino tener una lógica de lo que se está escribiendo y lograr un estándar mejorado, que facilite un total entendimiento por parte de los involucrados.

1.6. Estructura de los expedientes tecnológicos del ERP

Todo flujo de trabajo que se desarrolla durante la construcción de la arquitectura, debe estar organizado y/o estructurado con el objetivo de crear un lugar en el cual depositar la información organizada. Un lugar que constituya un espacio para documentación, que evite sobrescribir documentos, código, etc., que lleve un control de versiones y que guíe el proceso de desarrollo del software en general. El expediente tecnológico es una carpeta destinada a este fin.

El expediente tecnológico se realiza con el objetivo de estructurar y guiar organizacionalmente la construcción de la arquitectura del software. Este queda conformado a partir de siete vistas que conforman la arquitectura de Cedrux, quedando estructurado como se muestra en la figura 1.2.



Figura 1.2 Expediente tecnológico de la arquitectura de Cedrux

El expediente consta de una carpeta para cada una de las vistas de la arquitectura y una para recoger todos los aspectos relacionados con la gestión de la calidad debido a que en cada una de las áreas se definen estándares y listas de chequeo a utilizar por las demás líneas de desarrollo. Dentro de cada una de estas carpetas se encuentra la documentación asociada a cada disciplina, organizada en una serie de subcarpetas que tienden a ser comunes para cada arquitectura.

1.7. Descripción de la arquitectura de subsistema Capital Humano

El subsistema Capital Humano está compuesto fundamentalmente por los componentes referidos a continuación.

Persona: Manejan todos los datos personales necesarios a tener en cuenta cuando un individuo se convierte en trabajador.

Trabajador: Encargado de gestionar los datos respectivos a los trabajadores. Administra datos contables así como los movimientos de alta, baja y/o reubicación en un puesto de trabajo.

Nómina: Procesa la nómina de un área de trabajo, teniendo en cuenta las incidencias y los pagos adicionales de los trabajadores.

Submayores: Actualiza los submayores de vacaciones y retenciones de un trabajador y verifica que el saldo del submayor coincida.

Puesto de Trabajo: Gestiona los puestos de trabajo asociados a un cargo que pertenecen a un área determinada.

Pagos Adicionales: Administra los pagos adicionales de un trabajador y los asocia a un puesto de trabajo.

Incidencias: Registra las incidencias de un trabajador por un período de pago.

Régimen de Trabajo y Descanso: Define los días en que debe laborar el trabajador que se acoja al mismo. El régimen también define de cada día de trabajo las sesiones y la hora de inicio y fin de cada sesión.

Cierre: Realiza el proceso de cierre contable teniendo en cuenta datos de la nómina.

Configuración: Contiene los componentes generales de configuración del subsistema.

Entender la arquitectura del subsistema Capital Humano beneficia el trabajo a realizar y favorece los resultados que se obtendrán posteriormente. Por la importancia que esto implica es preciso analizar cada detalle de su composición y entender los aspectos fundamentales que la caracterizan a gran escala.

1.8. Herramientas a utilizar

Visual Paradigm 6.3

Visual Paradigm para UML (Lenguaje Unificado de Modelado) es una herramienta ampliamente utilizada en el mundo del software que permite a los profesionales modelar sus diseños. Posibilita la integración de aplicaciones empresariales a las bases de datos. Es capaz de generar código e ingeniería inversa para lenguajes como el PHP (Lenguaje de programación web). Permite manejar grandes estructuras de manera eficiente, sólo requiere una configuración de escritorio común. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegues (8). Esta herramienta posee mucha importancia, ya que sirve de apoyo para generar el Modelo de Datos y Diccionario de Datos; artefactos fundamentales en la vista de la arquitectura de datos del subsistema Capital Humano. Además facilita el trabajo con el manejo de las tablas, sus relaciones, disparadores y funciones de la base de datos.

PostgreSQL 8.2

PostgreSQL es una fuente poderosa, abierta al sistema de la base de datos correlativo. Posee más de 15 años de desarrollo activo y una arquitectura probada, ha ganado una reputación fuerte para la fiabilidad, integridad de los datos y exactitud. Corre en todos los sistemas operativos incluidos Linux y Windows. También apoya el almacenamiento de grandes objetos binarios, incluso imágenes, sonidos y videos. Tiene interfaces nativas de la programación para los lenguajes C++, Java.net, Perl, Python, Ruby, entre otros y una documentación excepcional. Entre sus principales características pueden encontrarse:

- Disponible totalmente sin costo alguno.
- Disponible para los Sistemas Operativos UNIX y Windows.
- Soporte total del Modelo Relacional de Bases de datos.
- Extensiones propias a SQL para realizar consultas sobre la base de datos.
- Dependencias entre objetos, integridad referencial.
- Soporta valores no atómicos como dominio de un campo. (9)

Con esta herramienta se desarrollará el artefacto: Pruebas de Conceptos de la vista de datos. Su uso brinda un fácil trabajo con la base de datos al tratar sus tablas y atributos.

Conclusiones

Con este capítulo se ampliaron los conocimientos sobre los temas relacionados con la documentación de la arquitectura de software, al estudiar las vistas, métodos de descripción arquitectónica, específicamente el método que propone RUP de las 4+1 vistas arquitectónicas, importancia de describir la arquitectura y reglas para realizar una buena descripción. Además, se estudió la estructura que poseen los expedientes tecnológicos en el ERP, la cual está compuesta por las vistas que conforman la arquitectura de Cedrux y basada en el método de descripción planteado anteriormente. También se fomentó la estructura que posee Capital Humano basada en sus componentes para lograr una buena relación con el medio. Con esto se sentaron las bases para realizar una buena actualización de las vistas de sistema y datos de la arquitectura del subsistema Capital Humano de Cedrux y se dio respuesta a un objetivo específico. Por ello, se puede comenzar el desarrollo de los artefactos que conforman cada una estas vistas en los próximos capítulos.

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Introducción

La vista de sistema de la arquitectura de software del subsistema Capital Humano representa una proyección simétrica de alto nivel de los procesos de arquitectura de negocio que se trabaja, expresada en: elementos, conectores, configuraciones y restricciones. Recoge la documentación formal e imprescindible con la que se debe contar para la creación de la arquitectura de sistema. Estos son documentos de identificación y descripción de los componentes. Este capítulo va a estar regido por la guía de actividades propuesta por el equipo de arquitectura del proyecto ERP-Cuba. Donde se definen cuatro artefactos principales que responderán a la línea de Capital Humano, los cuales son: Documento Línea Base del Subsistema, Especificación de Componentes, Criticidad y Complejidad de los componentes y por último la Matriz de Integración.

2.1 Expediente de sistema del subsistema Capital Humano

En el Capítulo anterior se especificó la estructura que poseen los expedientes tecnológicos del subsistema Capital Humano. En este epígrafe se muestra cómo están estructurados los expedientes de sistema del subsistema (Figura 2.1), expedientes que recogen los artefactos que se deben generar para conformar la estructura correspondiente a la documentación.

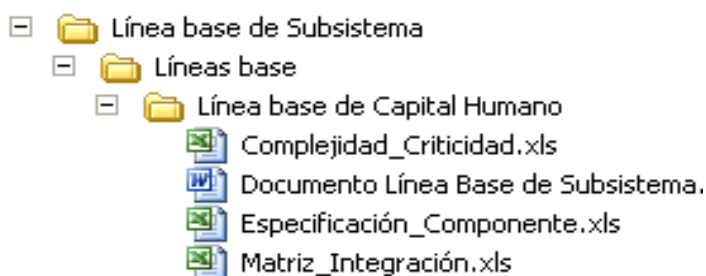


Figura 2.1 Estructura del expediente de sistema del subsistema Capital Humano

Cada uno de los artefactos generados en la vista de sistema de Capital Humano, conforman parte de la documentación con que debe contar el sistema en su completa conformación, tanto en su finalización

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

como en el proceso de desarrollo. Por ello, es de vital importancia ir actualizando cada uno de los artefactos a medida que ocurran cambios, pues estos rigen una serie de aspectos de mucho peso para el desarrollo del sistema.

2.2 Línea Base del Subsistema

El objetivo del documento Línea Base del Subsistema es delimitar las responsabilidades arquitectónicas del subsistema Capital Humano y establecer la comunicación y dependencias entre los componentes que lo integran, así como plasmar la criticidad y complejidad de los componentes con vista a su reutilización y prioridad en el desarrollo. El mismo está destinado a todos los arquitectos de componentes, de sistema, de datos y jefes de línea del proyecto que participan en el flujo de actividades para la conformación de la línea base de la arquitectura. Es un artefacto que posee gran peso por la estructura tan amplia que lo conforma.

- **Línea base de la arquitectura:** Son las definiciones más generales desde el punto de vista arquitectónico del sistema, a partir de las cuales se edifica toda la solución y la implementación.

A continuación se procederá a actualizar el artefacto, donde se desarrollarán los demás artefactos de la vista.

2.2.1 Especificación de Componentes

En este artefacto se listan todos los componentes definidos en el subsistema y de cada uno de ellos se especificará su descripción basada en el objetivo y sus principales características, los servicios que provee con sus detalles, código del mismo, el cual estará compuesto por: <<Subsistema>> <<Componente>> <<No. consecutivo>>, así como los atributos de entrada y salida de cada uno de los servicios. El objetivo de este artefacto es tener registrada la formalización de cada componente y la relación de los servicios que brinda. A continuación, en la figura 2.2 se muestra la especificación de los componentes Configuración y Persona:

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Componentes	Descripción	Servicios que brinda	Código	Descripción	Entrada	Salida
Configuración	Contiene los componentes generales de configuración del subsistema	InsertarDocumento1	CHDocumento01	Inserta un documento (por el momento se utiliza al registrar contratos y movimientos de nómina)	idtipodocumento	Objeto Documento ir
		ObtenerImpuesto	CHImpuesto01	Obtener impuestos dado un set de pa	Objeto stdClass con los siguientes atributos: idimpuesto, codigo, denom, porcentaje, idelementocomun (este es el id del centro de costo), idcuenta, idotr cuenta,	Arreglo de objetos
Persona	Administra todos los datos de la persona.	guardarPersona	CHPersona01	Guardar Personas al ejecutar Carga In	Objeto stdClass con los sigue	idpersona
		BuscarPersonaPor	CHPersona02	Buscar persona dado un set de parám	1. Objeto stdClass con los siguientes atributos: idpersona, numid, nombre, primerapellido, segundoapellido, idcategoriapersona. Cada uno de estos atributos son opcionales, si el objeto llega	Arreglo de objetos

Figura 2.2 Especificación de los componentes Configuración y Persona del subsistema Capital Humano.

Además, en el documento se recogen los distintos patrones utilizados para la implementación de los componentes Persona, Trabajador, Pagos Adicionales, Incidencias, Puesto de Trabajo, Nómina, Submayores, Configuración, Régimen de descanso y trabajo y Cierre. Estos patrones son: Alta Cohesión, Bajo Acoplamiento, Experto, Controlador y Creador.

En este artefacto se muestra cómo se redujo la cantidad de servicios, evitando duplicaciones y parametrizándolos, de modo que un servicio recibe un objeto con posibles atributos a filtrar. De esta forma se optimiza la eficiencia y limpieza del código.

Para una mayor comprensión consultar el documento Especificación de Componentes del subsistema Capital Humano ver Anexo 2.

2.2.2 Análisis de la Complejidad y Criticidad

El documento Complejidad y Criticidad está compuesto por todos los componentes definidos en el subsistema y por cada uno de ellos los requisitos funcionales que implementan. Además, contiene la complejidad de esos requisitos la cual oscila entre 1 y 10, la cantidad de servicios que brinda cada componente y el tipo de componentes el cual puede ser: Núcleo, Configuración, Integración, Tecnología o Prueba, donde:

Núcleo:

Representa aquellos componentes con la responsabilidad de abstraer las principales funcionalidades del negocio de la organización, comúnmente agrupan la mayor parte de los conceptos del modelo conceptual y en términos arquitectónicos agrupan los requisitos funcionales del sistema que solicita el cliente.

Configuración:

Representa aquellos componentes con la responsabilidad de abstraer las configuraciones y parametrizaciones estáticas o dinámicas del sistema, así como las características de los estándares a utilizar en la conversión de formato y validaciones de procesos del negocio.

Integración:

Representa aquellos componentes con la responsabilidad de abstraer la estrategia de integración y colaboración del sistema, así como los flujos y subflujos de trabajo. Otra de las responsabilidades arquitectónicas que descansan en este grupo de componentes o elementos arquitectónicos, es la interoperabilidad con otras plataformas tecnológicas afines al sistema en cuestión o de otro tipo como las plataformas ofimáticas, pasarelas bancarias, sistema de gestión de entidades o sistemas legados de la organización que requieren interacción con el sistema en modelación. Esta categoría asume además, los componentes asociados a características horizontales en la arquitectura de sistema, a la implementación de patrones de integración para mejorar la calidad del diseño arquitectónico y la aplicación de los patrones GRASP (patrones para asignación de responsabilidades) y los componentes creados con el objetivo de eliminar lazos de dependencia funcional entre componentes o para implementar políticas de reutilización específicas de la arquitectura de sistema. Esta característica posee un alto impacto en los elementos tecnológicos del diseño, pues en ellos recae la responsabilidad de la actividad del proceso, pero el elemento tecnológico debe garantizarlo el grupo de componentes arquitectónicos tecnológicos.

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Tecnología:

Representa aquellos componentes con la responsabilidad de abstraer características puramente tecnológicas, que sirven de base tecnológica al resto de los componentes del sistema. Tienen la responsabilidad de abstraer las características no funcionales, con el objetivo de lograr una separación arquitectónica y una mejor especialización entre los diferentes grupos de arquitectos de soluciones, permitiéndose una mejor trazabilidad en la arquitectura del sistema de la arquitectura del negocio. Estos grupos de componentes no forman parte en sí de la arquitectura de sistema, aunque la integración de estos con el resto de los grupos de componentes es importante para cada una de las áreas que trabajan. Generalmente son constituidos en frameworks tecnológicos que permiten una alta reutilización de las características tecnológicas, además de una alta productividad en el desarrollo, es el grupo de componentes más asociado con la arquitectura tecnológica.

Prueba:

Representa aquellos componentes con la responsabilidad de abstraer las pruebas unitarias y de integración del resto de los componentes de los otros grupos. Por el costo y el esfuerzo que requieren generalmente son asociados solamente a los componentes núcleos. Con el objetivo de poder maquetar la solución y garantizar que el núcleo funcional es válido, para a partir de una aceptación de la maqueta del producto subcontratar el resto de las características del sistema, los niveles de parametrización, interoperabilidad e integración a incluir en la solución.

A partir de estos datos se determinan la complejidad y criticidad donde:

- **La complejidad de cada componente** = cantidad de requisitos + (la complejidad de los requisitos del componente * 4) + (la cantidad de servicios del componente * 2).
- **La criticidad de cada componente** = cantidad de componentes que dependen + la sumatoria de la criticidad de los componentes dependientes.

En la figura 2.3 se muestra la complejidad y criticidad para el componente Persona del subsistema Capital Humano.

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Componente	Cant	No.	Requisitos	Código	Complejic	Servicios	Componentes de	Tipo de Com	Complejida	Criticidad de Integraci
Persona	20	6	Adicionar centro de estudio	CapitalHuma	5	2	5	Núcleo	440	3663
		7	Modificar centro de estudio	CapitalHuma	5					
		8	Eliminar centro de estudio	CapitalHuma	5					
		9	Listar centro de estudios	CapitalHuma	5					
		10	Adicionar título	CapitalHuma	5					
		11	Modificar título	CapitalHuma	5					
		12	Eliminar título	CapitalHuma	5					
		13	Listar título	CapitalHuma	5					
		14	Adicionar persona	CapitalHuma	7					
		15	Modificar datos de la perso	CapitalHuma	7					
		16	Eliminar datos de persona	CapitalHuma	5					
		17	Listar personas	CapitalHuma	5					
		18	Buscar persona	CapitalHuma	5					
		19	Registrar características pe	CapitalHuma	5					
		20	Registrar contacto	CapitalHuma	5					
		21	Registrar datos de nacieme	CapitalHuma	5					
		22	Registrar nivel educacional	CapitalHuma	5					
		23	Registrar residencia	CapitalHuma	5					
		24	Imprimir listado de persona	CapitalHuma	5					
		25	Imprimir datos de una pers	CapitalHuma	5					

Figura 2.3 Complejidad y criticidad del componente persona.

El objetivo fundamental de este artefacto es evaluar por cada componente su complejidad, así como qué tan importante es desde el punto de vista de la integración. Ambos criterios son variables de peso en el momento de definir una prioridad de implementación.

Para una mayor comprensión consultar el documento Complejidad y Criticidad de los componentes del subsistema Capital Humano. Ver Anexo 1.

2.2.3 Priorización de los componentes

Para decidir qué componentes desarrollar primero se tuvo en cuenta la cantidad de dependencias que generan cada uno de ellos y la criticidad de los componentes dependientes, donde el resultado de esta suma proporciona la criticidad del componente (calculada en el artefacto anterior) y a su vez define su prioridad. Por ello se consultó el artefacto Criticidad Y Complejidad donde se analizó por cada componente su criticidad, la cual va a decidir la prioridad de desarrollo de los componentes (Tabla 2.1).

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Tabla 2.1: Componentes priorizados

Componente	Criticidad	Prioridad
Configuración	3663	1
Persona	3663	1
Pagos Adicionales	3663	1
Régimen de Trabajo y Descanso	3663	1
Puesto de Trabajo	3161	2
Trabajador	1618	3
Incidencias	1120	4
Submayores	595	5
Nómina	585	6
Cierre	29	7

2.2.4 Mapa de componentes

El mapa de componentes contendrá todos los componentes definidos en el subsistema, así como las distintas dependencias existentes entre ellos. Su objetivo es obtener una vista general del subsistema a partir de las dependencias e integraciones de los componentes. Permitiendo evaluar visualmente por cada componente su complejidad, así como qué tan importante es desde el punto de vista de la integración. Figura 2.4.

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

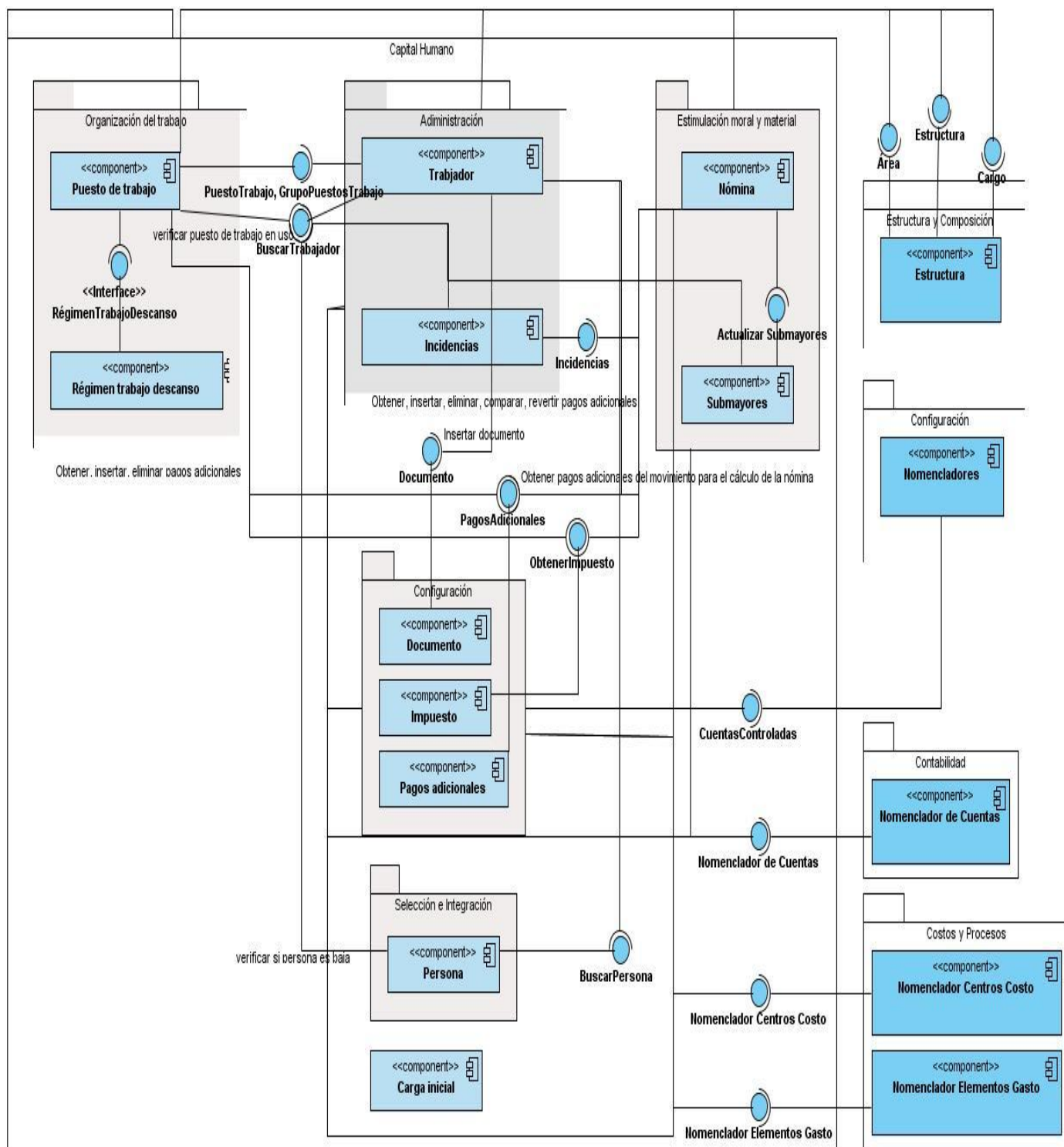


Figura 2.4 Mapa de componentes del subsistema Capital Humano

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

2.2.5 Diseño de clases

En el artefacto Línea de Base del Subsistema se muestra además, la estructura de cómo están conformadas las clases del diseño por cada uno de los componentes. A continuación se relacionan las clases de los componentes Régimen de Trabajo y Descanso y Pagos Adicionales.

Régimen de Trabajo y Descanso

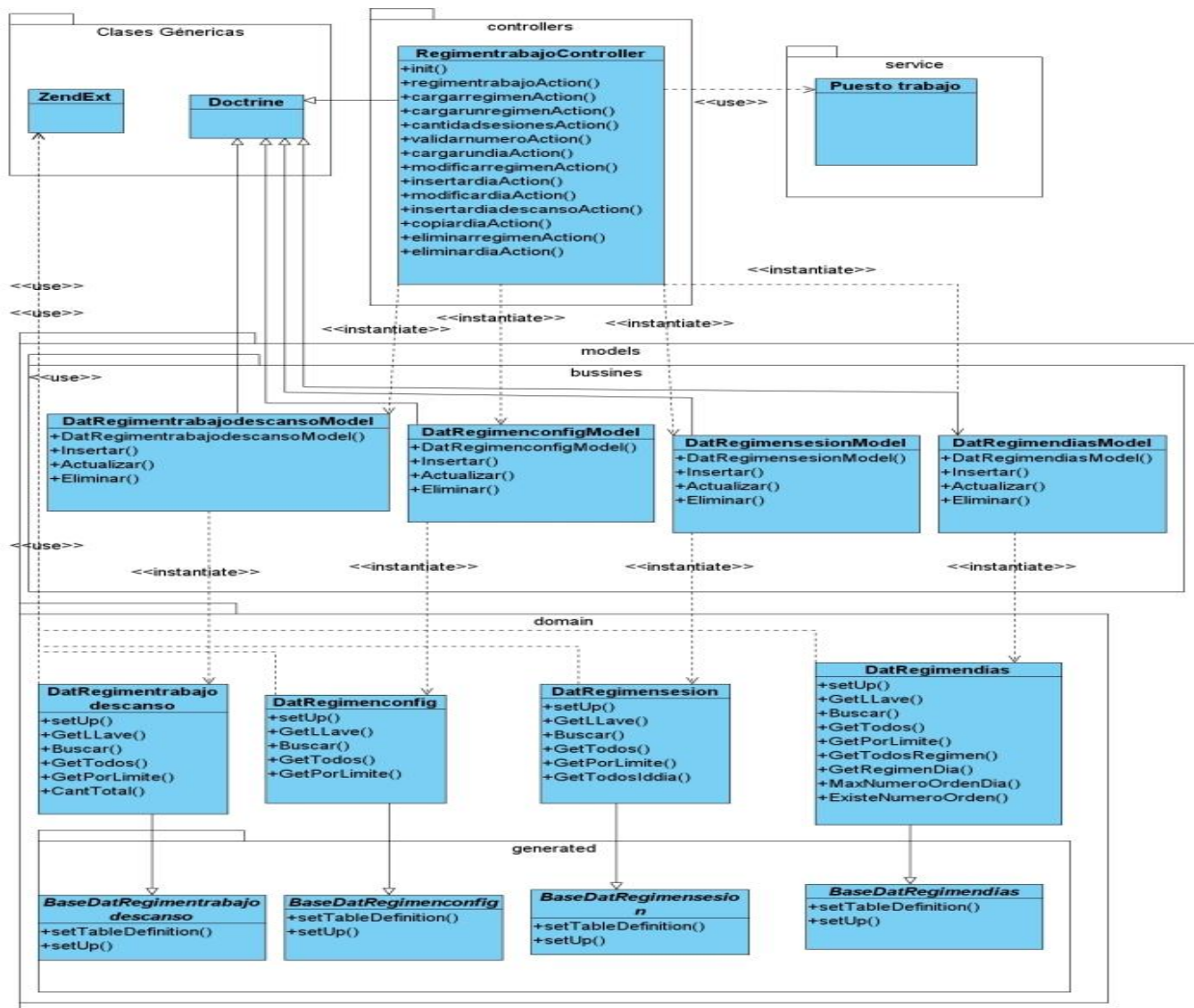


Figura 2.5 Diseño de clases del componente Régimen de trabajo y descanso.

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Pagos Adicionales

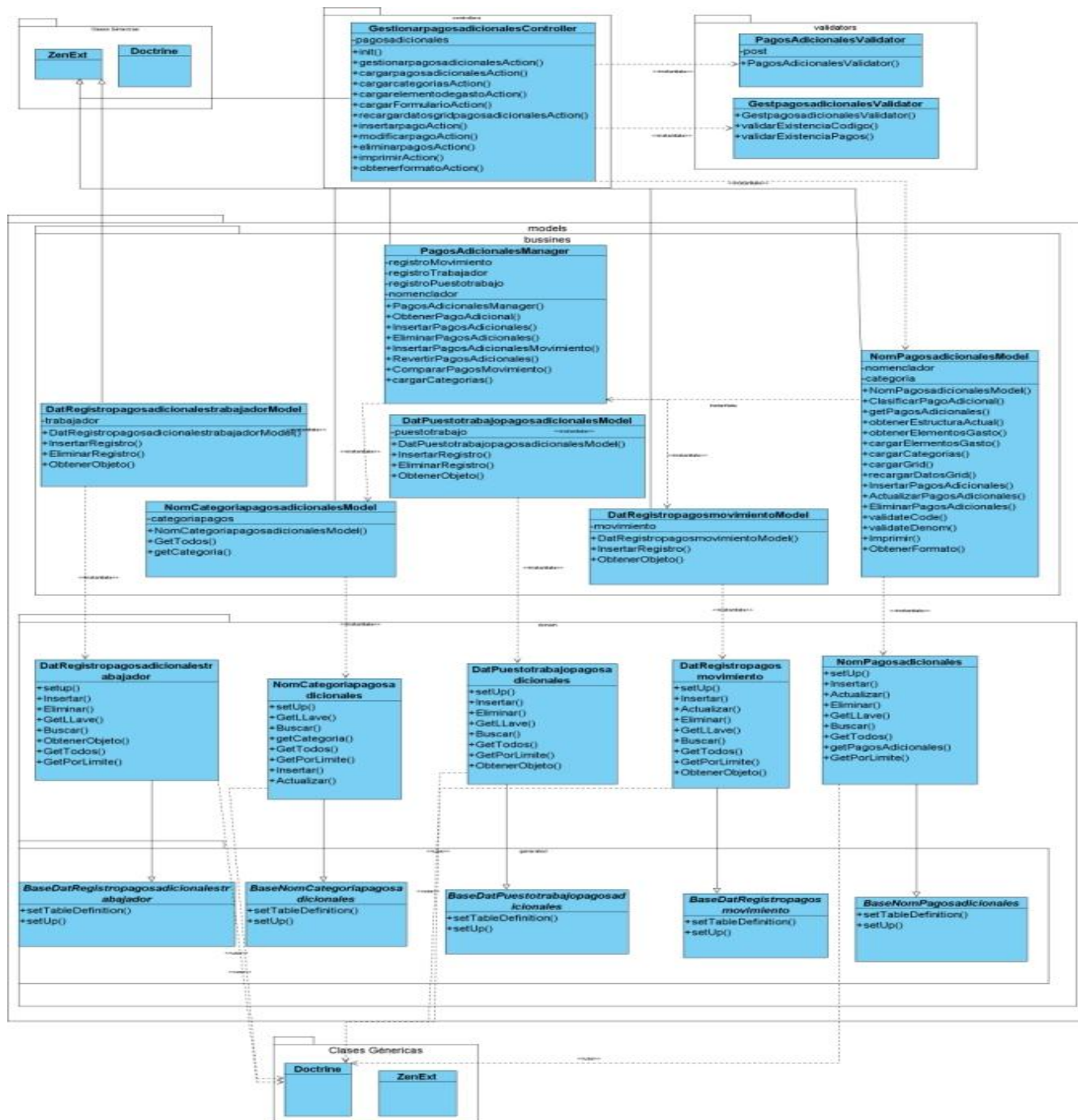


Figura 2.6 Diseño de clases del componente Pagos Adicionales.

Para seguir consultando las clases de los restantes componentes, consultar el Anexo 3

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

2.2.6 Matriz de Integración

La Matriz de Integración contiene todos los componentes definidos en el subsistema de forma matricial, donde en las intercepciones se especifican los servicios que brindan y consume cada componente. Tiene como objetivo tener registradas las relaciones entre componentes mediante servicios y saber cuáles son esos servicios. A continuación se muestra la matriz elaborada para los componentes Persona, Trabajador, Puesto de Trabajo, Pagos Adicionales, Incidencias, Nómina, Submayores Configuración, Régimen de Trabajo y Descanso y Cierre. Figura 2.7.

Componentes	Configuración	Brindan							
		Persona	Trabajador	Puesto de Trabajo	Régimen de trabajo y descanso	Pagos Adici	Incidencias	Nómina	Submayores Cierre
Configuración									
Persona			CHTrabajado						
Trabajador	CHDocument	CHPersona02		CHPuestoTrabajo01, CHPuestoTrabajo02, CHPuestoTrabajo03, CHPuestoTrabajo04		CHPagos01, CHPagos02, CHPagos03, CHPagos04, CHPagos05, CHPagos06	CHIncidencias07		CHSubmayores02, CHSubmayores03
Puesto de Trabajo	CHImpuesto01		CHTrabajador03		CHRegimen01	CHPagos01, CHPagos02, CHPagos04, CHPagos07			
Régimen de trabajo									
Pagos Adicionales									
Incidencias	CHImpuesto01		CHTrabajador02, CHTrabajador03	CHPuestoTrabajo04				CHNomina02, CHNomina04, CHNomina05	

Capítulo II Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Consumen	Nómina	CHImpuesto01	CHTrabajador02, CHTrabajador03			CHPagos03	CHIncidencias02, CHIncidencias03, CHIncidencias04, CHIncidencias05, CHIncidencias06, CHIncidencias08	CHSubmayores04, CHSubmayores06, CHSubmayores09, CHSubmayores10, CHSubmayores13, CHSubmayores14, CHSubmayores15, CHSubmayores16
	Submayores		CHPersona02, CHTrabajador03			CHPagos03	CHNomina03, CHNomina04, CHNomina06, CHNomina07	

Figura 2.7 Matriz de Integración.

Con este artefacto se tendrá una clara visión de la integración de cada uno de los componentes del subsistema Capital Humano y cuáles aún no están integrados. Esto ayudará a concebir un mejor trabajo por parte de los programadores, teniendo una idea de qué se tendrá que realizar.

De esta manera queda actualizado el artefacto Línea Base del subsistema Capital Humano. Culminando además con la realización de los artefactos necesarios para conformar la vista de sistema y dando cumplimiento a un objetivo específico.

CapítuloII Vista de la Arquitectura de Sistema del Subsistema Capital Humano

Conclusiones

En este capítulo se actualizó la vista arquitectónica de sistema del subsistema Capital Humano basada en los principales artefactos, los cuales son: Especificación del Componente, Complejidad y Criticidad, Matriz de Integración y el Documento Línea Base. Se determinó la prioridad para cada componente, se elaboró el mapa de componentes analizando la integración a través de los servicios que brindan o consumen los componentes, se mostró la estructura que poseen las clases del diseño del sistema así como sus relaciones por cada componente. De esta manera, se da cumplimiento a un objetivo específico planteado para dar solución a este trabajo.

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Introducción

En el capítulo I se brindan aspectos relacionados con la estructura que poseen los expedientes de CedruX y se actualizaron los artefactos de la vista de sistema de la arquitectura del subsistema Capital Humano en el capítulo II. Este capítulo III estará basado en actualizar los artefactos de la vista de la arquitectura de datos, que también conforman parte de dichos expedientes. De tal forma que al final del trabajo quede actualizada la arquitectura de datos y sistema de la línea y se logre trabajar a partir de una documentación para mejorar el entendimiento del equipo de trabajo.

3.1 Expediente de datos del subsistema Capital Humano

En el primer capítulo se especificó la estructura que poseen los expedientes tecnológicos del subsistema Capital Humano. En este epígrafe se muestra cómo están estructurados los expedientes de datos del subsistema Capital Humano (Figura 3.1), expedientes que recogen los artefactos a generar para lograr conformar la estructura correspondiente a la documentación. Estos artefactos son: La Matriz de Trazabilidad, Diccionario de Datos y Pruebas de Conceptos.



Figura 3.1 Estructura del expediente de datos del subsistema Capital Humano

3.2 Estándares

Existen una serie de estándares ya definidos en la vista de datos que son de suma importancia tenerlos en cuenta para lograr entender lo ya diseñado anteriormente, entre ellos sólo se analizarán los necesarios para seguir desarrollando este trabajo.

3.2.1 Normas de comentariado

Es una necesidad comentar todo lo que se realice dentro de la base de datos, es decir, establecer las pautas que conlleven a lograr un código más legible y reutilizable de manera que se pueda aumentar su mantenibilidad a lo largo del tiempo.

- Los comentarios serán tabulados en forma consecutiva y con el siguiente formato:
Descripción seguida del Nombre de la persona que la realiza. Ejemplo: *PostgreSQL COMMENT ON FUNCTION* "public". "prueba" () *IS* 'comentario de la función' "Pepe"; *COMMENT ON DOMAIN* "public"."domain1" *IS* 'comentario del dominio' "Pepe".
- Es importante ser correcto y no perder el tiempo, comentando solo lo necesario para transmitir la idea. Los comentarios deben ser simples y directos.
- Se comentará mientras se programe, para lograr eliminar mayores costos en el tiempo.
- Los comentarios se actualizarán al mismo tiempo que el código.
- En caso de que se quiera emitir algún comentario especial en cualquier segmento del código utilizar: /* Comentario */.

Este estándar apoyará el desarrollo del trabajo al brindar la estructura utilizada por los programadores para documentar el código. Entender el mismo elimina cualquier ambigüedad que pueda surgir referente al código documentado.

3.2.2 Estándares de nomenclatura

- **Apariencia de los esquemas:** El nombre del esquema se debe de escribir completo en minúscula, comenzando por el prefijo "mod" que significa módulo, seguidamente un guión bajo "_" y finalmente el nombre del módulo.
Ejemplo: mod_capitalhumano.

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

- **Nombres de las tablas:** El nombre que se empleará en cada tabla debe escribirse con todas las letras en minúscula y consecutivamente sin tener la presencia de guiones u otros caracteres especiales. Además, debe ser escrito de forma sencilla pero al mismo tiempo explícito, tal que con sólo leerlo se reconozca el propósito de la misma.

Ejemplo: dat_persona.

Como se observa antes del nombre de cada tabla se añade un prefijo, el cual es diferente en dependencia del tipo de tabla que sea. Los prefijos a utilizar son los siguientes:

dat_: Se utiliza en tablas que almacenan la mayor cantidad de datos de la entidad.

nom_: Se utiliza en tablas nomencladoras.

his_: Se utiliza en tablas que almacenan datos por largos períodos de tiempo y que sólo son utilizados para análisis esporádicos. (Tablas Históricas).

seg_: Se utiliza en tablas que almacenan control de acceso de usuarios y opciones de acceso de uno o varios sistemas. (Tablas de Seguridad)

conf_: Se utiliza en tablas que almacenan parámetros de configuración del sistema. (Tablas de Configuración)

tmp_: Se utiliza para tablas que almacenan datos transitorios. (Tablas Temporales)

res_: Se utiliza para las tablas resúmenes, empleadas en los reportes.

- **Estructura de los campos:** Los nombres se escribirán completamente en minúscula y al igual que el nombre de las tablas sin presentar guiones ni caracteres especiales. Debe ser explícito de forma tal que con sólo leerlo se comprenda el propósito del mismo. Cada campo deberá tener una descripción del mismo, de forma que se explique de forma sencilla cuál será su función en la tabla. En caso de que sea una llave primaria se le antepondrá el prefijo “**id**” y seguidamente el nombre del campo, así mismo si es llave foránea.

Ejemplo

“**idpersona**”:

Identificador de la persona.

“**idcolorpiel**”:

Identificador del color de piel.

“**nombre**”:

Nombre de la persona.

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

“**edad**”: Edad de la persona.

- **Nombre de las llaves primarias:** Los nombres de las restricciones se escribirán con minúscula. Comenzando con el identificador (id) seguido del nombre de la tabla.

Ejemplo: **idpersona**: Llave primaria de la tabla “**dat_persona**”.

- **Nombre de las llaves foráneas:** Los nombres de las llaves foráneas se escriben con minúscula.

Ejemplo: **idcolorojos**: Llave foránea de la tabla “**dat_colorojos**”:

- **Nombres de las funciones, triggers (disparadores), tipos de datos y vistas:** El nombre debe ser igual que en el caso de los nombres de las tablas y de los campos explícitos, de forma tal que con sólo leerlo se reconozca el propósito del mismo. Al igual que en caso de las tablas se utilizarán prefijos según el caso específico.

f_: Se utiliza en el caso de las funciones.

ft_: Se utiliza en el caso de las funciones utilizadas en los **triggers**.

t_: Se utiliza en el caso de los **triggers**.

En el caso de los **triggers** usados para manejar la concurrencia, los mismos estarán formulados de la siguiente forma: empezarán con “**t_**” puesto que son **triggers**, pero seguidamente le pondrá la palabra “**UPDATE**”, para de esta forma reconocer que el mismo lo que hace es actualizar el campo utilizado para la concurrencia y posteriormente el nombre del mismo.

td_: Se utiliza en el caso de los tipos de datos.

v_: Se utiliza para el caso de las vistas.

- **Nombre de los tablespace:** El nombre se definirá todo en minúscula, empezando con la letra “**tbs**”, seguido un guión bajo y el nombre del módulo junto en minúscula.

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Ejemplo:

d_modcapitalhumano: Dominio que pertenece al módulo Capital Humano.

- **Nombres de las secuencias:** Los nombres de las secuencias empezarán con el prefijo “**sec**” seguidamente tendrán un guión bajo “_”, posteriormente le continuará el nombre de la tabla a la que pertenece dicha secuencia, después del nombre de la tabla se le añadirá un guión bajo “_”, acompañado del sufijo “**seq**”.

Ejemplo:

“**sec_persona_seq**”

“**sec_trabajador_seq**”


El estándar trata todo lo referente a las abreviaturas de los nombres del modelo de datos del subsistema. Brinda la estructura que poseen los nombres de tablas, atributos y triggers, evitando ambigüedades y ofreciendo un claro entendimiento para el desarrollo de este trabajo.

La principal importancia de la vista de datos de la arquitectura, es que brinda en forma de plantilla una serie de pasos comunes a seguir por los arquitectos de datos, en pro de lograr uniformidad en el momento de realizar el diseño del modelo de datos o de modificarlo. Todos estos estándares ya definidos, poseen gran importancia en el momento de actualizar los artefactos de la vista de datos, pues permiten entender claramente la estructura de los componentes que conforman el subsistema Capital Humano.

















3.3 Descripción de las clases persistentes de Capital Humano

Después de un profundo análisis y discusión por parte de los analistas y especialistas funcionales, en conjunto con el equipo de datos y a partir de los diagramas de clases del diseño se identificaron las clases persistentes. A continuación se mencionarán dichas clases así como su propósito.





















Tabla 3.1 Descripción de las clases persistentes de Capital Humano.

Nombre	Documentación
 nom_gradoescolaridad	Almacena todo lo referente a los niveles escolares que existen. Ejemplo: 6to grado (primario), 9no grado



















Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

	(Secundario), 12mo (preuniversitario), superior o universitario.
 his_hispagosadicionales	Almacena el historial de los pagos adicionales
 nom_gruposanguineo	Almacenan los grupos sanguíneos conocidos hasta el momento
 nom_niveleducacional	Almacena los niveles escolares que puede tener una persona
 his_hisregistroretenciones	Almacena un historial del registro de retenciones
 nom_titulo	Almacena los tipos de títulos que existen
 dat_niveleducacional	Almacena el nivel educacional de una persona
 his_trabajadores	Almacena el historial de trabajadores
 his_hissubmayorvac	Almacena un historial de submayores de vacaciones
 nom_centroestudios	Almacena los centros de estudios existentes
 nom_categoriadocente	Almacena las categorías docentes que puede tener una persona.
 dat_chdirecciones	Almacena las direcciones de las personas
 nom_operacionsub	Almacena los tipos de operaciones que se hacen sobre los submayores
 nom_tiporeubicacion	Almacena los tipos de reubicación que puede tener un trabajador
 nom_tipocontacto	Almacena los tipos de contactos (personal, trabajo, provisional, otros)
 nom_tipomediocontacto	Almacena los tipos de medios de contactos (teléfono, fax, correo, beeper)
 dat_contacto	Almacena los contactos de una persona
















Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

 dat_personaespecialidad	Almacena la especialidad de la persona
 dat_personanacimiento	Almacena todos los datos del nacimiento de una persona
 dat_chfonetica	Almacena la fonética del nombre de las personas (sonido)
 dat_foto	Almacena las fotos de las personas
 nom_categoriapersona	Almacena las categorías de una persona (externo, interno)
 dat_persona	Almacena los datos de las personas. Datos como: nombre, apellidos, sexo, datos sobre características físicas, número de identidad, fecha de nacimiento, lugar de nacimiento, grado científico, nivel escolar, entre otros.
 dat_caracteristicaspersonales	Almacena las características personales
 nom_sexo	Almacena los sexos (masculino y femenino)
 dat_estadocivilpersona	Almacena el estado civil de la persona (soltero, casado)
 dat_submayorvac	Almacena el acumulado en día y en importe de cada trabajador
 nom_colores	Almacena los colores existentes
 dat_personatallaje	Almacena las tallas de las personas
 nom_estadocivil	Almacena el estado civil de la persona. Ejemplo: casado o soltero.
 dat_personaresidencia	Almacena los datos de la residencia de la persona
 nom_categoriacolores	Almacena la relación de colores y categorías
 nom_categoriacolor	Almacena las categorías del color (piel, pelo, ojos)
 nom_valorestalla	Almacena las tallas existentes de la persona
 nom_estatuvivienda	Almacena el estado de la vivienda (particular, vinculada)
 nom_tipodireccion	Almacena el tipo de dirección (rural, urbana)
 dat_personadocidentidad	Almacena todos los documentos de identificación de una persona



Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

 nom_tiporesidencia	Almacena los tipos de residencia (provisional, oficial)
 nom_formaconvivencia	Almacena los datos de convivencia de la persona (agregado, propietario, conviviente, asignado)
 nom_tipoperiodopago	Almacena el tipo de período de pago (mensual, quincenal)
 nom_tipovinculacion	Almacena el tipo de vinculación (producción, servicios, apoyo, administración)
 dat_datoscontables	Almacena los datos contables de los trabajadores
 dat_trabajador	Almacena los trabajadores
 nom_tipodocidentidad	Almacena los tipos de documentos de identidad (CI, pasaporte, carné del MININT, carné de las FAR)
 nom_tipodocumento	Almacena el tipo de documento que se genera en el sistema (contrato, movimiento de nómina, pre-nómina)
 nom_tipomovnomina	Almacena los tipos de movimientos de nómina que se pueden hacer
 dat_expedientetrabajador	Almacena los datos relacionados a los documentos generados por el sistema asociados al trabajador
 dat_documento	Almacena los documentos
 dat_modelomovnomina	Almacena el modelo de movimiento de nómina
 nom_detallesmovimiento	Almacena los detalles de un motivo de un movimiento de nómina
 nom_motivomovnomina	Almacena los motivos de movimiento de nómina
 dat_contrato	Almacena los datos relacionados a un contrato de un trabajador
 nom_estado	Almacena el estado de los documentos: contrato, movimiento de nómina, trabajador (cerrado, confirmado, activo, inactivo, revertido)
 nom_impuesto	Almacena los impuestos
 dat_puestotrabajo	Almacena los puestos de trabajo existentes en la empresa

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

 nom_sistemapago	Almacena los tipos de pagos existentes
 dat_puestogrupo	Almacena la asignación de un puesto de trabajo a un grupo de puesto de trabajo
 dat_registropagosmovimiento	Almacena los pagos adicionales asociados a un movimiento de nómina
 dat_regimenconfig	Almacena la configuración del régimen de trabajo
 dat_puestotrabajopagosadicionales	Almacena los pagos adicionales asociados al puesto de trabajo
 nom_grupopuestotrabajo	Almacenan grupos de puestos que sean similares
 dat_regimentrabajodescanso	Almacena la secuencia de trabajo y descanso de un trabajador
 dat_regimendia	Almacena el régimen de trabajo por día de un trabajador
 dat_regimensesion	Almacena el régimen de una sección de un trabajador
 nom_tiporelacionlaboral	Almacena las relaciones laborales existentes (contrato, elección, designación)
 nom_tipocontrato	Almacena los diferentes tipos de contrato que puede tener un trabajador
 dat_registropagosadicionalestrabajador	Almacena los pagos adicionales asociados al trabajador
 nom_pagosadicionales	Almacena los pagos adicionales
 nom_categoriapagosadicionales	Almacena las categorías que puede tener un pago adicional
 nom_gradocientifico	Almacena los grados científicos que puede tener una persona.
 nom_tipoincidencia	Almacena los diferentes tipos que pueden ser las incidencias

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

 nom_extrasocial	Almacena las procedencias sociales que se conocen. Ejemplo: (obrero, campesino, etc.)
 nom_configuracionsubmayorvac	Almacena la configuración (identificador de la cuenta y porcentaje) del submayor de vacaciones

Esta estructura ayudará a entender lo que ya está realizado, dando un margen de cómo están estructuradas las clases persistentes y exponiendo una pequeña descripción de las mismas, que posteriormente se abordarán. Con este análisis se podrá desplazar por cada una de las tablas teniendo una lógica de lo que se está haciendo, sin ambigüedades. Esto facilitará el entendimiento del Modelo de Datos

3.4 Modelo de Datos

El Modelo de Datos de la línea Capital Humano estaba compuesto por 67 tablas, cada una representa una clase persistente. Además con 35 tablas nomencladores que son utilizadas para agrupar datos con características similares ya sean gestionables o no. Otras 26 tablas de datos utilizadas para el almacenamiento de los datos que se gestionan en las empresas y 6 tablas que figuran historiales, donde se almacenan datos históricos de seis de las tablas de datos, con el objetivo de no tener que manipular todos los datos en cada consulta sobre las tablas de las cuales guardan su historial.

Actualmente y debido a los cambios mencionados en la introducción a este trabajo en beneficio de una buena gestión, el modelo de datos del subsistema Capital Humano sufrió grandes cambios. Debido a eso queda estructurado por 76 tablas donde 42 son nomencladores y otras 30 de datos gestionables en la empresa, además de 4 tablas que figuran historiales. Estos datos fueron actualizados en el artefacto: Diccionario de Datos tratado en el próximo epígrafe.

3.5 Diccionario de Datos

El objetivo de este artefacto es tener un listado organizado de todos los datos que pertenecen al subsistema Capital Humano. Este recoge una breve descripción de cada una de las tablas, así como los atributos que las componen y las relaciones con otras tablas del subsistema, especificando la cardinalidad y la multiplicidad. Además en este artefacto se describen los disparadores (Tabla 3.2) y las funciones por las cuales está conformada la base de datos (Tabla 3.3), lo que ayuda al programador en gran escala al brindarle la estructura y la funcionalidad que posee el código.

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Tabla 3.2 Descripción de disparadores

Nombre	Documentación
t_hissubmayorvac	Disparador que después de insertar actualizar o eliminar en la tabla dat_submayorvac ejecuta la función ft_hissubmayorvac ()
t_histpagosadicionales	Disparador que después de insertar actualizar o eliminar en la tabla nom_pagosadicionales ejecuta la función ft_histpagosadicionales ()
t_updatedatoscontables	Disparador que antes de actualizar en la tabla dat_regimendia ejecuta la función ft_chequear ()
t_updatedia	Disparador que antes de actualizar en la tabla dat_datoscontables ejecuta la función ft_chequear ()
t_updatecontrato	Disparador que antes de actualizar en la tabla dat_contrato ejecuta la función ft_chequear ()
t_updatecontacto	Disparador que antes de actualizar en la tabla dat_contacto ejecuta la función ft_chequear ()
t_updatecaracteristicaspersonales	Disparador que antes de actualizar en la tabla dat_caracteristicaspersonales ejecuta la función ft_chequear ()
t_submayorvac	Disparador que antes de actualizar en la tabla dat_submayorvac ejecuta la función ft_chequear ()
t_updateexpedientetrabajador	Disparador que antes de actualizar en la tabla dat_expedientetrabajador ejecuta la función ft_chequear ()
t_updateniveleducacional	Disparador que antes de actualizar en la tabla dat_niveleducacional ejecuta la función ft_chequear ()
t_updatepersonaespecialidad	Disparador que antes de actualizar en la tabla dat_personaespecialidad ejecuta la función ft_chequear ()

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

t_updatepersonaidentidad	Disparador que antes de actualizar en la tabla dat_personadocidentidad ejecuta la función ft_chequear ()
t_updatepersonaresidencia	Disparador que antes de actualizar en la tabla dat_personaresidencia ejecuta la función ft_chequear ()
t_updatepuestogrupo	Disparador que antes de actualizar en la tabla dat_puestogrupo ejecuta la función ft_chequear ()

Tabla 3.3 Descripción de la función f_listado_regimentrabajodescanso

f_listado_regimentrabajodescanso (codigoone : varchar, usuario : varchar)	
Documentación	<p>Función que dado un código de estructura retorna el listado del régimen de trabajo y descanso de los trabajadores. Mostrando:</p> <ul style="list-style-type: none"> Entidad Código de la entidad Régimen Cadena Hora Dirección Teléfono Correo Usuario
Esquema	mod_capitalhumano
Declaración	<pre>CREATE FUNCTION f_listado_regimentrabajodescanso (codigoone varchar, usuario varchar) RETURNS AS ' DECLARE regimen record; estructura numeric; temp record; numerofila numeric;</pre>

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

```
BEGIN

select e.idestructura into estructura
from mod_estructuracomp.dat_estructura e
where e.codigo = codigoone
union
select p.idestructuraop
from mod_estructuracomp.dat_estructuraop p
where p.codigo = codigoone;

temp = mod_capitalhumano.f_datosestructura (estructura);
select setval ('mod_capitalhumano.sec_numerar_seq',0) into numerofila;

FOR regimen in
(
SELECT
nextval ('mod_capitalhumano.sec_numerar_seq')::numeric as NroDeOrden,
estr.denominacion as "Entidad",
estr.codigo as "Codigo",
drtd.denom as "regimenTD",
ARRAY (select (case when drd1.trabajo = 1 then 'T'
when drd1.trabajo = 0 then 'D' end)::varchar from
mod_capitalhumano.dat_regimentrabajodescanso drd1
inner join mod_capitalhumano.dat_regimendia drd ON drtd.idregimentrabajodescanso =
drd1.idregimentrabajodescanso
WHERE drtd.denom = drd1.denom) as "cadena",
sum (drd.jornada),
temp.telefono,
temp.direccion,
temp.correo,
usuario


from mod_capitalhumano.dat_regimentrabajodescanso drtd
inner join mod_capitalhumano.dat_regimendia drd ON drtd.idregimentrabajodescanso =
drd.idregimentrabajodescanso
inner join mod_estructuracomp.dat_estructura estr on drtd.idestructuracomun =
estr.idestructura
where
```

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

	<pre> drtd.idestructuracomun = estructura AND drtd.fechaeliminado is null GROUP by estr.denominacion, estr.codigo, drtd.denom)Loop return next regimen; end loop; END ' LANGUAGE plpgsql </pre>
Nombre del Procedimiento	f_listado_regimentrabajodescanso
Comentario	descripción
Tipo de Procedimiento	Devoluciones de Resultados
Modelo de Datos	2

Cada una de las tablas que conforman la base de datos del subsistema Capital Humano, también generan una descripción en este artefacto como se especificó anteriormente. Esto mejorará el entendimiento de la estructura del modelo de datos. A continuación se describen algunas tablas del subsistema como son `dat_persona` y `dat_trabajador`, además se describen algunas relaciones de cada una de estas clases.

Tabla 3.4 Descripción de la tabla `dat_persona`

Nombre	Valor
Modelo de Datos	Físico
Clase asignada	 <code>Dat_persona</code>
Documentación	En esta tabla se guardan los datos de las personas. Datos como: nombre, apellidos, sexo, datos sobre características físicas, número de identidad, fecha de nacimiento, lugar de nacimiento, grado científico, nivel escolar, entre otros.
Esquema	<code>mod_capitalhumano</code>

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Llave Primaria	dat_persona_pkey
----------------	------------------

Tabla 3.5 Descripción de atributos de la tabla dat_persona

Nombre	Tipo de Dato	Limitaciones	Anulable	Documentación
idpersona	numeric (19)	PK	No	Identificador de la persona
numid	varchar (11)		No	Carné identidad de la persona
nombre	varchar (30)		No	Nombre de la persona
papel	varchar (30)		No	Primer apellido de la persona
sapel	varchar (30)		No	Segundo apellido de la persona
idsexo	numeric (19)	FK (nom_sexo.idsexo)	No	Identificador del sexo de la persona
version	numeric (6)		Sí	Utilizado para la concurrencia
idcategoriapersona	numeric (19)	FK (nom_categoriapersona.idcategoriapersona)	Sí	Identificador de la categoría de la persona
fechaeliminado	date (0)		Sí	Se pone fecha de eliminado para no eliminar físicamente a la persona de la base de datos

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

segundonombre	varchar (30)		Sí	Segundo nombre de la persona
---------------	--------------	--	----	------------------------------

Tabla 3.6 Descripción de relación entre la tabla dat_persona y nom_sexo



dat_persona_fk1 : Relación	
Desde	 nom_sexo
Documentación	Relación de uno a muchos, entre las tablas dat_persona y nom_sexo con idpersona como llave primaria en dat_persona y llave foránea en nom_sexo. Donde cada persona tiene un determinado grupo de sexo
Identificación	Falso
En Eliminar	No Acción
En Actualización	No Acción
Multiplicidad a	0..*
Multiplicidad de	1
Asociación	0
Modelo de Datos	2

Tabla 3.7 Descripción de relación entre la tabla dat_persona y nom_categoriapersona

dat_persona_fk2 : Relación	
Desde	 nom_categoriapersona
Documentación	Relación de uno a muchos, entre las tablas dat_persona y nom_categoriapersona con idpersona como llave primaria en dat_persona y llave foránea en nom_categoriapersona. Donde cada persona tiene un determinado grupo categoría de persona
Identificación	Falso
En Eliminar	No Acción
En Actualización	No Acción
Multiplicidad a	0..*
Multiplicidad de	0..1
Asociación	0
Modelo de Datos	2

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Tabla 3.8 Descripción de la tabla dat_trabajador


Nombre	Valor
Modelo de Datos	Físico
Clase Asignada	 Dat_trabajador
Documentación	Almacena todos los datos de los trabajadores
Esquema	mod_capitalhumano
Llave Primaria	dat_trabajador_pkey

Tabla 3.9 Descripción de atributos en la tabla dat_trabajador

Nombre	Tipo de Dato	Limitaciones	Anulable	Documentación
idpersona	numeric (19)	FK (dat_persona.idpersona)	No	Identificador de persona
idpuestotrabajo	numeric (19)	FK (dat_puestotrabajo.idpuestotrabajo)	No	Identificador de puesto de trabajo
idestado	numeric (19)	FK (nom_estado.idestado)	No	
idtrabajador	numeric (19)	PK	No	Identificador de trabajador
numexpediente	varchar (11)		No	Número de expediente
fechaeliminado	date (0)		Sí	Se utiliza para no eliminar físicamente de la base de datos
version	numeric (6)		Sí	Se utiliza para la concurrencia
salariotrabajador	numeric (8)		No	Salario escala del

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

				trabajador
principal	numeric (1)		Sí	
idtipovinculacion	numeric (19)	FK (nom_tipovinculacion.idtipovinculacion)	No	
fechainicioregimen	date (0)		Sí	
idtipoperiodopago	numeric (19)	FK (nom_tipoperiodopago.idtipoperiodopago)	No	

Tabla 3.10 Descripción de relación entre tabla dat_trabajador y dat_submayorvac



fkdat_submay719895 : Relación	
A	 dat_submayorvac
Documentación	Relación de uno a muchos, entre las tablas dat_trabajador y dat_submayorvac con idtrabajador como llave primaria en dat_trabajador y llave foránea en dat_submayorvac
Identificación	Falso
En Eliminar	No Acción
En Actualización	No Acción
Multiplicidad a	0..*
Multiplicidad de	1
Asociación	0
Modelo de Datos	2

Tabla 3.11 Descripción de relación entre tabla dat_trabajador y nom_contrato

dat_contrato_fk3 : Relación	
A	 dat_contrato
Documentación	Relación de uno a muchos, entre las tablas dat_trabajador y dat_contrato con idtrabajador como llave primaria en dat_trabajador y llave foránea en dat_contrato
Identificación	Falso
En Eliminar	No Acción

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

En Actualización	No Acción
Multiplidad a	0..*
Multiplidad de	1
Asociación	0
Modelo de Datos	2

Para continuar consultando las descripciones de las tablas de este artefacto consultar el Anexo 4

3.6 Matriz de Trazabilidad

La Matriz de Trazabilidad es uno de los artefactos que generan los arquitectos de datos. Se desarrolla por la presencia de datos en las clases que son de subsistemas diferentes. En el caso específico de la línea de Capital Humano, la Matriz de Trazabilidad es de gran ayuda, pues este subsistema necesita información que se encuentra representada en otros subsistemas. Estos datos se utilizan en más de un componente y en más de una clase dentro de cada componente. En esto radica su importancia y aplicabilidad, ya que sirve para tener el control de cuál subsistema proviene cada dato y dónde se utiliza. Además, sirve de guía al arquitecto de datos para el desarrollo o mantenimiento de la base de datos. Para realizar la matriz se analizaron cada una de las tablas la base de datos con que cuenta el subsistema Capital Humano y su relación con tablas de otros subsistemas. De esta forma, se elabora la Matriz de Trazabilidad que se muestra a continuación en la tabla 3.2.

Tabla 3.12 Matriz de Trazabilidad del subsistema Capital Humano

Esquemas Entrada→	Capital Humano				
	tabla origen	Variable	tabla destino	valor por defecto	estado de integración
Esquemas Salida↓	nom_filaestruc	idestructur acomun	dat_submayorvac	no null	hecho
	nom_filaestruc	idestructur acomun	dat_regimentrabajodescanso	no null	hecho

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Estructura y Composición	nom_filaestruc	idestructuracomun	dat_submayorvac	no null	hecho
	nom_filaestruc	idestructuracomun	nom_configuracionsubmayorvac	no null	hecho
	nom_filaestruc	idestructuracomun	dat_trabajador	no null	hecho
	nom_filaestruc	idestructuracomun	nom_pagosadicionales	no null	hecho
Nomencladores	nom_dpa	iddpa	dat_chdirecciones	no null	no hecho
	nom_especialidad	idpespecialidad	dat_personanaespecialidad	no null	hecho
	nom_dpa	iddpa	dat_personanacimiento	no null	hecho
	nom_pais	idpais	dat_personanacimiento	no null	hecho
	nom_dpa	iddpa	dat_personaresidencia	no null	hecho
	dat_elementocomun	idelementocomun	nom_pagosadicionales	no null	hecho
	dat_elementocomun	idelementocomun	nom_impuesto	no null	hecho
Contabilidad	nom_cuenta	idcuenta	nom_impuesto	no null	No hecho

La Matriz de Trazabilidad es de gran importancia, pues brinda la posibilidad de agrupar todos los atributos que no son propios del subsistema Capital Humano, además permite conocer a qué subsistema pertenece dicho atributo.

Para consultar el documento íntegro de la Matriz de Trazabilidad, consultar el Anexo 5

3.7 Pruebas de Conceptos

Las Pruebas de Conceptos constituyen una etapa absolutamente crítica en el lanzamiento de nuevos productos. En ellas se pueden utilizar enfoques cualitativos o cuantitativos. Estos últimos sin embargo, resultan especialmente relevantes para la investigación de mercados que pretenden hacer predicciones. (8) En este artefacto se analizan todas las funciones o consultas que dan respuesta a los requisitos funcionales del sistema. Además, por cada función o consulta se midieron parámetros como: el tiempo de ejecución, las filas devueltas y el costo de la función.

Para el llenado de los datos se utilizó la herramienta *EMS Generador de datos 2005 para PostgreSQL*, la cual constituye una herramienta poderosa en la generación de datos de pruebas para base de datos construidas en PostgreSQL. Permite seleccionar las tablas para la generación de los datos, definir para cada campo el rango de valores admisibles, además permite llenar la base de datos de forma eficiente y rápida teniendo en cuenta la integridad referencial de los datos.

Se generaron para la realización de las pruebas de concepto un volumen de 5000 tuplas para cada una de las tablas de datos. A continuación se mostrarán las pruebas realizadas a las funciones:

f_listado_regimentrabajodescanso: la cual retorna un listado del régimen de trabajo y descanso y
f_listado_persona: la cual retorna un listado de personas.

Requisito 1: Listar régimen de trabajo y descanso

Función: f_listado_regimentrabajodescanso (codigoone character varying, usuario character varying)

Descripción: Función que retorna un listado del régimen de trabajo y descanso dado un código de estructura.

Tablas relacionadas:

mod_capitalhumano.dat_regimentrabajodescanso

mod_capitalhumano.dat_regimendia

mod_estructuracomp.dat_estructura

Resultados:

Costo=0.00..260.00

Total de filas=5035

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

Filas devueltas = 608

Tiempo de ejecución total: 253.523 ms

Requisito 2: Listar personas

Función: f_listado_persona (usuario : varchar)

Descripción: Función que retorna un listado de personas

Tablas relacionadas:

mod_capitalhumano.dat_persona.

mod_capitalhumano.nom_categoriapersona.

mod_capitalhumano.nom_sexo.

Resultados:

Costo=0.66..497.64

Filas=5090

Filas devueltas = 5090

Tiempo de ejecución total: 355.303 ms

Con las pruebas de conceptos realizadas se trata de comprobar que el diseño de la base de datos cumple con tiempos de respuesta bajos, lo que constituye la base para garantizar que la misma funcione eficientemente y evitar que se afecten los atributos de calidad. Con los resultados alojados en el artefacto, se pudo comparar que los tiempos de respuesta de ejecución de las consultas de la base de datos, son significativamente bajos, debido a la eficiencia de cada una de ellas.

Para consultar el artefacto en su composición completa consultar el Anexo 6.

Conclusiones

En este capítulo se analizaron algunos estándares necesarios para actualizar la descripción de la vista de datos. Se describieron las clases persistentes, describiéndose una serie de atributos que pertenecen al

Capítulo III Vista de Arquitectura de Datos del Subsistema Capital Humano

subsistema Capital Humano. Además, como artefacto de la vista de datos se actualizó la Matriz de Trazabilidad, que garantiza saber en todo momento qué información se maneja en las tablas provenientes de otros subsistemas. El Diccionario de Datos que describe todo lo referente a tablas atributos, relaciones, triggers y funciones. Además, del artefacto Pruebas de Conceptos donde se calcula el costo de las funciones y el tiempo de ejecución, conformando así la vista de la arquitectura de datos y dando cumplimiento a un objetivo específico.

Conclusiones generales

- Se realizó un estudio acerca de los principales elementos de descripción arquitectónica, sentando las bases para realizar una buena actualización de las vistas de datos y sistema del subsistema Capital Humano.
- Se actualizaron los artefactos: Línea Base del Subsistema, Complejidad y Criticidad, Especificación de Componentes y la Matriz de Integración. Quedando conformada la vista de sistema del subsistema Capital Humano.
- Se actualizaron los artefactos: Diccionario de Datos, Matriz de Trazabilidad y las Pruebas de Conceptos del Diseño. Quedando conformada la vista de datos del subsistema Capital Humano.

Recomendaciones

Después de concluido este trabajo se recomienda:

- Analizar cada uno de los artefactos generados tanto de datos como de sistema, de tal manera que su uso favorezca a la línea de vida del proyecto y facilite el trabajo de los desarrolladores.
- Actualizar cada cambio que surja a medida que el desarrollo del proyecto avance para lograr una descripción que respalde todo el desarrollo del software.

Referencias bibliográficas

1. Reynoso. (2004). Arquitectura de Software. Recuperado el 4 de 2010, de <http://carlosreynoso.com.ar/arquitectura-de-software/>
2. Pressman. (2002). software engineering pressman solutions. Recuperado el 4 de 2010, de <http://www.vagos.es/showthread.php?t=537302>
3. Garlan. (6 de 2000). arquitectura de software. Recuperado el 4 de 2010, de An Introduction to Software Architecture: http://www.worldlingo.com/ma/enwiki/es/Software_architecture
4. Claments. Documenting Software Architectures.
5. Kruchten. (1995). The 4+1 View Model of Software Architecture Software, IEEE Institute of Electrical and Electronics Engineers.
6. Gómez, D. (11 de 2008). Cómo documentar la arquitectura de software basado en (Fontdevila, 2003). Recuperado el 4 de 2010, de <http://www.dosideas.com/noticias/metodologias/298-como-documentar-la-arquitectura-de-software.html>
7. Fo (Cleveland, 1995)a. (2003). reglas para una buena documentación basadas en [Clements 2003]. Recuperado el 4 de 2010, de <http://materias.fi.uba.ar/7500/fontdevilla-tesisingenieriainformatica.pdf>
8. Paradigm, V. (2006). isual-paradigm.com. Recuperado el 4 de 2010, de <http://www.visual-paradigm.com/product/vpuml>.
9. Oduardo Gómez, L. B. (2008). Análisis y diseño del Componente del Sistema de Ingresos dentro del ERP Posta. La Habana: Universidad de las Ciencias Informáticas: s. n., 2008.
10. Alagón, J. (17 de 8 de 2001). El Papel de las Pruebas de Conceptos en el Lanzamiento de Nuevos Productos. Recuperado el 4 de 2010, de <http://segmento.itam.mx/Administrador/Uploader/material/Pruebas%20de%20Concepto.PD>

Bibliografía

- 1 Ávila, F. A. (2008). La Gestión Documental en Cuba. Recuperado el 4 de 2010
- 2 Cáceres, A. M. (21 de 7 de 2006). Facultad de ingeniería, Universidad de Don. Recuperado el 4 de 2010, de <http://www.galeon.com/rcruz0423/docs/case.pdf>.
- 3 Cleveland, G. (2005). Overview of Document Management Technology. Obtenido de Ottawa, Canadá.
- 4 documental. R. d. (s. f.). sitio Revista de gestión documental. Recuperado el 4 de 2010, de <http://www.revistagestiondocumental.com/2009/05/21/la-biblioteca-virtual-miguel->
- 5 Espinoza, H. (2005). Una alternativa de DBMS Open Source. Recuperado el 4 de 2010, de PostgreSQL.
- 6 Gómez-Guillamón, F. (s. f.). La gestión documental y la norma ISO 15489: 2001. Recuperado el 4 de 2010, de La gestión documental y la norma ISO 15489: 2001.
- 7 González, E. (2007). Obtenido de <http://www.idg.es>. <http://www.idg.es>.
- 8 León, F. R. (3 de 8 de 2009). Recuperado el 4 de 2010, de <http://phylevn.mexrom.net>
- 9 (Gómez-Guillamón, 2005) <t/data/files/files/SQLiteFonasol.pdf>.
- 10 PostgreSQL. P. d. (8 de 2009). PostgreSQL. Recuperado el 4 de 2010, de Pagina del SGBD PostgreSQL: <http://www.postgresql.org/>.
- 11 web. M. d. (26 de 10 de 2007). Maestros del web. [Online]. Recuperado el 4 de 2010, de <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
11. Cleveland, G. (1995). Overview of Document Management Technology. Overview of Document Management Technology. Canadá, Ottawa: s. n., 1995.
12. Gómez-Guillamón, F. (2005). La gestión documental y la norma ISO 15489: 2001 Record Management.
13. Hofmeister, C. (1999). Describing Software Architecture with UML. San Antonio: En Proceedings of the First Working IFIP Conference on Software Architecture, IEEE Computer Society Press.
14. Pfleeger, S. L. (2002). Ingeniería de Software: Teoría y Práctica. Madrid.
15. Platt, M. (2002). microsoft.com. Recuperado el 4 de 2010, de <http://msdn.microsoft.com/architecture/default.aspx?pull=/library/en-us/dnea/html/eaarchover.asp>
16. Reynoso, C. B. (2 de 2004). Willy Dev. Recuperado el 5 de 2010, de Universidad de Buenos Aires: www.willydev.net/descargas/prev/IntroArq.pdf
17. SEI, S. E. (s. f.). Software Engennering Institute. Recuperado el 5 de 2010, de Carnegie Mellon University: <http://www.sei.cmu.edu/>

18. Shaw, M. (IEEE Transactions on Software Engineering, 1995.). Abstractions for Software Architecture and Tools to Support Them.
19. Steve, V. (1993). A cursory overview and comparison of four Architecture Description.

Anexos

Anexo 1: Complejidad y Criticidad

Este documento se puede consultar en su versión original en la siguiente dirección.

[http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente de la AS/Línea base de Subsisistema/Líneas base/Línea base de Capital Humano/Complejidad Criticidad.xls](http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente_de_la_AS/Línea_base_de_Subsisistema/Líneas_base/Línea_base_de_Capital_Humano/Complejidad_Criticidad.xls)

Anexo 2: Especificación de Componentes

Este documento se puede consultar en su versión original en la siguiente dirección.

[http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente de la AS/Línea base de Subsisistema/Líneas base/Línea base de Capital Humano/Especificación Componente.xls](http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente_de_la_AS/Línea_base_de_Subsisistema/Líneas_base/Línea_base_de_Capital_Humano/Especificación_Componente.xls)

Anexo 3: Clases del Diseño

Estructura de las clases del diseño de los componentes: Trabajador, Puesto de Trabajo, Régimen de trabajo y descanso, Nómina, Submayores y Cierre. El documento de la Línea Base íntegro se encuentra en la siguiente dirección.

[http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente de la AS/Línea base de Subsisistema/Líneas base/Línea base de Capital Humano/Documento Línea Base de Subsisistema.doc](http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_sistema/Expediente_de_la_AS/Línea_base_de_Subsisistema/Líneas_base/Línea_base_de_Capital_Humano/Documento_Línea_Base_de_Subsisistema.doc)

Puesto de Trabajo

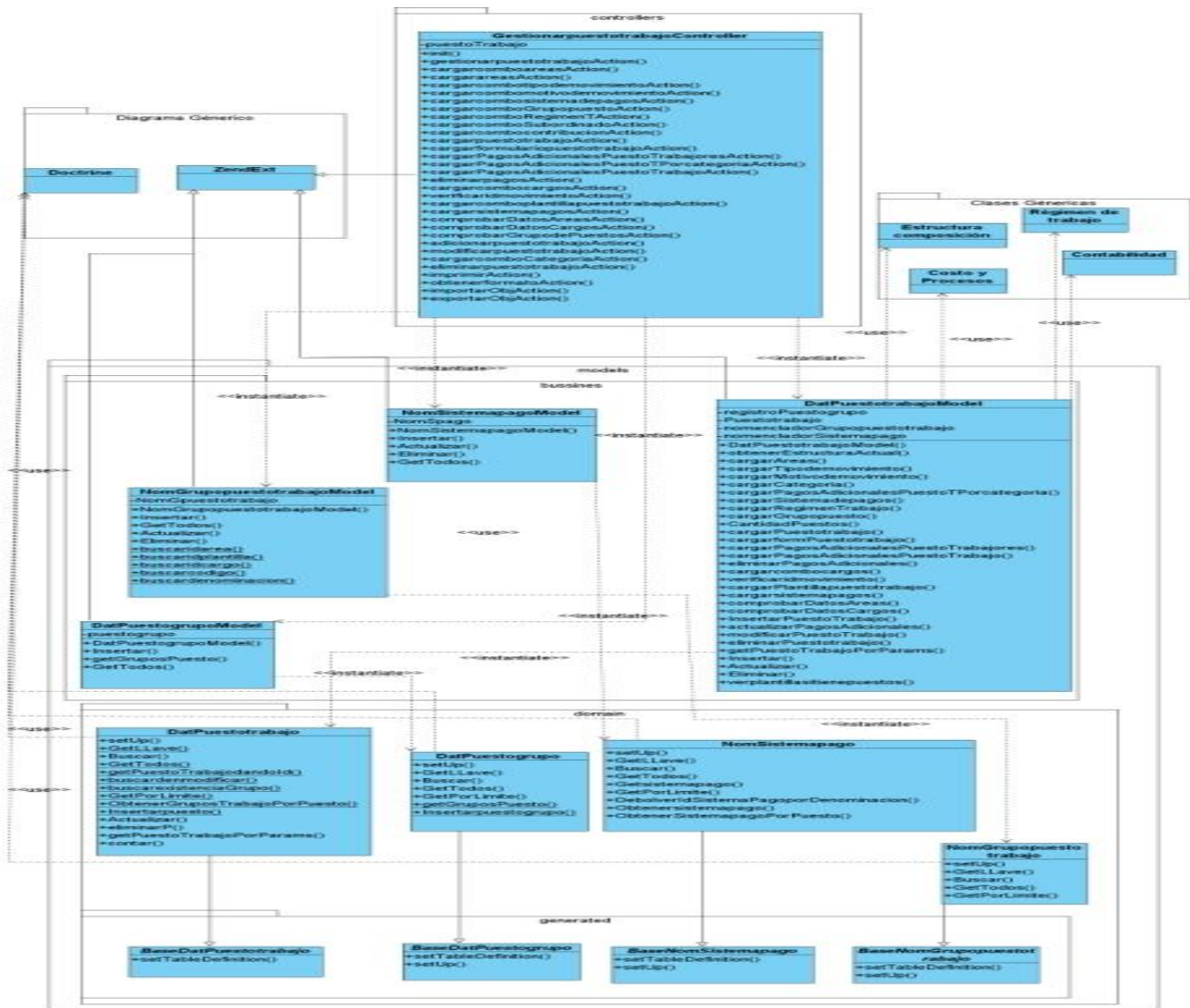


Figura 4.1 Diseño de clases del componente Puesto de Trabajo.

Trabajador

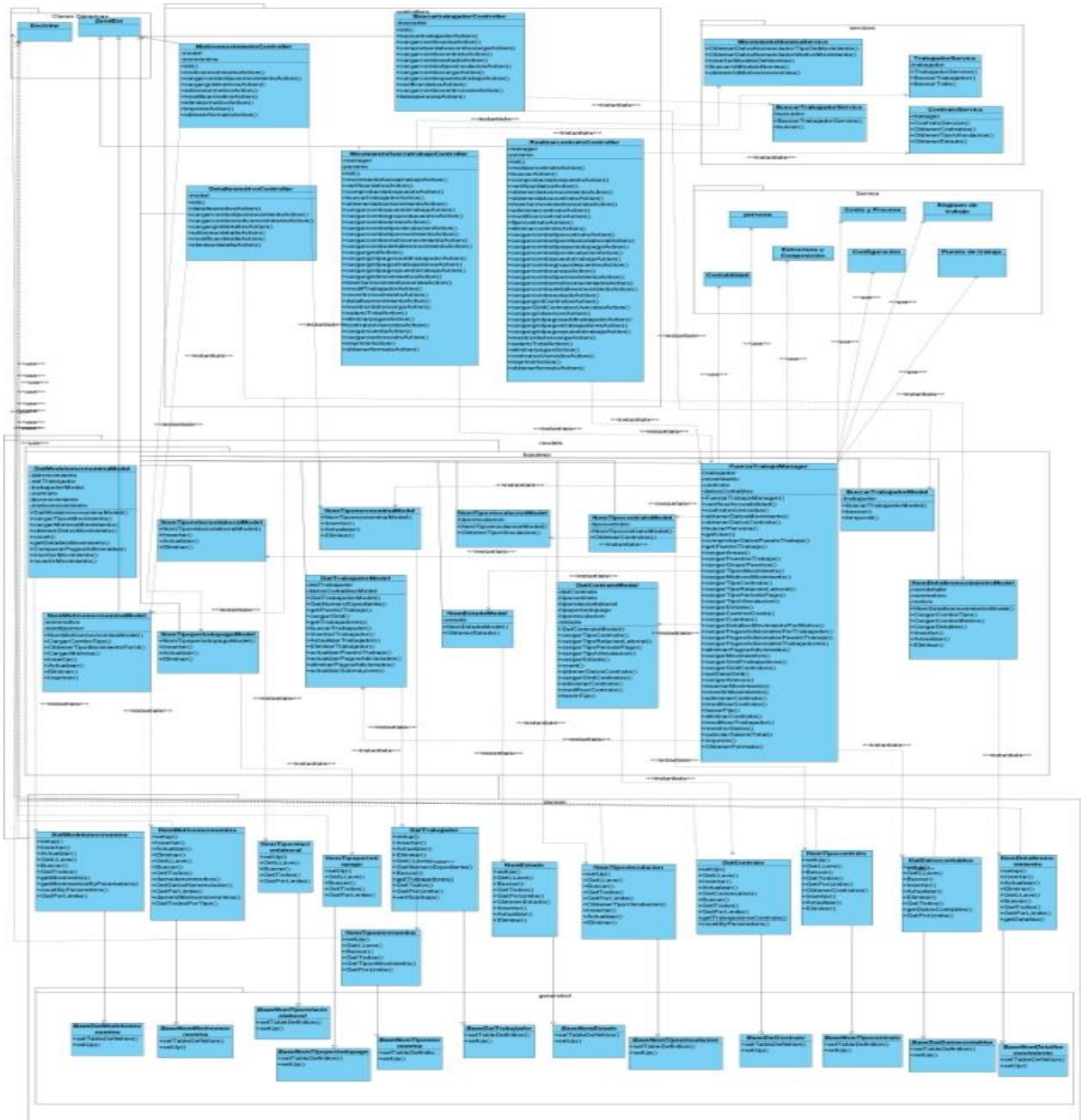


Figura 4.2 Diseño de clases del componente Trabajador.

Trabajador

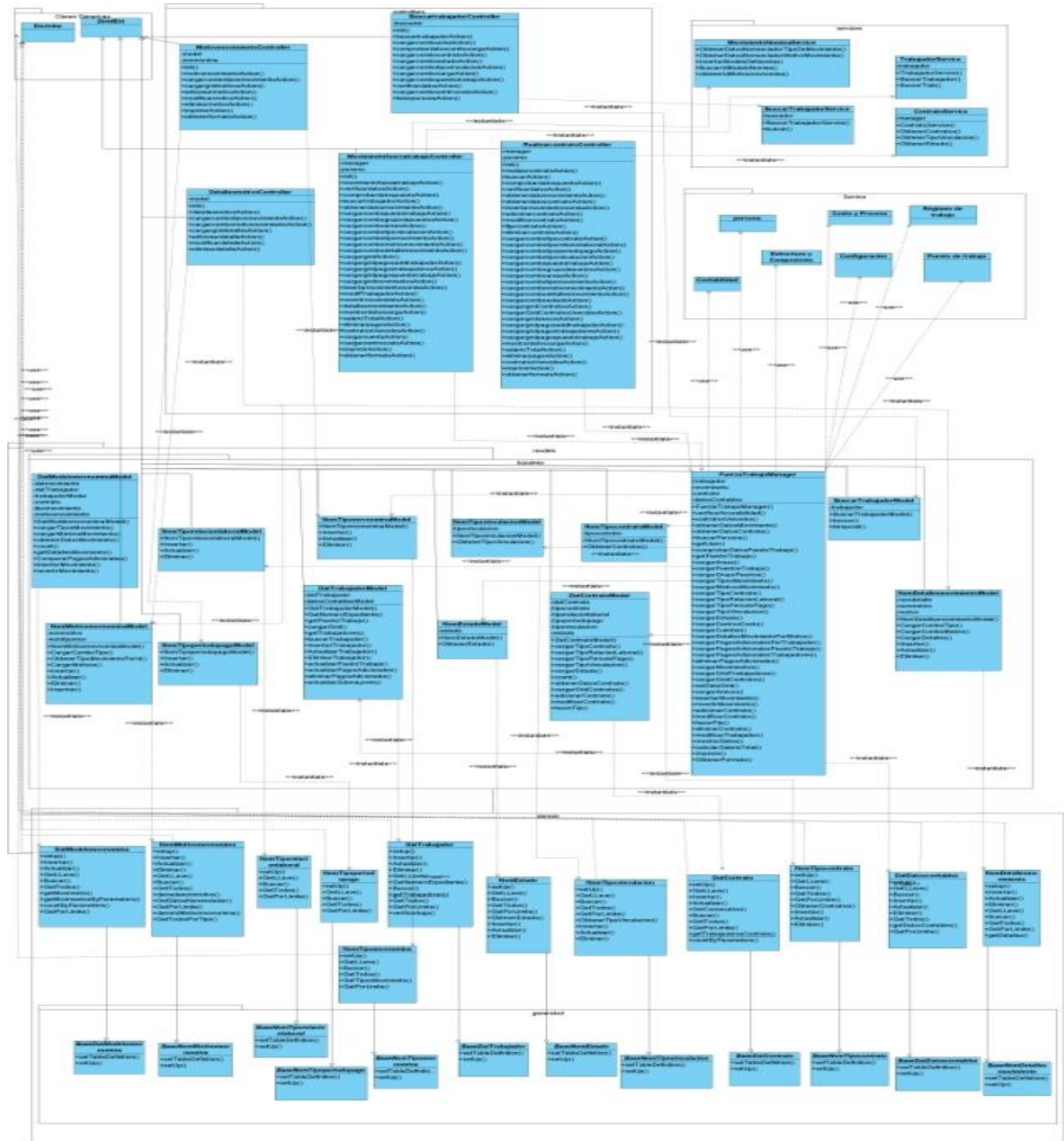


Figura 4.3 Diseño de clases del componente Trabajador.

Nómina

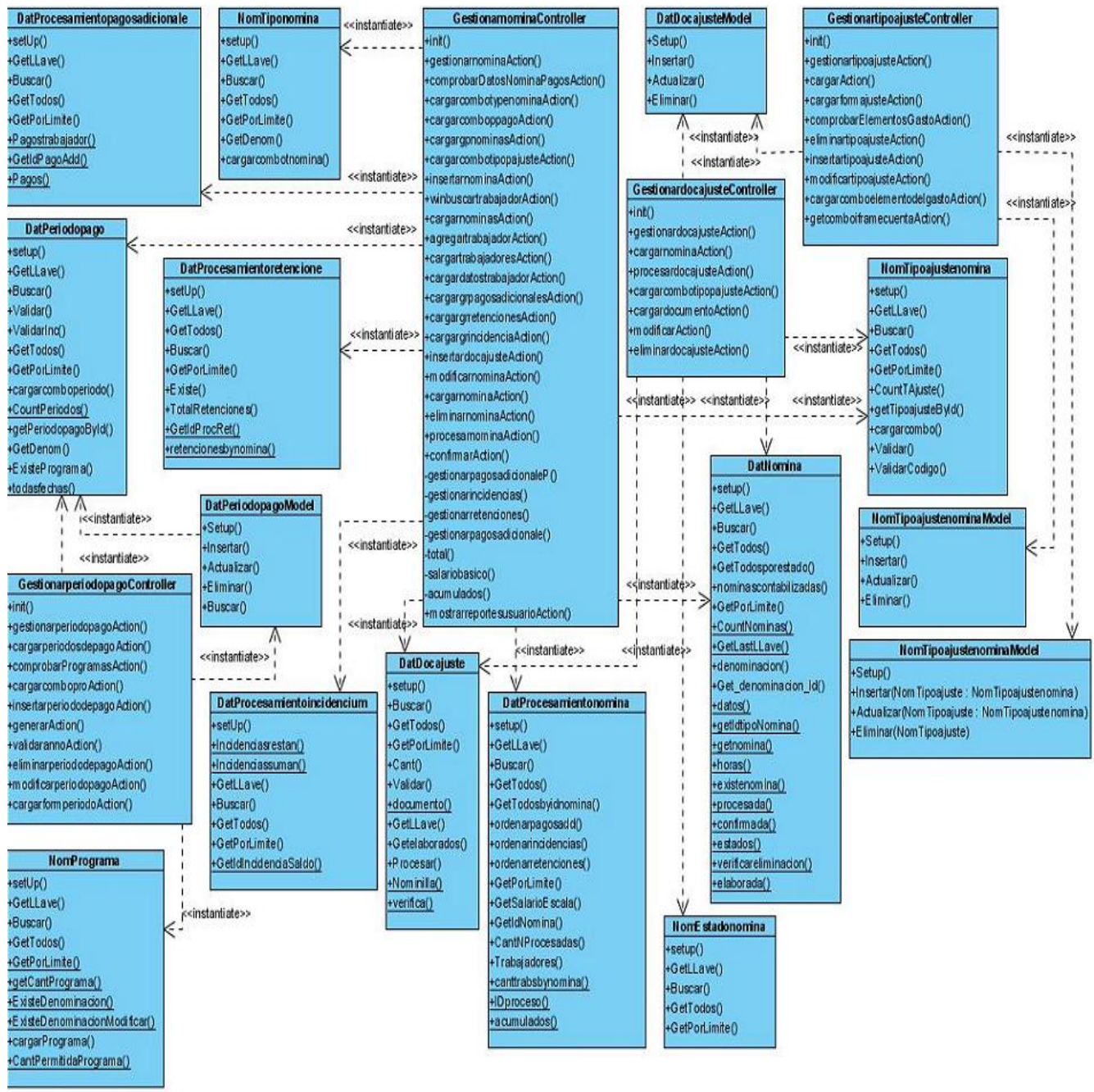


Figura 4.4 Diseño de clases del componente Nómina.

Submayores

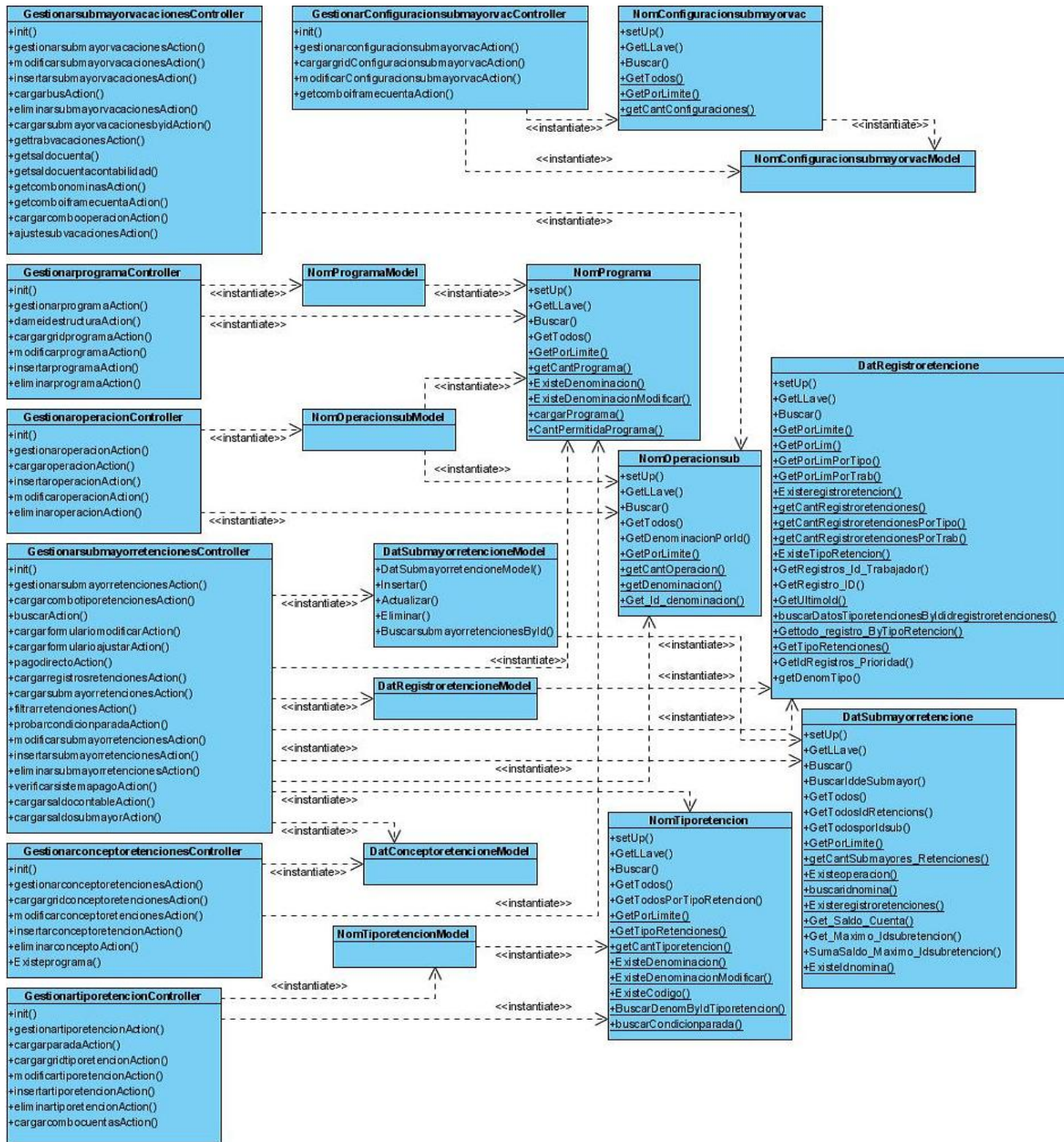


Figura 4.5 Diseño de clases del componente Submayores.

Cierre

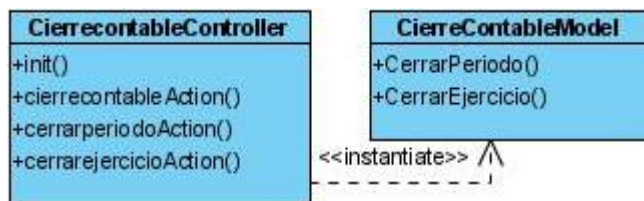


Figura 4.6 Diseño de clases del componente Cierre.

Incidencias

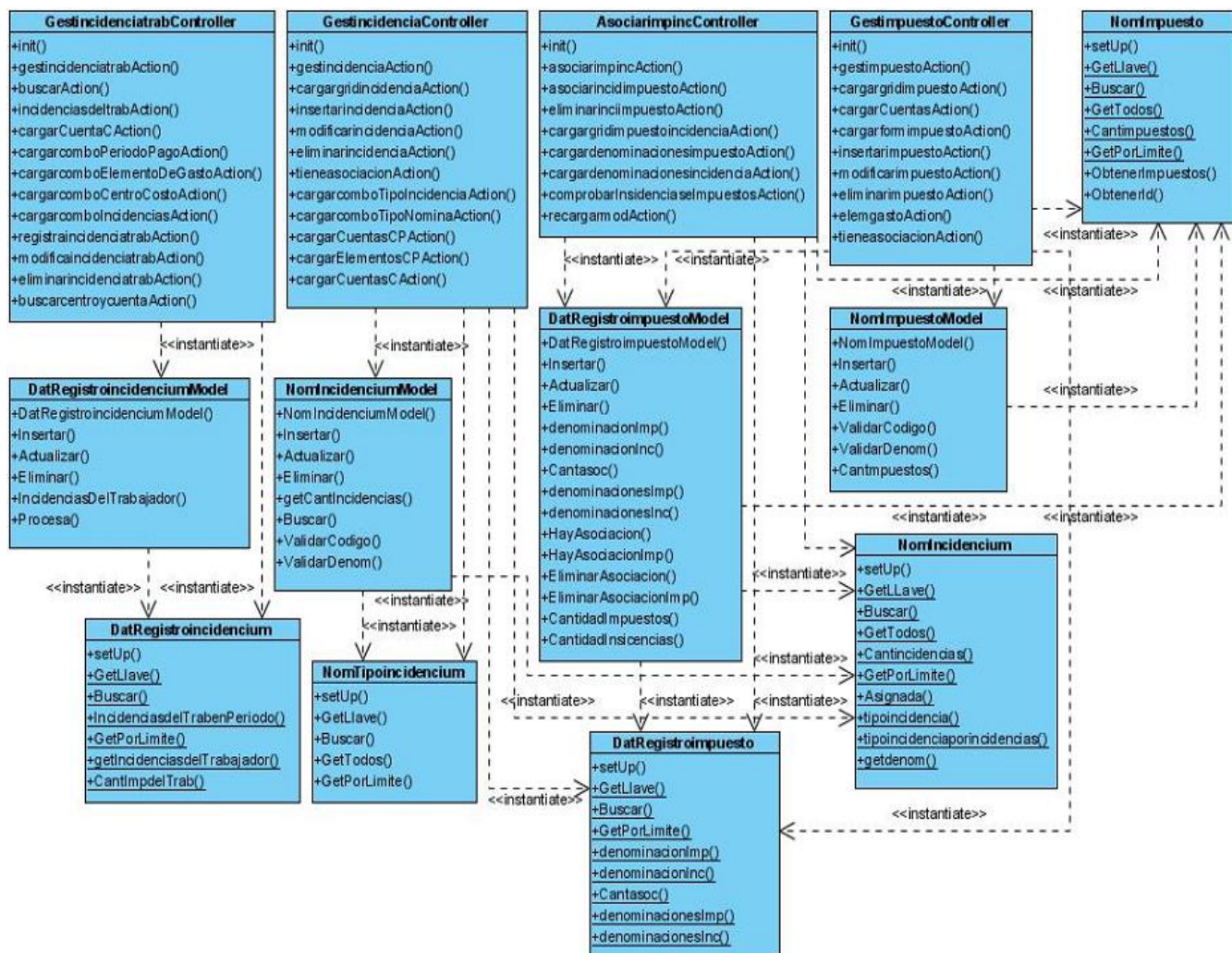


Figura 4.7 Diseño de clases del componente Incidencias.

Anexo 4: Diccionario de Datos

Descripción de algunas tablas del Diccionario de Datos. El documento en su conformación completa se encuentra en la siguiente dirección

[http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_Datos/Expediente-Datos/DAT-Expediente/DAT-Subsistemas Lineas/L_CapitalHumano/Mod_Capitalhumano/Diccionario_datos/ERP-ARQ_Diccionario_Datos- \(CH\) v1.0.1.doc](http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_Datos/Expediente-Datos/DAT-Expediente/DAT-Subsistemas Lineas/L_CapitalHumano/Mod_Capitalhumano/Diccionario_datos/ERP-ARQ_Diccionario_Datos- (CH) v1.0.1.doc)

Anexo 5: Matriz de Trazabilidad

El documento íntegro se encuentra en la siguiente dirección

http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_Datos/Expediente-Datos/DAT-Expediente/DAT-Subsistemas Lineas/L_CapitalHumano/Mod_Capitalhumano/Matriz_trazabilidad/ERP-ARQ_Matriz_Trazabilidad v1.0.0.xls

Anexo 6: Prueba de Concepto

Este documento se puede localizar en la siguiente dirección.

http://10.12.179.3:3389/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/Arquitectura_Datos/Expediente-Datos/DAT-Expediente/DAT-Subsistemas Lineas/L_CapitalHumano/Mod_Capitalhumano/Pruebas_Concepto_Diseño/ERP-ARQ_Pruebas_Concepto_Diseño v1.0.0.doc

Glosario de términos

Artefactos: Representa una parte de la información que es producida, modificada, o usada por un proceso, define un área de responsabilidad y está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos.

Capital Humano: Conjunto de conocimientos, experiencias, habilidades, sentimientos, actitudes, motivaciones, valores y capacidad para hacer, portados por los trabajadores para crear más riquezas con eficiencia.

Cardinalidad: Relación entre las tablas de una base de datos Ejemplo (cardinalidad de uno a muchos).

Disparadores: Los disparadores (triggers) son funciones definidas por el diseñador de la base de datos que se ejecutan cuando se producen ciertas operaciones en las bases de datos.

Esporádicos: Representa un aspecto que suceda con poca frecuencia, no es regular y ocurre aisladamente sin relación alguna con otros casos anteriores o posteriores.

Frameworks: es una estructura conceptual y tecnológica de soporte definida normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Hardware (soporte físico): Utilizada generalmente para describir los artefactos físicos de una tecnología. Es el conjunto de elementos físicos que componen una computadora Disco Duro, CD-ROM, entre otros.

IEEE (Institute of Electrical and Electronics Engineers): Asociación técnico-profesional mundial dedicada a la estandarización. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación e ingenieros en telecomunicación.

ISO: Organización Internacional de Normalización.

Modelo Relacional: El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

RUP (Racional Unified Process): El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo software iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

Servicios: Es un conjunto de actividades que buscan responder a una o más necesidades de un cliente.

Servidor: Constituye una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

SEI: Software Engennering Institute (Instituto Federal Estadounidense de Investigación y Desarrollo).

SQL: es un lenguaje formal declarativo, estandarizado ISO, para manipular información en una base de datos.

Software: Se conoce como software el equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de componentes lógicos necesarios para hacer posible la realización de tareas específicas, en contraposición a los componentes físicos del sistema llamados hardware.

Stakeholder: Parte interesada, es decir, cualquier persona o entidad que es afectada por las actividades de una organización; por ejemplo, los trabajadores de esa organización, sus accionistas, las asociaciones de vecinos, sindicatos, organizaciones civiles y gubernamentales.

Tablespace: Es la parte física de la base de datos utilizada para asignar almacenamiento

Tuplas: Secuencia ordenada de objetos, una lista con un número limitado de objetos.