DESARROLLO DE LA SOLUCIÓN DE SOFTWARE PARA APOYAR EL PROCESO DE GESTIÓN DE NÓMINA DE DOCENTES DE HORA CÁTEDRA EN LA UNIVERSIDAD DISTRITAL, SIGUIENDO LOS LINEAMIENTOS DEL PROCESO DE DESARROLLO SCRUM EN SU FASE DE INICIO, ELABORACIÓN Y CONSTRUCCIÓN

AUTOR:

JOSÉ MIGUEL RAMÍREZ SÁNCHEZ

PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTOR:

ING. ALEJANDRO PAOLO DAZA

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ, D.C
MAYO 2017

DESARROLLO DE LA SOLUCIÓN DE SOFTWARE PARA APOYAR EL PROCESO DE GESTIÓN DE NÓMINA DE DOCENTES DE HORA CÁTEDRA EN LA UNIVERSIDAD DISTRITAL, SIGUIENDO LOS LINEAMIENTOS DEL PROCESO DE DESARROLLO SCRUM EN SU FASE DE INICIO, ELABORACIÓN Y CONSTRUCCIÓN

AUTOR:

JOSÉ MIGUEL RAMÍREZ SÁNCHEZ

PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTOR:

ING. ALEJANDRO PAOLO DAZA

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BOGOTÁ, D.C
MAYO 2017

Índice General

CAPÍTULO 1. INTRODUCCIÓN.....1
CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA.....2
CAPÍTULO 3. OBJETIVOS.....3
3.1. Objetivo general.....3
3.2. Objetivos específicos.....3
CAPÍTULO 4. JUSTIFICACIÓN.....4
CAPÍTULO 5. MARCO TEÓRICO.....5
5.1. Metodología de desarrollo ágil SCRUM.....5
5.1.1. Componentes de SCRUM.....6
5.1.2. Fases.....7
5.1.3. Roles.....7
5.2. Elementos de SCRUM.....8
5.2.1. Product Backlog:.....8
5.2.2. Sprint Backlog:.....9
5.2.3. Sprint (Iteracion):.....9
5.2.4. Sprint Planning Meeting (Planificación de Sprint):.....10
5.2.5. Scrum Diario:.....11
5.2.6. Revisión de Sprint:.....12
5.2.7. Feedback:.....13
5.2.8. Release:.....14
5.2.9. Identificación de Funcionalidades del Software.....14
5.3. Historias de usuario.....14
5.3.1. Componentes de historia de usuario:.....14
5.3.2. Redacción de historia de usuario:.....15
5.3.3. Definición de terminado:.....15
5.4 Elementos, partes y estructura básica de la Nomina.....16
5.4.1 Salario Bruto.....17
5.4.2 Descuentos en la nomina.....18
CAPÍTULO 6. ALCANCES Y LIMITACIONES.....19
6.1. Alcances.....19

6.2. Limitaciones.....19
CAPÍTULO 7. METODOLOGÍA.....20
CAPÍTULO 8. RECURSOS.....25
CAPÍTULO 9. PRESUPUESTO.....26
CAPÍTULO 10. CRONOGRAMA.....27
CAPÍTULO 11. MODELO DE NEGOCIO.....29
11.1 Registro de Persona en el Banco de Proveedores.....29
11.2 Registro del Contrato de Docente de Vinculación especial en el sub-sistema de contratación......29
11.3 Registro de Nominas.....30
11.4 Registro y Calculo de Preliquidaciones......32
11.5 Registro de Novedades......42
11.6 Registro de Liquidación......44
CAPÍTULO 12. MODELO DE DATOS.....45
12.1 Descripción del esquema TITAN......47
12.1.1 Tabla nomina.....47
12.1.2 Tabla Preliquidación.....47
12.1.3 Tabla Detalle Preliquidación.....47
12.1.4 Tabla Concepto.....48
12.1.5 Tabla Concepto por Persona.....48
12.1.6 Tabla Liquidación.....48
12.1.7 Tabla Detalle Liquidación.....48
12.1.8 Tabla Tipo Nomina.....48
12.1.8 Tabla Tipo Vinculación.....48
CAPÍTULO 13. ARQUITECTURA DE LA APLICACIÓN.....49
13.1 Arquitectura General.....49
CAPÍTULO 14. CONCLUSIONES.....57
CAPÍTULO 15. BIBLIOGRAFIA.....58

Índice de Figuras

Figura 1 Epic Nomina Docentes Vinculación Especial. 21
Figura 2 Epic Arquitectura TITAN. 21
Figura 3 Ejemplo Historia de Usuario. 22
Figura 4 Ejemplo Registro diario de Actividades. 23
Figura 5 Cronograma de Actividades. 27
Figura 6 Diagrama de Proceso: Registro de una Nóminas. 31
Figura. 7 Registro de preliquidación. 33
Figura. 8 Generación de preliquidación. 35
Figura. 9 Diagrama de flujo: calculo preliquidación HC Salarios. 37
Figura. 10 Diagrama de flujo: calculo preliquidación HC Honorarios. 38
Figura. 11 Registro de una novedad. 43
Figura. 12 Registro de la Liquidación. 44
Figura. 13 Esquema Titán. 46
Figura 14. Diagrama de la Arquitectura general del sistema. 49
Figura 15. Diagrama de arquitectura de los sub-sistemas que interviene en la nómina. 51
Figura 16 Arquitectura de la información RULER. 52
Figura 17 Arquitectura Interna TITAN. 53
Figura 18 Módulo de cálculo de liquidación. 55

Índice de Tablas

Tabla 1 Recursos del Proyecto**.** 25
Tabla 2 Presupuesto del proyecto, rol desarrollador**.** 26
Tabla 3 Tarifas retención en la fuente. 40
Tabla 4 Descuentos por parafiscales obligatoria. 41
Tabla 5 Devengos por finalización del contrato. 41

CAPÍTULO 1. INTRODUCCIÓN

La Universidad Distrital Francisco José de Caldas es un ente autónomo de Educación Superior fundada en el año 1948, cuya misión se centra en las funciones básicas de docencia, investigación y extensión; para cumplir a cabalidad con estas funciones, requiere del apoyo de talento humano como docentes, administrativos y contratistas, quienes deben tener una vinculación contractual con la Institución para que puedan ser retribuidos monetariamente por sus servicios. (misión universidad distrital, s.f.)

Para cualquier persona natural vinculada a la universidad bajo cualquiera de los tipos de vinculación que en esta se manejen, se debe efectuar de forma periódica un proceso de gestión de nómina integral, donde se realice la liquidación de los correspondientes salarios, honorarios, prestaciones, aportes patronales, seguridad social, prestaciones sociales y para fiscales según sea necesario.

El proceso que se lleva a cabo en la actualidad para la gestión de nómina en la institución varía según el tipo de vinculación contractual que tenga la persona natural. Con el fin de llevar a cabo la ejecución de cada uno de los procesos subyacentes de las nóminas para cada tipo de vinculación la universidad actualmente cuenta con diferentes herramientas informáticas, estas van desde hojas de cálculo de Excel hasta herramientas más robustas con soporte en bases de datos Oracle, en donde ambos casos tienen como problemas comunes la baja interoperabilidad con otros componentes de software que son requeridos para realizar los pagos y causaciones financieras.

Con el fin de realizar mejoras y unificación en el proceso de gestión de nómina de la universidad distrital, la oficina asesora de sistema (OAS), en su potestad de crear nuevas soluciones de software para el ente educativo, crea el proyecto denominado “Solución de software para apoyar el proceso de gestión de nómina de la universidad distrital”, realizando un proceso de llamado y selección a estudiantes de últimos semestres que cursan ingeniería de sistemas que deseen participar en el mismo, para llevar el rol de desarrolladores.

Este proyecto en el que se participará con el rol de desarrollador, tiene como objetivo analizar, diseñar y construir una solución modular, integral y escalable que permita apoyar los procesos relacionados a la gestión de nóminas de la Universidad Distrital, soportado en tecnologías libres siguiendo el proceso de desarrollo SCRUM y de esta manera contar con una única herramienta tecnológica que soporte cualquier tipo de vinculación contractual con la Institución.

CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA

En cualquier entidad, ya sea de carácter público o privado, es necesario llevar un control sobre los diferentes factores que afectan el pago a sus empleados, ya que la remuneración de las actividades laborales por parte de la entidad a sus funcionarios garantiza el cumplimiento de las funciones que estos desempeñan y por tanto acercan a la entidad en cuestión al cumplimiento de su misión y visión. Dependiendo del tamaño de la organización así mismo será la robustez y complejidad del proceso de pago a sus empleados, además, se deben tener en cuenta las diferentes leyes impuestas por estatutos gubernamentales y normativas internas de la propia organización para realizar el proceso de liquidación y pago de los salarios a sus funcionarios, lo cual hace que dicho proceso crezca a un más en complejidad.

Debido a lo anterior la Universidad Distrital Francisco José de Caldas como entidad estatal también posee procesos de liquidación para el pago de sus empleados y por su tamaño este proceso es bastante complejo. La Oficina Asesora de Sistemas (OAS) detecto que este proceso dentro de la Universidad se realizaba de manera poco eficaz ya que actualmente en la universidad se cuenta con varios tipos de vinculación de sus funcionarios, para los cuales el proceso de liquidación es trabajado de forma independiente generando inconvenientes de tiempo, control y complejidad a la hora de realizar este proceso. Por tanto la OAS en el proceso de centralización de la información en la universidad ha decidido poner en marcha el proyecto TITAN en su primera versión, junto a otros proyectos que buscan solucionar los problemas de redundancia de datos y agilizar los procesos de vinculación, liquidación, pago de nóminas entre otros dentro de la universidad. Debido a que actualmente no se cuenta con la anteriormente mencionada centralización de datos, se hace necesario realizar una tediosa validación de la información para generar los respectivos pagos a los funcionarios, esto porque en la universidad se manejan distintos tipos de información, en diferentes dependencias, relacionada con el proceso de liquidación de las nóminas.

Los anteriormente mencionados problemas ocasionan que el proceso de liquidación de las nóminas que actualmente se lleva a cabo dentro de la universidad tenga demoras innecesarias, afectando el proceso de liquidación y pago a las personas naturales que cuentan con vinculación contractual con la universidad, como lo son los funcionarios, docentes y contratistas, o aquellos a cargo de procesos de liquidación y pago en áreas de recursos humanos y financieros.

CAPÍTULO 3. OBJETIVOS

3.1. Objetivo general

Desarrollar un software modular, integral y escalable que permita apoyar los procesos relacionados a la gestión de nóminas de las personas naturales vinculadas como profesores de horas catedra en la Universidad Distrital Francisco José de Caldas, siguiendo los lineamientos propios del proceso de desarrollo SCRUM.

3.2. Objetivos específicos

- Desarrollar entregas parciales y funcionales del producto las cuales sean realizadas según su prioridad y permitan evidenciar avances concretos en la realización de la solución de software final, para cumplir así con los objetivos de cada una de las iteraciones que componen el proceso de desarrollo SCRUM

- Aplicar pruebas a cada una de las entregas parciales desarrolladas a través del proceso, determinando si se están cumpliendo o no los requerimientos y especificaciones determinadas en los procesos de análisis y diseño, para realizar las correcciones pertinentes a tiempo y así obtener un producto de software que esté acorde a las peticiones de usuarios e interesados en el modelo de negocio de sistema de nómina de horas catedra.

- Realizar la documentación del proceso de desarrollo basada en los requerimientos y la arquitectura planteadas por proyectos anteriores, para llevar registros completos de lo que está siendo realizado y que estos sirvan como referencia para futuros equipos de desarrollo que necesiten entender y utilizar el producto de software resultante.

- Complementar los procesos de análisis, levantamiento de requerimientos y arquitectura realizados en proyectos anteriores desde el punto de vista del desarrollador para generar cambios pertinentes que mejoren y aporte a la más completa construcción del producto de software resultante.

CAPÍTULO 4. JUSTIFICACIÓN

Realizar la liquidación de nómina en cualquier entidad, ya sea pública o privada, es una de las partes fundamentales en su funcionamiento, ya que de esto depende el pago oportuno a los empleados que la conforman por las labores realizadas durante un mes específico. En la actualidad, la universidad Distrital Francisco José de Caldas no cuenta con un sistema integral de nómina que unifique el proceso que se lleva a cabo para la liquidación y generación de órdenes de pago para los diferentes tipos de contrato bajo los cuales están vinculados los empleados de la misma, el proceso que se lleva en la actualidad para realizar la liquidación de las nóminas de la universidad es tecnológicamente atrasado, lo cual genera problemas en la integridad, seguridad y agilidad del proceso, que a futuro puede verse reflejado en problemas administrativos afectando económica y administrativamente las actividades de la universidad.

Debido a lo anterior, en la universidad la Oficina Asesora de Sistemas (OAS) decidió poner en marcha el proyecto Titán en paralelo con otros proyectos que no solo buscan mejorar el proceso de liquidación de las nóminas en la universidad, sino que también buscan la centralización de la información que se maneja dentro de la misma universidad, reduciendo la replicación de datos en las bases de datos y agilizando los procesos administrativos dentro de la universidad.
Durante el desarrollo del proyecto Titán se deben seguir los lineamientos propuestos por los arquitectos de software, los cuales se enfocan en la adaptabilidad que debe tener el software a los cambios legales tanto internos como externos que afecten la generación de las liquidaciones, la fácil usabilidad que debe tener el producto final, la fiabilidad de los resultados de los procesos que se realicen dentro de este y la debida documentación de los módulos desarrollados.

La OAS está utilizando la metodología ágil Scrum, donde el rol a tomar por parte del autor de este anteproyecto es el de desarrollador que, además de generar los artefactos de software necesarios para el proyecto, también analizan y adaptan los documentos generados anteriormente por los arquitectos, todo esto debido a los requerimientos cambiantes del proyecto y de algunos detalles de la implementación.

CAPÍTULO 5. MARCO TEÓRICO

5.1. Metodología de desarrollo ágil SCRUM

A mediados de los 80 Takeuchi y Nonaka publican un artículo titulado “The New Product Developroent Game” en el cual se hace énfasis en una nueva forma de gestionar proyectos cuyos elementos principales son la agilidad, flexibilidad y la incertidumbre.

En principio Nonaka y Takeuchi estudiaron empresas tecnológicas que aun estando en el mismo entorno de otras, estas tenían la capacidad de realizar productos en menos tiempo, con una buena calidad y con costes inferiores.

De este modo observaron empresas como Honda, HP, Cannon entre otras y se percataron de que en estas empresas sus productos no seguían un proceso donde dentro cada una de sus fases era supervisada por un equipo especializado, más bien se partía de algunos requisitos muy generales y el producto era realizado por un equipo multidisciplinar que seguía el proceso desde el comienzo hasta el final. Esta forma de trabajo en equipo fue comparada con la colaboración que hacen los jugadores de Rugby en una formación denominada SCRUM (Manuel, T ., Cristina, D, 2014, p.32).

SCRUM hace su primera aparición como metodología destinada a los productos tecnológicos en 1993 debido a que Jeff Sutherland la aplico a un modelo de desarrollo de software en Ease/Corporation.

SCRUM es una metodología adecuada para aquellas empresas en donde el desarrollo de productos se realiza en entornos que se caracterizan por tener:

- **Incertidumbre:** no se plantean objetivos detallados, solo una idea general. Este hecho genera un reto y da autonomía al equipo, dando también cierto grado de motivación al mismo.

- **Auto-Organización:** en este caso los equipos no necesitan de roles para su gestión interna, pero para esto el equipo en conjunto debe ser autónomo en la consecución de estrategias que solucionen los problemas que se les presenten durante el desarrollo del producto, deben tener instinto de auto-superación, deben estar en permanente mejora de las soluciones desarrolladas con anterioridad y deben tener auto-enriquecimiento, donde al ser equipos multidisciplinares se ven auto-complementados los conocimientos de cada uno de los miembros.
- **Control Moderado:** los controles son suficientes para evitar problemas dentro de los grupos de trabajo, en este caso se basa en crear escenarios de autocontrol mutuo entre los integrantes lo que permite una alta generación de creatividad y espontaneidad.
- **Transmisión del conocimiento:** los conocimientos generados en los proyectos de la organización son transmitidos entre los miembros de cada uno de estos.

Como SCRUM es una metodología de desarrollo ágil, se basa en la creación de ciclos breves para el desarrollo, tradicionalmente llamadas iteraciones pero que en SCRUM también son conocidos como “sprints”.

Como SCRUM es una metodología de desarrollo ágil, como se mencionó con anterioridad, sus ciclos se componen de las siguientes fases:

- **Concepto:** Se debe definir el producto al que se quiere llegar con el proyecto y de ser necesario sus sub-proyectos para que posteriormente sea asignado a un equipo o equipos los cuales se encargaran de su desarrollo.
- **Especulación:** En esta fase se hace una recopilación de la información obtenida en el proceso y se establecen los límites para el desarrollo del producto como costes, agendas y límites de tiempo.
- **Exploración:** Se empiezan a añadir las características del producto que emergieron en la fase de Especulación.
- **Revisión:** En esta fase el equipo de trabajo se reúne para revisar y analizar los avances que se han tenido respecto a los objetivos planteados para el producto.
- **Cierre:** Esta fase es en donde se hace la entrega de una versión del producto objetivo del proyecto, como lo que se entrega en esta fase puede ser una versión no final del producto, esta no necesariamente marca el final del proyecto, sino una entrega de una versión a la cual se le seguirán haciendo cambios para acercar su funcionamiento al producto final deseado.

5.1.1. Componentes de SCRUM

Para entender de una manera apropiada la metodología, es necesario describir las fases de las que está compuesta así como los roles de cada uno de los roles que se asumen dentro de la misma, a continuación se describirán estos ítems involucrada (llera S, Fernando L, 2014, p.35).

5.1.2. Fases

1. **Planificación del Backlog:** Esta es la fase inicial del proyecto, en esta se define un documento en donde se verán registrados cuales son los requisitos del sistema a desarrollar asignando a cada uno de estos su respectiva prioridad.
   Además se planificara el Sprint 0 del proyecto, definiendo las tareas para el grupo en esa iteración inicial.
2. **Seguimiento del Sprint:** Para esta fase se hace revisiones diarias del avance en las tareas asignadas para revisar que tanto se ha avanzado en estas, que tareas podrían asignarse para la siguiente iteración y que inconvenientes han surgido y cuales se deben solucionar para continuar con el desarrollo del proyecto.
3. **Revisión del Sprint:** Para cada finalización de un Sprint, se hace una revisión de los avances que se han tenido respecto al producto final deseado y se realizan pequeñas demostraciones de estos avances.

5.1.3. Roles

Los roles dentro de esta metodología están definidos en 2 tipos, los cuales son inspirados en el chiste sobre un cerdo y una gallina que quieren montar un restaurante y que de forma resumida concluye que el cerdo está comprometido y la gallina apenas involucrada (llera S, Fernando L, 2014, p.35), a continuación se describen estos roles:

1. **Los cerdos:** Son los involucrados en el proyecto , que como el cerdo del chiste, están comprometidos con el mismo y con el proceso de SCRUM:
   a. **Product Owner:** Es quien conoce todo lo relacionado con el cliente y lo que este quiere como producto final, es quien se encarga de describir las ideas del cliente para el proyecto y ordenarlas por prioridad en el Product Backlog.
   b. **ScrumMaster:** Es quien se encarga de velar porque el proceso de desarrollo y la metodología funcionen, también se encarga de solucionar cualquier inconveniente que afecte el desarrollo del proyecto e interactúa con el cliente.

    c. **Equipo de desarrollo:** Por lo general es un grupo pequeño de personas, los cuales tienen autoridad para organizar y tomar decisiones para conseguir su objetivo.
2. **Las gallinas:** Son los involucrados en el proyecto que no son parte del proceso de SCRUM pero a los cuales es necesario realizar una retroalimentación del desarrollo para poder revisar y planear cada Sprint:
    a. **Usuarios:** Son los destinatarios del producto final.
    b. **Stakeholders:** Son los interesados en el proyecto, a quienes el resultado final del mismo les brindara algún tipo de beneficio.
    c. **Managers:** Son quienes toman las decisiones finales, también son quienes los objetivos y requisitos que tiene el proyecto.

5.2. Elementos de SCRUM

SCRUM tiene una cantidad mínima de elementos formales debido a su naturaleza ágil, a continuación se describe cada uno de ellos:

5.2.1. Product Backlog:

Este es el elemento principal de SCRUM, el cual también es conocido como Pila del Producto. Este es una lista de las características que debe tener el producto el cual va enfocado el proyecto, mantenida y priorizada por el Product Owner. La priorización que se le dé a cada característica es muy importante debido a que según esta es como se determinara el orden en el cual el equipo de trabajo del proyecto transformara estas características en el producto funcional acabado.
La prioridad que se le da a cada ítem es responsabilidad única del Product Owner, ya que este es quien conoce el negocio a profundidad, el equipo de trabajo puede dar algunas sugerencias respecto a la prioridad que se le dan a los elementos del Product Backlog pero es el Product Owner quien tiene la última palabra sobre la asignación de las prioridades de la lista (Oficina Asesora de Sistemas, 2016, p.17).

Existen otras formas de medir la prioridad de los ítems del Product Backlog, una de ellas es el cálculo del beneficio del beneficio económico generado en función a la inversión realizada en para el cumplimiento del ítem. Aunque se trata de una operación matemática simple, el cálculo del valor generado por la adición de una característica al producto se hace realmente compleja, pero una vez identificada, el cálculo se hace realmente fácil.

Sin importar si se realiza la priorización de los ítems del Product Backlog se realiza por alguno de los métodos anteriormente descritos, cada ítem también se podrá

ver afectado por el riesgo asociado a cada uno. De esta manera la construcción iterativa de cada característica mitigando riesgos de forma implícita, dejando los ítems con mayor nivel de riesgo al principio y las de menor riesgo para etapas futuras.

Debido a lo anterior, la lista es ordenada por nivel de riesgo dejando los ítems de mayor nivel de riesgo al principio con un nivel de detalle mayor, ya que estos son los más susceptibles a ser alterados o reemplazados.

5.2.2. Sprint Backlog:

Esta es una lista de algunos ítems del Product Backlog seleccionados para trabajar sobre ellos durante un Sprint en específico.
El resultado de cada Sprint debe ser un incremento funcional potencialmente entregable. *Incremento funcional* porque es una característica funcional nueva (o modificada) de un producto que está siendo construido de manera evolutiva. El producto crece con cada Sprint. *Potencialmente entregable* porque cada una de estas características se encuentra lo suficientemente validada y verificada como para poder ser desplegada en producción (o entregada a usuarios finales) si así el negocio lo permite o el cliente lo desea involucrada (llera S, Fernando L, 2014, p.40).

5.2.3. Sprint (Iteracion):

Las iteraciones en Scrum se conocen como Sprints. Scrum, como todos los enfoques ágiles, es un proceso de desarrollo incremental e iterativo. Esto significa que el producto se construye en incrementos funcionales entregados en periodos cortos para obtener feedback frecuente.

En general, Scrum tiene una duración de Sprint de entre 1 y 2 semanas y el objetivo será mantener esta duración constante a lo largo del desarrollo del producto, lo que implicará que la duración de una iteración no cambie una vez que sea establecida.

Como excepción se pueden presentar aquellas situaciones donde el equipo mismo decida probar con iteraciones más largas o más cortas, pero siempre entre 1 y 2 semanas. Esta decisión se basa principalmente en la volatilidad del contexto: mientras más volátil sea (negocio cambiante, requerimientos desconocidos, etc.) más corta será la duración del Sprint. Lo importante es recordar que se logra mayor ritmo y previsibilidad teniendo Sprints de duración constante.

Se encontrarán situaciones en donde el equipo de desarrollo se atrase o se

adelante. En estos casos, la regla del timeboxing no nos permitirá modificar (adelantar o postergar) la fecha de entrega o finalización del Sprint. La variable de ajuste en estos casos será el alcance del Sprint, esto es, en el caso de adelantarse se deberá incrementar el alcance del Sprint agregando nuevos Product Backlog Items (PBIs) y reducirlo en el caso de retraso.

5.2.4. Sprint Planning Meeting (Planificación de Sprint):

Al comienzo de cada Sprint se realiza una reunión de planificación del Sprint donde serán generados los acuerdos y compromisos entre el equipo de desarrollo y el Product Owner sobre el alcance del Sprint.

Esta reunión de planificación habitualmente se divide en dos partes con finalidades diferentes: una primera parte estratégica y enfocada en el “qué”, y una segunda parte táctica cuyo hilo conductor principal es el “cómo”.

Parte uno: ¿Qué trabajo será realizado? El objetivo buscado durante esta parte de la reunión es identificar “qué” es lo que el equipo de desarrollo va a realizar durante el Sprint, es decir, todos aquellos Product Backlog Items (PBIs) que el equipo se comprometerá a transformar en un producto funcionando y utilizable o en otras palabras: incremento funcional potencialmente entregable.

El Product Owner y el equipo de desarrollo deben participar de esta parte de la reunión como protagonistas principales. El Scrum Master, al tiempo que facilita la reunión, también debe asegurar que cualquier stakeholder del proyecto que sea requerido para profundizar en detalles esté presente o sea contactado.

El equipo de desarrollo utiliza su capacidad productiva (también conocida como Velocidad o Velocity), obtenida de los Sprints pasados, para conocer hasta cuánto trabajo podría comprometerse a realizar. Esto determinaría en un principio cuáles serían los Product Backlog Items (PBIs) comprometidos en este Sprint. La razón es que cada uno de los ítems del Product Backlog debe ser discutido para entender cuáles son sus criterios de aceptación y así conocer en detalle qué se esperará de cada uno. De esta manera, el equipo de desarrollo discutirá con el Product Owner sobre cada Product Backlog Items (PBIs) y generará un compromiso de entrega para aquellos que considera suficientemente claros como para comenzar a trabajar y que además podrían formar parte del alcance del Sprint que está comenzando. A esto se lo conoce como planificación basada en compromisos o Commitment-based Planning. Al finalizar esta primera parte de la reunión, los stakeholders involucrados (si los hubiese) se retirarán, dejando así al

producto Owner, al Scrum Master y al equipo de desarrollo para que den comienzo a la segunda parte de esta reunión, que se describe a continuación.

Parte dos: ¿Cómo será realizado el trabajo? Durante este espacio de tiempo el equipo de desarrollo determinará la forma en la que llevará adelante el trabajo. Esto implica la definición inicial de un diseño de alto nivel, el cual será refinado durante el Sprint mismo y la identificación de las actividades que el equipo en su conjunto tendrá que llevar a cabo.

Se espera que el diseño sea emergente, es decir, que surja de la necesidad del equipo de desarrollo a medida que éste avance en el conocimiento del negocio. Por esta misma razón es que indicamos la no necesidad de realizar un diseño completo y acabado de lo que será realizado durante el Sprint. En cambio, se buscará un acuerdo de alto nivel que será bajado a detalle durante la ejecución de la iteración.

Esto mismo sucede con las actividades del Sprint, es decir que no es estrictamente necesario enumerar por completo todas las actividades que serán realizadas durante la iteración ya que muchas aparecerán a medida que se avanza. Adicionalmente, las actividades deben durar un día. Esto permitirá detectar bloqueos o retrasos durante las reuniones diarias.

Al finalizar esta reunión, el equipo habrá arribado a un Sprint Backlog o Commited Backlog que representa el alcance del Sprint en cuestión. Este Sprint Backlog es el que se coloca en el taskboard (pizarra de actividades) del equipo. Se dará comienzo al desarrollo del producto para este Sprint (Oficina Asesora de Sistemas, 2016, p.12).

5.2.5. Scrum Diario:

Uno de los beneficios de Scrum está dado por el incremento de la comunicación dentro del equipo de proyecto. Esto facilita la coordinación de acciones entre los miembros del equipo de desarrollo y el conocimiento “en vivo” de las dependencias de las actividades que realizan.

Por otro lado, se requiere además aumentar y explicitar los compromisos asumidos entre los miembros del equipo de Proyectos Ágiles con Scrum desarrollo y dar visibilidad a los impedimentos que surjan del trabajo que está siendo realizando y que muchas veces nos impiden lograr los objetivos (Oficina Asesora de Sistemas, 2016, p.13).

Estos tres objetivos: 1) incrementar la comunicación 2) explicitar los compromisos y 3) dar visibilidad a los impedimentos, son logrados mediante las reuniones diarias de Scrum (Daily Scrums). Estas reuniones tienen, como su nombre lo indica, una frecuencia diaria y no deberían llevar más de 10 minutos. Estos 10 minutos son un timebox, es decir, que no se pueden superar.

A la reunión diaria acude el ScrumMaster y el equipo de trabajo. En el caso de que sea necesario, se podrá requerir la presencia del Product Owner y de los stakeholders. De todas maneras, se intenta que sea una reunión abierta donde cualquier interesado en escuchar lo que sucede pueda participar en calidad de observador. Se recomienda que los observadores no participen activamente en la reunión, y mucho menos, que soliciten a los miembros del equipo justificación del progreso y explicación de los problemas.

Esta reunión es facilitada por el Scrum Master. Todos y cada uno de los miembros toman turnos para responder las siguientes tres preguntas, y de esa manera comunicarse entre ellos:

1. ¿Qué hice desde la última reunión diaria hasta ahora?

2. ¿En qué voy a estar trabajando desde ahora hasta la próxima reunión diaria?

3. ¿Qué problemas o impedimentos tengo?

Es importante destacar que en ningún momento se trata de una reunión de reporte de avance o status al Scrum Master ni a otras personas. Por el contrario, es un espacio de estricta comunicación entre los miembros del equipo de desarrollo. El objetivo de la primera pregunta (¿qué hice?) es verificar el cumplimiento de los compromisos contraídos por los miembros del equipo en función del cumplimiento del objetivo del Sprint. La finalidad de la segunda pregunta (¿qué voy a hacer?) es generar nuevos compromisos hacia el futuro. Cuando hablamos de compromisos, hacemos referencia a aquéllos que los miembros del equipo asumen ante sus compañeros (Oficina Asesora de Sistemas, 2016, p.14).

La última pregunta (¿qué problemas?) apunta a detectar y dar visibilidad a los impedimentos. Estos impedimentos no se resuelven en esta reunión, sino en posteriores. Es responsabilidad del ScrumMaster que se resuelvan lo antes posible, generando las reuniones que sean necesarias e involucrando a las personas correctas.

En el caso de que los Product Backlog Items (PBIs) del Sprint se hubiesen podido dividir en actividades de menos de un día: si una de estas actividades se encuentra en progreso durante dos reuniones diarias seguidas (con 24hs de separación) claramente se advierte un retraso.

5.2.6. Revisión de Sprint:

Al finalizar cada Sprint se realiza una reunión de revisión del Sprint (Sprint Review), donde se evalúa el incremento funcional potencialmente entregable construido por el equipo de desarrollo (el “qué”). En esta reunión el Equipo Scrum y los Stakeholders revisan el resultado del Sprint. Cuando decimos “resultado” hablamos de “producto utilizable” y “potencialmente entregable” que el los interesados utilizan y evalúan durante esta misma reunión, aceptando o rechazando así las funcionalidades construidas.

Los Stakeholders evalúan el producto construido y proveen feedback. Este feedback puede ser acerca de cambios en la funcionalidad construida o bien nuevas funcionalidades que surjan al ver el producto en acción.

Toda la retroalimentación que los stakeholders aporten debe ser ingresada como PBIs en el Product Backlog. Para esto, los PBIs nuevos deben ser priorizados con respecto a todos los ya existentes en el Product Backlog. También es necesario que estos nuevos PBIs sean estimados antes de incluirlos como parte del Product Backlog ya que el Product Owner deberá decidir cuáles de los PBIs existentes cuya estimación de costo es similar a la de los nuevos PBIs deben ser eliminados para no incurrir en el incremento desmedido del alcance (Scope Creeping): si se agrega trabajo entonces debemos quitar trabajo de otro lado. El Product Owner cuenta para esto con la priorización de los ítems del Backlog como herramienta para la toma de este tipo de decisiones (Oficina Asesora de Sistemas, 2016, p.14).

En el caso de que una funcionalidad sea rechazada, el PBI correspondiente reingresa al Product Backlog con máxima prioridad, para ser tratado en el siguiente Sprint. La única excepción a esta regla es que el Product Owner, por decisión propia, prefiera dar mayor prioridad a otros. En este caso, nada debe salir del Backlog ya que esto no sería considerado como un incremento en el alcance.

Al finalizar la revisión del producto, es recomendable definir la fecha de la próxima reunión de revisión, que corresponderá al final del Sprint siguiente. De este modo ya se tendrán las agendas bloqueadas a tal fin.

5.2.7. Feedback:

En una metodología como Scrum, la retrospección del equipo es el corazón de la mejora continua y las prácticas emergentes. Mediante el mecanismo de retrospección, el equipo reflexiona sobre la forma en la que realizó su trabajo y los acontecimientos que sucedieron en el Sprint que acaba de concluir para mejorar sus prácticas. Todo esto sucede durante la reunión de feedback (Oficina Asesora de Sistemas, 2016, p.15).

Esta reunión tiene lugar inmediatamente después de la reunión de revisión. Mientras que la reunión de revisión se destina a revisar el producto (el “qué”), la feedback se centra en el
Proceso (el “cómo”). Este tipo de actividad necesita un ambiente seguro donde el Equipo Scrum pueda expresarse libremente, sin censura ni temores, por lo cual se restringe solo al Equipo de Desarrollo, al ScrumMaster y el Product Owner. En el caso de que se requiera la participación de stakeholders o gerentes, estos podrán ser convocados.

Valiéndose de técnicas de facilitación y análisis de causas raíces, se buscan tanto fortalezas como oportunidades de mejora. Luego, el Equipo Scrum decide por consenso cuáles serán las acciones de mejora a llevar a cabo en el siguiente Sprint. Estas acciones y sus impactos se revisarán en la próxima reunión de feedback.

5.2.8. Release:

El release se compone de dos procesos

- Envío de entregables: los Accepted Deliverables se les entregan a los Socios relevantes. Un acuerdo formal llamado Working Deliverables Agreement documenta la finalización con éxito del Sprint. (Oficina Asesora de Sistemas, 2016, p.15).
- feedback del proyecto: En este proceso, que completa el proyecto, los socios y miembros principales del del Equipo Principal de Scrum se reúnen para hacer una feedback del proyecto e identificar, documentar e internalizar las lecciones aprendidas. A menudo, estas lecciones llevan a la documentación de Agreed Actionable Improvement, que se aplicará en futuros proyectos.

5.2.9. Identificación de Funcionalidades del Software

Para realizar la identificación de las funcionalidades se deberá tener en cuenta todos los roles identificados, efectuando sucesivas “pasadas” por todos los procesos de negocio y evaluando que cada uno de los roles involucrados en ellos cuenten con las funcionalidades requeridas para la realización de sus objetivos. Al igual que la identificación de roles, esta actividad se realiza en forma colaborativa junto al Product Owner y la mayor cantidad de miembros del equipo posible.

5.3. Historias de usuario

Las Historias de Usuario surgieron en eXtremme Programming (XP) como una respuesta a una situación habitual en los proyectos de desarrollo de software: los

clientes o especialistas de negocio se comunican con los equipos de desarrollo a través de extensos documentos conocidos como especificaciones funcionales. A su vez, las especificaciones funcionales son la documentación de supuestos y están sujetas a interpretaciones, lo que causa malos entendidos y que finalmente el software construido no se corresponda con la realidad esperada.

5.3.1. Componentes de historia de usuario:

Una Historia de Usuario se compone de 3 elementos, también conocidos como “las tres Cs” de las Historias de Usuario:

- Card (Ficha): Toda historia de usuario debe poder describirse de manera corta en una ficha. Si una Historia de Usuario no puede describirse en ese tamaño, es una señal de que estamos traspasando las fronteras y comunicando demasiada información que debería compartirse cara a cara.
- Conversación: Toda historia de usuario debe tener una conversación con el Product Owner. Una comunicación cara a cara que intercambia no solo información sino también pensamientos, opiniones y sentimientos.
- Confirmación: Toda historia de usuario debe estar lo suficientemente explicada para que el equipo de desarrollo sepa qué es lo que debe construir y qué es lo que el Product Owner espera. Esto se conoce también como Criterios de Aceptación.

5.3.2. Redacción de historia de usuario:

La redacción para una historia de usuario debe cumplir los siguientes criterios:

- Primera Persona: La redacción en primera persona de la Historia de Usuario invita a quien la lee a ponerse en el lugar del usuario.

- Priorización: Tener esta estructura para redactar la Historia de Usuario ayuda al Product Owner a priorizar. Si el Product Backlog es un conjunto de ítems como “Permitir crear un evento tentativo”, “Confirmar un evento tentativo”, “Notificar al responsable de logística”, “Ver el estado de inscripciones”, etc. el Product Owner debe trabajar más para comprender cuál es la funcionalidad, quién se beneficia y cuál es el valor de la misma (Oficina Asesora de Sistemas, 2016, p.16).

- Propósito. Conocer el propósito de una funcionalidad permite al equipo de desarrollo plantear alternativas que cumplan con el mismo propósito en el

caso de que el costo de la funcionalidad solicitada sea alto o su construcción no sea viable.

5.3.3. Definición de terminado:

También conocido como Definition of Done, es el conjunto de características que una Historia de Usuario debe cumplir para que el equipo de desarrollo pueda determinar si ha terminado de trabajar en ella. Un típico criterio de “Terminado” podría ser:

- Todos los criterios de aceptación funcionan correctamente
- Todos los archivos fuentes están en el repositorio de código fuente y el build se ejecutó exitosamente.
- El Product Owner da su visto bueno de la funcionalidad construida antes de llegar a la Review Meeting.

5.4 Elementos, partes y estructura básica de la Nomina

En las empresas la nómina es usada como un sistema contable para mantener un registro del salario, cargo, descuentos y otros gastos que pueda generar cada uno de los empleados vinculados con la empresa. La nómina viene siendo el documento que se le que es entregado mensualmente a cada uno de los empleados de una determinada empresa en el cual aparecen en detalle su salario neto, junto con los diferentes descuentos que se le puedan aplicar al mismo, ya sean de ley o por algún tercero (Oficina Asesora de Sistemas, 2013, p.4).

El formato estándar que debe tener una nómina es determinado por la normatividad vigente, el cual define una estructura y un contenido mínimo que debe ser respetado en cualquier caso. Este debe incluir al menos las siguientes características:

- Datos identificativos de la empresa, dirección del centro de trabajo y código cuenta cotización en el que está el trabajador incluido.

- Datos básicos del trabajador, tipo de contrato, categoría, antigüedad en la empresa.

- Periodo de liquidación al que corresponde dicha nómina.

- Detalle de las percepciones salariales y extra salariales que componen la retribución bruta del trabajador.

- Detalle de las deducciones que se le practican al salario bruto, bien marcadas por la legislación vigente, bien por otro tipo de deducciones que haya que aplicarle a la nómina, como anticipos o, embargos en la nómina.

- Líquido a percibir, dado que la nómina tiene consideración de documento acreditativo del pago de salarios cerrando los pagos pendientes al trabajador para el periodo estipulado.

- Detalle de las bases de cotización de la nómina

- Lugar de emisión y firma y sello por la empresa y trabajador.

Cabe resaltar que en determinados casos los anteriores ítems no son obligatorios, por ejemplo, la firma del trabajador no es necesaria cuando el pago de su salario se realiza con el depósito en un banco (Oficina Asesora de Sistemas, 2013, p.4).

5.4.1 Salario Bruto

El salario bruto está compuesto por todos aquellos conceptos que deben ser abonados por parte del empleador al empleado de forma periódica, pero esta periodicidad en el pago no siempre se cumple, puede que en un determinado momento un empleado ya no forme parte de la nómina de la empresa y se le deba pagar por el periodo trabajado así este no haya cumplido con los periodos establecidos de pago de la empresa (lo común es un mes).

Otro concepto a definir para entender el salario bruto es el Salario Base, que corresponde al pago mensual mínimo que se tiene que realizar para un trabajador dentro de la categoría en la que está encuadrado. Este salario base esta también acompañado de Complementos, que son cantidades adicionales que complementan al salario base en el caso de que se cumplan unos determinados requisitos de productividad, cumplimiento horario, productividad, trabajos penosos, nocturnos en días festivos, aunque no en todos los casos son aplicados.

Para los Docentes de vinculación especial pertenecientes a las nóminas de horas catedra honorarios y salarios según el decreto 1279 de 2002, en su artículo 4º *"Los profesores de hora cátedra de las universidades Estatales u oficiales distintas a la universidad Nacional de Colombia no son empleados públicos docentes de*

régimen especial ni pertenecen a la carrera profesoral y, por consiguiente, sus condiciones salariales y prestacionales no están regidas por el presente Decreto, sino por las reglas contractuales que en cada caso se convengan, conforme a las normas internas de cada universidad, con sujeción a lo dispuesto en las disposiciones constitucionales y legales vigentes...”

Con base en lo anterior la Universidad Distrital Francisco José de Caldas emite el Acuerdo 11 de 2002 en su artículo 3º donde establece lo siguiente:

“Son docentes de vinculación especial aquellos que, sin pertenecer a la carrera docente, están vinculados temporalmente a la Universidad.

Los docentes de vinculación especial son:

a) Ocasionales: Tiempo completo y medio tiempo.
b) De Hora cátedra
c) Visitantes
d) Expertos”

El cálculo del salario para los docentes de vinculación especial en la Universidad Distrital están fijados según el Acuerdo 12 de 2002, en su artículo 2º donde el valor del punto para el cálculo del valor de la hora de los docentes de vinculación especial está fijado por el gobierno nacional para cada año (Oficina Asesora de Sistemas, 2013, p.4).

5.4.2 Descuentos en la nomina

Tal y como hemos explicado anteriormente, en la nómina nos podemos encontrar dos tipos de descuentos diferentes, bien los descuentos obligatorios por ley o bien los descuentos que se deban aplicar por cualquier otro tipo de normativas.
En el caso de los descuentos por ley, tenemos dos grupos de deducciones diferentes, que se destinan al pago de la seguridad social a cargo del trabajador, como los pagos a cuenta que el propio trabajador tiene que realizar por impuestos u otros conceptos (Oficina Asesora de Sistemas, 2013, p.6).

CAPÍTULO 6. ALCANCES Y LIMITACIONES

6.1. Alcances

- El proyecto lleva a cabo las fases de inicio, elaboración y construcción de la metodología ágil de desarrollo SCRUM/OAS.

- Se realizarán pruebas funcionales y no funcionales de software con el fin de cumplir con las especificaciones establecidas por el grupo analista y el grupo de arquitectura. Dichas pruebas se realizan finalizando cada iteración (Sprint), ya que la finalización de estas culmina con un producto funcional.

- El proyecto estará estructurado en 24 iteraciones, la primera iteración consiste en detallar el modelo de negocio junto con la documentación de análisis y diseño construida anteriormente; las iteraciones restantes serán para la elaboración, construcción y pruebas del producto de software.

- Documentar y llevar a cabo los respectivos cambios en los componentes del software que cambien las especificaciones, de acuerdo con las características de la metodología ágil de desarrollo SCRUM/OAS.

6.2. Limitaciones

- La solución del software será desarrollada con base en el framework “Beego” mediante el lenguaje de programación “Go” definido e implementado en la Oficina Asesora de Sistemas.

- La solución de software estará basada en componentes de software libre dando respuesta a políticas distritales e institucionales.

- Realizar y documentar el número de componentes de software de acuerdo a la arquitectura y diseño pertinentemente comunicada y establecida en el proyecto frente a la gestión de nóminas en la Universidad Distrital y siguiendo los lineamientos del proceso de desarrollo SCRUM/OAS.

CAPÍTULO 7. METODOLOGÍA

La metodología implementada para el proyecto TITAN propuesta por la Oficina Asesora de Sistemas (OAS) de la universidad Distrital Francisco José de Caldas es la metodología de desarrollo ágil SCRUM.

A continuación se especifica el conjunto de actividades a desarrollar por el rol de desarrollador a través de este proyecto, ya que mediante este rol se desarrolló el proceso de pasantías del autor del presente anteproyecto:

Sprint's semanales

Desde el comienzo del proyecto, se llevan a cabo reuniones periódicas cada semana las cuales son llamadas Sprint semanal y son un importante elemento en la metodología SCRUM.
En cada sprint son asignadas tareas a cada desarrollador a partir de las prioridades de cada requerimiento, esta prioridad la define el Product Owner quien está presente en cada reunión y también es quien lidera al grupo de desarrollo. Es importante aclarar que este Product Owner durante el desarrollo del proyecto fue interno, es decir era una persona de la oficina designada como líder del proyecto que a su vez tenía la función de Product Owner.
En cada una de estas reuniones también se socializa los avances que se obtuvieron con la tarea asignada en el Sprint pasado y esta información es recopilada y registrada en Tuleap, una herramienta de seguimiento para las tareas de los proyectos desarrollados en la OAS. En estas reuniones también está presente el SCRUM Master quien se encarga de revisar que la metodología esté funcionando y se encarga de resolver los inconvenientes que impidan el desarrollo del proyecto.

Generación de Epic's

Para poder asignar tareas, antes se deben crear las Epic's que contendrán las historias de usuario, dichos Epic's contienen las historias relacionadas con una actividad particular de la cual no conocemos exactamente su duración, pero que se descompone en historias de usuario que son pequeñas partes del desarrollo de las cuales si conocemos estos aspectos. En las figuras 1 y 2 se toman como ejemplo las Epic's Nomina Docentes Vinculación Especial y Arquitectura TITAN definidas para el proyecto de pasantías del cual el presente documento hace parte.



Figura 1. Epic Nomina Docentes Vinculación Especial. Fuente: Realizado por el Autor(es)



Figura 2. Epic Arquitectura TITAN. Fuente: Realizado por el Autor(es)

Generación de las historias de usuario

Durante cada Sprint pueden asignarse nuevas historias de usuario a un Epic especifico, estas historias de usuario como se vio anteriormente son la representación de lo que el equipo de desarrollo o desarrollador debe construir, cada historia de usuario está dividida en tareas asignadas a los desarrolladores cada semana. En la figura 3 se ve un ejemplo de una historia de usuario asignada durante la pasantía.



Figura 3. Ejemplo Historia de Usuario. Fuente: Realizado por el Autor(es)

Para cada tarea se debían tener comentarios diarios para que asi el Scrum Master pudiera tener control sobre las actividades desarrolladas cada semana, un ejemplo del registro diario de dichas actividades se observa en la figura 4 donde se registran las correspondientes actividades de la tarea ***"Cargar reglas en nuevo API".***

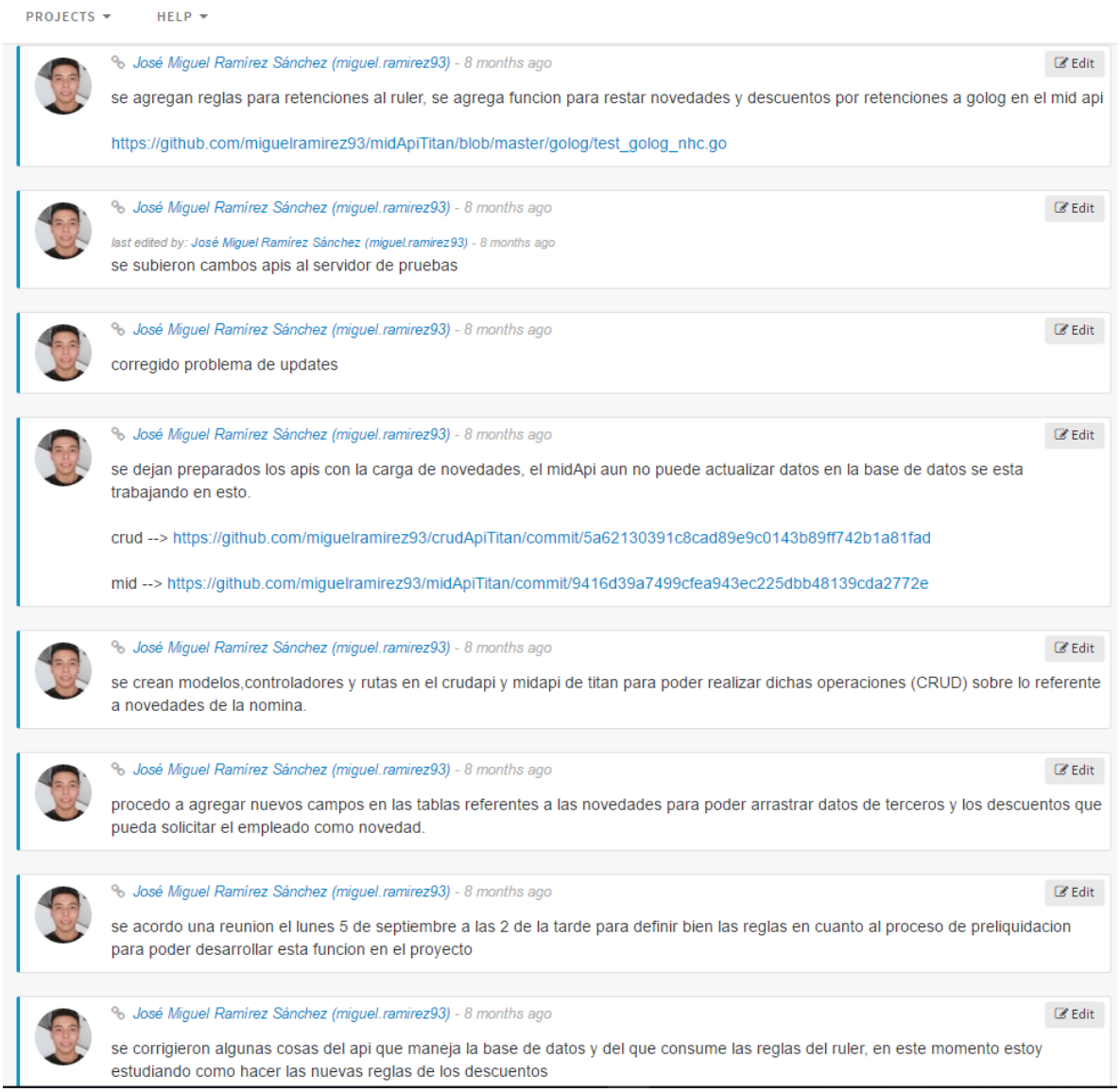


Figura 4. Ejemplo Registro diario de Actividades. Fuente: Realizado por el Autor(es)

Generación Kanban Task

En caso de realizar actividades referentes al proyecto que no estén registradas como tareas, se genera un Kanban Task que es un elemento que permite registrar estas actividades para que sirva de soporte para la validación de la realización de dicha tarea, estos Kanban task pueden servir como soporte para:

- Registrar reuniones con integrantes del mismo u otro proyecto para tratar temas relacionados a las tareas asignadas.
- Registrar solicitudes de herramientas, scripts o cambios en estructuras tanto del proyecto al que pertenece el desarrollador como de proyectos externos.
- Registro de reuniones con los Stakeholders para definir parámetros referentes a las actividades relacionadas con las tareas asignadas o funcionalidades del módulo en el que se esté trabajando.

Socialización mensual de avances en el proyecto

Un día de cada mes se realiza una reunión, a la que acuden los integrantes de los equipos de trabajo de cada proyecto que se está realizando en la OAS, durante estas reuniones se socializan los avances, contratiempos y soluciones que se han generado durante el mes para cada tarea asignada. Esto sirve de retroalimentación para los demás equipos de trabajo pues puede que un problema se presente en más de uno de los proyectos y alguno de los equipos de trabajo ya lo haya solucionado.

CAPÍTULO 8. RECURSOS

Se contempla en la Tabla 1, los recursos que se estiman serán necesarios en el proceso de desarrollo a la solución de software mediante el rol desarrollador.

Tipo de Recurso	Recurso	Cantidad
Humano	Desarrollador	5
Infraestructura	Computador	1
Infraestructura	Control de versiones, Entorno Beego y avance del proyecto en Tuleap.	1
Infraestructura	Almacenamiento disponible en google drive y Dropbox para guardar documentación	2 Gigas
Tiempo	Meses	4

Tabla 1 Recursos del Proyecto. Fuente: Oficina Asesora de Sistemas

CAPÍTULO 9. PRESUPUESTO

El presupuesto que se contempla desde el punto de vista del grupo desarrollador se evidencia en la tabla 2, expresada de la siguiente manera:

Tipo de Recurso	**Recurso**	**Cantidad**	**Tiempo**	**Valor mensual unitario**	**Valor cuatrimestral unitario**	**Valor total**
Humano	Desarrollador	1	4 meses	2.000.000	8.000.000	8.000.000
Humano	DBA	1	4 meses	3.000.000	12.000.000	12.000.000
Humano	Scrum Master	1	4 meses	2.895.000	11.580.000	11.580.000
Humano	Líder de proyecto	1	4 meses	3.100.000	12.400.000	12.400.000
Humano	Líder de desarrollo	1	4 meses	2.895.000	11.580.000	11.580.000
Infraestructura	Servicio de internet	1	4 meses	100.000	400.000	400.000
Infraestructura	Alquiler de computadores	1	4 meses	100.000	400.000	400.000
Infraestructura	Servicio de luz	1	4 meses	40.000	160.000	160.000
Transporte	Transporte	1	4 meses	80.000	320.000	320.000
Varios	Imprevistos.	1	4 meses	100.000	600.000	X6.820.800
Presupuesto Total:						**63.660.800**

Tabla 2 Presupuesto del proyecto, rol desarrollador. Fuente: Oficina Asesora de Sistemas

CAPÍTULO 10. CRONOGRAMA

El proyecto está planeado para que se desarrolle en 24 Sprint's, cada Sprint dura 1 semana, la duración del proyecto será aproximadamente de seis meses. En la siguiente imagen se muestra el cronograma de todo el proyecto en general (gestión de requerimientos, arquitectura y desarrollo). La fecha de inicio del proyecto está sujeta a la aprobación de los anteproyectos y la firma del contrato de pasantías, los 6 meses que se plantearon como periodo desarrollo empiezan en el momento que el proyecto curricular del visto bueno a lo descrito a lo largo de este documento**.**

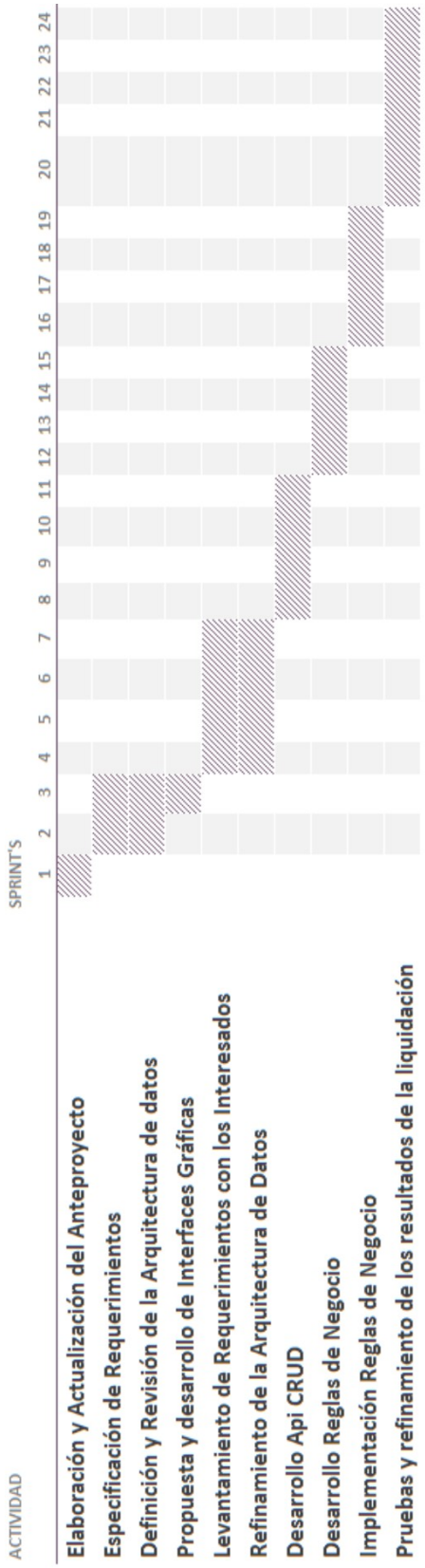


Figura. 5 Cronograma de Actividades. Fuente: Realizado por Autor(es)

CAPÍTULO 11. MODELO DE NEGOCIO

Para comprender el desarrollo que se llevó a cabo durante el periodo en que se realizó el proyecto de Gestión Integral de Nomina, más exactamente enfocado a la liquidación de docentes de vinculación especial (Horas catedra y Honorarios) dentro de la Universidad Distrital Francisco José de Caldas, en el cual el autor del presente proyecto de grado estuvo involucrado con el rol de desarrollador, se deben comprender los procesos que se llevan a cabo para realizar dicha liquidación, estos abarcan desde el registro en el banco de proveedores de los docentes , pasando por el registro de la información de los contratos correspondientes de los mismos en el sub-sistema de contratación, hasta el consumo de dicha información para realizar los cálculos dentro del proceso de liquidación a cada individuo. A continuación se describirán cada uno de estos procesos, los cuales fueron planteados por todo el equipo de trabajo del proyecto TITAN (Interesados, Arquitecto de Software y Desarrolladores).

11.1 Registro de Persona en el Banco de Proveedores

Para poder realizar el proceso de liquidación de las respectivas nóminas, primero se deben registrar los datos personales de cada persona en el banco de proveedores, es importante aclarar que el objetivo del presente proyecto de grado no es el de realizar este proceso de registro de personas o de los contratos pero estos sub-procesos hacen parte del modelo de negocio, ahora bien, el proceso comienza registrando los datos personales de cada persona en el banco de proveedores, estos datos serán usados posteriormente en el proceso de liquidación como identificadores de cada persona, de esta manera es más fácil la identificación de inconsistencias por parte del usuario final que va a realizar la liquidación de la(s) nóminas para un periodo específico. Estos datos no serán directamente consumidos por TITAN, estos serán referenciados en la información de cada contrato en el sub-sistema de contratación del cual se hará una breve explicación más adelante.

11.2 Registro del Contrato de Docente de Vinculación especial en el sub-sistema de contratación.

Una vez se tiene el registro de los datos personales de la persona en el banco de proveedores se procede a registrar los datos del contrato, en este caso los contratos son registrados con el tipo “Docente de vinculación especial -DVE” y también con los sub-tipos “Salarios” y “Honorarios”, según el Acuerdo 11 de 2002 en su artículo 3º se establece que “Son docentes de vinculación especial aquellos

que, sin pertenecer a la carrera docente, están vinculados temporalmente a la Universidad" (REF) , estos tipos varían en la forma en que se liquida mes a mes los pagos base que se estipulan en el contrato con base a aportes parafiscales normativos en la Universidad, a cada contrato se le asigna un número único para cada vigencia (entiéndase vigencia como el año en curso donde se efectúa el contrato), otra de las características que tienen los contratos de los docentes de vinculación especial es que tienen una duración de 4,5 meses periodo para el cual se destina cierta cantidad de dinero al contrato de cada persona, es decir se reserva el presupuesto para pagar cada contrato durante dichos 4,5 meses, esta reserva varía entre ambos tipos de contrato para los docentes de vinculación especial de la siguiente manera :

- **Salarios:** Se destina además del pago correspondiente a los 4,5 meses un porcentaje extra (20,02 % durante la vigencia 2016) para el pago de conceptos como primas y cesantías.
- **Honorarios:** Solo se destina el pago total de los 4,5 meses sin tomar en cuenta pagos de conceptos como primas y cesantías.

A demás de lo anterior, el salario de una persona vinculada a la Universidad Distrital como Docente de Vinculación Especial viene dada también según el Artículo 2º del Acuerdo 12 de 2002 por el valor que se le da a cada punto que obtiene la persona en la fase pre-contractual, este valor según el artículo está dado por el gobierno nacional durante cada vigencia. Es importante recalcar que este proceso no hace parte del objetivo del presente documento, pero su entendimiento es importante para comprender el proceso de liquidación de las nóminas.

11.3 Registro de Nominas

Una vez se ha aclarada la forma en que se vinculan los docentes de vinculación especial, se procederá a generar el registro de cada una de las nóminas, para este caso en específico las de profesores de vinculación especial en la modalidad de horas catedra, a continuación se muestra un diagrama de proceso para este caso:

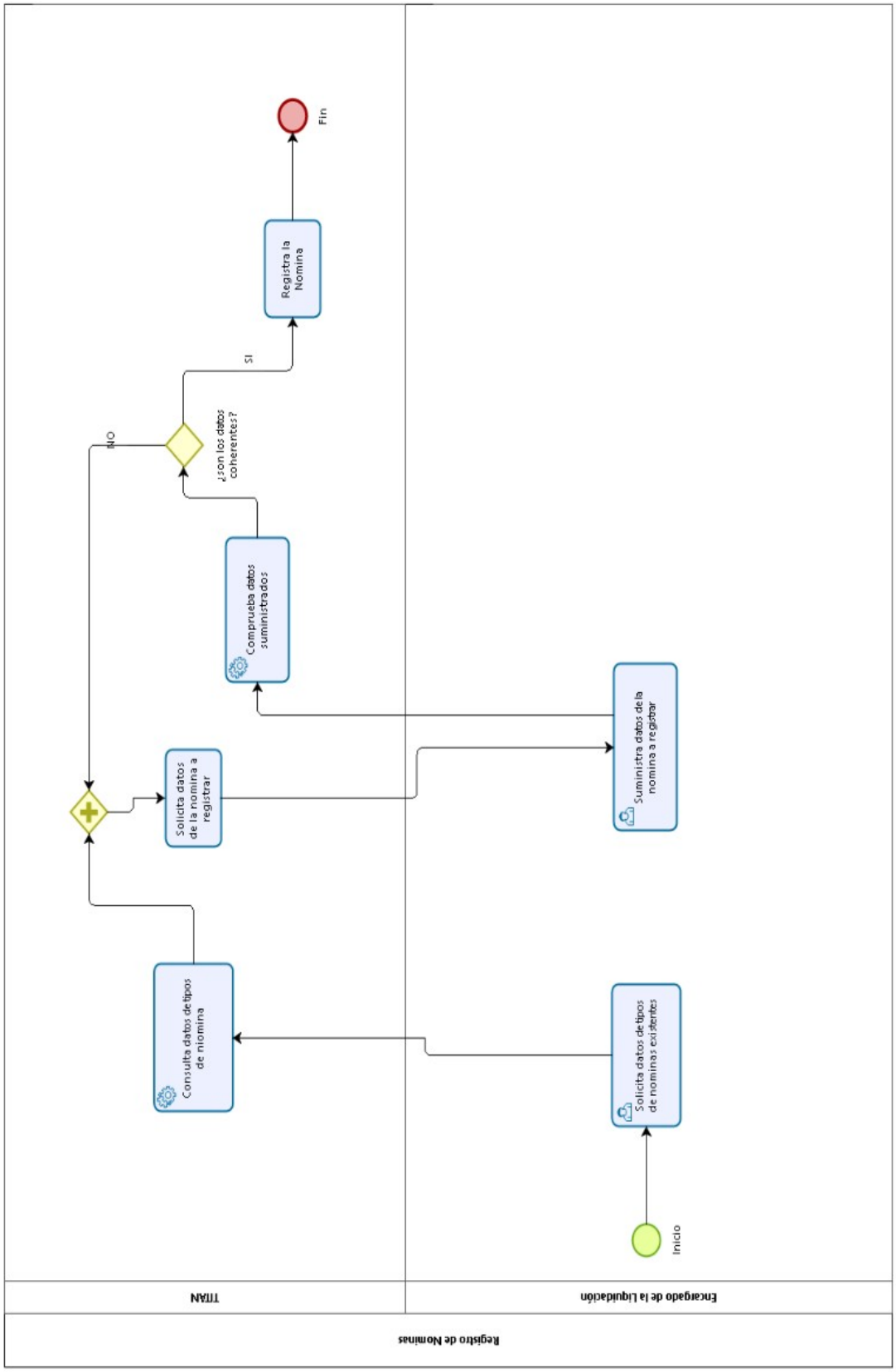


Figura. 6 Diagrama de Proceso: Registro de Nóminas. Fuente: Realizado por Autor(es)

En este proceso, quien realiza las liquidaciones de las nóminas debe registrar para cada vigencia la nómina o nominas que desea liquidar, este proceso inicia seleccionando el tipo de nómina al cual se le va a realizar las preliquidaciones y liquidaciones durante el periodo de la vigencia, estas se encuentran registradas en la base de datos en el esquema TITAN del cual se hará una explicación más adelante. Una vez se ha seleccionado el tipo se deben registrar los datos correspondientes a esta nómina, estos son:

- Nombre de la nómina.
- Descripción
- Tipo de vinculación

Una vez hecho esto al registrar el sistema asignara la vigencia automáticamente a dicha nómina.

11.4 Registro y Calculo de Preliquidaciones.

Una vez se ha registrado la nómina para una correspondiente vigencia, se procede a registrar las preliquidaciones que harán parte de esta nomina durante la vigencia de la misma, estas preliquidaciones, este proceso se evidencia en el diagrama de la Figura 7 a continuación:

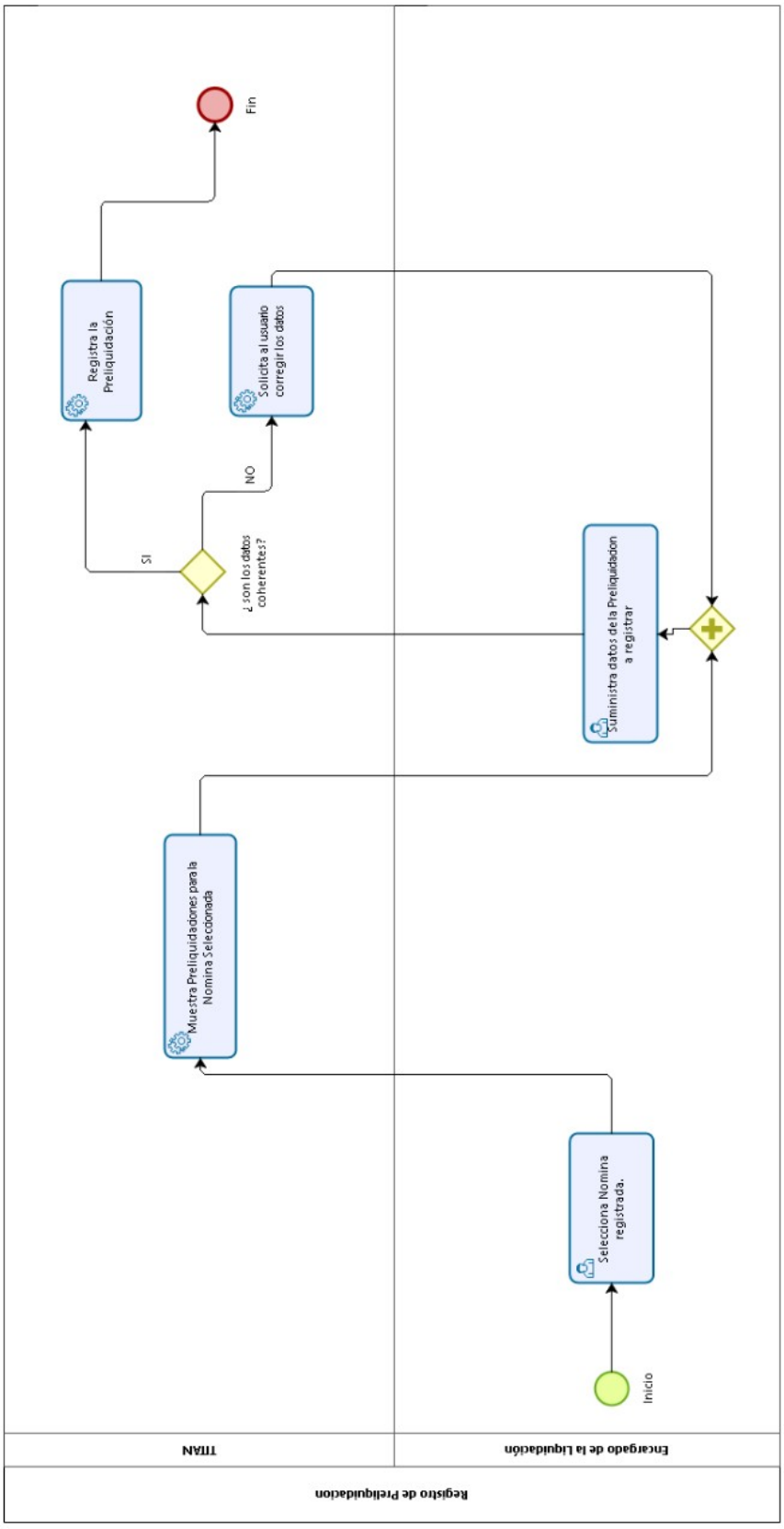


Figura. 7 Registro de preliquidación. Fuente: Realizado por Autor(es)

Este proceso comienza cuando el responsable de la liquidación selecciona la nómina para la cual desea registrar una preliquidación, una vez se ha seleccionado dicha nomina el sistema le muestra las preliquidaciones registradas para esta y la opción de registrar una nueva, los datos que el responsable de la liquidación debe suministrar al sistema para registrar una preliquidación son:

- Nombre para la Preliquidación.
- Descripción.
- Periodo a Preliquidar.

Una vez se ha registrado la preliquidación, el responsable de la liquidación ya tendrá disponible la opción de generar los cálculos para cada persona con base en las reglas establecidas para este proceso, a continuación se evidencia el proceso del cálculo de la preliquidación en el diagrama de la figura 8:

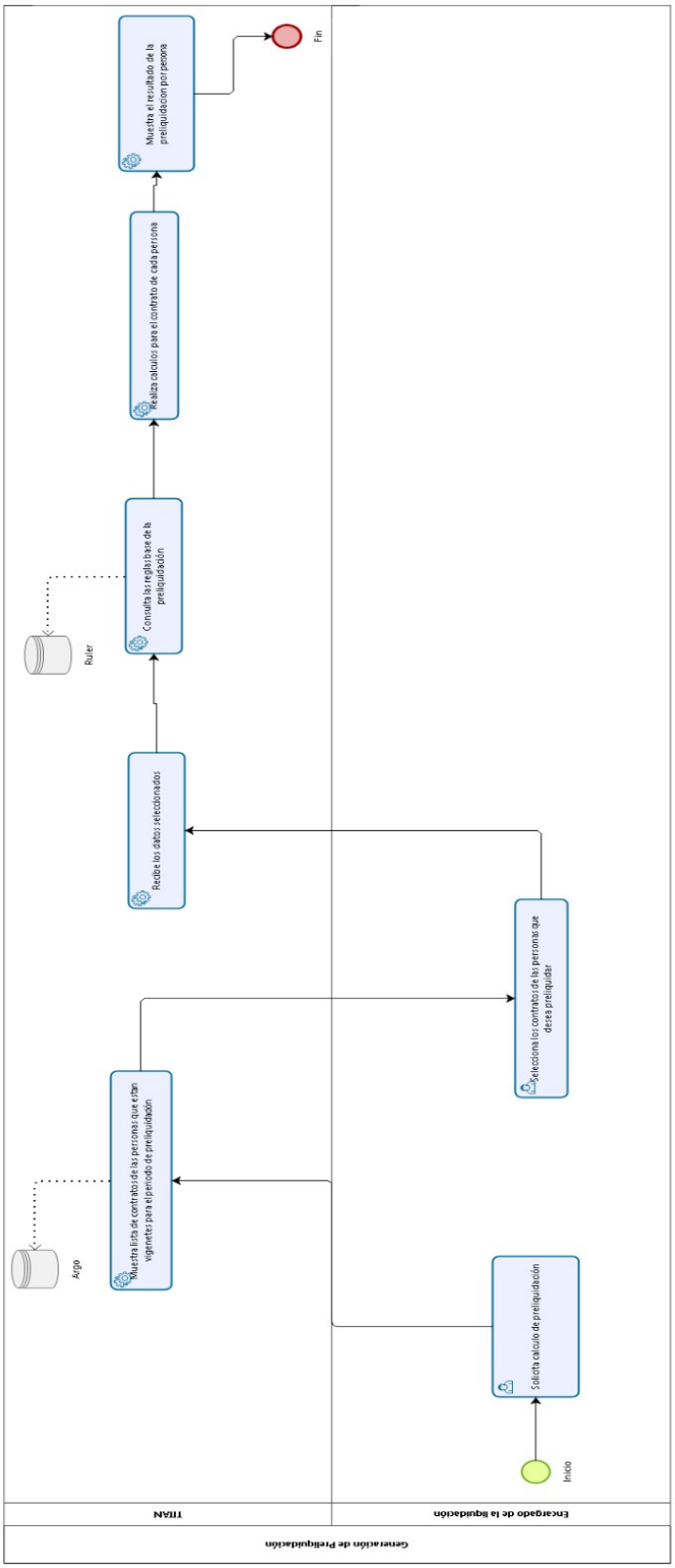


Figura. 8 Generación de preliquidación. Fuente: Realizado por Autor(es)

El proceso de preliquidación comienza cuando el encargado de la liquidación termina de registrar una preliquidación o selecciona una que previamente ha sido registrada, al seleccionar la opción de generar la preliquidación el sistema procederá a realizar las operaciones indicadas para realizar los respectivos cálculos para cada contrato correspondioente a cada persona que se encuentren en el banco de proveedores y en el sub-sistema de contratación, las condiciones para que un contrato DVE (Docente de vinculación especial) entre en los cálculos de la preliquidación que se esté realizando son:

- El docente debe estar registrado en el banco de proveedores.
- El contrato del docente debe estar registrado en el sub-sistema de contratación.
- Las fechas de inicio y finalización del contrato deben estar dentro del rango del rango de la preliquidación.
- Los parámetros de la vigencia actual deben estar registrados dentro de las reglas lógicas.

Las reglas que se usan para el proceso de preliquidación se presentan a continuación mediante dos diagramas de flujo, uno para Horas catedra Salarios y el otro para Honorarios:

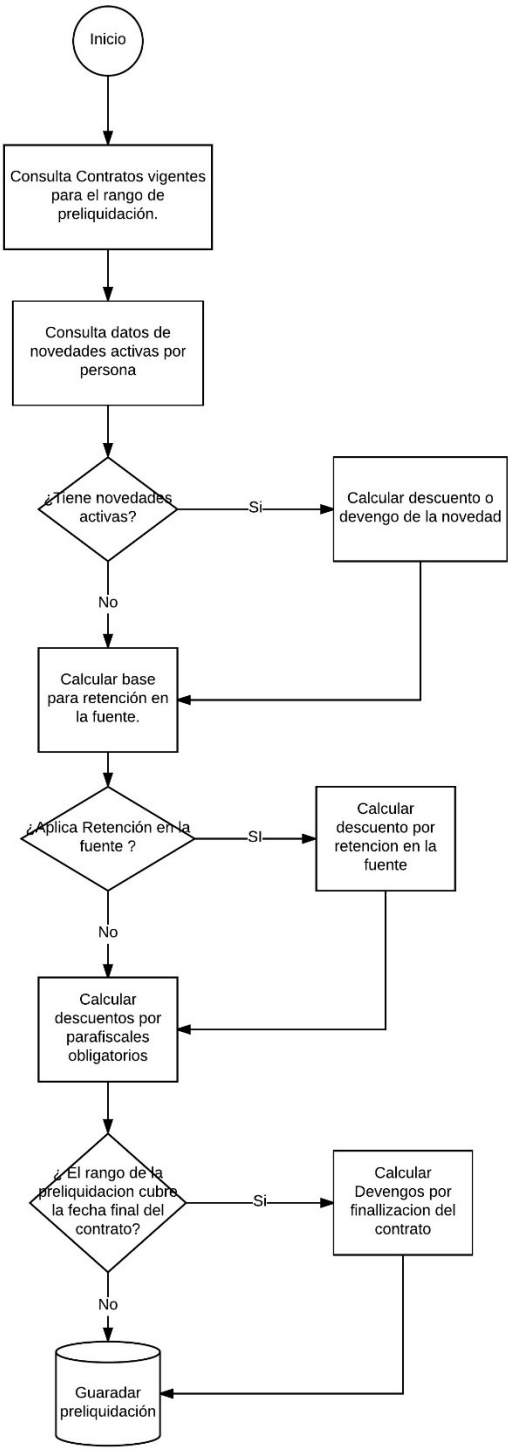


Figura. 9 Diagrama de flujo: calculo preliquidación HC Salarios. Fuente: Realizado por Autor(es)

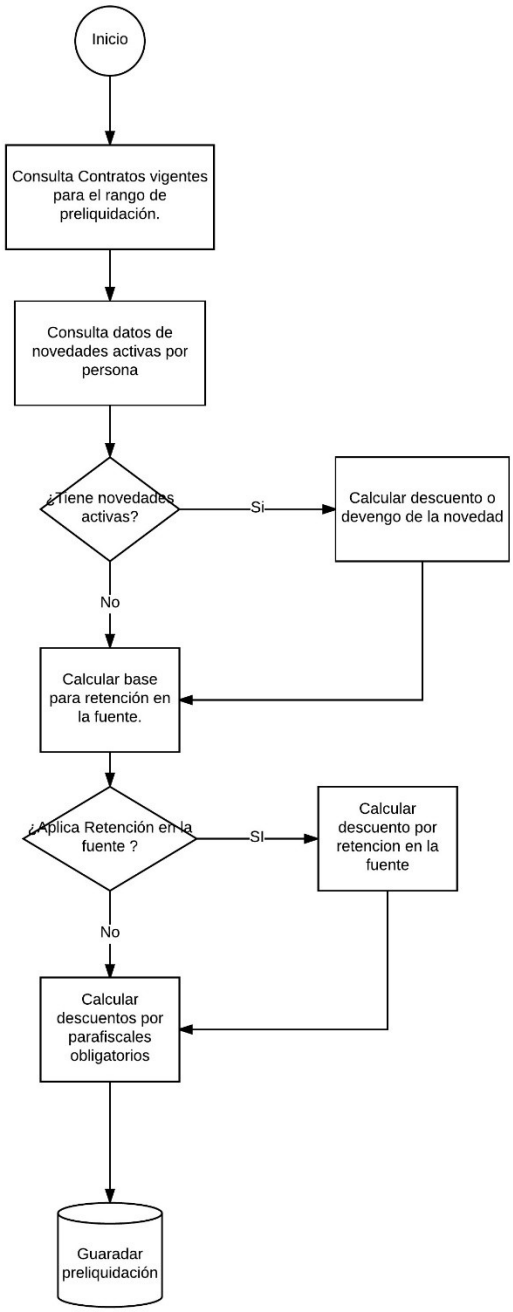


Figura. 10 Diagrama de flujo: calculo preliquidación HC Honorarios. Fuente: Realizado por Autor(es)

Los anteriores diagramas (Figura 9 y 10) representan el flujo del proceso mediante el cual se calcula el valor a pagar a cada persona que tiene un contrato como DVE en la Universidad Distrital, como se puede observar el proceso es similar en ambos casos (Salarios y Honorarios) con pequeños cambios para cada proceso, a continuación se explicara de forma general este cálculo para ambos casos:

- **Consulta Contratos Vigentes:** Se consulta a partir del tipo de nómina a preliquidar la información de todos los contratos que se encuentren vigentes durante el rango de la preliquidación, es decir que la fecha de inicio y final estén dentro de dicho rango, estos serán los contratos que se tendrán en cuenta para el cálculo a partir de este punto.

- **Consulta de Novedades Vigentes:** Una vez se tienen los datos de los contratos de las personas a preliquidar se procede a consultar para cada persona las novedades que pueda tener para el periodo de la preliquidación, este proceso se realiza de manera similar al de la consulta de los contratos, tomando en cuenta las fechas de inicio y final de la novedad, de esta manera si la novedad no se encuentra dentro del rango de la preliquidación esta no será tomada en cuenta si las fechas registradas de su rango de validez no están dentro del rango de la preliquidación.

- **Calculo del salario Base:** Un proceso intermedio para el cálculo de la preliquidación es el Cálculo del salario Base, este viene dado según (Oficina Asesora de Sistemas, 2016, p. 6) como:

  - $$valor\ mensual\ del\ contrato = \frac{Valor\ del\ contrato}{número\ de\ meses}$$

  Este cálculo se realiza mediante la información consultada en pasos anteriores del proceso relacionada con el contrato y se realiza este cálculo para cada contrato que entra en el rango de la preliquidación.

- **Calculo de base para retención en la fuente:** Este cálculo se realiza de dos maneras diferentes, pues para las nóminas de Docentes de Vinculación Especial se pueden aplicar dos tipos de retención: articulo 383 y articulo 384, la retención del artículo 383 se calcula tomando un porcentaje del pago base de la persona, para el periodo 2016 este porcentaje correspondió al 25%, una vez se hace este cálculo se le resta dicho porcentaje al valor del pago base y por último se redondea ese resultado, la formula se expresa de la siguiente manera:

- $Base\ rete\ 383 = Round\left(valor\ mensual\ del\ contrato - \left(valor\ mensual\ del\ contrato * 0.25\right)\right)$

El cálculo de la base para la retención del artículo 384 es un poco más sencilla en comparación con la del artículo 383 pues esta base es valor del salario base.

Ambas bases son usadas para calcular el descuento por retención en la fuente, proceso que será descrito en el siguiente punto.

- **Calculo de la retención en la fuente:** la retención en la fuente se calcula con base en el valor del UVT para un periodo en específico, esta es una unidad utilizada para representar valores tributarios, este valor es usado para encontrar un valor porcentual en una tabla determinada para cada tipo de retención dividiendo el valor del pago base sobre el valor de UVT del periodo así se encuentran a cuantos UVT equivale el pago, este valor se aplica en la formula a aplicar para el cálculo de este descuento. A continuación se muestra como ejemplo la tabla de valores de UVT para la retención 383:

Articulo 383 Tarifa				
inferior	**superior**	**tarifa**		**extra**
0	**95**	0%		
95	**150**	19%		
150	**360**	28%	**MAS**	**10 UVT**
360	**+**	33%	**MAS**	**69 UVT**

Tabla 3. Tarifas retención en la fuente. Fuente: Oficina Asesora de Sistemas

En la anterior tabla se muestra los límites superiores e inferiores y la tarifa correspondiente según el número de UVT correspondiente al pago base de cada persona en la preliquidación, una vez se tiene este valor se aplica una formula especial para cada tipo de retención para poder conocer el valor del descuento por este concepto de la siguiente manera:

- $Descuento\ 383 = \left(\left(\left(Valor\ salrio\ en\ UVT - inferior\right) * tarifa\right) + extra\right) * Valor\ UVT$

- $Descuento\,384 = (tarifa * valor\,UVT)$

- **Calculo por parafiscales obligatorios:** Estos son tratados como descuentos de ley, es decir, deben ser calculados y aplicados en todas las preliquidaciones que se hagan y según la normatividad vigente, para el periodo de 2016 se tuvieron en cuenta los siguientes descuentos de ley:

Nombre	**Formula**	**Nomina**
Rete ICA	$\frac{Valor\ pago\ base * 9.66}{1000}$	Salarios y Honorarios
Estampilla UD	$Valor\ pago\ base * 0.01$	Salarios y Honorarios
Pro-Cultura	$Valor\ pago\ base * 0.005$	Salarios y Honorarios
Adulto Mayor	$Valor\ pago\ base * 0.005$	Salarios y Honorarios

Tabla 4. Descuentos por parafiscales obligatorios. Fuente: Oficina Asesora de Sistemas

- **Calculo de devengos por final de contrato (Salarios Únicamente):** Este cálculo se realiza cuando el sistema detecta que el periodo de preliquidación cubre la fecha final del contrato de alguna de las personas que entran en la preliquidacion a realizar, estos cálculos se resumen en la siguiente tabla:

Nombre	**Formula**
Prima de Vacaciones	$Valor\,del\,contrato * 0.05555$
Prima de Navidad	$Valor\,del\,\frac{contrato * 1.0555}{12}$
Cesantías	$Valor\,del\,\frac{contrato * 1.1433}{12}$
Interés por Cesantías	$Valor\,del\,\frac{contrato * 0.137}{12}$

Tabla 5. Devengos por finalización del contrato. Fuente: Oficina Asesora de Sistemas

11.5 Registro de Novedades.

En el proceso de preliquidación es importante conocer que no solo al pago de cada persona le afectan los descuentos y devengos de ley, también pueden darse casos donde la persona solicita que se le descuente directamente algún tipo de pago por prestamos u otros factores externos, también puede darse el caso de realizar ajustes al pago por errores en pagos anteriores o incluso bonificaciones que se le dan a la persona que no necesariamente se le dan a los demás, para estos casos se registran dichos descuentos o devengos como novedades para ser aplicadas una o varias veces en distintas preliquidaciones, a continuación se muestra un diagrama que resume dicho proceso:

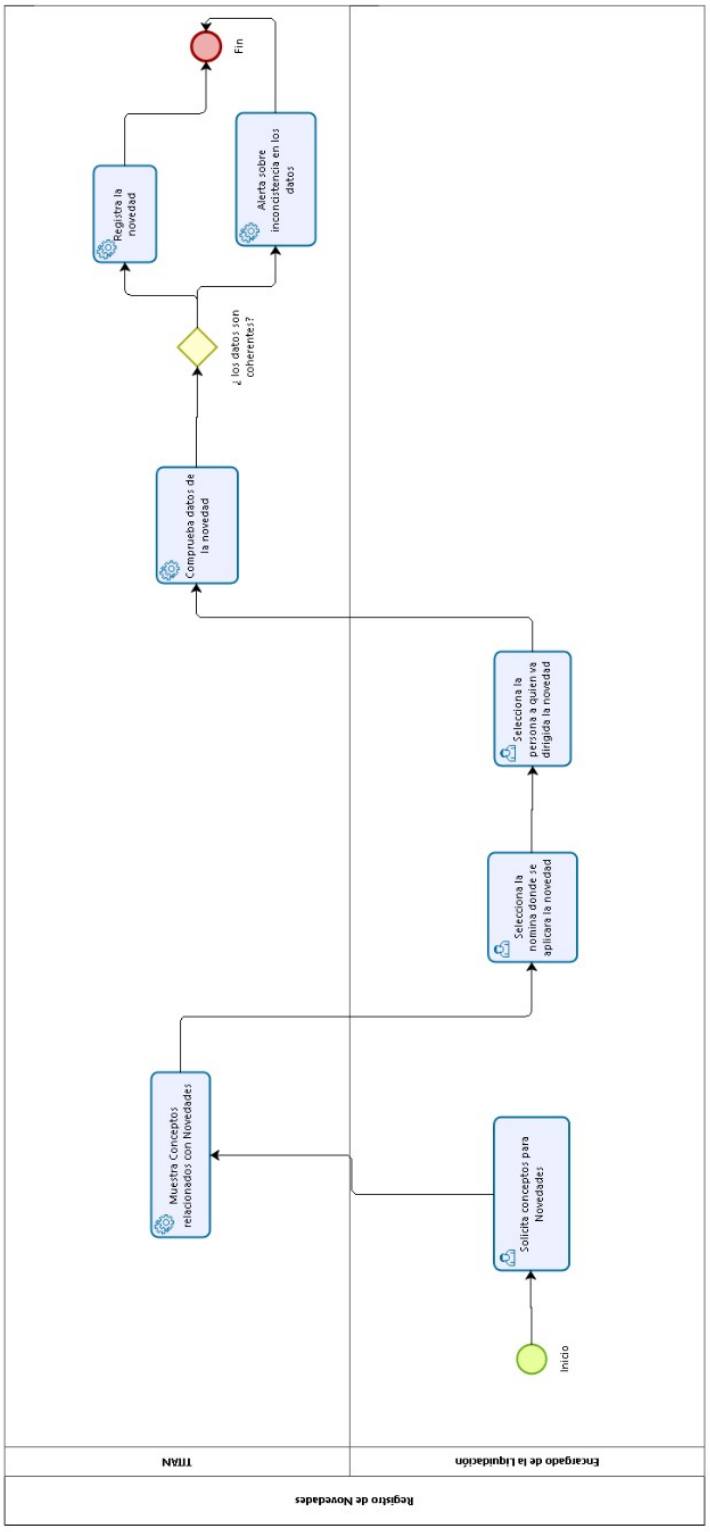


Figura. 11 Registro de una novedad. Fuente: Realizado por Autor(es)

Durante este proceso, el encargado de la liquidación debe primero asegurarse de que los datos del concepto que se pretende aplicar estén registrados, una vez se hace esta comprobación, este debe seleccionar la nómina a la cual va dirigida la novedad, esto se hace debido a que puede que una persona esté en varias nominas con contratos diferentes y se desea sabe a qué pago se debe descontar o sumar dicha novedad. Una vez se tienen estos datos selecciona a la persona a quien va dirigida la novedad y se le da un tipo (Descuento o Devengo), una naturaleza (Porcentaje o Fijo), el rango de tiempo donde se hará efectiva esta novedad y el valor a descontar. Una vez hecho esto el sistema estará en la capacidad de agregar esta novedad al cálculo en la preliquidación de la nómina indicada en el registro de la novedad.

11.6 Registro de Liquidación.

Este proceso se realiza una vez se tiene una preliquidación calculada, este es la parte final de la Liquidación de la nómina, para llegar a esta parte pueden haberse generado varias veces la misma preliquidación, como ya se han hecho los cálculos de los descuentos, devengos y novedades para cada persona en la preliquidación, el registro de la liquidación se convierte en una operación relativamente sencilla, este proceso se resume en el siguiente diagrama:

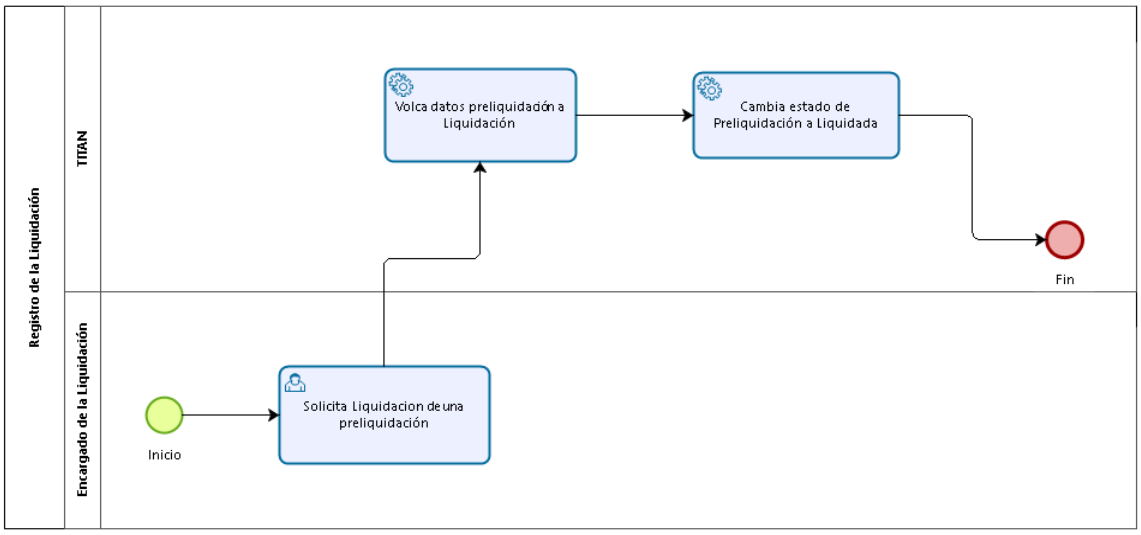


Figura. 12 Registro de la Liquidación. . Fuente: Realizado por Autor(es)

CAPÍTULO 12. MODELO DE DATOS

La idea fundamental de los proyectos que se realizaron en la OAS (Oficina Asesora de Sistemas) dentro de la universidad Distrital Francisco José de Caldas durante el periodo de realización de las pasantías presentadas por el autor del presente documento fue la centralización de la información y la eliminación de la redundancia en los datos presentes en cada uno de los sub-sistemas que hacen parte de este macro-proyecto, esta centralización de la información obliga al trabajo cooperativo entre integrantes de los equipos de trabajo de los diferentes sub-sistemas, como fue el caso del proyecto TITAN tema principal de este documento, donde a partir de la información presente en el sub-sistema de contratación (Argo) y el banco de proveedores (Agora) se puede sustraer la información relevante para los cálculos y diferenciación de los resultados de las liquidaciones y preliquidaciones, a continuación se muestra la estructura del esquema TITAN y se realiza una breve explicación de este mismo:

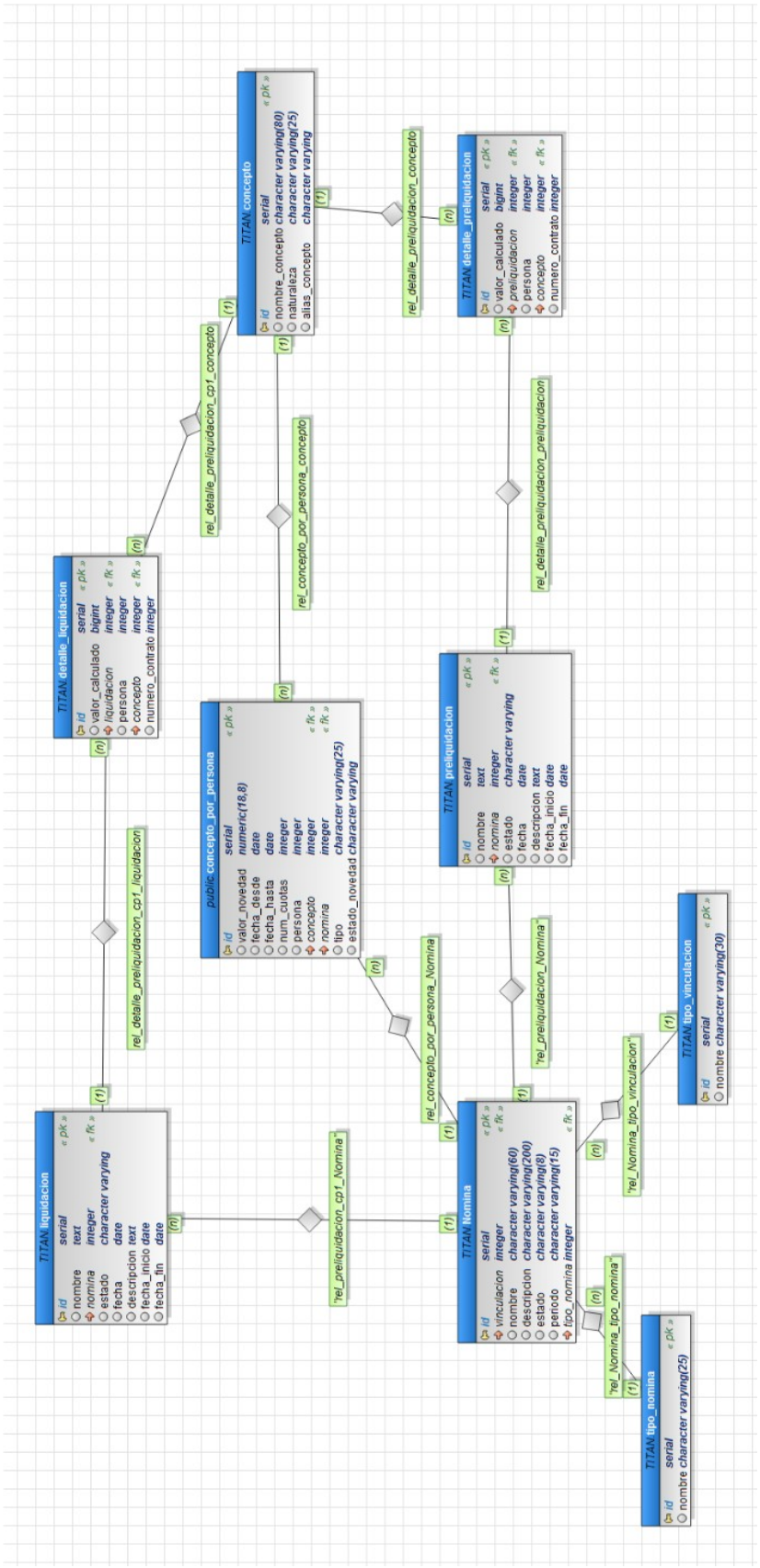


Figura. 13 Esquema Titán. Fuente: Realizado por Autor(es)

12.1 Descripción del esquema TITAN.

Este esquema es en donde se guardan los datos referentes a las preliquidaciones y liquidaciones, está compuesto por diferentes tablas relacionadas entre sí para brindar un mayor control en las operaciones que se realizan a nivel de negocio dentro del proceso de liquidación y pago de las diferentes nominas existentes dentro de la Universidad Distrital, a partir de este punto se describirán cada una de las tablas tomando en cuenta la función que cumple a nivel operativo.

12.1.1 Tabla nomina

Esta tabla representa las nóminas que pueden ser registradas por vigencia dentro del sistema, esta tabla tiene una relación 1 … n con la tabla preliquidación, en donde a una nómina está relacionada con 0 o más preliquidaciones. En esta se registran datos como la descripción, el tipo de vinculación y la vigencia de cada nómina.

12.1.2 Tabla Preliquidación

Esta representa los datos generales de las preliquidaciones que se registren para una nómina en específico, datos como Nombre, Descripción, la nómina a la cual pertenece y el periodo de liquidación del que hace parte, una parte importante de esta tabla es el periodo de liquidación, este está compuesto por dos fechas, la de inicio y la de finalización, este dato es el usado en los cálculos internos del sistema para saber cuántos días se le pagaran en esta preliquidación a las personas que hacen parte de la nómina que se está liquidando.

12.1.3 Tabla Detalle Preliquidación

Esta tabla, como su nombre lo indica, representa el detalle de cada preliquidación en donde se enlaza la información de persona natural mediante un identificador único de tipo serial, esto por motivos de integridad ya que la asignación de esta llave primaria a el número de documento de cada persona puede generar inconsistencias a la hora de realizar algún cambio en este, además de esto tiene una relación de cardinalidad 1 … 1 con concepto debido a que en este detalle se almacena el valor de cada uno de estos conceptos calculado durante el proceso de preliquidación , de esta manera se puede obtener un informe de cada concepto aplicado a cada persona durante la preliquidación.

12.1.4 Tabla Concepto

Esta tabla guarda todos los conceptos que son aplicables dentro de la preliquidación y liquidación, estos conceptos pueden ser de dos tipos (Descuento o Devengo) y esto está limitado por una restricción CHECK dentro de esta tabla.

12.1.5 Tabla Concepto por Persona

Esta tabla es donde se registran los datos de las novedades aplicadas a cada persona de forma individual, esta contiene datos como el valor, la naturaleza (fijo o porcentaje), el Concepto al cual se aplicara, la nómina donde será efectivo y el rango donde será válida esta novedad. Esta tabla es la usada durante la preliquidación para obtener las novedades que se encuentren activas durante el periodo de la preliquidación para cada persona y así tomarlas en cuente durante este proceso.

12.1.6 Tabla Liquidación

Esta tabla es donde se guardan los datos de una preliquidación que ha pasado la revisión y se ha decido que esta lista para ser liquidada, aquí se volcán los datos de la tabla preliquidación para hacer el cierre de esta.

12.1.7 Tabla Detalle Liquidación

En esta tabla, al igual que en la tabla Liquidación, es donde se guardan los datos de los detalles de preliquidación que han pasado la revisión y se ha decido que son los datos correctos para ser liquidados.

12.1.8 Tabla Tipo Nomina

Esta tabla almacena los diferentes tipos de nóminas que pueden haber dentro de la lógica del negocio, a parte la creación de esta tabla permite hacer al modelo flexible a la hora de la aparición de un nuevo tipo de nómina dentro del modelo de negocio.

12.1.8 Tabla Tipo Vinculación

Esta tabla almacena los tipos de vinculación que existen dentro del modelo de negocio de la Universidad, parte la creación de esta tabla permite hacer al modelo flexible a la hora de la aparición de un nuevo tipo de vinculación dentro del modelo de negocio.

CAPÍTULO 13. ARQUITECTURA DE LA APLICACIÓN

13.1 Arquitectura General

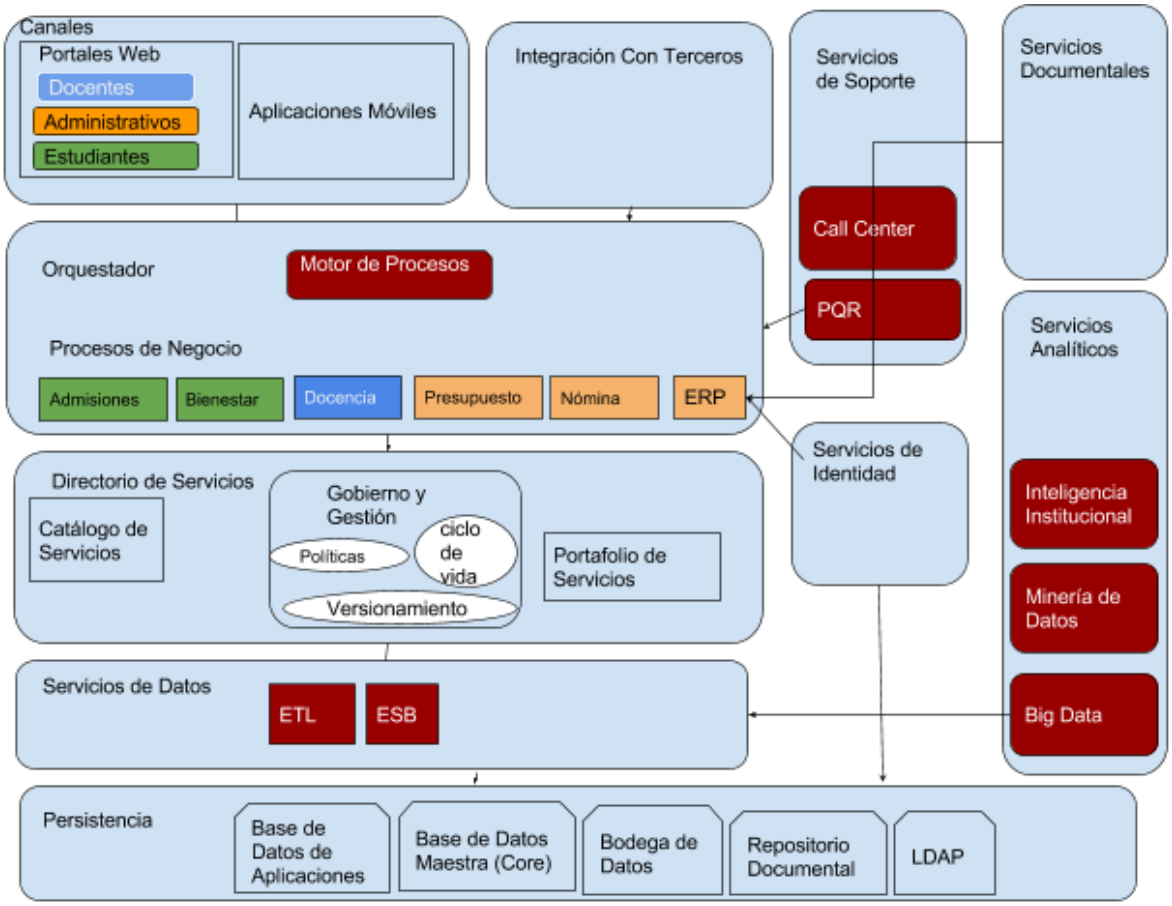


Figura 14. Diagrama de la Arquitectura general del sistema. Fuente: Oficina Asesora de Sistemas

En el anterior diagrama se resume la arquitectura propuesta para el macro sistema de centralización de la información de la cual hace parte el sistema integral de Nomina: TITAN, el cual es el objetivo de explicación del presente documento. En este diagrama se explica la interconectividad del sistema implementado una arquitectura orientada a servicios, de esta manera se reduce el tiempo de desarrollo y la complejidad en la interacción entre sistemas, permitiendo realizar una mejor integración entre sistemas a medida que pasan de etapa de desarrollo a producción, también presentan una mayor compatibilidad con servicios externos.

13.1.1 Sub-Sistemas

Se define en el Orquestador (ver figura 10) algunos de los subsistemas que integran la anteriormente mencionada centralización, dentro de estos se encuentra:

- **Admisiones:** Este sub-sistema se encarga de lo referente a la admisión de nuevos estudiantes tomando en cuenta la normatividad vigente dada tanto por la Universidad como por el Gobierno Nacional.
- **Docencia:** Es el que se encarga de lo referente a la contratación docente, ya sea de planta o de vinculación especial.
- **Presupuesto:** Este sub-sistema es el encargado de manejar todo lo referente a presupuesto destinado para el funcionamiento de la universidad.
- **Nomina:** Este es el sub-sistema que se encarga de realizar el proceso de liquidación de cada una de las diferentes nominas existentes en la universidad, este interactúa con los anteriormente descritos sub-sistemas para obtener la información necesaria para el proceso de liquidación.

Estos Sub-Sistemas son propuestos como solución a la reducción de la redundancia de datos presentes en los Sistemas informáticos que actualmente funcionan dentro de la Universidad a nivel académico y administrativo. Lo que se busca con la arquitectura propuesta en la Oficina Asesora de Sistemas (OAS) es facilitar la interoperabilidad de los sistemas informáticos tanto internos en la institución como servicios externos que pueda requerir la misma.

13.1.2 Arquitectura General De los Sub-Sistemas que intervienen en el proceso de liquidación de las nóminas.

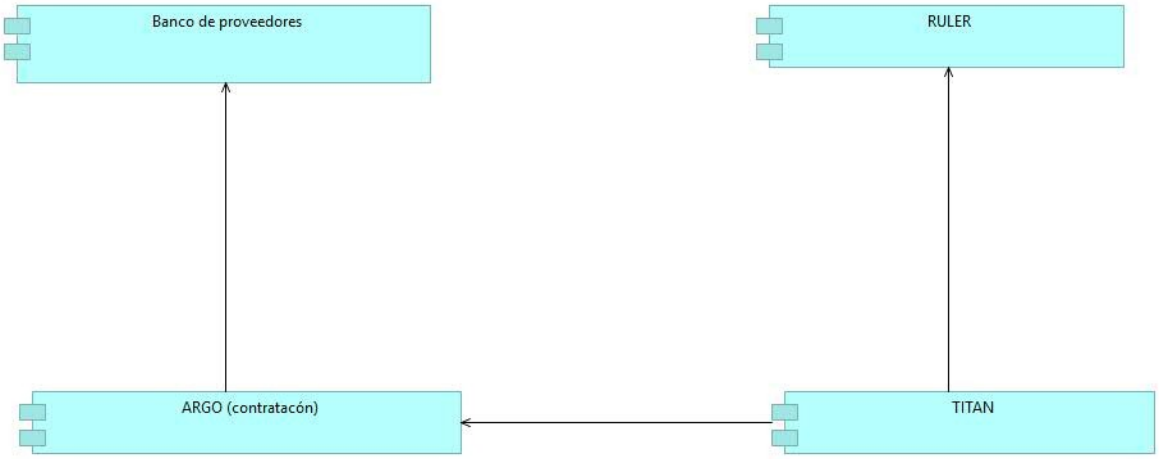


Figura 15. Diagrama de arquitectura de los sub-sistemas que interviene en la nómina. Fuente: Realizado por el Autor(es)

En el diagrama representado en la Figura 15 se puede ver la forma general en que cada sub-sistema involucrado interactúa en el proceso de generación y cálculo de la liquidación de las nóminas, Dicha interacción se da de la siguiente manera:

13.1.2.1 Argo (Contratación).

El módulo de ARGO, como se ha explicado a lo largo del presente documento, es el encargado de la gestión de los diferentes tipos de contrato que se le asocian a una persona (para este caso en particular proveedor) este almacena dicha información, información que es importante conocer a la hora de realizar los cálculos pertinentes en la liquidación mensual, por este motivo y debido a la separación de responsabilidad que se mantiene bajo la arquitectura REST que se maneja en la construcción de estos Sub-Sistemas, TITAN debe hacer uso de los servicios desarrollados en ARGO para consumir la información relacionada en este caso a las personas contratadas bajo la modalidad DVE (Docentes de Vinculación Especial) y la información de los contratos de esta modalidad que les pertenecen. Es importante aclarar que la información que se consume de ARGO es únicamente la de los contratos debido a que la información personal de cada persona se encuentra almacenada dentro del Banco de proveedores con el fin de , como se mencionaba con anterioridad, reducir la redundancia de información dentro de las bases de datos de la Universidad.

13.1.2.2 Banco de proveedores.

Este Sub-Sistema, para el proceso de la liquidación de la nómina, tiene como objetivo prestar el servicio de consulta de la información personal de la persona en cuestión a la que hace referencia un contrato que se pretende liquidar para su pago mensual. Esta información es usada para referenciar a la persona a quien corresponde el cálculo para posteriormente en un proceso ajeno al de la liquidación expedir las órdenes de pago que hacen efectiva la remuneración del docente en cuestión por el mes liquidado en el sistema. Este servicio no es directamente consumido por TITAN, sino que, este es un proceso intermedio en el servicio prestado por AGORA para enlazar los datos de un contrato especifico con la persona a la cual corresponde este mismo.

13.1.2.3 Ruler.

El Modulo RULER es común para todos los demás Sub-Sistemas, este es el encargado de proveer el servicio de consulta de las reglas escritas en lenguaje de programación lógica definidas para cada proceso de negocio que se efectúa mediante alguno de los Sistemas anteriormente mencionados.

En este caso, el RULER funciona como un Sub-Sistema de consulta, por este motivo este también presta servicios REST como se definió en la arquitectura, este servicio lo que hace es mediante una petición devolver la información de las reglas lógicas a aplicar basándose en un dominio de conocimiento, este proceso se evidencia mejor en la Figura 16 donde se describe la arquitectura de la información de esta parte del proceso:



Figura 16 Arquitectura de la información RULER Fuente: Realizado por el autor(es).

Como se puede ver, un dominio dentro de la arquitectura del RULER está compuesto por varios predicados, estos predicados contienen los datos de las reglas lógicas que rigen un proceso de negocio en específico y que serán interpretadas por un motor de procesamiento lógico inmerso dentro de un módulo

en cada uno de los subsistemas que integran la centralización de la que se ha hablado con anterioridad.

13.1.2.4 TITAN.

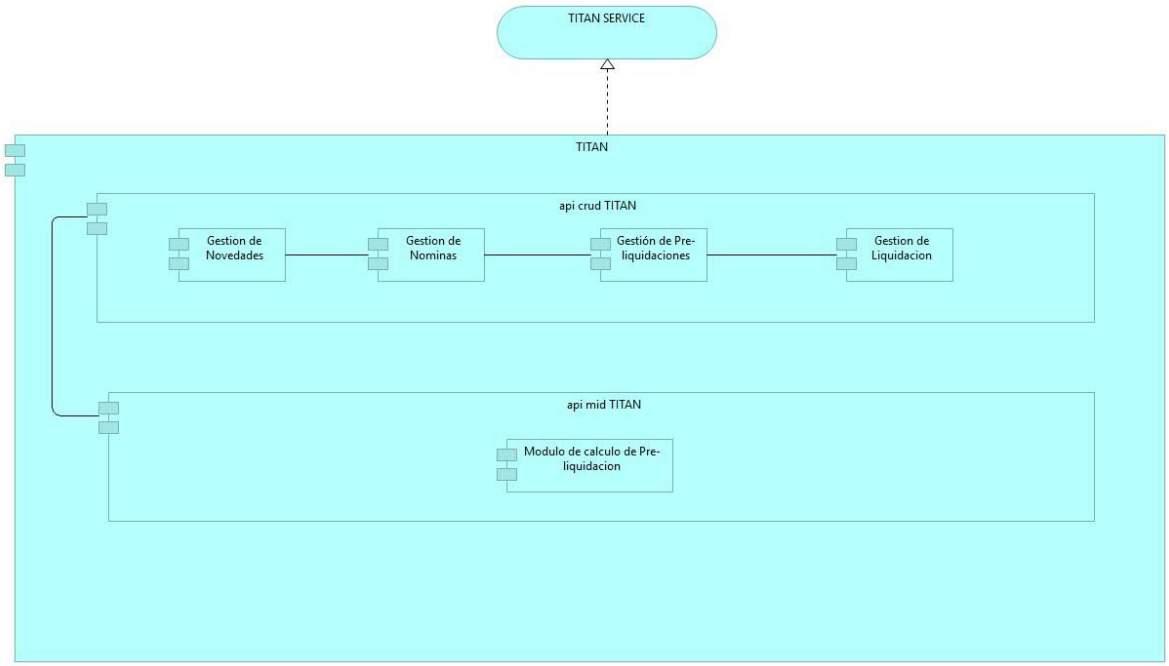


Figura 17. Arquitectura Interna TITAN. Fuente: Realizada por el Autor(es)

En el anterior diagrama se describe de forma gráfica la arquitectura que lleva el modulo del Sub-Sistema TITAN, esta arquitectura es similar para los demás Sub-Sistemas con cambios en los módulos internos que las componen, en general los módulos de cada uno de los Sub-Sistemas están compuestos por dos API REST, uno que es el encargado de las operaciones sobre la base de datos (api CRUD) y otro que se encarga de operaciones intermedias, enfocándose en la lógica del negocio (api MID).

En primer lugar, el api **CRUD** se encarga de prestar los servicios de guardado, modificación, borrado y consulta de datos guardados en el esquema perteneciente al Sub-Sistema, estos servicios serán suministrados a los Módulos que necesiten consultar dicha información en sus procesos internos.

Esta consulta se realiza usando un Mapeador de estructuras que representan las tablas presentes en los esquemas de la base de datos, el mapeador que actualmente se usa dentro del proyecto es el ORM (Object-Relational mapping) dentro de este se declaran estructuras que van a contener los datos de cada tabla, las relaciones entre tablas se manejan en forma de apuntadores, referencias a espacios de memoria donde se almacena los datos dentro de otra estructura, la forma en que el mapeador realiza esta operación es realizando una consulta SQL con JOINS entre tablas, dichos JOINS son realizados por medio de los PK que se declaran en las estructuras, de esta manera el mapeador sabe cómo relacionar las tablas y posteriormente las estructuras para realizar las operaciones que sean demandadas mediante los servicios que el api presta.

El api CRUD de TITAN por su parte está compuesto por cuatro módulos distintos, estos son:

- **Gestión de nóminas:** Este módulo se encarga de realizar todas las operaciones de las nóminas, estas son consultadas de los contratos y su información relacionada mediante peticiones a los servicios expuestos en el módulo ARGO como ya se describió en puntos anteriores, la función específica de este módulo es la de listar las nóminas registradas por vigencia, también permite registrar nominas según tipos de contratación.
- **Gestión de Pre-liquidaciones:** Dentro de este módulo se realizan las Pre-liquidaciones referentes a una nómina, estas preliquidaciones son simples registros que, como se describió en el capítulo 12, referencian los datos generales de un cálculo de preliquidación, estos datos son usados en las reglas lógicas que se aplican en el proceso de generación de cálculos por contrato durante el periodo que indique la preliquidación para los docentes de vinculación especial (DVE) y para las demás nóminas. Para cada preliquidación se guarda un detalle por descuento calculado a un contrato en específico, estos descuentos son calculados en el api MID del cual se aclaran sus aspectos más relevantes más adelante.
- **Gestión de Liquidación:** Este es el punto final del proceso de liquidación y de donde los otros Sub-Sistemas tomaran los servicios que necesitan, pues en este es donde se realizan los cierres de las nóminas mes a mes, es decir este módulo provee los servicios de consulta de los datos resultado para realizar los pagos de los sueldos de los contratos tipo DVE y los demás que existen dentro de la Universidad.

Por otra parte, el módulo TITAN también está compuesto por un api intermedio entre el **CRUD API** y el **RULER API** este se denomina dentro la arquitectura de la aplicación como **MID API,** para el caso de TITAN este solo contiene un módulo,

este módulo es el encargado de realizar operaciones intermedias de consulta de información propia de la nómina DVE que, como se mencionó con anterioridad, comprende la información básica del docente vinculado al contrato, las fechas de inicio y finalización del contrato junto con el valor total del contrato consultado. El proceso de cálculo que se lleva a cabo en este módulo del sub-sistema se describe de manera detallada en la Figura 18 a continuación:

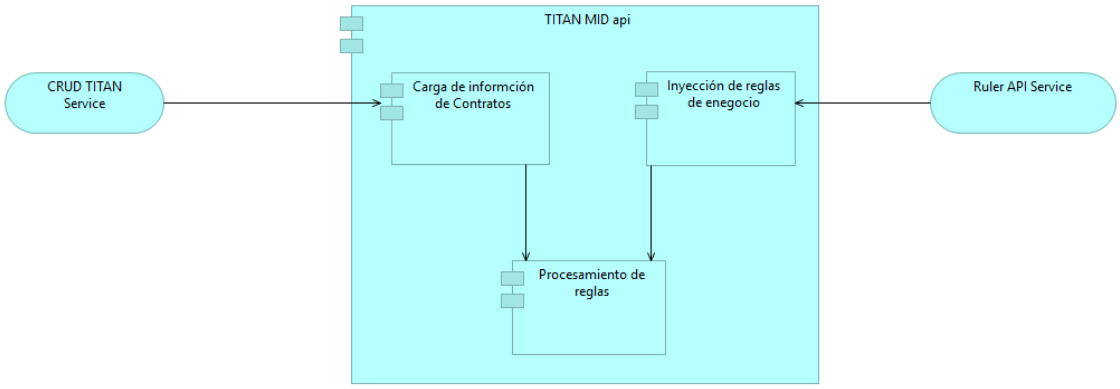


Figura 18. Módulo de cálculo de liquidación. Fuente: Realizada por el Autor(es).

El funcionamiento del MID API en TITAN comprende tres fases que a continuación serán descritas:

- **Fase de carga de información de los contratos:** En esta se realiza la consulta de la información de los contratos registrados con el tipo DVE en el sub-sistemas de contratación ARGO, mediante un servicio REST asociado, tomando en cuenta las fechas de inicio y finalización de los contratos.
- **Fase de Inyección de reglas de negocio:** en esta fase se realiza la petición al Ruler API, mediante su respectivo servicio REST asociado, de las reglas lógicas asociadas al dominio de la preliquidación objetivo (Horas catedra Honorarios y Salarios para este caso), y este resultado es consumido por el MID API de TITAN para realizar los cálculos posteriores con base en dichas reglas.
- **Fase de procesamiento de reglas:** en esta fase se toman los datos consumidos de los servicios anteriormente mencionados (ARGO y RULER), en primera instancia se debe dar un formato previo a los datos de ARGO, ya que se deben escribir como hechos de programación lógica debido a que estas reglas serán procesadas en un motor lógico que funciona en el lenguaje Go llamado **GOLOG,** este formateo no es más que un

encadenamiento de cadenas de caracteres del tipo *HECHO(X, Y).* este encadenamiento permite que el proceso automáticamente escriba los hechos que alimentaran las reglas que realizan los cálculos de las liquidaciones de cada contrato. Una vez se realiza este formateo de reglas se pasan dichas reglas como una sola cadena de caracteres a una función de la librería de **GOLOG,** esta función se encarga de realizar la compilación y procesamiento de las reglas. De esta manera se podrá llamar posterior mente a cualquiera de las reglas de negocio y esta retornara el dato a una variable dentro del módulo que hizo el llamado, para este caso se llama la reglas que trae el valor de la liquidación y los valores de los descuentos que a este le corresponden.(Lezcano, M., & Valdés, G., 2016)

CAPÍTULO 14. CONCLUSIONES

Durante el periodo de tiempo mediante el cual se llevaron a cabo las pasantías de las cuales el presente documento sirve como informe, se realizaron diferentes etapas del diseño e implementación del módulo TITAN como se evidencio en puntos anteriores. Dichas etapas fueron revisadas, en cuanto a la parte de diseño, durante los sprint semanales realizados siguiendo los lineamientos de la metodología de desarrollo ágil SCRUM/OAS, donde se consiguió un avance significativo en cuanto a tiempo de desarrollo ya que lo planteado al principio del proyecto era un plazo de 24 Sprint's para la entrega del software liquidando nominas HC y esto se consiguió en un periodo de 16 Sprint's.

Durante cada sprint se pasó por revisiones de los modelos de datos y reglas de negocio hasta tener un producto que cumplía con las características a nivel de diseño y desarrollo, durante dichas etapas se hicieron varias mejoras al proceso de desarrollo en general definiendo generalizaciones y módulos que pueden ser usados en otros proyectos. Uno de los avances que se tuvo durante este desarrollo fue la modificación de la librería **GOLOG** donde para realizar los cálculos de las nóminas según las reglas de negocio (ver Capitulo 14) se deben realizar redondeos en los cálculos realizados, esta función no está presente en la librería **GOLOG** la cual se usó para el procesamiento de las reglas de negocio escritas en lenguaje de programación lógico, por este motivo se debió hacer una modificación en algunos de los módulos de esta librería donde se logró agregar nuevas funciones por parte del equipo de desarrollo. Esto genero un avance importante en la implementación de reglas de negocio basadas en programación lógica dentro del software lo cual reduce las modificaciones directas al código de la aplicación.

Otro de los avances que consiguieron respecto a este proyecto es la generalización del **MID API,** dicha generalización se realizó con el fin de que al agregar una nueva nomina con nuevas reglas o consultas no afectara el proceso desarrollado para las demás, de este modo el proceso de liquidación mediante TITAN es extensible a nuevos tipos de nómina con diferentes reglas de negocio gracias a la ya nombrada generalización realizada y al método de realizar los procesos mediante reglas de negocio cargados desde un api aparte como lo es el **RULER API.**

CAPÍTULO 15. BIBLIOGRAFIA

- Oficina Asesora de Sistemas. (2016), Proceso de Desarrollo SCRUM/OAS, Universidad Distrital Francisco José de Caldas, Colombia.

- Gallego, M. T. (2012). Metodología Scrum. *Gestión de Proyectos Informáticos, http://openaccess. uoc. edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria. pdf*.

- Guillermo Pantaleo. (2015). Metodologías ágiles. En Ingeniería de Software (92-93). Buenos Aires, Argentina: Alfaomega.

- Universidad Distrital Francisco José de caldas (2003), acuerdo 003, Colombia.

- Oficina asesora de sistemas. (2013), Levantamiento de la primera versión del proceso de parametrización y liquidación de nómina soportado en el aplicativo de nómina de funcionarios y pensionados de la Universidad Distrital. Colombia.

- Delgado Colmenares, M. R. (2017). DISEÑO DE UNA HERRAMIENTA TECNOLOGICA PARA LA LIQUIDACION DE NOMINA Y EL MEJORAMIENTO EN EL CALCULO DE HORAS EXTRAORDINARIAS Y RECARGOS ADICIONALES EN LA EMPRESA TOSCHEM DE COLOMBIA LTDA. SECCIONAL BARRANCABERMEJA.

- *Flores Andrade, V. A., & Galarza Bolaños, V. R. (2016). Análisis, Diseño, Construcción e Implementación en ambiente de pruebas de los módulos de nómina, liquidación de personal y reportería para manejo de información en el Área de Talento Humano de la Superintendencia de Control de Poder de Mercado (SCPM) (Bachelor's thesis)*.

- The Open Group. (2013). ArchiMate® 2.1 Specification. (2017), de The Open Group Sitio web: http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html#_Toc371945245

- Flaishans, J., Galvin, M., Ignatius, A., Kuan, C., Laniak, G. F., Wolfe, K., & Purucker, T. (2016). Environmental Models as a Service: Enabling Interoperability through RESTful Endpoints and API Documentation.

- Cao, H., Falleri, J. R., Blanc, X., & Zhang, L. (2016, October). JSON Patch for Turning a Pull REST API into a Push. In International Conference on

Service-Oriented Computing (pp. 435-449). Springer International Publishing.

- Lezcano, M., & Valdés, G. (2016). Consideraciones sobre la construcción de sistemas expertos utilizando el lenguaje Prolog. Revista Facultad de Ingeniería, (15), 111-118.