



**iBossS: FRAMEWORK PARA LA GESTIÓN INDIVIDUAL DEL TRABAJO DE
UN DESARROLLADOR DE SOFTWARE BASADO EN LOS PRINCIPIOS
Y LAS PRÁCTICAS DE AUTOGESTIÓN**

PROYECTO DE GRADO

**ARIEL ARMANDO RUIZ ZUÑIGA
EIDER MAURICIO ARISTIZÁBAL ERAZO**

**Asesor de Investigación
Msc. LILIANA GÓMEZ ARENAS**

**FACULTAD DE INGENIERÍA
DEPARTAMENTO ACADÉMICO DE TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIONES
MAESTRÍA EN GESTIÓN DE INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2012**

**iBossS: FRAMEWORK PARA LA GESTIÓN INDIVIDUAL DEL TRABAJO DE
UN DESARROLLADOR DE SOFTWARE BASADO EN LOS PRINCIPIOS Y LAS
PRÁCTICAS DE AUTOGESTIÓN**

**ARIEL ARMANDO RUIZ ZUÑIGA
EIDER MAURICIO ARISTIZÁBAL ERAZO**

**Trabajo de grado para optar al título de Máster en Gestión de Proyectos y
Tecnología con Énfasis en Ingeniería de Software**

**Asesor de Investigación
Msc. LILIANA GÓMEZ ARENAS**

**FACULTAD DE INGENIERÍA
DEPARTAMENTO ACADÉMICO DE TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIONES
MAESTRÍA EN GESTIÓN DE INFORMÁTICA Y TELECOMUNICACIONES
SANTIAGO DE CALI
2012**

Nota de aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Santiago de Cali, 21 de agosto de 2012

CONTENIDO

RESUMEN	10
1. INTRODUCCIÓN	11
1.1 CONTEXTO DE TRABAJO	11
1.2 PLANTEAMIENTO DEL PROBLEMA	14
1.3 OBJETIVOS	14
1.3.1 Objetivo general	14
1.3.2 Objetivos específicos.....	15
1.4 RESUMEN DE RESULTADOS OBTENIDOS	15
1.5 ORGANIZACIÓN DEL DOCUMENTO	20
2. MARCO TEORICO	21
2.1 AUTOGESTIÓN	21
2.2 CARACTERÍSTICAS DEL TRABAJO INDIVIDUAL DEL PROGRAMADOR EN LA FASE DE DESARROLLO DE SOFTWARE	38
2.3 LA AUTOGESTIÓN Y EL DESARROLLADOR DE SOFTWARE	43
2.4 ESTADO DEL ARTE SOBRE FRAMEWORK DE AUTOGESTION	46
3. DESARROLLO DE LA PROPUESTA	49
3.1 REFERENTES	49
3.2 PROPÓSITO	49
3.3 CARACTERÍSTICAS DEL FRAMEWORK iBossS	50
3.4 ARQUITECTURA CONCEPTUAL DE iBossS	59
3.5 ARTEFACTOS iBossS	60
3.6 HERRAMIENTA iBossS	65
3.6.1 Requerimientos	65
3.6.2 Actores	66

3.6.3 Vista de Casos de Uso	66
3.6.4 Vista de Diagrama de Clases	75
3.6.4.1 Paquete: Automanagement.iBossS.Excecution.....	76
3.6.4.2 Paquete: Automanagement.iBossS.Plan	76
3.6.4.3 Paquete: Automanagement.iBossS.Configuration.....	77
3.6.5 Vista de Desarrollo	78
3.6.6 Vista de Despliegue	80
3.6.7 Desarrollo	81
3.7 ARTEFACTOS Y HERRAMIENTAS Vs. PRÁCTICAS DE AUTOGESTIÓN ..	85
4. PRUEBA DE CONCEPTO DEL FRAMEWORK PROPUESTO	87
4.1 CARACTERÍSTICAS DE LA PRUEBA.....	87
4.2 EQUIPO EVALUADOR	89
4.3 RESPUESTAS Y ANÁLISIS DE RESULTADOS	90
4.4 RESUMEN DE LAS OBSERVACIONES.....	100
5. CONCLUSIONES	102
6. TRABAJO FUTURO.....	104

LISTA DE CUADROS

Cuadro 1. Porcentaje del tipo de causas de problemas en las áreas de proceso	12
Cuadro 2. Lineamientos de iBossS	16
Cuadro 3. Lista de Artefactos y herramientas por capas del Framework	18
Cuadro 4. Síntesis Prácticas de Autogestión	51
Cuadro 5. Criterio de relación: Mejora continua	54
Cuadro 6. Criterio de relación: Planeación del Trabajo	54
Cuadro 7. Criterio de relación: Producción de entregables de software	55
Cuadro 8. Criterio de relación: Manifiesto iBossS	56
Cuadro 9. Grupos de Criterios y Lineamientos del Framework de Autogestión	57
Cuadro 10. Lista de Artefactos por capas del Framework	61
Cuadro 11. Fases propuestas para el SDLC	67
Cuadro 12. Tipos de actividades propuestos	69
Cuadro 13. Prácticas de Peter Drucker	86
Cuadro 14. Equipo Evaluador	89
Cuadro 15. Resultados pregunta número 1	91
Cuadro 16. Resultados Pregunta Número 2	92
Cuadro 17. Resultados pregunta número 3	93
Cuadro 18. Resultados pregunta número 4	94
Cuadro 19. Resultados pregunta número 5	95
Cuadro 20. Resultados pregunta número 6	96
Cuadro 21. Resultados pregunta número 7	97
Cuadro 22. Resultados pregunta número 8	98
Cuadro 23. Resultados pregunta número 9	99
Cuadro 24. Resultados pregunta número 10	100

LISTA DE TABLAS

Tabla 1. Escala y rangos de valoración	88
--	----

LISTA DE FIGURAS

Figura 1. Arquitectura conceptual del framework de autogestión	17
Figura 2. Ciclo de Autogestión en iBoossS	19
Figura 3. Entorno de trabajo con iBossS	19
Figura 4. Estructura del Proceso PSP	43
Figura 5. Proceso de cambio en una arquitectura auto gestionada	47
Figura 6. Estructura básica de un LOOP de realimentación	47
Figura 7. Contexto de trabajo.....	52
Figura 8. Arquitectura conceptual del framework de autogestión	60
Figura 9. Diagrama Casos de uso Configuración de iBossS	67
Figura 10. Diagrama Casos de uso Planeación de Entregables.....	69
Figura 11. Diagrama Casos de uso Producción de Entregables de Software	71
Figura 12. Casos de uso Mejora Continua y Cierre de Tareas	73
Figura 13. Vista de Diagrama de Clases	75
Figura 14. Vista de Diagrama de Componentes	78
Figura 15. Vista de despliegue.....	80
Figura 16. Interfaz gráfica herramienta iBossS	82
Figura 17. Estructura del Paquete	82
Figura 18. Estructura propuesta.....	83
Figura 19. Interfaz gráfica iBossS.plan	85
Figura 20. Resumen resultados pregunta número 1	91
Figura 21. Resumen Resultados Pregunta Número 2	92
Figura 22. Resumen resultados pregunta número 3.....	93
Figura 23. Resumen resultados pregunta número 4.....	94
Figura 24. Resumen resultados pregunta número 5.....	95
Figura 25. Resumen resultados pregunta número 6.....	96
Figura 26. Resumen resultados pregunta número 7.....	97
Figura 27. Resumen resultados pregunta número 8.....	98
Figura 28. Resumen resultados pregunta número 9.....	99
Figura 29. Resumen pregunta número 10	100

LISTA DE ANEXOS

1. Manifiesto iBossS	107
2. Procedimiento de Autogestión iBossS	111
3. Documento Maestro - MD	118
4. Manual de Usuario	120
5. Matrices de Comparación	121

RESUMEN

El interés en la calidad del software ha generado varias propuestas, todas ellas inmersas dentro de modelos, estándares o normas que buscan asegurar la consecución de productos con excelentes características. Sin embargo, como se pudo verificar en la publicación titulada ¿Cuáles son las causas de los problemas de los proyectos de software?, de la IEEE Computer Society (LEHTINEN & MÄNTYLÄ, 2011), se siguen presentando problemas que afectan notablemente los proyectos y muchos, como lo demuestran estudios recientes, están relacionados con las prácticas y la calidad del trabajo ejecutado por el desarrollador de software.

Éstas fueron las razones que motivaron la búsqueda que permitió encontrar en los principios y en las prácticas de autogestión una alternativa, que podría ser utilizada para involucrar al desarrollador de software con el logro de los indicadores de calidad que se proponga una organización o proyecto de software.

El trabajo que realiza el programador y la autogestión se convierten en el foco de atención del marco teórico, encontrando así definiciones que guían el tipo de resultados que se quieren lograr, como el del “profesional en autogestión” y “ser su propio jefe”. En la primera, el individuo es capaz de planificar, organizar, dirigir y controlar su propio trabajo, en la segunda, la autogestión se formula como una alternativa que le permitiría a un programador tener una vida de trabajo verdaderamente satisfactoria y gratificante. En común, todas están respaldadas en teorías y experiencias que sustentan que el éxito de algunas personas se debe a la capacidad de auto gestionarse, lo que les permite planear la búsqueda de la excelencia y ser gestores conscientes de su triunfo.

Estas definiciones se unen para proponer iBossS como un framework que intenta vincular la teoría de autogestión, dentro del contexto de trabajo del desarrollador de software, utilizando para ello dos herramientas y una serie de artefactos que le permitirán al programador llevar el control de sus labores y lo más importante, sobre la visión del éxito: motivarlo a estar continuamente retroalimentando y potencializando sus fortalezas; aplicando para lograrlo, un conjunto de lineamientos derivados de los principios y las prácticas de autogestión.

Finalizando, se incluyen los resultados de la prueba de concepto a la que fue sometido el framework, para tener elementos de juicio que permitieran evidenciar las conclusiones y recomendaciones que se presentan en el proyecto.

1. INTRODUCCIÓN

1.1 CONTEXTO DE TRABAJO

La IEEE Computer Society tiene una publicación titulada: ¿Cuáles son las causas de los problemas de los proyectos de software? (LEHTINEN & MÄNTYLÄ, 2011), donde se muestra con cifras concretas que la inexperiencia y el uso inadecuado de las metodologías de trabajo, son la causa de la mayoría de los problemas. En la investigación mencionada se tomaron proyectos de cuatro compañías de software de tamaño mediano, que les permitieron identificar catorce tipos de causas que afectan los proyectos, teniendo en cuenta las siguientes áreas: Ingeniería de Requerimientos (Rq), Gestión de Personal (Gp), Desarrollo (Ds), Pruebas (Pr), Gestión de Cambios (Gc) y Release (Rl), entre las que se encuentran la de desarrollo y la de pruebas; que son en esencia las causas que más involucran a los programadores y de las cuales se tomarán los datos que enmarcan las variables más relevantes para ésta sustentación. En el Cuadro 1. Porcentaje del tipo de causas de problemas en las áreas de proceso se muestra un resumen donde se puede identificar que las causas señaladas del problema están muy relacionadas con el desempeño laboral y profesional del programador, específicamente la forma de trabajar y la calidad de sus tareas, en ese orden de ideas, se formula la pregunta: ¿Qué puede estar pasando?, al respecto, las conclusiones del artículo resaltan que se le debe dar mucha importancia a la forma de trabajo de la persona.

Lo anterior dirige la atención hacia la importancia que tienen las cualidades del desempeño del creador del código fuente y a la forma en la que los procesos organizacionales buscan mejorar la calidad del trabajo del individuo. En este punto, siguiendo los referentes de estándares o modelos del Software Engineering Institute (SEI), aparece como respuesta el modelo de referencia Personal Software Process (PSP), proporcionando una metodología de trabajo individual que les permite a los ingenieros de software realizar una planificación y seguimiento de su trabajo para lograr productos de alta calidad. En consecuencia, las organizaciones de software no deben ser ajenas a la existencia del PSP, como lo señala Pat Ferguson, quien muestra en su informe la efectividad y los grandes resultados que se logran cuando se trabaja con PSP, relatando los resultados de tres casos de estudio, entre ellos Motorola, que permitieron concluir que el éxito está supeditado a la planificación minuciosa y a una disciplina constante por parte de los involucrados, especialmente en lo concerniente al manejo y administración del tiempo (FERGUSON, HUMPHREY, KHAJENOORI, MACKE, & MATVYA, 1997, pág. 30).

Cuadro 1. Porcentaje del tipo de causas de problemas en las áreas de proceso

Tipo de Causa		Áreas de Proceso						Total	
		Ds	Pr	Otras					
				Rq	Gp	Gc	RI		Ot*
Prácticas y métodos de trabajo		4,30	5,60	0,80	2,30	0,90	1,50	0,30	15,70
Educación y Experiencia		5,10	2,90	3,40	2,30	0,30	2,60	1,40	18,00
Dificultad de la tareas		3,10	2,90	0,60	0,30	0,50	2,00	0,20	9,60
Calidad de las tareas		0,90	2,90	3,20	3,50	0,20	0,60	0,30	11,70
Otros tipos de Causa	Producto Existente	2,50	0,90	0,30	0,30	0,30	2,20	2,00	8,50
	Recursos y Cronogramas	1,90	2,50	0,60	1,10	0,20	0,50	0,80	7,40
	Valores y responsabilidad	1,50	1,40	0,90	1,50	0,50	0,60	0,50	6,90
	Procesos organizacionales	0,60	1,20	0,00	0,20	1,10	1,40	1,10	5,60
	Políticas organizacionales	0,50	0,80	0,50	0,20	0,30	0,80	0,50	3,40
	Cooperación entre personas	0,80	0,00	1,40	0,90	0,20	0,00	0,00	3,20
	Clientes	0,30	0,20	1,50	0,00	0,00	0,50	0,80	3,20
	Herramientas	0,20	1,50	0,20	0,30	0,20	0,20	0,00	2,50
	Prioridad de las tareas	0,80	0,20	0,00	0,20	0,90	0,00	0,30	2,30
Seguimiento del gerente de proy.	0,20	0,20	0,00	0,80	0,30	0,20	0,20	1,70	
Subtotal		22,60	23,10	13,40	13,90	5,70	13,00	8,20	100
Total		22,60	23,10	54,3					100

* Otros
Fuente LEHTINEN 2011. Percentages of the type of causes in software process areas, pág. 4

Entonces, si existen este tipo de modelos, ¿qué podría argumentar la propuesta del presente trabajo de grado? Al respecto, se mencionan algunas afirmaciones preliminares de una investigación donde se señala que el tiempo requerido, la energía y atención al detalle que se necesita para implementar PSP, imponen pesadas cargas al trabajo típico de ingeniería, respaldándose en resultados de laboratorio que toman información de la labor de los programadores después de un tiempo de haber tomado el curso de capacitación en PSP (PHILIP, 1997). También es de analizar elementos de las conclusiones obtenidas de un estudio donde se menciona que para las pequeñas y medianas empresas de software les resulta difícil adaptar estándares o modelos derivados del SEI* y la ISO, justificando la necesidad de buscar nuevos marcos de trabajo respaldados en Software Process Improvement (SPI), para lograr mejoras oportunas y rentables

* SEI (Software Engineering Institute) se utiliza para hacer referencia a los modelos como CMMI y PSP que están a cargo de este Instituto.

en las pequeñas y medianas empresas (PYME) (PINO, GARCÍA, & PIATTINI, 2007). Estas referencias permiten pensar en la posibilidad de buscar otros elementos o herramientas que permitan lograr, desde el trabajo individual de las personas, los estándares de calidad deseados.

En este orden de ideas y fuera del contexto de la Ingeniería del Software, muchos autores como Gary Hamel, recomiendan abandonar el enfoque jerárquico (Gerente, supervisor, jefe, etc.) para administrar las organizaciones del último siglo y proponen un nuevo esquema basado en la autogestión, sobre el cual la calidad y responsabilidad recaen en el individuo (GARY, 2011). También lo señala Peter F. Drucker, cuando brinda varios elementos relacionados con la eficiencia, el manejo del tiempo y la productividad que logra una persona cuando se autogestiona, haciendo alusión a lo que él denomina el Trabajador del Conocimiento (DRUCKER , Management Challenges for the 21st Century, 1999). Y es aquí donde nace la base que motiva este proyecto de grado, porque permite que la autogestión se convierta en el punto de partida para lograr una estructura que oriente el trabajo individual del desarrollador del software para buscar calidad y alta productividad.

Esta visión logró mayor fuerza cuando se analizó que las características que presenta el Trabajador del Conocimiento también fueron del interés del autor de PSP, Watts S. Humphrey, reconocido pensador en temas relacionados con la Ingeniería del Software. Humphrey en sus artículos de la serie titulada “Ser su propio jefe”, presenta una serie de interrogantes y comentarios que abren la reflexión sobre la importancia que tiene la autogestión, para tratar de convencer a los desarrolladores a que lo hagan, recalcando que es la única manera de lograr tener una vida de trabajo verdaderamente satisfactoria y gratificante (SEI, 2009).

Saliendo ya del contexto de las propuestas teóricas para intentar concretar las expectativas de lo que se puede lograr con la autogestión, se inició la búsqueda de fuentes que permitieran evidenciar los efectos de su aplicación, logrando verificar que la salud es el sector que más beneficios reporta. Se encontraron casos de estados, como Wenster Australia, que han implementado varios framework's de autogestión para el manejo de enfermedades crónicas, argumentando que la estrategia de vincular al paciente en sus tratamientos genera muy buenos resultados (Department of Health, State of Western Australia, 2011, pág. 5). Por último y para reforzar el interés en la autogestión, sobresale como alguna de las experiencias empresariales, el caso de la filial Colombiana de Merck, la cual aumentó la productividad de su planta y elevó la autoestima y niveles de liderazgo de sus empleados, al apostarle a la creación de equipos autogestionados dentro de sus líneas de producción (Gestion Humana.com, 2007).

Otro de los factores a resaltar dentro de las motivaciones se sustenta en las grandes oportunidades que presenta el sector del software para Colombia. Visto en cifras, según la Federación Colombiana de la Industria de Software y Tecnologías Informáticas (FEDESOFTE), la industria del software alcanzó en el país un crecimiento del 7.7% frente al 8.9% de Latinoamérica y podría generar un incremento de dos puntos porcentuales en la economía colombiana si llegase a crecer diez puntos. Esta perspectiva está supeditada al reto de que este segmento sea una industria con todos los estándares y todas las certificaciones que se requieren para poder competir (FEDESOFTE, 2012). Hablando de oportunidades, estos datos son alentadores si se tienen en cuenta los desafíos que tendrá el sector software frente a la consolidación del tratado de libre comercio (TLC) con Estados Unidos, al convertirse en uno de los mercados a los que Colombia aspira sacarle el mejor provecho. Aspectos que hacen más atractiva la idea de pensar en fortalecer la calidad desde la forma y la disposición con la que el desarrollador gestiona su trabajo.

1.2 PLANTEAMIENTO DEL PROBLEMA

Según las cifras presentadas en el contexto sobre las debilidades relacionadas con el trabajo individual de un desarrollador y sus posibles consecuencias negativas dentro de los proyectos de software, se evidencia la oportunidad de buscar un mecanismo que le permita al programador adoptar las prácticas y los principios de autogestión que han sido implementadas exitosamente, especialmente en el sector de la salud, para lograr mejoras en la calidad, productividad y en general en el desempeño de su trabajo.

1.3 OBJETIVOS

1.3.1 Objetivo general

Construir un framework que permita incorporar los principios y las prácticas de autogestión en el contexto del trabajo individual del desarrollador de software.

1.3.2 Objetivos específicos

- Compilar los principios y las prácticas más representativas de autogestión para contextualizarlas en el proceso de desarrollo de software.
- Generar un marco de trabajo con artefactos de autogestión para ser aplicados por un desarrollador de software.
- Construir una herramienta basada en el marco de trabajo definido anteriormente, que automatice y facilite al desarrollador aplicar las prácticas de autogestión.
- Realizar una prueba de concepto del framework con un equipo de desarrolladores de software.

1.4 RESUMEN DE RESULTADOS OBTENIDOS

Se destaca que el trabajo desarrollado permitió el logro del objetivo general, definiendo las características de un Framework que permite gestionar el trabajo individual de un desarrollador de software aplicando los principios y las prácticas de autogestión.

Es importante aclarar que las consultas del estado del arte y del marco teórico indican que la autogestión se puede trabajar a nivel organizacional, de equipos e individual, pero el interés de este proyecto se basa en buscar cómo se podría llegar a tener un profesional en autogestión en el campo del desarrollo del software, para que se logren mejores resultados desde la persona, sin descartar que sería ideal llevar el framework a nivel de equipos y de organizaciones.

Este marco de trabajo ha recibido el nombre de iBossS y se hizo necesaria la ejecución del siguiente proceso, el cual se detalla en el Capítulo Tres.

1.4.1. Selección de prácticas y principios de autogestión. Se hace recopilación y lectura de las publicaciones realizadas por un grupo de importantes autores referidos en el marco teórico, tales publicaciones permitieron identificar y seleccionar un grupo de prácticas y principios de autogestión.

1.4.2. Contextualizar el Trabajo del Desarrollador de Software. Se buscaron referencias bibliográficas que aportarán características o elementos importantes

para poder identificar el contexto de trabajo del desarrollador teniendo en cuenta las especificaciones de la ingeniería del software. En este caso, se seleccionaron el Código de ética de la ingeniería del software propuesto por la ACM y la IEEE-CS y las Prácticas Genéricas para el desarrollo individual propuestas por el modelo PSP (Personal Software Process).

1.4.3. Validación de las prácticas de autogestión en el contexto de trabajo del desarrollador. Con las características identificadas dentro del contexto de trabajo del desarrollador de software, se procedió a utilizarlas como instrumento para validar y relacionar cada práctica y principio de autogestión.

1.4.4. Definición de criterios y lineamientos. La validación de las prácticas en el contexto de trabajo del desarrollador, se convirtieron en un referente que permitió identificar los criterios más relevantes para definir las características del framework, buscando principalmente la definición de lineamientos específicos que orienten el trabajo del desarrollador de software enmarcados en las prácticas y los principios de autogestión:

- **Criterio Mejora continua:** Prácticas que comprenden la mejora a nivel personal y profesional.
- **Criterio Planeación de requerimientos:** Prácticas que sugieren la planeación del trabajo a realizar durante el desarrollo de un requerimiento.
- **Criterio Producción de entregables de software:** Prácticas y principios aplicables durante la producción de los productos de trabajo de un desarrollador de software.
- **Criterio Manifiesto iBossS.** Prácticas y principios de autogestión que se deben aplicar para lograr un cambio permanente y trascendental en la forma de ser.

Éstos criterios se utilizaron en un proceso analítico que agrupa, sustituye, cohesiona y enmarca dentro del trabajo del desarrollador de software los lineamientos que se propone sigan los desarrolladores, como se muestra en el Cuadro 2.

Cuadro 2. Lineamientos de iBossS

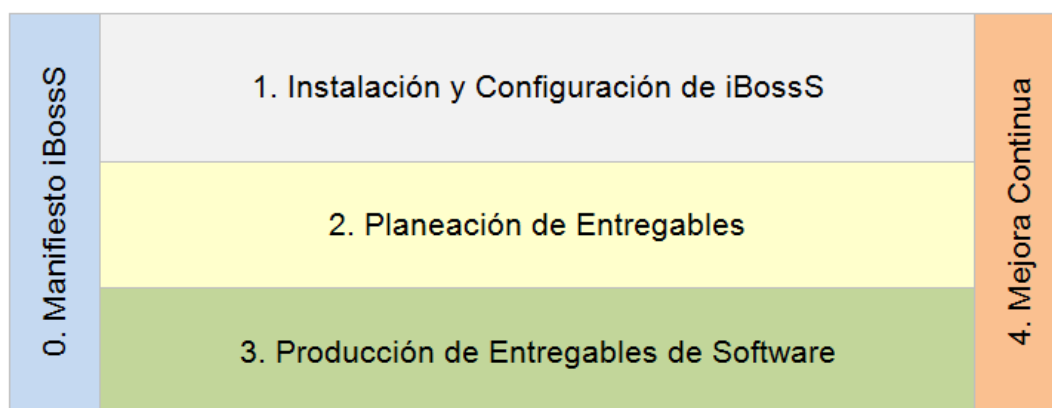
Grupos de Criterios	Lineamiento
Manifiesto iBossS	Manifiesto de autogestión
Mejora Continua	Procedimiento de autogestión

Grupos de Criterios	Lineamiento
	Identificación de Distractores del trabajo
	Lecciones Aprendidas
	Gestión del tiempo
Planeación de Requerimientos	Plan de Trabajo
Producción de Entregables de Software	Área de Trabajo
	Artefactos de los Procesos de calidad
	Artefactos de los Procesos de autogestión
Fuente: Los Autores	

1.4.5. Arquitectura conceptual. Los criterios y lineamientos permitieron definir la arquitectura conceptual del framework que se muestra en la Figura 1.

1.4.6. Artefactos y Herramientas. Teniendo en cuenta cada una de las capas de la arquitectura conceptual del framework, se definieron una serie de herramientas y artefactos que pretenden guiar sistemáticamente la aplicación de las prácticas de autogestión por parte del desarrollador de software. La lista de artefactos y herramientas se muestran en el Cuadro 3.

Figura 1. Arquitectura conceptual del framework de autogestión



Fuente: Los autores

1.4.7. Artefactos vs Prácticas. Para tener mayor claridad de los resultados obtenidos, se consideró necesario justificar desde la perspectiva de la autogestión, las características que se dieron a cada uno de los artefactos y las herramientas. Para ellos se diseñaron unas matrices que permiten validar la implementación que hace iBossS de las prácticas de autogestión. Éstos se detallan en el anexo 5.

Cuadro 3. Lista de Artefactos y herramientas por capas del Framework

Artefactos y herramientas de autogestión	Capa del Framework	
<ul style="list-style-type: none"> - Manifiesto iBossS - iBossS - Procedimiento - Documento Maestro 	0. Manifiesto iBossS - 4. Mejora Continua -	<ul style="list-style-type: none"> - 1. Instalación y Configuración de la herramienta
<ul style="list-style-type: none"> - iBossS - iBossS.Plan 		<ul style="list-style-type: none"> - 2. Planeación de Entregables
<ul style="list-style-type: none"> - iBossS - Procedimiento - Documento Maestro 		<ul style="list-style-type: none"> - 3. Producción de entregables de software
Fuente: Los Autores		

1.4.8. Ciclo de Autogestión en iBossS. En la Figura 2 se ilustra en forma de ciclo, como el framework abarca los dos ámbitos del desarrollador: el personal y el laboral, buscando que el éxito y la calidad sean el foco de atención que se logra a través de un proceso de retroalimentación permanente.

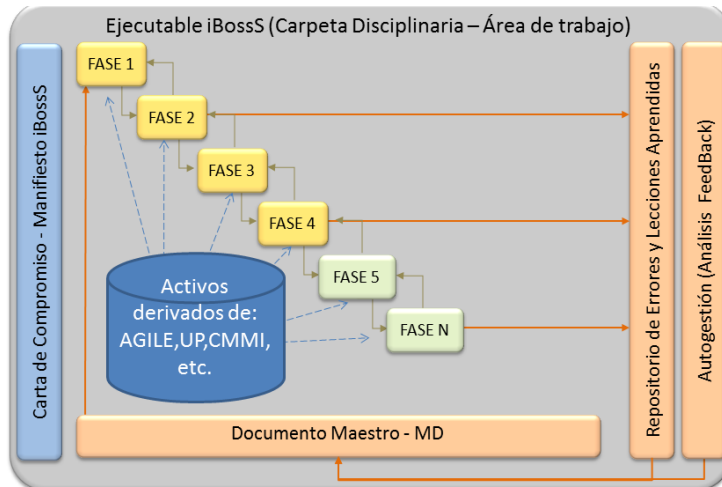
Figura 2. Ciclo de Autogestión en iBossS



Fuente: Los Autores

1.4.9. Entorno de Trabajo con iBossS. La Figura 3 representa como iBossS se integra en el contexto de desarrollo. Se busca que en las fases de planeación o construcción, el Desarrollador en su proceso de retroalimentación lleve sus experiencias a las lecciones aprendidas, para que a futuro, cuando tenga que construir un nuevo requerimiento, en el documento maestro tenga la información que le permita evitar errores y hacer que el trabajo sea más productivo.

Figura 3. Entorno de trabajo con iBossS



Fuente: Los Autores

1.5 ORGANIZACIÓN DEL DOCUMENTO

El presente documento se encuentra organizado de la siguiente manera:

1. **Capítulo 1 - Introducción:** Se hace una descripción de lo que se quiere lograr con este trabajo. Apoyándose en el contexto de trabajo, y el planteamiento del problema.
2. **Capítulo 2 - Marco Teórico:** Se identificarán las prácticas de autogestión propuestas por reconocidos autores en el tema. También se identificarán las características del trabajo individual del programador en la fase de desarrollo de software. Finalmente se describe la relación que existe entre la autogestión y la ingeniería del software.
3. **Capítulo 3 - Desarrollo de la propuesta:** En este capítulo se, ejecuta un proceso analítico mediante el cual se logran validar y aplicar las prácticas de autogestión en el contexto del desarrollo de software para planear, describir y construir las características, artefactos y herramientas del framework de autogestión iBossS.
4. **Capítulo 4 - Prueba de Concepto:** Se seleccionaron 8 profesionales en el desarrollo de software a los que se les hace una presentación del framework iBossS y la intención de contribuir con la mejora en la calidad del software mediante su utilización. Finalmente, se cierra el capítulo con una tabulación de los resultados obtenidos.
5. **Capítulo 5 - Conclusiones:** Se relacionan una serie de conclusiones sobre la experiencia lograda durante el desarrollo de este trabajo y sobre la importancia y algunas falencias de iBossS.
6. **Capítulo 6 - Trabajo Futuro:** Se describe lo que se podría realizar con iBossS y que sería interesante ampliar por quien a criterio considere que este trabajo es importante.

2. MARCO TEORICO

Los antecedentes y fundamentos teóricos del presente trabajo están relacionados con dos focos temáticos: La Autogestión y Las Características del trabajo individual del programador en la fase de desarrollo de software. Con ellos se conforma el marco de referencia que sustenta la posibilidad de poder aplicar los principios y prácticas de autogestión en el contexto del desarrollador de software.

Para los referentes de autogestión se evalúan dos tipos de prácticas, las que tienen que ver con el enfoque de aplicación personal y las otras que lo han hecho como estrategia organizacional.

Después se mencionan los aportes que definen claramente las características de los procedimientos a ser ejecutados por el programador y algunos principios que están directamente relacionados con las buenas prácticas. La intención es poder especificar las actividades que representan la labor que ejecuta el desarrollador paso a paso.

Se tiene también un punto en el que se logra describir cómo se llegó a concebir la idea de unir la teoría de autogestión con el desarrollador de software, partiendo de dos definiciones: “Trabajador del Conocimiento” y “Ser su Propio Jefe”.

Finalizando, en el estado del arte, se hace un análisis de cuatro framework para autogestión que se han desarrollado en diferentes contextos, con el fin de introducir algunos elementos válidos que ayuden a aclarar las características de los resultados de la propuesta.

2.1 AUTOGESTIÓN

2.1.1 Peter Ferdinand Drucker. Es considerado el padre del management, falleció en 2005 a la edad de 95 años, fue abogado y tratadista austriaco, nacionalizado como ciudadano americano y ampliamente reconocido por sus aportes en temas relacionados con la administración de empresas.

Dentro de sus obras se destaca Management Challenges for the 21st Century (Desafíos de la gerencia en el siglo XXI), donde describe de forma sencilla; a nivel de preguntas, cuáles deben ser las prácticas de autogestión (Managing Oneself) que debe aplicar un Trabajador del Conocimiento, para poder sobrevivir a los

cambios turbulentos que imponen los nuevos tiempos y así lograr el éxito. Porque según Drucker “...*todos, inclusive los menos dotados, debemos aprender a autogestionarnos, a manejarnos a nosotros mismos, a hacer autogestión*” (DRUCKER , 1999).

Las preguntas se abordan desde los cinco cuestionamientos que se listan a continuación, los cuales permiten precisar las prácticas, principios y valores que necesita una persona aplicar para gestionarse a sí misma.

1. Preguntarse: ¿Quién soy yo? ¿Cuáles son mis fortalezas? ¿Cómo me desempeño?
2. Preguntarse: ¿A dónde pertenezco?
3. Preguntarse: ¿Cuál es mi contribución?
4. Responsabilidad sobre las relaciones
5. Planificar la segunda mitad de su vida

El tratamiento que se hace para cada una de estas preguntas, a través de ejemplos y de recomendaciones, genera las siguientes prácticas:

- **Haga un análisis feedback periódicamente:** Este método se utiliza para identificar claramente las fortalezas, se parte de la pregunta ¿Cuáles son mis fortalezas?, e indica que cada vez que se tenga que tomar una decisión o ejecutar una acción clave, se deben escribir a nivel de expectativas lo que se espera va a ocurrir. Pasados nueve o doce meses, se deben comparar los resultados con las expectativas y sobre un análisis consiente se pasara a identificar las fortalezas y las debilidades. Las fortalezas van orientadas a poder responder donde se es competente y las debilidades donde no se logran buenos resultados en el desempeño. Este análisis se desarrolla con el ánimo de identificar varias implicaciones que orientaran la acción o el proceso de feedback:
 - Concentrarse en las fortalezas.
 - Trabajar en mejorar las fortalezas.
 - Superar la ignorancia paralizante (Arrogancia).
 - Corregir los malos hábitos (Que generan pérdida de eficiencia).
 - Practicar los buenos modales.
 - Reconocer que no se puede o no se es capaz de hacer (Humildad).

- **Ser consciente de las características bajo las cuales usted presenta mejor desempeño:** Consiste en tener claridad sobre las cosas en las que se logra un buen desempeño y en cuáles no. El análisis de feedback ayuda notablemente en este reconocimiento, pero el desempeño es un asunto de personalidad, lo que quiere decir que la persona desde su forma de ser tiene que poder determinar en que es buena o en qué es mala para saber dónde logrará mejores resultados y dónde no debe participar.
- **Definir el estilo de comunicación (lector u oyente):** Se refiere a la necesidad de poder identificar la forma en la que se interpreta y se participa de los procesos de comunicación, si se es un lector o un oyente. El oyente es aquel que tiene la capacidad de participar espontáneamente, demostrando dominio del tema al describir y explicar una situación o idea en pocas frases, en cambio, el lector tendrá que conocer previamente los temas a tratar para lograr el control, escribiendo sobre sus percepciones antes de discutir o participar. Según el autor, es difícil que una persona pueda ser las dos cosas a la misma vez y que es un error procurar ser algo contrario a lo que ya se es.
- **Entender cómo se aprende:** Identificar si se es un lector u oyente están muy relacionados con la forma en la que se aprende, hay muchas personas que aprenden sea escuchando, leyendo, escribiendo, hablando o de muchas otras formas, pero lo más importante como clave del desempeño, es poner en práctica la respuesta para la pregunta ¿Cómo aprendo?.
- **Definir la capacidad de trabajar en equipo:** Es necesario definir si se está en la capacidad de trabajar en equipo o individualmente. Si existe la capacidad de poder participar en equipos también se debe identificar si se hace en el rol de coach o mentor.
- **Prever si se está en la capacidad de tomar decisiones o de asesorar:** En cuanto al desempeño y a la toma de decisiones, se debe saber si se está en la capacidad de seguir el papel de tomador de decisiones o de asesor. Un gran número de personas se desempeñan mejor como asesores, pero no pueden tolerar la carga y la presión de tomar una decisión. Y un buen número de otras personas, en contraste, necesitan asesores que las obliguen a pensar; luego pueden tomar sus propias decisiones y ejecutarlas con rapidez, con confianza en sí mismas y con coraje.

- **Identificar las características de su desempeño en diferentes entornos de trabajo:** Es necesario saber cuál será el comportamiento frente al estrés o la presión y en diferentes entornos: empresas grandes o pequeñas, para reconocer las características del desempeño en diferentes situaciones.
- **Identificar cuáles son sus valores personales:** La compatibilidad que exista entre los valores de la persona y los de la organización tiene una relación directa con la eficacia y el desempeño, por ende la necesidad de poder identificar los valores de la persona para saber si pueden coexistir con los de la organización. Esto es muy importante, no solamente para definir el éxito en las labores, sino para determinar si lo que se hace está de acuerdo con el proyecto de vida individual.
- **Decidir cuál es el lugar a ocupar dentro de la organización a partir de sus habilidades:** Ya se han establecido varios elementos relacionados con el buen desempeño, lo que quiere decir que ya se está en la capacidad de poder describir concretamente cual será el comportamiento frente a las oportunidades, las ofertas o proyectos que se presenten, porque la persona podrá estar segura de decir “esto es lo que soy” y “esto es lo que soy capaz de hacer”.
- **Contribuir con el éxito de la organización:** La intención es ir más allá de lo que le digan o le asignen, es establecer la diferencia buscando contribuir con los mejores resultados o con las decisiones acertadas que la situación requiera. Para ello, se debe establecer un plan donde se defina qué hacer, dónde y cómo comenzar y qué metas y plazos fijar, sobre tres elementos distintivos:
 - a. **Requerimientos de desempeño:** identificar las fortalezas, formas de actuar y los valores adecuados para atender la situación.
 - b. **Contribuciones:** definir cuáles serán los aportes o las contribuciones frente a lo que se debe hacer.
 - c. **Resultados esperados:** consientes de la forma en la que se abordará la situación se deben identificar los resultados a lograr. Estos deben ser visibles y, en la medida de lo posible, cuantificables.
- **Responsabilizarse de las relaciones interpersonales:** El primer secreto de la eficacia está en conocer y poder entender a las personas con las que se

trabaja. Lo primero es aceptar que las otras personas también tienen fortalezas, formas de comportarse y valores, por tal motivo hay que observar, conocer y analizar cómo se comportan las otras personas, para evitar conflictos y fracasos. Lo segundo es asumir la responsabilidad de la comunicación, para que todas las personas sepan que es lo que los demás saben hacer y cómo lo hacen.

- **Gestionar la segunda mitad de su vida:** Saber que se hará en la segunda mitad de la vida es de las últimas tareas, porque gestionarse a sí mismo es también pensar en iniciar una segunda carrera después de haber alcanzado la primera cúspide, buscando dentro de las expectativas de vida laboral una oportunidad para sí mismo y para la sociedad. El objetivo de esta tarea es comenzar a gestionar la segunda mitad de la vida antes de llegar a ella, es decir ahora mismo, y frente al ¿Cómo hacerlo?, se pueden analizar las siguientes respuestas:
 - a. **Empezar realmente una nueva carrera:** a menudo implica cambiar de organización, de línea de trabajo o estudiar una nueva profesión.
 - b. **Desarrollar una carrera paralela:** Consiste en desarrollar una actividad en forma paralela, generalmente aportando a organizaciones sin ánimo de lucro.
 - c. **Ser un emprendedor social:** Lo que se puede ver representado en la creación de nuevas empresas, fundaciones u organizaciones sin ánimo de lucro.

2.1.2 John C. Marshall y Bob McHardy. Son los autores del libro Principios de Autogestión: La clave para el éxito personal y profesional. Marshall es un reconocido psicólogo organizacional y líder de pensamiento innovador, es el presidente y fundador de The Self-Management Group.

Ellos también refieren que el éxito de algunas personas se debe a su capacidad de auto gestionarse; pues fueron capaces de planear y ser gestores conscientes de su triunfo.

Motivan la aplicación de la autogestión justificando que todos en algún momento lo hemos hecho y que gracias a ello se ha logrado la felicidad y la oportunidad de ser más exitosos.

Mencionan que la autogestión se logra aplicando los siguientes procesos: (MARSHALL & MCHARDY, 1999)

- Determinar sus propias metas, objetivos y expectativas.
- Resolver lo que tienen que hacer: su comportamiento, la estrategia o el plan de acción.
- Comprometerse.
- Evaluar el desempeño de la actividad.
- Evaluar los resultados.
- Buscar recursos para la auto-mejora y el desarrollo.
- Evaluar el proceso en general, durante un período de tiempo, para decidir si continuar, ajustar o terminar.

Confrontan la idea que cuando se nace el deseo de supervivencia viene desde una motivación totalmente interna, pero que al pasar los años se tiende a depender de influencias externas que complican y condicionan las decisiones que hay que tomar día a día. Qué se pierde el control de la propia vida, porque lo que piensan otras personas tiene la capacidad de modificar su curso. *“Tenemos la capacidad de elegir, pero aprendemos a que otros lo hagan por nosotros”* (MARSHALL & MCHARDY, 1999, pág. 18).

Ante éste problema justifican que la solución está en la autogestión, porque a través de ella se logrará recuperar una madurez que permita volver a la auto-determinación para poder tomar las propias decisiones y dirigir las capacidades hacia el logro de los objetivos personales. Sostienen que para lograrlo se deben aplicar nueve principios, los cuales desde la lectura pueden resultar simples y de fácil asentimiento, pero que eso no es suficiente, que lo más importante es colocarlos en práctica día a día, de manera sistemática, para lograr una estrategia que permita fortalecer y potenciar los beneficios de la autogestión. Estos son los nueve principios:

- **Concéntrese en optimizar su rendimiento, esa es la clave del éxito.** Existen diferencias y una estrecha relación entre dos enfoques muy comunes: el rendimiento y los resultados.

El enfoque por resultados se deriva de motivaciones externas, por lo tanto puede ser influenciado y desconoce o no tiene en cuenta las circunstancias y la forma en la que se consiguen; desmeritando en algunas ocasiones las características del desempeño y el grado de satisfacción o motivación que se tuvo para lograrlos. Es más, esta predisposición a medir por los resultados

puede llevar a que estos sean falsificados o a que las acciones para lograrlos no sean las más adecuadas.

En cambio, el enfoque por rendimientos obedece a motivaciones internas y por lo tanto puede ser controlado. En ese sentido, el éxito se define en relación al potencial de la persona, mientras que en el otro caso se mide en términos de lo que se consigue.

Sobre su correlación se precisa que uno es consecuencia del otro, los resultados son fundamentales y es la forma de maximizar el rendimiento, puesto que es el rendimiento el que conduce a los resultados y son éstos la evidencia disponible de la calidad y la cantidad de una actuación. Sobre ésta caracterización se enfatiza la importancia de trabajar bajo el enfoque del rendimiento, justificando que una persona si puede llegar a ser responsable de su rendimiento más no puede llegar a ser plenamente responsable de los resultados por lo inciertos o incontrolables que pueden llegar a ser. Así se concreta al rendimiento como el primer principio que se debe seguir para aplicar autogestión, porque es el mecanismo que permite acoplar la búsqueda de resultados con la responsabilidad que existe sobre las acciones que se siguen para obtenerlos, teniendo en cuenta el compromiso, la motivación y la disciplina que hay inmersa en ello.

El principio del rendimiento contiene implícitos tres factores que garantizan el éxito: talento, actitud y oportunidad, los cuales se propone sean vistos como una ecuación que los multiplica ($\text{rendimiento} = \text{talento} \times \text{actitud} \times \text{oportunidad}$), de tal forma que si uno de ellos es cero, no habrá como resultado rendimiento. Entonces para ser exitoso no basta con tener el talento y la actitud correcta, también es necesario buscar las oportunidades, descartando así que estas sean cuestión de suerte y al contrario de lo que se dice, probar que éstas obedecen a factores de selección y de control del desempeño.

- **Las expectativas que usted defina determinan su rendimiento.** Se deben establecer los propios límites; el éxito es medido a partir de las propias expectativas. Aunque puedan existir estereotipos a seguir como ejemplos, lo más importante es alcanzar las propias expectativas y el autogestor lo sabe.

“Si piensas que puedes o que no puedes: tienes la razón”, es una frase pronunciada por Henry Ford y ampliamente utilizada para referir la duda que existe en el logro de las expectativas. Es una afirmación que permite lanzar el interrogante *¿Dónde nacen este tipo de dudas?*, y esto suele ser porque pensamos que las expectativas son causas externas, que vienen de deseos y reglas que otras personas ponen, como los profesores, el jefe o en ocasiones los amigos o los familiares, pero esta es otra manera equivocada de ver las

cosas, no hay que dar lugar a que la gente defina las expectativas aprovechando ciertas circunstancias u oportunidades, que al contrario, lo que se debe hacer es demostrar pensada y determinadamente la capacidad de fijar las propias expectativas, porque si este deseo no viene desde el interior, no habrán creencias que ayuden a forjar las metas y a superar las adversidades que puedan impedir el logro de lo que se quiere. Las expectativas se convierten entonces, en la fuerza persistente que viene desde adentro, para lograr lo que se desea, como el deseo de vivir.

De esta forma las expectativas y el rendimiento se convierten en la manera directa de lograr la productividad deseada, porque se sabe que se alcanzó una meta cuando el rendimiento alcanza el objetivo, permitiendo así que periódicamente el rendimiento fije nuevas expectativas.

- **El refuerzo en sus actividades crea hábitos.** Todo comportamiento que se refuerza positivamente tiende a repetirse y a generar un condicionamiento que promueve la creación de hábitos, estos a la vez se convierten en códigos de conducta que refuerzan e internalizan fácilmente los comportamientos aceptables. Esta práctica de autogestión promueve al auto-refuerzo como el modelo ideal para dirigir las acciones que pueden estar influenciadas por agentes externos, pero aclarando que se debe tener cuidado cuando el refuerzo externo genere inconsistencias o contradicciones con las motivaciones internas, porque esto puede llevar a situaciones de tensión o de estrés.
- **Automotívese constantemente durante la consecución de sus metas.** El poder que entrega la automotivación nunca para, es la forma de tomar el control, por eso la motivación es un aspecto crítico dentro de la autogestión. Se debe actuar impulsado y atraído por la automotivación, el impulso es el deseo de alcanzar la meta y la atracción la ejerce la idea de pensar que tan cerca se está de lograrla.

En el tema de la motivación están los elementos externos que supuestamente motivan, como el actuar por el premio, la recompensa y los incentivos, los cuales en última instancia se pueden reducir a términos puramente materiales que se acaban y no dejan huella. Estos condicionan y tiñen el comportamiento, ocultando la verdadera razón del por qué se hacen las cosas y no son verdadera motivación, son autoridades externas que determinan la forma de actuar. Por eso la ruta a seguir debe venir desde el interior, donde se coloque de primera mano al individuo antes que las razones externas y los deseos de

otras personas, convirtiendo la automotivación en la fuerza interna que nunca para y que permitirá el alcance del éxito, obviamente con recompensas, pero lo más importante: con la sensación del logro constante de la excelencia individual.

- **Maximizar el rendimiento estableciendo prioridades en sus tareas.** Realice una sola tarea al tiempo, si existen varias tareas, priorícelas y resuélvalas en el orden definido. La energía y el interés de hacer las cosas se deben enfocar sobre las prioridades, “no se queme”, tenga en cuenta que debe velar por su salud y que tiene una familia, éstos son aspectos que deben ser incluidos dentro de sus compromisos.
- **Gestionar el esfuerzo haciendo seguimiento a sus planes.** Consiste en saber llevar el único camino que conducirá al logro de los resultados y a la consolidación del éxito: El esfuerzo. Desde el comienzo se ha hecho hincapié en la obligación que tienen los autogestores de ser responsables de su rendimiento y de la rendición de cuentas de sus resultados. Para ello el plan incluye unas metas y unos objetivos específicos que permiten enfocar el futuro y como se resaltó en las primeras líneas, el esfuerzo es la ruta; a mayor esfuerzo mayor éxito. La gestión del esfuerzo permite especificar la escala del paso a paso que se debe seguir para el logro de los resultados, evitando de cierta forma, la perversa razón de justificar lograr los objetivos sin importar la forma del cómo se llega a ellos. Para lograrlo se debe primero analizar qué aspectos son controlables y cuáles no, asumiendo el control de los primeros, teniendo en cuenta las capacidades y el esfuerzo que se necesita, obteniendo con claridad lo que se tiene que hacer, recordando siempre la recomendación de tener en cuenta las fortalezas. En este caso los no controlables darán una visión de los límites que se tienen para saber qué decisión tomar.
- **Perfeccionar la autoconfianza.** Perfeccione la confianza en sí mismo para desarrollar todo su potencial. La confianza está muy relacionada con la motivación, depende de la fuerza interior y no de influencias externas, descartando así la posibilidad de que otras fuerzas ejerzan presión sobre el comportamiento. Cuando la autoestima es alta, se espera un mejor desempeño, porque la persona tiene más claros sus niveles de rendimiento y el alcance de sus expectativas. La gestión también debe incluir la confianza que se tiene en los demás.

- **Comprométase con el logro de los objetivos planeados.** El compromiso debe ser un ingrediente constante en el proceso, no se debe llegar a los resultados o a las conclusiones para poder evidenciar el grado de compromiso que se tuvo frente al proceso. Hay que resaltar que existe una gran diferencia entre “yo debo hacer” y “yo debería hacer”, porque si se analiza, sólo la primera indica un compromiso verdadero y representa por ende la esencia de lo que es la autogestión, la otra denota una acción frágil; fácil de disuadir y de incumplir. Nuevamente recalcar que para el cumplimiento de éste principio existe la necesidad de mantener presente el compromiso con los demás.
- **Tome el control de su vida gestionando su tiempo.** La gestión del tiempo se ofrece habitualmente como un medio para que la gente tome el control de sus vidas, para mejorar su rendimiento o para aumentar su eficiencia, especificando claramente para el tiempo disponible, con sus horarios tipo reloj, las actividades a desarrollar. Suena convencional, pero la autogestión lo que busca es hacer el mejor uso del tiempo disponible colocando toda la energía para hacer lo que se tiene que hacer y no perder el tiempo pensando en lo que “podría” hacerse.

2.1.3 Gary Hamel. Es un norteamericano experto en gestión, nombrado como el pensador de negocios más influyente del mundo y el experto mundial en estrategia de negocios. Reconoce la importancia de las teorías de Drucker y en particular sus aportes al concepto de la economía del conocimiento (THE WALL STREET JOURNAL, 2009). Este autor tiene gran incidencia sobre la concepción de la autogestión vista desde las organizaciones, se enfoca en una serie de temáticas que resaltan la intención de poder llegar a tener organizaciones altamente eficaces y profundamente humanas, dirigidas por profesionales en autogestión, que manejan y coordinan todo, sin la necesidad de tener jefes (GARY, The big idea: First Lets Fire all the Managers, 2011, pág. 4).

En su artículo “La gran idea: En primer lugar vamos a despedir a todos los jefes”, publicado por Harvard Business Review en diciembre de 2011, se muestra cómo la procesadora de alimentos Morning Star logra, a partir de la autogestión, ser una empresa líder por más de dos décadas, sin jefes, títulos o promociones.

Cada persona que trabaja en Morning Star es su propio gerente: planifica, organiza, dirige y controla su trabajo, es un Profesional en Autogestión y lo hace de la siguiente manera:

- Crea una misión personal, la socializa y la negocia con los compañeros.
- Es gestor de su misión.
- Negocia las responsabilidades con sus compañeros para poder construir una red de responsabilidades recíprocas. Registra los compromisos adquiridos en una carta llamada “Carta de Entendimiento con los Compañeros” redactada en primera persona.
- Es responsable de adquirir las herramientas que necesite para su trabajo.
- Identifica los procedimientos o problemas que atentan con su misión o verifica si su comportamiento afecta la misión de alguno de sus compañeros, realiza las correcciones necesarias y construye un adecuado engranaje.
- Se centra en los beneficios no en las actividades.
- Es motivado e impulsado por su misión y sus compromisos.
- Rinde cuentas de lo previamente negociado con los compañeros.
- Ejerce la libertad con sabiduría, siendo capaz de calcular el impacto de sus decisiones.
- Siempre se pregunta:
 - ¿Cuál es el valor que ayudará a crear para sus compañeros?
 - ¿Cuáles son los problemas que ayudará a resolver a sus compañeros?

Esta es la descripción de algunas de las prácticas que desarrollan los profesionales en autogestión:

- **Crear una misión personal.** Cada persona debe elaborar una declaración de su propia misión, señalando como contribuirá con el objetivo de la organización e identificando como cumplirá con las expectativas de calidad y con las métricas de rendimiento. Igualmente será responsable de cumplirla y de gestionar las capacitaciones, recursos y colaboraciones que necesite.
- **Ser innovador sin encajarse en un manual de funciones.** No es necesario encajar sobre un marco de funciones, el comportamiento de un individuo, porque esa persona debe tener la oportunidad de asumir mayores responsabilidades y de sugerir mejoras a medida que se desarrollan nuevas

habilidades y se gana más experiencia. La gente suele impulsar el cambio y la innovación fuera de su estrecho marco de trabajo.

- **Fomentar la competencia en pro de los beneficios de la contribución.** Consiste en competir para lograr reconocimiento porque se puede contribuir más, por tener nuevas habilidades o formas de servir a los compañeros, no por beneficios personales como las promociones o el ascenso en las jerarquías.
- **Buscar el éxito pensando primero en la libertad.** Cada persona es libre de hacer lo que verdaderamente desea, sin necesidad de ser empujado, teniendo en cuenta el cumplimiento de las responsabilidades adquiridas y el disfrute de la efectividad operacional.
- **Establezca metas claras y actúe con transparencia en la información.** Toda decisión sabia se toma sobre información clara, lo que facilita el control de los resultados.

El trabajo basado en la autogestión, le ha permitido a Morning Star lograr una serie de ventajas, de las que se destacan:

- **Más iniciativa.** Las funciones que se dan son generales, lo que les da a las personas la oportunidad y la libertad de actuar, de ser proactivos y de favorecer su reputación.
- **Más experiencia.** La autogestión ayuda a que una persona desarrolle sus habilidades, siendo responsable de la calidad de su trabajo.
- **Mayor flexibilidad.** Con el grado de autonomía que brinda la autogestión la persona tiene la capacidad de adaptarse a cualquier realidad.
- **Mayor compañerismo.** Los compromisos son adquiridos y socializados en común acuerdo con los compañeros. Entonces con los compañeros se acuerda, se trabaja y se controla.
- **Mejor juicio.** Cuando la persona planea y propone sus metas, conforme a la misión promulgada, no tiene que ser empujada, ella sabe tomar la decisión más acertada, sabe que es lo que le conviene más, recuerde que todos son gerentes.

- **Más lealtad.** La práctica de la autogestión le permite a una persona empoderarse de los objetivos y de la razón de ser de la organización.

Morning Start también admite que hay desventajas al aplicar este tipo de modelo:

- **Problemas de adaptación.** Comienza diciendo que la autogestión no es del agrado de todos, porque para muchos, especialmente para los que han pasado su vida dentro de organizaciones jerárquicas, les es muy difícil ajustarse a este tipo de desafíos.
- **Desafío en la rendición de cuentas.** Si el compromiso y las exigencias para con los compañeros no son verdaderos, la autogestión puede ser una conspiración de la mediocridad.
- **Largos periodos de inducción.** Se necesita tiempo para encajarse en el sistema, aproximadamente un año.

2.1.4 Jeremy Hunter y J. Scott Scherer. Ratifican la importancia de los pensamientos de Drucker y su visión de la autogestión y el trabajador del conocimiento. Hunter es un reconocido PhD dedicado a la formación en autogestión y a la transformación de la mente del ejecutivo. Escriben el capítulo número once titulado “Knowledge Worker Productivity and The Practice of Self-Management”, el cual hace parte del libro “The Drucker Difference: What the World's greatest management thinker means to today's business leaders.” Sus reflexiones permiten cubrir otros elementos que favorece notablemente este marco teórico, porque precisan una serie de argumentos que clarifican el alcance de los mecanismos a seguir para implementar adecuadamente la autogestión y poder lograr todas sus bondades (YAMAWAKI, MACIARIELLO, & PEARCE, 2009, págs. 175-194).

Toman, en la introducción del capítulo, comentarios de algunos detractores que ponen en tela de juicio el alcance de la teoría de autogestión, pero logran a partir de ellos, concientizar que los fracasos se han dado porque las estrategias aplicadas siguen quedándose en elementos del contexto externo y omiten profundizar en la esencia de lo que es la autogestión: el aporte del ser, lo que se convierte en un excelente punto de referencia para este proyecto. Sobre esta conclusión concretan una serie de aclaraciones y explicaciones que son muy importantes para los principios:

- **Minimice los asuntos internos (inestabilidad emocional, erosión moral, pérdida de la atención) que afectan su productividad.** La autogestión ayuda a que las personas vean cómo los procesos internos tienen un impacto directo en su comportamiento externo. La pérdida de la productividad a menudo se remonta a sucesos externos que interrumpen el pensamiento claro y las interacciones socio afectivas que generan conductas desadaptativas, como en estos casos:
 - **Inestabilidad emocional.** Cuando en las estructuras jerárquicas se hace uso del poder para infundir miedo.
 - **Erosión moral que afecta el trabajo en equipo y el rendimiento.** Sucede cuando los jefes hacen uso del sarcasmo y pierden cualquier sentido de la solidaridad.
 - **Perdida de la atención.** Como por ejemplo, reuniones poco productivas o la presencia de jefes a los que se les ha perdido el respeto.

Esto se había planteado antes, pero la diferencia es que aquí los autores unen la gestión de la productividad con puntos de vista contemporáneos del desarrollo humano, como la neurociencia y otros avances de la medicina y la psicología, para concretar la verdadera transformación del individuo que quiere llevar la autogestión a la práctica.

- **La Atención Plena y la Autogestión.** Esta unión trae los aportes de la neurociencia, afirmando que la autogestión se inicia en el sistema nervioso, directamente en el cerebro y se requiere tratar unos procesos internos para lograrlo:
 - **La atención es el fundamento de la autogestión.** La atención es la raíz misma del juicio, del carácter y de la voluntad; una persona para ser dueña de sí misma lo primero que debe controlar es su atención. Comenzando por la imperiosa necesidad de entrenar la percepción para poder ver los supuestos ocultos y las nuevas posibilidades, porque la forma en la que vemos las cosas influye en como las entendemos y cómo podemos responder a ellas.
 - **Atención concentrada: el enfoque es el poder.** Diseñe sistemas y entornos de trabajo que le permitan conectarse y mantener la atención sobre la labor realizada, si se hace esto sistemáticamente se fortalece la corteza prefrontal del cerebro, la cual está directamente relacionada con la atención y es llamada el CEO* interior. Evitando al máximo elementos que

* Chief Executive Officer (Director Ejecutivo)

obstaculicen el enfoque de la atención, como la multitarea, porque esto lo que puede hacer es debilitarlo, “la multitarea daña la productividad, las relaciones y el cerebro.” Las investigaciones demuestran que las múltiples tareas reducen el rendimiento, los errores aumentan, disminuye la atención disponible y aumenta la probabilidad de que las emociones de desorganización, como el miedo y la ansiedad, abrumen la función cerebral. Las personas que evitan la multitarea manifiestan que logran conexiones más significativas con sus colegas y seres queridos, convirtiéndose así en mejores oyentes: perciben mejor.

- **La Meditación de la Concentración: Fortalecimiento del CEO interior.** Para mejorar la atención existen métodos probados basados en la práctica de la meditación de la concentración. Esta técnica de meditación fortalece la capacidad del cerebro porque le permite enfocarse, convirtiéndose así en el antídoto de la multitarea, también se demostró por la escuela de medicina de la Universidad de Harvard, que fortifica su corteza prefrontal.

- **La atención y el abandono sistemático de hábitos desadaptativos.** Para lograr que la atención se convierta en la base del desarrollo de la concentración y que pueda ser utilizada como una herramienta para otras tareas; como dirigir las cosas a conciencia. Se necesita ir abandonando sistemáticamente hábitos desadaptativos que se quedan en el subconsciente y que según la neurobiología pueden ir dominando el lado consciente, por la predisposición cerebral a moverse por las acciones repetitivas y las creencias, perjudicando notablemente la productividad. Pero entonces si existe esta predisposición, ¿Qué se tiene que hacer para romperla?
 - **Neuroplasticidad: Recableando la Red.** Esta teoría plantea nuevamente que la atención es la estrategia que le permite al cerebro transformar su comportamiento, para ello es necesario que el autogestor dirija la atención hacia un nuevo comportamiento para romper el patrón antiguo e ir desmantelando poco a poco el comportamiento no deseado.
 - **El aprendizaje continuo.** Es un principio de la autogestión que le permite a la persona ser curiosa y ser abierta al mundo. Esta forma de pensar permite asumir una visión más amplia, empática y más esperanzadora, que racionalmente se centra en las posibilidades y oportunidades. Se plantea que tiene mucha relación con la teoría de la Mentalidad de Crecimiento propuesta por el psicólogo Carol Dweck, la cual sostiene que el talento puede ser cultivado porque todos están en la capacidad de aprender y perfeccionarse.

- **Ser consciente de las emociones reactivas.** Las emociones reactivas suceden de forma automática e irracional, motivadas por la disposición de la propia defensa. La auto-gratificación, la ira, el miedo, la ansiedad, la lujuria y el deseo son algunos ejemplos. Para evitarlas, lo mejor es conocerse y prever que no sucedan, lo que nos lleva nuevamente a enfocarse en la atención porque es la forma de observarse así mismo dentro de un episodio emocional para analizar cuál sería la respuesta.

2.1.5 Self-Management Institute. Es un instituto dedicado a la investigación y capacitación en temas de autogestión. Sostienen que la autogestión es la piedra fundamental y el método más eficaz y eficiente de organizar a la gente. Es la forma en la que las familias se organizan, en la que las economías prosperan. Tienen varias publicaciones donde se tratan los principios que ellos han construido y sobre los cuales basan su razón de ser. Ofrecen un buen número de artículos escritos por miembros y colaboradores, entre los que se destacan: Doug Kirkpatrick quien aporta muchos elementos en el tema de los principios y la teoría de autogestión, LoriKane Handout con información de equipos de trabajo autogestionados y Chris J. Rufer, su fundador y el promotor de la filosofía del trabajo basado en autogestión (Self-Management Institute, 2012).

2.1.5.1 Doug Kirkpatrick. Para las organizaciones que aplican el modelo de la autogestión, esta se define como la filosofía que le permite estar compuesta por individuos con libertad y autonomía, que realizan las funciones tradicionales de gestión (planificación, organización, coordinación, personal, dirección, control), sin jerarquías o autoridades arbitrarias y unilaterales. Se rigen por dos principios fundamentales:

- Los individuos no deben usar la fuerza contra otras personas o sus bienes.
- Las personas deben respetar y cumplir los compromisos que han hecho con los demás.

Otras características que resalta el autor sobre la autogestión son las siguientes:

- Proporciona a las personas todo el poder que necesitan desde el primer día de trabajo para llevar a cabo su misión, y nadie tiene la autoridad para despedirlos unilateralmente.

- No todo el mundo puede prosperar en una organización autogestionada, por ejemplo: no será fácil para los que tratan de ejercer el poder y la autoridad (quieren ser "jefes") y los que tienen que ser constantemente dirigidos para cumplir con sus responsabilidades. Es particularmente propicia para aquellos que toman la iniciativa, tienen habilidades para comunicarse y trabajar bien con los demás.
- Para ser un líder en una organización autogestionada se debe asumir el reto de desarrollar relaciones de alta calidad, tener un excelente comportamiento, labrar siempre las conductas ejemplares, ser comunicativo, imaginativo y con mucha iniciativa. También influye la experiencia y las competencias.
- Para que una persona sea eficaz se requiere que cumplan con sus compromisos: a largo plazo y a corto plazo. Sólo manteniendo los compromisos se puede generar confianza en los demás y prosperar en un ecosistema de alto rendimiento.

2.1.5.2 LoriKane Handout. Analiza la situación de los equipos autogestionados y extrae una serie de características que permiten añadir características para las prácticas de autogestión, sobre referencias que respaldan lo dicho en este capítulo pero que son esenciales porque respaldan la importancia de trabajar en equipo.

- **Mantenga el foco en lo más importante.** Esto lo incluye a usted y a otras personas, como sus clientes, amigos y compañeros.
- **Crear en sus compañeros de trabajo y mantener buenas relaciones personales con ellos.** Ésta es la característica que permite que los equipos de trabajo se mantengan a través del tiempo, hace que las personas se apoyen mutuamente, buscando el éxito, el aprendizaje y el crecimiento continuo, proyectando seres más conscientes, creativos, inteligentes y unidos. Con este tipo de trabajo se haría más fácil lograr las siguientes características en un persona que se autogestiona:
 - Ser más abierto y visible, comunicando los logros y compromisos ante el equipo.
 - Facilita la inclusión y la adaptación continua.
 - Facilita el reconocimiento de las limitaciones.

2.2 CARACTERÍSTICAS DEL TRABAJO INDIVIDUAL DEL PROGRAMADOR EN LA FASE DE DESARROLLO DE SOFTWARE

Una buena guía de referencia para poder caracterizar dentro del framework las particularidades de la fase de desarrollo de software, desde la perspectiva del trabajo individual del programador, son algunas de las propuestas de Watts S. Humphrey, dentro de las que se incluyen dos de sus obras más importantes: The Personal Software ProcessBody of Knowledge y The Personal Software Process (PSP).

Estos libros definen una serie de principios y de procesos en los que se basa PSP, permitiendo conocer las razones que deben motivar u orientar la misión que se asume cuando se decide crear código fuente y la forma de cómo controlar el trabajo.

Humphrey se convierte así, en uno de los autores con una relevancia considerable dentro de los resultados que se logran dentro del framework, por eso se incluye este punto, en el que se conocerán cuáles serán sus aportes.

2.2.1 Watts S. Humphrey. Es reconocido internacionalmente por sus aportes en temas relacionados con la Ingeniería del Software y por su carrera llena de éxitos como miembro del SEI (Software Engeenering Institute), donde dirigió y concretó el desarrollo de proyectos muy importantes: The Personal Software Process (PSP) y Team Software Process (TSP), también participó del proyecto que hoy ha evolucionado a lo que se llama Capability Maturity Model Integration (CMMI).

También se ha retomado el tema que él ha denominado “Ser su propio jefe”, descrito en una de sus columnas de la revista del SEI, donde toma como referencia a Peter Drucker para generar una serie de preguntas que permiten evidenciar su visión de la aplicación de la autogestión en el contexto de trabajo del desarrollador. Define desde sus 50 años de experiencias, cuáles serían las ventajas de llevar el Trabajador del Conocimiento de Drucker a los procesos de calidad en el desarrollo de software (SEI, 2009).

2.2.2 The Personal Software Process Body of Knowledge. Encapsula los conocimientos y los principios básicos de la ingeniería de software que se convierten en los fundamentos de PSP, también ofrece una guía que orienta a los

profesionales del software en la forma que se pueden llevar a cabo métodos disciplinados para mejorar su proceso de trabajo (SEI, 2010).

Provee una serie de definiciones que describen elementos básicos que se van a tratar o a considerar dentro del framework de autogestión, como los siguientes:

Proceso: Secuencia de pasos que un profesional experto debe seguir para realizar una tarea específica.

Proceso definido: Es una secuencia documentada de los pasos necesarios para hacer un trabajo específico. Los procesos se definen por lo general para los trabajos que se realizan en varias ocasiones y hay que hacerlo de la misma manera cada vez que se lleve a cabo.

Proceso básico: Tiene tres fases.

1. *Planificación:* Elaborar un plan para hacer el trabajo.
2. *Desarrollo:* Realizar el trabajo.
 - a. Definir los requerimientos
 - b. Diseñar el programa
 - c. Revisar el diseño y corregir todos los defectos
 - d. Escribir el código del programa
 - e. Revisar el código y corregir todos los defectos
 - f. Construir o elaborar y corregir todos los defectos
 - g. Probar el programa y corregir todos los defectos
3. *Postmortem:* Se comparan los resultados reales con el plan, se actualizan los datos de registro del proceso, se produce un informe y se documentan todas las ideas para fortalecer la mejora de procesos.

También se encontraron una serie de argumentos que soportarían algunos lineamientos que cubre el framework de autogestión, como por ejemplo:

Fidelidad en el proceso: A veces llamada “Disciplina de Proceso” o “Cumplimiento de Procesos”, y se define como el grado en el que los individuos siguen su proceso personal. El objetivo de recalcar la fidelidad

en el proceso es para buscar mejorar el rendimiento del trabajo y la realización de productos de mejor calidad. A menos que el proceso sea seguido fielmente, la mejora de procesos no es posible.

La calidad del producto se rige por la calidad del proceso utilizado para su construcción. No es suficiente definir un proceso de alta calidad, se necesita que los individuos también sigan fielmente ese proceso para el desarrollo del producto. Crear y utilizar consistentemente un proceso de alta calidad se traducirá en la obtención de productos de alta calidad. La calidad del producto, a su vez, tiene un efecto directo sobre la capacidad del individuo para cumplir con el calendario y los objetivos presupuestados para el producto.

Las medidas claves del rendimiento del proceso personal son:

- La capacidad para cumplir con los compromisos programados para la entrega prometida.
- La calidad de los contenidos entregados.

Post-mortem: Un análisis post-mortem de los trabajos realizados cuando finaliza una fase o proyecto, proporciona una valiosa información, incluyendo datos de los proyectos actualizados en tiempo, tamaño, defectos y horario.

Mejora de procesos: Las razones para la aplicación de mejoras en el proceso son la mejora de la previsibilidad y la calidad de la prestación, reducir el tiempo del ciclo y mantener o mejorar la productividad.

Principios de Planificación: Las personas que realizan el trabajo son las más idóneas para planificarlo.

- Las personas siempre deben desarrollar un plan de trabajo antes de comprometerse con iniciar un proyecto. Cuando los individuos están involucrados en el desarrollo del plan, lo más probable es que se han comprometido con ese plan.

- Los planes deben basarse en procesos definidos y en datos históricos, se hacen con un nivel de detalle que sea apropiado para el trabajo que se va a realizar.
- Cuando es difícil hacer un plan preciso, lo mejor es comenzar con un plan preliminar y volver a planear a menudo.
- Cuando el plan no se ajusta al trabajo, se debe revisar el plan.

Planificación y Seguimiento de la Calidad del Software: Describe la necesidad de producir productos que satisfagan las necesidades de los usuarios.

2.2.3 The Personal Software Process. Consiste en un conjunto de métodos, formas y secuencias de pasos que muestran como un Ingeniero de Software puede planificar, medir y gestionar su trabajo. Proporciona un framework de trabajo disciplinado que puede ser utilizado en varios puntos relacionados con la producción de software (HUMPHREY, 2000).

Se citan en primer lugar los principios que rigen el marco de trabajo, los cuales se consideró prudente incluir dentro de los lineamientos que se han definido para el framework de autogestión:

- Cada ingeniero es diferente, para ser más eficaz debe planear su trabajo y debe basar sus planes en sus experiencias personales.
- Para mejorar constantemente su rendimiento, el ingeniero debe utilizar procesos personales bien definidos y debidamente medidos.
- Para producir productos de calidad, los ingenieros deben sentirse personalmente responsables de la calidad de sus productos. Los productos que son superiores no son producidos por error, los ingenieros deben esforzarse por hacer un trabajo de calidad.
- Cuesta menos encontrar y corregir un defecto en un proceso temprano que más tarde.

- Es más eficaz prevenir los defectos que encontrarlos y corregirlos.
- La forma correcta es siempre la manera más rápida y barata de hacer un trabajo.

También se adoptó la descripción del flujo de procesos que tiene la estructura del proceso de PSP para definir las actividades que se quieren incluir dentro del framework de autogestión, como se muestra en la Figura 1.

El flujo del proceso involucra las siguientes tareas:

Requerimientos. Se comienza con la declaración de la sentencia de los requerimientos y será la base para el plan. Por supuesto que la precisión de la estimación y el plan estarán fuertemente influenciados por la capacidad del ingeniero en saber sobre el trabajo a realizar.

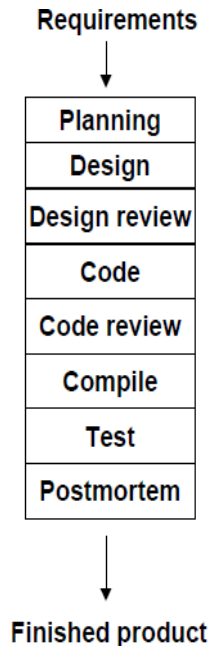
Diseño Conceptual. Para hacer una estimación y un plan, los ingenieros deben definir en primer lugar cómo el producto será diseñado y construido. Sin embargo, desde la fase de planificación es demasiado pronto para realizar un diseño completo de productos, por consiguiente, los ingenieros elaboraran lo que se llama un diseño conceptual. Esta es una primera aproximación, más tarde, durante la fase de revisión del diseño, los ingenieros analizar alternativas de diseño.

Desarrollar el producto. En la etapa denominada desarrollar el producto, los ingenieros hacen el trabajo de programación real: codificación, compilación y prueba. Si bien este trabajo no se considera normalmente como parte del proceso de planificación, los ingenieros deben utilizar los datos de este proceso para hacer planes de futuros.

Post-mortem. Después de completar un trabajo, los ingenieros realizarán un análisis post-mortem de la obra. En la autopsia, que actualizará el resumen del plan del proyecto con datos reales, se calculara cualquier dato de calidad o de rendimiento y se revisara qué tanto se cumplió el plan. Asimismo, revisará los defectos encontrados en la compilación y pruebas, y actualizará sus listas de control de

revisión personal para ayudarse a encontrar y corregir defectos similares en el futuro.

Figura 4. Estructura del Proceso PSP



Fuente: (HUMPHREY, 2000, pág. 7)

2.3 LA AUTOGESTIÓN Y EL DESARROLLADOR DE SOFTWARE

La autogestión aparece en el contexto del desarrollo de software desde los planteamientos de las metodologías ágiles, donde se destaca que la autogestión es una cualidad que todos los equipos de trabajo deben tener. Por ejemplo, en el manifiesto ágil, en algunos de sus principios se refiere que las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados y que es necesario que en un determinado momento los equipos reflexionen sobre cómo pueden ser más efectivos para perfeccionar y ajustar el comportamiento (Beck & Otros, 2001). En este orden de ideas, se encontró que Scrum crea las condiciones necesarias para que los equipos reconozcan sus capacidades, señalando que se logra obtener un mayor grado de autogestión cuando los equipos son capaces de definir la cantidad de trabajo a realizar, asumiendo responsabilidad sobre el desempeño y encontrando maneras para mejorarlo (Keith, 2010, pág. 159).

Estos aspectos indican la importancia de la autogestión desde la perspectiva de los equipos, pretendiendo lograr mejores resultados, lo que demanda que los integrantes también conozcan qué es y cuáles son los alcances de la teoría de autogestión para que el éxito de los equipos no se vea como un hecho aislado, sino como una consecuencia de la participación de profesionales en autogestión.

Esta aplicación de la autogestión es totalmente válida y de mucha importancia, pero como se resalta dentro de los trabajos futuros, es una etapa a la que se debe llegar después de analizar y conocer otras propuestas que llevan la autogestión al nivel de los equipos de trabajo, como lo recomienda el Self-Management Institute, en especial LoriKane Handout (Self-Management Institute, 2012).

En este trabajo se quiere resaltar la intención de vincular la teoría de autogestión en los procesos de mejora de la ingeniería de software, que hace Watts Humphrey. Él propone que la construcción de software es el escenario ideal para laborar sin la presencia de jefes autocráticos, invitando a los programadores a que sean sus propios jefes, siguiendo la noción del Trabajador del Conocimiento que documenta Peter Drucker (SEI, 2009, pág. 187).

Humphrey se refería al término “Ser su Propio Jefe” (Being Your Own Boss), el cual mencionó en la revista del SEI, donde publicó en la columna titulada “Watts New?”, cinco artículos donde involucra, dentro de su visión de lo que tiene que ser el trabajo de desarrollo de software, los conceptos del Trabajador del Conocimiento.

Entonces, el Trabajador del Conocimiento se convierte en ese vínculo que permitió unir la visión y el enfoque de éstos dos autores, para crear una asociación entre las prácticas y los principios que se han tomado de la autogestión, con la arquitectura conceptual que se requiere para el framework. Otro aspecto importante de esta información, es que se pudo comprobar que la teoría de autogestión tiene una gran trascendencia, cuando se comprueba que el Trabajador del Conocimiento hace parte de un término que cobra gran relevancia en lo que hoy se conoce como la “Sociedad del Conocimiento”. Por consiguiente, se puede afirmar que todos los intentos por aplicar autogestión están contribuyendo, de cierta manera, con el fortalecimiento de la nueva sociedad.

2.3.1 Trabajador del Conocimiento. Drucker desde el año 1969, comienza a hablar de uno de los temas de mayor incidencia en la actualidad: “La Sociedad del Conocimiento”, sosteniendo que lo más importante para lograr atender las exigencias del mundo actual, es aprender a aprender, concretando que la sociedad del conocimiento se construye a partir de los Trabajadores del

Conocimiento, los cuales se dedican a crear e innovar, buscando siempre alcanzar la mayor productividad a partir del conocimiento y de su capacidad de auto gestionarse.

Esa fue la forma de llegar a la teoría de autogestión, conociendo qué es el Trabajador del Conocimiento y Drucker lo describe en su libro “Los Desafíos para la Gestión del Siglo 21 (Management challenges for the 21st century)”, donde se destacan las siguientes características que resaltan la cualidad de auto gestionarse: (DRUCKER , Management Challenges for the 21st Century, 1999, pág. 142)

- La responsabilidad de la productividad está en el propio trabajador del conocimiento. Para lograrlo tienen que ser autónomo y auto gestionarse.
- El trabajador del conocimiento tendrá que auto gestionarse para saber dónde será más productivo, para aprender a desarrollarse y para mantenerse mentalmente joven durante una vida laboral de cincuenta años.

2.3.2 Ser su propio jefe. Humphrey comienza refiriendo estudios del por qué los proyectos de software no son exitosos, concluyendo que en la mayoría de los casos, el fracaso se da por el grado de desmotivación y por lo poco gratificante que resulta para los desarrolladores su participación en el proyecto.

Esta conclusión soporta las razones expuestas en el planteamiento del problema, porque Humphrey desde sus cincuenta años de experiencia, admite que el éxito del proyecto no se puede ver únicamente desde el cumplimiento de los cronogramas o de los costos, que también influye saber que tan exitoso fue el proyecto desde la perspectiva profesional, el punto de vista técnico y el ambiente de trabajo que comparten los desarrolladores.

Sobre ésta problemática, Humphrey enmarca una solución, colocando de antemano que lo primero que hay que hacer para evitar entornos de trabajo desagradables, es impedir la participación de jefes autocráticos, proponiendo la búsqueda de nuevos estilos de gerencia, en la que todo el equipo de trabajo participe, asegurando que para lograrlo se necesita el concurso del Trabajador del Conocimiento y su capacidad de auto gestionarse, aplicando cuatro recomendaciones:

- Saber qué es lo que se está haciendo.
- Entender cómo se hace.
- Medir lo que se ha logrado.
- Saber cuánto tiempo llevo hacerlo.

2.4 ESTADO DEL ARTE SOBRE FRAMEWORK DE AUTOGESTION

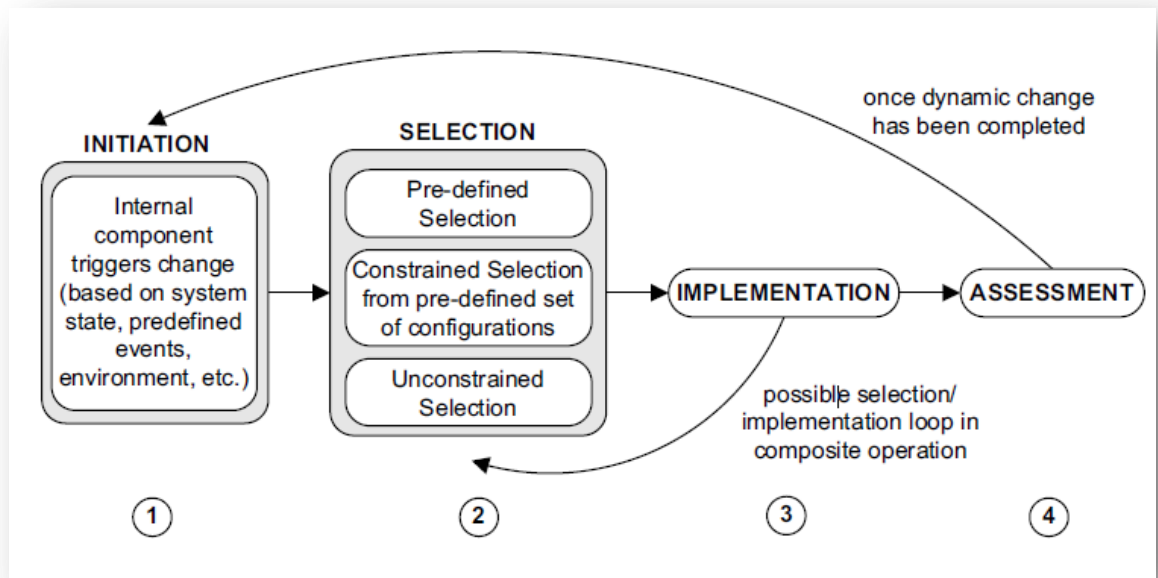
Para complementar el marco teórico, se muestran los resultados de la búsqueda exhaustiva que se hizo de frameworks de autogestión, encontrando muchos resultados aplicados en diferentes contextos y en el caso específico del desarrollo de software, los tratados por el desarrollo ágil, pero no se encontró un referente que precise directamente la aplicación de autogestión a nivel individual y que involucre directamente el trabajo del programador.

Se considera necesario hacerlo, porque es bueno mostrar los ejemplos donde se han logrado buenos resultados y porque de cierta manera, sirvieron para tener un acercamiento a la forma y diseño del framework que se quiere lograr. Se comienza mostrando ejemplos de circuitos y se termina con los casos de aplicación en las organizaciones.

2.4.1. Autogestión en Circuitos Digitales y Sistemas de Información. Se trata de procesos de automatización basados en software y electrónica digital. En general se encontró que estos casos conciben la autogestión como el mecanismo que les permite a los sistemas retroalimentarse, buscando eliminar labores de administración externa, principalmente del hombre, para ser sistemas autosuficientes que se auto regulan y se adaptan a las necesidades del entorno; siguiendo el ejemplo de algunos sistemas biológicos. La mayoría definen cuatro componentes para estos sistemas: analizador, diagnosticador, planeador y ejecutor. (HAYDARLOU, OVEY, & OVEREINDER, 2006) .

Otros afirman que las arquitecturas de sistemas auto gestionados implementan cambios al interior de su composición (Association for Computing Machinery, 2004). Por lo cual deben ser dinámicas y abiertas al cambio.

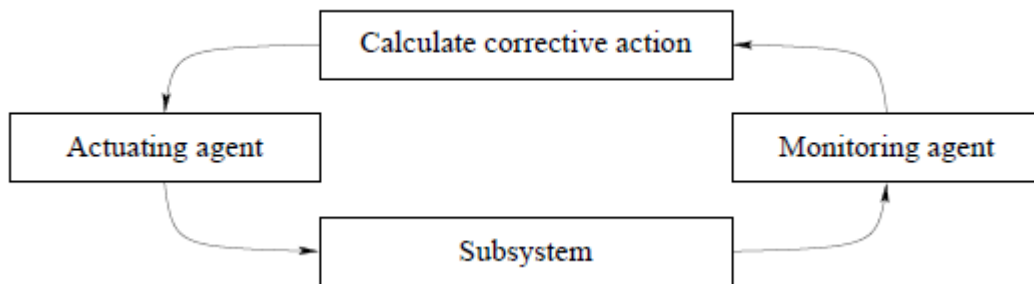
Figura 5. Proceso de cambio en una arquitectura auto gestionada



Fuente: A survey of self-management in Dynamic Software Architecture

Un mecanismo básico para implementar autogestión son los LOOPS (Roy, 2006) consiste en tres elementos que interactúan juntos en un subsistema. Un elemento que monitorea el estado, otro que calcula una acción correctiva y otro que aplica esta acción correctiva en el sistema:

Figura 6. Estructura básica de un LOOP de realimentación



Fuente: Self-Management and the Future of Software Design

2.4.2. Framework for self-management for Chronic Disease. La medicina y la psicología se caracterizan por ser los casos con el mayor número de ejemplos,

constatando que son grandes los esfuerzos que se hacen en el sector de la salud para intentar minimizar los riesgos que corren las personas con enfermedades crónicas. Se han implementado propuestas de cuidado y prevención basadas en autogestión, por ejemplo en Australia, como en otras parte del mundo, las enfermedades crónicas constituyen la mayor carga de problemas que deben ser atendidas por los organismos del sector de la salud, convirtiéndose en una de las grandes preocupaciones porque no se tiene la capacidad para responder a la cantidad de requerimientos que demandan este grupo de pacientes.

Estos países han encontrado en la autogestión una gran alternativa que les ha permitido implementar una serie de estrategias que cambian los factores conductuales de las personas afectadas, minimizando así los riesgos y controlando los efectos de la enfermedad. En estos casos la autogestión les permite a los pacientes ser parte activa del tratamiento, ayudándoles a reconocer sus fortalezas y debilidades para disminuir el impacto de la enfermedad y lograr adquirir nuevos comportamientos que mejoran notablemente su estilo de vida. En estos ejemplos se logra evidenciar como la aplicación de los principios de autogestión salvan vidas y les permite a los organismos de salud pública ser más eficaces (Department of Health, State of Western Australia, 2011).

2.4.3. Autogestión en la educación. En estas propuestas también intentan que un estudiante realice un seguimiento de su propia conducta, ya sea con el propósito de mejorar o perfeccionar un comportamiento o habilidad. Se define como una estrategia de intervención educativa que busca que los estudiantes controlen y cambien su propio comportamiento, que sean ellos los agentes de su propio cambio en lugar de depender de los adultos para lograrlo. Se destaca en este caso la experiencia de las universidades Lehigh y de California, las cuales han incluido la autogestión dentro de su proyecto "REACH" el cual trabaja con estudiantes que tienen profundas necesidades sociales, emocionales y de comportamiento (LEHIGH UNIVERSITY).

2.4.4. Autogestión en las organizaciones. Existen muchas experiencias pero se aprovecha nuevamente este espacio para relatar la experiencia de la empresa procesadora de alimentos Morning Star la cual logra, a partir de la autogestión, que cada trabajador sea su propio gerente, planificando, organizando, dirigiendo y controlando la misión que se ha propuesto para colaborar con el éxito de la organización y el bienestar personal (GARY, The big idea: First Lets Fire all the Managers, 2011).

3. DESARROLLO DE LA PROPUESTA

La propuesta describe, en primera instancia los referentes y el propósito, para darle al lector un resumen y una idea de los elementos que se han tenido en cuenta para llegar a la definición de las características del framework y para que conozca cuáles fueron los motivos que llevaron a su concepción y a entender porque se ha llamado iBossS, que significa “I’m my Boss of Software”.

3.1 REFERENTES

El Framework iBossS se fundamenta en dos líneas base que permiten consolidar el propósito y el contenido del modelo propuesto. La primera está relacionada con el proceso de desarrollo de software y los principios señalados en The Personal Software Process Body of Knowledge, que se consideran básicos, porque son cercanos a las labores específicas del desarrollador y son importantes en el logro de trabajos de alta calidad (SEI, 2010, págs. 15-16). La segunda línea permite definir el valor de la autogestión tratándola desde dos perspectivas; la individual y la grupal (organizaciones y equipos de trabajo auto gestionados). En la individual se incluye a Peter Drucker quien logra, desde una serie de cuestionamientos, hacer inferir características y principios que le permitirían a un trabajador auto gestionarse para lograr el éxito (DRUCKER , Management Challenges for the 21st Century, 1999, págs. 163-195) y el libro Principles of Self-Management: The key to personal and professional success (MARSHALL & MCHARDY, 1999). En la grupal aparece The Self-Management Institute con una amplia gama de artículos ofrecidos en su portal web por miembros que trabajan el tema, entre los que se destacan Gary Hamel y su fundador Chris Rufer (Self-Management Institute, 2012). Estas dos líneas se unen por sugerencia del mismo Watts S. Humphrey, cuando propone que una forma de contribuir con la mejora de los procesos de desarrollo de software es traer la autogestión y la teoría de Peter Drucker al contexto del programador, buscando que sea su propio jefe, “Being Your Own Boss”, de donde proviene el nombre del Framework.

3.2 PROPÓSITO

Para enmarcar el propósito del framework se comenzará definiendo el significado de la palabra autogestión en el contexto de iBossS. La autogestión se define como la iniciativa que le permitirá al desarrollador de software asumir con libertad y autonomía el control de sus actos, visionando siempre la excelencia de sus

acciones sobre las responsabilidades éticas que su profesión le exige, actuando siempre en pro del éxito personal y de la organización para la cual trabaja; siguiendo la estrategia de gestionarse a sí mismos para llegar a ser lo que Humphrey denominó “Su propio jefe” (SEI, 2009, pág. 175).

El framework contendrá entonces, un consenso de los principios y las prácticas de autogestión que buscarán concientizar, educar y principalmente, motivar a los desarrolladores de software sobre la importancia que representa para lograr el éxito, su actitud frente a la vida y al trabajo, porque no se concibe el logro de la excelencia sin un recorrido lleno de metas y triunfos que impacten positivamente el mundo en el que se desenvuelven, teniendo en cuenta especialmente el entorno en el que se desenvuelven y las obligaciones éticas atribuibles a las profesiones relacionadas con la Ingeniería de Software.

3.3 CARACTERÍSTICAS DEL FRAMEWORK iBossS

Para obtener las características de iBossS se hizo necesaria la ejecución de cuatro pasos, los cuales permitieron definir los lineamientos que enmarcan las prácticas y los principios de autogestión en la disciplina del desarrollo de software:

3.3.1 Paso 1: Compilar Prácticas y principios de Autogestión. A partir de la lectura de los documentos y las publicaciones de los autores referidos en el marco teórico, como Peter Drucker, se extraen las recomendaciones más relevantes de la autogestión, las cuales se compilaron en tres grandes grupos:

- **Principios y valores:** Los cuales dictan las actitudes y comportamientos que debe asumir la persona que desea auto gestionarse para lograr sus metas y garantizar la armonía y la fraternidad en el trabajo.
- **Prácticas a nivel general:** Son un conjunto de prácticas y principios que se sugieren para aplicar las prácticas de autogestión.
- **Prácticas a nivel de equipo:** Son un conjunto de prácticas que recomiendan la existencia de un líder externo para apoyar un equipo auto gestionado.

Posteriormente, se tabularon, analizaron y sintetizaron, dando como resultado veintiuna prácticas de autogestión que son implementadas por el framework propuesto. Éstas se muestran en el Cuadro 4 y se refieren en el **Anexo 2 Procedimiento de Autogestión iBossS** en la sección 10. Actividades de Autogestión.

Cuadro 4. Síntesis Prácticas de Autogestión

Prácticas de Autogestión Sintetizadas	Peter F. Drucker	John C. Marshall Bob McHarday	Gary Hamel	Jeremy Hunter y J. Scott Scherer	Doug Kirkpatrick	LoriKane Handout
Gestione adecuadamente su tiempo.	X	X				
Concéntrese en optimizar su rendimiento.	X	X		X		
Defina su misión y objetivos y consiga los recursos necesarios para trabajar en función de ellos.	X		X			
Planifique su propio trabajo.	X	X	X			
Comprométase con el logro de los objetivos planeados.		X				
Traducir los planes de trabajo en acciones concretas.	X	X	X	X	X	X
Asumir la responsabilidad de comunicar.	X			X		
Asuma responsabilidad por las decisiones tomadas.	X	X	X		X	
Concentrarse en las oportunidades.	X			X		
Coordine sus actividades con los demás miembros de su equipo	X				X	X
Cree y mantenga buenas relaciones personales con sus					X	X
Llame a las personas por su nombre.	X				X	X
Desarrolle altas cualidades de relaciones, excelente desempeño, comunicación y visión. De ésta manera será visto como un	X	X	X	X	X	X
Haga un análisis Feedback periódicamente teniendo en cuenta sus expectativas y decisiones tomadas a nivel técnico,	X					
Incorpore prácticas de autogestión en los procesos de negocio de su organización.	X		X			X
Sea asertivo a la hora de resolver conflictos laborales de tal manera que a futuro éstos no afecten su trabajo.	X				X	
Mantenga el enfoque en lo que más le importa.	X	X	X	X		
Realice una sola tarea al tiempo, Si existen varias tareas, priorícelas y resuélvalas en el orden definido.	X			X		
Mantenga la atención en su trabajo, identificando y reduciendo las distracciones.	X	X		X		
Haga uso de herramientas informáticas para facilitar el uso de las prácticas de autogestión.			X		X	X
Todos los principios influyen en el modo de operar de la persona por lo que deben ser interiorizados.	X	X	X	X	X	X
Fuente: Los autores						

En esta síntesis se puede evidenciar que la mayoría de las prácticas se tomaron de la propuesta de Peter Drucker y que están respaldadas por los otros autores referidos en el marco teórico.

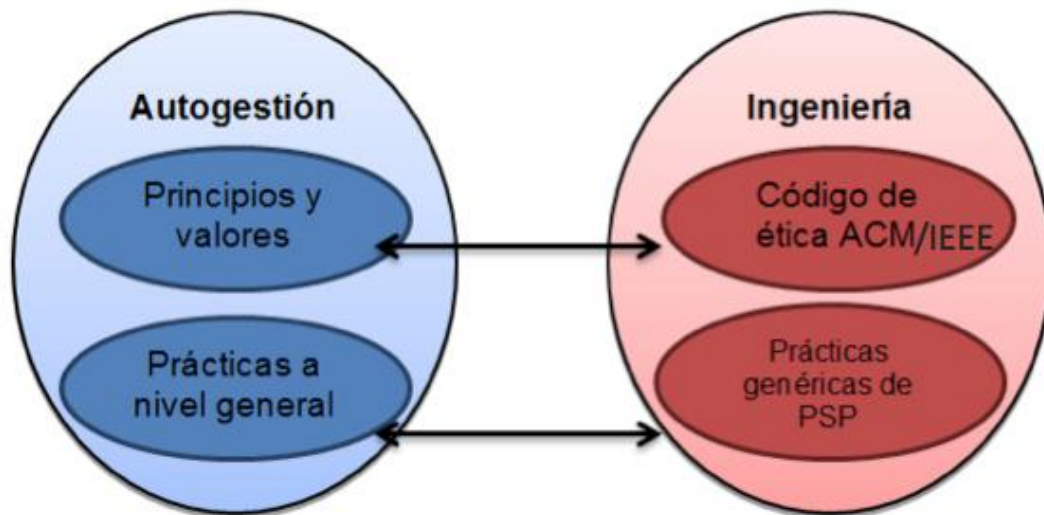
3.3.2 Paso 2: Contextualizar el Trabajo del Desarrollador de Software.

Teniendo en cuenta que uno de los objetivos de este proyecto de grado es compilar los principios y las prácticas de autogestión para contextualizarlas en el proceso de desarrollo de software, se seleccionaron de los grupos descritos anteriormente, los dos primeros: Principios y valores y Prácticas a nivel general.

El tercer grupo se omitió porque hace referencia a un tema no considerado dentro del alcance del proyecto, muy relacionado con el trabajo en equipo propuesto por TSP y considerado como trabajo futuro.

Para encontrarle aplicabilidad a éstos dos grupos en el entorno de trabajo del desarrollador, se buscaron las teorías que se asemejan dentro del contexto de la ingeniería del software. En este caso, como se muestra en la Figura 5, se encontró una relación muy directa entre el Código de ética de la ingeniería del software propuesto por el ACM y la IEEE-CS y las Prácticas genéricas para el trabajo individual de software tomado del modelo PSP.

Figura 7. Contexto de trabajo



Fuente: Los Autores

3.3.3 Paso 3: Validar el contexto de autogestión. Como las prácticas y los principios de autogestión previamente identificados deben ser validados ante la disciplina del Desarrollo de Software, se procedió a relacionarlas frente a un

referente auténtico y reconocido en el Desarrollo de Software, como lo es el Personal Software Process (PSP). Para ello se definieron una serie de criterios que permitieran integrarlos ampliamente, en un comienzo se trabajó con tres, pero posteriormente se observó que no todas las prácticas seleccionadas estaban inmersas en el modelo propuesto por PSP, entendiéndose que esto sucede porque Humphrey está orientado más en la ingeniería del proceso y no en las características personales del desarrollador, lo que permitió concluir que las que no se podían cruzar son prácticas o principios que hacen parte fundamental de la esencia de la autogestión porque están muy relacionadas con el ser y la forma de pensar y de ver la vida, Esto orientó la iniciativa de vincularlas dentro del framework como un nuevo criterio que cubriera estas prácticas y se le llamó el Manifiesto iBossS, soportando así que la autogestión es un estilo de vida que se debe asumir como una filosofía que debe acompañar el comportamiento de los desarrolladores que asuman el framework, entonces se definieron en total cuatro criterios:

- **Mejora continua:** Prácticas de autogestión que comprenden la mejora y el desarrollo a nivel personal y profesional.
- **Planeación de requerimientos:** Prácticas de autogestión que sugieren la planeación del trabajo a realizar durante el desarrollo de un requerimiento.
- **Producción de entregables de software:** Prácticas y principios de autogestión que se pueden aplicar durante la producción de los productos de trabajo de un desarrollador de software.
- **Manifiesto iBossS.** Prácticas y principios de autogestión que se deben aplicar para lograr un cambio permanente y trascendental en la forma de ser.

Cada uno de estos criterios se trabajó en los siguientes cuadros, donde se muestra como se relacionaron las prácticas de autogestión con las prácticas de la Ingeniería de Software.

Cuadro 5. Criterio de relación: Mejora continua

Práctica de Autogestión	Práctica de Ingeniería de software
Planifique su propio trabajo.	Para mejorar auténticamente su trabajo, los ingenieros deben usar procesos personales bien definidos y cuantificados.
Mantenga la atención en tu trabajo, identificando y reduciendo las distracciones.	Cada ingeniero es diferente; para ser más eficiente, debe planificar su trabajo basándose en datos tomados de su propia trayectoria profesional. Para obtener productos de calidad, el ingeniero debe asumir la responsabilidad personal de la calidad de sus productos. Los buenos productos no se obtienen por azar, sino como consecuencia de un esfuerzo positivo para hacer un trabajo de calidad. Trabajar bien es siempre la forma más rápida y económica.
Haga un análisis Feedback periódicamente teniendo en cuenta sus expectativas y decisiones tomadas.	Las medidas claves del rendimiento del proceso personal son: • La capacidad para cumplir con los compromisos programados para la entrega prometida • La calidad de los contenidos entregados
Concentrarse en las oportunidades.	
Gestione adecuadamente su tiempo.	
Fuente: Los Autores	

Cuadro 6. Criterio de relación: Planeación del Trabajo

Práctica de Autogestión	Práctica de Ingeniería de software
Haga un análisis Feedback periódicamente teniendo en cuenta sus expectativas y decisiones tomadas.	Planificar el trabajo: Las personas que hacen el trabajo son las más indicadas para planear el trabajo. Los individuos siempre deben desarrollar un plan antes de comprometerse con iniciar un proyecto. Los planes deben ser procesos definidos y con datos

Práctica de Autogestión	Práctica de Ingeniería de software
Planifique su propio trabajo.	históricos. Cuando el plan no se ajusta al trabajo hay que revisar.
Fuente: Los Autores	

Cuadro 7. Criterio de relación: Producción de entregables de software

Práctica de Autogestión	Práctica de Ingeniería de software
Defina una misión y objetivos consiga los recursos necesarios para trabajar en función de ellos.	Trabajar bien es siempre la forma más rápida y económica.
Mantenga el enfoque en lo que le importa.	Para obtener productos de calidad, el ingeniero debe asumir la responsabilidad personal de la calidad de sus productos. Los buenos productos no se obtienen por azar, sino como consecuencia de un esfuerzo positivo para hacer un trabajo de calidad.
Realice una sola tarea al tiempo, Si existen varias tareas, priorícelas y resuélvalas en el orden definido.	Planificar el trabajo: Las personas que hacen el trabajo son las más indicadas para planear el trabajo. Los individuos siempre deben desarrollar un plan antes de comprometerse con iniciar un proyecto. Los planes deben ser procesos definidos y con datos históricos. Cuando el plan no se ajusta al trabajo hay que revisar.
Traducir los planes de trabajo en acciones concretas.	Principio de Calidad: Para producir productos de calidad las personas deben sentirse personalmente responsables de la calidad de sus productos. Con el fin de

Práctica de Autogestión	Práctica de Ingeniería de software
Gestione adecuadamente su tiempo	crear productos de calidad consistente, las personas deben ser disciplinadas en la elaboración y seguimiento de los planes, el seguimiento y la gestión de su tiempo personal, y mantener la calidad como la principal prioridad.
Coordine sus actividades con los demás miembros de su equipo de trabajo.	
Haga uso de herramientas informáticas para facilitar el uso de las prácticas.	
Incorpore prácticas de autogestión en los procesos de negocio de su organización.	
Concéntrese en optimizar su rendimiento.	
Fuente: Los Autores	

Cuadro 8. Criterio de relación: Manifiesto iBossS

Práctica de Autogestión	Práctica de Ingeniería de software
Todos los principios influyen en el modo de operar del Desarrollador de Software.	Cada ingeniero es diferente; para ser más eficiente, debe planificar su trabajo basándose en datos tomados de su propia experiencia profesional.
Sea asertivo a la hora de resolver conflictos laborales de tal manera que a futuro éstos no afecten su trabajo.	Es más efectivo evitar los defectos que detectarlos y corregirlos. Como ejemplos: Codificación por Pares, Propiedad Colectiva de Código y Cliente en Sitio.
Comprométase con el logro de los objetivos planeados.	
Asumir la responsabilidad de comunicar.	

Práctica de Autogestión	Práctica de Ingeniería de software
Asuma responsabilidad por las decisiones	
Cree y mantenga buenas relaciones personales con sus compañeros de trabajo.	
Llame a las personas por su nombre.	
Desarrolle altas cualidades de relaciones, excelente desempeño, comunicación y visión. De ésta manera será visto como un verdadero líder.	
Fuente: Los Autores	

3.3.4 Paso 4: Especificar Lineamientos. Sobre los criterios y las prácticas relacionadas en el proceso anterior, se crearon los lineamientos de autogestión para el contexto del Desarrollo de Software. Estos lineamientos se convierten en la parte estructural y fundamental que le permite al framework iBossS preservar la teoría de autogestión. Se dice que son estructurales porque más adelante se verificará, en el punto 3.6.1, que se convierten en los requerimientos sobre los que se fundamenta el análisis y el diseño de la herramienta para iBossS. Para comprender como se logró obtenerlos, en el Cuadro 6 se muestran para cada Grupo de Criterios los lineamientos obtenidos, teniendo en cuenta el Manifiesto iBossS.

Cuadro 9. Grupos de Criterios y Lineamientos del Framework de Autogestión

Grupos de Criterios	Lineamiento
Fundamentos de autogestión	Manifiesto iBossS
Mejora Continua	Procedimiento de Autogestión
	Identificación de Distractores del Trabajo
	Lecciones Aprendidas
	Gestión del tiempo
Planeación de Requerimientos	Plan de Trabajo

Grupos de Criterios	Lineamiento
Producción de Entregables de Software	Área de Trabajo
	Artefactos de los Procesos de Calidad
	Artefactos de los Procesos de Autogestión
Fuente: Los Autores	

3.3.4.1 Lineamiento 1: Manifiesto de autogestión. Debe existir un código donde se expresen los principios y prácticas de autogestión que le permitan al Desarrollador de Software conocer la filosofía y los objetivos que se quieren lograr con el uso del framework de autogestión. Éste debe ser de fácil acceso y recordación para que el desarrollador lo consulte e interiorice.

3.3.4.2 Lineamiento 2: Procedimiento de autogestión. Debe existir un procedimiento de autogestión que permita incorporar las prácticas de autogestión en la planeación y regulación de las acciones que debe ejecutar el Desarrollador de Software.

3.3.4.3 Lineamiento 3: Identificación de Distractores del Trabajo. Debe proporcionarse un mecanismo que permita identificar las razones por las cuales se presentan distractores que generan interrupciones en el trabajo, para posteriormente actuar en la reducción de éstas.

3.3.4.4 Lineamiento 4: Lecciones Aprendidas. Debe proporcionarse un mecanismo que permita registrar las lecciones aprendidas a partir de experiencias propias y de los compañeros, corrección de errores, recomendaciones, oportunidades, aplicación de buenas prácticas, entre otras, e incentivar la cultura de revisarlas periódicamente.

3.3.4.5 Lineamiento 5: Plan de Trabajo. Debe proporcionarse un mecanismo apoyado en el Procedimiento de Autogestión, herramientas y/o artefactos que permitan planificar, organizar y controlar la ejecución de las actividades que hará el Desarrollador de Software. Permitiendo también, llevar un registro del avance de la carga de trabajo planeada versus la carga de trabajo realizada, para poder conocer teniendo en cuenta la variable tiempo, los valores presupuestados y el avance percibido. Haciendo referencia a lo expuesto en la medición de tiempos que hace Watts Humphrey en el PSP BOK (SEI, 2010, pág. 36).

3.3.4.6 Lineamiento 6: Área de Trabajo. Debe proporcionarse un mecanismo práctico que permita gestionar los requerimientos, organizándolos teniendo en cuenta las prioridades y configurando un espacio de trabajo donde se puedan almacenar los artefactos y entregables. Así se tendrá acceso rápidamente a la información que se ha producido del trabajo realizado por el desarrollador.

3.3.4.7 Lineamiento 7: Artefactos de los Procesos de Calidad. Se deben poder utilizar y organizar en un repositorio los artefactos que una organización productora de software ha definido dentro de sus procesos de calidad (Documento de Análisis, Documento de Diseño, etc.) Para que puedan ser utilizados fácilmente a la hora de traducir los planes de trabajo en acciones concretas.

3.3.4.8 Lineamiento 8: Artefactos de los Procesos de Autogestión. Crear un conjunto de artefactos de autogestión que permitan concentrar y guiar la ejecución de las actividades del proceso que se está siguiendo para la realización de cada requerimiento de trabajo y a la vez poder evidenciar el uso de las prácticas de autogestión.

3.3.4.9 Lineamiento 9: Gestión del tiempo. Se debe crear un mecanismo práctico que permita registrar y gestionar adecuadamente el tiempo invertido en las actividades de ingeniería. Estas actividades estarán enmarcadas por la definición de los pasos del proceso que se hace en PSP y visualizadas en la Figura 1. El registro de tiempo debe hacerse indicando y consolidando las actividades productivas y no productivas en las que se emplea. Éste recurso es valioso y debe poderse evaluar continuamente para minimizar sus desviaciones.

3.4 ARQUITECTURA CONCEPTUAL DE iBossS

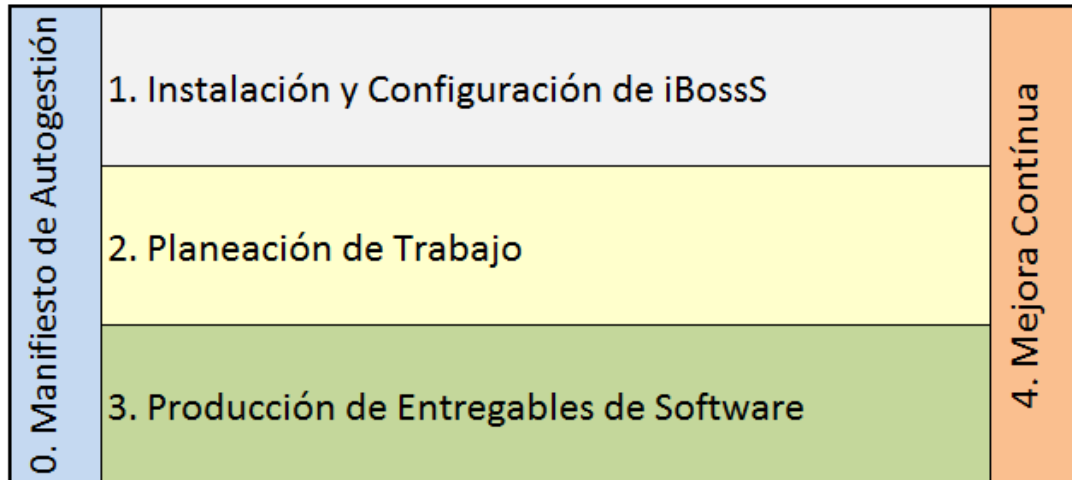
Teniendo en cuenta que iBossS es un marco de trabajo, con la arquitectura conceptual se quiere representar, en un nivel muy general, la dinámica que debe seguir el Desarrollador de Software para realizar sus labores utilizando iBossS.

Se diseñó teniendo en cuenta los lineamientos referidos en el Cuadro 6, con ellos se lograron definir las cinco capas que agrupan las diferentes tareas a ejecutarse, incluyendo la capa llamada instalación y configuración de la herramienta, porque como se indicó en los objetivos específicos, se desarrollará una herramienta piloto que deberá ser instalable en un computador.

La Figura 6 muestra la arquitectura conceptual y como se puede ver existen dos capas transversales, la Capa 0 y la Capa 4, con las que representan las prácticas

y principios que se deben seguir permanentemente en todas las etapas del trabajo. Las otras tres indican el paso a paso que se sigue para atender el desarrollo de un requerimiento. Después de tener la herramienta instalada (Capa 1), el programador hará la planeación de sus entregables (Capa 2) y llevará un registro del cumplimiento de las acciones planeadas (Capa 3).

Figura 8. Arquitectura conceptual del framework de autogestión



Fuente: Los autores

La organización de las capas indica que lo primero que debe hacer el desarrollador es reconocer el alcance y los fines de la aplicación de la teoría de autogestión asimilando y valorando el manifiesto, después instala y configura la herramienta iBossS, aclarando que la instalación se hace una sola vez, pero tendrá ejecutar iBossS cada vez que se tenga que desarrollar un nuevo requerimiento para la organización, esto es su ubicación y estructura. Posteriormente, avanzará progresivamente hasta las capas dos y tres. La capa de mejora continua representa que todo el trabajo se debe estar ejecutando sobre procesos de retroalimentación que permitan avanzar hacia los objetivos que propone la teoría de autogestión.

3.5 ARTEFACTOS iBossS

Teniendo en cuenta las capas de la Arquitectura Conceptual se construyeron los artefactos que se muestran en el Cuadro 7. Con cada uno de ellos se intenta dar cumplimiento a las prácticas de autogestión y a las tareas de desarrollo de

software que debe ejecutar el programador, incluido el artefacto para la sistematización del proceso.

Cuadro 10. Lista de Artefactos por capas del Framework

Artefactos	Capa del Framework	
<ul style="list-style-type: none"> - Manifiesto iBossS - Procedimiento de Autogestión - Carta de Compromiso - Documento Maestro - Artefacto para la sistematización: Herramienta iBossS 	0. Manifiesto de Autogestión - 4. Mejora Continua	<ul style="list-style-type: none"> - 1. Instalación y Configuración de la herramienta
<ul style="list-style-type: none"> - Artefacto para la sistematización: Herramienta iBossS 		<ul style="list-style-type: none"> - 2. Planeación de Entregables
<ul style="list-style-type: none"> - Procedimiento de Autogestión - Documento Maestro - Artefacto para la sistematización: Herramienta iBossS 		<ul style="list-style-type: none"> - 3. Producción de entregables de software
Fuente: Los Autores		

Como se ve en el Cuadro 7, se cuenta con cinco artefactos de trabajo, los cuales se listan y se describen a continuación:

1. **Manifiesto iBossS:** Principios y valores del comportamiento personal.
2. **Procedimiento de Autogestión iBossS:** Especificación paso a paso del procedimiento que se debe seguir para hacer autogestión con iBossS.
3. **Carta de compromiso:** Este instrumento contiene al Manifiesto iBossS (principios y valores), las expectativas, plan de vida, metas y logros enmarcados en sus experiencias laborales. Este documento representa el compromiso que tienen las personas que utilizarán el Framework de Autogestión iBossS para interiorizar la filosofía de autogestión en su forma de ser.
4. **Documento Maestro:** Archivo elaborado en Microsoft Word utilizado para registrar el seguimiento del procedimiento de autogestión en cada

requerimiento de desarrollo de software permite mejorar una lista de verificación de las actividades a tener en cuenta durante el desarrollo de un requerimiento.

5. **Artefacto para la Sistematización.** Es la herramienta iBoss desarrollada con macros en Microsoft Excel y será utilizada para asistir los procesos de autogestión que ejecutará el desarrollador de software, los aspectos concerniente a su diseño y desarrollado serán especificados en el punto número 3.6. La conforman dos plantillas:

- ✓ **iBossS:** Permite llevar un control de los parámetros y de los artefactos que hacen parte del framework, incluyendo aquellos que se utilicen para definir los requerimientos a ser desarrollados.
- ✓ **iBossS.Plan:** Extiende a la plantilla anterior, permitiendo la realización de la planeación y seguimiento de los trabajos a ser ejecutados para el desarrollo del requerimiento. Incluye informes relacionados con indicadores de gestión del valor devengado.

3.5.1 Manifiesto iBossS. En el Anexo 1 se presenta el manifiesto en toda su extensión, es un documento que hará parte permanente del framework y resalta los valores y principios que deben acompañar el comportamiento del desarrollador que se autogestiona, a continuación se presenta un resumen:

Presentación. Este instrumento contiene los principios y valores que rigen el comportamiento de las personas que utilizarán el Framework de Autogestión iBossS. Sobre esta premisa se deduce que los resultados esperados de su implementación, están directamente relacionados con la forma de ser de la persona.

Declaratoria. Todo Desarrollador de Software que haga uso de iBossS se propone proyectar su éxito laboral y personal aplicando los principios de autogestión que contiene el presente manifiesto y a estar sujeto a las normas éticas que su profesión establece.

Principios de Autogestión. iBossS comprende el siguiente grupo de principios:

- Principios propios
- Principios Inmutables
- Principios generales

Principios propios. Estos principios deben ser identificados por la persona.

Principios inmutables. Representan los criterios de la autogestión que resguardan los beneficios que se quieren lograr con la implementación de iBossS y se convierten en la base que cimientan el entorno en el que se debe desarrollar el framework.

- No ejercer poder en contra de la voluntad de sus compañeros de trabajo.
- Respetar y mantener los compromisos establecidos con los demás.
- Creer en ti mismo.
- Desarrollar altas cualidades de relaciones, excelente desempeño, comunicación y visión.
- Respetar los principios de autogestión.

Principios Generales. Soportan o apoyan la puesta en marcha de las prácticas o procesos de autogestión:

- Creer en sus compañeros de trabajo y mantener buenas relaciones personales con ellos.
- Conocer tus propias fortalezas.
- Ser consciente de las características bajo las cuales usted presenta mejor desempeño.
- Auto motívese constantemente durante la consecución de sus metas.
- Comprométase a cumplir las actividades que planea llevar a cabo.
- Planifique su propio trabajo.
- Mantenga el enfoque en lo que más importa.
- Identifique su estilo de comunicación (Lector u Oyente)
- Incorpore prácticas de autogestión en los procesos de negocio de su organización.

3.5.2 Procedimiento de Autogestión iBossS. Este procedimiento es un artefacto primordial del framework, se define como una secuencia documentada de los pasos que se proponen para adelantar la implementación de iBossS. A continuación se hace una descripción a groso modo de los diez pasos que se deben seguir, el detalle de cada uno y las acciones que se deben ejecutar se pueden observar en el Anexo 2, donde se muestra el formato de la Plantilla propuesta.

Los pasos a considerar dentro del procedimiento son los siguientes:

- **Control de versiones.** Se especifica la versión que se está trabajando a partir del formato propuesto por los autores del presente proyecto. Cada vez que se hagan las modificaciones se registraran los datos respectivos.
- **Propósito.** Se describe la razón que justifica la existencia y el propósito del procedimiento, teniendo en cuenta la especificación de las actividades de configuración y autogestión del framework.
- **Referencias.** Define la ruta física de los archivos que hacen parte de las referencias conceptuales del framework.
- **Conceptos.** Se definen los conceptos, términos, siglas y demás términos que necesiten ser aclarados para evitar ambigüedades y confusiones.
- **Configurar la herramienta iBossS.** Se definen los parámetros de configuración que permitirán llevar un registro de la productividad. Estos se refieren a las características con las que opera la hoja “Productividad” de la herramienta, entre las que se incluye el tiempo, especificación de los artefactos del sistema de calidad de la organización, las actividades de trabajo, entre otras.
- **Configurar la estructura del área de trabajo.** Estas especificaciones del procedimiento le permitirán configurar el área de trabajado dentro de la herramienta, la cual se hace en la hoja principal e incluye elementos como: configuración primaria, plantillas y documentación.
- **Configurar el Documento Maestro.** Este parámetro de configuración le permitirá llevar un registro de las actividades más relevantes del Ciclo de Vida del software que se están controlando con iBossS. El desarrollador es libre de definir las actividades dependiendo de los procedimientos que lleve a cabo la organización.
- **Actividades de Autogestión.** Se debe tener en claro que la autogestión debe cubrir todos los aspectos laborales; pensándolo bien de toda la vida. Por consiguiente este punto sugiere una serie de tips que le permitirán llevar un registro minucioso en el iBossS Plan, en las áreas de trabajo y en el documento maestro. Con esto se buscará apoyar el actuar disciplinado de la autogestión.
- **Hoja de Seguimiento y control.** Se detallan una serie de instrucciones que le permitirán verificar la evolución del trabajo ejecutado, con una gráfica que

le ayudará a conocer el consumo del tiempo empleado. Es muy importante que esta parte del procedimiento se haga con toda la atención del caso, porque involucra registro de tiempos y de avances.

- **Actividades de Ingeniería.** Se hacen las recomendaciones que se necesitan seguir para poder llevar el avance del proyecto de desarrollo de software.

3.5.3 Carta de Compromiso: Este instrumento contiene al Manifiesto iBossS (principios y valores), las expectativas, plan de vida, metas y logros enmarcados en sus experiencias laborales. Este documento representa el compromiso que tienen las personas que utilizarán el Framework de Autogestión iBossS para interiorizar la filosofía de autogestión en su forma de ser. El artefacto se incluye como Artefacto de Autogestión y se llama CartaDeCompromiso.V1.0.docx.

3.5.4 Documento Maestro – MD. Es un documento que se instancia en cada requerimiento para registrar la ejecución del Procedimiento de Autogestión que se sigue para cada requerimiento, determinando el estado de su evolución. En él se consignan las dudas, expectativas, modificaciones y los conflictos que se tengan frente a los diferentes equipos de trabajo que pueden trabajar en un proyecto. Este documento sirve además para garantizar la adherencia de la labor del desarrollador con el sistema de gestión de la calidad porque contiene una lista de chequeo que permite registrar la evolución del trabajo. El artefacto se incluye como Anexo 3.

3.5.5 Artefacto para la sistematización. Todo lo relacionado con la herramienta está contenido en el punto 3.6.

3.6 HERRAMIENTA iBossS

Para dar cumplimiento al tercer objetivo específico, en esta ocasión se desarrollará un prototipo piloto de la herramienta, tomando para su diseño la arquitectura 4+1 de Philippe Kruchten y su desarrollo se hará utilizando la programación de macros en Microsoft Excel 2007. Los macros son módulos escritos en Lenguaje VBA - Visual Basic for Applications.

3.6.1 Requerimientos

Los lineamientos definidos previamente, son los requerimientos del sistema:

- Lineamiento 1: Manifiesto de autogestión.
- Lineamiento 2: Procedimiento de autogestión.
- Lineamiento 3: Identificación de Distractores del Trabajo.
- Lineamiento 4: Lecciones Aprendidas.
- Lineamiento 5: Plan de Trabajo.
- Lineamiento 6: Área de Trabajo.
- Lineamiento 7: Artefactos de los Procesos de Calidad.
- Lineamiento 8: Artefactos de los Procesos de Autogestión.
- Lineamiento 9: Gestión del tiempo

3.6.2 Actores

- **Desarrollador:** El desarrollador es quien utilizará 100% el framework iBossS.
- **Responsable del Sistema de Gestión de la Calidad:** Interviene en los procesos de identificación de los activos organizacionales que apoyan las labores de construcción de software, además su conocimiento en el sistema de gestión de la calidad de la organización es importante para configurar las listas de verificación del documento maestro.

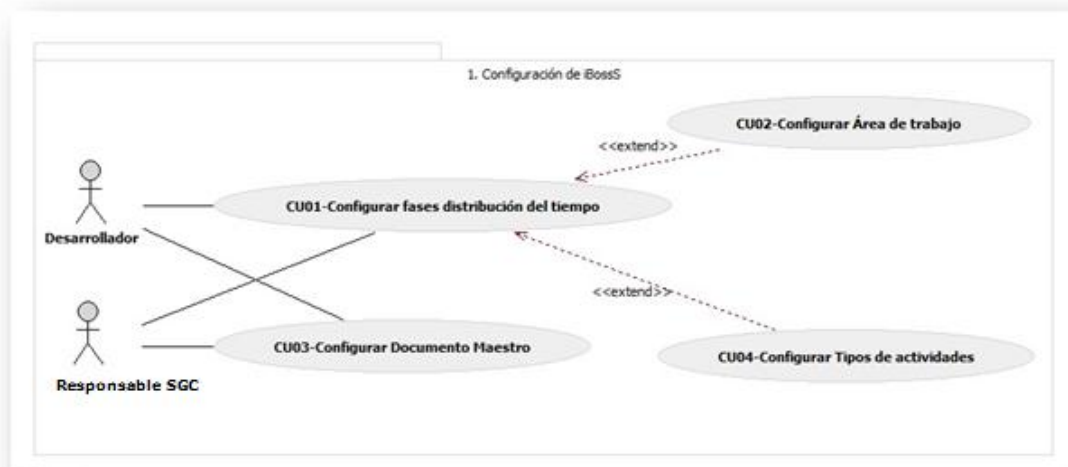
3.6.3 Vista de Casos de Uso

Se definen cuatro paquetes, cada uno relacionado con las capas de la Arquitectura Conceptual definida en el punto 3.4, para identificar los paquetes se utilizaron los colores que identifican cada una de las capas:

- **Paquete Configuración de iBossS.** Utilizando color gris se diagraman los casos de uso que atienden las funcionalidades de la instalación y configuración.
- **Paquete Planeación de Entregables.** Con color amarillo se representan los casos de uso que permiten hacer una planeación del trabajo a desarrollar en cada nuevo requerimiento.
- **Paquete Producción de Entregables.** Se muestran en color verde y representan las acciones que se ejecutan para la ejecución del plan de trabajo.
- **Paquete Mejora Continua.** Los casos de uso de este paquete se pintan con color naranja.

3.6.3.1 Paquete Configuración de iBossS. En este contenedor de casos de uso, se hace una descripción funcional de los escenarios involucrados en la Configuración del framework iBossS una vez se ha instalado la herramienta en el equipo del usuario final, como se muestra en el diagrama de la Figura 7. Para mayor información sobre la instalación de iBossS, vea el manual del usuario en el Anexo 4.

Figura 9. Diagrama Casos de uso Configuración de iBossS



Fuente: Los autores.

3.6.3.2 CU01-Configurar fases de distribución del tiempo. Permite configurar las fases que se utilizarán para clasificar la distribución de las actividades a realizar para cumplir con cualquier requerimiento, entre ellas se encuentran las fases del SDLC (Systems Development Life Cycle).

En el Cuadro 8 se hace una propuesta inicial de las actividades para la fase del SDLC.

Cuadro 11. Fases propuestas para el SDLC

Fases productivas	Descripción
FALTA REPORTAR	Identifica la cantidad de tiempo que falta por reportar en iBossS.
TRABAJO EXTRA	Identifica la cantidad de tiempo que se ha trabajado

Fases productivas	Descripción
	por fuera del horario laboral.
EVAL	Tiempo invertido en la evaluación de un requerimiento.
DIS	Tiempo invertido en el diseño de un requerimiento.
RDIS	Tiempo invertido en la revisión del diseño de un requerimiento.
CONS	Tiempo invertido en la fase de construcción de un requerimiento.
RCOD	Tiempo invertido en la fase de revisión del código de un requerimiento.
PRUEBAS	Tiempo invertido en la fase de pruebas unitarias.
ERROR	Tiempo invertido en errores, detección y corrección.
AUTOGESTION	Tiempo invertido en autogestión.
INTERR	Actividades que son importantes para el funcionamiento de la organización pero que no aportan en las labores del ciclo productivo de construcción de software. Aclarando que el desarrollador tiene la posibilidad de hacer una clasificación más detallada de estas actividades.
ALMUERZO	Tiempo invertido en almuerzo.
PERSONAL	Tiempo invertido en temas personales.
OTRO	Tiempo invertido en otras actividades.
Fuente: Los Autores	

Nota: Estas fases deben ser definidas con los líderes de procesos de la organización.

3.6.3.3 CU02-Configurar Área de trabajo. Define la estructura de directorios y plantillas de autogestión y del sistema de gestión de la calidad para establecer el área de trabajo personal para cada requerimiento. Esta funcionalidad se acompaña con el responsable del sistema de gestión de la calidad quien previamente debe haber identificado los activos organizacionales que serán instanciados en los directorios que aquí se definan.

Para mayor información vea el numeral 3.6.7.3.

3.6.3.4 CU03-Configurar Documento Maestro. Permite configurar las listas de chequeo para las actividades que según los procedimientos definidos por el sistema de gestión de la calidad, son necesarias para el cumplimiento del ciclo productivo en la construcción del software.

Para mayor información vea el Documento Maestro en el Anexo 3.

3.6.3.5 CU04-Configurar Tipos de actividades. Establece los tipos de actividades que comúnmente se utilizan en el contexto de trabajo del desarrollador. Con los tipos de actividades se permitirá un llenado rápido e inteligente de la actividad que se reporta en la componente TimeLog de iBossS. Se hace uso del identificador \$Sn donde n es un número consecutivo. \$Sn permite mediante la tecla tabulador desplazarse rápidamente en la escritura del texto de la actividad. El Cuadro 9 presenta una lista de actividades propuestas.

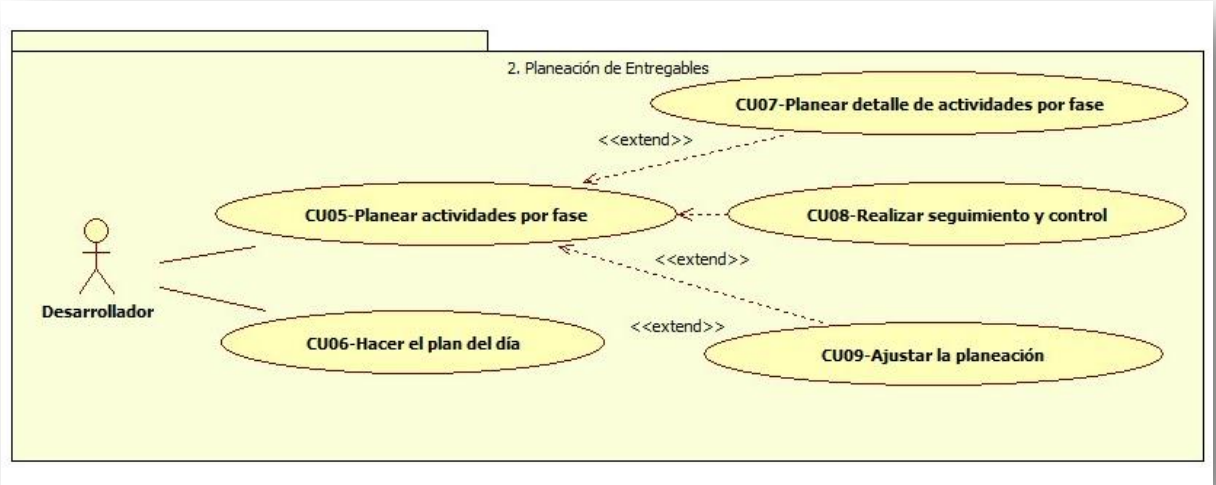
Cuadro 12. Tipos de actividades propuestos

Tipos de actividades
Café
Break
Almuerzo
WC
Pregunta de %S1 sobre %S2
Reunión de %S1 sobre %S2
Construcción del requerimiento %S1
Diseño del requerimiento %S1
Fuente: Los Autores

Nota: Los tipos de actividades iniciales deben ser definidas con los líderes de procesos de la organización.

3.6.3.6 Paquete: Planeación de Entregables. En este contenedor se hace una descripción funcional de los casos de uso involucrados en la Planeación de los entregables que se comprometan en la elaboración de un requerimiento de software. Como se muestra en el diagrama de la Figura 8.

Figura 10. Diagrama Casos de uso Planeación de Entregables



Fuente: Los autores.

3.6.3.7 CU05-Planear actividades por fase. Proporciona un mecanismo para realizar una planeación simple de las actividades necesarias para cumplir un requerimiento agrupando por las fases de distribución del tiempo que se configuraron en el caso de uso *CU01-Configurar fases de distribución del tiempo*.

3.6.3.8 CU06-Hacer el plan del día. Establece las actividades que se planean realizar en el día, de tal manera que estas se puedan marcar como realizadas.

3.6.3.9 CU07-Planear detalle de actividades por fase. Este caso de uso extiende al caso de uso CU05-Planear actividades por fase.

Para la elaboración de un requerimiento de software en el que el usuario considere necesario un mayor control, se debe poder hacer una planeación más detallada de las actividades necesarias para cumplir con un entregable y en función del tiempo. Se deben poder generar gráficas de la evolución del trabajo en función de indicadores de costos.

Para mayor información sobre la planeación detallada, diríjase al **Anexo 4 Manual del usuario** en la sección Planeación.

3.6.3.10 CU08-Realizar seguimiento y control. Este caso de uso extiende al caso de uso CU05-Planear actividades por fase. Cada vez que se cumplan las fechas de control, es necesario revisar el tiempo real vs el tiempo planeado para determinar el valor devengado del tiempo y conocer el estado de la evolución del requerimiento. Se utilizará el indicador EV/AC (valor devengado / costo actual) con el cual si el valor es por encima del 100% nos indica que estamos bien en el período de tiempo analizado, si está por debajo del 100% es necesario tomar acciones para controlar la evolución del requerimiento.

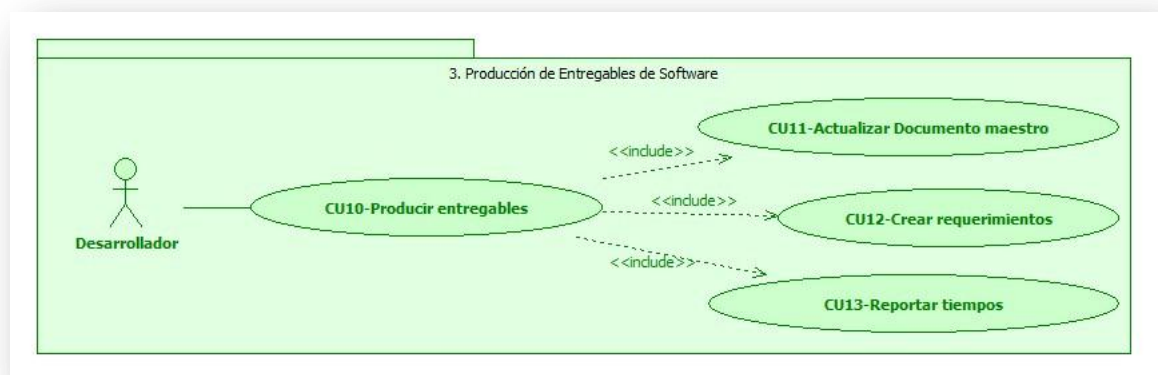
Para mayor información sobre la planeación detallada, diríjase al **Anexo 4 Manual del usuario** en la sección seguimiento y control.

3.6.3.11 CU09-Ajustar la planeación. Este caso de uso extiende al caso de uso CU05-Planear actividades por fase.

Permite crear, modificar, eliminar e insertar nuevas actividades para el cumplimiento de los entregables de software.

3.6.3.12 Paquete: Producción de Entregables de Software. En este contenedor se hace una descripción funcional de los casos de uso de producción para la producción de entregables de software.

Figura 11. Diagrama Casos de uso Producción de Entregables de Software



Fuente: Los autores.

3.6.3.13 CU10-Producir entregables. El sistema proporciona el área de trabajo con los activos organizacionales referenciados (ver descripción del artefacto *Paquete ejecutable iBossS* en el capítulo 3) de tal manera que se puedan copiar y diligenciar en la producción de entregables de software, para lo cual el usuario deberá conocer la ubicación de los requerimientos para los que está trabajando. Y deberá trabajar en el cumplimiento de las componentes requeridas por las plantillas organizacionales. Estos activos organizacionales son los entregables que el usuario va produciendo en el transcurrir del tiempo.

3.6.3.14 CU11-Actualizar Documento Maestro. Permite marcar la lista de verificación de las actividades que se van cumpliendo a medida que se avanza en la producción de entregables de software.

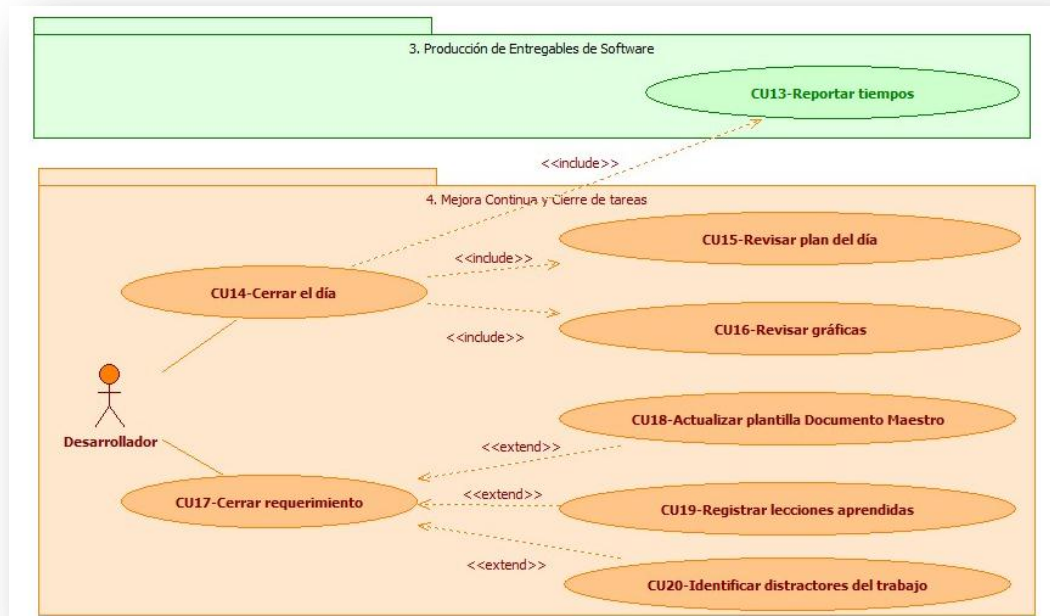
- Para mayor información vea el capítulo 3 artefacto Documento Maestro – MD.
- El documento maestro se encuentra en el **Anexo 3 Documento Maestro**.

3.6.3.15 CU12-Crear requerimientos. Permite la creación de un requerimiento, para ello instancia la estructura de directorios (área de trabajo) configurada. Se solicitan los datos básicos para la creación del requerimiento: Identificador, Nombre, Alias, ruta (por defecto se utiliza la configurada).

3.6.3.16 CU13-Reportar tiempos. Inserta, actualiza y/o elimina en un sistema legado, los tiempos registrados agrupándolos en actividades por usuario y por día.

3.6.3.17 Paquete: Mejora Continua y Cierre de Tareas. En este contenedor se hace una descripción funcional de los casos de uso de Mejora continua y Cierre de tareas. Como se muestra en la Figura 9.

Figura 12. Casos de uso Mejora Continua y Cierre de Tareas



Fuente: Los autores

3.6.3.18 CU14-Cerrar el día. Cada vez que el usuario reporta una nueva actividad en el mismo día, lo cierra e identifica con color verde claro si el día sábado o domingo, de lo contrario lo pinta con color azul claro. En caso de encontrar solapamientos de tiempos entre las actividades registradas, los pinta en color rojo.

3.6.3.19 CU15-Revisar plan del día. Muestra las actividades planeadas para el día actual, y permite que el usuario las modifique.

3.6.3.20 CU16-Revisar gráficas. Muestra las gráficas de distribución del tiempo en el día seleccionado por el usuario.

3.6.3.21 CU17-Cerrar requerimiento. Comprime el contenido de cada directorio hijo del área de trabajo del requerimiento que se está cerrando. Y Muestra un reporte de distribución del tiempo de todo el requerimiento.

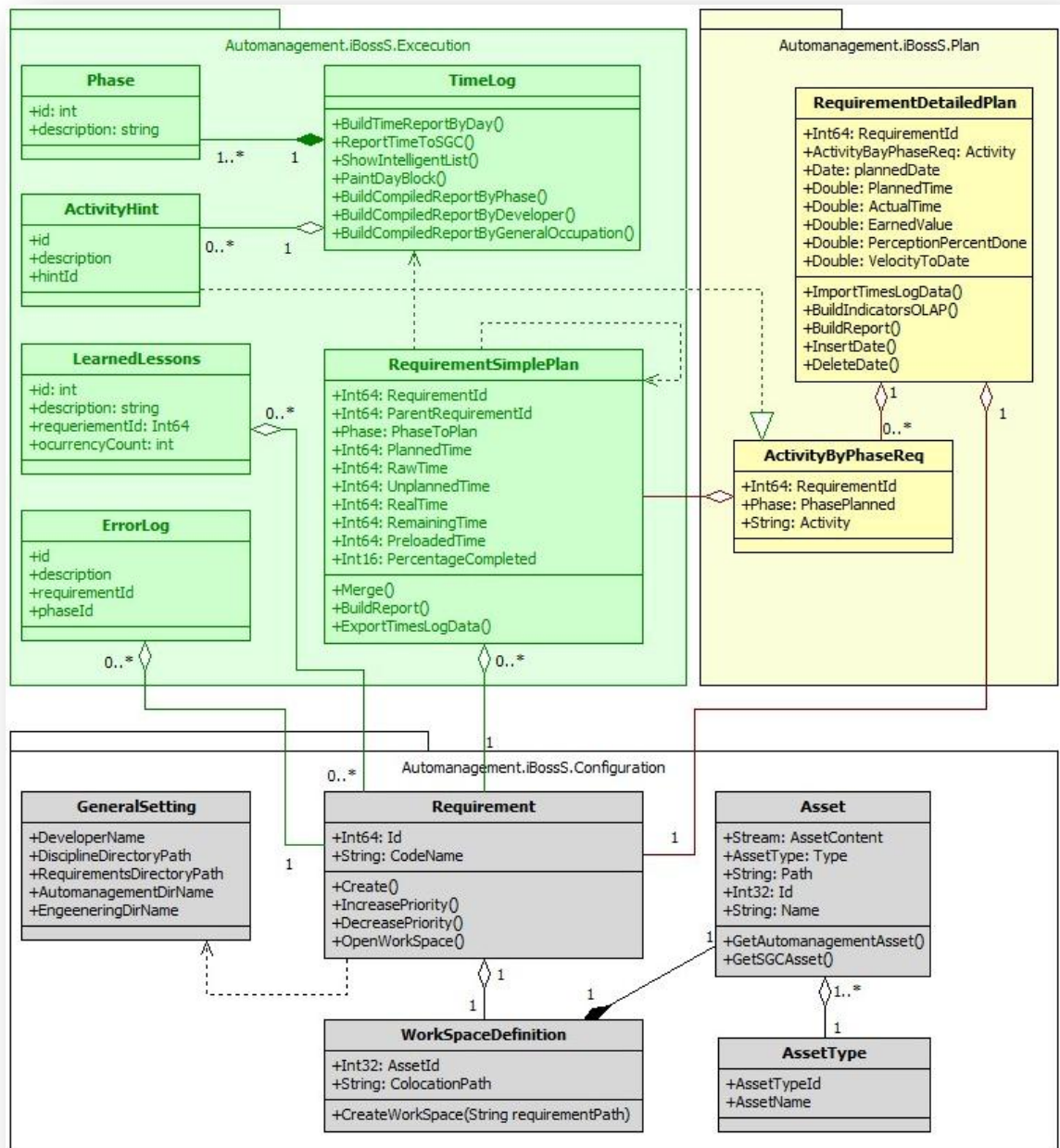
3.6.3.22 CU18-Actualizar plantilla Documento Maestro. Crea, elimina y modifica las actividades configuradas en el documento maestro. De tal manera que en un nuevo requerimiento de trabajo haya mejor desempeño y calidad dada la experiencias vivida por el usuario durante el desarrollo del requerimiento.

3.6.3.23 CU19-Registrar lecciones aprendidas. Crea, elimina y modifica las lecciones que el usuario aprendió en el desarrollo del requerimiento. Solicitando los datos básicos: Identificador del requerimiento, Nombre de la lección aprendida, descripción de la lección aprendida y cantidad de veces (Ocurrencia) que se presentó la oportunidad de mejora.

3.6.3.24 CU20-Identificar distractores del trabajo. Muestra un reporte de todas las actividades tipo INTERRUPCIÓN que se presentaron durante el desarrollo del requerimiento. Esto con el fin de que el usuario tome medidas preventivas y correctivas para mejorar la atención y calidad de su trabajo personal.

3.6.4 Vista de Diagrama de Clases

Figura 13. Vista de Diagrama de Clases



Fuente: Los autores

3.6.4.1 Paquete: Automanagement.iBossS.Excecution

3.6.4.1.1 Clase: Phase. Identifica una fase o agrupador de la distribución del tiempo.

3.6.4.1.2 Clase: ActivityHint. Representa un acceso rápido a una actividad planeada, el hint se accede con el carácter \$.

3.6.4.1.3 Clase: LearnedLessons. Representa una lección aprendida durante el desarrollo de software.

3.6.4.1.4 Clase: ErrorLog. Representa el log de errores, que se van encontrando en cualquier fase del desarrollo de software.

3.6.4.1.5 Clase: TimeLog. Representa un instante de tiempo invertido por fase, día, hora inicial, hora final y por requerimiento.

3.6.4.1.6 Clase: RequirementSimplePlan. Gestiona la planeación simple de un requerimiento, registrando las actividades en su orden precedente hasta el logro de los entregables a producir.

3.6.4.2 Paquete: Automanagement.iBossS.Plan

3.6.4.2.1 Clase: RequirementDetailedPlan. Permite hacer seguimiento y control y gestionar las actividades que componen un la elaboración de un requerimiento, registrar los tiempos planeados, reales, valor devengado y demás variables de gestión del valor ganado.

Para una descripción detallada de los indicadores de gestión del valor ganado, Ver **Anexo 3 Manual del Usuario** en la sección seguimiento y control.

3.6.4.2.2 Clase: ActivityByPhaseReq. Representa una actividad que se gestiona con indicadores de gestión del valor devengado.

Para una descripción detallada de los indicadores de gestión del valor ganado, Ver **Anexo 3 Manual del Usuario** en la sección seguimiento y control.

3.6.4.3 Paquete: Automanagement.iBossS.Configuration

3.6.4.3.1 Clase: GeneralSetting. Gestiona y aplica las configuraciones generales del área de trabajo de autogestión, de desarrollos nuevos de la estructura de cada nuevo requerimiento.

3.6.4.3.2 Clase: Requirement. Representa un requerimiento de software.

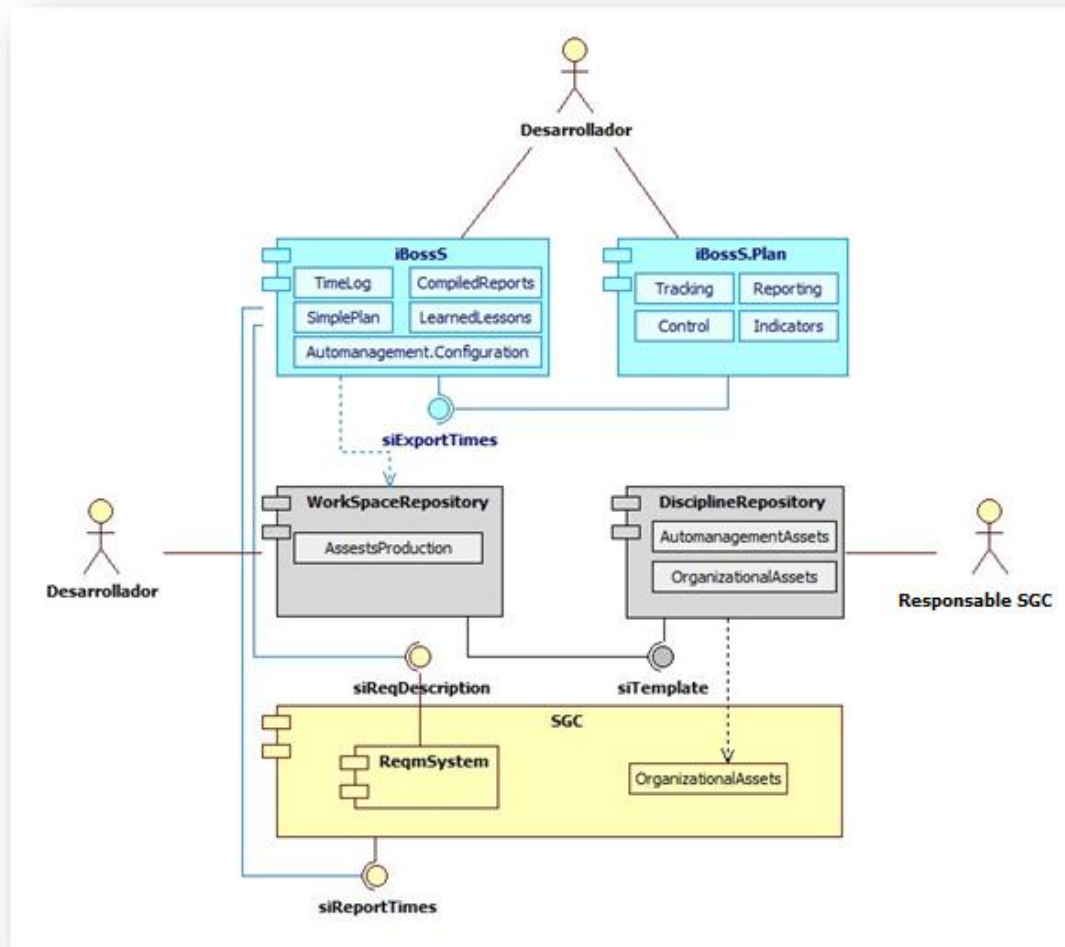
3.6.4.3.3 Clase: WorkspaceDefinition. Gestiona la configuración de la estructura del área de trabajo de cada nuevo requerimiento. También permite la creación de la estructura de un nuevo requerimiento.

3.6.4.3.4 Clase: Asset. Flujo de bits que representa un activo organizacional o de autogestión.

3.6.4.3.5 Clase: AssetType. Representa un tipo de activo que puede ser organizacional o de autogestión.

3.6.5 Vista de Desarrollo

Figura 14. Vista de Diagrama de Componentes



Fuente: Los autores

3.6.5.1 Componente: iBossS. Aplicativo elaborado en macros de Microsoft Excel que permitirá integrar y asistir los procesos de autogestión en del desarrollo de software. Vea la descripción de esta plantilla en el capítulo 3.

3.6.5.2 Componente: iBossS.Plan. Aplicativo elaborado en macros de Microsoft Excel que permitirá planear y gestionar la evolución del trabajo mediante indicadores de gestión del valor ganado.

Para una descripción detallada de los indicadores de gestión del valor ganado, Ver **Anexo 3 Manual del Usuario** en la sección seguimiento y control.

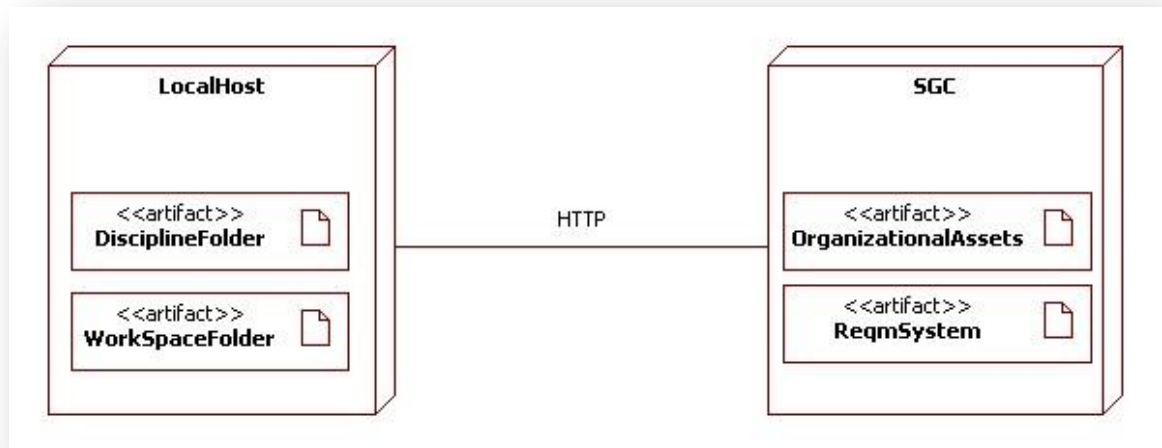
3.6.5.3 Componente: WorkspaceRepository. Representa el directorio de trabajo de cada requerimiento nuevo que se instancia según la configuración realizada en la componente iBossS.

3.6.5.4 Componente: DisciplineRepository. Representa el directorio de donde se alojan los artefactos de la filosofía de autogestión y del framework iBossS.

3.6.5.5 Componente: SGC. Sistema legado donde se encuentran las plantillas organizacionales que se instanciarán en cada nuevo requerimiento, además se hace referencia al sistema de gestión de requerimientos con el cual iBossS podría integrarse para magnificar las posibilidades de gestionar requerimientos

3.6.6 Vista de Despliegue

Figura 15. Vista de despliegue



Fuente: Los autores

3.6.6.1 Nodo: Localhost. Este es el computador donde se instala la herramienta. Ésta consta de dos artefactos principales que pueden estar distintos directorios. El artefacto DisciplineFolder contiene todas las plantillas de autogestión propuestas para iBossS y el artefacto WorkspaceFolder es el directorio de trabajo donde se producen los entregables de cada nuevo requerimiento de trabajo.

3.6.6.2 Nodo: SGC. Es el servidor donde se encuentran las plantillas organizacionales que se instanciarán en cada nuevo requerimiento, además se hace referencia al sistema de gestión de requerimientos con el cual iBossS podría integrarse para magnificar las posibilidades de gestionar requerimientos.

Este nodo representa el repositorio con el cual iBossS podría integrarse para reportar los tiempos que se registran en el transcurrir del desarrollador de software.

3.6.7 Desarrollo

El resultado final de la etapa de desarrollo permitió lograr la herramienta iBossS con las siguientes características físicas:

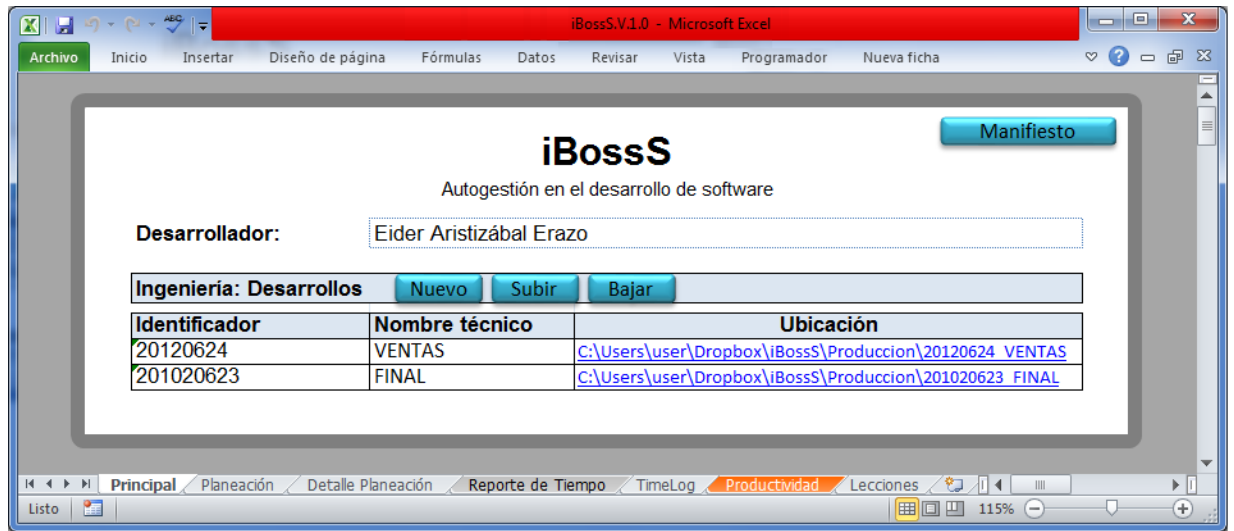
- Plantilla iBossS
- Herramienta iBossS.Plan
- Paquete ejecutable iBossS
- Espacio de trabajo
- Artefactos de autogestión y del sistema de gestión de la calidad

3.6.7.1 Herramienta iBossS. Esta es la herramienta principal que se utilizará para asistir los procesos de autogestión en del desarrollo de software, su apariencia se muestra en la Figura 14, pero la especificación de sus características y el funcionamiento está detallado en el Anexo 4 Manual del Usuario.

En esta plantilla se pueden crear requerimientos, priorizarlos, hacer una planeación sencilla, registrar tiempos de las actividades realizadas durante la producción de entregables, revisar reportes etc. Está compuesta por las siguientes hojas:

- Principal
- Planeación
- Detalle Planeación
- Reporte de Tiempo
- TimeLog
- Productividad
- Lecciones Aprendidas.

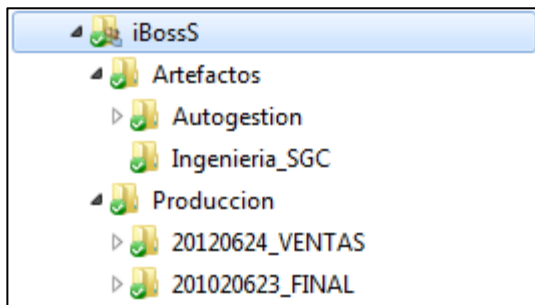
Figura 16. Interfaz gráfica herramienta iBossS



Fuente: Los Autores

3.6.7.2 Paquete ejecutable iBossS. El paquete ejecutable iBossS es una carpeta que se debe instalar (copiar) en el computador del desarrollador de software, en ella se aloja todo el cuerpo teórico de autogestión y la herramienta iBossS, de tal manera que el desarrollador de software comience a producir entregables para el Sistema de Gestión de la Calidad. La estructura final de la carpeta se puede ver en la Figura 17. Estructura del Paquete

Figura 17. Estructura del Paquete



Fuente: Los autores

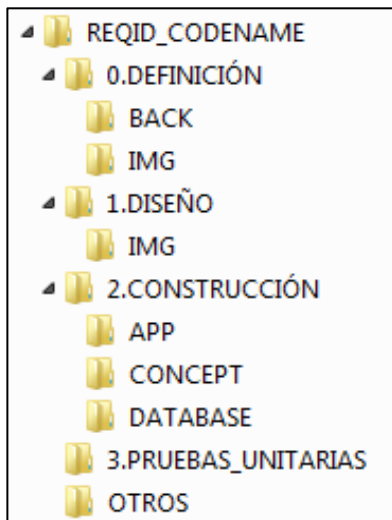
La carpeta iBossS está dividida en subcarpetas:

- **iBossS:** Carpeta raíz del framework de autogestión iBossS.

- **Artefactos:** Carpeta principal de los artefactos para apoyar la planeación del trabajo y la producción de los entregables.
 - **Autogestión:** Contiene los artefactos y herramientas de autogestión
 - **Ingeniería_SGC:** Contiene los artefactos que la organización defina para los procesos de desarrollo de software.
- **Producción:** Aquí se alojan los requerimientos que se producen, a manera de ejemplo, se listan los siguientes:
 - 20120624_VENTAS
 - 201020623_FINAL

3.6.7.3 Espacio de trabajo. El espacio de trabajo es otra estructura de carpetas diferente a la descrita anteriormente. El espacio de trabajo corresponde a la estructura que se arma cada vez que se crea un nuevo requerimiento. Aquí se alojará una copia de cada documento plantilla que apoya la labor del desarrollador de software. Se propone una estructura que puede ser modificada en iBossS (para modificaciones vea el manual de usuario).

Figura 18. Estructura propuesta



Fuente: Los autores

- **REQID_CODENAME** – Carpeta raíz del requerimiento que se desarrolla.
- **0. DEFINICIÓN.** Carpeta donde se alojan los archivos referentes a la definición del requerimiento que se desarrolla, Por ejemplo, aquí se podría alojar una plantilla del sistema de gestión de la calidad para definición de requerimientos.
 - BACK – Carpeta donde se alojan las copias de los archivos que requieran respaldo

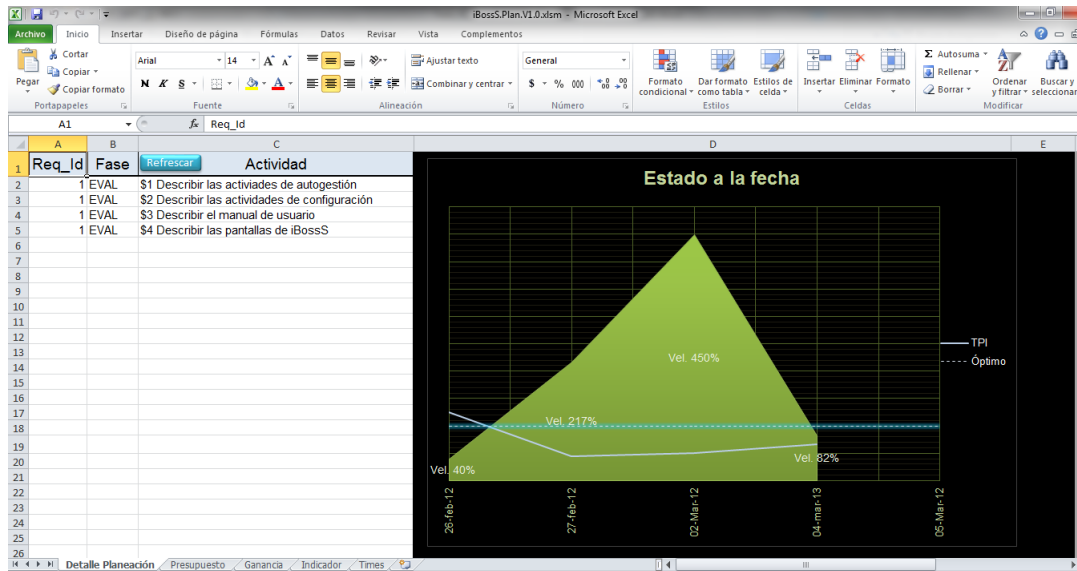
- IMG – Carpeta donde se alojan las imágenes de los prototipos como por ejemplo pantallas del requerimiento que se desarrolla.
- **1. DISEÑO.** Carpeta donde se alojan los archivos referentes al diseño del requerimiento que se desarrolla, por ejemplo un documento de arquitectura del sistema de gestión de la calidad o una plantilla de arquitectura UML.
 - IMG: Carpeta donde se alojan las imágenes del diseño entre ellas, pantallas, vistas de arquitectura, procesos etc.
- **2. CONSTRUCCIÓN.** Carpeta donde se aloja el código fuente del requerimiento que se desarrolla.
 - **APP** – Carpeta donde se aloja el código de la aplicación
 - **CONCEPT** – Carpeta donde se aloja el código de componentes de prueba o código de prueba para propuestas o alternativas.
 - **DATABASE** – Carpeta donde se alojan archivos de base de datos como scripts SQL DDL, DML etc.
- **3.PRUEBAS_UNITARIAS** – Carpeta donde se alojan los archivos y evidencias de pruebas de unidad.
- **OTROS** – Carpeta para almacenar otros archivos referentes al requerimiento que se desarrolla.

3.6.7.4 Herramienta iBossS.Plan. Esta herramienta permite hacer una planeación más detallada del requerimiento que se desarrolla, proporcionando una gráfica que representa la velocidad del trabajo que se realiza. Para ello se basa en los indicadores de gestión del valor ganado. La fuente de datos para hacer el análisis y cálculo de indicadores los toma de los registros del TimeLog que residen dentro de iBossS. Está compuesta por las hojas:

- Detalle de planeación
- Presupuesto
- Ganancia
- Indicador
- Times

Su presentación gráfica se muestra en la Figura 14. La especificación de sus características y el funcionamiento esta detallado en el Anexo 4.

Figura 19. Interfaz gráfica iBossS.plan



Fuente: Los Autores

3.7 ARTEFACTOS Y HERRAMIENTAS Vs. PRÁCTICAS DE AUTOGESTIÓN

En el Anexo 5 se muestran las matrices que se utilizaron para verificar la relación que existe entre las prácticas de autogestión propuestas por los diferentes autores referidos en el marco teórico y los artefactos y la herramienta que conforman el Framework iBossS, de este trabajo se pudo constatar que en un alto porcentaje, las prácticas son cubiertas.

Para tener una ilustración de cómo se hizo el proceso, se muestra en el Cuadro 12, la matriz que se utilizó para validar las practicas o principios propuesto por Peter Drucker, verificando que se cumple con 11 de las prácticas que se habían identificado, dando como resultado, para este caso, un cubrimiento del 92%.

Cuadro 13. Prácticas de Peter Drucker

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
1	Haga un análisis Feedback periódicamente					X	X
1.1	Concentrarse en sus fortalezas		X	X			
1.2	Trabajar en mejorar las fortalezas.	X		X			
1.3	Superar la ignorancia paralizante (Arrogancia)	X		X			
1.4	Corregir los malos hábitos (Que generan pérdidas de eficiencia)		X	X	X	X	
1.5	Practicar los buenos modales al comunicarse con las personas	X	X	X			
1.6	Reconocer que no se puede o no se es capaz de hacer (Humildad)		X	X	X	X	
2	Ser consciente de las características bajo los cuales usted presenta mejor desempeño	X	X			X	X
3	Definir el estilo de comunicación (lector u oyente)	X			X	X	
4	Entender cómo se aprende (lector u oyente)	X			X	X	
5	Definir la capacidad de trabajar en equipo	No aplica para éste trabajo de tesis.					
6	Prever si se está en la capacidad de tomar decisiones o de asesorar						
7	Identificar las características de su desempeño en diferentes entornos de trabajo	X					
8	Identificar cuáles son sus valores personales.		X				
9	Decidir cuál es el lugar a ocupar dentro de la organización		X				
10	Contribuir con el éxito	X					
11	Responsabilizarse de las relaciones interpersonales	X					
12	Gestionar la segunda mitad de su vida		X				

Fuente: Los Autores

4. PRUEBA DE CONCEPTO DEL FRAMEWORK PROPUESTO

La prueba de concepto se diseñó para validar las expectativas del framework, sobre la pretensión de impactar positivamente el desempeño del desarrollador de software con una propuesta que intenta aprovechar los principios y las prácticas de autogestión para buscar la excelencia y un mejor rendimiento del programador de software.

Para realizarla la prueba de concepto se diseñó un plan con las siguientes actividades:

- Características de la prueba.
- Equipo evaluador.
- Respuestas y análisis de resultados.
- Resumen de las observaciones.

4.1 CARACTERÍSTICAS DE LA PRUEBA

Se planteó que la prueba fuera realizada por personas que se dediquen al desarrollo de software en diferentes empresas de la ciudad, tomando como característica importante la experiencia. Esta decisión se consideró conveniente, porque iBossS está estrechamente ligado con la forma de ser y la forma en la que la persona asume sus responsabilidades, colocando el primer foco de atención en la actitud de cada uno de los individuos.

Entonces, como primer paso se programó una charla que buscó dar a entender el alcance que busca la autogestión y el sentido de iBossS. Después se presentó un cuestionario con diez preguntas, las cuales deberían ser calificadas teniendo en cuenta la escala y los rangos mostrados en la

La escala “Bueno”, entre 4.6 y 5.0, permitirá verificar que se ha cumplido con el objetivo propuesto y que el Framework iBossS satisface el alcance esperado. En el caso contrario, si se califica como “Deficiente”, menor o igual a 2.0, se podrá deducir que la propuesta no cumplió con las expectativas.

Para efectos de lograr un equilibrio en la escala y para dar la oportunidad a que los evaluadores propongan desde su experiencia y desde su percepción de las cosas, se incluyeron las escalas “Regular” y “Aceptable”. En la primera, para valores de 2.1 a 3.5, el evaluador tendrá la oportunidad de calificar que al framework le faltan mejoras considerables para acercarse a lo propuesto y la

segunda, con calificación entre 3.6 y 4.5, permitirá sugerir que faltan algunos ajustes para llegar al objetivo.

Las preguntas y sus respuestas se presentan en el punto de análisis de resultados.

Tabla 1. Escala y rangos de valoración.

La escala “Bueno”, entre 4.6 y 5.0, permitirá verificar que se ha cumplido con el objetivo propuesto y que el Framework iBossS satisface el alcance esperado. En el caso contrario, si se califica como “Deficiente”, menor o igual a 2.0, se podrá deducir que la propuesta no cumplió con las expectativas.

Para efectos de lograr un equilibrio en la escala y para dar la oportunidad a que los evaluadores propongan desde su experiencia y desde su percepción de las cosas, se incluyeron las escalas “Regular” y “Aceptable”. En la primera, para valores de 2.1 a 3.5, el evaluador tendrá la oportunidad de calificar que al framework le faltan mejoras considerables para acercarse a lo propuesto y la segunda, con calificación entre 3.6 y 4.5, permitirá sugerir que faltan algunos ajustes para llegar al objetivo.

Las preguntas y sus respuestas se presentan en el punto de análisis de resultados.

Tabla 1. Escala y rangos de valoración

Escala	Rango
Deficiente	Calificación ≤ 2
Regular*	2.1 \leq Calificación ≤ 3.5
Aceptable**	3.6 \leq Calificación ≤ 4.5
Bueno	4.6 \leq Calificación ≤ 5.0

* Mejoras considerables

** Se necesitan algunos ajustes

Fuente Los Autores

4.2 EQUIPO EVALUADOR

Se solicitó la colaboración a diez profesionales dedicados al desarrollo de software con una experiencia superior a los cinco años, trabajan para empresas dedicadas a la industria del software y con estándares de calidad para los que se requiere un excelente desempeño.

En el Cuadro 14, se indican los perfiles y la experiencia de los desarrolladores que tuvieron la gentileza de colaborar con la prueba y de dar a conocer sus apreciaciones.

Cuadro 14. Equipo Evaluador

Desarrollador	Perfil Profesional	Experiencia
Luis Eduardo Ferro	Ingeniero de Sistemas. Tecnólogo en Sistemas. Oracle Certified Professional Java Programmer (OCPJP)	Experiencia como desarrollador de Software por más de cinco años. Trabaja actualmente como Desarrollador Senior en COOMEVA EPS
Brayham Camilo Peña	Ingeniero de Sistemas Tecnólogo en Sistemas Programador en Java	Experiencia de 6 años como desarrollador de software, trabaja actualmente para la empresa IT SOLUTIONS.
Ricardo Alejandro Morales Pinilla	Ingeniero de Sistemas	Cinco años de experiencia. Empresa actual AXTEL.
Alvaro Hernán Buitrago	Ingeniero de Sistema	15 años de experiencia como desarrollador OpenSystems, 10 como analista de infraestructura en Colombina.
Víctor Manuel Barona Guzmán	Ingeniero de Sistemas	7 años de experiencia. Trabajo actual Ingeniero de Desarrollo PRPÁ.
Julio César Robles Uribe	Ingeniero de Sistemas	18 años de experiencia en OpenSystems.
Karina Johana Cerón	Ingeniero de Sistemas	Tres años de experiencia.

Desarrollador	Perfil Profesional	Experiencia
		Empresa actual OPEN SYSTEMS.
Israel Perdomo	Ingeniero de Sistemas	3 años de experiencia como Desarrollador Junior de OpenSystems, en el equipo de Framework
Fuente Los autores		

4.3 RESPUESTAS Y ANÁLISIS DE RESULTADOS

Para determinar los aportes y el impacto de iBossS, los asistentes respondieron las siguientes preguntas. Cada uno de los puntos se analizó y se tabuló haciendo uso de cuadros de resumen y figuras que contienen gráficos circulares; los cuales permiten representar los resultados para corroborar las conclusiones que acompañan las preguntas.

Se destaca que para cada pregunta, el evaluador tenía la posibilidad de retroalimentar a los autores con sus comentarios u observaciones.

Los resultados de la evaluación para cada una de las preguntas es el siguiente:

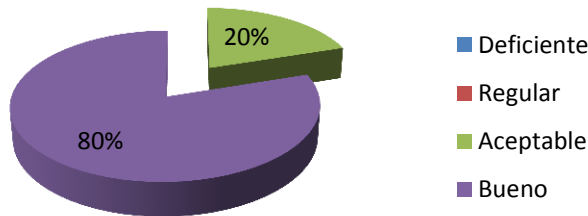
1. ¿Usted reconoce la importancia y el alcance de la teoría de autogestión?

En el Cuadro 15. Resultados pregunta número 1 y en la Figura 20. Resumen resultados pregunta número 1, se muestra que el 80% está de acuerdo en señalar que reconocen la importancia de la autogestión, razón que permite validar que las personas harán una evaluación más acertada del impacto o del alcance del framework. El otro 20% recomendó mejorar algunos aspectos para concretar y definir los elementos de la teoría de autogestión basándose en ejemplos más cercanos al tema del desarrollo de software o de tecnología.

Cuadro 15. Resultados pregunta número 1

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	2,0	20%
Bueno	8,0	80%
Promedio: 4,8		
Escala: Bueno		
Fuente Los Autores		

Figura 20. Resumen resultados pregunta número 1



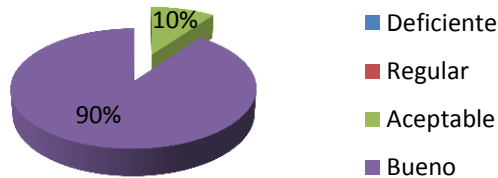
2. ¿Reconoce el aporte que puede llegar a representar dentro del trabajo del Desarrollador de Software el Manifiesto iBossS que contiene el presente framework?

El Cuadro 16. Resultados Pregunta Número y la Figura 21. Resumen Resultados Pregunta Número 2, muestran que el 90% comparten la idea de que el código de ética es muy importante y que su aplicación le permitiría al programador evitarse muchos problemas, también es de anotar que el 10% que calificó cómo regular lo hizo registrando una observación, la cual se considera prudente, porque se señala que hay otros elementos que sería prudente incluir, en especial aquellos relacionados con los estándares de calidad.

Cuadro 16. Resultados Pregunta Número 2

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	1,0	10%
Bueno	9,0	90%
Promedio: 4,9		
Escala: Bueno		
Fuente Los Autores		

Figura 21. Resumen Resultados Pregunta Número 2



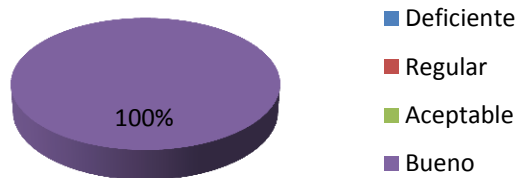
3. ¿Usted reconoce el alcance del framework iBossS al intentar incluir dentro del contexto del trabajo del desarrollador de software la teoría de autogestión?

El 100%, como se muestra en el Cuadro 17. Resultados pregunta número 3 y la Figura 22. Resumen resultados pregunta número , reconoce la intención del framework, para facilitar al desarrollador de software que auto gestione su propio trabajo como un mecanismo que le permita cumplir con los compromisos o como lo señalaba una de las observaciones anotadas por uno de los evaluadores, algunos no están conscientes de la importancia que representa la palabra planear. Cabe resaltar que para dar mayor claridad a esta pregunta se utilizó la descripción del concepto “profesional en autogestión” que hace Gary Hamel, cuando lo define como el gerente que planifica, organiza, dirige y controla su trabajo.

Cuadro 17. Resultados pregunta número 3

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	0,0	0%
Bueno	10,0	100%
Promedio: 5		
Escala: Bueno		
Fuente Los Autores		

Figura 22. Resumen resultados pregunta número 3



4. ¿Considera usted que los lineamientos de autogestión especificados en la herramienta iBossS son suficientes y pueden llegar a convertirse en un mecanismo válido que contribuya con el buen desempeño del desarrollador?

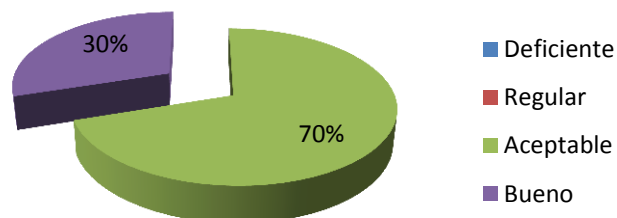
Para entender esta pregunta cabe recordar que los lineamientos considerados dentro del framework son los siguientes: Manifiesto iBossS, Procedimiento de Autogestión, Identificación de Distractores de Trabajo, Lecciones Aprendidas, Plan de Trabajo, Área de Trabajo, Artefactos de Proceso de Calidad y Artefactos de los Procesos de Autogestión. En este caso el Cuadro 18. Resultados pregunta número 4 y la Figura 23. Resumen resultados pregunta número 4 demuestran que el 70% opina que es regular aclarando que el caso de los distractores de trabajo deberían tener mayor relevancia porque según su experiencia, estos se convierten en un elemento que afecta la productividad y que son pocas las veces en que este tema ha sido tratado, por consiguiente y teniendo en cuenta el promedio de 4.3

que muestra el Cuadro 18, se asume como positiva la evaluación entendiendo que se están considerando elementos que contribuyen con el buen desempeño.

Cuadro 18. Resultados pregunta número 4

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	7,0	70%
Bueno	3,0	30%
Promedio: 4,3		
Escala: Regular		
Fuente Los Autores		

Figura 23. Resumen resultados pregunta número 4



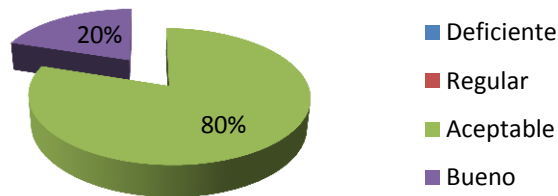
5. ¿Cree usted que la herramienta iBossS controla las tareas mínimas que se deben ejecutar dentro la fase de desarrollo de software?

En el Cuadro 19. Resultados pregunta número 5 y en la Figura 24. Resumen resultados pregunta número 5 se demuestra nuevamente que el 80% califica esta pregunta como regular, pero analizando las observaciones, muchos coinciden en la necesidad de hacer mayor énfasis en las tareas de pruebas y en la posibilidad de que estas sean automatizadas, pero se aclaró que este no es un alcance de la propuesta para esta versión y que cabe dentro de las recomendaciones y trabajos futuros.

Cuadro 19. Resultados pregunta número 5

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	8,0	80%
Bueno	2,0	20%
Promedio: 4,2		
Escala: Regular		
Fuente Los Autores		

Figura 24. Resumen resultados pregunta número 5



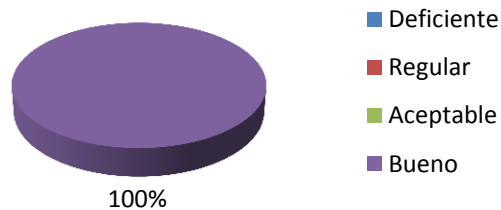
6. ¿Considera usted que iBoss contribuiría con la productividad del desarrollador software?

Todos, el 100% como se muestra en el Cuadro 20 y la Figura 25, coinciden en que iBossS si contribuiría con la productividad del desarrollador y como lo indican las observaciones recopiladas en la pregunta, muchos de los problemas se generan por la negligencia que se tiene frente a la planeación.

Cuadro 20. Resultados pregunta número 6

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	0,0	0%
Bueno	10,0	100%
Promedio: 4,6		
Escala: Bueno		
Fuente Los Autores		

Figura 25. Resumen resultados pregunta número 6



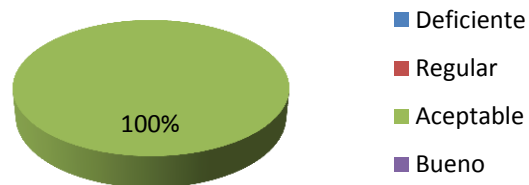
7. ¿Considera usted que iBossS contribuiría con la calidad del producto software?

En esta pregunta, el Cuadro 20 y la Figura 24 muestran que el 100% calificó dentro del rango de regular con un promedio de 4.0, aunque leyendo las observaciones de mejora se comprende que las recomendaciones se hacen sobre la propuesta de inmersión de estándares como ISO o CMMI para validar la calidad, pero estas propuestas estarían fuera de la convicción que dio origen a iBossS. En conclusión si se descartan estas recomendaciones se deduciría que si hay una contribución con la calidad del producto software.

Cuadro 21. Resultados pregunta número 7

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	10,0	100%
Bueno	0,0	0%
Promedio: 4		
Escala: Aceptable		
Fuente Los Autores		

Figura 26. Resumen resultados pregunta número 7



8. ¿Usted está de acuerdo en que los artefactos y herramientas que acompañan el framework iBossS aportan al desarrollo de las tareas y al objetivo de aplicar autogestión en el contexto del desarrollador de software?

Para comprender la pregunta se explicaron los siguientes:

- Carta de Compromiso
- Framework iBossS
- Documento Maestro
- iBossS Plan
- Herramienta iBossS
- Procedimiento de Autogestión
- Artefactos de los procesos de calidad
- Artefactos de los procesos de autogestión

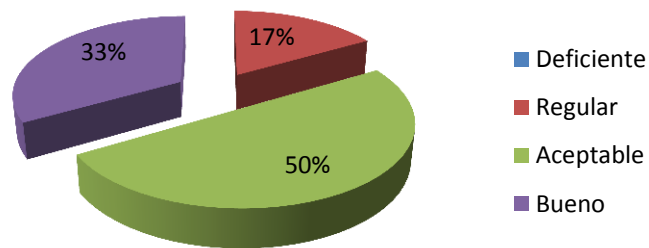
Cada uno de éstos está descrito en el parágrafo 3.2.7.

En el Cuadro 22 y la Figura 27, se observa que el 40% está opinando que iBossS si aporta, el 40% considera que se deben hacer algunos ajustes y hay un 20% que opina que se deben hacer mejoras considerables. El análisis comienza por estos últimos y enfatizan en sus observaciones que sería necesario e importante definir detalladamente los artefactos que se van a manejar dentro de los procesos de calidad, en este caso se aclaró, que estos artefactos dependen del sistema de gestión de la calidad o de los procesos propios de cada empresa, por eso se definen únicamente que tipo de artefactos se pueden manejar. Lo mismo opinan el grupo que pertenecen al 40% con la diferencia de que aconsejan que se deberían establecer algunos artefactos para evitar que su ausencia límite del uso del framework. En conclusión y tomando el promedio de 4.2, se asume que iBossS si aporta con los ítem considerados en la pregunta número ocho y que lo que se necesita es ajustar los artefactos relacionados con la calidad.

Cuadro 22. Resultados pregunta número 8

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	2,0	20%
Aceptable	6,0	60%
Bueno	4,0	40%
Promedio 4,2		
Escala: Aceptable		
Fuente Los Autores		

Figura 27. Resumen resultados pregunta número 8



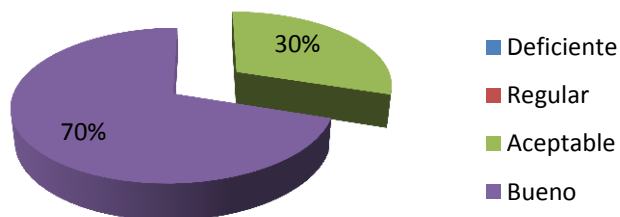
9. En general, ¿Considera usted que el framework iBossS aporta con la mejora del trabajo del Desarrollador de Software?

El 70% que muestraN el Cuadro 23 y la Figura 28 opinan que iBossS si aporta con la mejora del trabajo del desarrollador y el 30% restante alude que se debe buscar suplir o justificar las observaciones hechas en las preguntas anteriores para lograr mejores resultados.

Cuadro 23. Resultados pregunta número 9

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	3,0	30%
Bueno	7,0	70%
Promedio		4,6
Escala: Bueno		
Fuente Los Autores		

Figura 28. Resumen resultados pregunta número 9



10. ¿Usted recomendaría iBossS a una empresa desarrolladora de software como una herramienta que puede contribuir con la mejora de los procesos y el logro de los indicadores de calidad?

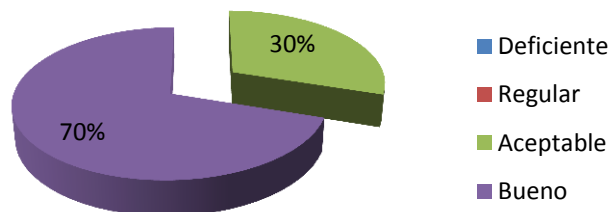
El Cuadro 24 y la Figura 29 muestran que el 70% recomendaría iBossS como una propuesta que puede contribuir con la calidad y la mejora de resultados en el

proceso de desarrollo. El 30%, como en el caso inmediatamente anterior, recomiendan aplicar las observaciones de mejora para lograr un impacto real de iBossS en la calidad y como conclusión, analizando el promedio de 4.6, se podría deducir que iBossS si contribuye el objetivo propuesto.

Cuadro 24. Resultados pregunta número 10

Escala	Resultado	Porcentaje
Deficiente	0,0	0%
Regular	0,0	0%
Aceptable	3,0	30%
Bueno	7,0	70%
Promedio		4,6
Escala: Bueno		
Fuente Los Autores		

Figura 29. Resumen pregunta número 10



4.4 RESUMEN DE LAS OBSERVACIONES

El equipo que realizó la prueba retroalimentó el trabajo mediante las siguientes observaciones:

- Elaborar o concretar un ejemplo que permita ver claramente la forma en la que se puede hacer autogestión por parte de un desarrollador.
- Es necesario que el manifiesto ayude a lograr más conciencia con los procesos de estándares y certificados de la calidad.

- Muy importante, porque como la describen la autogestión no solo aplica para el trabajo, aplica para toda la vida.
- El tema de los distractores es muy importante, por lo menos para el caso de los programadores, porque en la dinámica laboral aparecen muchas cosas, reuniones, cambios de tareas y otras cosas que afectan notablemente el rendimiento.
- Uno de los temas que más tiempo consume son las pruebas. Este framework debería incluir propuestas de automatización de prueba.
- Sería bueno para poder validar los criterios de calidad hablar de la relación entre iBossS, la ISO o CMMI.
- Sería bueno incluir dentro de los artefactos plantillas o modelos por defecto que sirvan para orientar el trabajo. Ejemplo formato de validación de requerimientos.
- Hace falta tiempo para conocer más sobre el framework, pero en este momento es pertinente hacer los ajustes o justificarlos para validar la posible implementación en una empresa.

5. CONCLUSIONES

- Se concluye que el objetivo general se logró satisfactoriamente porque el Framework iBossS permite involucrar las prácticas y los principios de autogestión en el contexto de trabajo del desarrollador de software, teniendo en cuenta los resultados de la pregunta número tres de la prueba de concepto.
- No todas las recomendaciones de autogestión son susceptibles de aplicar explícitamente en la disciplina del desarrollo de software, pero como son parte fundamental de su filosofía, también fueron tomadas como principios y valores a tener en cuenta dentro del Manifiesto de Autogestión para iBossS.
- De las prácticas de autogestión seleccionadas se quieren resaltar tres como relevantes, porque representan las actividades más importantes a desarrollar para cumplir con la intención de iBossS: 1) Gestione adecuadamente el tiempo, 2) Defina su misión y objetivos y consiga los recursos necesarios para trabajar en función de ella y 3) Incorpore prácticas de autogestión en los procesos de negocio de su organización.
- El éxito en la aplicación del framework iBossS depende de la voluntad de cada individuo en crear hábitos personales, los que con el transcurrir del tiempo se harán más fuertes y traerán mejores resultados de calidad al desarrollador, quien al ser más metódico requerirá menos de la asistencia sistematizada de iBossS.
- La componente TimeLog y su registro de las actividades, se convierte en un recurso muy importante para cumplir con las características de gestión del tiempo dentro del framework, pero también es un tema bien delicado, esto es porque un desarrollador que forme parte de una organización, legalmente no tiene derecho a poseer información personal. Es decir que los jefes podrían consultarla, consolidarla y criticarla; esta situación sería un acto molesto que atenta contra la integridad del programador y que podría anular el uso de este tipo de herramientas.
- La utilización de iBossS requiere la participación activa de un experto en temas motivacionales que sensibilice a los programadores sobre las bondades que ofrece el framework y la superación personal que se puede llegar a lograr. Por lo cual se recomienda buscar el apoyo de la gerencia de Recursos humanos para tener una adopción masiva de iBossS en el corto plazo.
- Del trabajo realizado para obtener el capítulo tres, se pudo abstraer un proceso que podría ser utilizado para crear frameworks de autogestión en otras

disciplinas que necesiten tener un software para sistematizar la adopción de la teoría de autogestión. El Proceso tendría los siguientes pasos:

1. **Seleccionar:** Las prácticas generales de la disciplina objetivo.
2. **Relacionar:** Mediante un diagrama cartesiano las prácticas de la disciplina objetivo y las de autogestión.
3. **Tipificar:** A juicio del experto en la disciplina objetivo se establecen los criterios para tipificar las relaciones creadas.
4. **Agrupar:** Separar las prácticas de ambas disciplinas según sean los tipos de relación.
5. **Definir:** Mediante una lectura de cada grupo creado, se podrán definir los requisitos de la herramienta informática a crear.
6. **Validar:** La aplicabilidad de los requerimientos creados y así anular aquellos que no aporten valor.
7. **Construir:** La herramienta de autogestión.

6. TRABAJO FUTURO

- El método para determinar el cumplimiento de un requerimiento se expresa en términos del tiempo transcurrido vs el tiempo planeado, sería útil poder establecer también el cumplimiento de un requerimiento en términos del cumplimiento de los entregables acordados entre el programador y el cliente.
- Una empresa que desarrolla software cuyo nivel de madurez expresado bajo el modelo CMMI sea Inicial, podría considerar el uso de iBossS y proyectar una iniciativa hacia el establecimiento, definición y madurez de sus procesos. Por lo que se requiere establecer un mapeo entre iBossS y el modelo CMMI para establecer y motivar la valoración a la que se podría llegar.
- Técnicamente, iBossS podría evolucionar para programarse en un entorno operable en la nube como DropBox o en una infraestructura con más valor como Office 365, esto aumentaría la usabilidad del framework. Por ejemplo, podría ser de mucha utilidad acceder a los servicios de iBossS desde un celular, tableta o cualquier dispositivo con conexión a internet. Al abrirse hacia la nube, entonces aparecerá el contexto de la autogestión en equipos de desarrollo de software, donde sus miembros podrían mediante el dinamismo y la flexibilidad de iBossS aplicar prácticas de desarrollo Ágiles y/o TSP.
- Se necesita trabajar la autogestión a nivel de equipos por lo que se hace interesante investigar y proponer a fondo prácticas de autogestión a nivel de equipo validadas frente a TSP y las metodologías Ágiles. De ésta manera se podría ampliar el espectro de uso de iBossS desde una perspectiva individual a una grupal.
- Plantear la propuesta de iBossS ante el SMI (Self Management Institute) para buscar el apoyo directo de los expertos, con el objetivo de convertirlo en una herramienta robusta, abierta y gratuita que contribuya con la calidad del software a escala mundial.

BIBLIOGRAFIA

- THE WALL STREET JOURNAL*. (2009). Recuperado el 18 de Enero de 2012, de WORLD BUSINESS FORUM: <http://online.wsj.com/ad/article/wbf-hamel>
- Association for Computing Machinery. (2004). A survey of self-management in dynamic software architecture. *The Open University*, 7.
- Beck, K., & Otros. (2001). *Manifiesto for Agile Software Development*. Recuperado el 6 de 11 de 2012, de Principles behind the Agile Manifiesto: <http://www.agilemanifesto.org/principles.html>
- Department of Health, State of Western Australia. (2011). *WA Chronic Conditions Self-Management Strategic Framework 2011–2015*. Perth: C'wth Australia.
- DRUCKER , P. F. (1999). *Management Challenges for the 21st Century*. PerfectBoun.
- DRUCKER, P. F. (1999). *Managing Oneself*. Harvard Business School Publishing Corporation.
- FEDESOFTE. (24 de enero de 2012). www.fedesoft.org. Recuperado el 29 de enero de 2012, de <http://www.fedesoft.org/novedades/software-sector-pyme-que-promete>
- FERGUSON, P., HUMPHREY, W., KHAJENOORI, S., MACKE, S., & MATVYA, A. (1997). Results of Applying the Personal Software Process. *IEEE Computer*, 24-31.
- GARY, H. (2011). *First Lets Fire all the Managers*. Harvard Business Review.
- GARY, H. (2011). *The big idea: First Lets Fire all the Managers*. Harvard Business Review.
- Gestion Humana.com. (2007). *Autogestión: todos los empleados son líderes*. Bogotá: LEGIS.
- HAYDARLOU, A., OVEY, M., & OVEREINDER, B. (2006). Using Semantic Web Technology for Self-Management of Distributed Object-Oriented Systems. *IEEE Computer Society*.
- HUMPHREY, W. (2000). *The Personal Software Process*. Pittsburgh: Carnegie Mellon University.

- IEEE-CS/ACM. (1999). *Association for Computing Machinery*. Recuperado el 12 de enero de 2012, de Software Engineering Code of Ethics: <http://www.acm.org/about/se-code-s>
- IEEE-CS/ACM. (s.f.). *Association for Computing Machinery*. Recuperado el 12 de enero de 2012, de Software Engineering Code of Ethics: <http://www.acm.org/about/se-code-s>
- Keith, C. (2010). *Agile game development with Scrum*. Boston: Pearson Education.
- LEHIGH UNIVERSITY. (s.f.). *Center for Promoting Research Practice*. Recuperado el 16 de Marzo de 2012, de Self-Management: <http://www.lehigh.edu/projectreach/teachers/self-managemnt/sm-implement.htm>
- LEHTINEN, T., & MÄNTYLÄ, M. (2011). *What Are Problem Causes of Software Projects?* IEEE Computer Society.
- MARSHALL, J. C., & MCHARDY, B. (1999). *Principles of Self-Management: The key to personal an professional success*. Canada.
- PHILIP, J. (1997). *Project LEAP: Lightweight, Empirical, Anti-measurement dysfunction, and Portable Software Developer Improvement*. Department of Information and Computer Science, University of Hawaii.
- PINO, F., GARCÍA, F., & PIATTINI, M. (2007). *Software process improvement in small and medium software enterprises: a systematic review*. UCLM-INDRA Research and Development Institute, University of Castilla-La Mancha: Springer Science+Business Media.
- Roy, P. V. (2006). *Self Management and the Future of Software Design*. 17.
- SEI. (2009). *The Watts New? Collection: Columns by the SEI's Watts Humphrey*. Carnegie Mellon University 2009.
- SEI. (2010). *The Personal Software Process Body of Knowledge*. Pittsburgh: Carnegie Mellon University.
- Self-Management Institute. (2012). *The Self-management Institute*. Recuperado el 12 de Enero de 2012, de <http://self-managementinstitute.org>
- YAMAWAKI, H., MACIARIELLO, J. A., & PEARCE, C. L. (2009). *The Drucker Difference: What the World's Greatest Management Thinker Means to Today's Business Leaders*. McGraw-Hill.

ANEXO 1

1. Manifiesto iBossS

La seriedad inmersa en este tipo de instrumentos se fundamenta en los objetivos que se buscan con su aplicación y porque contienen la declaración de los principios y valores que rigen el comportamiento de las personas que lo asumen, orientando el alcance de los compromisos y de las responsabilidades que se adquieren a nivel personal, profesional y social.

Sobre este lineamiento se deduce que los resultados esperados de la implementación del framework iBossS, están directamente relacionados con la forma de ser de la persona, lo que resalta en primera instancia la necesidad de involucrar dentro de iBossS una figura que le permita al desarrollador conocer y apropiarse de una filosofía de vida que le puede cambiar la forma de ver y de hacer las cosas. Esto es exactamente lo que busca la autogestión, por eso se toma de referente a Drucker, porque el señala que la autogestión “es la oportunidad que tendrá un trabajador de mantenerse comprometido y productivo a través de sus propias habilidades, las cuales le permitirán cultivar una profunda comprensión de sí mismo; no sólo de cuáles son sus fortalezas y debilidades, sino también de cómo se aprende, cómo se trabaja con otros, cuáles son los propios valores y dónde se puede hacer la mayor contribución. Porque sólo cuando se opera a partir de fortalezas se puede alcanzar la verdadera excelencia” (DRUCKER, Managing Oneself, 1999, pág. 2).

CARTA DE COMPROMISO

Este instrumento contiene los principios, valores, expectativas y plan de vida que representan el comportamiento de las personas que utilizarán el Framework de Autogestión iBossS. Por lo tanto, los resultados esperados de su implementación, están directamente relacionados con la forma de ser de la persona. El cumplimiento de los siguientes ítems sugieren la formalización de la carta de compromiso:

- [Manifiesto iBossS](#)
- [Plan de Vida](#)
- [Compromisos Laborales](#)
- [Procedimiento de Autogestión](#)

Manifiesto iBossS

Declaratoria. Todo Desarrollador de Software que haga uso de iBossS se propone proyectar su éxito laboral y personal aplicando los principios de autogestión que contiene el presente manifiesto y a estar sujeto a las normas éticas que su profesión establece.

Principios propios

Describa sus propios principios.

- <Descripción de principio propio #1>
- <Descripción de principio propio #2>
- <Descripción de principio propio #n>

Principios inmutables

- Los individuos no deben usar la fuerza contra otras personas o sus bienes.
- Respetar y mantener los compromisos establecidos con los demás.
- Creer en ti mismo.
- Desarrollar altas cualidades de relaciones, excelente desempeño, comunicación y visión.
- Respetar los principios de autogestión.

Principios generales

- Creer en sus compañeros de trabajo y mantener buenas relaciones personales con ellos.
- Conocer tus propias fortalezas.
- Ser consciente de las características bajo las cuales usted presenta mejor desempeño.
- Auto motive constantemente durante la consecución de sus metas.
- Comprométase a cumplir las actividades que planea llevar a cabo.
- Planifique su propio trabajo.
- Mantenga el enfoque en lo que más importa.
- Identifique su estilo de comunicación (Lector u Oyente)
- Incorpore prácticas de autogestión en los procesos de negocio de su organización.

Plan de Vida

Diligencie el cronograma de su plan de vida o proyecto de vida:

Actividad	Año Estimado de cumplimiento	Cumplido (S/N)
Primera mitad de mi vida		
Actividad a		
Actividad b		
Actividad c		
Segunda mitad de mi vida		
Actividad d		
Actividad e		

Misión

<Describa aquí su misión de vida>

Visión

<Describa aquí su visión de vida>

Motivaciones

<Describa sus motivaciones de vida>

Fortalezas

<Describa aquí sus puntos fuertes>

Debilidades

<Describa aquí sus puntos débiles>

Compromiso laboral 1

Misión Organizacional

<Describe la misión de la organización para la que labora>

Visión Organizacional

<Describe la visión de la organización para la que labora>

Mis expectativas

<Describe sus expectativas en la organización>

A donde quiero llegar

<Describe hacia dónde quiere llegar en la organización para la que labora>

Logros obtenidos

<Describe los logros que alcanzó en la organización>

Compromiso laboral 2

Misión Organizacional

<Describe la misión de la organización para la que labora>

Visión Organizacional

<Describe la visión de la organización para la que labora>

Mis expectativas

<Describe sus expectativas en la organización>

A donde quiero llegar

<Describe hacia dónde quiere llegar en la organización para la que labora>

Logros obtenidos

<Describe los logros que alcanzó en la organización>

ANEXO 2

2. Procedimiento de Autogestión iBossS

1. Control de versiones.

Fecha	Versión	Autor
18-Ago-2012	1.0	Eider Mauricio Aristizábal Ariel Armando Ruiz

2. **Propósito.** Describir las actividades de configuración y autogestión en el desarrollo de software.

3. Referencias.

- MD_Documento Maestro.docx
- iBossS.Plan.V1.0.xlsm
- iBossS.V1.0.xlsm

4. Conceptos.

- **SGC:** Sistema de Gestión de la Calidad
- **iBossS:** Componente de auto gestión que le permite acceder a toda la información disciplinaria del desarrollo de los requerimientos, análisis de la distribución de su tiempo; y registro de tiempos ante el SGC. Es una herramienta de trabajo construida en macros de Excel. Para mayor información vea el documento "**MarcoDeReferencia/Anexos/2.Manual_de_usuario.docx.**"
- **iBossSPlan:** Componente de autogestión que le permite planear y monitorear las actividades de un requerimiento mediante indicadores de gestión básicos como: PV (Valor Planeado), AC (Costo Actual), EV (Valor Devengado) entre otros. Ésta componente se encuentra en la carpeta raíz del área de trabajo de cada requerimiento.

- **Ciclo de Vida de Desarrollo de Sistemas:** Fases requeridas para el desarrollo de software. El SLDC (Software Life Development Cycle) varía según el método de Desarrollo de Software. entre ellos: Software de prototipos, Aplicaciones de diseño mixto (JAD), Desarrollo rápido de aplicaciones (RAD), Extreme Programming (XP), Microsoft Solutions Framework (MSF) etc.
- **Activos Organizacionales:** Documentos plantilla definidas por la organización.

5. Configurar la herramienta iBossS: Tenga en cuenta los siguientes parámetros:

- **Frecuencia:** Una vez o cuando los cambios en el SGC lo requieran.
- **Inducción:** Esta configuración se realiza en la página "Productividad" de iBossS.
 1. Establezca el valor para el parámetro "minutos del día" que define la organización.
 2. Identifique los activos organizacionales correspondientes con el Ciclo de Vida de Desarrollo de Sistemas, entre ellos: Documento de análisis, Documento de diseño, Estándar de construcción, Plantilla de arquitectura etc. Y péguelos en el área de trabajo \iBossS\Artefactos\Ingenieria_SGC.
 3. Identifique y edite las fases y actividades globales del ciclo de vida de desarrollo de sistemas que apliquen en el SGC para el desarrollo de software. El aplicativo iBossS en la página "Productividad" ya cuenta con unas propuestas.
 4. Identifique e inserte las fases que tienen productividad neta para la labor de desarrollo de software por ejemplo en iBossS están Definición, Diseño, Desarrollo, Pruebas unitarias etc.
 5. Identifique e inserte las fases que tienen productividad neta para la empresa pero que no corresponden con la labor del desarrollo de software, Si es necesario controlar la distribución detallada de ese tiempo tipifíquelas en la hoja "Productividad", de lo contrario, utilice la fase "Interrupción" que viene predeterminada en iBossS. El propósito de ésta fase es registrar todo lo que le compete a la organización pero que no está en su lista de requerimientos a desarrollar o no le apoya en el desarrollo de éstos.
 6. Identifique e inserte las fases que no representan labores productivas para la organización ni para el desarrollador. Si es necesario controlar la distribución detallada de ese tiempo tipifíquelas en la hoja "Productividad", de lo contrario simplemente utilice la fase "Otro" que viene predeterminada en iBossS.
 7. Programe la macro para obtener la descripción del requerimiento a partir de su identificador. Buscando el método GetReqDescription en iBossS y

editándolo para que se conecte con la base de datos del aplicativo de gestión de requerimientos de la organización.

8. Programe la macro para registrar tiempos desde iBossS. Esto aplica si la organización para la que usted labora cuenta con una herramienta de registro de tiempos.

6. Configurar la estructura del área de trabajo. Tenga en cuenta los siguientes parámetros:

- **Frecuencia:** Una vez.
- **Inducción:** Esta configuración se realiza en la página principal de iBossS en la sección área de trabajo.
 1. Establezca la ruta raíz del directorio donde se encuentra la carpeta "Disciplina" de la fundamentación teórica de "Autogestión".
 2. Establezca la ruta raíz del directorio donde serán creados los requerimientos nuevos. Ésta ruta puede ser modificada según sea necesario, por ejemplo, para agrupar los requerimientos por año o por mes incluya el año o mes en la ruta. La ruta que se defina será la sugerida por iBossS durante la creación de un nuevo requerimiento y podrá ser cambiada en ese momento.
 3. Pegue los documentos plantilla que la organización utiliza en la carpeta "Disciplina/Plantillas SGC".
 4. Registre el nombre y la extensión de los activos organizacionales del paso anterior en la sección "Plantillas Comunes". Los documentos propios de autogestión están ya incluidos en este apartado

7. Configurar el Documento Maestro. Tenga en cuenta los siguientes parámetros:

- **Frecuencia:** Una vez o durante mejoras al procedimiento de auto gestión
- **Inducción:** El Documento Maestro se configura con las actividades relevantes del Ciclo de Vida de Desarrollo de Software descritas en los procesos y procedimientos organizacionales
 1. Identifique los procedimientos propios del ciclo de vida de desarrollo de sistemas que estén definidos en el Sistema de Gestión de la Calidad.
 2. Léase con el apoyo del responsable del Sistema de Gestión de la Calidad para registrar en el documento maestro las actividades relevantes de tal manera que éstas se puedan marcar como listas de verificación.
 3. Utilice verbos en infinitivo para las entradas de la lista de chequeo.

Clasifique las listas de chequeo según las fases del modelo del ciclo de vida de desarrollo de software.

8. Hoja Seguimiento y control

1. Para hacer un seguimiento de la evolución de un requerimiento, vea la página “Planeación” de iBossS. Allí visualizará en una sola gráfica el consumo de tiempo del requerimiento en función de las fases del SLDC, también visualizará en color rojo, cuando una fase se está saliendo de los tiempos planeados, si esto ocurre, debe informarle a su cliente a menos que pueda adelantar tiempo.
2. En la página “Detalle de Planeación” de iBossS, escriba las actividades a realizar para cumplir con cada fase de desarrollo del requerimiento, haga uso del símbolo \$ seguido de un consecutivo para enumerar las actividades.

Para requerimientos grandes

3. Haga uso del archivo "iBossSPlan" que está en la raíz del área de trabajo del requerimiento.
4. Llene la tabla “**PV - Avance presupuestado (Horas)**” con las horas presupuestadas según cada fecha de corte (Fechas en las que usted espera hacer seguimiento y control) que está en la página “Presupuesto”
5. Asigne el valor del porcentaje de avance percibido, en cada fecha de corte, esto hágalo en la tabla “**Avance percibido (%)**” que está en la página “Ganancia”.
6. Observe la gráfica “Velocidad” que está en la página “Detalle Planeación”, si las curvas están cerca de la línea Óptimo, es porque todo está bien. Si el área velocidad está por debajo de la línea Óptimo entonces hay problemas, de lo contrario es posible que esté presupuestando por encima los tiempos planeados por cada fase del SLDC.

9. Actividades de Ingeniería.

Éstas actividades son dinámicas y se incluyen las siguientes a manera de ejemplo:

1. Llene la lista de chequeo del documento maestro a medida que se avanza en el desarrollo del software.
2. Haga el bosquejo de la tabla de contenido del diseño (en un archivo de texto), organizándolo por componentes de software, para ello puede apoyarse en el diagrama de componentes de la arquitectura.

3. Léase la definición y el alcance del requerimiento de tal manera que extraiga las sentencias y las ubique de manera sencilla en cada componente software.
4. Convierta cada sentencia en la descripción técnica y detallada que instruya la forma en que programará sistema final.

10. Actividades de Autogestión. Tenga en cuenta los siguientes parámetros:

- **Frecuencia:** En el día a día del desarrollador
- **Inducción:** Esto se realiza con todos los productos de trabajo de autogestión (iiBossS, iBossSPlan, Área de trabajo, Documento Maestro y Carta de compromiso) que apoyan el actuar disciplinario del desarrollador que se auto gestiona.
 1. Diligencie el documento carta de compromiso si aún no lo ha realizado.
 2. Antes de abordar un requerimiento, identifique que trabajos de los que se han realizado con anterioridad son similares y revise sus Documentos Maestros esto le dará una idea más clara del nuevo trabajo a realizar en cuanto a la planeación, los recursos, el conocimiento requerido y las características técnicas, etc.
 3. Diligencie el formato Documento Maestro definiendo las expectativas que usted tiene con el requerimiento a abordar. Si lo desea, puede negociar estas expectativas con su jefe.
 4. Planifique su trabajo, en la hoja de planeación simple del iBossS.
 5. Asista con el Documento Maestro a todas las reuniones y refiérase a las personas con su nombre exacto.
 6. Asista con el Documento Maestro físico si lo tiene, puede ser un block de notas o una tableta con los archivos en papel.
 7. Haga el plan del trabajo del día siguiente en la página “TimeLog” de iBossS (Basándose en la planeación simple), puede basarse en la planeación hecha en iBossSPlan (ver manual de usuario de iBossSPlan).
 8. Revise el plan de trabajo del día y visualice vagamente el plan de trabajo del día siguiente, puede basarse en la planeación hecha en iBossSPlan (ver manual de usuario de iBossSPlan).
 9. Al final del día o al inicio del siguiente, reporte los tiempos <si el SGC tiene la herramienta de tiempos>.
 10. Al final del día haga un cierre de autogestión revisando que las fases corresponden con los identificadores de los requerimientos y que el tiempo es adecuado según lo laborado en el día.
 11. Priorice las actividades, simplemente con subir y bajar los accesos de los requerimientos, esto se realiza en la página principal del iBossS.

12. Al inicio del día, revise el plan de trabajo y ejecútelo. Inicie abriendo el Documento Maestro del requerimiento.
13. Revise que notas tiene pendientes en el Documento Maestro y determine si debe gestionarlas.
14. Llene la lista de verificación en el orden descrito dentro del Documento Maestro, esto durante el transcurrir del SDLC.
15. Cuando termine un requerimiento, haga un análisis feedback revisando si las expectativas que se plantearon inicialmente se cumplieron, en este instante usted puede identificar sus puntos fuertes y débiles, para consignarlos en la Carta de Compromiso.
Revise el registro de tiempos realizado junto con la descripción de las actividades para identificar que prácticas relevantes de su profesión puede mejorar, suprimir o incorporar. Por ejemplo, puede encontrar falencias y hacer ajustes al Documento Maestro, esto con el fin de no cometer los mismos errores en el futuro.

No olvide tener en cuenta las siguientes prácticas de autogestión:

1. Gestione adecuadamente su tiempo.
2. Concéntrese en optimizar su rendimiento.
3. Defina su misión y objetivos y consiga los recursos necesarios para trabajar en función de ellos.
4. Planifique su propio trabajo.
5. Comprométase con el logro de los objetivos planeados.
6. Traducir los planes de trabajo en acciones concretas.
7. Asumir la responsabilidad de comunicar.
8. Asuma responsabilidad por las decisiones tomadas.
9. Concentrarse en las oportunidades.
10. Coordine sus actividades con los demás miembros de su equipo de trabajo.
11. Cree y mantenga buenas relaciones personales con sus compañeros de trabajo.
12. Llame a las personas por su nombre.
13. Desarrolle altas cualidades de relaciones, excelente desempeño, comunicación y visión. De ésta manera será visto como un verdadero líder.
14. Haga un análisis Feedback periódicamente teniendo en cuenta sus expectativas y decisiones tomadas a nivel técnico, profesional y personal.
15. Incorpore prácticas de autogestión en los procesos de negocio de su organización.
16. Sea asertivo a la hora de resolver conflictos laborales de tal manera que a futuro éstos no afecten su trabajo.
17. Mantenga el enfoque en lo que más le importa.
18. Realice una sola tarea al tiempo, Si existen varias tareas, priorícelas y resuélvalas en el orden definido.

19. Mantenga la atención en su trabajo, identificando y reduciendo las distracciones.
20. Haga uso de herramientas informáticas para facilitar el uso de las prácticas de autogestión.
21. Todos los principios influyen en el modo de operar del Desarrollador de Software por lo que deben ser interiorizados.

ANEXO 3

3. Documento Maestro - MD

Documentos | Documentos | Documento Maestro - MD

- Asista a todas las reuniones con su MD
- Lívese el MD físico si lo tiene, puede ser una libreta tableta con los archivos en papel
- Al final o inicio del día, si su SGC tiene herramienta para reportar tiempos, hágalo
- Priorice las actividades -usando iBoss-

• REQ_ID : Es el identificador del requerimiento
• CODE NAME: Es el nombre del requerimiento
• Módulo: El área del sistema o componente principal al que pertenece el requerimiento

Notas Importantes

- Nota 1
- Nota 2

<REQ_ID> - CODE NAME

Mis Expectativas

Negocie las expectativas con su jefe y tipifíquelas.

Descripción	Tipo

Causas de desfase

Pretende registrar las razones por la cuales se genera el desfase del requerimiento

Registrar las razones por la cuales se genera el desfase del requerimiento:

#	Fase	Descripción	Causa
1.	<FASE>	Descripción de la causa del desfase	
2.			

Conflictos de integración

Cada vez que un requerimiento tiene algún tipo de relación (Directa, Indirecta, etc.) se deben registrar las expectativas que se tienen de los requerimientos que dependo y las que tienen los requerimientos que dependen de éste. Debe ser una descripción muy breve. Si requiere ahondar, utilice un formato de acta de reunión de su SGC.

Grupo o equipo Involucrado

Descripción del "conflicto"

Grupo o equipo Involucrado

Descripción del "conflicto"

Procedimiento Plantilla

Utilice las siguientes marcas ('X' la actividad es completa '✓' la actividad no aplica '*' Actualmente se está trabajando.

Observación: Las actividades que se listan a continuación, dentro del análisis y descubrimiento del requerimiento, son producto de la experiencia de los autores, pero el desarrollador está en la libertad de colocar las actividades que dicten la metodología que está utilizando o el marco de procesos que sigue el SGC de la organización para la cual trabaja.

Análisis y descubrimiento del requerimiento		
#	Actividad	CHK
1.	Crear carpeta diligenciar la cabecera del Documento Maestro	
2.	Realizar la lectura completa del requerimiento (Solo la lectura).	
3.	Elaborar un StoryBoard (Guión gráfico) en papel o digital (Leyendo nuevamente).	
4.	Elaborar un MER (Solo entidades lógicas de negocio, sin involucrar modelo de datos físico)	
5.	Elaborar un diagrama de relaciones acerca del requerimiento	
6.	Elaborar diagramas (Conceptuales y del Requerimiento)	
7.	Elaborar un diagrama de relaciones del MER (Solo entidades lógicas de negocio, sin involucrar modelo de datos físico)	
8.	Limpiar el requerimiento: <ol style="list-style-type: none"> Cohesión: ¿Hay secciones del requerimiento que deberían estar en otro documento?, si es así, diligenciar la sección de problemas de integración 	
9.	Elaborar una presentación (Prototipo) del requerimiento.	
10.	Hacer la revisión de la presentación	
Diseño Preliminar		
#	Actividad	CHK
11.	Definir el requerimiento en el documento de arquitectura del Sistema de Gestión de la Calidad	
12.	Crear estructura de directorios para alojar artefactos en fase de diseño	
13.	Elaborar la arquitectura 4+1 (Bosquejo)	
14.	En el documento de arquitectura, Detallar lo que en la definición de requerimiento solicita	
15.	Revisión de pares	
16.	Casos de uso	
17.	Pantallas	
Diseño Final		
#	Actividad	CHK
18.	Hacer el esqueleto "Estructura global del documento en fase de diseño"	
19.	Revisión de Casos de Uso	
20.	Arquitectura 4+1 (Afinar)	
21.	Diseño de Pantallas	
22.	Solicitar revisión de diseño a un compañero	
23.	Estimar el tiempo que puede tomar la construcción del trabajo	
24.	Elaborar Casos de prueba	
Construcción		
#	Actividad	CHK
25.	Bloquear unidades de código a utilizar	
26.	Escribir código bajo buenas prácticas <Recomendadas por el sistema de gestión de la calidad>	
27.	Elaborar pruebas de unidad (Con los casos de prueba definidos)	
28.	Elaborar revisión de pares	
29.	Actualizar documento de diseño	

ANEXO 4

4. Manual de Usuario

Manual del usuario iBossS – iBossSPlan

Este manual se incluye en formato digital dentro de la carpeta de la tesis, su tabla de contenidos es la siguiente:

Contenido

Registro de tiempos.....	1
Visualizar el reporte de tiempo por requerimiento.....	2
Estadísticas	2
Planeación.....	2
Planeación Sencilla	2
Detalle de Planeación	4
Seguimiento y Control.....	4
Planear el tiempo.....	4
Seguimiento y Control.....	5
Configuración.....	8
Productividad.....	8
Configuración General	8
Disciplina Root	8
WorkSpace Root	8
Plantillas: Activos Organizacionales.....	8
Área de trabajo.....	9

ANEXO 5

5. Matrices de Comparación

Artefactos y Herramientas Vs. Prácticas de Autogestión

Matriz para Peter Drucker

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
1	Haga un análisis Feedback periódicamente					X	X
1.1	Concentrarse en sus fortalezas		X	X			
1.2	Trabajar en mejorar las fortalezas.	X		X			
1.3	Superar la ignorancia paralizante (Arrogancia)	X		X			
1.4	Corregir los malos hábitos (Que generan pérdidas de eficiencia)		X	X	X	X	
1.5	Practicar los buenos modales al comunicarse con las personas	X	X	X			
1.6	Reconocer que no se puede o no se es capaz de hacer (Humildad)		X	X	X	X	
2	Ser consciente de las características bajo los cuales usted presenta mejor desempeño	X	X			X	X
3	Definir el estilo de comunicación (lector u oyente)	X			X	X	
4	Entender cómo se aprende (lector u oyente)	X			X	X	
5	Definir la capacidad de trabajar en equipo	No aplica para éste trabajo de tesis.					
6	Prever si se está en la capacidad de tomar decisiones o de asesorar						
7	Identificar las características de su desempeño en diferentes entornos de trabajo	X					
8	Identificar cuáles son sus valores personales.		X				
9	Decidir cuál es el lugar a ocupar dentro de la organización		X				
10	Contribuir con el éxito	X					
11	Responsabilizarse de las relaciones interpersonales	X					
12	Gestionar la segunda mitad de su vida		X				

Fuente: Los Autores

Matriz para John C. Marshall y Bob McHarday

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
13	Concéntrese en optimizar su rendimiento, esa es la clave del éxito	x	x	x	x	x	x
14	Las expectativas que usted defina determinan su rendimiento		x		x	x	x
15	El refuerzo en sus actividades crea hábitos			x	x		
16	Auto motive constantemente durante la consecución de sus metas	x	x				
17	Maximizar el rendimiento estableciendo prioridades en sus tareas					x	
18	Gestionar el esfuerzo haciendo seguimiento a sus planes		x			x	x
19	Perfeccionar la autoconfianza	x					
20	Comprométase con el logro de los objetivos planeados		x				
21	Tome el control de su vida gestionando su tiempo					x	x

Fuente: Los Autores

Matriz para Prácticas de Gary Hamel

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
22	Crear una misión personal	x					
23	Elaborar una carta de compromiso	x					
24	Ser innovador sin encajarse en un manual de funciones	x					
25	Fomentar la competencia en pro de los beneficios de la contribución	x					
26	Buscar el éxito pensando primero en la libertad	x					
27	Establezca metas claras y actúe con transparencia en la información	x				x	x

Fuente: Los Autores

Matriz para Jeremy Hunter y J. Scott Scherer

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
28	Minimice los asuntos internos (inestabilidad emocional, erosión moral, pérdida de la atención) que afectan su productividad	x	x			x	
29	Mantener la atención a punto						
29.1	La atención es el fundamento de la autogestión			x		x	
29.2	Atención concentrada: el enfoque es el poder	x	x	x	x	x	x
30	Meditación de la Concentración: Fortalecimiento del CEO interior						
30.1	Neuroplasticidad: Recableando la Red						
30.2	El aprendizaje continuo	x		x	x	x	
30.3	Ser consciente de las emociones reactivas	x					

Fuente: Los Autores

Matriz para Doug Kirkpatrick

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
31	Los individuos no deben usar la fuerza contra otras personas o sus bienes	x					
32	Las personas deben respetar y cumplir los compromisos realizados	x					

Fuente: Los Autores

Matriz para LoriKane Handout

Práctica No.	Práctica / Principio de Autogestión	Framework iBossS					
		Artefacto				Herramienta	
		Manifiesto iBossS	Carta de compromiso	Procedimiento iBossS	Documento Maestro	iBossS	iBossS.Plan
33	Mantenga el foco en lo más importante		x		x		
34	Crea en los otros miembros del equipo	x					

Fuente: Los Autores