



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

SISTEMA WEB PARA MONITOREO DE ACTIVIDADES, GESTION Y  
CONTROL DE PAGOS ADICIONALES PARA SERTECPET S.A.

Autor

Augusto César Yépez Dávila

Año  
2017



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

SISTEMA WEB PARA MONITOREO DE ACTIVIDADES, GESTION Y CONTROL  
DE PAGOS ADICIONALES PARA SERTECPET S.A.

Trabajo de Titulación presentado en conformidad a los requisitos establecidos  
para optar por el título de Ingeniero en Sistemas de Computación e Informática

Profesor guía

Mgs. Santiago Ramiro Villarreal Narvárez

Autor

Augusto César Yépez Dávila

Año

2017

## DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

---

Santiago Ramiro Villarreal Narváez

Master Science Technologies, Sante a Finalite Recherche

Mention Informatique Specialite Web Intelligence

CC: 1713980074

## DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

---

Adonis Ricardo Rosales García  
Máster en Informática Aplicada  
CC: 1756883144

## DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

---

Augusto César Yépez Dávila

CC: 1715853410

## AGRADECIMIENTOS

Agradezco a mi familia, a mi esposa y a mis hijos por siempre apóyame y darme fuerzas para continuar y terminar la carrera de Ingeniería en sistemas de computación e informática.

## DEDICATORIA

Dedico el presente trabajo de titulación a toda mi familia, que siempre han sabido apoyarme y darme fuerzas para seguir adelante.

## RESUMEN

El presente trabajo de titulación surge por la necesidad de optimizar el control de actividades diarias de los colaboradores de las áreas de operaciones y proyectos, así también como por la necesidad de la urgente automatización del proceso de pagos adicionales, que en la actualidad se lleva a cabo en fichas obsoletas de Excel.

El objetivo principal de este proyecto de titulación consiste en la implementación de un sistema web que permita monitorear actividades, gestionar y controlar pagos adicionales para las áreas de operaciones y proyectos de la compañía Sertecpet S.A., esto debido a la constante sobrecarga de trabajo que tiene el área de nómina para la realización de este tedioso proceso.

Para el desarrollo de este sistema se utilizó la metodología de desarrollo de software ágil Scrum. Conjuntamente con Bonita BPM que es una de las mejores herramientas de código abierto para automatizar procesos del negocio y flujos de trabajo. Adicionalmente se utilizó los más modernos frameworks de interfaz de usuario, como lo son angularJS, KendoUI y Bootstrap, con el fin de mejorar la experiencia de usuario.

## ABSTRACT

The present titling work arises from the need to optimize the control of daily activities for employees in the areas of operations and projects, as well as the need for the automation of the additional payments process, which is currently carried out in obsolete Excel tabs.

The main objective of this titling project is the implementation of a web system to monitor activities, manage and control additional payments for the areas of operations and projects of the company Sertecpet SA, due to the constant overload of work Payroll area for carrying out this tedious process.

For the development of this system were used the agile software development methodology Scrum. Together with Bonita BPM that it's one of the best open source tools to automate business processes and workflows. Additionally were use the most modern user interface frameworks, such as angularJS, KendoUI and Boostrap, in order to improve the user experience.

# ÍNDICE

Antecedentes .....	1
Alcance.....	2
Justificación.....	4
Objetivos .....	5
Objetivo general.....	5
Objetivos específicos .....	5
1. Marco Teórico .....	6
Introducción .....	6
1.1. Metodologías tradicionales.....	6
1.2 Metodologías ágiles .....	9
1.2.1 Equipo Scrum .....	11
1.2.2 Dueño del producto .....	11
1.2.3 Equipo de Desarrollo .....	11
1.2.4 Eventos Scrum: .....	12
1.3 Tecnologías actuales .....	13
1.4 Arquitectura BPM en general .....	14
1.5 Arquitectura MVC .....	16
1.5.1 Ciclo de vida MVC .....	16
1.5.2 Ventajas y desventajas.....	17
1.6 Lenguajes de programación.....	18
1.7 Integración de sistemas .....	19
1.7.1 Concepto de integración.....	19
1.7.2 Tipos de integración .....	19
1.8 Conclusiones parciales del capítulo .....	25

2	Análisis y diseño del sistema .....	26
2.1	Scrum .....	26
2.1.1	Datos generales.....	26
2.1.2	Historias de usuario.....	27
2.1.3	Product Backlog.....	30
2.1.4	Iteraciones .....	31
2.2	Diagramas de Casos de Uso.....	32
2.3	Diagrama Entidad Relación.....	40
2.4	Análisis y diseño de la integración con sistema Squarenet.....	41
2.4.1	Diagrama de integración.....	42
2.5	Análisis e integración unidireccional con sistema biométrico...	42
2.5.1	Diagrama de integración.....	43
2.6	Arquitectura del sistema .....	44
2.7	Conclusiones parciales capítulo 2 .....	47
3	Desarrollo, resultados y pruebas del sistema .....	49
3.1	Desarrollo del sistema .....	49
3.2	Resultados .....	61
3.3	Pruebas de caja negra .....	62
3.4	Pruebas de caja blanca.....	71
4	Conclusiones y recomendaciones .....	79
4.1	Conclusiones.....	79
4.2	Recomendaciones.....	80
	REFERENCIAS.....	81
	ANEXOS .....	83

## ÍNDICE DE FIGURAS

Figura 1. Ciclo de vida BPM.....	15
Figura 2. Ciclo de vida MVC.....	16
Figura 3. Esquema de integración por transferencia de archivos.....	20
Figura 4. Esquema de integración por bases de datos compartidas.....	21
Figura 5. Esquema de integración por RCP.....	23
Figura 6. Esquema de integración por mensajería.....	24
Figura 7. Caso de uso generar organizador de personal.....	33
Figura 8. Registrar actividades diarias.....	36
Figura 9. Aprobar conjunto de actividades diarias.....	38
Figura 10. Diagrama Entidad Relación.....	40
Figura 11. Diagrama de integración con sistema Squarenet.....	42
Figura 12. Diagrama de integración con sistema Biométrico.....	44
Figura 13. Diagrama de arquitectura de Bonita BPM.....	45
Figura 14. Diagrama de arquitectura de sistema de pagos adicionales.....	47
Figura 15. Organizador de personal.....	49
Figura 16. Wizard de conector con Squarenet flujo organizador de personal parte 1.....	50
Figura 17. Wizard de conector con Squarenet flujo organizador de personal parte 2.....	51
Figura 18. Wizard de conector con Squarenet flujo organizador de personal parte 3.....	52
Figura 19. Wizard de conector con Squarenet flujo organizador de personal parte 4.....	53
Figura 20. Flujo de ingreso de actividades diarias para operaciones y proyectos.....	54
Figura 21. Wizard integración con sistema Biométrico, flujo de ingreso de actividades diarias parte 1.....	54

Figura 22. Wizard integración con sistema Biométrico, flujo de ingreso de actividades diarias parte 2.....	55
Figura 23. Formulario de registro de actividades diarias.....	56
Figura 24. Widget personalizado botón continuar, flujo registro de actividades diarias.....	56
Figura 25. Flujo de aprobaciones gerenciales.....	57
Figura 26. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 1 .....	58
Figura 27. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 2 .....	58
Figura 28. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 3 .....	59
Figura 29. Widget de grilla personalizada con Bonita BPM UI Designer.....	60
Figura 30. Calculo horas extraordinarias y suplementarias.....	60
Figura 31. Vista resumen y detalle de actividades mensuales por colaborador, flujo aprobaciones gerenciales .....	61

## ÍNDICE DE TABLAS

Tabla 1. Diferencias entre metodologías tradicionales y ágiles.....	9
Tabla 2. Datos de las iteraciones del proyecto.....	26
Tabla 3. Roles Scrum.....	27
Tabla 4. Instalación de Bonita BPM en ambiente de desarrollo .....	28
Tabla 5. Configuración de Bonita BPM .....	28
Tabla 6. Configuración de la base de datos .....	29
Tabla 7. Pruebas del ambiente de desarrollo .....	29
Tabla 8. Muestra de las primeras 7 historias de usuario del Product Backlog. ....	30
Tabla 9. Prueba de caja negra del módulo login. ....	62
Tabla 10. Prueba de caja negra del módulo creación de usuarios.....	64
Tabla 11. Prueba de caja negra del módulo importación/exportación de organización .....	67
Tabla 12. Prueba de caja negra del módulo instalación de flujos.....	69
Tabla 13. Prueba de caja negra sobre el módulo de aplicaciones, creación de vista mantenimiento tabla clientes.....	70
Tabla 14. Prueba de caja blanca, operación read sobre la tabla cliente .....	72
Tabla 15. Prueba de caja blanca, operación create sobre la tabla cliente .....	73
Tabla 16. Prueba de caja blanca, operación update sobre la tabla cliente .....	75
Tabla 17. Prueba de caja blanca, operación delete sobre la tabla cliente.....	76

## **Antecedentes**

Sertecpet es una empresa enfocada en generar soluciones energéticas integrales para el sector energético y hidrocarburífero con tecnología de punta, comprometidos con el medio ambiente además, siguiendo los más altos estándares internacionales de calidad, salud y seguridad ocupacional.

Actualmente Sertecpet no cuenta con ningún sistema automatizado para controlar las actividades diarias de sus colaboradores de las áreas de operaciones y proyectos que se encuentran en zonas remotas del oriente ecuatoriano. Al estar apartados estas locaciones no se puede controlar de manera eficiente las actividades diarias de estos colaboradores por diversos motivos como son en muchas ocasiones la falta de comunicación, la dificultad del camino de ingreso a estos sitios, entre otros. Estos colaboradores son los únicos que reciben remuneraciones adicionales, por lo que se ha decidido la creación de un sistema web que determine múltiples indicadores para la determinación de pagos de bonos, horas extras, días extras y días adicionales para talento humano, que sería el principal beneficiado debido a que disminuiría considerablemente la carga de trabajo actual que requiere este proceso. Por otra parte las gerencias de operaciones y proyectos podrían utilizar la información del sistema para realizar planes de optimización de recursos y trabajos realizados en estas locaciones.

Es muy importante conocer las actividades diarias de los colaboradores, debido a que siempre se puede optimizar el trabajo realizado en campo, controlando que los colaboradores siempre estén trabajando y no estén en standby o descanso en horas o días de jornada laboral, esto por lo tanto llevaría a disminuir costos operativos, y mejorar la productividad del trabajo, lo cual beneficiaría a toda la compañía.

## **Alcance**

El alcance de este proyecto es la creación de una aplicación web, enfocada principalmente al personal de operaciones y proyectos, quienes regularmente trabajan en zonas apartadas de las instalaciones de la empresa, por lo que se busca llevar un mejor control de actividades diarias y pagos adicionales, para que a corto plazo aprovechar de mejor manera el recurso humano.

Para la elaboración de este proyecto se utilizará la metodología ágil SCRUM, la cual “es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto” (SCRUM, 2015). Lo que se pretende es realizar entregas parciales del producto software, hasta llegar a una versión final del mismo, pasando por todo el proceso SCRUM.

El sistema se lo desarrollará en tres módulos que se dispondrán en orden numérico (módulo 1, módulo 2 y módulo 3), pero no siguen un orden cronológico, ya que dependen el uno del otro.

Como primer módulo, se tendrá la construcción del sistema como tal, en esta primera etapa se implementará por primera vez en la compañía un gestor de procesos del negocio, lo cual permitirá abrir futuros desarrollos orientados a la automatización de procesos para la empresa. La selección del lenguaje de programación del lado del servidor, se la realizara una vez escogido el gestor de procesos de negocio, esto debido a que cada BPM está ligado a un lenguaje de servidor base. Por otra parte como lenguaje de programación del lado del cliente, se utilizara como base a Javascript, y posteriormente se pueden ir agregando más lenguajes de programación del lado del cliente, los cuales permitan facilitar el desarrollo.

Como motor de base de datos se utilizará a SQL Server, esto debido a que es un estándar en la construcción de sistemas que utiliza la compañía, además que facilitara la integración con cualquier otro sistema actual de la compañía.

El segundo módulo será la integración con el sistema SQUARENET, la cual será una integración de tipo bidireccional. Lo que se pretende realizar primero en esta integración es la consulta de información personal del colaborador, mediante el consumo de un servicio web proporcionado por el sistema SQUARENET, el cual tendrá información requerida para interactuar, controlar, restringir y a su vez mostrar información del empleado. La segunda etapa de esta integración se la realizará al final de la etapa de desarrollo, esto debido a que requiere extraer diversa información del sistema con el fin de generar un archivo CSV o XML, que contenga data de todos los colaboradores con las cantidades de bonos, días extras y días adicionales, para su posterior lectura y carga en el sistema SQUARENET, haciendo uso de la integración mediante transferencia de archivos.

Finalmente en el tercer módulo, se realizará la integración con el sistema biométrico de asistencia, esta integración será una integración unidireccional, ya que se consultará información del colaborador, con el fin de controlar que no se pueda ingresar las actividades diarias sin antes haberse registrado primero en el sistema biométrico.

Otro de los fines del sistema es que podrá servir como indicador gerencial, para la ayuda en la toma de decisiones del día a día acerca de las actividades necesarias para que las actividades relacionadas con las ocupaciones de los servicios petroleros ofertadas por la compañía sean más rentables, mediante la optimización de recursos.

## **Justificación**

Actualmente este proceso de gestión de nómina se controla mediante un formato establecido en fichas de Excel, que se realiza una vez al mes, donde los colaboradores que se encuentran en campo se encargan de llenar sus actividades, bonos, horas extras, días extras y días adicionales basados en el criterio propio o del supervisor, lo que podría causar en muchas ocasiones que se generen pagos de manera errónea a más de 600 colaboradores, dando lugar a que se generen grandes pérdidas económicas a la compañía.

El área de talento humano se basa en información emitida desde campo, la cual como se mencionó anteriormente es difícil de recolectar por medios electrónicos o físicos por la cantidad de colaboradores en las áreas de Operaciones y Proyectos, esto debido a que se tiene una gran cantidad de correos electrónicos, documentación física, las cuales se debe filtrar, copiar y pegar en un formato de ficha de Excel establecida por talento humano. Por otra parte esta información en muchas ocasiones se la envía retrasada por parte de los supervisores, lo que dificulta y retrasa el proceso de nómina. La intención de este proyecto es dinamizar, facilitar y asegurar este complejo proceso.

Con el sistema se tendrá información más eficaz, veraz y actualizada en menor tiempo, ya que será 100% digital y podrá ser auditada constantemente, trayendo ahorros a la empresa, además que el sistema será de alta confiabilidad porque son datos ingresados diariamente para mayor control del personal que labora en estas dos grandes áreas de la compañía.

## **Objetivos**

### **Objetivo general**

- Implementar un sistema web para automatizar el control de actividades diarias, y gestión de pagos adicionales para las áreas de operaciones y proyectos para Sertecpet S.A.

### **Objetivos específicos**

- Crear una aplicación web para automatizar el proceso de pagos adicionales.
- Crear la integración bidireccional, con el sistema de talento humano SQUARENET para la carga mensual de pagos adicionales.
- Crear la integración con el sistema biométrico de asistencias para el control diario de ingreso de actividades.
- Validar el correcto funcionamiento del sistema, haciendo uso de las pruebas de caja negra y caja negra.

# 1. Capítulo I. Marco Teórico

## Introducción

En este primer capítulo se contemplará todo el marco teórico del presente trabajo de titulación. Primero analizara y se seleccionara una metodología de desarrollo, realizando una comparación de las metodologías ágiles vs tradicionales, para así seleccionar la metodología que se utilizará a lo largo del desarrollo del presente trabajo de titulación.

Así mismo se analizara un BPM, revisando la arquitectura de los BPM en general, y escogiendo el que mejor se adapte a las necesidades de la compañía, el cual vendrá acompañado con un lenguaje de programación base del lado del servidor, y uno o varios lenguajes de programación del lado del cliente.

Finalmente se observaran las conclusiones parciales del capítulo, en donde se mirara los motivos por los cuales se seleccionaron los leguajes de programación y el motor de base de datos que se utilizan el este trabajo de titulación.

### 1.1. Metodologías tradicionales

Las metodologías tradicionales surgen como una guía para garantizar la creación de un producto con un alto nivel de calidad. Estas metodologías tienen como pilar una gestión predictiva, lo cual quiere decir que parten de unos requisitos iniciales poco cambiantes, para cumplir con estos requisitos, se configura un plan adecuado de recursos y tiempo necesarios. Estas metodologías definen un conjunto de etapas secuenciales en las que se indican las operaciones que se van a realizar, el tiempo que van a llevar, y el costo de las mismas.

Algunas características relevantes de las metodologías tradicionales son:

- Los requisitos son puntuales durante todo el proyecto.
- No están sujetos a cambios en los requisitos iniciales.
- Documentación extensa.
- Desarrollo dividido en fases y cuyo conjunto posee un ciclo de vida.
- Desarrollos en largos periodos de tiempo.
- Gestión predictiva.

La primera metodología tradicional es la metodología en cascada, la cual se caracteriza debido a que todas sus etapas son secuenciales, y se debe finalizar una etapa antes de que la otra inicie, y cada etapa está compuesta por personas o grupos de trabajo especializados para cada fase. Las fases de esta metodología son las siguientes:

- Análisis y definición de requerimientos
- Diseño del sistema
- Implementación y pruebas de unidad
- Integración y pruebas de sistema
- Operación y mantenimiento

El principal inconveniente de esta metodología, es que los requisitos deben estar bien definidos desde el principio del proyecto, y estos no pueden cambiar o modificarse, cosa que muy pocas veces es aplicable, ya que el cliente no siempre plantea bien sus requerimientos desde del principio, sino que estos varían.

Otra metodología tradicional muy conocida es RUP (Rational Unified Proces), la cual pertenece a IBM, esta metodología es la resultante de la combinación de varias metodologías y se vio influenciada por métodos como es espiral. Está basada en Objectory, metodología creado por Iván Jacobson.

RUP posee tres módulos centrales, los cuales contestan a las preguntas de: Quien hace el proceso, que productos de trabajo se van a realizar, que documentos y modelos se van a producir y cómo se van a realizar las tareas.

Posee cuatro fases, las cuales son:

- **Inicio:**  
Se establece el objetivo del sistema y se recogen los requisitos del usuario.
  
- **Elaboración:**  
Reducción de costos y riesgos.
  
- **Construcción:**  
Partiendo de la arquitectura se realiza la implementación.
  
- **Transición:**  
Se verifica que el software cumpla con las expectativas y se realiza un feedback.

Sin embargo esta metodología también requiere gran cantidad de documentación y recursos, por lo que sigue siendo poco usada en épocas actuales, y por lo cual se ha visto la necesidad de mejorar las metodologías tradicionales y eliminar la gran cantidad de documentación generada.

## 1.2 Metodologías ágiles

En el transcurso de esta década se ha comenzado con un progresivo interés por las metodologías ágiles de desarrollo de software, dejando de lado el esquema tradicional de desarrollo, el cual es más efectivo para proyectos con de gran tamaño, con respecto al tiempo y recursos.

“En la década de 1990 el descontento con estos enfoques engorrosos de la ingeniería de software condujo a algunos desarrolladores de software a proponer nuevos “métodos ágiles”, los cuales permitieron que el equipo de desarrollo se enfocara en el software en lugar del diseño y la documentación. Los métodos ágiles se apoyan universalmente en el enfoque incremental para la especificación, el desarrollo y la entrega del software.” (Sommerville, I. 2011).

El objetivo primordial del desarrollo ágil es la entrega incremental del producto software operativo, para así disminuir notablemente la documentación que quizá nunca se emplee, con esto se permite que el equipo de desarrollo se enfocare en el software en lugar del diseño y la documentación.

A continuación se visualizara algunas de las más importantes diferencias entre metodologías ágiles y tradicionales:

Tabla 1.

Diferencias entre metodologías tradicionales y ágiles.

<b>Metodologías tradicionales</b>	<b>Metodologías ágiles</b>
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código

Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Tomado de (Figuerola, R., Solís, C., Cabrear, A., 2008)

Para el desarrollo de esta tesis se seleccionó la metodología de desarrollo ágil Scrum, la cual tiene un proceso que ayuda a la administración y control del desarrollo de software. El desarrollo con esta metodología, se realiza de forma iterativa e incremental, con lo que cada iteración es un ciclo corto de construcción del software. Al finalizar cada iteración, se termina una pieza de software funcional, la cual incorpora nueva funcionalidad. Por lo general las iteraciones tienen una duración entre 2 y 4 semanas.

### **1.2.1 Equipo Scrum**

Los equipos auto organizados, seleccionan su mejor manera de trabajar, por lo que pueden ser multifuncionales y no dependen de terceros para llevar a cabo su trabajo, sino solo del propio equipo de trabajo, lo cual permite mayor flexibilidad, brindando una mejor productividad y creatividad.

### **1.2.2 Dueño del producto**

El dueño del producto es el encargado de la gestión del Product Backlog, la cual incluye:

- Enunciar los elementos del Product Backlog.
- Ordenar estos elementos para alcanzar los objetivos planteados.
- Mejorar el desempeño del equipo de trabajo.
- Asegurarse que el Product Backlog es entendible para todo el equipo.

El dueño del producto tiene la potestad de realizar estas actividades o delegar a alguien del equipo para que las realice, por lo que es importante que se respete las decisiones tomadas por el Dueño del Producto.

### **1.2.3 Equipo de Desarrollo**

Todo el Equipo de Desarrollo debe estar en sinergia para la optimización de la eficiencia y efectividad del equipo. Algunas características importantes de estos equipos son la auto organización y multifuncionalidad de los miembros del equipo.

#### **1.2.4 Eventos Scrum:**

Un evento Scrum es el Sprint, el cual es el núcleo del Scrum, en el cual se crea producto terminado y funcional. Cada Sprint surge de las reuniones planificadas (Sprint Planning Meeting). Cada Sprint debe durar como máximo un mes, el Sprint Planning Meeting debe durar un máximo de 8 horas.

Otro evento del Scrum es el Objetivo del Sprint (Sprint Goal), el cual es una meta preestablecida del Sprint y es alcanzada por la implementación del Product Backlog.

El Scrum Diario (Daily Scrum), es otro evento Scrum, el cual consiste en una reunión diaria de una duración máxima de 15 minutos, para verificar el avance o dificultades en el desarrollo del producto, y además para planificar las actividades del día siguiente.

La revisión de Sprint (Sprint Review), en la cual se realiza una evaluación de lo que fue todo el Sprint, se recomienda que no se extienda más allá de 4 horas.

Finalmente el último evento Scrum es la Retrospectiva de Sprint (Sprint Retrospective), cuyo propósito es la inspección del último Sprint, además de identificar y organizar los elementos más importantes que salieron bien y las posibles opciones de mejoras, para así crear un plan de mejora para todo el Equipo.

### 1.3 Tecnologías actuales

En la actualidad, debido al crecimiento de internet, y la alta demanda en las necesidades de manejo y control de información, se ha visto la necesidad de que los sistemas evolucionen de un esquema tradicional, de aplicaciones solo para escritorios a aplicaciones para la web o dispositivos móviles. Para así impulsar el uso de dispositivos móviles para uso cotidiano, los cuales, al conectarse a internet, permiten la creación de nuevas e interesantes aplicaciones que han venido revolucionando la comunicación.

Para alcanzar esta etapa primero se vio la necesidad de evolucionar las tecnologías web tradicionales para el procesamiento de hipertexto y hojas de estilos, los cuales son HTML, CSS y Javascript. La Web actual es el resultado de los continuos esfuerzos de una comunidad web abierta que ha ayudado y evolucionar estas tecnologías web a las actuales que con HTML5, CSS3.

Las nuevas características del HTML5, fueron pensadas para sustituir la dependencia de los plug-in multimedia, por estándares abiertos, permitiendo a las aplicaciones web comportarse más nativamente. En ejemplo claro de eso es las cuevas característica de soportar audio y video en el navegador, sin la necesidad de instalar ningún plug-in que en muchas ocasiones aumentaba el tiempo de carga de la página, con lo era Abode Flash o Microsoft Silverlighth para contenido multimedia.

Otra característica del estándar HTML5 es la habilidad de construir aplicaciones web que se puedan comportar más como aplicaciones nativas, por ejemplo los drag-and-drop archivos y los local storage permiten a desarrolladores construir aplicaciones basadas en web. Así también los elementos Canvas los cuales permiten combinar vectores gráficos, imágenes, audio y video en un espacio predefinido de la página. Otro gran característica de es la geo localización.

Con respecto al CSS3, la mayor diferencia con CSS2, es la separación de los módulos, dividido en varios documentos llamados módulos. Cada módulo cuenta con nuevas capacidades, sin afectar la compatibilidad de la versión anterior, cuatro de ellos han sido recomendaciones formales, y se componen de lo siguiente:

- Media Queris
- Selectores nivel 3
- Namespaces

Para el desarrollo del presente trabajo de titulación se utilizará la tecnología web HTML5, en conjunto con la tecnología web para hojas de estilos CSS3, esto debido a que son tendencias de la web actual.

## **1.4 Arquitectura BPM en general**

Un Business Process Management “orienta los esfuerzos para la optimación de los procesos de la empresa, en busca de mejorar la eficiencia y la eficacia por medio de la gestión sistemática de los mismos.” Estos procesos son modelados y automatizados con el fin optimizar la productividad y eficiencia de una organización, mediante la optimización de sus procesos de negocio. Un BPM aborda los problemas típicos del día a día en el negocio, teniendo como ventajas:

- Incremento de la productividad
- Agilidad en tiempos de respuesta a problemas
- Flexibilidad en los procesos
- Fortalecimiento y optimización de recursos

El ciclo de vida de un BPM es el siguiente:

- **Diseño del proceso:**

Aquí se comprenden las necesidades del usuario.

- **Modelado del proceso:**

En esta etapa se modela un proceso del negocio.

- **Ejecución del proceso:**

En esta etapa se automatiza un proceso a través de una herramienta BPM.

- **Monitoreo del proceso:**

Aquí se da seguimiento a los procesos, y también se analiza la información de su ejecución.

- **Optimización del proceso:**

En este punto se recolectan los datos para la posible optimización del negocio.

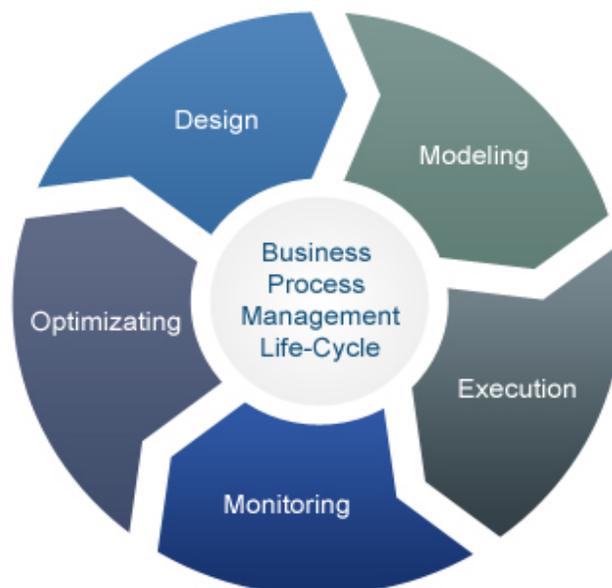


Figura 1. Ciclo de vida BPM

Tomado de (Uribe, C., Bello, J., Núñez, O. , 2012).

## 1.5 Arquitectura MVC

Es un tipo de arquitectura de software separa los datos y transacciones complejas de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes diferentes, los cuales son:

- **Modelo:**

El modelo representa la información manejada por la aplicación, en otras palabras el modelo son los datos que el sistema provee al usuario o a la aplicación.

- **Vista:**

La vista es la representación gráfica que ve e interactúa con el usuario.

- **Controlador:**

El controlador es el encargado de manejar y responder solicitudes que el usuario realiza, procesando la información y datos necesarios, modificando el modelo en caso de que se necesite.

### 1.5.1 Ciclo de vida MVC

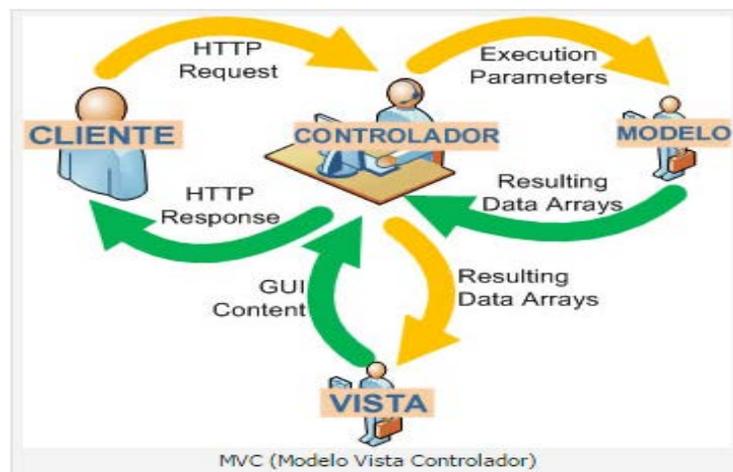


Figura 2. Ciclo de vida MVC

Tomado de (Quijano, J. , 2011).

Este ciclo de vida inicia cuando el usuario realiza una solicitud al controlador, con información que el usuario desea realizar, para así entonces el controlador delegue la tarea hacia el modelo que es el encargado de realizar operaciones sobre la información que le corresponde, para así devolver la información al controlador, para finalmente retornar la información hacia la vista, la cual interpreta ese resultado en forma visual.

### **1.5.2 Ventajas y desventajas**

Las principales ventajas son las siguientes:

- Independencia de los datos.
- Vistas siempre actualizadas, gracias al modelo que realiza este proceso automático.
- Las vistas pueden ser modificadas sin que estas afecten directamente al modelo.
- El patrón MVC es flexible en cuanto a cambios, por lo que lo hacen mantenible, modificable y extensible en comparación con otros patrones similares.

**Algunas desventajas son las siguientes:**

- Mayor tiempo de desarrollo inicial, ya que se deben construir mayor número de clases.

- Obligatoriamente se debe seguir el paradigma orientado a objetos por lo que su implementación es más costosa y difícil en comparación a otros lenguajes que no siguen este paradigma.

## 1.6 Lenguajes de programación

El lenguaje base del lado del servidor para Bonita BPM es Java, y “como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación. El paradigma de programación del lenguaje Java se basa en el concepto de programación orientada a objetos (OOP)” (Steven, 2012).

Algunas de las principales características del lenguaje Java son:

- Lenguaje orientado a objetos
- Fuertemente tipado
- Soporte de concurrencia multi-hilo
- Portable
- Dinámico
- Alto rendimiento

Bonita BPM está compuesta de 3 partes principales:

- **Bonita Studio:** el cual permite al usuario modificar gráficamente los procesos de negocio siguiendo el estándar BPMN, además de tener la capacidad de conectar los procesos a otros sistemas de información tales como servicios de mensajería, ERP, ECM, bases de datos para generar una aplicación de negocios autónoma accesible como formulario web.

- **Bonita BPM Engine:** El cual es una JAVA API que permite al usuario interactuar programáticamente con el proceso o los procesos.
- **Bonita User Experience:** es un portal web que permite a cada usuario gestionar todas sus tareas y procesos en las cuales está involucrado.

Como lenguaje de programación del lado del cliente, Bonita BPM funciona a través de la tecnología de framework de javascript AngularJs, para la creación de aplicaciones dinámicas y potentes.

## 1.7 Integración de sistemas

### 1.7.1 Concepto de integración

La idea de realizar una integración de sistemas, es hacer que “aplicaciones separadas trabajen juntas para producir un conjunto unificado de funcionalidad” (Wesley A. 2011, p 63). Estas aplicaciones no necesitan estar necesariamente en el mismo lugar, sino pueden estar dispersas geográficamente en cualquier parte del mundo, además de que pueden ser ejecutadas en múltiples plataformas, haciendo que la integración de los sistemas sea una tarea difícil.

### 1.7.2 Tipos de integración

Existen muchos tipos de integración de aplicaciones, y cada uno puede ser mejor que otro, esto dependiendo de las necesidades y las tecnologías

disponibles por una empresa. Estos enfoques se pueden resumir en cuatro principales estilos de integración:

El primero es la transferencia de archivos, en el cual cada aplicación genera y comparte archivos de datos para que otra aplicación lea esos archivos y los consuma.

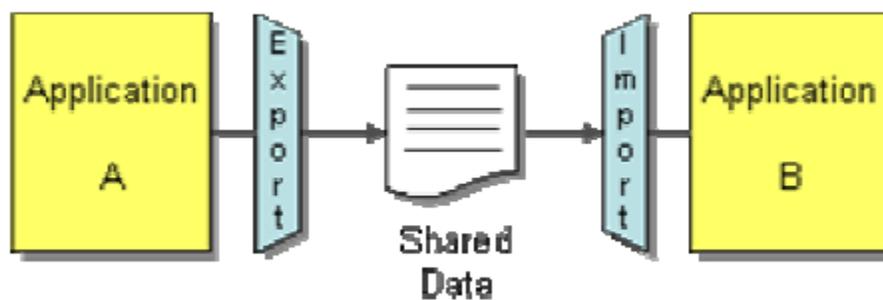


Figura 3. Esquema de integración por transferencia de archivos

Tomado de (Wesley, 2011, p. 66)

#### **Ventajas:**

- Simplicidad de integración.
- Rapidez en la implementación.
- Independencia de la plataforma.

#### **Desventajas:**

- Lapsos de desincronización.
- Información no siempre actualizada.
- Sincronización frecuente de los archivos.
- Pérdida de integridad de los datos.

- Seguridad mínima.

El segundo tipo de integración son las Bases de datos compartidas, este tipo de integración se comparte la base de datos en común entre las dos aplicaciones. Aquí cada aplicación dispone de su propia base de datos, y la sincronización de datos es mucho más rápida que en la transferencia de archivos, por lo tanto el rango de errores posibles es bastante reducido.

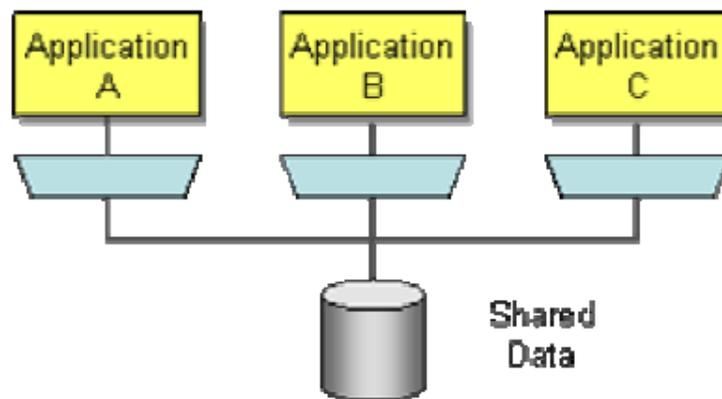


Figura 4. Esquema de integración por bases de datos compartidas

Tomado de (Wesley, 2011, p. 69)

#### **Ventajas:**

- Rapidez de consulta.
- Lapso de actualización pequeño.
- Pocos errores.
- Formato de consulta SQL universal de la bases de datos.

**Desventajas:**

- Disonancia semántica debido al uso compartido de la misma base de datos.
- Diseño de base de datos difícil de implementar.
- Dificultad en compartir la misma base de datos entre distintas organizaciones.
- Seguridad escasa.

El tercer tipo de integración es la invocación a un procedimiento remoto en la cual no se toma solo en cuenta al intercambio de datos entre aplicaciones, las cuales pueden ser realizadas por la transferencia de archivos o por la base de datos compartida, lo cual no es en muchas ocasiones suficiente. Muchas veces se requiere un “mecanismo para una aplicación pueda invocar una función en otra aplicación, pasar los datos que necesitan ser compartidos y la invocación de la función le dice al receptor aplicación la forma de procesar los datos” (Wesley A, 2011, p71).

La invocación a un procedimiento remoto aplica en principio de encapsulamiento en la integración de aplicaciones, manteniendo la integridad de los datos en cada aplicación. Existen diversas llamadas a procedimientos remotos dependiendo de los diversos lenguajes de programación, estos se los realiza con los estándares SOAP y XML, que funcionan fácilmente a través del protocolo HTTP.

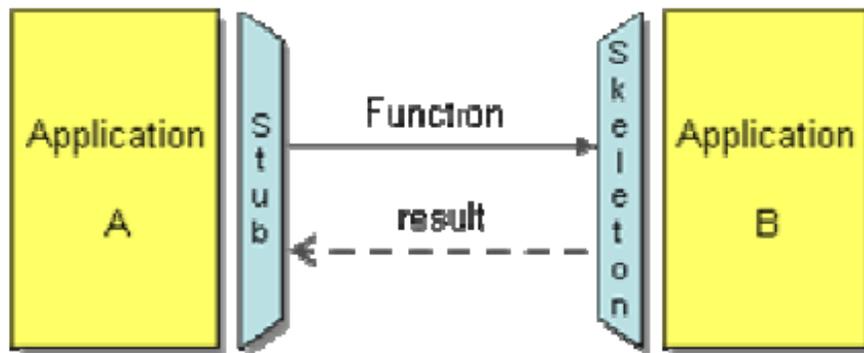


Figura 5. Esquema de integración por RCP

Tomado de (Wesley, 2011, p. 71)

#### **Ventajas:**

- Independencia de la plataforma.
- Funcionan en Internet a través de los servicios web.
- Integridad de los datos.
- Independencia de bases de datos.

#### **Desventajas:**

- La llamada del RPC no siempre responde.
- Si la llamada al RPC es remota puede tardar mucho en ocasiones
- Dependencia mutua entre las dos aplicaciones

El ultimo tipo de integración es la integración por mensajería, en la cual se tiene “cada aplicación conectada a un sistema común de mensajería, intercambio de datos e invocación de comportamiento usando mensajes” (Wesley, A. ,2011, p. 65). La mensajería asíncrona es fundamentalmente una reacción pragmática a los problemas de distribución

sistemas. El envío de un mensaje no requiere de ambos sistemas para estar arriba y listos al mismo tiempo.

Mediante el envío de mensajes pequeños con frecuencia, también permiten a las aplicaciones colaborar conductualmente así como compartir datos.

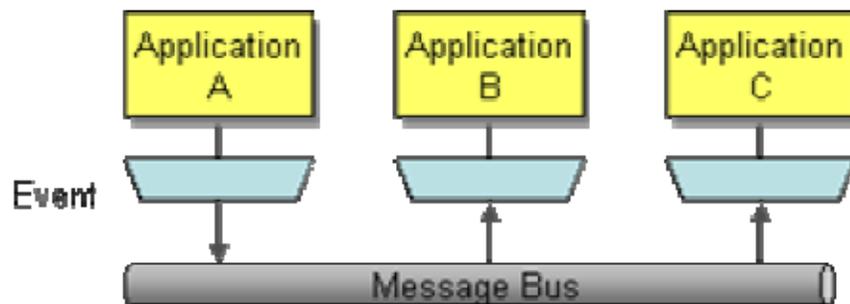


Figura 6. Esquema de integración por mensajería

Tomado de (Wesley, 2011, p. 73)

#### **Ventajas:**

- Respuesta rápida.
- Pocos errores en su ejecución.
- Pocos problemas de inconsistencia.
- Ejecución asíncrona.

#### **Desventajas:**

- No hay como enviar paquetes de datos.
- Conocimiento exacto del receptor y remitente para que no se en rute mal la información.
- El emisor y el receptor deben ponerse de acuerdo acerca del formato de datos

## 1.8 Conclusiones parciales del capítulo

Para el desarrollo del presente trabajo de titulación se utilizara como lenguaje de programación del lado del servidor a Bonita BPM basado en JAVA, ya que es libre y no tendrá ningún costo para la compañía. En base a la selección IDE de desarrollo, los lenguajes de programación de lado del cliente son JQuery, y AngularJs, los cuales permitirán interactuar de manera más intuitiva con los documentos HTML.

Se utilizará la metodología de desarrollo ágil SCRUM, debido a que se tiene ya se tiene conocimiento del funcionamiento de esta metodología. Como motor de base de datos se utilizará a SQL server, ya que este es un estándar de la compañía.

Para la integración requerida con el sistema SQUARENET, se utilizara el tipo de integración por transferencia de archivos, ya que es la que mejor se adapta al tipo de desarrollo que se va a realizar, esto debido a que el sistema SQUARENET, ya cuenta con un módulo para leer y cargar archivos para el pago de nómina, por lo que no se debe realizar ningún otro desarrollo por parte del proveedor del sistema de nómina.

## 2 Capítulo II. Análisis y diseño del sistema

### 2.1 Scrum

El objetivo de esta parte es colocar los datos y herramientas utilizados en el desarrollo de este trabajo de titulación, siguiendo la metodología ágil Scrum.

#### 2.1.1 Datos generales

Tabla 2.

Datos de las iteraciones del proyecto

Datos de Sprints	
Número de Sprints	36
Número real de Sprints	38
Duración de cada Sprint	7 días
Horas de cada Sprint	40

Se decidió realizar cada sprint con una duración de tan solo 7 días, aunque lo ideal según la metodología es de 14 a 30 días. Esto se lo decidió debido a que las reuniones con el Scrum master y con el cliente se las realizaron semanalmente durante todo el desarrollo.

Tabla 3.

## Roles Scrum

Asignación de roles	
Rol	Encargado
Product Owner	Augusto Yépez
Scrum Master	Santiago Villarreal
Desarrollador	Augusto Yépez

**2.1.2 Historias de usuario**

Cada historia de usuario corresponde a cada tarea en el Product Backlog y se detallan a continuación las primeras tareas del sprint 1.

**Reunión de planeación del sprint 1:**

El primer día antes del primer sprint se realizó la selección de las historias de usuario, según la prioridad dada por el cliente. En conjunto con el Scrum master, se seleccionaron las tareas necesarias para completar el sprint, según las horas disponibles y carga del desarrollador. Como resultado de esta planeación se sacaron las cuatro primeras historias de usuario, para realizarlas en el primer sprint de 7 días.

**Objetivo del sprint 1:**

El objetivo del primer sprint fue tener instalado y configurado Bonita BPM y su base de datos SQL Server.

Tabla 4.

Instalación de Bonita BPM en ambiente de desarrollo

Historia de Usuario		
<b>Numero:</b> HI-001	<b>Rol:</b> Programador	<b>Fecha:</b> 24/07/2015
<b>Nombre de la historia:</b> Instalación de Bonita BPM en ambiente de desarrollo		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 2	
<b>Programador responsable:</b> Augusto Yépez		
<b>Descripción:</b> Instalación de Bonita BPM en el ambiente de desarrollo local.		
<b>Validación:</b> Se instaló Bonita BPM localmente.		

Tabla 5.

Configuración de Bonita BPM

Historia de Usuario		
<b>Numero:</b> HI-002	<b>Rol:</b> Programador	<b>Fecha:</b> 27/07/2015
<b>Nombre de la historia:</b> Configuración de Bonita BPM		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 3	
<b>Programador responsable:</b> Augusto Yépez		
<b>Descripción:</b> Configurar el IDE Bonita BPM para que funcione en el ambiente de desarrollo.		
<b>Validación:</b> Se configuró a Bonita BPM para su funcionamiento local.		

Tabla 6.

Configuración de la base de datos

Historia de Usuario		
<b>Numero:</b> HI-003	<b>Rol:</b> Programador	<b>Fecha:</b> 28/07/2015
<b>Nombre de la historia:</b> Configuración de la base de datos SQL Server		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 1	
<b>Programador responsable:</b> Augusto Yépez		
<b>Descripción:</b> Aquí se pretende realizar la configuración de la Base de datos SQL server, para que funcione con Bonita BPM.		
<b>Validación:</b> Se verificó que la base fue configurada satisfactoriamente.		

Tabla 7.

Pruebas del ambiente de desarrollo

Historia de Usuario		
<b>Numero:</b> HI-004	<b>Rol:</b> Programador	<b>Fecha:</b> 29/07/2015
<b>Nombre de la historia:</b> Pruebas de ambiente de desarrollo		
<b>Prioridad en negocio:</b> 1	<b>Estimación:</b> 1	
<b>Programador responsable:</b> Augusto Yépez		
<b>Descripción:</b> Aquí se realizaran las primeras pruebas del ambiente de desarrollo.		
<b>Validación:</b> Se realizó la prueba de un login exitoso y creación de un usuario de pruebas en el ambiente de desarrollo.		

### 2.1.3 Product Backlog

La siguiente tabla muestra solo un fragmento del Product Backlog, para visualizarlo completo se puede ir al Anexo 1.

Tabla 8.

Muestra de las primeras 7 historias de usuario del Product Backlog.

A continuación se muestran las 7 historias de usuario, que corresponden a los dos primeros sprints, del desarrollo.

ID de Historia	Nombre de la Historia	Fecha de Culminación	Duración (Días)	Sprint	Comentarios
HI-001	Instalación de Bonita BPM en ambiente de desarrollo.	24/07/2015	2	1	
HI-002	Configuración de Bonita BPM.	27/07/2015	3	1	
HI-003	Configuración de la base de datos SQL Server.	28/07/2015	1	1	
HI-004	Pruebas de ambiente de desarrollo.	29/07/2015	1	1	
HI-005	Reunión con encargado de talento humano para	30/07/2015	1	2	

	entender el proceso.				
HI-006	Diseño del modelo Entidad Relación	03/08/2015	4	2	
HI-007	Creación del diagrama físico de base de datos.	05/08/2015	2	2	

#### 2.1.4 Iteraciones

En el presente Trabajo de Titulación se realizaron un total de 38 Sprints, a continuación se presenta solo un sprint. El resto de iteraciones se los encontrara en el Anexo 2.

Tabla 9.

#### Sprint 1

Fecha de inicio	Fecha de finalización
23/jul/2015	29/jul/2015

Tarea	Responsable	Estado	Horas Planeadas	Horas Realizadas	Fecha de culminación
Instalación de Bonita BPM en ambiente de desarrollo	Augusto Yépez	Realizado	6	8	24/07/2015

Configuración de Bonita BPM	Augusto Yépez	Realizado	8	8	27/07/2015
Configuración de la base de datos SQL Server	Augusto Yépez	Realizado	4	6	28/07/2015
Pruebas de ambiente de desarrollo	Augusto Yépez	Realizado	2	5	29/07/2015

## 2.2 Diagramas de Casos de Uso

Para el primer caso de uso el cual es *Generar Organizador de Personal*, existe un primer sub flujo que se encarga de esta primera tarea, la cual servirá como punto de partida para el flujo de *Registrar Actividades Diarias*.

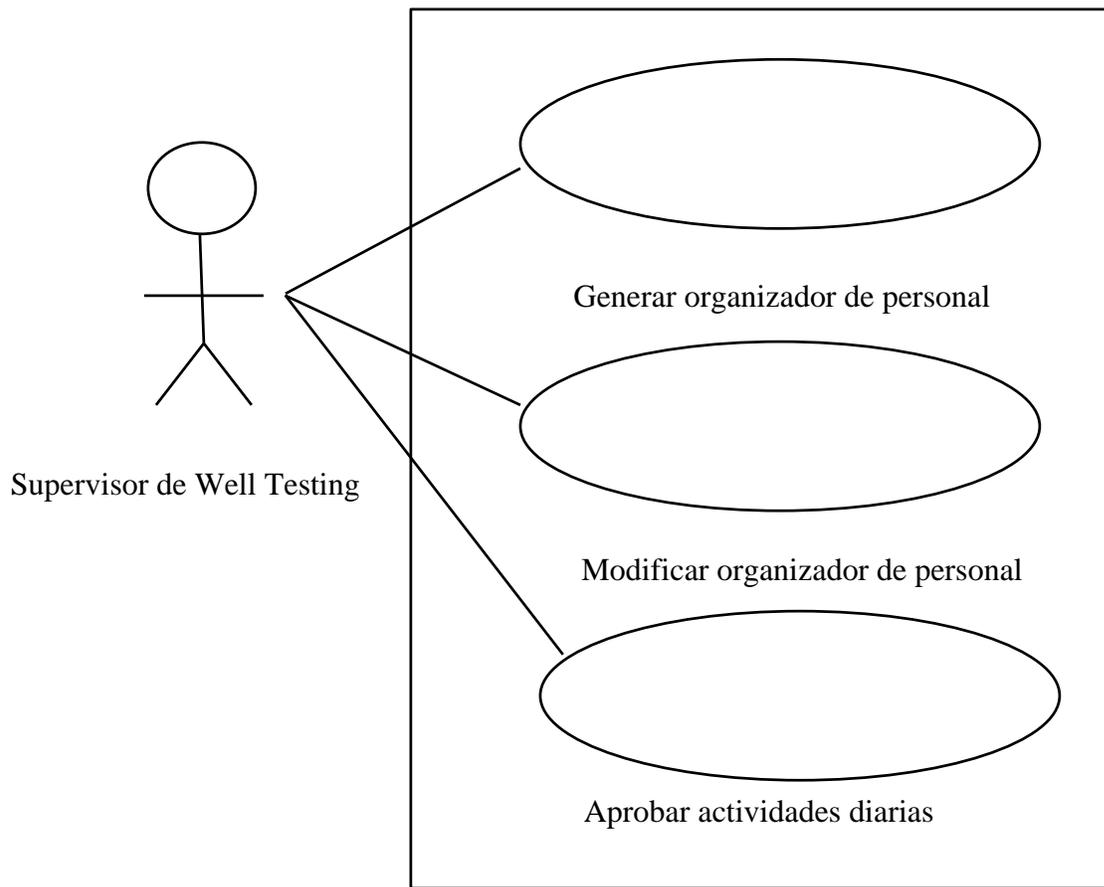


Figura 7. Caso de uso generar organizador de personal

Tabla 10.

Caso de uso generar organizador de personal

CU-001	Generar organizador de personal
Descripción	Crear el organizador diario de personal
Actores	Supervisor de Well Testing
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El supervisor de Well Testing ingresa al sistema.</li> <li>2. El supervisor de Well Testing da clic en Flujos, para iniciar.</li> <li>3. El muestra la primera pantalla que dice si se va a realizar una creación o modificación.</li> </ol>

	<p>4.1 Si el supervisor selecciona nueva creación del organizador, el sistema genera un nuevo registro por cada MTU activa.</p> <p>4.2 El supervisor selecciona la fecha del organizador.</p> <p>4.3 El supervisor llena todos los campos y selecciona el pozo, las personas encargadas del turno de la mañana, noche y cambio de turno.</p> <p>4.4 El supervisor da clic en guardar.</p> <p>4.5 El sistema valida que la fecha no está ya registrada.</p> <p>4.6 El sistema valida que las mismas personas no se repitan en el turno del día, noche y cambio de turno.</p> <p>4.7 El sistema valida que las personas no se repitan el otras MTU's.</p> <p>4.8 Si pasa todas las validaciones el sistema guarda los datos.</p>
Excepciones	Un mismo colaborador si puede constar en la turno del día y noche, pero no en el cambio de turno a la vez.

Tabla 11.

## Caso de uso modificar organizador de personal

CU-002	Modificar organizador de personal
Descripción	Modificar el organizador de personal
Actores	Supervisor de Well Testing
Secuencia Normal	<p>1. El supervisor de Well Testing ingresa al sistema.</p> <p>2. El supervisor de Well Testing da clic en Flujos, para iniciar.</p> <p>3. El muestra la primera pantalla que dice si se va a realizar una creación o modificación.</p> <p>4.1 Si el supervisor selecciona modificación del organizador, el sistema lee los datos del organizador de la fecha actual.</p> <p>4.2 La fecha del organizador se presentara bloqueada.</p> <p>4.3 El supervisor modifica la información requerida.</p> <p>4.4 El supervisor da clic en guardar.</p>

	<p>4.5 El sistema valida que las mismas personas no se repitan en el turno del día, noche y cambio de turno.</p> <p>4.6 El sistema valida que las personas no se repitan el otras MTU's.</p> <p>4.7 Si pasa todas las validaciones el sistema guarda los datos.</p>
Excepciones	<p>Un mismo colaborador si puede constar en la turno del día y noche, pero no en el cambio de turno a la vez.</p> <p>Solo se puede modificar el organizador el mismo día de su fecha de creación.</p> <p>El supervisor no podrá cambiar la fecha del organizador en caso de modificación del mismo.</p>

Para el segundo flujo que es *Registrar Actividades Diarias*, se tiene un primer nivel de aprobación por parte de los supervisores.

Tabla 12.

Caso de uso aprobar actividades diarias

CU-003	Aprobar actividades diarias
Descripción	Aprobación del formulario de registro de actividades diarias
Actores	Supervisor de operaciones
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El supervisor de operaciones ingresa al sistema.</li> <li>2. El supervisor da clic en su bandeja de entrada y abre la actividad a aprobar.</li> <li>3. Se muestra información del colaborador no modificable y el formulario de registro de actividad diaria previamente llenada por el colaborador.</li> <li>4. El supervisor modifica la información ingresada por el colaborador en caso de ser necesario.</li> <li>5. El sistema valida que el colaborador se encuentre activo en la compañía y en sus jornada</li> </ol>

	<p>6. El sistema valida que el colaborador se encuentre activo en la compañía y en sus jornadas.</p> <p>8. Si pasan todas las validaciones el sistema continúa al siguiente paso y guarda la información ingresada.</p> <p>9. El sistema notifica vía correo al colaborador su actividad fue aprobada o negada.</p>
Excepciones	

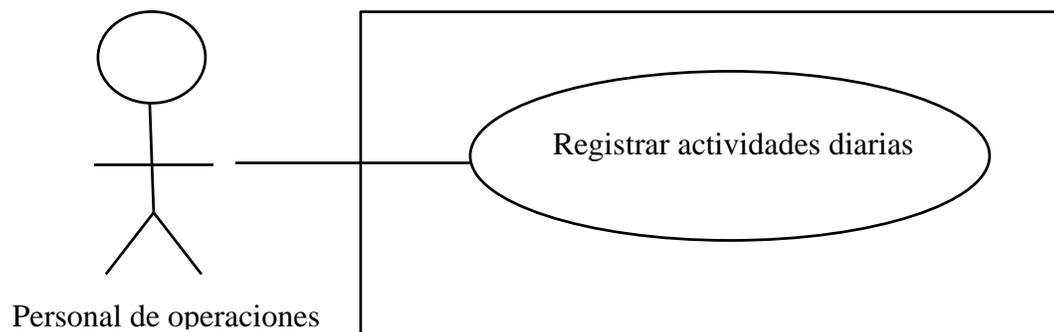


Figura 8. Registrar actividades diarias

Tabla 13.

Caso de uso registrar actividades diarias

CU-004	Registrar actividades diarias
Descripción	Formulario de registro de actividades diarias
Actores	Personal operaciones
Secuencia Normal	<p>1. El personal de operaciones ingresa al sistema.</p> <p>2. El personal de operaciones da clic en Flujos, para iniciar.</p> <p>3. Se muestra información del colaborador no modificable y un formulario de registro de actividad diaria.</p>

	<p>4. Se llenan los campos del formulario y comentarios en caso de que se desee agregar información adicional.</p> <p>5. El sistema valida que si la persona que está llenando el formulario pertenece al centro de costos de Well Testing , primero este ingresado en el organizador de personal.</p> <p>6. El sistema valida que solo exista una actividad registrada en el mismo día.</p> <p>7. El sistema valida que el colaborador se encuentre activo en la compañía y en sus jornada.</p> <p>8. Si pasan todas las validaciones el sistema continúa al siguiente paso y guarda la información ingresada.</p> <p>9. El sistema notifica vía correo al colaborador que ingreso su actividad diaria y a su supervisor.</p>
Excepciones	

Tabla 14.

## Caso de uso validar y aprobar actividades diarias

CU-005	Validar y aprobar actividades diarias
Descripción	Validación del registro de la actividad diaria por el supervisor a cargo
Actores	Supervisores de campo
Secuencia Normal	<p>1. El personal de operaciones ingresa al sistema.</p> <p>2. El personal de operaciones da clic en Flujos, para iniciar.</p> <p>3. El muestra información del colaborador no modificable y un formulario de registro de actividad diaria.</p> <p>4. Se llenan los campos del formulario y comentarios en caso de que se desee agregar información adicional.</p> <p>5. El sistema valida que si la persona que está llenando el formulario pertenece al centro de costos de Well Testing , primero este ingresado en el organizador de personal.</p>

	<p>6. El sistema valida que solo exista una actividad registrada en el mismo día.</p> <p>7. El sistema valida que el colaborador se encuentre activo en la compañía y en sus jornada.</p> <p>8. Si pasan todas las validaciones el sistema continúa al siguiente paso y guarda la información ingresada.</p> <p>9. El sistema notifica vía correo al colaborador que ingreso su actividad diaria y a su supervisor.</p>
Excepciones	

Para el tercer flujo que es *Aprobación Gerencial*, se tienen dos niveles de aprobaciones, el primer nivel corresponde a la aprobación del gerente de campo y el segundo corresponde a la aprobación del gerente de operaciones.

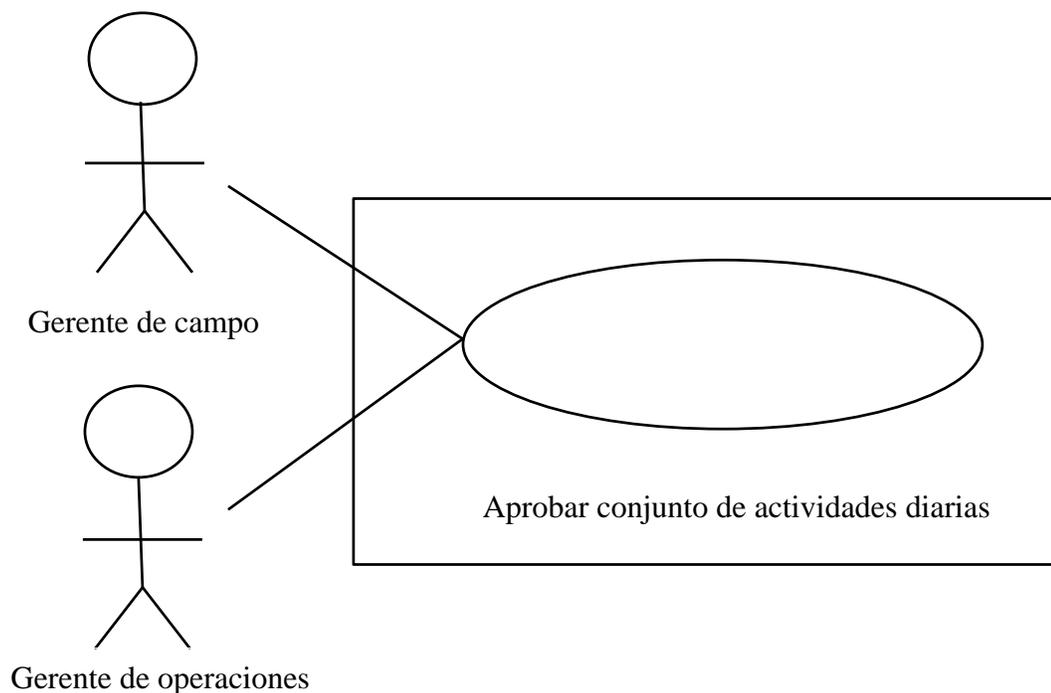


Figura 9. Aprobar conjunto de actividades diarias

Tabla 15.

## Caso de uso validar y aprobar conjunto de actividades diarias

CU-006	Validar y aprobar conjunto de actividades diarias
Descripción	Validación mensual del conjunto de registros de las actividades diarias por parte del gerente de campo.
Actores	Gerente de campo
Secuencia Normal	<ol style="list-style-type: none"> <li>1. El gerente de campo ingresa al sistema.</li> <li>2. El gerente de campo da clic en Flujos, para iniciar.</li> <li>3. Se muestra el detalle mensual de las actividades diarias por colaborador.</li> <li>4. Se selecciona(n) si se aprueban todas las actividades o si se rechazan algunas.</li> <li>5. El sistema notifica que si se está seguro de no aprobar todas las actividades en caso de no aprobar alguna.</li> <li>8. Se guarda la información ingresada.</li> <li>9. El sistema notifica vía correo al gerente de campo y operaciones que aprobaron las actividades mensuales de los colaboradores.</li> </ol>
Excepciones	

## 2.3 Diagrama Entidad Relación

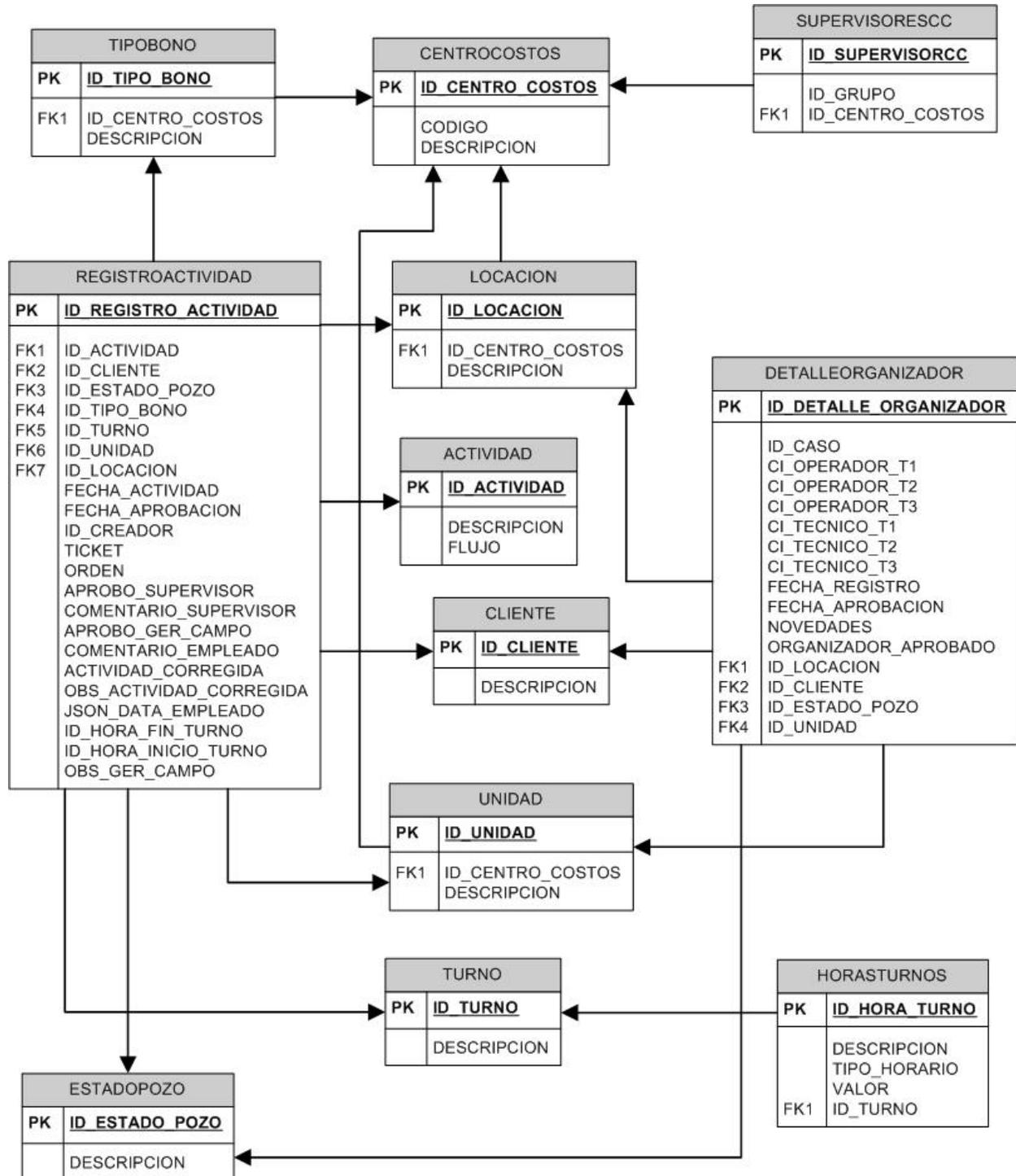


Figura 10. Diagrama Entidad Relación

## 2.4 Análisis y diseño de la integración con sistema Squarenet

Para el intercambio de información entre el sistema de pagos adicionales y el sistema de pago de nómina Squarenet se contempló las distintas posibilidades de integración de sistemas disponibles, para lo cual se realizaron varias reuniones con personal de nómina que usa el sistema Squarenet, y adicional se tuvo una reunión con el personal externo de la compañía encargado de los desarrollos de módulos y administración del sistema Squarenet, y con lo cual por motivos económicos y de tiempo se definió que no se realizaría ningún desarrollo, ni creación de módulos adicionales a los ya existentes del sistema Squarenet.

A continuación se describen varios de los motivos más importantes por los cuales se tomó la decisión de realizar una integración de los dos sistemas por transferencia de archivos.

- El sistema Squarenet cuenta con un módulo ya desarrollado de lectura de archivos para carga de pagos de horas extras, horas suplementarias y pago de bonos.
- El sistema Squarenet no cuenta con módulos para realizar una integración por servicios.
- La sincronización y carga de la información hacia Squarenet solo se la realiza una vez al mes.
- El sistema Squarenet es un sistema externo a la compañía, es decir no fue desarrollado por la empresa.
- No se requerirá realizar ninguna solicitud de cambio o creación de un nuevo módulo para la integración.
- La generación del archivo por el sistema de pago de nómina se lo puede hacer sin mayor dificultad.
- No se gastará dinero ni tiempo en el desarrollo de nuevos módulos.
- La base de datos del sistema Squarenet no debe ser modificada.

### 2.4.1 Diagrama de integración

La figura 8 describe el diagrama de integración con el sistema Squarenet, en donde el sistema de pagos adicionales genera un archivo plano en formato .CSV, el cual es primero leído de la base de datos, para su posterior proceso por el sistema de pagos adicionales, y finalmente es exportado a una ruta determinada, para que de ahí pueda ser leída e importado por el sistema Squarenet, y guardado en base de datos para la generación de los pagos para los colaboradores.

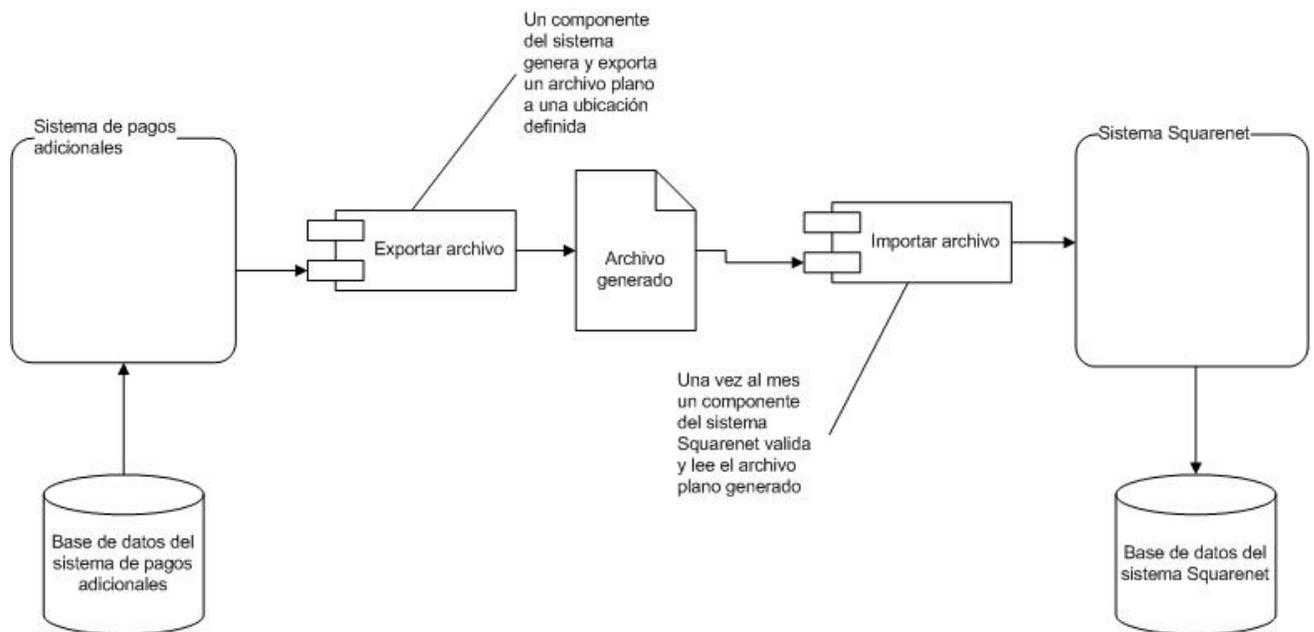


Figura 11. Diagrama de integración con sistema Squarenet

## 2.5 Análisis e integración unidireccional con sistema biométrico

Para esta integración unidireccional se validó mediante reuniones con personal de talento humano y personal del sistema biométrico, que una integración por base de datos sería la más factible, ya que el sistema biométrico no cuenta con ningún

componente que permita realizar consultas del personal que se registró por el biométrico.

Con el único fin de restringir el ingreso de actividades diarias al personal de campo, se realizó esta integración, para validar que primero el personal se haya registrado en el sistema biométrico, para posteriormente puedan llenar sus actividades diarias.

Los principales motivos por los que se seleccionó la integración por base de datos son los siguientes:

- El sistema biométrico no cuenta con ningún servicio web que permita realizar una consulta del personal.
- El sistema biométrico no cuenta con ningún modulo o interfaz para generar un archivo plano, necesario para una integración por archivos.
- El sistema solo cuenta con una interfaz gráfica que solo permite la visualización de los timbrajes de los colaboradores.

Por estos motivos la única opción posible para consultar si un colaborador se registró en el biométrico, es la consulta directa a la base de datos del sistema biométrico.

### **2.5.1 Diagrama de integración**

En la figura 9 se puede apreciar el diagrama de integración unidireccional con el sistema biométrico de asistencia. Lo que se realiza en esta integración es la consulta basada en el número de cedula y fecha, para posteriormente guardarla en base de datos, con el fin saber si el colaborador está registrado en el sistema biométrico, para en base a esto habilitar el ingreso de actividades diarias del colaborador.

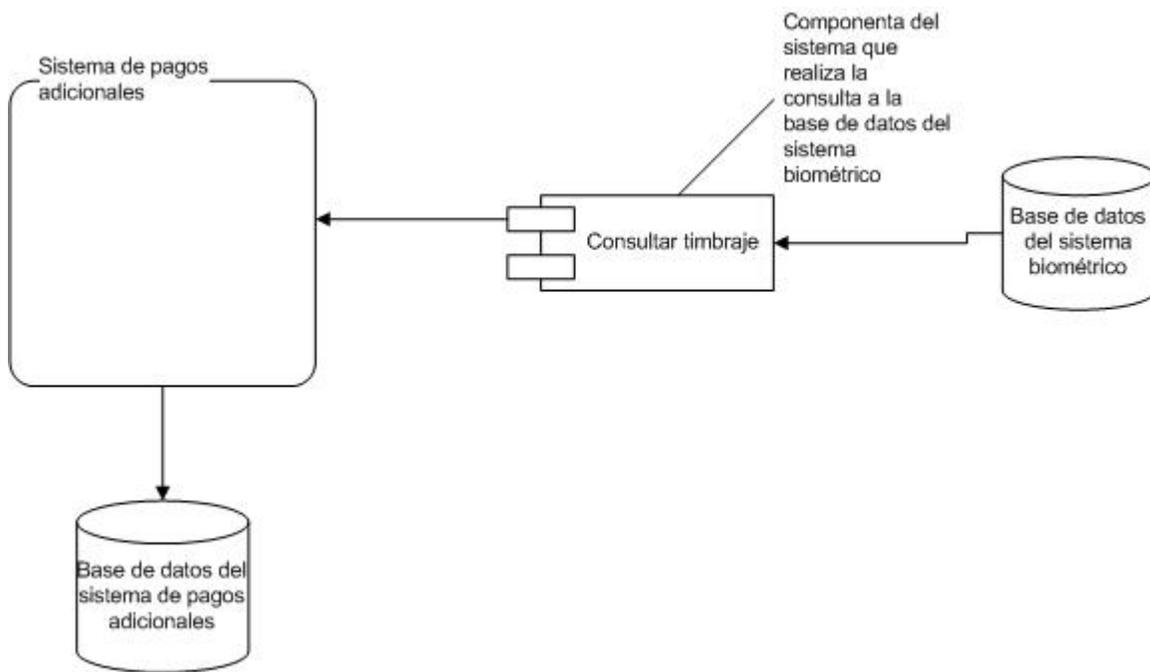


Figura 12. Diagrama de integración con sistema Biométrico

## 2.6 Arquitectura del sistema

Bonita BPM consiste en un conjunto de aplicaciones open source, las cuales permiten la automatización de procesos de negocio de una organización. En la figura 10 se representa el diagrama de arquitectura de Bonita BPM.

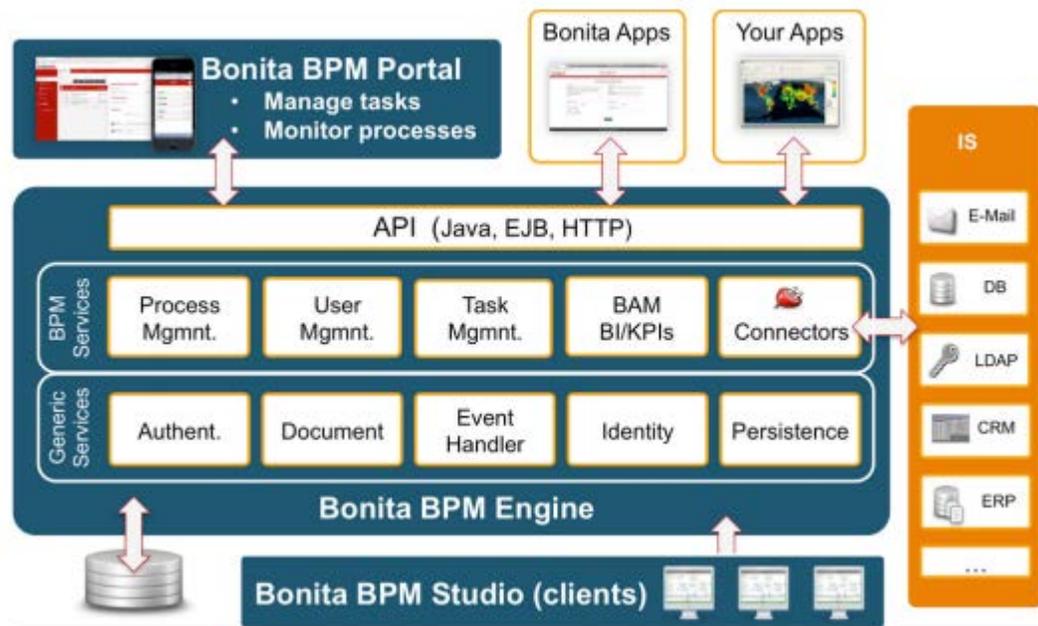


Figura 13. Diagrama de arquitectura de Bonita BPM

Tomado de (Prats, A. , 2016)

La parte principal de esta arquitectura es el Bonita Engine, la cual se representa en el centro de la figura 7, la cual está dividida en *servicios genéricos* y *servicios generales*. Dentro de los servicios generales están los módulos de autenticación, gestión documental, manejador de eventos, identidad y persistencia de información. En el segundo grupo de servicios BPM, se encuentran los módulos de administración de procesos, administración de usuarios y tareas, así también como la gestión de indicadores, y por último los conectores hacia otros sistemas.

Entre las principales características de Bonita es que cuenta con una API, la cual se tiene acceso al motor vía aplicaciones desarrolladas por Bonita o también otras aplicaciones de terceros, lo cual, hace que se adapte de manera más fácil a cualquier organización.

Bonita posee un entorno de trabajo denominado Open Studio que incluye tres componentes:

- a) Bonita BPM Studio orientado a modelar los procesos de negocio.
- b) Bonita BPM Engine dedicado a la ejecutar los procesos previamente modelados.
- c) Bonita BPM Portal utilizado por los usuarios como interfaz para acceder a los procesos y trabajar con los mismos.

La arquitectura del sistema de pagos adicionales posee tres capas, las cuales son la capa de presentación del sistema, la capa de lógica del negocio, y la capa de base de datos. El sistema cuenta con un módulo de administración y también tiene un módulo de reportes y consultas, los cuales interactúan con la base de datos y muestran o alteran datos de la base de datos del sistema, como se lo muestra en la figura 11. Toda la lógica del negocio se la realiza dentro del Motor de Bonita BPM.

A continuación se presenta un diagrama de la arquitectura del sistema de pagos adicionales:

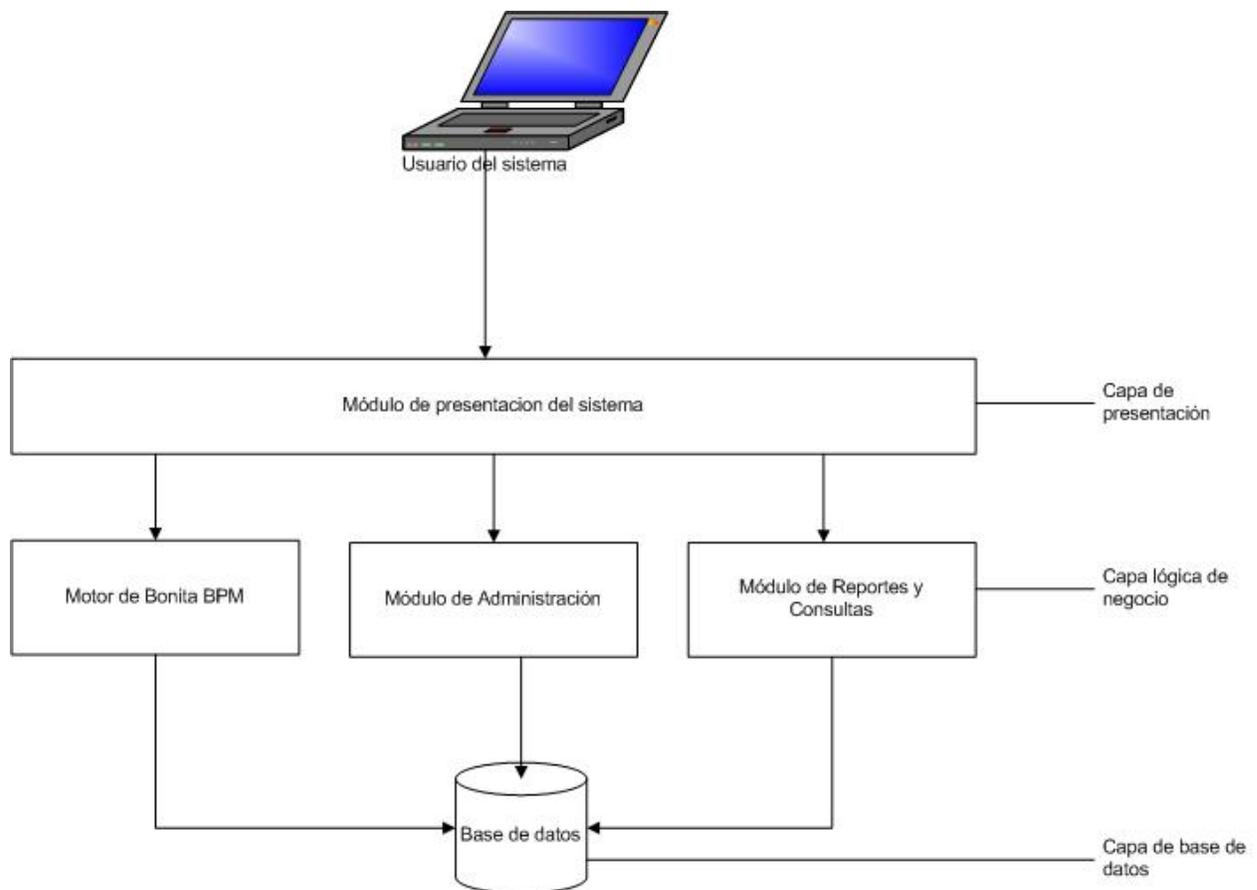


Figura 14. Diagrama de arquitectura de sistema de pagos adicionales

## 2.7 Conclusiones parciales capítulo 2

A lo largo del capítulo 2 se evidenció la utilización de la metodología ágil Scrum, la cual se pudo constatar con los objetos de Scrum como lo son la planificación del sprint, la meta del sprint. Así mismo se analizó el primer sprint cuya duración fue de 7 días, en el cual se pretendía configurar el ambiente de desarrollo.

Además en el presente capítulo se pudo apreciar el diagrama entidad relación del sistema, así también como la arquitectura que maneja Bonita BPM.

Finalmente en este capítulo se pudo apreciar los diagramas de integración, tanto con el sistema Squarenet, como con el sistema Biométrico de asistencia. Se pudo constatar que la primera integración con el sistema Squarenet, será una integración bidireccional por transferencia de archivos. En cuanto a la segunda integración, esta será una integración unidireccional, en la que básicamente el sistema consultará si el colaborador tiene un registro en el sistema biométrico, para que se le permita ingresar sus actividades diarias.

## 3 Capítulo III. Desarrollo, resultados y pruebas del sistema

### 3.1 Desarrollo del sistema

Para el desarrollo de este proyecto se utilizó la herramienta Bonita BPM Studio, la cual provee un conjunto de herramientas para la modelación y simulación de procesos y flujos de trabajo. El motor de Bonita BPM provee una serie de servicios BPM y APIs.

El desarrollo del sistema se lo llevo a cabo en un total de 38 sprints. A partir del sprint 3 se arrancó con el modelado de los tres flujos que componen el sistema. El primero de ellos es el organizador de personal, el cual es el punto de partida necesario para que los colaboradores se les habilite el ingreso de actividades diarias. En este flujo los supervisores de campo organizan diariamente al personal que ingresa a cada MTU en diversas locaciones, por lo que ingresan al sistema, y arrancan el flujo organizador de personal, en donde deben seleccionar la fecha, la locación, la unidad MTU, la compañía para la cual realizan el trabajo, la actividad en la cual se encuentra la MTU, y el personal técnico y operador a cargo en los turnos diurnos y nocturnos, como lo muestra la figura 12.

Bienvenido: HILBER SANTIAGO MELO FIALLOS | Usuario | Configuración

Tareas Casos Procesos

### Organizador de personal

Fecha: 2017-01-30

POZO	UNIDAD	COMPANIA	ACTIVIDAD	TURNO 1 OPERADOR(NOCHE)	TURNO 1 TECNICO(NOCHE)	TURNO 3 OPERADOR(DIA)
APAIIKA 11	MTU 1	PETROAMAZONAS EP	PRODUCCION	ACEVEDO VALLE HERNAN IVAN	ALCOSER SANGA JAIRO JAMIER	CARPIO OCHOA CARLOS E
APAIIKA 11	MTU 2	RIO NAPO	PRODUCCION	ALBUJA JACOME RAFAEL EDUARDO	BOHORQUEZ CUEVA ANGEL GABRIEL	ZUÑIGA PIZARRO MILTON L

Continuar

Figura 15. Organizador de personal

El desarrollo de este flujo se contempla desde el sprint 3 hasta el sprint 6, los cuales pueden ser revisados detenidamente en el anexo 2. Para el desarrollo de este flujo se hizo uso de los componentes provistos por Bonita BPM Studio. Uno de estos componentes son los conectores los cuales permiten realizar una consulta a una base de datos externa o incluso hacia la misma base de datos del mismo sistema. A continuación se presentan capturas del wizard generado por este componente, en el cual se extrae información como la cedula y los nombres de los colaboradores del sistema Squarenet de talento humano, para traerlos al sistema de pagos adicionales:

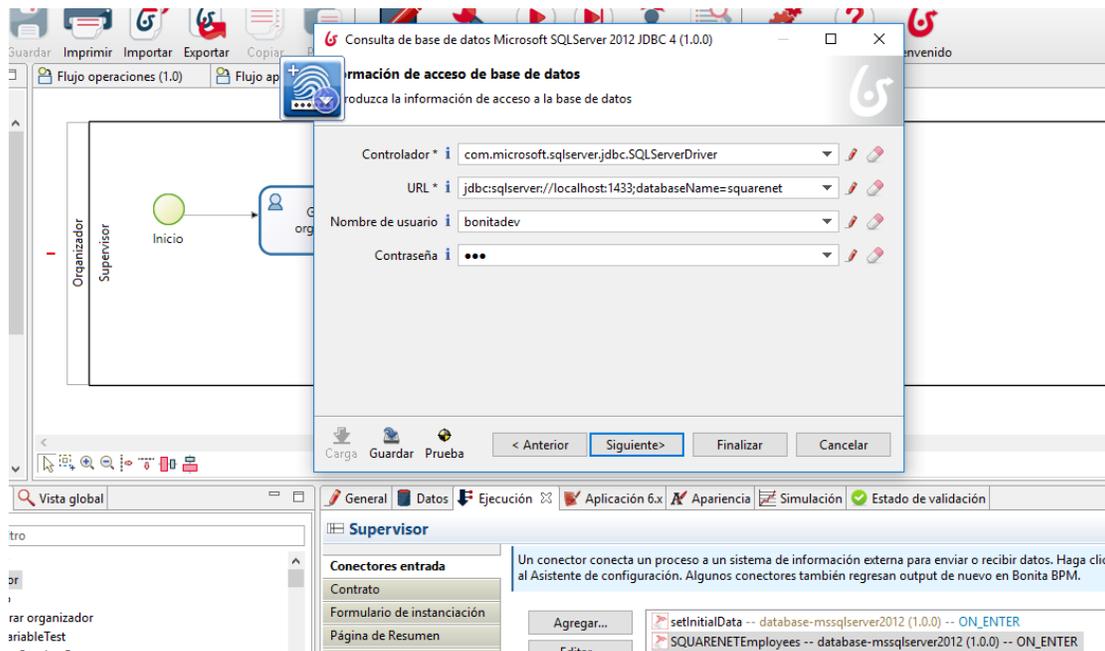


Figura 16. Wizard de conector con Squarenet flujo organizador de personal parte 1

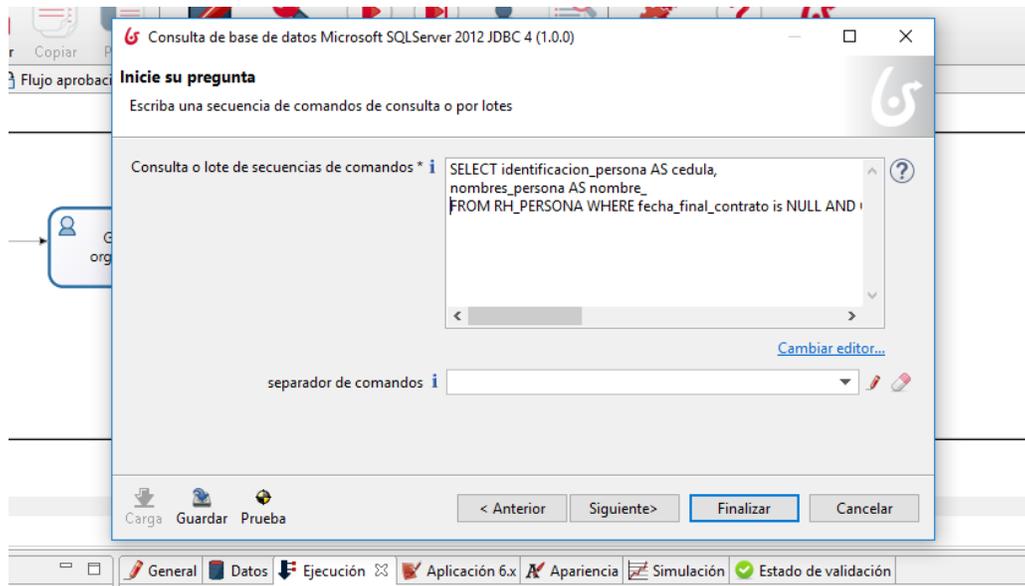


Figura 17. Wizard de conector con Squarenet flujo organizador de personal parte 2

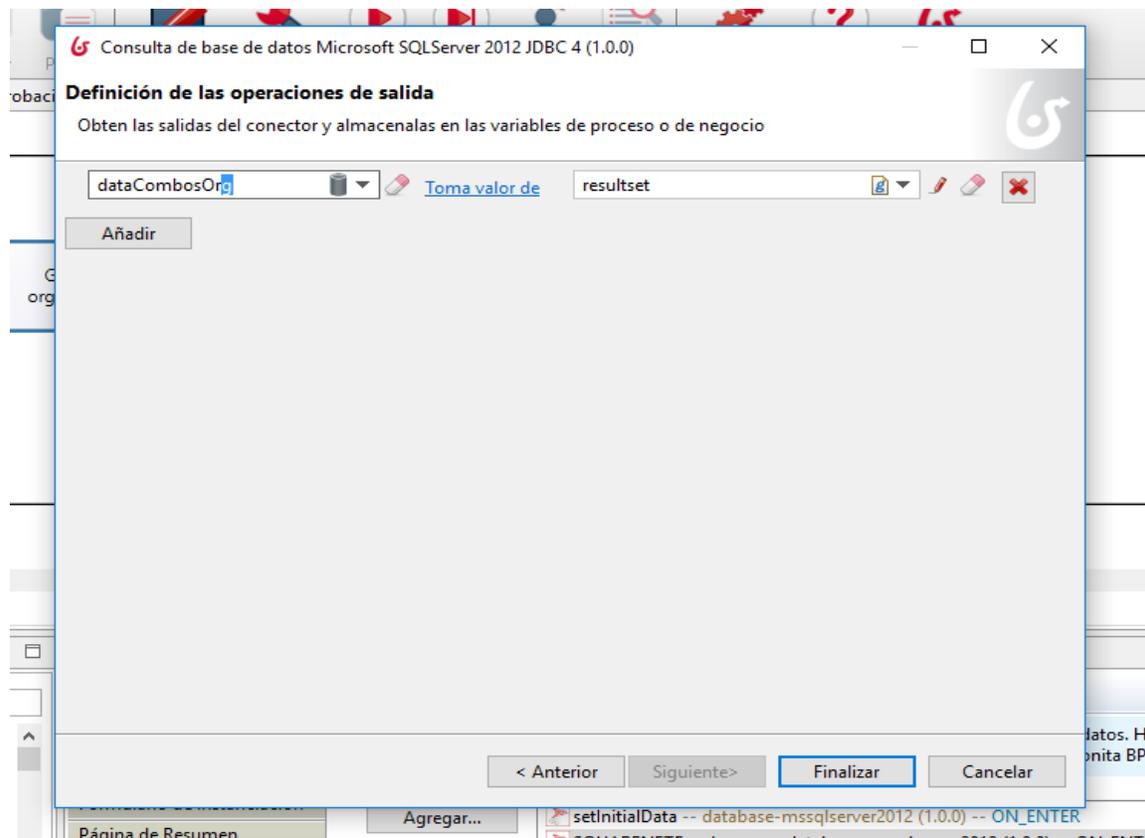


Figura 18. Wizard de conector con Squarenet flujo organizador de personal parte 3

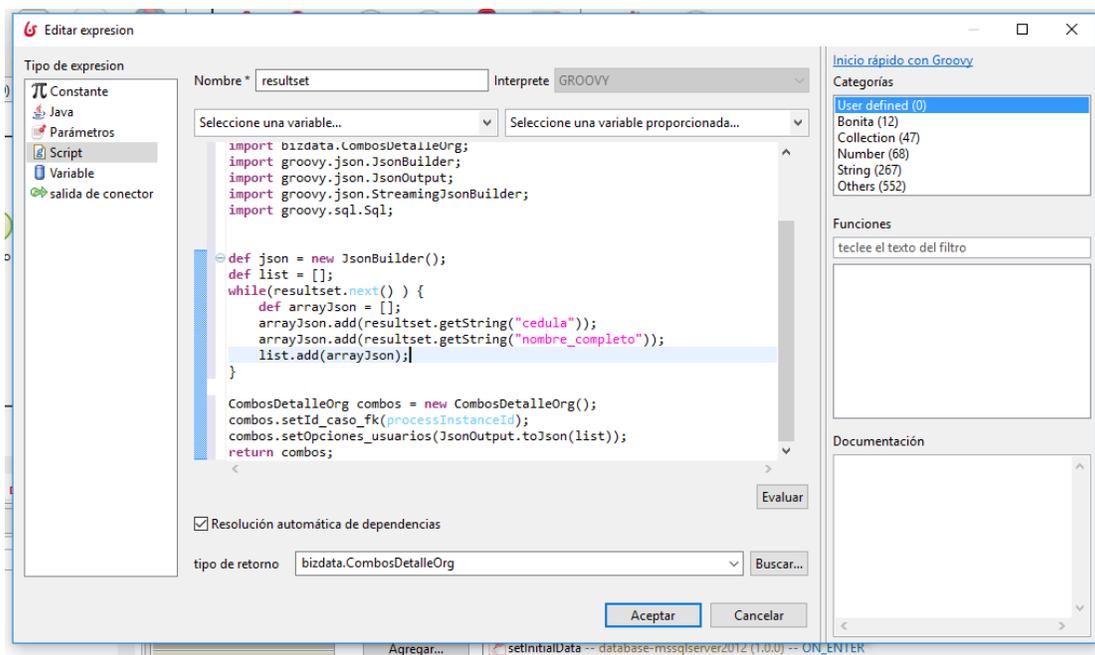


Figura 19. Wizard de conector con Squarenet flujo organizador de personal parte 4

Dentro del segundo flujo, el cual sirve para ingreso de actividades diarias de los colaboradores, se realiza la integración por base de datos compartida con el sistema biométrico de asistencia. Para esta integración se hizo uso una vez más de los conectores provistos por Bonita BPM Studio. Este conector es llamado antes de dar inicio al flujo de registro de actividades diarias, el cual extrae solo los registros nuevos que tiene el colaborador en el sistema biométrico, para así permitir o negar el ingreso de sus actividades diarias. A continuación se muestra el wizard a través del cual se trae la información del colaborador del sistema biométrico, para así llevar esta información hacia el sistema de pagos adicionales.

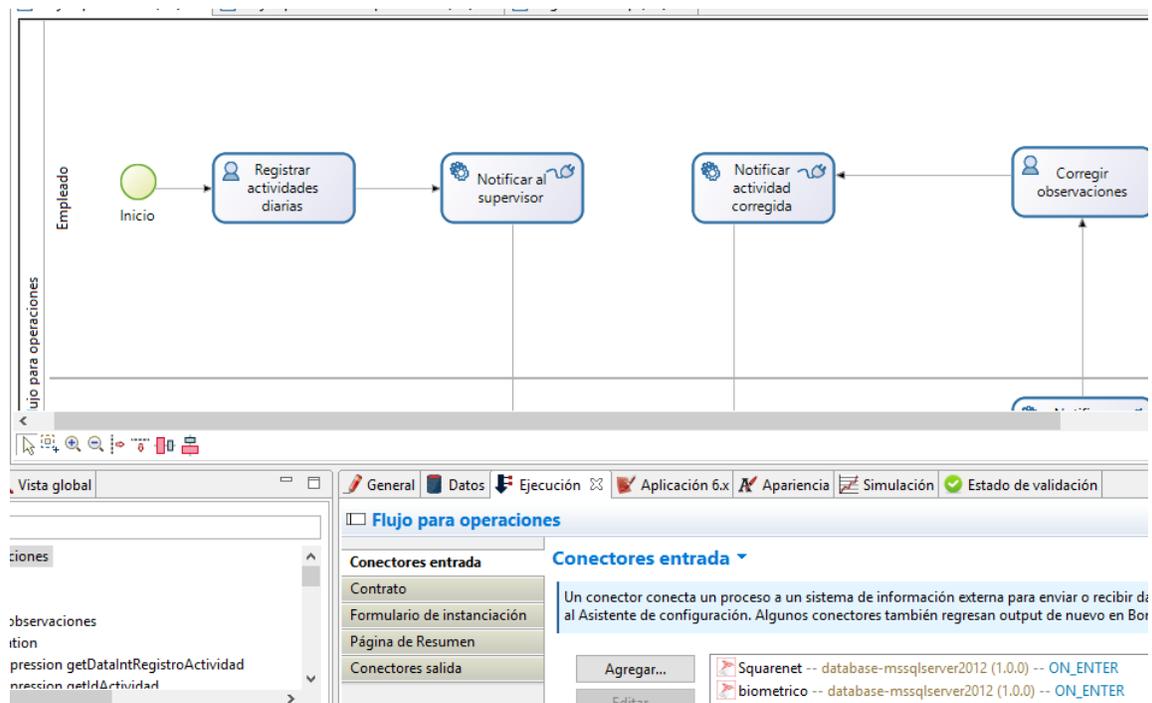


Figura 20. Flujo de ingreso de actividades diarias para operaciones y proyectos

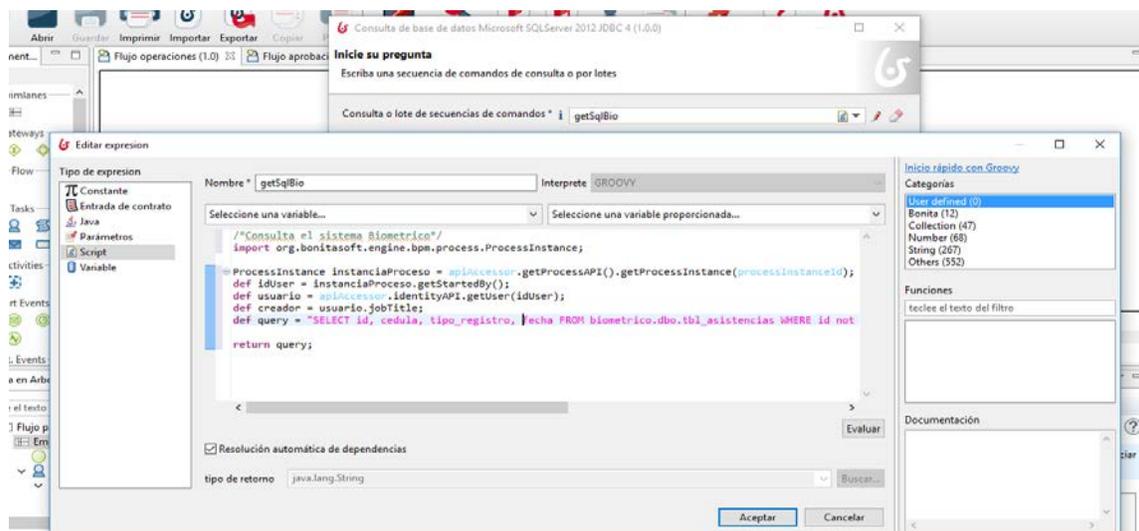


Figura 21. Wizard integración con sistema Biométrico, flujo de ingreso de actividades diarias parte 1

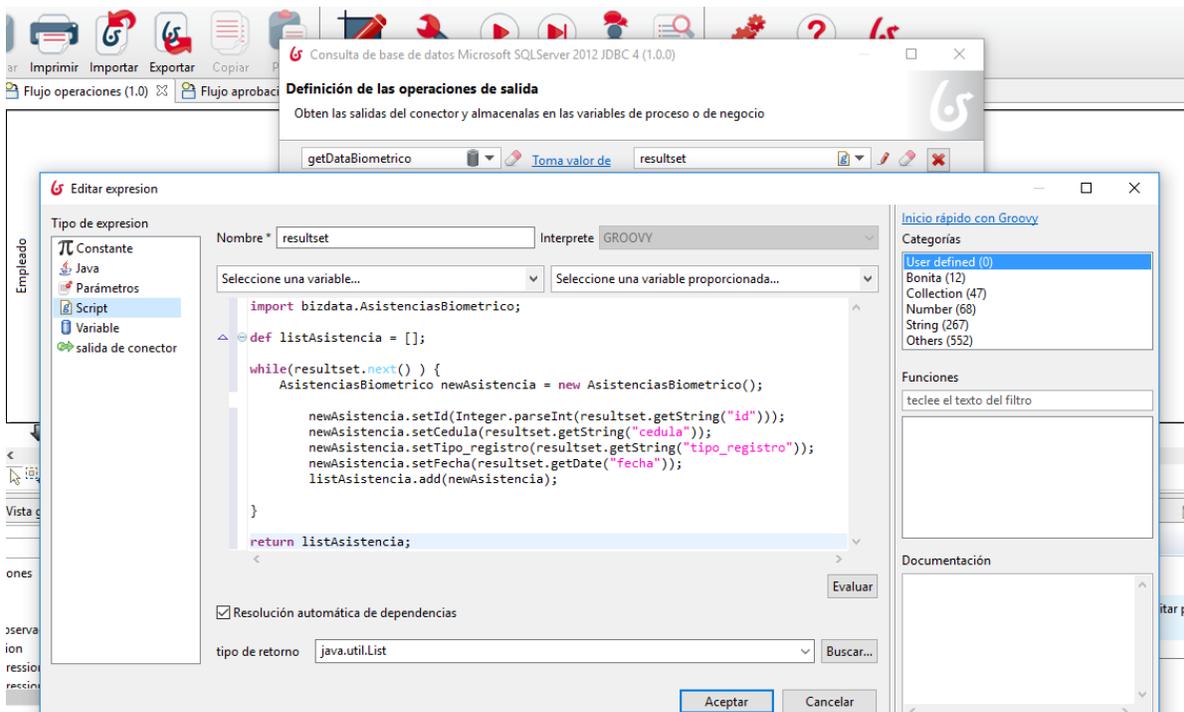


Figura 22. Wizard integración con sistema Biométrico, flujo de ingreso de actividades diarias parte 2

Una vez que se tiene la información localmente en la base de datos, lo que se realizó es la implementación de un widget personalizado para el botón continuar en el formulario de ingreso de actividades diarias. Aquí se realiza la consulta que valida si el colaborador está habilitado o no para realizar el ingreso de su actividad diaria en la fecha que el colaborador seleccionó. Así mismo también se realizan varias validaciones que le permitirán al colaborador continuar o no con el proceso.

Bonitasoft Bienvenido: FABU

Tareas Casos Procesos

Fecha ingreso: 2016-07-12 00:00:00

Centro costos: WELL TESTING

Supervisor a cargo: Seleccione una opción

Fecha salida:

Carga: OPERADOR DE WELL TESTING

Sucursal: Quito

Registro de actividades

Actividad: TURNO Estado: PRODUCCION

Turno: TURNO 1 Hora de inicio: 19:00 Hora de fin: 19:00

Bono: B Nro. Unidad: MTU 1

Locacion: APAIKA 1 Cliente: PETROAMAZONAS EP

Requerimiento\*

Fecha actividad: 2017-01-30

Comentario

Continuar

Figura 23. Formulario de registro de actividades diarias

Controller

```

23  */
24  function doRequest(method, url, params) {
25
26      var notificationWidget = $("#notification").kendoNotification().data("kendoNotification");
27      notificationWidget.options.position.top=0;
28      notificationWidget.options.position.bottom=null;
29      notificationWidget.options.position.right=null;
30      notificationWidget.options.width="100%";
31      $scope.method=method;
32      $scope.url=url;
33      $scope.params=params;
34      var centroCostos = $scope.properties.codCentroCostos;
35      if(centroCostos == "02010101"){
36          if($scope.properties.dataToSend.fecha_actividad == undefined){
37              $scope.properties.dataToSend.fecha_actividad=$("#kendo_datapicker").val();
38          }
39
40          if(typeof($scope.properties.dataToSend.fecha_actividad) != "string" ){
41              var fechaAct = $scope.properties.dataToSend.fecha_actividad;
42          }else{
43
44

```

Propiedades

+ Crear

textoBoton

Etiqueta: Texto del boton

Tratar como Valor dinámico

Tipo: text

Valor predeterminado: Continuar

targetUrlOnSuccess

Etiqueta: Url retorno

Tratar como Valor dinámico

Tipo: text

Valor predeterminado: /bonifa

dataFromSuccess

Etiqueta: Respuesta exitosa

Tratar como Valor dinámico

Tipo: text

Valor predeterminado:

dataFromError

Etiqueta: Respuesta fallida

Tratar como Valor dinámico

Tipo: text

Valor predeterminado:

Figura 24. Widget personalizado botón continuar, flujo registro de actividades diarias

El tercer flujo del sistema es el flujo de aprobaciones gerenciales, en el cual el gerente de campo en turno, y el gerente de operaciones realizan la aprobación mensual de las actividades diarias previamente llenadas en el transcurso de todo el mes por cada uno de los colaboradores. Este flujo se lo puede mirar en la figura 22.

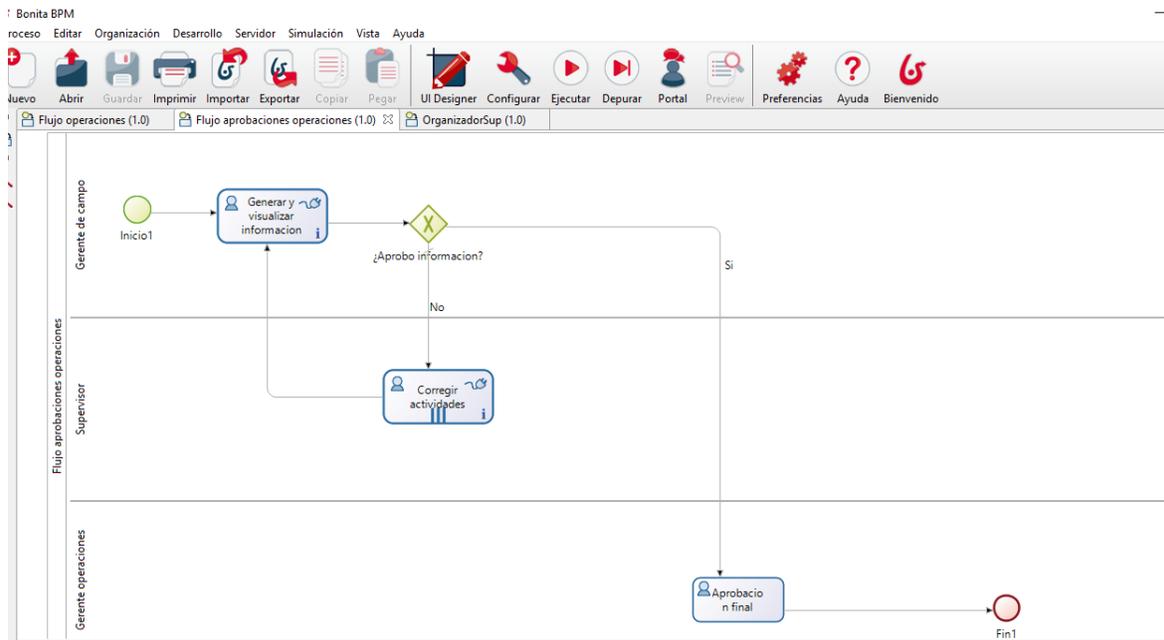


Figura 25. Flujo de aprobaciones gerenciales

Dentro de este flujo se realiza la integración por transferencia de archivos con el sistema Ssquarenet, para lo cual primero se extrae la información de todas las actividades diarias del mes, para así poderlas mostrar en formato resumido hacia los gerentes tanto de campo como de operaciones. Para extraer esta información una vez más se hace uso de los conectores para extraer el listado de las actividades, y ponerlas en una variable de proceso para así ser posteriormente mostrada. De la figura 23 a la 25 se puede ver el wizard del conector que extrae el listado de actividades diarias.

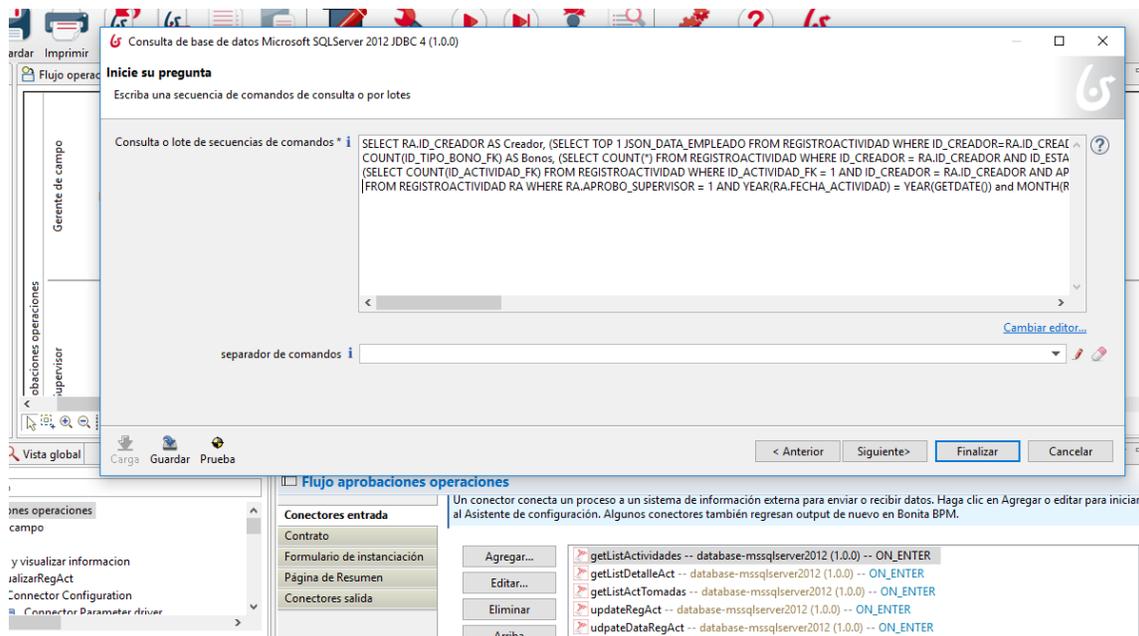


Figura 26. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 1

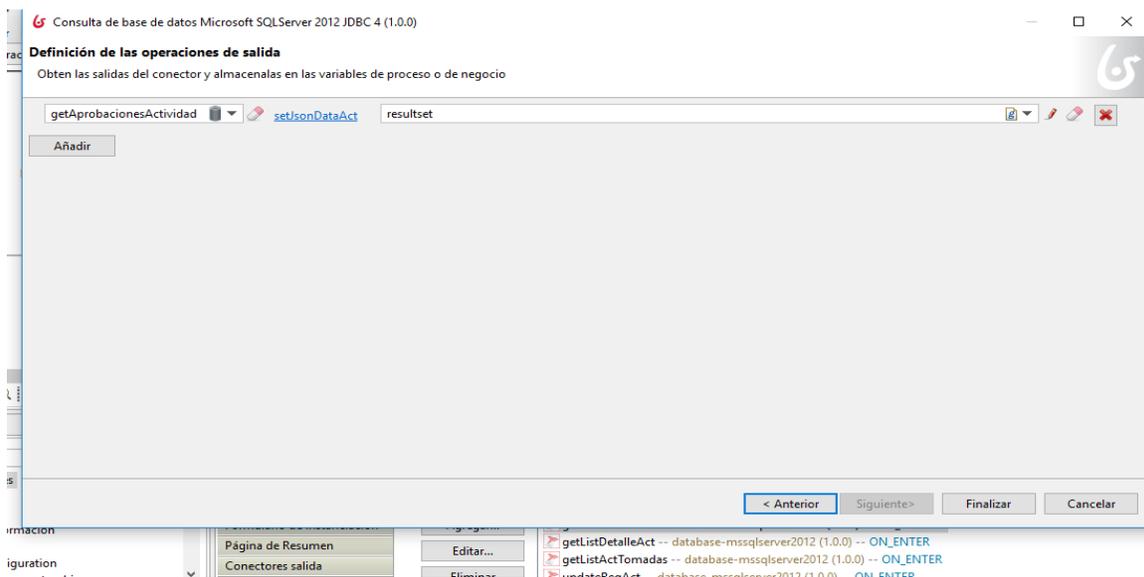


Figura 27. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 2

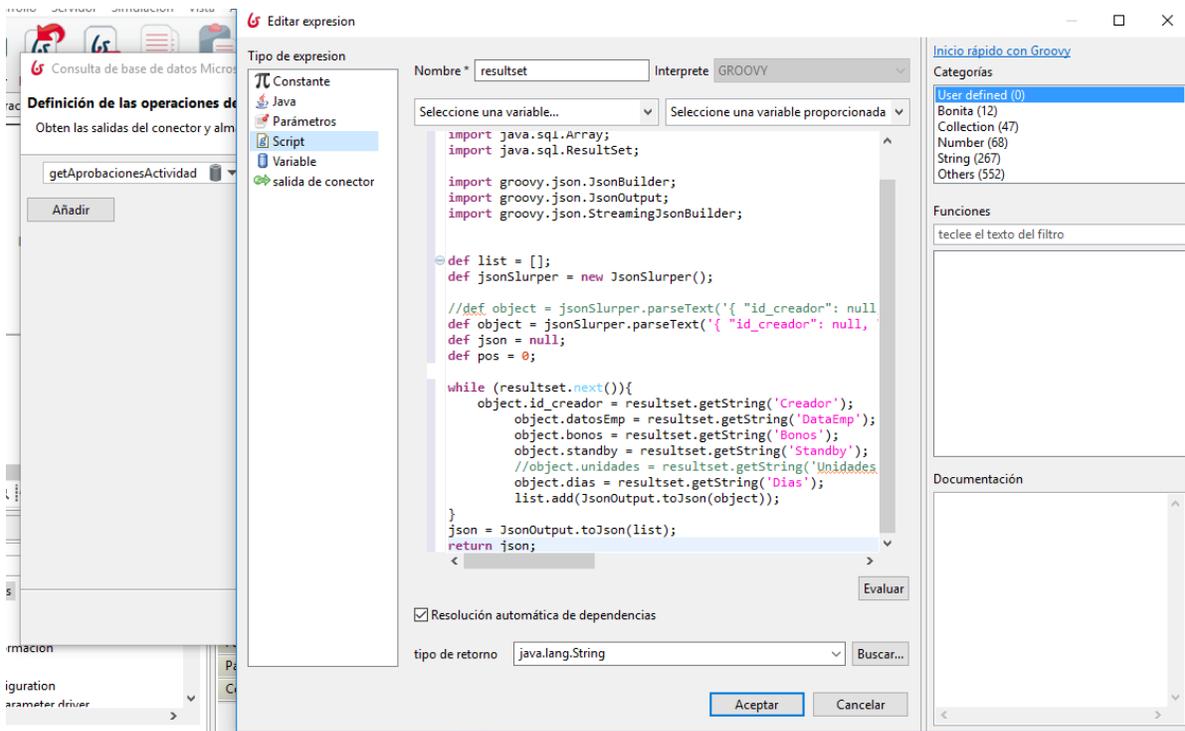


Figura 28. Wizard de obtención de listado de actividades aprobadas, flujo aprobaciones gerenciales, parte 3

Para poder mostrar la información consolidada de las actividades diarias por colaborador se construyó un widget personalizado que muestra tanto la información consolidada por colaborador, así como también el resumen. La figura 26 muestra la grilla personalizada, realizada con el editor de widget personalizados de Bonita BPM UI Designer.

**EDITOR WIDGETS PERSONALIZADOS** kendoGrid Guardar

**Descripción**

Descripción del widget

**Plantilla**

```

1 | <kendo-grid options="gridOptions" k-rebind="gridOptions.selectable">
2 |   <div k-detail-template>
3 |     <div kendo-grid k-options="detailGridOptions(dataItem)" k-on-change="handleChange(data, dataItem, columns)"></div>
4 |   </div>
5 | </kendo-grid>
6 | <!--options="detailGridOptions" k-rebind="detailGridOptions.selectable"-->

```

**Controller**

```

1 | function(scope, $http) {
2 |   var responseContext = $http.get("../API/tpn/userTask/"+"$scope.properties.idTarea+"/"context");
3 |   var horaInicio=null;
4 |   var horaFin=null;
5 |   var totalHoras=null;
6 |   var H1 = 0;
7 |   var H25 = 0;
8 |   var H50 = 0;
9 |   var H100 = 0;
10 |   var colaboradorAnterior=null;
11 |   var arregloTotalHoras=[];
12 |   responseContext.success(function(data, status, headers, config) {
13 |     var url = data.gridAprobacionActividad.ret_line;
14 |     var responsePromise = $http.get("/bonita/"+url);
15 |
16 |     responsePromise.success(function(data, status, headers, config) {

```

**Recursos**

Nombre	Tipo
http://kendo.cdn.telerik.com/2015.2.805/styles/kendo.common.min.css	CSS
http://kendo.cdn.telerik.com/2015.2.805/styles/kendo.metro.min.css	CSS
http://kendo.cdn.telerik.com/2015.2.805/js/kendo.all.min.js	JavaScript

**Módulos angular requeridos**

kendo directives

**Propiedades**

**idTarea**  
Etiqueta: Id tarea  
Tratar como: Valor dinámico  
Tipo: text  
Valor predeterminado:

**dataGrillaResumen**  
Etiqueta: Grilla resumen  
Tratar como: Enlace bidireccional

**dataGrillaDetalle**  
Etiqueta: Grilla detalle  
Tratar como: Enlace bidireccional

aprGerCampo

Figura 29. Widget de grilla personalizada con Bonita BPM UI Designer

Dentro de este Widget se realiza el cálculo de horas extraordinarias y suplementarias por colaborador, que posteriormente servirá como dato para el archivo plano en formato CSV generado por el sistema, y el cuál posteriormente será leído por el sistema Squarenet de talento humano. La figura 27 muestra el cálculo de horas al 25%, 50% y 100% correspondientes a las horas extraordinarias y suplementarias.

**Controller**

```

37 |   dato.HoraInicio = jsonGeneral.hora_inicio;
38 |   dato.HoraFin = jsonGeneral.hora_fin;
39 |   /***** calculo horas 25%, 50%, 100%*****/
40 |   if(index == 0){
41 |     colaboradorAnterior = parseInt(dato.Creador);
42 |   }else{
43 |     colaboradorAnterior = parseInt(JSON.parse(dataActDetalle[index-1]).id_creador);
44 |   }
45 |   if(colaboradorAnterior != parseInt(dato.Creador)){
46 |     var horasColaborador = {};
47 |     horasColaborador.H25=H25;
48 |     horasColaborador.H50=H50;
49 |     horasColaborador.H100=H100;
50 |     horasColaborador.creador=colaboradorAnterior;
51 |     arregloTotalHoras.push(horasColaborador);
52 |     H25 = 0;
53 |     H50 = 0;
54 |     H100 = 0;
55 |   }
56 |
57 |   horaInicio = parseInt(dato.HoraInicio);
58 |
59 |

```

**idTarea**  
Etiqueta: Id tarea  
Tratar como: Valor dinámico  
Tipo: text  
Valor predeterminado:

**dataGrillaResumen**  
Etiqueta: Grilla resumen  
Tratar como: Enlace bidi

**dataGrillaDetalle**  
Etiqueta: Grilla detalle  
Tratar como: Enlace bidi

**aprGerCampo**  
Etiqueta: Aprobó Gerer  
Tratar como: Enlace bidi

Figura 30. Calculo horas extraordinarias y suplementarias

La vista resumen y detalle de las actividades diarias por colaborador que los gerentes tanto de campo como de operaciones ven es la que se muestra en la figura 28.

FECHA DE ACTIVIDAD	ACTIVIDAD	ESTADO	TIPO DE BO...	HORA DE INI.	HORA DE FIN	LOCACION	CLIENTE	FECHA APROBACI...
2016-08-25 00:00:00.0	TURNO	PRODUCC...	B	19	7	GUANTA 13	PARDALIS	2016-08-28 00:00:00.0
2016-08-26 00:00:00.0	TURNO	PRODUCC...	B	7	16	GUANTA 13	PETROAMAZONAS EP	2016-08-28 00:00:00.0
2016-08-27 00:00:00.0	TURNO	PRODUCC...	DB	12	19	APAICA 11	PETROECUADOR	2016-08-28 00:00:00.0

FECHA DE ACTIVIDAD	ACTIVIDAD	ESTADO	TIPO DE BO...	HORA DE INI.	HORA DE FIN	LOCACION	CLIENTE	FECHA APROBACI...
2016-08-23 00:00:00.0	TURNO	PRODUCC...	B	19	7	APAICA 11	PETROAMAZONAS EP	2016-08-28 00:00:00.0
2016-08-24 00:00:00.0	TURNO	PRODUCC...	B	20	6	DRAGO NORTE 36	RIO NAPO	2016-08-28 00:00:00.0
2016-08-25 00:00:00.0	TURNO	PRODUCC...	DB	19	4	CONONACO 40	REPSOL	2016-08-28 00:00:00.0
2016-08-26 00:00:00.0	TURNO	PRODUCC...	B	21	7	LAGO 60	PETROECUADOR	2016-08-28 14:22:00.0
2016-08-27 00:00:00.0	TURNO	PRODUCC...	DB	20	7	LAGO 60	PARDALIS	2016-08-28 14:22:00.0

Figura 31. Vista resumen y detalle de actividades mensuales por colaborador, flujo aprobaciones gerenciales

### 3.2 Resultados

Se optimizó el control de actividades diarias de los colaboradores de las áreas de operaciones y proyectos. Esto debido a que los colaboradores de dichas áreas deben realizar diariamente su ingreso de actividades para el pago de bonos y horas extras, las cuales quedarán registradas en la base de datos del sistema, impidiendo su manipulación.

Con la automatización del proceso de pagos adicionales, se ha logrado disminuir la carga laboral para el personal de nómina, que realizaban antes esta tediosa tarea cada fin de mes, que en conjunto con los supervisores de cada área iban revisando la ficha de Excel y correos de respaldos de los supervisores, para posteriormente

generar un archivo plano en formato .CSV de cada uno de los colaboradores, describiendo el conteo de bonos y horas extras de cada colaborador. Ahora la generación automática de este archivo que se la puede ver en el Anexo 4, ahorra mucho tiempo.

En el Anexo 5 se puede mirar una carta de conformidad redactada y firmada por la persona encargada del proyecto, describiendo la aportación a la compañía del desarrollo realizado.

### 3.3 Pruebas de caja negra

Son un enfoque de “pruebas donde los examinadores no tienen acceso al código fuente de un sistema o sus componentes. Las pruebas se derivan de la especificación del sistema.” (Sommerville, 2011, p. 744).

Estas pruebas fueron realizadas para poder validar que el resultado obtenido sea igual al resultado esperado.

Para una primera prueba se puso a prueba el módulo de login al sistema, obteniendo como resultado lo siguiente

Tabla 16.

Prueba de caja negra del módulo login.

Módulo de login					
ID	Descripción	Pasos	Resultado esperado	Resultado obtenido	Estado

P1	Probar el correcto funcionamiento del login al sistema, ingresando un usuario incorrecto.	<ol style="list-style-type: none"> <li>1.- Ir a la pantalla de login.</li> <li>2.- Introducir en el campo nombre de usuario "admin".</li> <li>3.- Ingresar en el campo contraseña "install".</li> <li>4. Dar clic en el botón "Inicio de sesión".</li> </ol>	Usuario o contraseña inválidos	Usuario o contraseña inválidos	Aprobado
P2	Probar el correcto funcionamiento del login al sistema, ingresando una contraseña incorrecta.	<ol style="list-style-type: none"> <li>1.- Ir a la pantalla de login.</li> <li>2.- Introducir en el campo nombre de usuario "admin".</li> <li>3.- Ingresar en el campo contraseña "install".</li> <li>4. Dar clic en el botón "Inicio de sesión".</li> </ol>	Usuario o contraseña inválidos	Usuario o contraseña inválidos	Aprobado
P3	Probar el correcto funcionamiento del login al sistema, ingresando un usuario y contraseña correctos.	<ol style="list-style-type: none"> <li>1.- Ir a la pantalla de login.</li> <li>2.- Introducir en el campo nombre de usuario "install".</li> <li>3.- Ingresar en el campo contraseña "install".</li> </ol>	Login exitoso	Login exitoso	Aprobado

		4. Dar clic en el botón "Inicio de sesión".			
--	--	---	--	--	--

La segunda prueba se la realizó sobre el módulo de creación de usuarios, para lo cual se creó un nuevo usuario en el sistema.

Tabla 17.

Prueba de caja negra del módulo creación de usuarios.

Módulo de creación de usuarios					
ID	Descripción	Pasos	Resultado esperado	Resultado obtenido	Estado
P1	Probar la correcta validación del formulario de creación de un usuario.	1.- Dar clic en el menú de "Organización". 2.- Seleccionar la opción "Usuarios". 3.- Dar clic en el botón "Crear". 4.- Dejar en blanco todos los campos requeridos, que son: Nombre de usuario, Contraseña, Confirmar contraseña, Nombre, Apellido.	Validación incorrecta	Validación incorrecta	Aprobado

		5.- Dar clic en "Crear".			
P2	Probar el correcto funcionamiento del login al sistema, ingresando una contraseña incorrecta.	<p>1.- Dar clic en el menú de "Organización".</p> <p>2.- Seleccionar la opción "Usuarios".</p> <p>3.- Dar clic en el botón "Crear".</p> <p>4.- Llenar todos los campos requeridos, de Nombre de usuario, Contraseña, Confirmar contraseña, Nombre, Apellido.</p> <p>5.- Dar clic en "Crear".</p>	Creación de usuario exitosa	Creación de usuario exitosa	Aprobado
P3	Probar el login del nuevo usuario creado	<p>1.- Ir a la pantalla de login.</p> <p>2.- Introducir en el campo nombre de usuario "test".</p> <p>3.- Ingresar en el campo contraseña "test".</p> <p>4. Dar clic en el botón "Inicio de sesión".</p>	Usuario sin suficientes privilegios	Usuario sin suficientes privilegios	Aprobado

P4	Asignación de un grupo y rol al usuario creado	<p>1.- Dar clic en el menú de "Organización".</p> <p>2.- Seleccionar el usuario recién creado "test".</p> <p>3.- Dar clic en el botón "Mas ..."</p> <p>4.- Ir a la opción membresía, y dar clic en el botón "Anadir".</p> <p>5.- Seleccionar el grupo al cual pertenece el usuario (colaboradores)</p> <p>6.- Seleccionar el rol del usuario (member).</p> <p>7.- Dar clic en el botón "Anadir"</p>	Membresía agregada exitosamente	Membresía exitosa	Aprobado
P5	Probar el login del nuevo usuario creado, con su membresía asignada	<p>1.- Ir a la pantalla de login.</p> <p>2.- Introducir en el campo nombre de usuario "test".</p> <p>3.- Ingresar en el campo contraseña "test".</p>	Login exitoso	Login exitoso	Aprobado

		4. Dar clic en el botón "Inicio de sesión".			
--	--	---	--	--	--

La tercera prueba se la realizo en el módulo de importación/exportación de organización.

Tabla 18.

Prueba de caja negra del módulo importación/exportación de organización

Módulo de importación/exportación de organización					
ID	Descripción	Pasos	Resultado esperado	Resultado obtenido	Estado
P1	Probar la importación de una organización.	1.- Loguearse en el sistema como administrador. 2.- Dar clic en el menú de "Organización". 3.- Seleccionar la opción "Importación/exportación" 4.- Dar clic en seleccionar un archivo 5.- Seleccionar el archivo Serpetpet.xml 6.- Dar clic en el botón "Importar"	Importación correcta	Importación correcta	Aprobado
P2	Verificar la correcta	1.- Dar clic en el menú de "Organización".	Verificación correcta	Verificación correcta	Aprobado

	importación de la organización	2.- Seleccionar la opción "Usuarios" 3.- Verificar que los nuevos usuarios consten en el listado de usuarios			
P3	Probar la exportación de la organización actual	1.- Loguearse en el sistema como administrador. 2.- Dar clic en el menú de "Organización". 3.- Seleccionar la opción "Importación/exportación" 4.- Dar clic en el botón "Exportar" 5.- Verificar que se descargue un archivo con extensión .xml	Exportación correcta	Exportación correcta	Aprobado
P4	Verificar la correcta exportación de la organización	1.- Abrir la ubicación del archivo descargado 2.- Validar de las información de los usuarios este en el formato xml correcto	Verificación correcta	Verificación correcta	Aprobado

La cuarta prueba se la realizó sobre el módulo de instalación de flujos, la cual requiere un perfil administrador de la plataforma.

Tabla 19.

Prueba de caja negra del módulo instalación de flujos

Módulo de instalación de flujos					
ID	Descripción	Pasos	Resultado esperado	Resultado obtenido	Estado
P1	Probar la instalación correcta de un flujo.	1.- Loguearse en el sistema como administrador. 2.- Dar clic en el menú de "BPM". 3.- Seleccionar la opción "Procesos" 4.- Dar clic en el botón "Instalar", ubicado al lado superior derecho. 5.- Seleccionar el archivo "Organizador.bar" 6.- Dar clic en el botón "Instalar"	Instalación correcta	Instalación satisfactoria	Aprobado
P2	Asignar un grupo al flujo instalado.	1.- Dar clic en el tab actores. 2.- Dar clic en el botón "+" del ítem "Grupos" 3.- Dar clic en el combo de selección "Seleccionar grupos". 4.- Buscar el grupo "Supervisores". 5.- Dar clic en el botón "Aplicar".	Asignación de grupo correcta	Asignación correcta	Aprobado

P3	Habilitar el flujo instalado.	1.- Dar clic en la pestaña "General". 2.- Dar clic en el botón "Estado de activación".	Cambio de estado de "Disabled" a "Enabled"	Cambio correcto a estado "Enable"	Aprobado
P4	Verificar que se pueda iniciar el flujo instalado.	1.- Logearse como supervisor. 2.- Dar clic en el tab "Procesos". 3.- Seleccionar el flujo llamado "Organizador". 4.- Dar clic en el botón "Iniciar".	Apertura de grilla del organizador del personal	Grilla del organizador de personal mostrada	Aprobado

La quinta prueba se la realizó sobre el módulo aplicaciones, la cual requiere un perfil administrador de la plataforma.

Tabla 20.

Prueba de caja negra sobre el módulo de aplicaciones, creación de vista mantenimiento tabla clientes.

Módulo de instalación de flujos					
ID	Descripción	Pasos	Resultado esperado	Resultado obtenido	Estado
P1	Crear una nueva aplicación.	1.- Loguearse en el sistema como administrador. 2.- Dar clic en el menú de "Aplicaciones". 3.- Dar clic en el botón "Nuevo". 4.- Llenar el formulario "Crear una aplicación".	Nueva aplicación en blanco	Nueva aplicación en blanco	Aprobado

		5.- Dar una URL amigable. 6.- Seleccionar el perfil de acceso para la aplicación. 7.- Dar clic en el botón “crear”.			
P2	Configurar la aplicación.	1.- Dar clic en el botón “...” en donde se edita las configuraciones. 2.- Dentro del bloque “Paginas” dar clic en el botón “Añadir”. 3.- Seleccionar la página “Administrador” 4.- Darle una url amigable para la página de mantenimiento de la tabla clientes. 5.- Dar clic en el botón “Añadir”.	Configuración correcta	Configuración satisfactoria	Aprobado
P3	Probar aplicación creada.	1.- Logearse como administrador. 1.- Ingresar a cualquier navegador. 2.- Digitar la url “http://localhost:8686/bonita/apps/administrador/clientes/”	Cambio de estado de “Disabled” a “Enabled”	Cambio correcto a estado “Enable”	Aprobado

### 3.4 Pruebas de caja blanca

“Las pruebas de caja blanca o pruebas estructurales son un enfoque sistemático a las pruebas donde se usa el conocimiento del código fuente del programa para diseñar pruebas de defecto.” (Sommerville, 2011, p. 585).

Estas pruebas se las realizó sobre el módulo de administración, validando las operaciones CRUD (Create, Read, Update, Delete).

Tabla 21.

Prueba de caja blanca, operación read sobre la tabla cliente

Operación read sobre la tabla cliente				
<b>Nombre de la prueba</b>		Lectura datos tabla cliente		
<b>Descripción de la prueba</b>		Esta prueba pretende verificar el correcto funcionamiento de la lectura de información de la tabla cliente		
<b>Paso</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado Obtenido</b>	<b>Estado</b>
1	Abrir la administración de la tabla cliente.	Apertura de la página	Apertura de la página	Aprobado
2	Leer vía Ajax a la dirección /administradorBonita/rest/administrador/listadoClientes.	Invocación de método de lectura de la tabla cliente	Invocación de método de lectura de la tabla cliente	Aprobado
3	Ejecutar método <i>readCliente</i> enlazado al path <i>listadoClientes</i> .	Ingreso al método <i>readCliente</i> del controlador	Ingreso al método <i>readCliente</i> del controlador	Aprobado
4	Dentro del método se realiza una llamada al Entity Manager para poder	Instancia del Entity Manager	Instancia del Entity Manager	Aprobado

	acceder a la manipulación de datos de la tabla cliente.			
5	Se ejecuta la consulta en formato del Entity Manager con el método <i>createQuery</i>	Data de la tabla cliente	Data de la tabla cliente	Aprobado
6	Si no existe ningún error en la ejecución de la consulta, se asigna el resultado de la consulta a un arreglo de clientes	Arreglo de clientes	Arreglo de clientes	Aprobado
7	Se retorna un string en formato json	Retorno del resultado	Retorno del resultado	Aprobado
8	Se transforma la data en un objeto y se la muestra en una tabla	Renderizado de los datos	Renderizado de los datos	Aprobado

Tabla 22.

Prueba de caja blanca, operación create sobre la tabla cliente

Operación read sobre la tabla cliente				
<b>Nombre de la prueba</b>		Inserción de un nuevo registro sobre la tabla cliente		
<b>Descripción de la prueba</b>		Esta prueba pretende verificar el correcto funcionamiento de una inserción de un registro nuevo sobre la tabla cliente		
<b>Paso</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado Obtenido</b>	<b>Estado</b>
1	Abrir la administración de la tabla cliente.	Apertura de la pagina	Apertura de la pagina	Aprobado

2	Dar clic en el botón "Create".	Evento clic sobre el botón Create	Evento clic sobre el botón Create	Aprobado
3	Ingresar el id del cliente y una descripción y dar clic en el botón "Update"	Llamado al evento create	Llamado al evento create	Aprobado
4	Ejecución vía POST a la URL: /administradorBonita/rest/administrador/createCliente	Ejecución post	Ejecución post	Aprobado
5	Ejecutar método <i>createCliente</i> enlazado al path <i>createCliente</i> .	Ingreso al método <i>createCliente</i> del controlador	Ingreso al método <i>createClient</i> e	Aprobado
6	Dentro del método se realiza una llamada al Entity Manager para poder acceder a la manipulación de datos de la tabla cliente.	Instancia del Entity Manager	Instancia del Entity Manager	Aprobado
7	Se crea una instancia a la tabla cliente se asigna los datos de entrada.	Asignación de datos	Asignación de datos	Aprobado
8	Se ejecuta la persistencia y el commit que provee el Entity Manager.	Inserción de datos	Inserción de datos	Aprobado
10	Se retorna un string en formato json	Retorno del resultado	Retorno del resultado	Aprobado
11	Se actualiza la tabla	Renderizado de los datos	Renderizado de los datos	Aprobado

Tabla 23.

Prueba de caja blanca, operación update sobre la tabla cliente

Operación update sobre la tabla cliente				
Nombre de la prueba		Actualización de un registro sobre la tabla cliente		
Descripción de la prueba		Esta prueba pretende verificar el correcto funcionamiento de una actualización de un registro nuevo sobre la tabla cliente		
Paso	Descripción	Resultado esperado	Resultado Obtenido	Estado
1	Abrir la administración de la tabla cliente.	Apertura de la pagina	Apertura de la pagina	Aprobado
2	Dar clic en el botón Edit.	Evento clic sobre el botón Edit	Evento clic sobre el botón Edit	Aprobado
4	Ejecución vía POST a la URL: /administradorBonita/rest/administrador/editCliente/id_cliente	Ejecución post	Ejecución post	Aprobado
5	Ejecutar método <i>updateCliente</i> en el controlador enlazado al path <i>editCliente</i> .	Ingreso al método <i>updateCliente</i> del controlador	Ingreso al método <i>updateCliente</i>	Aprobado
6	Dentro del método se realiza una llamada al Entity Manager para poder acceder a la manipulación de datos de la tabla cliente.	Instancia del Entity Manager	Instancia del Entity Manager	Aprobado

7	Se crea una instancia a la tabla cliente se asigna los datos de entrada.	Asignación de datos	Asignación de datos	Aprobado
8	Se ejecuta find con el Entity Manager.	Ejecución de método find del Entity Manager	Ejecución de método find del Entity Manager	Aprobado
9	Se ejecutan los métodos merge y commit que provee el Entity Manager.	Actualización de datos	Actualización de datos	Aprobado
10	Se retorna un string en formato json	Retorno del resultado	Retorno del resultado	Aprobado
11	Se actualiza la tabla	Renderizado de los datos	Renderizado de los datos	Aprobado

Tabla 24.

Prueba de caja blanca, operación delete sobre la tabla cliente

Operación delete sobre la tabla cliente				
<b>Nombre de la prueba</b>		Borrado de un registro sobre la tabla cliente		
<b>Descripción de la prueba</b>		Esta prueba pretende verificar el correcto funcionamiento del borrado de un registro nuevo sobre la tabla cliente		
<b>Paso</b>	<b>Descripción</b>	<b>Resultado esperado</b>	<b>Resultado Obtenido</b>	<b>Estado</b>

1	Abrir la administración de la tabla cliente.	Apertura de la pagina	Apertura de la pagina	Aprobado
2	Dar clic en el botón Delete.	Evento clic sobre el botón Delete	Evento clic sobre el botón Delete	Aprobado
4	Ejecución vía POST a la URL: /administradorBonita/rest/administrador/deleteCliente/ id_cliente	Ejecución post	Ejecución post	Aprobado
5	Ejecutar método <i>deleteCliente</i> en el controlador enlazado al path <i>deleteCliente</i> .	Ingreso al método <i>deleteCliente</i> del controlador	Ingreso al método <i>deleteClient</i> e	Aprobado
6	Dentro del método se realiza una llamada al Entity Manager para poder acceder a la manipulación de datos de la tabla cliente.	Instancia del Entity Manager	Instancia del Entity Manager	Aprobado
7	Se ejecuta find con el Entity Manager.	Ejecución de método find del Entity Manager	Ejecución de método find del Entity Manager	Aprobado
8	Se ejecutan los métodos remove y commit que provee el Entity Manager.	Borrado de datos	Borrado de datos	Aprobado
9	Se retorna un string en formato json	Retorno del resultado	Retorno del resultado	Aprobado

10	Se actualiza la tabla	Renderizado de los datos	Renderizado de los datos	Aprobado
----	-----------------------	--------------------------	--------------------------	----------

## 4 Conclusiones y recomendaciones

### 4.1 Conclusiones

Se ha logrado implementar exitosamente el sistema web para el control de actividades diarias, y gestión de pagos adicionales para las áreas de operaciones y proyectos de la compañía Sertecpet S.A.

Se realizó la integración bidireccional por transferencia de archivos con el sistema de talento humano SQUARENET, que a través de la generación, y posterior lectura del archivo plano en formato CSV, carga los pagos adicionales de los colaboradores en el sistema SQUARENET, el cual ayuda a la gestión de nómina, control de asistencias y recursos humanos para la compañía.

Se tuvo éxito con la integración unidireccional por base de datos compartida con el sistema biométrico de asistencia, cuyo propósito era controlar el ingreso de actividades diarias, de colaboradores en jornada activa.

Mediante las pruebas de caja negra y de caja blanca se pudo validar el correcto funcionamiento del sistema

La utilización de Bonita BPM facilitó el desarrollo del sistema, a través de sus módulos de conexión a diversas bases de datos y también gracias a su módulo de construcción de interfaz de usuario, que usa las tecnologías actuales en interfaz de usuario como lo son AngularJs y Bootstrap.

## 4.2 Recomendaciones

Para futuros desarrollos con la herramienta Bonita BPM, y una base de datos SQL server, se recomienda leer detenidamente la documentación oficial de Bonita BPM, y así mismo la documentación de SQL server para conocer los posibles inconvenientes que se pueden generar.

Si se desea tener tablas en la base de datos que sean parametrizables, es necesario tomar en cuenta que Bonita BPM, no permite modificaciones de las tablas pertenecientes al Business Data Model (BDM), por lo que esto representa un desarrollo extra fuera de la herramienta del Bonita BPM Engine.

## REFERENCIAS

- Bahit, E. (2011). *POO y MVC en PHP*. Argentina: Creative Commons BY-NC-ND.
- Bello, J., Uribe, C., Nuñez, O. (2016) *Ciclo de vida BPM*. Recuperado el 2 de Octubre de 2016 de <https://bpmsosw.wordpress.com/2012/02/12/ciclo-de-vida-bpm/>
- Bonitasoft. (2016). *Custom Pages & Applications*. Recuperado el 15 de Octubre de 2016 de <https://www.youtube.com/watch?v=B34eJdgGdt4>
- Bonitasoft. (2016). *First steps after setup*. Recuperado el 29 de Enero de 2017 de <http://documentation.bonitasoft.com/6.x-7.2/first-steps-after-setup-0>
- Bonitasoft. (2016). *JBoss Bumble*. Recuperado el 5 de Septiembre de 2016 de <http://documentation.bonitasoft.com/6.x-7.2/jboss-bundle-3>
- Bootstrap. (2016). *Bootstrap 3.3.4 released*. Recuperado el 24 de mayo del 2015 de <http://blog.getbootstrap.com/>
- Henrik, K. & Mattias, S. *Kanban y Scrum obteniendo lo mejor de ambos*. U.S.A.: Media Inc. ISBN: 978-0-557-13832-6.
- Hohpe, G. y Woolf, B. (2012). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Boston, U.S.A.: Addison-Wesley ISBN: 9780133065107.
- Hohpe, G. y Woolf, B. (2004). *Enterprise Integration Patterns*. New York: Addison-Wesley.

IBM, (2016). Introducción a la programación Java, parte 1: Conceptos básicos del lenguaje Java. Recuperado el 2 de Octubre de 2016 de <http://www.ibm.com/developerworks/ssa/java/tutorials/j-introjava1>

Jeston, J. y Neils, J. (2014). *Business Process Management* (3 ra ed). New York: Routledge

Mero, K. (2011). *Fase de requerimientos*. Recuperado el 29 de Agosto de 2016 de <https://blogereducativo.wordpress.com/2011/09/07/fase-de-requerimientos/>

Proyectos ágiles (2015). *SCRUM*. Recuperado el 15 de Julio de 2015 de <http://www.proyectosagiles.org/que-es-scrum>

Sertecpet S.A. (2015), *Exploración y Producción de Gas y Petróleo*. Recuperado el 23 de mayo del 2015 de <http://www.sertecpet.net/business-areas>

Sertecpet S.A. (2015), *Resumen Corporativo*. Recuperado el 23 de mayo del 2015 de <http://www.sertecpet.net/about>

Sommerville, I. (2011). *Ingeniería de Software* (9.a ed.). Mexico: Person Educación.

SQUARENET. (2016). *Home - Noticias*. Recuperado el 24 de Noviembre de 2016 de <http://www.squarenet.com.ec/>

Torres, E. (2016). *MVC Modelo Vista Controlador*. Recuperado el 01 de Diciembre de 2016 de <https://www.scribd.com/doc/316735642/MVC-MODELO-VISTA-CONTROLADOR>

## **ANEXOS**

## Anexo 1 Product Backlog

ID de Historia	Nombre de la Historia	Fecha de Culminación	Duración (Días)	Sprint	Prioridad	Comentarios
HI-01	Instalación de Bonita BPM en ambiente de desarrollo	24/07/2015	2	1	1	
HI-02	Configuración de Bonita BPM	27/07/2015	3	1	2	
HI-03	Configuración de la base de datos SQL Server	28/07/2015	1	1	2	
HI-04	Pruebas de ambiente de desarrollo	29/07/2015	1	1	3	
HI-05	Reunión con encargado de talento humano para entender el proceso.	30/07/2015	1	2	2	
HI-06	Diseño del modelo Entidad Relación.	03/08/2015	4	2	1	

HI-07	Creación del diagrama físico de base de datos.	05/08/2015	2	2	2	
HI-08	Modelado del flujo de organizador de personal.	09/08/2015	4	3	1	
HI-09	Construcción de formulario de organizador de personal.	12/08/2015	3	3	2	
HI-010	Desarrollo de lógica del negocio para el organizador de personal.	19/08/2015	7	4	1	
HI-011	Desarrollo reglas de transición.	23/08/2015	4	5	1	
HI-012	Desarrollo de plantilla de notificación de ingreso del organizador de personal.	26/08/2015	3	5	2	
HI-013	Pruebas funcionales	28/08/2015	2	6	1	

	del organizador de personal.					
HI-014	Modelado del flujo de ingreso de actividades diarias para operaciones y proyectos.	02/09/2015	5	6	2	
HI-015	Construcción del formulario de ingreso de actividades diarias.	06/09/2015	4	7	2	
HI-016	Construcción de lógica de negocio para extracción de datos del colaborador del sistema SQUARENE T de talento humano.	09/09/2015	3	7	3	
HI-017	Construcción de lógica de negocio para extracción de datos del colaborador del sistema	12/09/2015	3	8	3	

	SQUARENE T de talento humano.					
HI-018	Construcción de lógica de negocio para integración con sistema biométrico.	21/09/2015	9	8	4	
HI-019	Construcción de lógica de negocio para integración con sistema biométrico.	26/09/2015	5	9	1	
HI-020	Pruebas de integración	28/09/2015	2	9	2	
HI-021	Validaciones del formulario de ingreso de actividades diarias.	03/10/2015	5	10	2	
HI-022	Construcción de plantilla de correo de ingreso de actividad diaria.	05/10/2015	2	10	3	
HI-023	Desarrollo de reglas de transición de primera	12/10/2015	7	11	1	

	actividad del flujo.					
HI-024	Construcción del formulario de aprobación de actividades diarias.	16/10/2015	4	12	1	
HI-025	Implementación de lógica de negocio para aprobación de actividades diarias.	19/10/2015	3	12	2	
HI-026	Implementación de lógica de negocio para aprobación de actividades diarias.	24/10/2015	5	13	1	
HI-027	Construcción de plantilla de correo de aprobación de actividad diaria.	26/10/2015	2	13	2	

HI-028	Desarrollo de reglas de transición de aprobación de actividad diaria.	02/11/2015	7	14	1	
HI-029	Construcción del formulario de corrección de actividades diarias.	06/11/2015	4	15	1	
HI-030	Implementación de lógica de negocio para corrección de actividades diarias.	09/11/2015	3	15	2	
HI-031	Implementación de lógica de negocio para corrección de actividades diarias.	13/11/2015	4	16	1	
HI-032	Construcción de plantilla de correo para corrección de	16/11/2015	3	16	2	

	actividad diaria.					
HI-033	Desarrollo de reglas de transición de corrección de actividad diaria.	23/11/2015	7	17	1	
HI-034	Pruebas funcionales flujo de ingreso de actividades para operaciones y proyectos.	26/11/2015	3	18	1	
HI-035	Modelado del flujo de aprobaciones gerenciales para operaciones y proyectos.	30/11/2015	4	18	2	
HI-036	Construcción de la vista resumen mensual por colaborador de aprobación de gerente de campo.	07/12/2015	7	19	1	

HI-037	Validaciones de la vista resumen de aprobación del gerente de campo.	14/12/2015	7	20	2	
HI-038	Desarrollo reglas de negocio para vista resumen de aprobación gerente de campo.	21/12/2015	7	21	1	
HI-039	Desarrollo reglas de negocio para vista resumen de aprobación gerente de campo.	24/12/2015	3	22	2	
HI-040	Desarrollo de plantilla de notificación aprobación de gerente de campo.	28/12/2015	4	22	2	
HI-041	Desarrollo reglas de transición de aprobación	04/01/2016	7	23	1	

	del gerente de campo.					
HI-042	Construcción de la vista corrección de actividades.	08/01/2016	4	24	1	
HI-043	Validaciones vista corrección de actividades	11/01/2016	3	24	2	
HI-044	Desarrollo de reglas de negocio para vista corrección de actividades.	16/01/2016	5	25	1	
HI-045	Desarrollo de plantilla de notificación de corrección de actividades.	18/01/2016	2	25	2	
HI-046	Desarrollo reglas de transición de corrección de actividades.	22/01/2016	4	26	1	
HI-047	Construcción de la vista resumen mensual por colaborador	25/01/2016	3	26	2	

	de aprobación de gerente de operaciones.					
HI-048	Construcción de la vista resumen mensual por colaborador de aprobación de gerente de operaciones.	29/01/2016	4	27	1	
HI-049	Validaciones de la vista resumen de aprobación del gerente de campo.	01/02/2016	3	27	2	
HI-050	Desarrollo reglas de negocio para vista resumen de aprobación gerente de operaciones.	08/02/2016	7	28	1	
HI-051	Desarrollo de plantilla de notificación	10/02/2016	2	29	2	

	aprobación de gerente de campo.					
HI-052	Desarrollo reglas de transición de aprobación del gerente de campo.	15/02/2016	5	29	2	
HI-053	Desarrollo de la integración por transferencia de archivos con el sistema SQUARENE T	22/02/2016	7	30	1	
HI-054	Desarrollo de la integración por transferencia de archivos con el sistema SQUARENE T	27/02/2016	5	31	1	
HI-055	Pruebas de la integración	29/02/2016	2	31	2	
HI-056	Pruebas del flujo de	03/03/2016	3	32	1	

	aprobaciones gerenciales.					
HI-057	Subida a producción	07/03/2016	4	32	2	
HI-058	Instalación de eclipse para desarrollo de módulo de administración y reportes.	27/09/2016	1	33	1	
HI-059	Construcción del modelo del módulo administración.	03/10/2016	6	33	2	
HI-060	Construcción de la vista del módulo administración.	06/10/2016	3	34	1	
HI-061	Construcción del controlador del módulo administración.	10/10/2016	4	34	2	
HI-062	Construcción del controlador del módulo	12/10/2016	2	35	1	

	administración.					
HI-063	Integración módulo de administración al sistema.	15/10/2016	3	35	2	
HI-064	Pruebas del módulo de administración.	17/10/2016	2	35	3	
HI-065	Construcción del modelo del módulo reporte.	21/10/2016	4	36	1	
HI-066	Construcción de la vista del módulo reporte.	24/10/2016	3	36	2	
HI-067	Construcción del controlador del módulo de reporte.	30/10/2016	7	37	1	
HI-068	Integración módulo de reporte al sistema.	02/11/2016	3	38	1	
HI-069	Pruebas del módulo de reporte.	06/11/2016	4	38	2	

## Anexo 2 Sprints

### Sprint 1

Fecha de inicio	Fecha de finalización
23/jul/2015	29/jul/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Instalación de Bonita BPM en ambiente de desarrollo	Augusto Yépez	Realizado	Descarga de Bonita BPM del sitio oficial	2	2	24/07/2015
			Subida de archivos al servidor de desarrollo	3	3	
			Instalación de Bonita BPM Studio	1	1	
			Instalación de Bonita BPM Plataform	2	2	
Configuración de Bonita BPM	Augusto Yépez	Realizado	Configuración de usuarios administradores	2	2	27/07/2015

			Configuración del host virtual	2	3	
			Cambio en configuración de jboss bundle para distinto JDBC	16	18	
Configuración de la base de datos SQL Server	Augusto Yépez	Realizado	Modificación de archivo standalone.xml para utilización de otra base de datos	2	2	28/07/2015
			Configuración de base de datos, siguiendo documentación oficial de Bonita BPM	4	3	
Pruebas de ambiente de desarrollo	Augusto Yépez	Realizado	Pruebas de conexión a la base de datos de desarrollo	3	4	29/07/2015

### Sprint 2

Fecha de inicio	Fecha de finalización
30/jul/2015	05/ago/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Reunión con encargado de talento humano para entender el proceso.	Augusto Yépez	Realizado	Reunión de con personal de nómina para conocer el flujo de pagos adicionales	2	3	30/07/2015
Diseño del modelo Entidad Relación.	Augusto Yépez	Realizado	Diseño del modelo Entidad Relación	26	20	03/08/2015
Creación del diagrama físico de base de datos.	Augusto Yépez	Realizado	Modelado de BDM	10	12	05/08/2015
			Exportación de BMD a servidor de desarrollo	2	1	

### Sprint 3

Fecha de inicio	Fecha de finalización
06/ago/2015	12/ago/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
---------------------	-------------	--------	-------	-----------------	------------------	----------------------

Modelado del flujo de organizador de personal.	Augusto Yépez	Realizado	Modelado del flujo del organizado de personal en Bonita BPM Studio	28	24	09/08/2015
Construcción de formulario de organizador de personal.	Augusto Yépez	Realizado	Construcción del formulario organizador de personal en UI Designer de Bonita BPM Studio	8	6	12/08/2015
			Generación de Widget personalizado para el organizador de personal	16	20	

### Sprint 4

Fecha de inicio	Fecha de finalización
13/ago/2015	19/ago/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de lógica del negocio	Augusto Yépez	Realizado	Construcción de la lógica para la	12	12	19/08/2015

para el organizador de personal.			obtención los datos			
			Construcción de lógica de llenado de datos en la tabla	8	8	
			Validaciones en la tabla	16	16	
			Pruebas de verificación	2	4	

### Sprint 5

Fecha de inicio	Fecha de finalización
20/ago/2015	26/ago/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de transición	Augusto Yépez	Realizado	Desarrollo de reglas de transición	24	20	23/08/2015
Desarrollo de plantilla de notificación de ingreso del organizador de personal.	Augusto Yépez	Realizado	Obtención dinámica de datos de supervisor	16	12	26/08/2015
			Creación de plantilla y configuración de componente	4	3	

			para el envío de correos			
--	--	--	--------------------------	--	--	--

### Sprint 6

Fecha de inicio	Fecha de finalización
27/ago/2015	02/sep/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Pruebas funcionales del organizador de personal.	Augusto Yépez	Realizado	Pruebas de funcionales del flujo organizado de personal	12	10	28/08/2015
Modelado del flujo de ingreso de actividades diarias para operaciones y proyectos.	Augusto Yépez	Realizado		34	36	02/09/2015

### Sprint 7

Fecha de inicio	Fecha de finalización
03/sep/2015	09/sep/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del formulario de ingreso de actividades diarias.	Augusto Yépez	Realizado	Diseño del formulario en el UI Designer de Bonita BPM	12	10	06/09/2015
			Construcción de Widget personalizado para envío de datos	20	18	
Construcción de lógica de negocio para extracción de datos del colaborador del sistema SQUARENET de talento humano.	Augusto Yépez	Realizado	Generación de consulta hacia sistema SQUARENET	8	8	09/09/2015
			Configuración de componente para conexión a base de datos SQUARENET	8	8	
			Lógica de llenado y recorrido de datos del colaborador	10	8	

## Sprint 8

Fecha de inicio	Fecha de finalización
10/sep/2015	21/sep/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción de lógica de negocio para extracción de datos del colaborador del sistema SQUARENET de talento humano.	Augusto Yépez	Realizado	20	20	8	12/09/2015
Construcción de lógica de negocio para integración con sistema biométrico.	Augusto Yépez	Realizado	Evaluación de base de datos del sistema Biométrico	8	8	21/09/2015
			Generación de query a base de datos del sistema Biométrico	8	6	
			Configuración de componente para	8	8	

			conexión a base de datos del sistema Biométrico			
			Lógica de recorrido y llenado de datos en base de datos del sistema de pagos adicionales	24	24	

### Sprint 9

Fecha de inicio	Fecha de finalización
22/sep/2015	28/sep/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción de lógica de negocio para integración con sistema biométrico.	Augusto Yépez	Realizado	Implementacion de lógica de negocia para control de ingreso de actividades	24	26	26/09/2015

Pruebas de integración	Augusto Yépez	Realizado		10	9	28/09/2015
------------------------	---------------	-----------	--	----	---	------------

### Sprint 10

Fecha de inicio	Fecha de finalización
29/sep/2015	05/oct/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Validaciones del formulario de ingreso de actividades diarias.	Augusto Yépez	Realizado	Lógica de validaciones en formulario de ingreso de actividades diarias	34	38	03/10/2015
Construcción de plantilla de correo de ingreso de actividad diaria.	Augusto Yépez	Realizado	Lógica dinámica para obtener datos del colaborador y supervisor	6	6	05/10/2015
			Construcción de la plantilla de correo	4	4	

## Sprint 11

Fecha de inicio	Fecha de finalización
06/oct/2015	12/oct/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de reglas de transición de primera actividad del flujo.	Augusto Yépez	Realizado	Desarrollo de reglas de transición	40	42	12/10/2015

## Sprint 12

Fecha de inicio	Fecha de finalización
13/oct/2015	19/oct/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del formulario de aprobación	Augusto Yépez	Realizado	Diseño de formulario con el UI Designer de Bonita BPM	8	8	16/10/2015

de actividades diarias.			Generación de widget personalizado para botón de envío de datos	18	20	
Implementación de lógica de negocio para aprobación de actividades diarias.	Augusto Yépez	Realizado	Implementación de lógica de negocio para aprobación de actividades	20	18	19/10/2015

### Sprint 13

Fecha de inicio	Fecha de finalización
20/oct/2015	26/oct/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Implementación de lógica de negocio para aprobación de actividades diarias.	Augusto Yépez	Realizado	Implementación de lógica de negocio para aprobación de actividades	38	36	24/10/2015
Construcción de plantilla de correo de	Augusto Yépez	Realizado	Lógica dinámica para obtener datos	10	8	26/10/2015

aprobación de actividad diaria.			del colaborador y supervisor			
			Construcción de la plantilla de correo	4	4	

### Sprint 14

Fecha de inicio	Fecha de finalización
27/oct/2015	02/nov/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de reglas de transición de aprobación de actividad diaria.	Augusto Yépez	Realizado	Implementación de reglas de transición	40	46	02/11/2015

### Sprint 15

Fecha de inicio	Fecha de finalización
03/nov/2015	09/nov/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del formulario de corrección de actividades diarias.	Augusto Yépez	Realizado	Modelado e implementación en UI Designer de Bonita BPM	12	14	06/11/2015
			Implementación de widget personalizado para envío de datos	20	18	
Implementación de lógica de negocio para corrección de actividades diarias.	Augusto Yépez	Realizado	Lógica de negocio para corrección de actividades diaria	18	20	09/11/2015

### Sprint 16

Fecha de inicio	Fecha de finalización
10/nov/2015	16/nov/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Implementación de lógica de negocio para corrección de actividades diarias.	Augusto Yépez	Realizado	Lógica de negocio para corrección de actividades diaria	30	28	13/11/2015

Construcción de plantilla de correo para corrección de actividad diaria.	Augusto Yépez	Realizado	Lógica dinámica para obtener datos de los supervisores	16	16	16/11/2015
			Construcción de la plantilla de correo	8	8	

### Sprint 17

Fecha de inicio	Fecha de finalización
17/nov/2015	23/nov/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de reglas de transición de corrección de actividad diaria	Augusto Yépez	Realizado	Desarrollo de reglas de transición de corrección de actividad diaria	40	36	23/11/2015

### Sprint 18

Fecha de inicio	Fecha de finalización
24/nov/2015	30/nov/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Pruebas funcionales flujo de ingreso de actividades para operaciones y proyectos	Augusto Yépez	Realizado	Pruebas flujo de operaciones y proyectos	16	14	26/11/2015
Modelado del flujo de aprobaciones gerenciales para operaciones y proyectos.	Augusto Yépez	Realizado	Modelado del flujo de aprobaciones gerenciales	32	28	30/11/2015

### Sprint 19

Fecha de inicio	Fecha de finalización
01/dic/2015	07/dic/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
---------------------	-------------	--------	-------	-----------------	------------------	----------------------

Construcción de la vista resumen mensual por colaborador de aprobación de gerente de campo.	Augusto Yépez	Realizado	Diseño e implementación en UI Designer de Bonita BPM	24	24	07/12/2015
			Creación de widget personalizado para envío de datos	32	32	

### Sprint 20

Fecha de inicio	Fecha de finalización
08/dic/2015	14/dic/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Validaciones de la vista resumen de aprobación del gerente de campo.	Augusto Yépez	Realizado	Diseño e implementación en UI Designer de Bonita BPM	24	20	14/12/2015
			Creación de widget personalizado para envío de datos	36	32	

### Sprint 21

Fecha de inicio	Fecha de finalización
15/dic/2015	21/dic/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de negocio para vista resumen de aprobación gerente de campo.	Augusto Yépez	Realizado	Reglas de negocio para vista resumen de aprobación gerente de campo	40	40	21/12/2015

### Sprint 22

Fecha de inicio	Fecha de finalización
22/dic/2015	28/dic/2015

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de negocio para vista resumen de aprobación	Augusto Yépez	Realizado	Reglas de negocio para vista resumen de aprobación gerente de campo	40	40	24/12/2015

gerente de campo.						
Desarrollo de plantilla de notificación aprobación de gerente de campo.	Augusto Yépez	Realizado	Lógica dinámica para obtener datos de los gerentes	18	18	28/12/2015
			Construcción de la plantilla de correo	8	8	

### Sprint 23

Fecha de inicio	Fecha de finalización
29/dic/2015	04/ene/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de transición de aprobación del gerente de campo.	Augusto Yépez	Realizado	Desarrollo reglas de transición de aprobación del gerente de campo	40	40	04/01/2016

## Sprint 24

Fecha de inicio	Fecha de finalización
05/ene/2016	11/ene/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción de la vista corrección de actividades.	Augusto Yépez	Realizado	Modelado implementación usando el UI Designer de Bonita BPM	10	12	08/01/2016
			Implementación de widget personalizado para el envío de datos	16	16	
Validaciones vista corrección de actividades	Augusto Yépez	Realizado	Validaciones vista corrección de actividades	14	16	11/01/2016

## Sprint 25

Fecha de inicio	Fecha de finalización
12/ene/2016	18/ene/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
---------------------	-------------	--------	-------	-----------------	------------------	----------------------

Desarrollo de reglas de negocio para vista corrección de actividades.	Augusto Yépez	Realizado	Reglas de negocio para vista corrección de actividades	36	34	16/01/2016
Desarrollo de plantilla de notificación de corrección de actividades.	Augusto Yépez	Realizado	Lógica dinámica para obtener datos de los gerentes	10	10	18/01/2016
			Construcción de la plantilla de correo	6	4	

### Sprint 26

Fecha de inicio	Fecha de finalización
19/ene/2016	25/ene/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de transición de corrección de actividades.	Augusto Yépez	Realizado	R reglas de transición de corrección de actividades	28	28	22/01/2016

Construcción de la vista resumen mensual por colaborador de aprobación de gerente de operaciones.	Augusto Yépez	Realizado	Modelado e implementación de UI Designer de Bonita BPM	20	22	25/01/2016
---	---------------	-----------	--	----	----	------------

### Sprint 27

Fecha de inicio	Fecha de finalización
26/ene/2016	01/feb/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción de la vista resumen mensual por colaborador de aprobación de gerente de operaciones.	Augusto Yépez	Realizado	Construcción de widget personalizado para el envío de datos	32	32	29/01/2016
Validaciones de la vista resumen de aprobación del gerente de campo.	Augusto Yépez	Realizado	Reglas de validaciones para la vista de aprobación de gerente de campo	20	18	01/02/2016

### Sprint 28

Fecha de inicio	Fecha de finalización
02/feb/2016	08/feb/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo reglas de negocio para vista	Augusto Yépez	Realizado	Reglas de negocio para vista resumen de	40	40	08/02/2016

resumen de aprobación gerente de operaciones.			aprobación gerente de operaciones			
---	--	--	-----------------------------------	--	--	--

### Sprint 29

Fecha de inicio	Fecha de finalización
09/feb/2016	15/feb/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de plantilla de notificación aprobación de gerente de campo.	Augusto Yépez	Realizado	Lógica dinámica para obtener datos de los gerentes	10	8	10/02/2016
			Construcción de la plantilla de correo	4	4	
Desarrollo reglas de transición de aprobación del gerente de campo.	Augusto Yépez	Realizado	Desarrollo reglas de transición de aprobación del gerente de campo	36	36	15/02/2016

### Sprint 30

Fecha de inicio	Fecha de finalización
16/feb/2016	22/feb/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de la integración por transferencia de archivos con el sistema SQUARENET	Augusto Yépez	Realizado	Obtención de los datos desde la base de datos mediante el componente	24	24	22/02/2016
			Conteo y cálculo de bonos y horas extras por colaborador	32	32	

### Sprint 31

Fecha de inicio	Fecha de finalización
23/feb/2016	29/feb/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Desarrollo de la integración por	Augusto Yépez	Realizado	Implementación de la	36	34	27/02/2016

trasferencia de archivos con el sistema SQUARENET			generación del archivo plano			
Pruebas de la integración	Augusto Yépez	Realizado	Pruebas de la integración	10	9	29/02/2016

### Sprint 32

Fecha de inicio	Fecha de finalización
01/mar/2016	07/mar/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Pruebas del flujo de aprobaciones gerenciales.	Augusto Yépez	Realizado	Pruebas del flujo de aprobaciones gerenciales	10	10	03/03/2016
Subida a producción	Augusto Yépez	Realizado		8	8	07/03/2016

### Sprint 33

Fecha de inicio	Fecha de finalización
27/sep/2016	03/oct/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
---------------------	-------------	--------	-------	-----------------	------------------	----------------------

Instalación de eclipse para desarrollo de módulo de administración y reportes.	Augusto Yépez	Realizado	Instalación de eclipse	8	5	27/09/2016
Construcción del modelo del módulo administración.	Augusto Yépez	Realizado	Diseño e implementación del modelo para el módulo de administración	44	42	03/10/2016

### Sprint 34

Fecha de inicio	Fecha de finalización
04/oct/2016	10/oct/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción de la vista del módulo administración.	Augusto Yépez	Realizado	Implementación de la vista del módulo de administración	20	86	06/10/2016
Construcción del controlador	Augusto Yépez	Realizado	Implementación de las funciones para	32	32	10/10/2016

del módulo administración.			el módulo de administración			
----------------------------	--	--	-----------------------------	--	--	--

### Sprint 35

Fecha de inicio	Fecha de finalización
11/oct/2016	17/oct/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del controlador del módulo administración.	Augusto Yépez	Realizado	Implementación de las funciones para el módulo de administración	16	16	12/10/2016
Integración módulo de administración al sistema.	Augusto Yépez	Realizado	Exportación de componentes	4	4	15/10/2016
			Agregar nueva aplicación desde un perfil administrador	16	16	
Pruebas del módulo de administración.	Augusto Yépez	Realizado	Pruebas del módulo de administración	10	9	17/10/2016

### Sprint 36

Fecha de inicio	Fecha de finalización
-----------------	-----------------------

18/oct/2016	24/oct/2016
-------------	-------------

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del modelo del módulo reporte.	Augusto Yépez	Realizado	Construcción del modelo del módulo reporte	36	36	21/10/2016
Construcción de la vista del módulo reporte.	Augusto Yépez	Realizado	Construcción de la vista del módulo reporte	20	24	24/10/2016

### Sprint 37

Fecha de inicio	Fecha de finalización
25/oct/2016	30/oct/2016

Tarea Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Construcción del controlador del módulo de reporte.	Augusto Yépez	Realizado	Construcción del controlador del módulo de reporte	50	48	30/10/2016

### Sprint 38

Fecha de inicio	Fecha de finalización
31/oct/2016	06/nov/2016

Historia de usuario	Responsable	Estado	Tarea	Horas Planeadas	Horas Realizadas	Fecha de culminación
Integración módulo de reporte al sistema.	Augusto Yépez	Realizado	Exportación de componentes	6	4	02/11/2016
			Agregar nueva aplicación desde un perfil administrador	16	16	
Pruebas del módulo de reporte.	Augusto Yépez	Realizado	Pruebas del módulo de reporte	32	28	06/11/2016

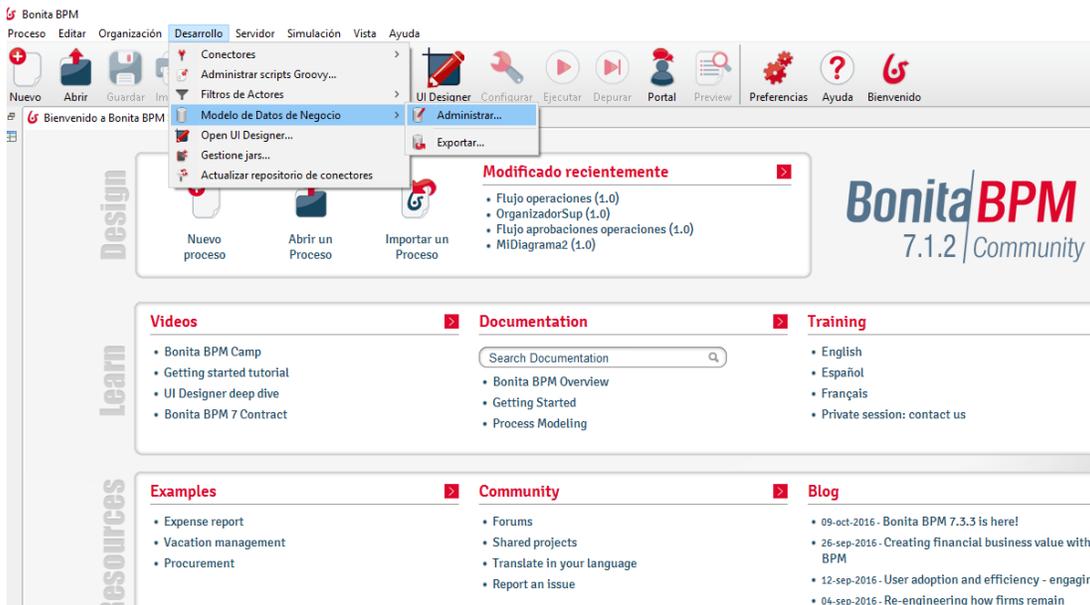
# Anexo 3 Manual técnico y de instalación del sistema

## 1. Pasos para la exportacion a produccion

### 1.1 Paso 1

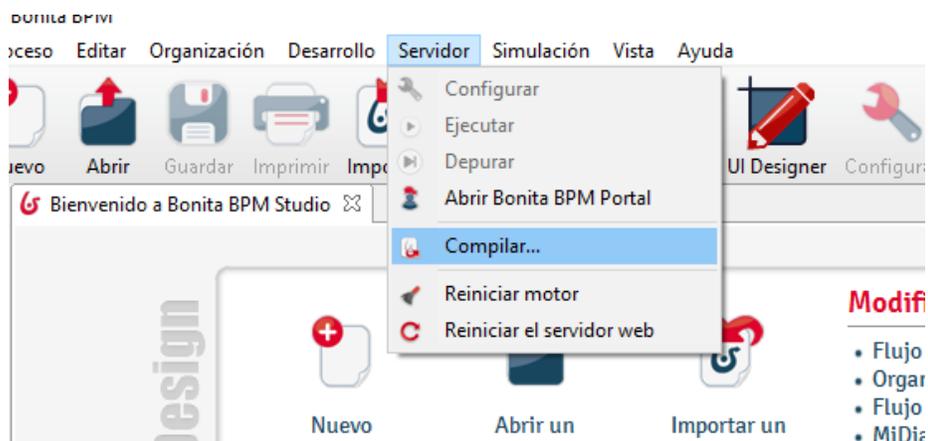
El primer paso a realizar una vez terminado el desarrollo es extraer la estructura de la Base del modelo de datos del negocio. Para esto dentro del editor de Bonita BPM Studio, nos vamos al menú Desarrollo y luego seleccionamos la opción “Modelo de Datos de Negocio”, con su sub opción “Exportar”.

Una vez que se dé clic en “Exportar”, Bonita BPM nos generara un archivo XML comprimido como zip con la estructura de la base de datos que posteriormente en producción tendrá que ser importada.



## 1.2 Paso 2

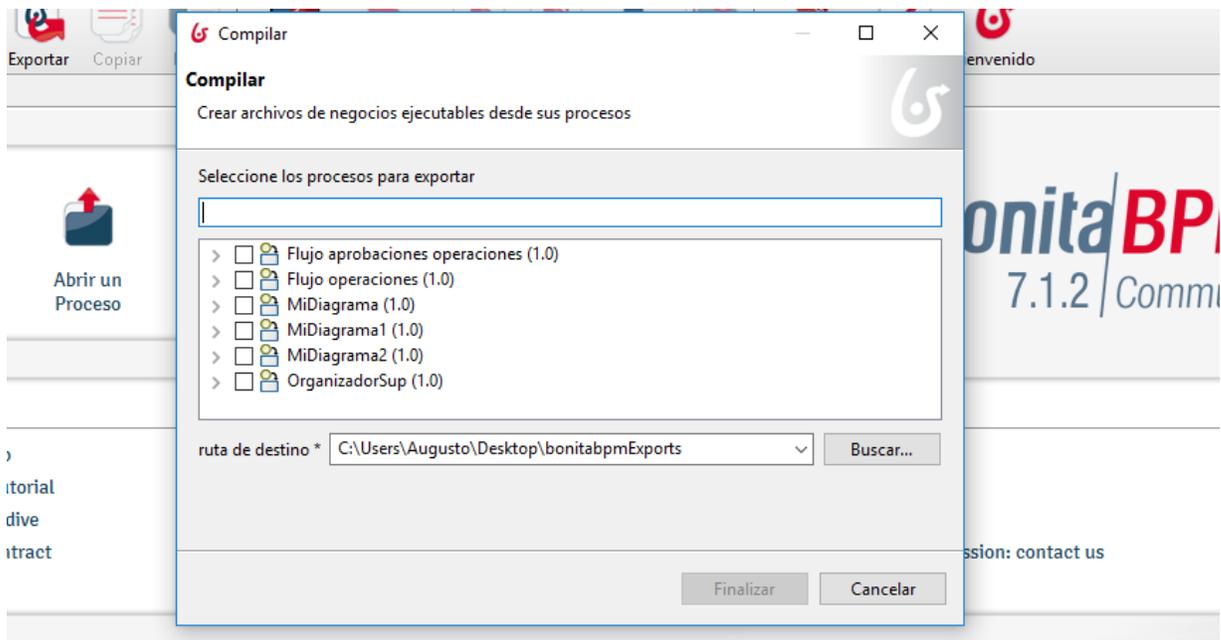
Como siguiente paso se deben exportar los flujos, dando clic en la opción “Servidor”, y seleccionando el submenú “Compilar” dentro de Bonita BPM Studio.



Se nos abra una ventana con la opción de los flujos a compilar. Se deben seleccionar los tres flujos relacionados para el proceso de pago de bonos y horas extras, los cuales son:

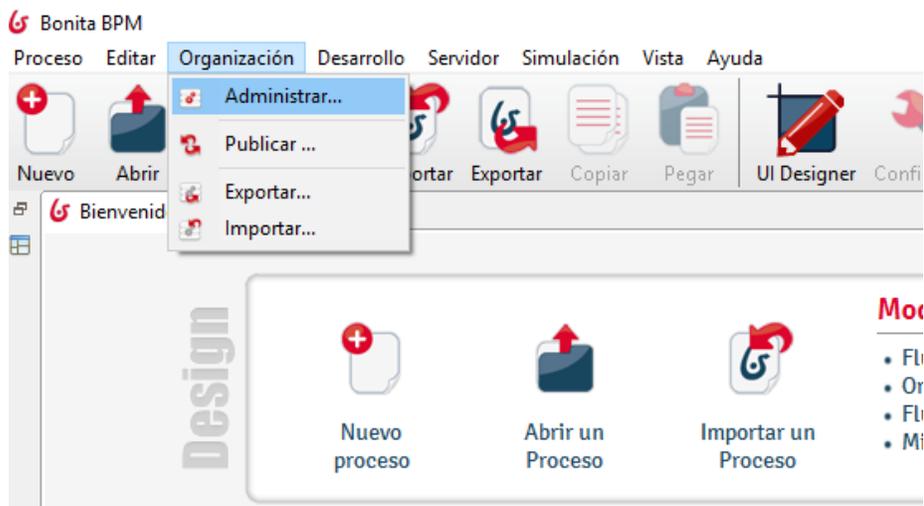
- Flujo operaciones
- Flujo aprobaicones operaciones
- OrganizadorSup

Una vez finalizada la exportación se crearan tres archivos con extensiones .bar, los cuales se deberán importar desde la plataforma de administración de Bonita BPM en producción.



### 1.3 Paso 3

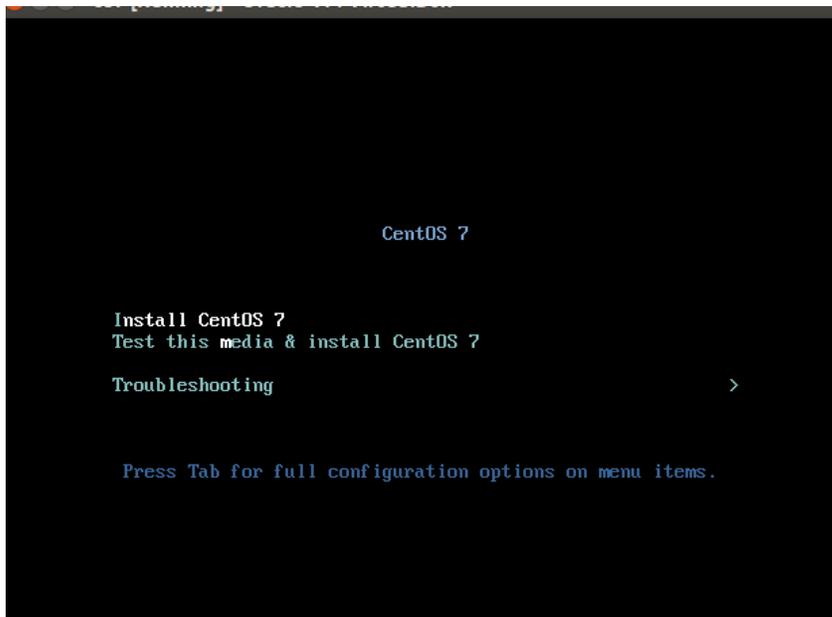
Como tercer paso y paso final en la exportación de componentes del Bonita BPM Studio, se debe ir al menú "Organización", y dar clic en la opción "Administrar", para poder exportar todos los usuarios y perfiles disponibles, para su posterior importación desde la administración de la plataforma.



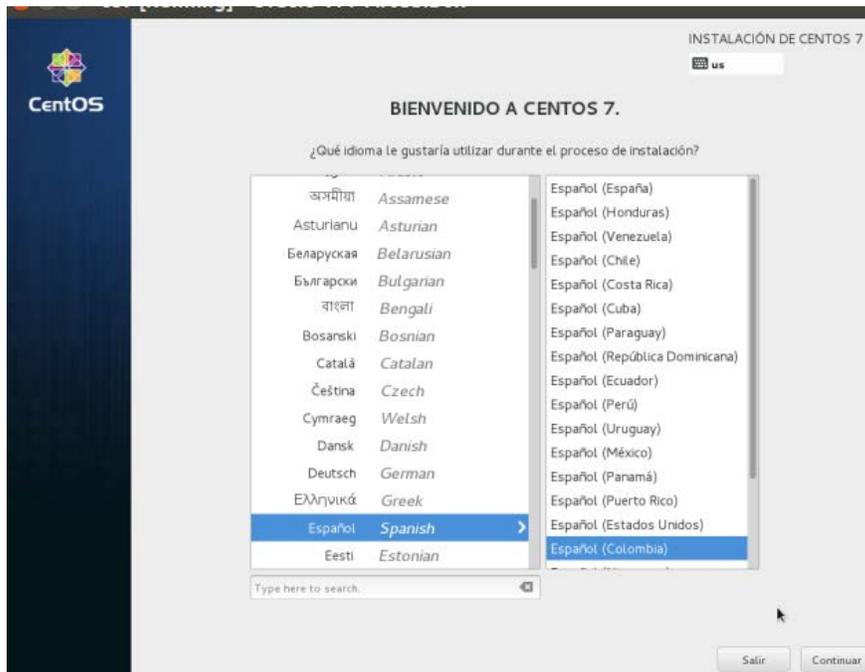
## 2. Instalación de servidor centos 7 y creacion de la base de datos

### 2.1 Instalación de Linux Centos 7

Lo primero que se realiza es ejecutar el instalador de Linux Centos 7.



Posteriormente seleccionar el idioma de instalación.



Después de dar clic en el botón “Continuar”, aparecerá una pantalla de Resumen de Instalación, en donde se deberá configurar el destino de la instalación, dando clic al botón “Destino de la instalación”.

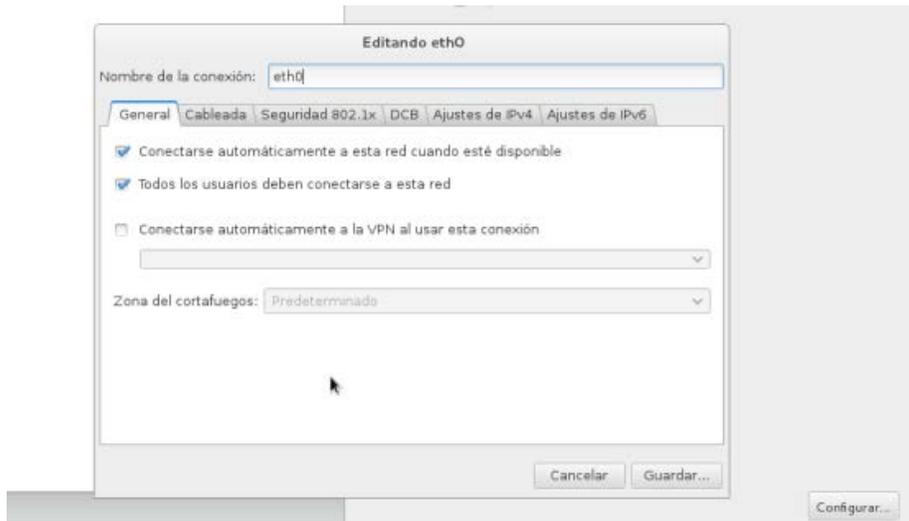


Una vez en la pantalla “Destino de la instalación”, se debe seleccionar el disco donde se va a instalar el sistema operativo y posterior a esto se debe dar clic en el botón “Listo” para regresar a la ventana anterior.

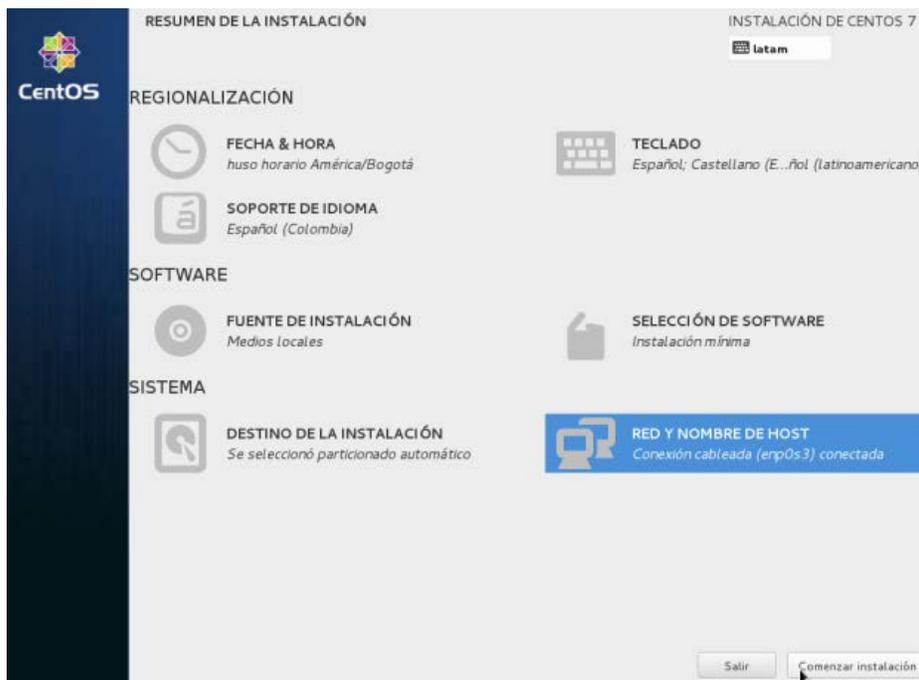
Después debemos configurar la Red y nombre de host, por lo de damos clic en la opción “Red y Nombre de Host”.



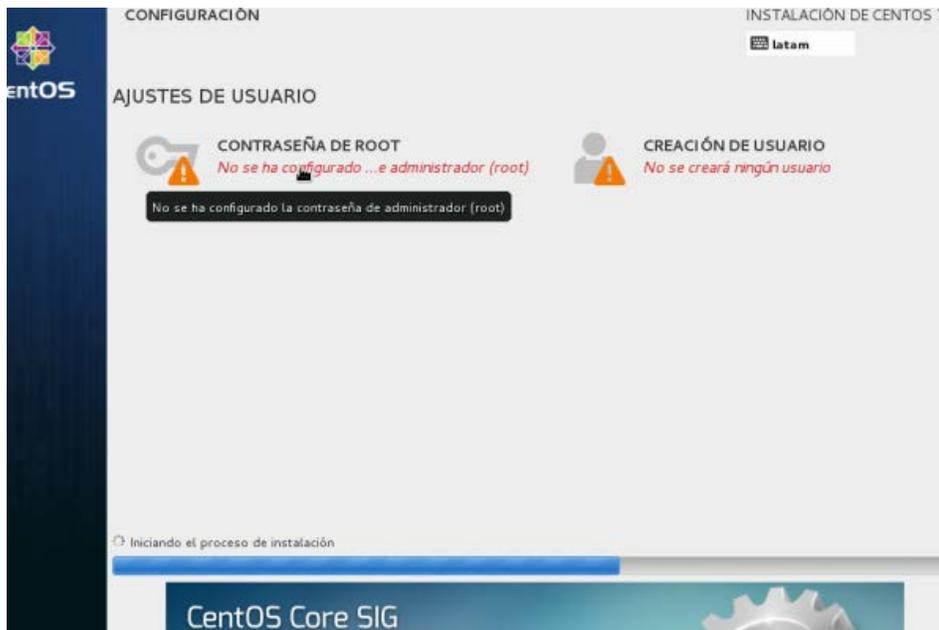
Ya en la ventana de Red y nombre de host, debemos configurar el nombre del host y luego damos clic al botón “Configurar”. Nos saldrá otra ventana donde debemos activar la opción “Conectarse automáticamente a esta red cuando esté disponible”, para finalmente dar clic en el botón “Guardar”.



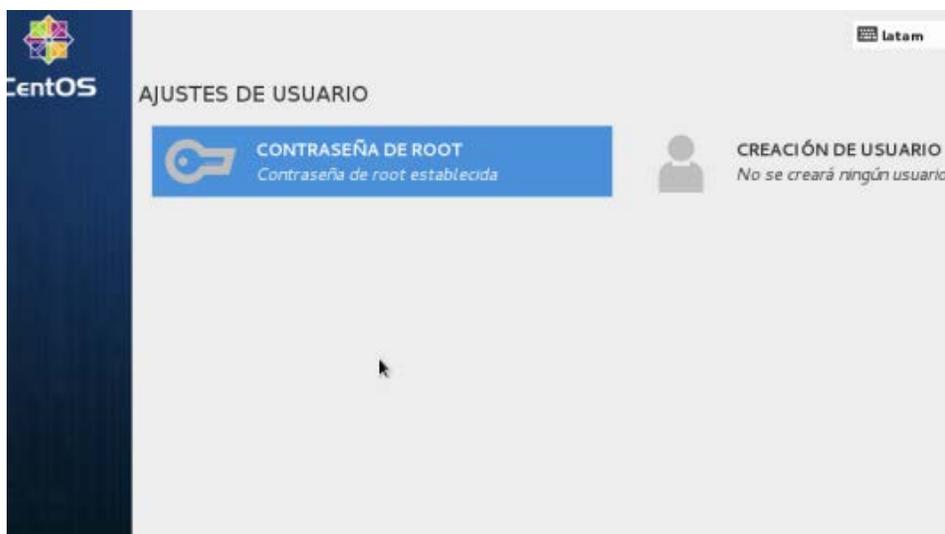
Una vez realizadas estas configuraciones ya podemos iniciar el proceso de instalación dando clic en el botón “Comenzar instalación”.



El instalador empieza a cargar el sistema operativo y nos muestra una ventana de ajustes de usuario, en donde debemos por seguridad configurar una contraseña para el usuario root del sistema operativo.

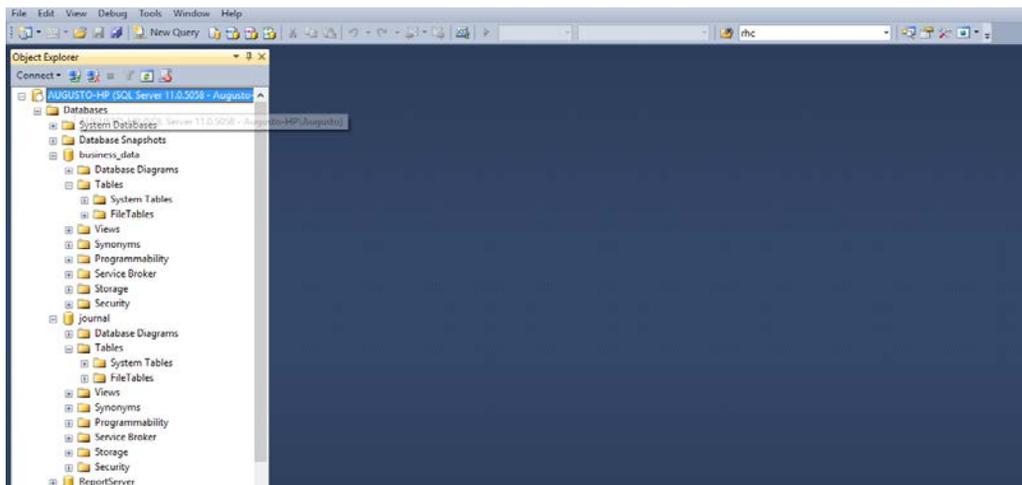


Finalmente ponemos una contraseña para el usuario root y esperamos que el instalador termine su proceso para poder acceder al sistema operativo.



## 2.2 Creación y configuración de la base de datos

Para la creación de la base de datos nos podemos apoyar en el SQL Server Management Studio. Lo abrimos y damos clic derecho en “Databases”, y seleccionamos la opción “New Data Base”, y le damos en el nombre de “business\_data”. Posterior a esto creamos otra base de datos que se llamara “journal”, las cuales serán usadas para el Modelo de datos de negocio y la otra para las tablas por defecto de Bonita BPM.



Posterior a la creación de las bases de datos se debe seguir una serie de pasos para la correcta configuración de SQL Server para que funcione con Bonita BPM. Estos pasos se los puede encontrar en la documentación de Bonita.

Bonitasoft Documentation - Legacy - 7.3+ 7.2 7.1 7.0 6.5 6.4 6.3 6.2 6.1 6.0 5.X

Search

**Bonita BPM 7.1**

- Application and Process Design
  - Installation
    - Bonita BPM installation overview
    - Bonita BPM Studio installation
    - Basic Bonita BPM platform installation
      - Hardware and software requirements
      - JBoss bundle
      - Tomcat bundle
      - Deploy bundle
      - Database configuration
        - Database configuration for business data
        - Bonita Home
        - Licenses
        - First steps after setup
    - Advanced Bonita BPM platform installation
    - Platform installation examples
      - Back up Bonita BPM Platform
    - Migration
      - Upgrade from Community to a Subscription edition
  - Bonita BPM Portal Administration
  - Development

**SQL Server**

XA Transactions

To support XA transactions, SQL Server requires a specific configuration. You can refer to MSDN for more information.

Here is the list of steps to perform (as an example, the database name BONITA\_BPM is used):

1. Make sure you have already downloaded and installed the Microsoft SQL Server JDBC Driver 4.0.
2. Copy the sqljdbc\_\*.jar from %JDBC\_DRIVER\_INSTALL\_ROOT%\sqljdbc\_4.0\enu\xa\ (x64 for 64 bit version of Windows, x86 for 32 bit version of Windows) to %SQLSERVER\_INSTALL\_ROOT%\Instance\_root\%SQLSERVER%\SQLSERVER\%SQLSERVER%\.
3. Copy/paste the content of xa\_install.sql file (located in %JDBC\_DRIVER\_INSTALL\_ROOT%\sqljdbc\_4.0\enu\xa) to SQL Server Management Studio's Query Editor.
4. Execute the query in the Query Editor.
5. To confirm successful execution of the script, open the "Object Explorer" and go to: Master > Programmability > Extended Stored Procedures. You should have 12 new procedures, each with a name starting with dbo.xp\_sqljdbc\_xa\_.
6. Assign the new role 'SqlJDBCXAUser' to the user who owns the Bonita BPM Engine database (bonitadev in our example). To do this, execute the following commands in SQL editor:
 

```
USE master;
GO

CREATE LOGIN bonitadev WITH PASSWORD = 'secret_password';
GO

CREATE USER bonitadev FOR LOGIN bonitadev;
GO

EXEC sp_addrolemember [SqlJDBCXAUser], 'bonitadev';
GO
```
7. In the Windows "Start" menu, select Administrative Tools > Services.
8. In the "Services" window, make sure that the Distributed Transaction Coordinator service is set to start automatically. If it's not yet started, start it.
9. Make sure that the other services it depends on, namely "Remote Procedure Call" and "Security Accounts Manager", are also set to start automatically.
10. Run the dcomcnfg command, or go to the "Start" menu, then Administrative Tools > Component Services.
11. In the left navigation pane, navigate to Component Services > Computers > My Computer > Distributed Transaction Coordinator.
12. Select and right-click on Local DTC and then Properties.
13. Click on Security tab. Ensure that the checkbox for Enable XA Transactions is checked.
14. Click Apply, then click OK.
15. Then stop and restart SQLServer.
16. Create the BONITA\_BPM database: CREATE DATABASE BONITA\_BPM GO.
17. Set bonitadev as owner of BONITA\_BPM database (use, for example, 'Microsoft SQL Management Studio')

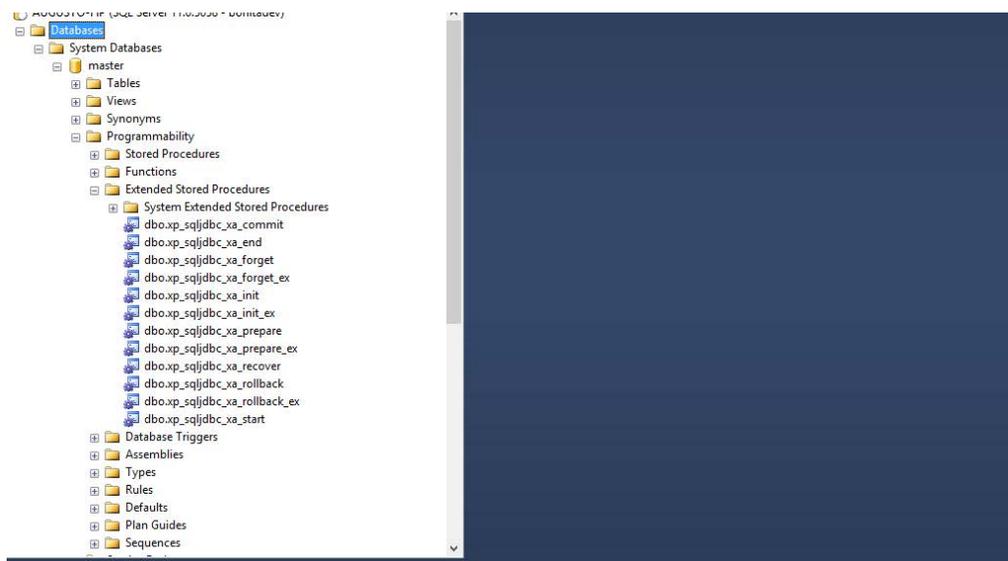
Recommended configuration for lock management

Run the script below to avoid deadlocks:

```
ALTER DATABASE BONITA_BPM SET SINGLE_USER WITH ROLLBACK IMMEDIATE
```

BACK TO TOP

Una vez seguidos esto paso se debe verificar que se hayan creado correctamente los Stored Procedures y el usuario del Bonita BPM Engine.

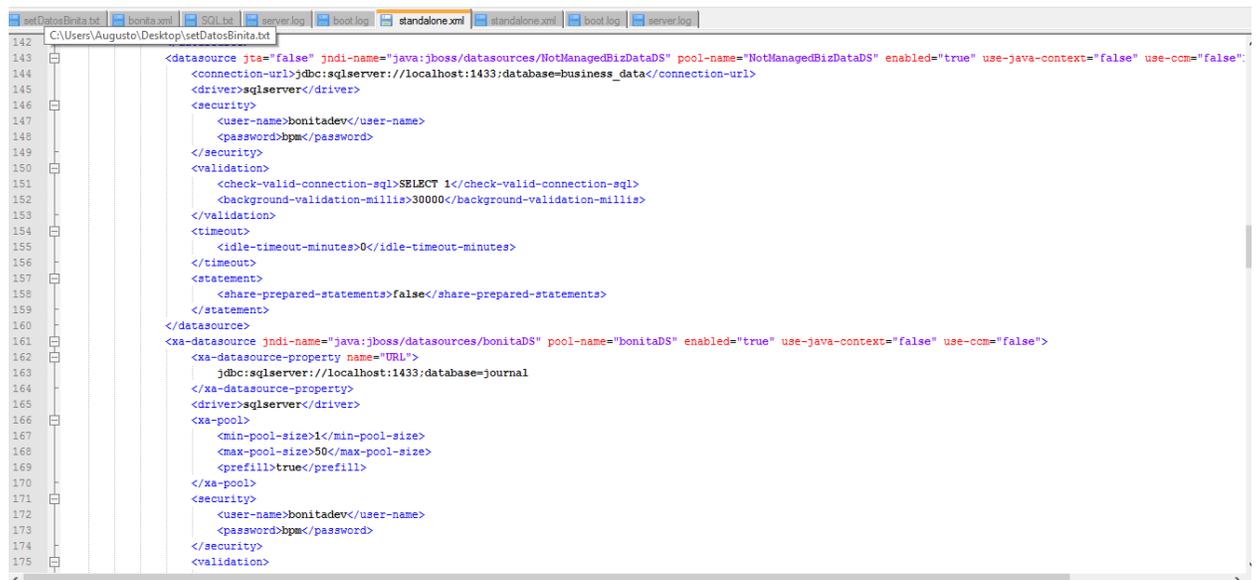


### 3. Configuración de Bonita BPM con JBoss bundle

### 3.1 Configuración de archivo standalone.xml

Una vez descargado del sitio oficial de BonitaSoft el JBoss bundle, se lo debe subir al servidor vía FTP, para su configuración.

Se requiere configurar el archivo sandalone.xml, para que el funcione con la configuración previa de SQL Server, por lo cual cambiamos la cadena de conexión, el nombre de usuario y la contraseña de la base de datos.



```
142 <datasource jta="false" jndi-name="java:jboss/datasources/NotManagedBizDataDS" pool-name="NotManagedBizDataDS" enabled="true" use-java-context="false" use-ccm="false">
143 <connection-url>jdbc:sqlserver://localhost:1433:database=business_data</connection-url>
144 <driver>sqlserver</driver>
145 <security>
146 <user-name>bonitadev</user-name>
147 <password>bpw</password>
148 </security>
149 <validation>
150 <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
151 <background-validation-millis>30000</background-validation-millis>
152 </validation>
153 <timeout>
154 <idle-timeout-minutes>0</idle-timeout-minutes>
155 </timeout>
156 <statement>
157 <share-prepared-statements>>false</share-prepared-statements>
158 </statement>
159 </datasource>
160 <xa-datasource jndi-name="java:jboss/datasources/bonitaDS" pool-name="bonitaDS" enabled="true" use-java-context="false" use-ccm="false">
161 <xa-datasource-property name="URL">
162 jdbc:sqlserver://localhost:1433:database=journal
163 </xa-datasource-property>
164 <driver>sqlserver</driver>
165 <xa-pool>
166 <min-pool-size>1</min-pool-size>
167 <max-pool-size>50</max-pool-size>
168 <prefill>true</prefill>
169 </xa-pool>
170 <security>
171 <user-name>bonitadev</user-name>
172 <password>bpw</password>
173 </security>
174 <validation>
175
```

### 3.2 Configuración de driver de conexión con SQL Server

Para una correcta configuración del driver de conexión se debe crear la siguiente estructura dentro de JBoss\_Home\modules\com\sqlserver\main. Copiamos el

driver de conexión dentro de la carpeta main y creamos un archivo module.xml con la siguiente estructura:

```
<module xmlns="urn:jboss:module:1.0" name="com.sqlserver">
  <resources>
    <resource-root path="sqljdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

Esto servirá para enlazar a JBoss con el driver de SQL server.



Finalmente el único paso restante es el levantar el servidor JBoss, para lo cual se ejecutamos vía comandos la sentencia “sh standalone.sh”, dentro de JBoss\_home\bin. Con esto levantaremos al servidor de aplicaciones JBoss.

A screenshot of a Windows file explorer window showing the contents of the JBoss bin directory. The path is 'Local Disk (C:) > BonitaBPM2 > bin'. The table below lists the files and their properties.

name	Date modified	Type	Size
add-user.sh	3/10/2012 12:14 AM	Shell Script	2 KB
appclient.bat	3/10/2012 12:14 AM	Windows Batch File	3 KB
appclient.conf	3/10/2012 12:14 AM	CONF File	2 KB
appclient.conf.bat	3/10/2012 12:14 AM	Windows Batch File	3 KB
appclient.sh	3/10/2012 12:14 AM	Shell Script	4 KB
domain.bat	3/10/2012 12:14 AM	Windows Batch File	4 KB
domain.conf	3/10/2012 12:14 AM	CONF File	3 KB
domain.conf.bat	3/10/2012 12:14 AM	Windows Batch File	4 KB
domain.sh	3/10/2012 12:14 AM	Shell Script	8 KB
jboss-cli.bat	3/10/2012 12:14 AM	Windows Batch File	2 KB
jboss-cli.sh	3/10/2012 12:14 AM	Shell Script	2 KB
jboss-cli.xml	3/10/2012 12:14 AM	XML Document	1 KB
jconsole.bat	3/10/2012 12:14 AM	Windows Batch File	3 KB
jconsole.sh	3/10/2012 12:14 AM	Shell Script	3 KB
jdr.bat	3/10/2012 12:14 AM	Windows Batch File	3 KB
jdr.sh	3/10/2012 12:14 AM	Shell Script	3 KB
run.bat	3/10/2012 12:14 AM	Windows Batch File	1 KB
run.sh	3/10/2012 12:14 AM	Shell Script	1 KB
standalone.bat	3/10/2012 12:14 AM	Windows Batch File	5 KB
standalone.conf	10/6/2015 4:45 PM	CONF File	3 KB
standalone.conf.bat	10/6/2015 4:45 PM	Windows Batch File	4 KB
standalone.sh	3/10/2012 12:14 AM	Shell Script	8 KB
vault.bat	3/10/2012 12:14 AM	Windows Batch File	2 KB

```
add-user.sh 3/10/2012 12:14 AM Shell Script
C:\WINDOWS\System32\cmd.exe
Calling "C:\BonitaBPM2\bin\standalone.conf.bat"
=====
JBoss Bootstrap Environment
JBOSS_HOME: C:\BonitaBPM2
JAVA: C:\Program Files\Java\jdk1.7.0_13\bin\java
JAVA_OPTS: -XX:+TieredCompilation -Dprogram.name=standalone.bat -Xms1024M -Xmx1024M -XX:MaxPermSize=256M -XX:+HeapDumpOnOutOfMemoryError -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djboss.server.default.config=standalone.xml -Dbonita.home="C:\BonitaBPM2\bin\..\bonita" -Dfile.encoding=UTF-8
=====
21:30:56,448 INFO [org.jboss.modules] JBoss Modules version 1.1.1.GA
21:31:00,263 INFO [org.jboss.msc] JBoss MSC version 1.0.2.GA
21:31:00,342 INFO [org.jboss.as] JBAS015899: JBoss AS 7.1.1.Final "Brontes" starting
```

## 4. Configuración de Bonita BPM Portal

### 4.1 Importación del modelo de datos del negocio

Una vez levantado el servidor de aplicaciones JBoss, se debe ir a la URL [bpm.sertecpet.net/bonita](http://bpm.sertecpet.net/bonita), para abrir el portal, y en el nombre de usuario y contraseña colocar install, ya que este es un usuario creado por defecto para la configuración inicial del portal.

Una vez logueado, damos clic en el menú “Servicios BPM”, y pausamos los servicios dando clic en el botón “Pausar”.



Después de pausados o detenidos los servicios, nos dirigimos a la ventana de “Modelo de Datos de Negocio”, para poder importar y activar las tablas del modelo del negocio.



Una vez que el sistema valida el archivo damos clic en el botón “Activar”, con lo cual recibiremos un mensaje de éxito de la importación.



## 4.2 Importación y configuración de la organización

Para poder importar la organización se debe ir al menú y seleccionar “Organización”, y dentro de este el sub menú “Importación / exportación”.



Una vez en esa ventana se debe dar clic en el botón importar, para realizar la importación de la organización ya existente.



Si todo está bien con la importación nos mostrara un mensaje de éxito de la importación.

### Importar una organización existente

**Advertencia:** Esto importará un archivo que contiene los datos de la organización entera. Tenga cuidado, su organización se fusionará con los datos existentes. En caso de conflicto, se da prioridad a el archivo importado

SERTECPET.xml 

Organización importada con éxito.

**IMPORTAR**

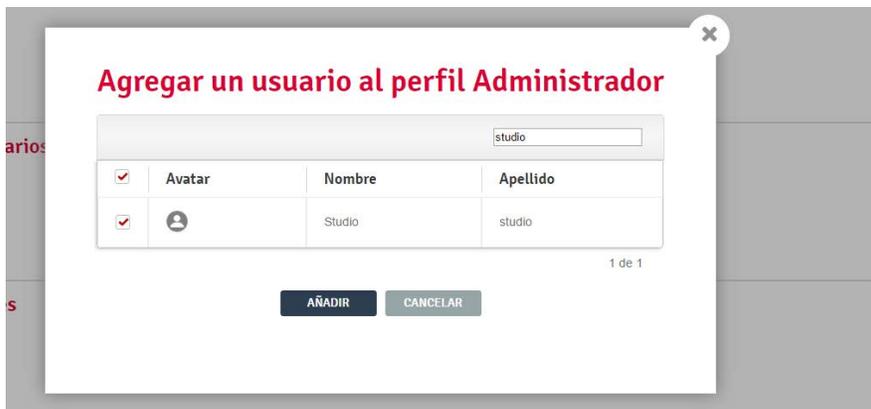
Se puede verificar que la importación fue realizada correctamente dando clic en “Organización”, “Usuarios”. Posterior a esta verificación se deben configurar los perfiles de usuario y administrador del sistema, para lo cual damos clic en el menú “Organización”, y luego en “Perfiles”.



Una vez en esta ventana se debe dar clic en cada uno de los perfiles, para poder configurarlos para cada usuario del sistema.

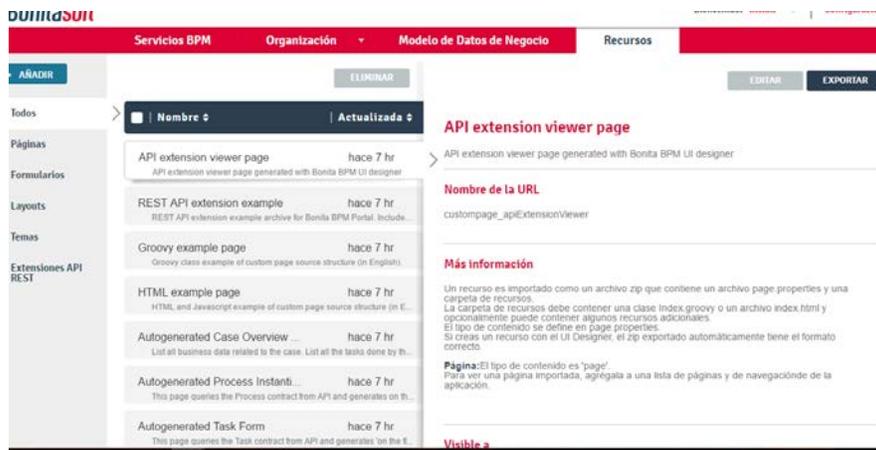


Posterior a esto se asignan los perfiles de administrador y usuario y sus respectivas membresías a todos los miembros de la organización, para que puedan acceder al sistema.

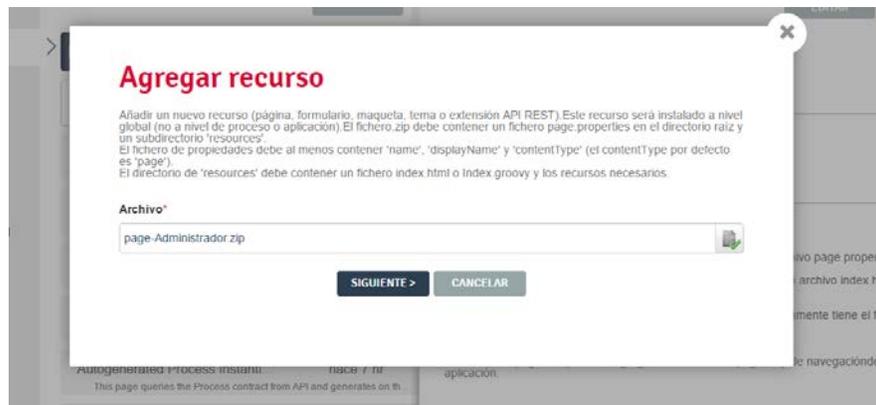


### 4.3 Importación y configuración de módulo de administración

Para realizar la importación del módulo de administración se debe ir al menú “Recursos”, y dar clic en el botón “Añadir”.



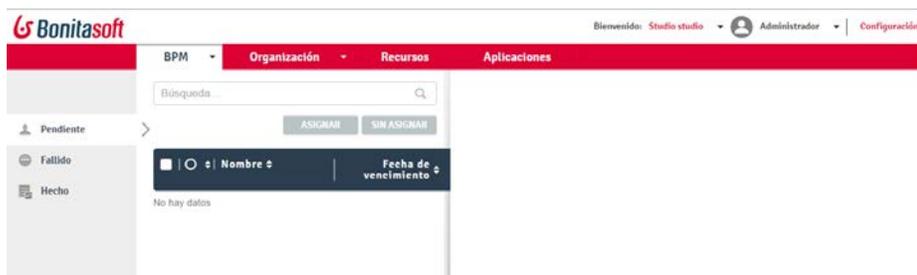
Una vez que el sistema valida que el archivo importado del módulo de administración está correcto, damos clic en el botón “Siguiente”, para terminar la importación.



Una vez terminada la importación se deben reanudar los servicios BPM, para lo cual damos clic en el menú “Servicios BPM” y posteriormente damos clic en “Reanudar”.



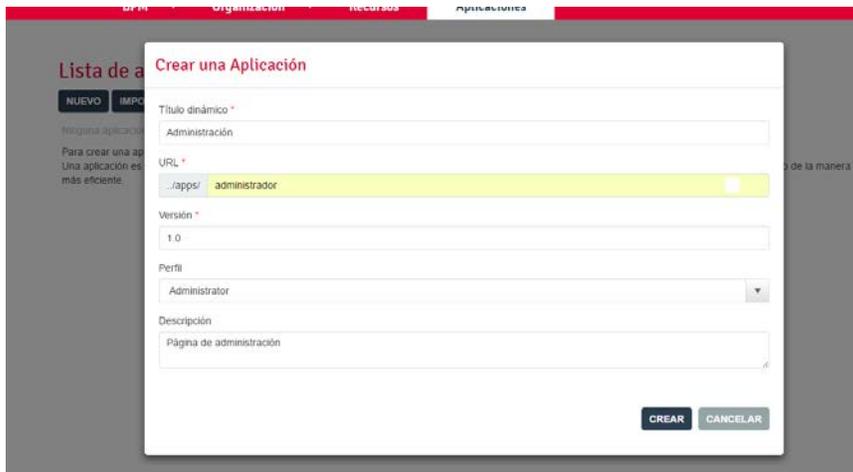
Posterior a esto nos salimos del sistema y nos volvemos a logear con el usuario Studio el cual tiene privilegios de administrador.



Nos dirigimos al menú "Aplicaciones", y damos clic en el botón "Nuevo".



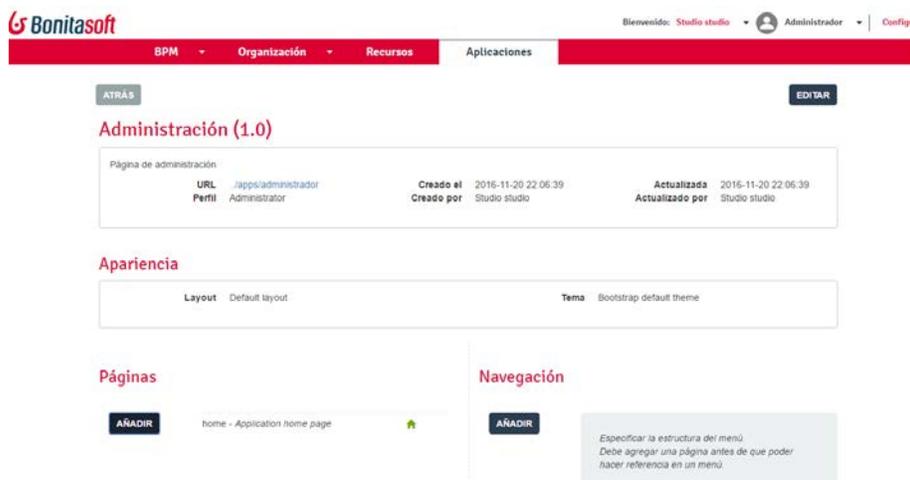
Ingresamos los datos de título, URL, perfil y descripción para nuestro módulo de administración.



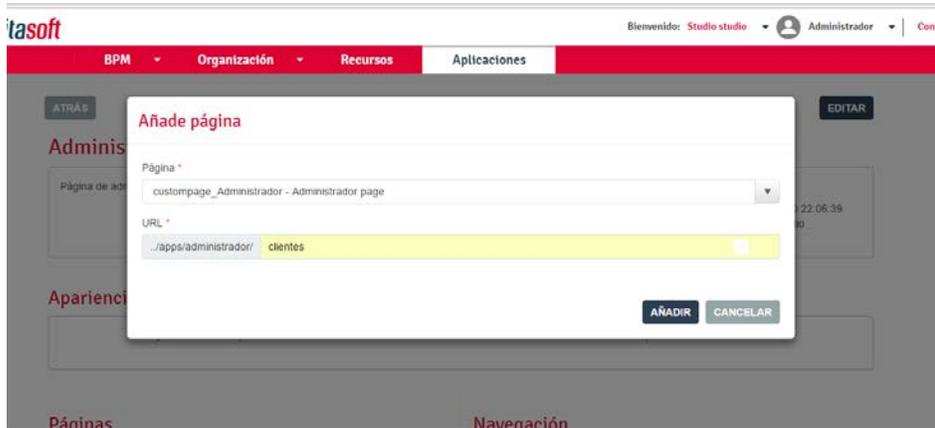
Una vez creada la aplicación se debe configurar, para lo cual se debe dar clic en “...” dentro del menú “acciones”.



Posterior a esto añadimos una página dando clic en el botón “Añadir”, dentro de la pestaña “Paginas”



Aquí seleccionamos la página previamente importada en los recursos, y se creamos una URL de acceso.



Finalmente podemos probar la página del módulo de administración.



## MANTENIMIENTO CLIENTE

Id Cliente	Descripcion	
1	PETROAMAZONAS EP33	<a href="#">Edit</a> <a href="#">Delete</a>
2	RIO NAPO	<a href="#">Edit</a> <a href="#">Delete</a>
3	PETROECUADOR	<a href="#">Edit</a> <a href="#">Delete</a>
4	PARDALIS	<a href="#">Edit</a> <a href="#">Delete</a>
5	REPSOL	<a href="#">Edit</a> <a href="#">Delete</a>
6	GENTE OIL	<a href="#">Edit</a> <a href="#">Delete</a>
7	IGAPO-HALLIBURTON	<a href="#">Edit</a> <a href="#">Delete</a>
0	NINGUNO	<a href="#">Edit</a> <a href="#">Delete</a>
9	testttt	<a href="#">Edit</a> <a href="#">Delete</a>

1 - 9 of 9 items

# Anexo 4 Manual de usuario del sistema

## 1. Ingreso tareas diarias para los colaboradores

Los colaboradores de la compañía deben ingresar a la URL <https://bpm.sertecpet.net/bonita>, y loguearse con su respectivo nombre de usuario y contraseña.

Formulario de login con el logo de Bonitasoft a la izquierda. A la derecha, el título "Formulario de login" está sobre un campo de "Usuario" con un ícono de persona y un campo de "Contraseña" con un ícono de candado. Debajo de los campos hay un botón rojo que dice "INICIO DE SESIÓN".

Una vez dentro del sistema se deben dirigir a la opción del menú "Procesos", y dar clic en el botón "Inicio".



Se abrirá un formulario para el ingreso de actividades diarias, el cual deberá ser llenados todos los campos obligatoriamente, con excepción del campo de observaciones, e cual es un campo opcional.

**Bonitasoft** Bienvenido: CARLOS ALBERTO AIZAGA LOOR Usuario Configuración

Tareas Casos Procesos

### Registro de empleados

**Datos empleado**

Cedula: 1717875197

Fecha ingreso: 2011-08-08 00:00:00

Centro costos: WELL TESTING

Supervisor a cargo: Seleccione una opción

Colaborador: AIZAGA LOOR CARLOS ALBERTO

Fecha salida:

Cargo: OPERADOR DE WELL TESTING

Sucursal: Quito

**Registro de actividades**

Actividad: TURNO

Estado: PRODUCCION

Turno: TURNO 1

Hora de inicio: 19:00

Hora de fin: 19:00

Llegará un correo de notificación al supervisor a cargo del colaborador, notificándole que el colaborador X ha registrado sus actividades diarias, para su aprobación.

para agosto\_360

Estimado EDISON MANUEL CHICAIZA PILAMUNGA, tiene una actividad pendiente de aprobación del colaborador CARLOS ALBERTO AIZAGA LOOR en su bandeja de entrada del sistema. El número de actividad es el 26003, si requiere mas información del mismo, favor ingresar al sistema.

## 2. Aprobación de tareas diarias de los colaboradores

El siguiente paso en el flujo es la aprobación de las actividades diarias de los colaboradores, para lo cual los supervisores ingresan al sistema y revisan su bandeja de tareas la cual está al lado derecho, y es similar a la bandeja de un correo. Aquí ubicaran todas las tareas pendientes de aprobación.



El supervisor deberá seleccionar la tarea que desea aprobar y dar clic en el botón “Hacer”. Se abrirá la información previamente llenada por el colaborador, a manera de visualización, y dándole la opción de modificación de casi todos los datos de ingreso, con excepción de la fecha de ingreso y el comentario realizado por el colaborador.

Una vez revisada y validada la información el supervisor deberá aprobar o rechazar la actividad diaria, e ingresar un comentario en caso de que así lo quiera, siendo este campo no obligatorio.

Aprobación de actividades

Comentario supervisor

Aprobar

Rechazar

Si aprueba la actividad le llegara un correo al supervisor notificándole que ha aprobado la actividad del colaborador. En caso de que se rechace la actividad, le llegara un correo de notificación al colaborador que ingreso la actividad.

para agosto\_000, agosto\_000 (2)

Estimado EDISON MANUEL CHICAIZA PILAMUNGA, usted ha aprobado la actividad del colaborador CARLOS ALBERTO AIZAGA LOOR. El número de actividad es el 26003, si requiere mas información del mismo, favor ingresar al [sistema](#).

### 3. Ingreso del organizador de personal

Este punto es solo realizado por los supervisores del área de Well Testing, en el cual los supervisores deberán llenar la información de los turnos, el personal asignado a cada MTU, la ubicación del personal, para qué compañía se está realizando el trabajo y que tipo de actividad se está realizando el colaborador. Esto tiene un formato similar una hoja de Excel manejada con anterioridad por los supervisores.

Organizador de personal

Fecha: 2016-04-19

POZZ	UNIDAD	COMPAÑIA	ACTIVIDAD	TURNO 1 OPERADOR(NOCHE)	TURNO 1 TECNICO(NOCHE)	TURNO 3 OPERADOR(DIA)	TURNO 3 TECNICO(DIA)	TURNO 2 OPERAD
APAKA 11	MTU 1	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 2	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 3	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 4	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 5	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 6	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 7	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 8	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 9	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 10	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 11	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 12	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 13	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 14	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 15	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 16	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 17	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 18	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 19	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE
APAKA 11	MTU 20	NINGUNO	PRODUCCION	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE	SELECCIONE

El sistema restringirá el ingreso de actividades diarias al personal de Well Testing, sin antes haber sido llenadas por el supervisor.

#### 4. Aprobación gerencial mensual

Todos los fines de mes los gerentes de campo deberán ingresar al sistema e iniciar el proceso de aprobación de actividades. Para esto se loguean en el sistema y se dirigen al menú “Procesos”, y después deberán dar clic en el botón “Inicio”, con lo que se generara una tabla resumen y detalle de las actividades de cada uno de los colaboradores.

En la tabla resumen se mostrara los datos de Cedula, Cargo, Nombre, Centro de costos, Nro. de bonos en el mes, número de días en standby, número de días trabajados, conteo de horas extras, horas suplementarias y complementarias.

The screenshot shows a web interface with a navigation bar (Tareas, Casos, Procesos) and a title "Aprobación de actividades". Below the title is a summary table with columns: CEDULA, CARGO, NOMBRE, CENTRO DE COSTOS, NRO BONOS EN MES, NRO DIAS STAND BY, and NRO DIAS. A "Continuar" button is visible below the table.

Si el gerente de campo desea ver el detalle las actividades diarias de cierto colaborador, deberá dar doble clic encima de la fila correspondiente al colaborador o a su vez en la flecha ubicada a lado izquierdo de campo "Cedula".

The screenshot shows a detailed view of activity approval. It features a summary row for each employee and a detailed table below. The summary row for employee 1711916724 shows: OPERADOR DE WELL TESTING, ACEVEDO VALLE HERNAN IVAN, WELL TESTING (antiguo), 3 bonos, 8 días, 1 día standby, 3 días trabajados, 0 horas extras, and 3 horas complementarias. The detailed table below lists activities with columns: FECHA DE ACTIVIDAD, ACTIVIDAD, ESTADO, TIPO DE BO..., HORA DE INI..., HORA DE FIN, LOCACION, CLIENTE, and FECHA APROBACION.

FECHA DE ACTIVIDAD	ACTIVIDAD	ESTADO	TIPO DE BO...	HORA DE INI...	HORA DE FIN	LOCACION	CLIENTE	FECHA APROBACION
2016-08-25 00:00:00.0	TURNO	PRODUCC...	B	19	7	GUANTA 13	PARDALIS	2016-08-28 00:00:00.0
2016-08-26 00:00:00.0	TURNO	PRODUCC...	B	7	16	GUANTA 13	PETROAMAZONAS EP	2016-08-28 00:00:00.0
2016-08-27 00:00:00.0	TURNO	PRODUCC...	DB	12	19	APAICA 11	PETROECUADOR	2016-08-28 00:00:00.0

FECHA DE ACTIVIDAD	ACTIVIDAD	ESTADO	TIPO DE BO...	HORA DE INI...	HORA DE FIN	LOCACION	CLIENTE	FECHA APROBACION
2016-08-23 00:00:00.0	TURNO	PRODUCC...	B	19	7	APAICA 11	PETROAMAZONAS EP	2016-08-28 00:00:00.0
2016-08-24 00:00:00.0	TURNO	PRODUCC...	B	20	6	DRAGO NORTE 36	RIO NAPO	2016-08-28 00:00:00.0
2016-08-25 00:00:00.0	TURNO	PRODUCC...	DB	19	4	CONONACO 40	REPSOL	2016-08-28 00:00:00.0
2016-08-26 00:00:00.0	TURNO	PRODUCC...	B	21	7	LAGO 60	PETROECUADOR	2016-08-28 14:22:00.0
2016-08-27 00:00:00.0	TURNO	PRODUCC...	DB	20	7	LAGO 60	PARDALIS	2016-08-28 14:22:00.0

Cuando se valide esta información el gerente de campo podrá aprobar todas las actividades de los colaboradores, o de una en una o a su vez también podrá negar todas o solo ciertas actividades.

Si el gerente niega las actividades, le llegara un correo al supervisor notificándole que se ha negado ciertas actividades diarias, para lo cual el supervisor deberá corregir y volverá aprobar las actividades diarias. En caso de que todas las

actividades sean aprobadas pasara a una aprobación final por el gerente de operaciones. El gerente de operaciones vera la misma información que el gerente de campo, pero únicamente tendrá la opción de aprobar todo o rechazar todo, con lo cual se terminaría el flujo en caso de aprobación. Si el gerente de campo rechaza, esto regresara hacia el gerente de campo para su validación.

## Anexo 5 Carta de conformidad



Quito, 25 de Noviembre de 2016

Señores:

**UDLA**

Presente.-

De nuestra consideración:

Por medio de la presente, yo Dra. Silvia Campaña, en calidad de Coordinadora de Nómina y Compensaciones de SERTECPET S.A., y encargada del proyecto, quiero indicar que el software implementado por el Sr. Augusto Cesar Yépez Dávila en su trabajo de titulación, está contribuyendo con la optimización del proceso de pagos de bonos y horas extras que realiza la compañía.

Atentamente,

  
SERTECPET  
Silvia Campaña  
Fecha:

Dra. Silvia Campaña

**COORDINADORA DE NOMINA Y COMPENSACIONES**



Sertecpet S.A Ecuador Av. Boy Abalo N07-25 y José Cevallos Casilla: 17-11-06537 Telf.: (052-3)354-930 Quito - Ecuador Cooz: Va Lago Agro Km. 8.7  
Email: info@sartecpet.net, sales@sartecpet.net, internacion@sartecpet.net, apoyo@sartecpet.net

