

**CREACIÓN DE NUEVA NOVEDAD DEL SISTEMA DE
INFORMACIÓN DE RECURSOS HUMANOS (SIRH)
CORRESPONDIENTE A LA NIVELACIÓN SALARIAL DE
DOCENTES Y TRABAJADORES OFICIALES**

CRISTHIAN DAVID MARÍN ANGULO

200834882

cristhian.marin@correounivalle.com

Director

Ángel Gracia Baños, Ph.D

angel.garcia@correounivalle.edu.co

Codirector

Pablo Astroz Avellaneda

pablo.astroz@correounivalle.edu.co

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Programa Académico de Ingeniería de Sistemas

Santiago de Cali, Enero 5 de 2015

Índice

AGRADECIMIENTOS	5
RESUMEN	6
1. INTRODUCCIÓN	7
2. OBJETIVOS	8
2.1. Objetivo General	8
2.2. Objetivos Específicos	8
3. MARCO TEÓRICO	9
3.1. Bases de Datos[1]	9
3.1.1. Sistema Gestor de Bases de Datos (SGBD)	9
3.1.2. Modelo Entidad-Relación (E-R)	10
3.1.3. Modelo Relacional	10
3.2. Modelo-Vista-Controlador[2]	11
3.3. Metodología Ágil[2]	12
3.3.1. Programación Extrema (XP)	12
3.4. Programación Orientada a Objetos	13
3.5. PHP (Hypertext Preprocessor)[3]	14
3.6. SQL (Structured Query Language)[1]	14
3.7. PostgreSQL[4]	15
4. ESTADO DEL ARTE	17
5. TRABAJO REALIZADO	18
5.1. Análisis	18
5.2. Diseño e Implementación	19
5.2.1. Pruebas	22
5.2.2. Retroactividad Docentes 1279	23
5.2.3. Retroactividad Docentes 115	24
5.3. Producción	25

6. CONCLUSIONES	26
7. TRABAJOS FUTUROS	27
8. BIBLIOGRAFÍA	28
9. ANEXOS	29
9.1. Historias de Usuario	29
9.2. Modelo E-R	37
9.3. Diagramas de Navegación	38

Índice de figuras

1.	Niveles de abstracción de un SGBD.[1]	9
2.	Modelo E-R con dos entidades, cliente y cuenta, y una relación, impostor.[1]	10
3.	Funcionamiento del modelo-vista-controlador.[2]	12
4.	Ciclo de vida de la metodología XP.[2]	13
5.	Componentes de PostgreSQL.[4]	15
6.	Captura tomada de SIRH, novedades de vinculación laboral.	17
7.	Captura tomada de SIRH, formato de puntos de los docentes que se rigen bajo el Decreto 1279.	19
8.	Captura tomada de SIRH, formato de puntos de los docentes que se rigen bajo la Resolución 115.	20
9.	Captura tomada de SIRH, el usuario “crithian_marin” con perfil de administrador y rol de “Recursos Humanos”.	20
10.	Captura tomada de SIRH, escaneo de la base de datos y reconocimiento de las nuevas tablas agregadas.	21
11.	Captura tomada de SIRH, integración de la novedad a los elementos correspondientes del sistema (docentes 1279 y 115, novedades de vinculación laboral).	21
12.	Captura tomada de Libre Office Calc, tabla de pruebas para corroborar que los resultados del sistema sean los correctos.	22
13.	Modelo E-R de la base de datos.	37
14.	Diagrama de navegación novedades.	38
15.	Diagrama de navegación retroactividad.	39

Índice de cuadros

1.	Tabla cliente.[1]	10
2.	Tabla cuenta.[1]	11
3.	Tabla impositor.[1]	11
4.	Acceder a la novedad de nivelación de los docentes 1279 y 115.	29
5.	Crear y guardar novedades de nivelación de los docentes 1279.	29
6.	Crear y guardar novedades de nivelación de los docentes 115.	30
7.	Aprobar novedades de nivelación de los docentes 1279 y 115.	30
8.	Anular novedades de nivelación de los docentes 1279 y 115.	31
9.	Consultar novedades de nivelación de los docentes 1279 y 115.	31
10.	Realizar retroactividad individual o por grupo de los docentes 1279 y 115.	32
11.	Realizar retroactividades de novedades de nivelación que no han sido liquidadas.	32
12.	Calcular el sueldo para los docentes 1279.	33
13.	Calcular el sueldo para los docentes 115.	33
14.	Evaluar la retroactividad con los todos los criterios involucrados.	34
15.	Ignorar los valores retroactivos negativos en la sumatoria de retroactivos de cada periodo de liquidación.	34
16.	Identificar las novedades que estén involucradas en retroactividades liquidadas.	35
17.	Almacenar el retroactivo total de cada criterio de liquidación.	35
18.	Liquidar correctamente el retroactivo generado.	36
19.	Generar reportes del proceso retroactivo.	36

Agradecimientos

Quiero agradecer primeramente a mi papá Jairo Marín Cardona que fue el que me permitió llegar tan lejos y me dio gran parte de la educación en mi infancia, estuvo ahí para apoyarme en todo y a pesar de que muchas veces no correspondí como era debido, nunca perdió la fe en mí. Tengo la suerte de tener un padre que fue capaz de sacar su tiempo para dedicarme a enseñarme todo lo que el podía, gracias a él ya sabía leer antes de llegar al jardín, gracias a él aprendí matemáticas, gracias a él fue que tuve el hábito de leer libros aunque cuando pequeño yo los rayara (ya perdí ese hábito). Espero poder retornar todo lo que me ha dado y acompañarlo de igual manera que él lo ha hecho hasta ahora.

No por ser menos importante quiero agradecer a mi madre Leidy Johanna Angulo, que a pesar por eventos de la vida no puedo compartir mucho con ella un hogar junto con mi papá, pero que de igual manera, siendo como una amiga a la vez me mostró su apoyo incondicional y preguntando todos los días “¿cómo va con la tesis pelaito?” me dio ánimos para seguir trabajando duro y lograr las metas. Ella es una persona alegre que a pesar de las dificultades nunca se deja decaer y sigue luchando por su familia (creo que saque algo de ella, a pesar de que me quejo mucho). Gracias por estar a mi lado, yo se las dificultades que has pasado por la familia.

Siguiendo quiero agradecer a mis dos hermanos Hanner y Juliana, que cuando tuve dificultades me ayudaron y que quiero estemos los tres así de unidos como siempre, tratándonos con “cariño” (es maltrato pero con amor).

Quiero agradecer a Yerminson Gonzalez por la amistad que me brindó y que me apoyó económicamente para poder ir todos los días a la universidad (por políticas de la universidad y distracción mía no recibí pago los últimos tres meses de la pasantía) y poder continuar con el trabajo. Es un buen partero y alguien que respeta a sus compañeros. Después de que comience a trabajar tengo pagarle toda la deuda.

Un agradecimiento a todos los profesores que con sus enseñanzas me permitieron culminar satisfactoriamente mi carrera, en especial el profesor Ángel García que me dió las oportunidades y el apoyo para realizar este trabajo de grado.

Y por último, teniendo dudas del por qué, quiero agradecerles a Marcel Sandoval y Eduardo Pinto, son buenos parceros y pues la verdad sólo los estoy colocando porque ellos me colocaron a mí, pero no importa, los aprecio al igual que al resto de compañeros con los que compartí el transcurso de la carrera (Jorge, Paola, Maria Andrea, Felipe, Cristina, Vasquez, Julian, etc).

Resumen

En este documento, se dará una descripción del proceso realizado durante la pasantía, la cual era realizar varios módulos prototipo: el primero que permitiera crear, aprobar y anular novedades de nivelación para los docentes del Decreto 1279[5] y la Resolución 115[6], y el segundo que realizara el proceso de retroactividad a partir de las novedades anteriores. Estos dos módulos fueron implementados dentro del Sistema de Información de Recursos Humanos (SIRH).

En las siguientes secciones se mostrará una breve descripción del SIRH, los objetivos a cumplir y la base existente a partir de la cual se desarrollaron los módulos. Antes de describir como se realizó la implementación, se informa de varios conceptos y tecnologías que se usaron en el transcurso de la pasantía, la cual estuvo compuesta de varias etapas de las cuales hubo contratiempos en algunas de ellas, por supuesto, se explicará las causas de estos.

Al final del documento están los anexos con datos y diagramas que se utilizaron en el desarrollo del software.

Palabras claves: Decreto 1279, Resolución 115, Sistema de Información de Recursos Humanos

1. INTRODUCCIÓN

El Sistema de Información de Recursos Humanos (SIRH) es una estructura web que permite consultar y procesar información referente a hojas de vida de empleados y dependientes, la estructura de la planta de personal, liquidación de nóminas, pagos y descuentos de nómina, aportes a entidades externas, liquidación de préstamos sociales, Decreto 1279 y Resolución 115, interfaz presupuestal, interfaz contable, selección y contratación, cuotas partes, cesantías, seguridad social, vacaciones, evaluación de desempeño, concursos, embargos, jubilaciones, certificados y procesos de la Oficina de Recursos Humanos.

Este sistema fue desarrollado por la Oficina de Informática y Telecomunicaciones (OITEL) y actualmente la misma le da soporte.

Uno de los procesos más críticos es la nivelación de docentes y trabajadores oficiales, el cual se ha estado realizando de manera manual causando un gran gasto de recursos debido a los múltiples requisitos que se deben cumplir y a la vez de lo diversos que son dependientes de las facultades en que trabajan especialmente los docentes. Por lo tanto se buscó crear un módulo que permita realizar estas operaciones de manera automática y que tenga en cuenta todos los casos posibles que puedan ocurrir a la hora de hacer la nivelación.

2. OBJETIVOS

2.1. Objetivo General

Diseñar, desarrollar y documentar un nuevo aplicativo perteneciente a los módulos de novedades y liquidación del Sistema de Información de Recursos Humanos (SIRH) para automatizar la nivelación de docentes y trabajadores oficiales.

2.2. Objetivos Específicos

- Diseñar e implementar la novedad para el registro de nivelaciones de docentes del decreto 1279 y resolución 115.
- Diseñar e implementar la operación de liquidación de excedentes de nivelación para docentes del decreto 1279 y resolución 115.
- Ajuste y/o implementación de reportes relacionados.

3. MARCO TEÓRICO

3.1. Bases de Datos[1]

Es un conjunto de datos que componen un mismo contexto, los cuales estan almacenados de manera estructurada para un uso posterior. Uno de sus propósitos es mostrar una visión abstracta de los datos a los usuarios.

3.1.1. Sistema Gestor de Bases de Datos (SGBD)

Se refiere al software que permite acceder a los datos almacenados y administrarlos de forma práctica y eficiente. Dado que generalmente se usan para dar fiabilidad en la información en las organizaciones, éstas poseen conceptos y técnicas para recuperar la información almacenada a pesar de las caídas del sistema y accesos no autorizados. Los SGBD poseen varios sistemas de abstracción debido a que no todos los usuarios tienen conocimiento sobre computadores e informática:

- **Nivel físico:** Aquí se almacenan realmente los datos, donde se describen las estructuras complejas de bajo nivel.
- **Nivel lógico:** Se describe que datos se almacenan y que relaciones existen entre estos.
- **Nivel de vistas:** Usa estructuras más simples y solo muestra una parte de la base de datos completa para que la interacción sea mas fácil.

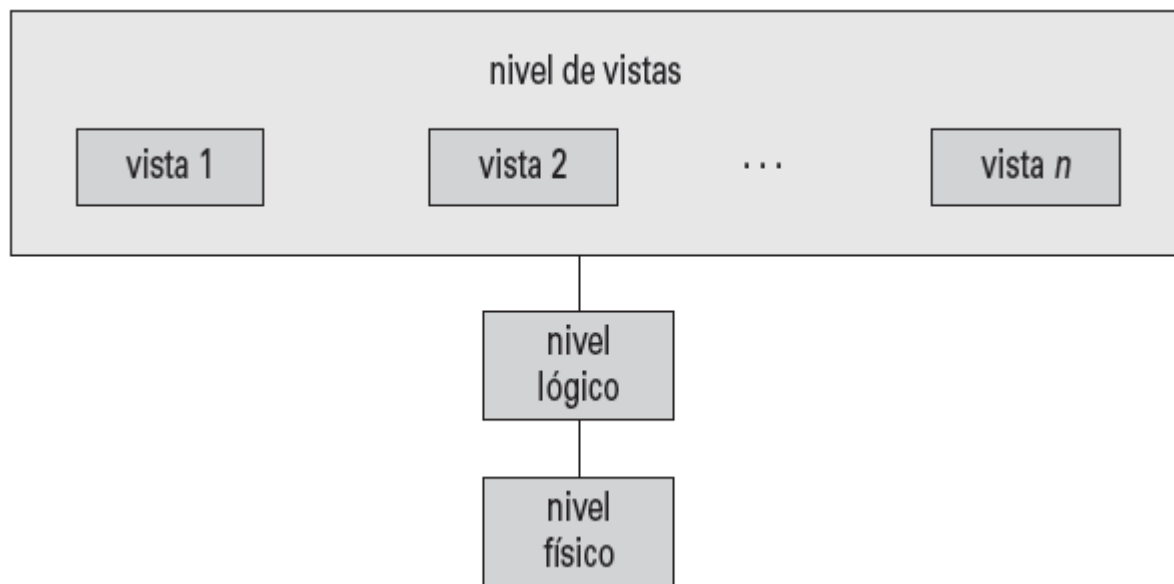


Figura 1: Niveles de abstracción de un SGBD.[1]

3.1.2. Modelo Entidad-Relación (E-R)

Se basa en como funciona la percepción en el mundo real a partir de unos conceptos básicos: entidades y relaciones. Las entidades se encuentran descritas por atributos, de los cuales existe uno adicional que identifica a dicha identidad. Mientras que una relación es la asociación de varias entidades. Estos conceptos se pueden representar de forma lógica mediante un diagrama E-R usando los siguientes componentes:

- Rectángulos para representar las entidades.
- Elipses para representar los atributos.
- Rombos que representan las relaciones entre las entidades.
- Líneas que unen los atributos con las entidades, y las entidades con las relaciones.

Por ejemplo a continuación se muestra dos entidades, con sus respectivos atributos y unidas bajo la relación impositor:

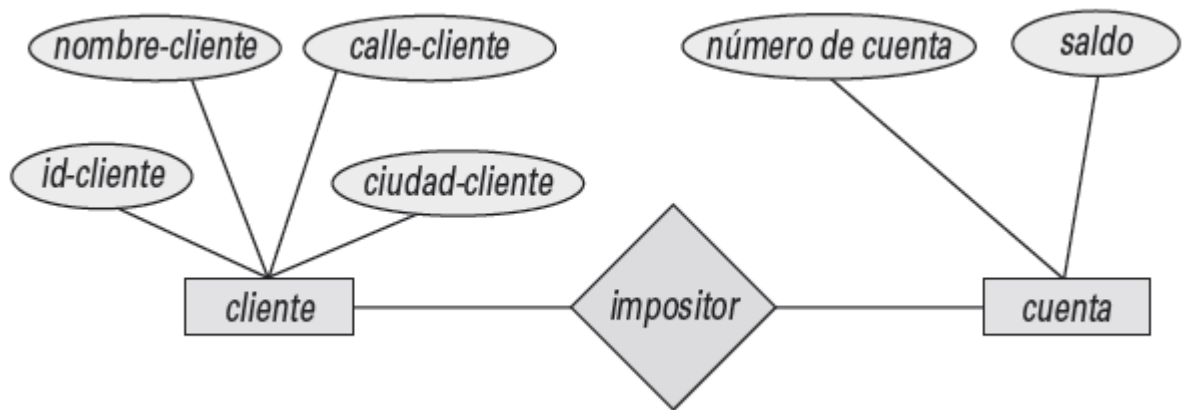


Figura 2: Modelo E-R con dos entidades, cliente y cuenta, y una relación, impositor.[1]

3.1.3. Modelo Relacional

A diferencia del modelo anterior, aquí se utilizan tablas para representar los datos y las relaciones entre ellos, además que esta en un nivel inferior de abstracción al modelo E-R. Los modelos de base de datos generalmente primero se diseñan en el diagrama E-R y luego se traducen al modelo relacional.

Ahora de acuerdo al ejemplo mostrado en el modelo E-R, las siguientes tablas muestran sus contrapartes en el relacional:

id-cliente	nombre-cliente	calle-cliente	ciudad-cliente
19.283.746	Camilo	Americas	Cali
01.928.374	Ana	Roosvelt	Medellín
67.789.901	Jorge	Vasquez Cobo	Bogotá
18.273.609	Claudia	Colombia	Santa Marta

Tabla 1: Tabla cliente.[1]

numero-cuenta	saldo
C-105	200
C-098	450
C-220	300
C-195	100

Tabla 2: Tabla cuenta.[1]

id-cliente	numero-cuenta
01.928.374	C-220
18.273.609	C-105
19.283.746	C-098
67.789.901	C-195

Tabla 3: Tabla impositor.[1]

3.2. Modelo-Vista-Controlador[2]

Este modelo describe una infraestructura de tres capas lo que permite separar la interfaz de navegación del comportamiento de la aplicación. El modelo contiene el contenido y la lógica de procesamiento de la aplicación, incluyendo el acceso a datos externos específicos. La vista contiene las funciones específicas de la interfaz para presentar el contenido y la lógica de procesamiento, así como los objetos de contenido. El controlador coordina el flujo de datos entre estos dos.

En resumen la vista es actualizada por el controlador con datos del modelo basados en las entradas del usuario. En la siguiente figura se puede observar como el controlador maneja las solicitudes del usuario seleccionando la vistas se aplique a dichas solicitudes, una vez determinada la solicitud se transmite al modelo un pedido de comportamiento, haciendo que este acceda a los datos almacenados. Una vez recuperados los datos, el controlador nuevamente escoge la vista apropiada para organizarlos y enviarlos al navegador del cliente.

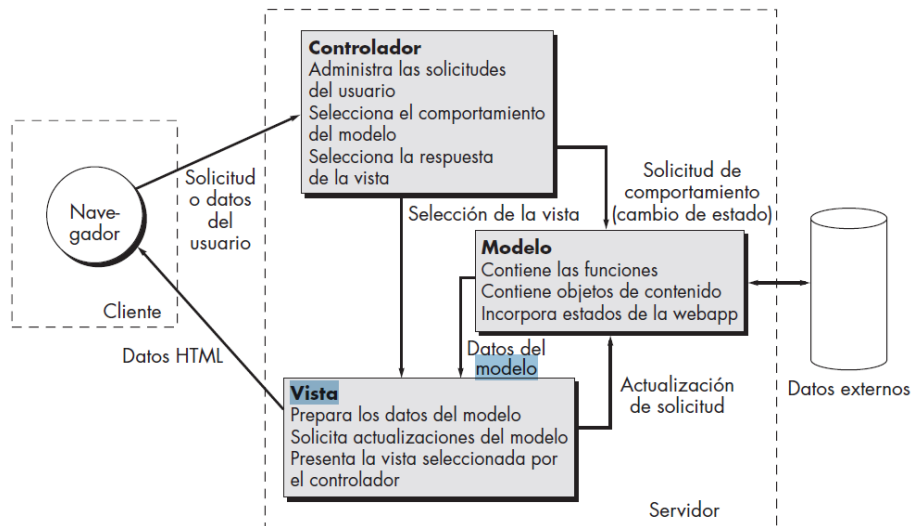


Figura 3: Funcionamiento del modelo-vista-controlador.[2]

3.3. Metodología Ágil[2]

Las metodologías ágiles son procesos que son adaptables a cambios futuros en los requerimientos de un software. Para esto debe existir una comunicación constante con el cliente para que exista una retroalimentación del progreso del sistema con la ayuda de prototipos funcionales.

La comunicación entre los integrantes del proyecto también es una base importante no solo para tener control en las diferentes etapas del software, sino que además facilita generar nuevas estrategias que permitan el avance eficiente del mismo. Por lo tanto, las personas que integran el proyecto deben estar motivadas y ser proactivas, ya que pueden ocurrir cambios a lo largo de todo el desarrollo.

3.3.1. Programación Extrema (XP)

Este proceso ágil está compuesto por cinco valores: comunicación, simplicidad, retroalimentación, valentía y respeto. Para lograr una comunicación eficaz entre los participantes, la colaboración es informal entre clientes y desarrolladores por medio de conceptos y así evitar la documentación voluminosa. La simplicidad restringe de que sólo se diseñen para resolver las necesidades inmediatas y no extender la implementación del código, ya si en un futuro ocurre un cambio se rediseña en dicho momento.

La retroalimentación se realiza por medio del software implementado, el cliente y los desarrolladores. Se hacen pruebas unitarias para dar una retroalimentación a los desarrolladores y se entregan incrementos al cliente para obtener las pruebas de aceptación a partir de las historias de usuario de las cuales se hablará mas adelante. Diseñar para hoy requiere valentía o disciplina debido a que se debe ser consciente de que es necesario suplir las necesidades inmediatas del cliente y que posiblemente los requerimientos futuros cambien mucho.

Y, finalmente se debe inculcar respeto entre los miembros del proyecto, tanto como para el cliente como para los desarrolladores, al igual que para el software en sí. A medida que se hacen entregas exitosas al proyecto esta confianza y respeto se va afianzando.

Las actividades que componen XP son:

- **Planeación:** Se escucha al cliente para entender el modelo de negocio, conocer las funcionalidades y características que se requieren. Esto conlleva a la creación de las historias de usuario que describen la salida, característica y funcionalidad a realizar, además se les asigna un valor para priorizar su desarrollo.
- **Diseño:** Guía la implementación de una historia de usuario mientras se escribe, siguiendo el principio de mantener sencilla la funcionalidad (nada más y nada menos, lo que se necesita)
- **Codificación:** La codificación se realiza para superar pruebas unitarias a partir de las historias de usuario, y de esta manera obtener una retroalimentación para luego si poder integrar el código.
- **Pruebas:** Las pruebas unitarias se deben hacer antes de codificar y es recomendable el uso de una estructura que automatice dichas pruebas. También estan las pruebas de aceptación que son por parte del cliente, quien aprueba las funcionalidades del software visibles para el.

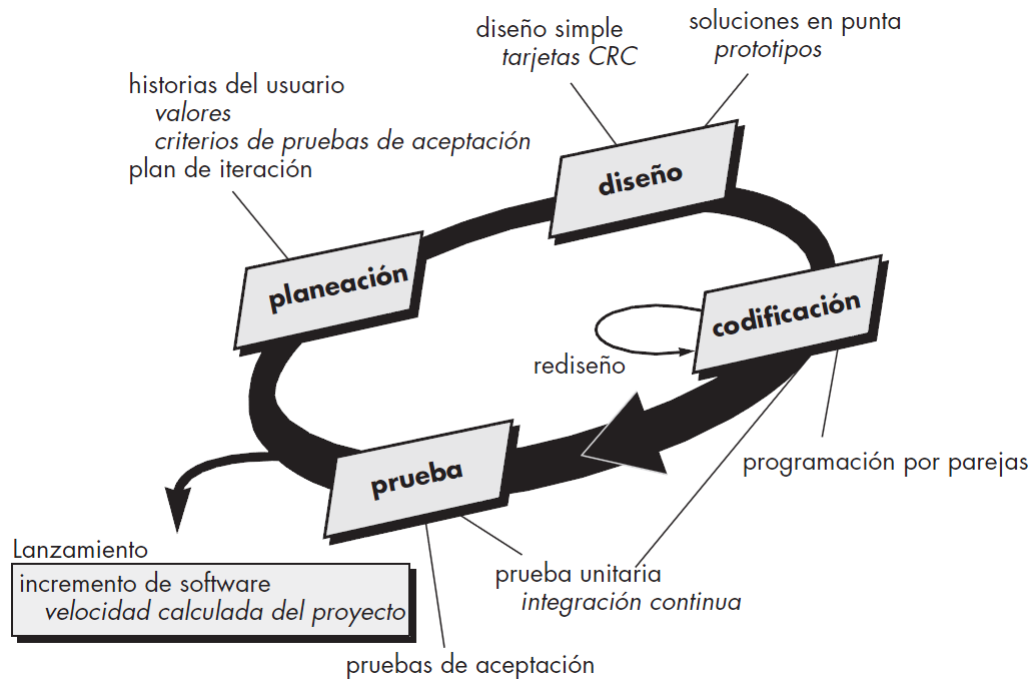


Figura 4: Ciclo de vida de la metodología XP.[2]

3.4. Programación Orientada a Objetos

La programación orientada a objetos es un paradigma que provee herramientas que permiten representar elementos en el espacio del problema como objetos, los cuales no estan restringidos a solo un tipo de problema y permite adaptar el software añadiendo mas objetos. Esto funciona como una analogía a los objetos en el mundo real[7].

Los objetos son entidades que poseen estado, comportamiento e identidad:

- **Estado:** Esta compuesto de datos o atributos que tienen asignados valores concretos.
- **Comportamiento:** Esta definido por los métodos a los cuales objeto sabe responder, es decir, las operaciones que puede realizar.
- **Identidad:** Son las propiedades que diferencian al objeto del resto.

La POO se basa en varios conceptos, entre ellos están[8]:

- **Herencia:** Es uno de los mecanismos de los lenguajes de programación basados en clases, en donde una clase deriva de otra extendiendo su funcionalidad.
- **Cohesión:** Es el grado de dependencia que existe entre dos o más objetos o unidades de software.
- **Abstracción:** Expresa las características esenciales de un objeto que lo distingue de los demás (atributos y métodos).
- **Polimorfismo:** Es la capacidad de enviar mensajes sintácticamente iguales a objetos de tipos distintos, y que a su vez estos sepan responder a dichos mensajes.
- **Acoplamiento:** Es la forma de agrupar dos o varios módulos de software.
- **Encapsulamiento:** Los objetos estan aislados del exterior como módulos para así ocultar su estado, de tal manera que solo es posible cambiarlo por medio de los métodos definidos por dicho objeto.

3.5. PHP (Hypertext Preprocessor)[3]

Es un lenguaje de programación de código abierto que se ejecuta del lado del servidor, adecuado para el desarrollo web de contenido dinámico y que puede ser incrustado en HTML. Al ejecutar el código en el lado del servidor se genera el contenido dinámico para enviar al cliente, este recibe el resultado de la ejecución sin saber el código que lo creó.

PHP es multiplataforma y puede emplearse en todos los sistemas operativos más conocidos, Linux, Windows, Mac OS, entre otros. Además permite la programación por procedimientos, orientada a objetos o la mezcla de ambas. Una de las características potentes de este lenguaje es el soporte a las distintas bases de datos tales como PostgreSQL, MySQL, Oracle, Microsoft SQL Server, etc. También existen extensiones que permite el acceso a documentos XML y expresiones regulares compatibles con Perl.

3.6. SQL (Structured Query Language)[1]

Es un lenguaje declarativo de acceso a base de datos relacionales el cual permite realizar varias operaciones en ellas. Su característica principal es el uso de algebra y cálculo relacional para realizar consultas para recuperar de forma sencilla información o modificarla, debido a que se centra al manejo de conjuntos de registros permitiendo una alta productividad y la posibilidad de poder realizar una orientación a objetos. Además de las características anteriores también posee:

- **Lenguaje de definición de datos:** Proporciona comandos para la definición y modificación de esquemas de relación y borrado de relaciones.

- **Lenguaje interactivo de manipulación de datos:** Incluye lenguajes de consultas que se basan en álgebra relacional y cálculo relacional de tuplas.
- **Integridad:** Incluye comandos para especificar restricciones que deben cumplir los datos almacenados.
- **Definición de vistas:** Incluye comandos para definir vistas.
- **Control de transacciones:** Incluye comandos para indicar el inicio y fin de una transacción.
- **SQL incorporado y dinámico:** Es posible incorporar instrucciones de SQL en otros lenguajes de programación como C, C++, Java, PHP, Pascal, etc.
- **Autorización:** Incluye comandos para especificar los derechos de acceso a las relaciones y las vistas.

La eficiencia al ejecutar una consulta depende mucho del motor de base de datos y otros factores, por ejemplo, el uso de índices acelera la una instrucción de consulta, pero realentiza la actualización de los datos, por lo que algunas veces dependiendo de la necesidad se prioriza el acceso indexado o la rápida actualización de la información.

3.7. PostgreSQL[4]

Es un sistema de gestión de bases de datos objeto-relacional distribuido bajo licencia BSD y código fuente libre. PostgreSQL usa el modelo cliente-servidor y utiliza multiprocesos para garantizar la estabilidad del sistema. Mientras que un proceso está escribiendo sobre una tabla, otros pueden acceder a la misma tabla sin necesidad de bloqueos, lo que le permite a cada usuario tener una visión consistente de lo que está haciendo ya que un error en uno de los procesos no afecta el resto y el sistema puede continuar funcionando normalmente a diferencia de otros SGBD.

A continuación se muestra los componentes más importantes de PostgreSQL:

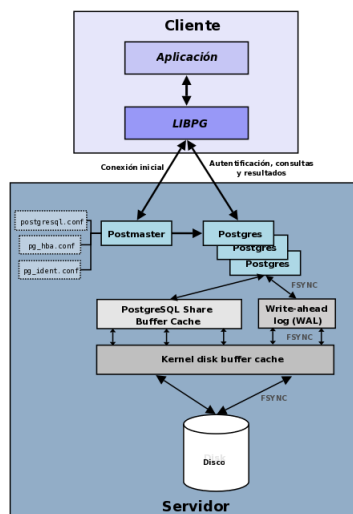


Figura 5: Componentes de PostgreSQL.[4]

- **Aplicación cliente:** Es el cliente que usa PostgreSQL como administrador de base de datos con conexión TCP/IP ó puertos locales.
- **Demonio postmaster:** Es el encargado de escuchar por un puerto las conexiones entrantes de los clientes y crear los procesos que se encargaran de autentificar estas peticiones, gestionar consultas y mandar resultados a las aplicaciones cliente.
- **Ficheros de configuración:** Los tres ficheros principales de configuración utilizados: postgresql.conf, pg_hba.conf y pg_ident.conf.
- **Procesos hijos postgres:** Procesos que se encargan de autentificar a los clientes, gestionar las consultas y mandar los resultados a las aplicaciones cliente.
- **PostgreSQL share buffer cache:** Memoria compartida para almacenar datos en caché.
- **Write-ahead log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- **Kernel disk buffer cache:** Caché de disco del sistema operativo.
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

4. ESTADO DEL ARTE

En la Universidad del Valle, la OITEL ha desarrollado varias novedades las cuales son eventos que se registran de los empleados tales como cambio de salario, cargo, actualización de credenciales, nombramientos administrativos, entre otras. Gracias a esto la universidad puede llevar un registro de los movimientos de vinculación, salariales, laborales e informativos de los empleados.

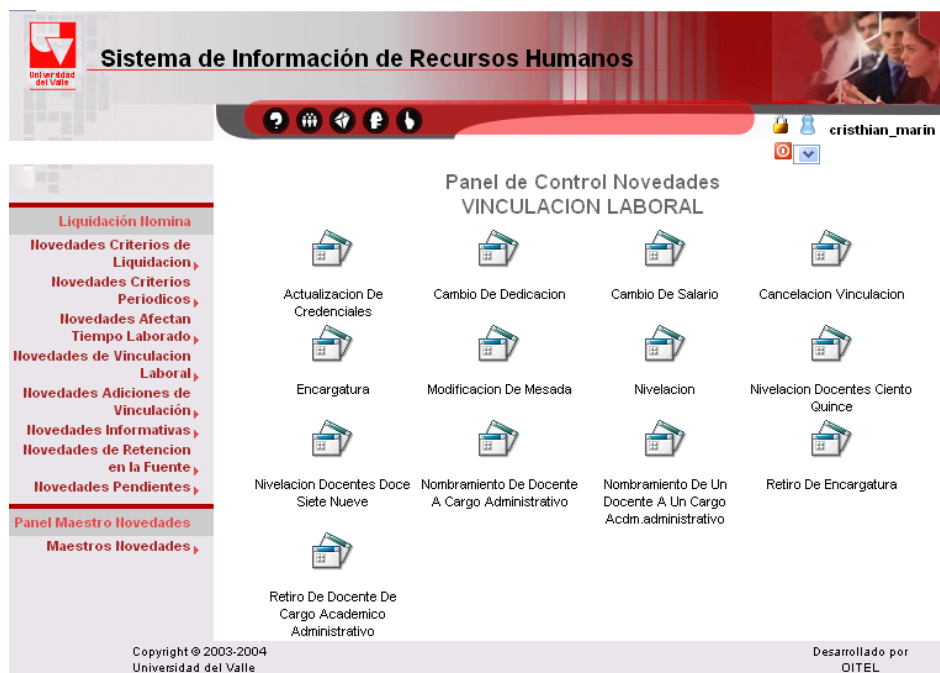


Figura 6: Captura tomada de SIRH, novedades de vinculación laboral.

Gracias a ello es posible realizar operaciones del día a día administrativo como lo es la liquidación de los sueldos, reportes administrativos, retroactivos, cambios de cargo o nivelaciones, entre muchos otros. Hasta el momento el sistema de SIRH cuenta con más de 90 novedades que se funcionan para los distintos tipos de empleados que laboran en la universidad: empleados públicos, empleados públicos no docentes, docentes e incluso monitores.

En el caso específico de la retroactividad, ya se contaba con este proceso para los empleados públicos no docentes, los cuales son empleados que además de suplir necesidades laborales o administrativas en la universidad pueden eventualmente realizan tareas de docencia pero sin ser profesores nombrados. El sueldo de los empleados públicos no docentes se maneja por medio de la novedad de cambio de salario y la categoría en la que se encuentre, por lo que la retroactividad solo funciona para una novedad por periodo de liquidación y calcula las diferencias por cada criterio del sueldo nuevo generado por dicha novedad y el sueldo liquidado.

El proceso de retroactividad permite ser realizado para un individuo en específico o el grupo total de empleados públicos no docentes teniendo en cuenta las novedades que aun no han sido liquidadas. Además al final se genera reportes con información del total retroactivo por cada criterio de liquidación evaluado. Estos resultados son almacenados en la base de datos y puede repetirse el proceso mientras que las novedades implicadas no hayan sido liquidadas como retroactivos.

5. TRABAJO REALIZADO

La OITEL trabaja con tres ambientes: local, desarrollo, pruebas y producción. De esta manera se queman varias etapas en el software y evitar al máximo los errores en el último ambiente el cual utiliza la comunidad universitaria.

El trabajo realizado consiste en cuatro etapas: análisis, diseño, implementación y producción.

5.1. Análisis

En esta etapa se habla con el cliente, en este caso la sección de Recursos Humanos de la Universidad del Valle, para conocer el modelo de negocio y la problemática a solucionar.

Lo primero que hay que entender es cómo se manejan los salarios de los profesores en la universidad, el cual a diferencia del resto de empleados públicos que trabajan en ella no es indicado por un sueldo como tal, sino que es asignado por medio de puntos y que luego se traducen en la remuneración de los docentes. En la Universidad del Valle existen dos tipos de docentes que se rigen bajo dos normas distintas, los que están manejados por el Decreto 1279 y los que acobijan bajo la Resolución 115; cada normativa indica cómo son asignados los puntos a los docentes y en qué cantidad de acuerdo a los estudios que hayan realizado, cargos administrativos, libros y artículos publicados, gestión docente, entre otros.

En la universidad este proceso inicia cuando los profesores envían los datos (con pruebas de acuerdo a los reglamentos nombrados anteriormente) actualizados con los cambios en su curriculum, que estén incluidos dentro de la normativa por la cual están regidos, al Consejo de Facultad. Cada facultad o instituto posee su propio Consejo de Facultad, el cual estudia los casos y las pruebas cada cierto tiempo, y así asignar los puntos correspondientes. Por ejemplo en el caso de los docentes que están bajo la Resolución 115 por cada año de postgrado en maestría o especialización se le adicionan 5 puntos hasta un límite de 20.

Después se envían la tabla de puntos de forma discriminada al CIARP (Comité Interno de Asignación y Reconocimiento de Puntaje), en donde se reúnen los representantes de cada facultad o instituto, el vicerrector académico, el vicerrector de investigaciones, un decano designado, el representante profesoral y la secretaria del mismo CIARP, para volver a estudiar los casos y hacer efectivo el aumento o disminución del puntaje. Este nuevo puntaje se ingresa al sistema, el cual tiene almacenado el valor del punto por cada tipo de docente (este valor puede cambiar cada cierto tiempo) y el cual se traduce en el nuevo salario que se le debe pagar.

El problema radica, en que este proceso que lleva hasta el cambio de puntos puede tomarse varios meses (o periodos de liquidación) y que el docente no pueda ver reflejado en su salario los cambios. Por lo tanto, es necesario un proceso de retroactividad en el cual se le debe liquidar a la persona desde que se aprobó la novedad de asignación de puntos hasta cuando fue efectivo el nuevo salario. Este proceso incluye los mismos factores a tener en cuenta al hacer una liquidación (vacaciones, gastos de representación, primas y bonificaciones).

Al momento de iniciar la pasantía solo existía en el sistema la retroactividad para empleados públicos no docentes, pero en el caso de los docentes 1279 y 115, este proceso se hacía de forma manual por parte de Recursos Humanos lo que conllevaba a costos de sobretiempo (muchos de los empleados llevan el trabajo a sus casas ya que no logran completar todas las tareas que deben entregar en un tiempo límite) o a errores humanos con mayor facilidad.

Además, existen varias diferencias entre el proceso que se lleva a cabo a los empleados públicos no docentes y los de nuestra problemática. El primero es que a diferencia de los públicos no docentes que la liquidación se maneja por salario y poseen la novedad de cambio de salario, los docentes 1279 y los 115 el salario es asignado por los puntos que luego se traduce en dinero, lo que lleva a otro problema, el cual es que tampoco existe una novedad para adicionar puntos (los puntos totales se asignan sin guardar registro de cuantos puntos de más fueron los que se adicionaron). La segunda diferencia es que las novedades de cambio de salario solo pueden ocurrir una vez por periodo de liquidación, mientras que las novedades de adición de puntos pueden ocurrir múltiples veces dentro de un mismo periodo de liquidación.

A continuación combinaremos el diseño y la implementación dentro de una misma sección, pues hubo bastantes iteraciones entre estas dos etapas (retroalimentación).

5.2. Diseño e Implementación

Lo primero que hay tener en cuenta de que se estaba trabajando sobre un sistema que se ha estado desarrollando por mas de diez años, y las nuevas funcionalidades no deben afectar de manera negativa lo que ya funciona. Teniendo en cuenta esto, fue aconsejable manejar la retroactividad de una forma similar a como funciona con los empleados públicos no docentes, al igual que las novedades. Tanto los formularios, como las tablas en la base de datos se deben basar en el formato de puntos para cada tipo de docente, los cuales se muestran a continuación:

VII. CATEGORIA Y PUNTOS DOCENTE

1. CATEGORIA DOCENTE

Categoría del docente: PROFESOR ASOCIADO
Ver Historial

2. PUNTOS POR CATEGORIA DEL PUNTO

Ver Historial

PUNTOS DE ACUERDO AL DECRETO 1279		
Criterios	Porcentaje	Cantidad Puntos
TITULOS OBTENIDOS	1	318
CATEGORÍA	1	74
EXPERIENCIA CALIFICADA	1	32.18
PRODUCCIÓN ACADÉMICA	1	147.5
DIRECCIÓN ACADÉMICA	1	0
ASIGNACIÓN ADICIONAL	1	1.76
DOCENCIA Y EXTENSION DESTACADA	1	18
DIFERENCIA PUNTOS	1	0
Total:		591.44

Figura 7: Captura tomada de SIRH, formato de puntos de los docentes que se rigen bajo el Decreto 1279.

VII. CATEGORIA Y PUNTOS DOCENTE

1. CATEGORIA DOCENTE

Categoría del docente: PROFESOR ASOCIADO
Ver Historial

2. PUNTOS POR CATEGORIA DEL PUNTO

Ver Historial

PUNTOS DE ACUERDO A LA RESOLUCIÓN 115		
Criterios	Porcentaje	Cantidad Puntos
ESTUDIOS TITULOS OBTENIDOS	1.5	45.9
LABOR DOCENTE	1	191.35
EXPERIENCIA PROFESIONAL	2	66.6
EXPERIENCIA ACADÉMICA	2	59.3
PRODUCCIÓN INTELECTUAL	4	21.9
DISTINCIÓN ACADÉMICA	4	0
IDIOMAS	1.5	0
Total:		385.05

Figura 8: Captura tomada de SIRH, formato de puntos de los docentes que se rigen bajo la Resolución 115.

Debido a que se estaba trabajando sobre un sistema ya hecho, era necesario entender cómo funciona de una manera general y estudiar un poco más a fondo las funcionalidades que se iban a afectar.

Los usuarios del sistema de SIRH, poseen varios tipos de roles y cada uno tiene ciertos permisos para acceder a varias funcionalidades. Tanto los roles como las funcionalidades son asignadas por el perfil administrador.

Sistema de Información de Recursos Humanos

Administración del Sistema

Panel de Control

Flujo De Aprobación

Panel de Control

Manejo de los Flujos de Aprobacion

CONSULTAR USUARIO DE APROBACIÓN

Login del Usuario cristhian_marin

Sub Rol del Usuario RECURSOS_HUMANOS

Elementos de la Estructura Asociados

Elemento Estructura: RECTORIA

Seleccionar para Eliminar: ☐

Adicionar Elemento
Eliminar Elemento Seleccionado

Actualizar
Restaurar
Borrar
Consultar
Nuevo

Copyright © 2003-2004
Universidad del Valle

Desarrollado por
OITEL

Figura 9: Captura tomada de SIRH, el usuario “cristhian_marin” con perfil de administrador y rol de “Recursos Humanos”.

Al indagar más sobre cómo crear una nueva novedad en el sistema, nos damos cuenta que el proceso para integrar la tabla de la base de datos, el formulario y el archivo de lógica de la novedad, se hace dentro de la misma interfaz de SIRH.

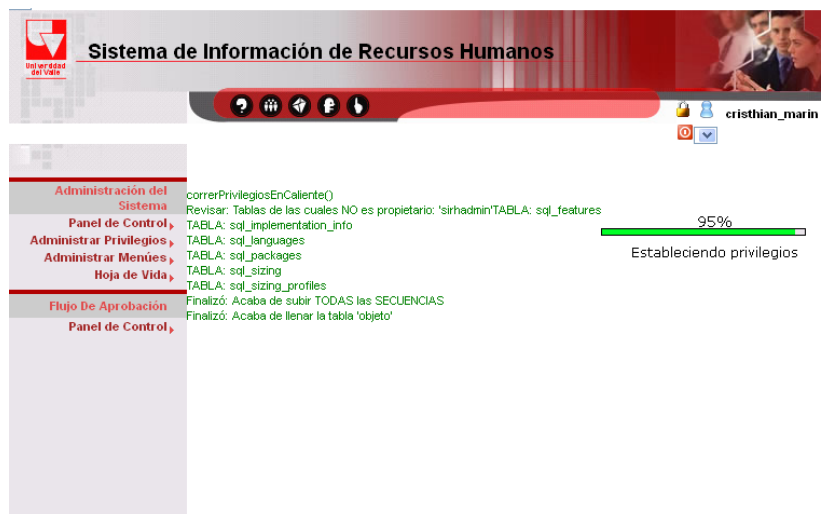


Figura 10: Captura tomada de SIRH, escaneo de la base de datos y reconocimiento de las nuevas tablas agregadas.

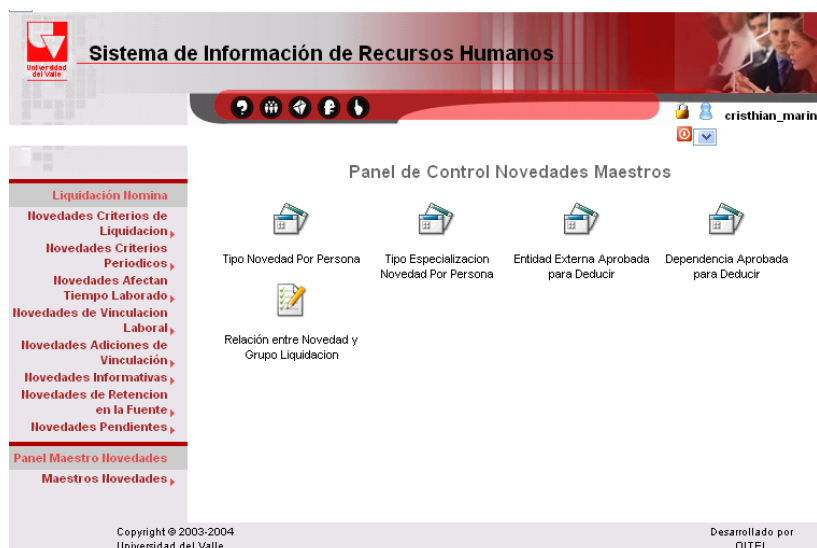


Figura 11: Captura tomada de SIRH, integración de la novedad a los elementos correspondientes del sistema (docentes 1279 y 115, novedades de vinculación laboral).

Como se dijo anteriormente los atributos de las tablas de cada novedad se basan en los formatos de puntos por cada tipo de docente, sin embargo se puede observar que aunque pertenezcan a un grupo llamado “Novedades” y posean muchas características en común (fecha, código, persona, etc.) fue necesario realizar una herencia de una tabla común llamada “novedadporpersona” ya que los formatos

de puntos son distintos. No solo eso, el resto de novedades ya realizadas siguen este mismo modelo y manejan formatos muy diferentes (por ejemplo, el manejo de los cargos de los empleados).

Los formularios se basan en otro ya realizado en cuanto a diseño. De igual manera el archivo que contiene la lógica, el cual gracias a que la interfaz de SIRH permite realizar parte del proceso como el manejo del flujo que debe seguir la novedad (guardar novedad, aprobar novedad y anular novedad, más información en la sección 9) solo es necesario codificar las consultas en la base de datos y las restricciones correspondientes al caso.

Esto fué relativamente fácil, pero la retroactividad a pesar de basarse en la retroactividad de los empleados públicos no docentes fue bastante complicada al punto de retrasar el progreso de la pasantía. La codificación que se realizó para los empleados públicos es de hace 10 años y además de que usa la clase “liquidación”, la cual no se podía modificar por riesgo a dañar el sistema. Al igual que con las novedades se debió conocer cómo estaba funcionando la retroactividad para los públicos no docentes.

El código de retroactividad es bastante complejo debido a que posee muchos flujos porque tienen que cumplir la normativa de la universidad. La otra razón que complicó el entendimiento de la clase de retroactividad, fué el pensamiento de querer entender todo lo que comprendía la clase en si y no lo que era realmente necesario para cumplir con los objetivos de la pasantía.

5.2.1. Pruebas

Antes de mencionar sobre la implementación de la retroactividad, es importante hablar sobre las pruebas, ya que gracias a estas en cierta forma ayudó a desarrollar el algoritmo final. Durante la codificación se hicieron pruebas causa-efecto, a medida que se iban programando caminos en los métodos que se creaban o se modificaban. Las pruebas unitarias proceden de las historias de usuario (ver en la sección 9), como por ejemplo el resultado de la retroactividad el cual se comprobaba con una tabla en Libre Office Calc:

A	B	C	D	E	F	G	H	I	J
Valor del punto			10745						
Porcentaje de gastos de representación			0.16						
Porcentaje de bonificación de servicios prestados			0.35						
Porcentaje de prima antigüedad			0.7						
SUELDO BASICO - 73									
Periodos de liquidación	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	
Liquidado	5187060	5371855	5371855	5817773	5817773	5817773	1357480	4460292	
Puntos nuevos	0	0	0	0	0	0	20	5	
Días laborados	30	30	30	30	30	30	7	23	
Días restantes	30	30	30	30	30	30	6	23	
Puntos totales	541.44	541.44	541.44	541.44	541.44	541.44	561.44	566.44	
Nueva liquidación	5817772.8	5817772.8	5817772.8	5817772.8	5817772.8	5817772.8	1407623.65	4666238.31	
Retroactivo previo	152500	0	0	0	0	0	0	0	
Retroactivo	478213	445918	445918	0	0	0	42980	205946	248927
									TOTAL
GASTOS REPRESENTACIÓN - 12									
Periodos de liquidación	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	
Liquidado	829930	859497	859497	217197	0	0	0	0	
Días laborados	30	30	30	30	30	30	7	23	
Días restantes	15	30	30	30	30	30	7	23	
Nueva liquidación	930843.65	930843.65	930843.65	217196.85	0	0	0	0	
Retroactivo previo	24400	0	0	0	0	0	0	0	
Retroactivo	26057	71347	71347	0	0	0	0	0	168750
									TOTAL
BONIFICACIÓN DE SERVICIOS PRESTADOS - 29									
Periodos de liquidación	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	
Liquidado	2105948	0	0	0	0	0	0	0	
Días laborados	30	30	30	30	30	30	7	23	
Días restantes	15	30	30	30	30	30	7	23	
Nueva liquidación	2362015.76	0	0	0	0	0	0	0	
Retroactivo previo	61915	0	0	0	0	0	0	0	
Retroactivo	66119	0	0	0	0	0	0	0	66119
									TOTAL

Figura 12: Captura tomada de Libre Office Calc, tabla de pruebas para corroborar que los resultados del sistema sean los correctos.

Otro tipo de pruebas más funcionales se hicieron con datos inventados para comprobar que el resultado sea correcto, o que ocurriera un fallo en el sistema a propósito. Sobre esto último cuando ocurría errores en el proceso de codificación y pruebas, debido a que se desconocía un método para automatizar pruebas que funcionara con el proyecto que estaba montado; para encontrar los fallos se usó la conocida metodología manual, es decir, imprimir datos en pantalla mientras se ejecutaba la aplicación.

5.2.2. Retroactividad Docentes 1279

Debido a que la retroactividad de los públicos no docentes solo funciona con una novedad por periodo de liquidación como se dijo anteriormente, se creó una copia para la retroactividad de los docentes 1279 y se modificó primero su funcionamiento para una sola novedad por periodo (solo cambios menores en consultas y variables), de esta manera tener una base más firme para partir y analizar la situación.

Después de que las pruebas salieran correctas, el siguiente paso fué modificar el código para que funcionara con varias novedades en un solo periodo. Al ya tener una base funcional más clara para una novedad, ocurre el problema que el código tiene demasiada dependencia entre sus métodos y a su vez de la clase “liquidación”, es decir, el código funcionaba de manera muy rígida para una sola novedad en un periodo de liquidación y en su momento no se veía la forma de modificarlo para varias novedades.

Por lo tanto, se pensó en una fórmula que permitiera generar una sola prenomina con todas las novedades y así es como si funcionara con una sola novedad. Se calculaba la retroactividad a partir de la diferencia entre la prenomina con todas las novedades en el periodo de liquidación y la nómina que se pagó, a partir de aquí se hacían los siguientes cálculos que se mostrará a continuación.

Llamamos r la retroactividad igual a la diferencia nombrada, rp_i la retroactividad proporcional a los puntos de cada novedad, D días liquidados de la nómina, p_i los puntos de cada novedad, P puntos totales, d_i días efectivos de la novedad dentro del periodo de liquidación y R la retroactividad definitiva del periodo retroactivo.

Primero se calculaba la proporción de cada novedad:

$$rp_i = r * \frac{p_i}{P}$$

Luego tenemos que, r_i es la retroactividad de cada novedad:

$$r_i = rp_i * \frac{d_i}{D}$$

Sea I la cantidad de novedades en el periodo de liquidación:

$$R = \sum_{i=1}^I rp_i * \frac{d_i}{D}$$

Al aplicar esta fórmula los primeras pruebas funcionaron, dando exacto tanto como para el sueldo básico como para el resto de criterios. Sin embargo, en pruebas posteriores el resultado en el sistema se desfasaba mucho del resultado de la tabla de pruebas.

Cómo ya se explicó anteriormente el funcionamiento de las pruebas, después de que el método anterior no funcionara se optó por realizar la retroactividad basado en las pruebas. A diferencia de las pruebas en donde ya tenemos los datos iniciales y los datos finales del todo el periodo retroactivo, al consultar en la base de datos los puntos iniciales ya no existen debido a las constantes adiciones y anulaciones de novedades. Pero esto no indica que no se pueda obtener, y gracias a que las novedades nunca se borran de la base de datos, podemos llegar por medio de simples sumas y restas al puntaje que se tenía al inicio del periodo de liquidación.

Con estos puntos iniciales podemos generar los salarios que se generan por cada novedad e ir generando la retroactividad iterativamente en cada periodo de liquidación. Este método generaba mayor complejidad, además hacía operaciones innecesarias cuando no existía novedades, por lo que fueron necesarios condicionales para optimizar la operación.

Después de varias iteraciones de pruebas y corrección de errores dados los diferentes casos, ocurrió otro problema, al liquidar la retroactividad esta no se guardaba en nómina por lo que siempre se podía realizar la misma operación de retroactividad sin tener en cuenta que había novedades que ya se habían pagado. Pero este problema, fue más difícil de solucionar ya que el diseño de la base de datos era muy estricto para que solo se pudiera guardar una novedad por criterio de liquidación, y la retroactividad de los docentes comprendía varias novedades.

Se propuso varias soluciones sin intentar modificar el modelo de la base de datos, de donde la única que fue viable era crear otra tabla en la base de datos en donde se guardaran los identificadores de liquidación y de las novedades, lo que en cierta forma era redundante pero permitió mantener el registro de las novedades ya pagadas.

5.2.3. Retroactividad Docentes 115

Lo anterior se duplicó a los docentes de la Resolución 115 y solo cambiando los criterios con que se evalúan los puntos (véase la). Sin embargo al realizar los resultados que debía generar las pruebas se encontró que la retroactividad era negativa, pues el sueldo con los nuevos puntos era menor que la cantidad liquidada en el periodo de liquidación respectivo. Aquí nos dimos cuenta que nuestro cliente (Recursos Humanos) no nos proporcionó toda la información en este caso y que existía una fórmula para calcular el salario a partir de los puntos, la cual se define como:

Sea smc salario mínimo por categoría, vp valor del punto, pp puntos ponderados, pmc puntaje mínimo por categoría y dh dedicación horaria.

$$S = (smc + vp * (pp - pmc)) * \frac{dh}{40}$$

Dado que pp es:

$$pp = \sum_{i=1}^7 c_i * p_i$$

Donde c_i es cada tipo de punto perteneciente a los docentes 115 (estudios títulos obtenidos, labor docente, experiencia profesional, experiencia académica, producción intelectual, idiomas y distinción académica) y p_i su respectivo porcentaje de ponderación.

Sin embargo al aprobar la novedad también fue necesario guardar una novedad de cambio de salario porque, a pesar de que se guardan puntos para los docentes 115, el salario no se genera automáticamente a partir de estos, sino que se guarda el salario de manera manual (como si fuera un empleado público no docente). Esto fue necesario para no cambiar el flujo de cómo estaba funcionando hasta el momento, sin embargo la retroactividad si se manejaba bajo las novedades de nivelación solamente aplicando la fórmula anteriormente descrita.

5.3. Producción

Como se mencionó anteriormente existen cuatro ambientes de trabajo. Después de superar las pruebas en los tres primeros, se pasa a producción donde se hace un seguimiento, actualización y mantenimiento del mismo dadas las normativas o circunstancias cambiantes.

6. CONCLUSIONES

- Fue una experiencia interesante porque nunca se había trabajado con un código tan complejo como lo es la retroactividad, durante el entendimiento del código y parte de la modificación para el problema a resolver fue frustrante pero se logró al final el objetivo.
- Aprender a comunicarse con los otros también fue parte del proceso, no solo para consultar sobre las dudas que se tenían sino también el hecho de poder expresar las ideas correctamente.
- Hubo poca documentación sobre SIRH, lo sabido fue gracias al ingeniero Alexander Varela de la OITEL que ayudó en el proceso y los comentarios en el código (los cuales no eran suficientes para entender).
- Muchas partes del sistema no siguen correctamente el patrón Modelo-Vista-Controlador, además partes del código no están bien moduladas lo que genera demasiada dependencia y más aun al realizar cambios al código.

7. TRABAJOS FUTUROS

- Volver a crear la liquidación en SIRH que permita mayor flexibilidad en las otras funcionalidades que la utilizan, es decir, que disminuya la dependencia y exista un mejor acoplamiento entre los módulos. Por ende al cambiar la liquidación, cambian los módulos que dependen de ella.
- Documentar mediante una API los desarrollos realizados sobre SIRH, para que de esta manera sea más fácil la adaptación a la hora de codificar nuevo código.
- Estandarizar la codificación de las variables, ya que actualmente no se maneja un mismo patrón lo cual dificulta la lectura del código.

8. BIBLIOGRAFÍA

Referencias

- [1] S. S. Abraham Silberschatz, Henry F. Korth, *Fundamentos de Bases de Datos. Cuarta edición*, C. F. Madrid, Ed.
- [2] R. S. Pressman, *Ingeniería de Software Un Enfoque Práctico. Séptima edición*, P. R. Vázquez, Ed., 2013.
- [3] Php. [Online]. Available: <http://php.net/>
- [4] PostgreSQL. (2010) Sobre postgresql. [Online]. Available: http://www.postgresql.org.es/sobre_postgresql
- [5] U. del Valle. (2002, Junio) Decreto 1279. [Online]. Available: http://viceacademica.univalle.edu.co/documentos/normatividad/dec_1279.pdf
- [6] ——. (1989, Septiembre) Resolución 115. [Online]. Available: http://viceacademica.univalle.edu.co/documentos/normatividad/res_115_cs.pdf
- [7] B. Eckel, *Thinking on C++, Volume 1, 2nd edition*, A. Stevens, Ed., 2000.
- [8] Wikipedia. (2014, Diciembre) Programación orientada a objetos. [Online]. Available: http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

9. ANEXOS

9.1. Historias de Usuario

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Acceder a la novedad de nivelación de los docentes 1279 y 115	Identificador	1
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir acceder a la novedad de nivelación para los docentes del Decreto 1279 y la Resolución 115 dentro del grupo de novedades de vinculación laboral.			
Observaciones			
Solo los actores pueden ver el acceso a esta novedad.			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 4: Acceder a la novedad de nivelación de los docentes 1279 y 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Crear y guardar novedades de nivelación de los docentes 1279	Identificador	2
Actor(es)	Rol de operario recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir crear y guardar en la base de datos novedades de nivelación para los docentes del Decreto 1279 dentro del grupo de novedades de vinculación laboral mediante un formulario basado en el formato de puntos de dichos profesores con los siguientes campos: empleado, fecha a partir de la cual es vigente la novedad; puntos por títulos obtenidos, categoría, experiencia calificada, producción académica, dirección académica, asignación adicional, docencia destacada y diferencia puntos.			
Observaciones			
Debe validar la información ingresada y no permitir ingresar una novedad con la misma fecha de vigencia que otra.			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 5: Crear y guardar novedades de nivelación de los docentes 1279.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Crear y guardar novedades de nivelación de los docentes 115	Identificador	3
Actor(es)	Rol de operario recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir crear y guardar en la base de datos novedades de nivelación para los docentes de la Resolución 115 dentro del grupo de novedades de vinculación laboral mediante un formulario basado en el formato de puntos de dichos profesores con los siguientes campos: empleado, fecha a partir de la cual es vigente la novedad; puntos por títulos obtenidos, labor docente, experiencia profesional, experiencia académica, producción intelectual, idiomas y distinción académica.			
Observaciones			
Debe validar la información ingresada y no permitir ingresar una novedad con la misma fecha de vigencia que otra.			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 6: Crear y guardar novedades de nivelación de los docentes 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Aprobar novedades de nivelación de los docentes 1279 y 115	Identificador	4
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir aprobar novedades siguiendo el Flujo Básico (operario recursos humanos - recursos humanos) y al final del flujo hacer efectivo los cambios de puntaje.			
Observaciones			
Para los docentes 115 también se genera dentro del sistema una novedad de cambio de salario (ya existente) para no generar conflictos.			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 7: Aprobar novedades de nivelación de los docentes 1279 y 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Anular novedades de nivelación de los docentes 1279 y 115	Identificador	5
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir anular novedades siguiendo el Flujo Básico (operario recursos humanos - recursos humanos) haciendo un borrado lógico en la base de datos.			
Observaciones			
<ul style="list-style-type: none"> ■ Si se aprueba las novedades al final del flujo se deshace los cambios hechos en los puntos totales. ■ Para los docentes 115 se anula también la novedad de cambio de salario. 			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 8: Anular novedades de nivelación de los docentes 1279 y 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	novedadporpersona
Nombre HU	Consultar novedades de nivelación de los docentes 1279 y 115	Identificador	6
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Media
Descripción			
El sistema debe permitir consultar las novedades de nivelación de los docentes del Decreto 1279 y la Resolución 115, y dar información del estado en que se encuentran dentro del flujo (en espera de aprobación, aprobada o anulada).			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 9: Consultar novedades de nivelación de los docentes 1279 y 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Realizar retroactividad individual o por grupo de los docentes 1279 y 115	Identificador	7
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe permitir realizar la retroactividad tanto para un solo docente, como para el grupo, ya sea docentes del Decreto 1279 o de la Resolución 115.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 10: Realizar retroactividad individual o por grupo de los docentes 1279 y 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Realizar retroactividades de novedades de nivelación que no han sido liquidadas	Identificador	8
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe realizar solamente retroactividades de las novedades de nivelación que aún no han sido liquidadas.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	–/–/–

Tabla 11: Realizar retroactividades de novedades de nivelación que no han sido liquidadas.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Calcular el sueldo para los docentes 1279	Identificador	9
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe calcular el sueldo multiplicando el último valor del punto por la cantidad de puntos del docente en el periodo que se este procesando.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/-

Tabla 12: Calcular el sueldo para los docentes 1279.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Calcular el sueldo para los docentes 115	Identificador	10
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe calcular el sueldo usando la siguiente fórmula: $S = (smc + vp * (pp - pmc)) * \frac{dh}{40}, \text{ donde } pp = \sum_{i=1}^7 c_i * p_i.$			
Observaciones			
<ul style="list-style-type: none"> ▪ S es sueldo. ▪ smc es sueldo mínimo por categoría. ▪ vp es valor del punto. ▪ pp es puntaje ponderado. ▪ c_i es cada criterio de punto: títulos obtenidos, labor docente, experiencia profesional, experiencia académica, producción intelectual, idiomas y distinción académica. ▪ p_i es la ponderación por cada criterio respectivo. ▪ pmc es el puntaje mínimo por categoría. ▪ dh es la dedicación horaria. 			
Fecha creación	12/03/2014	Fecha modificación	05/12/2014

Tabla 13: Calcular el sueldo para los docentes 115.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Evaluar la retroactividad con los todos los criterios involucrados	Identificador	11
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe calcular la retroactividad con todos los criterios involucrados en cada periodo de liquidación que se encuentran dentro del periodo retroactivo, tanto periódicos (sueldo básico, gastos de representación, bonificación de servicios prestados, bonificación de productividad académica, etc.) como anuales (prima antigüedad, prima servicios, vacaciones, prima vacaciones, etc.).			
Observaciones			
<ul style="list-style-type: none"> ■ Algunos criterios periódicos no se evalúan en cada periodo de liquidación como por ejemplo los gastos de representación, el cual solo se evalúa si el docente posee un cargo administrativo en dicho periodo. ■ Algunos criterios anuales no necesariamente se cumplen cada año como por ejemplo las vacaciones, ya que estas pueden ser adelantadas. 			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 14: Evaluar la retroactividad con los todos los criterios involucrados.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Ignorar los valores retroactivos negativos en la sumatoria de retroactivos de cada periodo de liquidación	Identificador	12
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema no debe tener en cuenta valores retroactivos negativos que se calculen en cada periodo de liquidación.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 15: Ignorar los valores retroactivos negativos en la sumatoria de retroactivos de cada periodo de liquidación.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Identificar las novedades que estén involucradas en retroactividades liquidadas	Identificador	13
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe almacenar en la base de datos las novedades con su respectiva retroactividad liquidada en la base de datos para que se puedan identificar en futuros procesos retroactivos.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 16: Identificar las novedades que estén involucradas en retroactividades liquidadas.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Almacenar el retroactivo total de cada criterio de liquidación	Identificador	14
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe almacenar en la base de datos los retroactivos totales de cada criterio de liquidación al final del periodo retroactivo.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 17: Almacenar el retroactivo total de cada criterio de liquidación.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Liquidar correctamente el retroactivo generado	Identificador	15
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Alta
Descripción			
El sistema debe liquidar correctamente el retroactivo liquidado, el cual es el total de la suma de los retroactivos de cada criterio.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 18: Liquidar correctamente el retroactivo generado.

Historia de usuario			
Autor	Cristhian David Marín	Módulo	retroactividad
Nombre HU	Generar reportes del proceso retroactivo	Identificador	16
Actor(es)	Rol de operario recursos humanos, rol de recursos humanos	Responsable	Cristhian David Marín
Iteración	–	Prioridad	Media
Descripción			
El sistema debe generar un reporte o reportes (en el caso de que se haga por grupos) al final del proceso de retroactividad discriminando el resultado por cada criterio de liquidación.			
Observaciones			
Fecha creación	12/03/2014	Fecha modificación	-/-/—

Tabla 19: Generar reportes del proceso retroactivo.

9.2. Modelo E-R

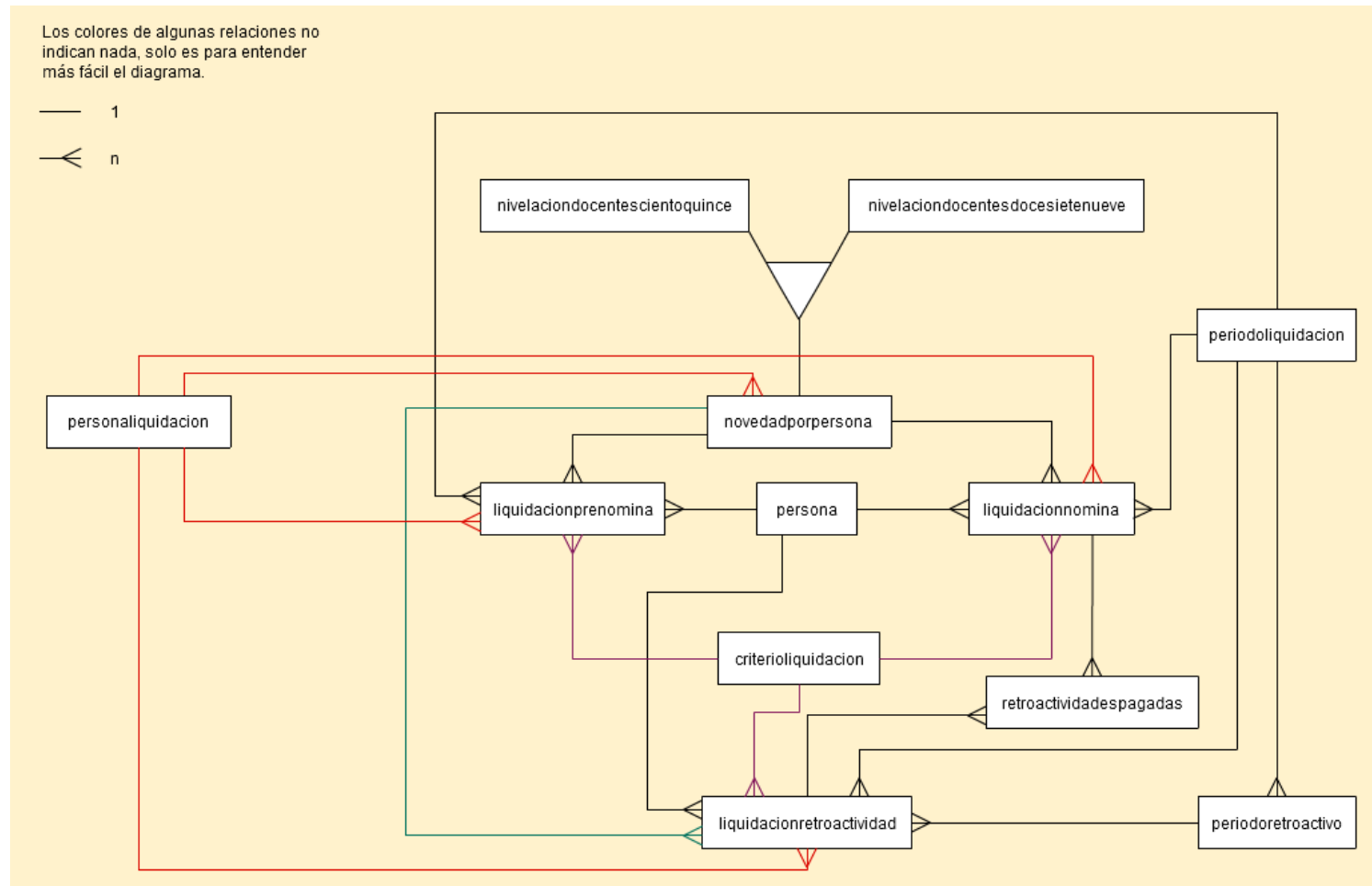


Figura 13: Modelo E-R de la base de datos.

9.3. Diagramas de Navegación

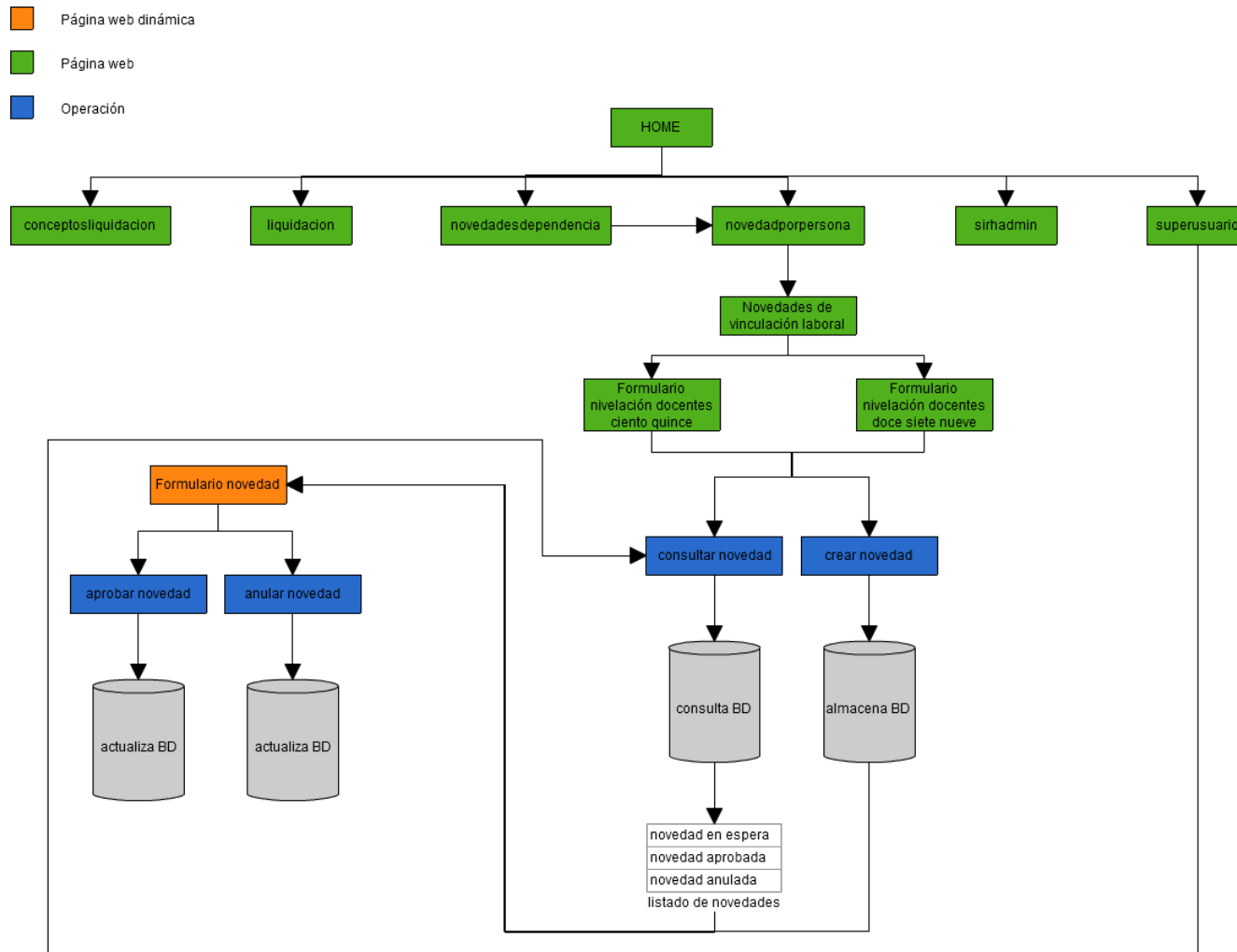


Figura 14: Diagrama de navegación novedades.

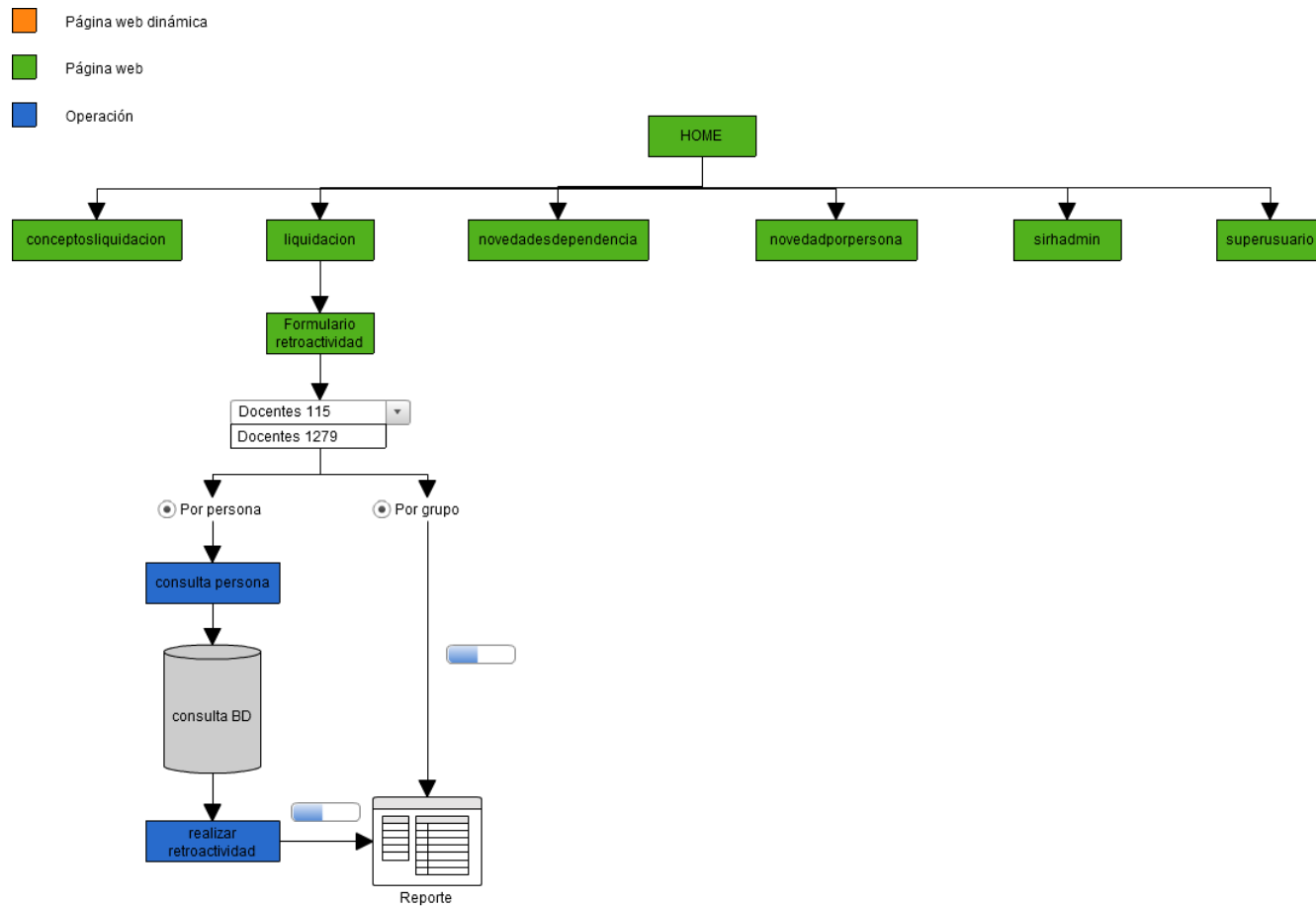


Figura 15: Diagrama de navegación retroactividad.