



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**MODELO DE DINÁMICA DE SISTEMAS PARA LA MEJORA DE
LA PLANIFICACIÓN Y DESARROLLO DE PROYECTOS DE
SOFTWARE QUE APLICAN EL MARCO DE TRABAJO SCRUM
EN LA REGIÓN DE PUNO**

TESIS

PRESENTADA POR:

VERARDO GRETA ROQUE

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS

PUNO – PERÚ

2019



NOMBRE DEL TRABAJO

MODELO DE DINÁMICA DE SISTEMAS PARA LA MEJORA DE LA PLANIFICACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE QUE APLICAN EL MARCO DE TRABAJO SCRUM EN LA REGIÓN DE PUNO

AUTOR

VERARDO GRETA ROQUE

RECuento de palabras

36634 Words

RECuento de caracteres

201773 Characters

RECuento de páginas

207 Pages

Tamaño del archivo

2.6MB

Fecha de entrega

Aug 18, 2024 7:41 PM GMT-5

Fecha del informe

Aug 18, 2024 7:44 PM GMT-5

● 10% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 10% Base de datos de Internet
- Base de datos de Crossref
- 3% Base de datos de trabajos entregados
- 1% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 10 palabras)

V°B°



Firmado digitalmente por
INGALUQUE ARAPA Marga Isabel
FAU 20145498170 hard
Motivo: Soy el autor del documento
Fecha: 18.08.2024 19:46:32 -05:00

Firmado digitalmente por
SOTOMAYOR ALZAMORA Guina
Guadalupe FAU 20145498170 hard
Motivo: Doy V° B°
Fecha: 19.08.2024 19:22:30 -05:00



DEDICATORIA

*Dedico este trabajo a mi madre **Gladys Roque**, a mi padre **Epifanio Greta** y a mis hermanos, a cada uno de ellos por su amor, apoyo y aliento en todo momento. Gracias por ser mi fuente de inspiración y por enseñarme a perseguir mis sueños con pasión y determinación. Este logro no habría sido posible sin su constante apoyo y confianza en mí.*

Verardo Greta Roque



AGRADECIMIENTOS

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera en la realización de esta tesis.

En primer lugar, A la Universidad Nacional del Altiplano, Facultad de Ingeniería Mecánica Eléctrica, Electrónica y Sistemas, Escuela Profesional de Ingeniería de Sistemas por su formación profesional. Agradecer a mi director/asesor de tesis, M.SC. Marga Isabel Ingaluque Arapa, por su orientación, paciencia y apoyo constante a lo largo de todo el proceso de investigación. Sus valiosas sugerencias y comentarios han sido fundamentales para lograr los objetivos propuestos.



ÍNDICE GENERAL

	Pág.
DEDICATORIA	
AGRADECIMIENTOS	
ÍNDICE GENERAL	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
ÍNDICE DE ANEXOS	
ACRÓNIMOS	
RESUMEN	17
ABSTRACT.....	18
CAPÍTULO I	
INTRODUCCIÓN	
1.1. DESCRIPCIÓN DEL PROBLEMA	19
1.2. DEFINICIÓN DEL PROBLEMA	20
1.2.1. Problemas específicos.....	20
1.3. IMPORTANCIA Y UTILIDAD DEL ESTUDIO	20
1.4. OBJETIVOS DE LA INVESTIGACIÓN.....	21
1.4.1. Objetivo general.....	21
1.4.2. Objetivos específicos	21
CAPÍTULO II	
REVISIÓN DE LITERATURA	
2.1. ANTECEDENTES DE LA INVESTIGACIÓN.....	22
2.2. BASE TEÓRICA.....	29



2.2.1. Sistema.....	29
2.2.2. Modelo	31
2.2.3. Simulación	32
2.2.3.1. Clasificación de la simulación	33
2.2.3.2. Metodología de Simulación	37
2.2.4. Dinámica de Sistemas	40
2.2.5. Metodología Dinámica de Sistemas	44
2.2.5.1. Fases de la Metodología Dinámica de Sistemas	45
2.2.5.2. Utilizando la Metodología de Dinámica de Sistemas	47
2.2.6. Proyecto	49
2.2.6.1. Fases de un Proyecto.....	50
2.2.6.2. Gestión de Riesgos.....	52
2.2.7. Métodos Ágiles	53
2.2.8. Marco de Trabajo Scrum	57
2.2.8.1 Roles.....	59
2.2.8.2 Bloques de Tiempo	62
2.2.8.3 Artefactos Principales de Scrum.....	65
2.2.8.4 Ciclo de Vida	67
2.3. MARCO CONCEPTUAL	69
2.4. HIPÓTESIS Y VARIABLES.....	73
2.4.1. Hipótesis General.....	73
2.4.2. Operacionalización de Variables	73
CAPÍTULO III	
MATERIALES Y MÉTODOS	
3.1. METODOLOGÍA DE LA INVESTIGACIÓN	75



3.2.	DISEÑO DE LA INVESTIGACIÓN	75
3.3.	POBLACIÓN Y MUESTRA.....	76
	3.3.1. Población	76
	3.3.2. Muestra	77
3.4.	MÉTODO DE RECOPIACIÓN DE DATOS.....	77
	3.4.1. Encuesta.....	77
3.5.	MÉTODO DE TRATAMIENTO DE DATOS.....	78
3.6.	MATERIAL EXPERIMENTAL.....	78
3.7.	DISEÑO ESTADÍSTICO PARA LA PRUEBA DE HIPÓTESIS.....	79

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1.	FASE DE CONCEPTUALIZACIÓN	81
	4.1.1. Diagrama de Influencia del Sub Modelo de Planificación	83
	4.1.2. Diagrama de Influencia del Sub Modelo de Puntos en la Producción.....	84
	4.1.3. Diagrama de Influencia del Sub Modelo de Ejecución de Tareas	85
	4.1.4. Diagrama de Influencia del Sub Modelo de Pruebas de Tareas	86
	4.1.5. Diagrama de Influencia del Sub Modelo de Presión del Equipo	88
	4.1.6. Diagrama de Influencia del Sub Modelo de Tareas Adicionales.....	89
	4.1.7. Diagrama de Influencia del Sub Modelo de Pruebas de Integración.....	90
	4.1.8. Diagrama de Influencia del Sub Modelo de Tiempo de Trabajo.....	91
	4.1.9. Diagrama de Influencia del Sub Modelo de Cansancio.....	92
	4.1.10. Diagrama de Influencia del Sub Modelo de Incremento de Experiencia de Equipo	93
4.2.	FASE DE FORMULACIÓN.....	94
	4.2.1. Sub Modelo de Planificación.....	95



4.2.1.1. Clasificación de Variables	95
4.2.1.2. Diagrama de Forrester.....	98
4.2.1.3. Ecuaciones del Sub Modelo.....	98
4.2.2. Sub Modelo de Puntos en la Producción	100
4.2.2.1. Clasificación de Variables	100
4.2.2.2. Diagrama de Forrester.....	102
4.2.2.3. Ecuaciones del Sub Modelo.....	102
4.2.3. Sub Modelo de Ejecución de Tareas.....	104
4.2.3.1. Clasificación de Variables	104
4.2.3.2. Diagrama de Forrester.....	107
4.2.3.3. Ecuaciones del Sub Modelo.....	107
4.2.4. Sub Modelo de Pruebas de Tareas	111
4.2.4.1. Clasificación de Variables	111
4.2.4.2. Diagrama de Forrester.....	113
4.2.4.3. Ecuaciones del Sub Modelo.....	113
4.2.5. Sub Modelo de Presión en el Equipo.....	115
4.2.5.1. Clasificación de Variables	115
4.2.5.2. Diagrama de Forrester.....	117
4.2.5.3. Ecuaciones del Sub Modelo.....	117
4.2.6. Sub Modelo de Tareas Adicionales	118
4.2.6.1. Clasificación de Variables	118
4.2.6.2. Diagrama de Forrester.....	120
4.2.6.3. Ecuaciones del Sub Modelo.....	120
4.2.7. Sub Modelo de Pruebas de Integración	121
4.2.7.1. Clasificación de Variables	121



4.2.7.2. Diagrama de Forrester.....	123
4.2.7.3. Ecuaciones del Sub Modelo.....	123
4.2.8. Sub Modelo de Tiempo de Trabajo	125
4.2.8.1. Clasificación de Variables	125
4.2.8.2. Diagrama de Forrester.....	127
4.2.8.3. Ecuaciones del Sub Modelo.....	127
4.2.9. Sub Modelo de Cansancio	129
4.2.9.1. Clasificación de Variables	129
4.2.9.2. Diagrama de Forrester.....	131
4.2.9.3. Ecuaciones del Sub Modelo.....	131
4.2.10. Sub Modelo de Incremento de Experiencia de Equipo.....	132
4.2.10.1. Clasificación de Variables	132
4.2.10.2. Diagrama de Forrester.....	134
4.2.10.3. Ecuaciones del Sub Modelo.....	135
4.3. FASE DE EVALUACIÓN.....	137
4.3.1. Validación Estructural	137
4.3.1.1. Verificación de la Estructura.....	137
4.3.1.2. Consistencia Dimensional.....	139
4.3.1.3. Pruebas de Condiciones Extremas	141
4.3.2. Validación Estadística.....	168
4.3.2.1. Modelado 1: “SISTEMA DE PLANIFICACIÓN DE RECURSOS”	168
4.3.2.2. Modelado 2: “APLICACIÓN MULTIPLATAFORMA PANDA COMPUTER”	174



4.3.2.3. Modelado 3: “APLICATIVO WEB ALAPAMAYO CLIENT.”	180
4.3.2.4. Análisis de Validación Estadística	187
4.4. PRUEBA DE HIPÓTESIS DE LA INVESTIGACIÓN	189
4.4.1. Planteamiento de la Hipótesis	189
4.4.2. Nivel de Significancia	189
4.4.3. Interpretación de Resultados	193
4.5. DISCUSIÓN	195
V. CONCLUSIONES	197
VI. RECOMENDACIONES	200
VII. REFERENCIAS BIBLIOGRÁFICAS	201



ÍNDICE DE TABLAS

	Pág.
Tabla 1 Elementos del Diagrama de Forrester de Vensim.	48
Tabla 2 Los Principios de los Métodos Ágiles.	55
Tabla 3 Operacionalización de Variables	74
Tabla 4 Análisis Estadístico de Datos Reales y Datos Simulados.....	188
Tabla 5 Significado de las Alternativas	190
Tabla 6 Resultados de la Encuesta Pre-Test y Post-Test	190
Tabla 7 Resultados de Medias Prueba T-Student.	192
Tabla 8 Resultados de la Prueba T-Student.	192



ÍNDICE DE FIGURAS

	Pág.
Figura 1 Modelo general de un sistema.....	29
Figura 2 Proceso de modelado.....	33
Figura 3 Resumen de los tipos de Simulación.....	36
Figura 4 Diagrama Influencia Básico	41
Figura 5 Estructura de realimentación negativa	42
Figura 6 Estructura de realimentación Positiva	44
Figura 8 Marco de Trabajo Scrum.....	59
Figura 9 Fases Scrum.....	69
Figura 10 Diseño del Modelo y sus Diez Sub Modelos	82
Figura 11 Diagrama de Influencia del Sub Modelo de Planificación.....	84
Figura 12 Diagrama de Influencia del Sub Modelo de Puntos en la Producción	85
Figura 13 Diagrama de Influencia del Sub Modelo de Ejecución de Tareas	86
Figura 14 Diagrama de Influencia del Sub Modelo de Pruebas de Tareas.....	88
Figura 15 Diagrama de Influencia del Sub Modelo de Presión del Equipo	89
Figura 16 Diagrama de Influencia del Sub Modelo de Tareas Adicionales	90
Figura 17 Diagrama de Influencia del Sub Modelo de Pruebas de Integración	91
Figura 18 Diagrama de Influencia del Sub Modelo de Tiempo de Trabajo	92
Figura 19 Diagrama de Influencia del Sub Modelo de Cansancio	93
Figura 20 Diagrama de Influencia del Sub Modelo de Incremento de Experiencia de Equipo.....	94
Figura 21 Diagrama de Forrester del Sub Modelo de Planificación	98
Figura 22 Diagrama de Forrester del Sub Modelo de Puntos en la Producción.....	102
Figura 23 Diagrama de Forrester del Sub Modelo de Ejecución de Tareas	107
Figura 24 Diagrama de Forrester del Sub Modelo de Pruebas de Tareas	113
Figura 25 Diagrama de Forrester del Sub Modelo de Presión en el Plazo	117
Figura 26 Diagrama de Forrester del Sub Modelo de Tareas Adicionales.....	120
Figura 27 Diagrama de Forrester del Sub Modelo de Pruebas de Integración	123
Figura 28 Diagrama de Forrester del Sub Modelo de Tiempo de Trabajo.....	127
Figura 29 Diagrama de Forrester del Sub Modelo de Cansancio	131



Figura 30 Diagrama de Forrester del Sub Modelo de Incremento de Experiencia de Equipo	134
Figura 31 Funciones en el software Vensim (Check Model y Units Check) para revisar sintaxis del modelo	139
Figura 32 Estado de la ecuación cantidadTareasPorPuntos	141
Figura 33 ModeladoEscenario1 Variables BurndownCharts y Puntos desarrollar ...	144
Figura 34 ModeladoEscenario1 Variable Cansancio	145
Figura 35 ModeladoEscenario1 Variables TAREASPLANIFICADAS y TAREASCODIFICADAS	146
Figura 36 ModeladoEscenario1 Variables PRUEBASREALIZAR y PruebasDiseñadas	147
Figura 37 ModeladoEscenario1 Variable PRUEBASINERROR	148
Figura 38 ModeladoEscenario1 Variables PRUEBASCONERRDETECTADOS ...	149
Figura 39 ModeladoEscenario1 Variables PRUEBASCONERRINTEGRACION y PRUEBASINTEGRACIONREALIZADAS	149
Figura 40 ModeladoEscenario2 Variables BurndownCharts y Puntos desarrollar ...	153
Figura 41 ModeladoEscenario2 Variable Cansancio	153
Figura 42 ModeladoEscenario2 Experiencia del Equipo de Trabajo	154
Figura 43 ModeladoEscenario2 Variable PresionPlazo.	155
Figura 44 ModeladoEscenario2 Variables TAREASPLANIFICADAS y TAREASCODIFICADAS	156
Figura 45 ModeladoEscenario2 Variables PRUEBASCONERRDETECTADOS ...	157
Figura 46 ModeladoEscenario2 Variables PRUEBASINERROR	158
Figura 47 ModeladoEscenario2 Variables PRUEBASCONERRINTEGRACION y PRUEBASINTEGRADAS	158
Figura 48 ModeladoEscenario3 Variables BurndownCharts y Puntos desarrollar ...	161
Figura 49 ModeladoEscenario3 Variable CANSANCIO	162
Figura 50 ModeladoEscenario3 Variable TAREASEXTRASDESARROLLADAS	163
Figura 51 ModeladoEscenario3 Variables TAREASCODIFICADAS y TAREASPLANIFICADAS	164
Figura 52 ModeladoEscenario3 Variables PRUEBASCONERRDETECTADOS ...	165
Figura 53 ModeladoEscenario3 Variables PRUEBASINERROR	165



Figura 54	ModeladoEscenario3 Variables PRUEBASCONERRRDEINTEGRACION.	166
Figura 55	ModeladoEscenario3 Variables PRUEBASINTEGRADAS	167
Figura 56	ValidacionModelado1 Variable Momento Inicio Sprints	169
Figura 57	ValidacionModelado1 Variable Puntos Según Sprints	170
Figura 58	ValidacionModelado1 Momentos de Inicio y Puntos por Sprints	171
Figura 59	ValidacionModelado1 Variable BURNDOWN CHARTS	172
Figura 60	ValidacionModelado1 Puntos y Tareas por Sprints	173
Figura 61	ValidacionModelado1 Pruebas con Errores y Pruebas sin Error	173
Figura 62	ValidacionModelado2 Variable Momento Inicio Sprints	175
Figura 63	ValidacionModelado2 Variable Puntos Según Sprints	176
Figura 64	ValidacionModelado2 Momentos de Inicio y Puntos por Sprints	177
Figura 65	ValidacionModelado2 Variable BURNDOWN CHARTS	178
Figura 66	ValidacionModelado2 Puntos y Tareas por Sprints	178
Figura 67	ValidacionModelado2 Pruebas con Errores Detectados y Pruebas sin Error	179
Figura 68	ValidacionModelado2 Prueba de Integración	180
Figura 69	ValidacionModelado1 Variable Momento Inicio Sprints	182
Figura 70	ValidacionModelado3 Variable Puntos Según Sprints	183
Figura 71	ValidacionModelado3 Momentos de Inicio y Puntos por Sprints	183
Figura 72	ValidacionModelado3 Variable BURNDOWN CHARTS	184
Figura 73	ValidacionModelado3 Puntos y Tareas por Sprints	185
Figura 74	ValidacionModelado3 Pruebas con Errores Detectados y Pruebas sin Error	186
Figura 75	ValidacionModelado3 Prueba de Integración	186
Figura 76	Resultados Pre y Post de la Planificación	191
Figura 77	Resultados Pre y Post del Desarrollo	191
Figura 78	Región de Aceptación Hipótesis.	193



ÍNDICE DE ANEXOS

	Pág.
ANEXO 1 Encuesta para la prueba de hipótesis	202
ANEXO 2 Declaración jurada de autenticidad de tesis	202
ANEXO 3 Autorización para el depósito de tesis en el Repositorio Institucional.....	202



ACRÓNIMOS

- SD: System Dynamics (Dinámica de Sistemas).
- ASD: Adaptative Software Development (Desarrollo de Software Adaptativo)
- FDD: Feature driven development (Desarrollo Basado en Funcionalidades)
- DSDM: Dynamic System Development Method (Método de Desarrollo de
Sistemas Dinámicos)



RESUMEN

En el presente trabajo de investigación, se plantea presentar un modelo de dinámica de sistemas, para proyectos de software que aplican el marco de trabajo Scrum en la región de Puno, con el objetivo general de Determinar que el Modelo de Dinámica de Sistemas mejora la planificación y desarrollo de proyectos de software que aplican el marco de trabajo mencionado. Para el desarrollo del modelo, se inició con la elaboración del diseño y construcción de los diferentes Diagramas de Influencia, obteniendo diez Sub Modelos, tales como: Sub Modelo de Planificación, Sub Modelo de Ejecución de Tareas, Sub Modelo de Ejecución de Tareas Adicionales, Sub Modelo de Pruebas de Integración, entre otras; los Diagramas de Forrester y el posterior modelado de los mismos fueron trabajados con Vensim PLE, que fue aplicado en los tres proyectos tomados como población. La investigación siguió el diseño Cuasi – Experimental, donde se demostró estadísticamente la hipótesis planteada, los resultados a partir de la prueba realizada demostraron un claro alto nivel de aceptación, finalmente se concluye que el Modelo de Dinámica de Sistemas en los proyectos de software que aplican el marco de trabajo Scrum, mejora la planificación y desarrollo de dichos proyectos en la región de Puno.

Palabras Clave: Dinámica de Sistemas, Modelo, Marco de Trabajo Scrum, Software.



ABSTRACT

In this research work, it is proposed to present a system dynamics model, for software projects that apply the Scrum framework in the Puno region. With the general objective of Determining that the System Dynamics Model improves the planning and development of software projects that apply the mentioned framework. For the development of the model, it began with the design and construction of the different Influence Diagrams, obtaining ten Sub Models, such as: Sub Planning Model, Sub Model of Execution of Tasks, Sub Model of Execution of Additional Tasks, Sub Model of Integration Tests, among others; Forrester Diagrams and their subsequent modeling were worked with Vensim PLE, which was applied in the three projects taken as a population. The research followed the Quasi - Experimental design, where the hypothesis was statistically demonstrated, the results from the test performed showed a clear high level of acceptance, it is finally concluded that the System Dynamics Model in the software projects that apply The Scrum framework improves the planning and development of these projects in the Puno region.

Keywords: Systems Dynamics, Model, Software, Scrum Framework.



CAPÍTULO I

INTRODUCCIÓN

1.1. DESCRIPCIÓN DEL PROBLEMA

En el ámbito de desarrollo de software en el cual aplican el marco de trabajo Scrum, tanto el lado de desarrolladores como clientes no son ajenos a los múltiples problemas que se generan en el transcurso de este tipo de proyectos; los sistemas actuales y su complejidad, los cambios repentinos en el ámbito de estos sistemas y diferentes modificaciones hechas en los requerimientos por parte de los clientes, y esto en pleno desarrollo del proyecto de software, generan que la planificación, desarrollo, administración y control resulten difíciles de estimar o evaluar.

Por lo tanto, un proyecto determinado y más aún en el ámbito del desarrollo de software, al tener varios factores para su correcta elaboración, tienden a desequilibrar los recursos y el mismo software como proyecto.

Una vez explicado todo lo anterior, queda claro que la simulación de dinámica de sistemas se aplica en diversos campos a la hora de elegir soluciones. Las ciencias empresariales, medioambientales y sociales son los campos en los que más se utiliza la simulación de dinámica de sistemas. La dinámica de sistemas es una herramienta muy útil para analizar los problemas que surgen en un mundo complejo y en constante cambio. Dado que los requisitos y los escenarios cambian con frecuencia, disponer de una herramienta que permita simular diversos escenarios y extraer distintas proyecciones con la dinámica del sistema sobre la gestión de los proyectos de desarrollo de software implementados con Scrum es, por tanto, una alternativa muy atractiva para los gestores y los equipos de trabajo. Esto se debe principalmente a que les permite evaluar cómo sus



decisiones afectan al proceso de desarrollo del proyecto sin afectar o poner en peligro el proyecto real o sus recursos.

1.2. DEFINICIÓN DEL PROBLEMA

¿El modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum, mejorará significativamente la planificación y desarrollo de dichos proyectos en la región de Puno?

1.2.1. Problemas específicos

- a) ¿En qué medida el uso del modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum en la región de Puno mejorará la planificación y desarrollo?
- b) ¿En qué medida se determinará el nivel de mejora de la planificación y desarrollo en los proyectos de software que aplican el marco de trabajo Scrum con el modelo de dinámica de sistemas?

1.3. IMPORTANCIA Y UTILIDAD DEL ESTUDIO

La importancia del presente proyecto de investigación radica en la obtención de información de tantos escenarios resultantes como uno requiera simular de un determinado proyecto de software que aplica el marco de trabajo Scrum. Ya que el modelo de dinámica de sistemas nos permite simular escenarios hipotéticos sin poner en riesgo un determinado proyecto real, aprovechando esta información resultante como uno de los puntos de referencia para la toma de decisiones dentro del proyecto.

Un aporte importante también se da en el potencial profesional de los desarrolladores de software, ya que el modelo conlleva al incremento y aprendizaje de experiencias y habilidades, apoyándose de los resultados de los diferentes escenarios simulados.



1.4. OBJETIVOS DE LA INVESTIGACIÓN

1.4.1. Objetivo general

Determinar que el modelo de dinámica de sistemas mejora la planificación y desarrollo de proyectos de software que aplican el marco de trabajo Scrum en la región de Puno.

1.4.2. Objetivos específicos

- Conceptualizar y Formular el modelo de dinámica de sistemas para proyectos de desarrollo de software que en su elaboración hacen uso del marco de trabajo Scrum.
- Evaluar y lograr la mejora de la planificación y desarrollo de proyectos de software que aplican el marco de trabajo Scrum con la influencia del modelo dinámico en la región de Puno.



CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Las siguientes investigaciones son las más relevantes que se consideraron como antecedentes, puesto que trabajan en las mismas líneas de tecnología y tratan conceptos altamente importantes enfocados a la misma dirección que nuestro actual proyecto.

- ❖ Cocco, Mannaro, Concas, & Marchesi, (2011), en su paper titulada “SIMULATING KANBAN AND SCRUM VS. WATERFALL WITH SYSTEM DYNAMICS”, desde la Universidad de Cagliari Italia; llegan a las siguientes conclusiones:

1. Este documento contiene un caso de validación y un experimento que utiliza un «Modelo de Simulación Dinámica para Proyectos de Software utilizando Scrum.» Los Scrum Masters y el equipo pueden utilizar el modelo para analizar cómo el uso de la metodología Scrum, en conjunto con los Bloques de Tiempo, Artefactos y Reglas, ha impactado su gestión bajo diferentes condiciones.
2. El modelo ofrece la posibilidad de modificar los valores de los parámetros tanto antes como durante la simulación. Se pueden establecer muchos parámetros antes del comienzo de cada simulación, incluyendo la duración y la velocidad de cada Sprint, el ritmo estimado de desarrollo de las tareas, los factores de fatiga, las limitaciones de tiempo, el número de miembros del equipo en función de su experiencia con la metodología, y cualquier tarea adicional que se prevea. El usuario puede cambiar el número de miembros del equipo que empiezan, clasificar al equipo como Juniors o Expertos en función de su experiencia con Scrum, y modificar la cantidad de horas programadas



modificando los valores de los parámetros mientras el proyecto está en marcha.

3. Por lo tanto, tras su validación y comprobación, se ha constatado que el modelo puede utilizarse como herramienta para evaluar los efectos de distintas medidas de gestión.

❖ Godoy, Belloni, Sosa, & Kotynski, (2014), en el paper titulada “EVALUACIÓN DE ALTERNATIVAS DE GESTIÓN EN PROYECTOS DE SOFTWARE DESARROLLADOS CON SCRUM UTILIZANDO DINÁMICA DE SISTEMAS”, en el Centro de Investigación en Tecnología de la Información y Comunicaciones de la Universidad Gastón Dachary Argentina; muestran las siguientes conclusiones:

1. En el presente trabajo se ha presentado un caso de validación y un experimento realizado con un “Modelo Dinámico de Simulación para Proyectos de Software que utilizan Scrum”. El modelo permite a los Scrum Masters y el Team analizar el efecto del uso conjunto de la metodología Scrum, Bloques de Tiempo, Artefactos y Reglas en la gestión de los mismos en diferentes escenarios.
2. El modelo ofrece flexibilidad de modificar los valores de los parámetros tanto previamente al inicio de la simulación como al momento de la ejecución de la misma. Dentro de los parámetros que se pueden establecer previo al inicio de cada simulación se encuentran: la duración y la velocidad de cada Sprint, la velocidad estimada de desarrollo de las tareas, Factores de Cansancio, de Presión en el plazo, Cantidad de integrantes del Team según su experiencia en la metodología y las tareas extras que se prevén puedan surgir. A través de la modificación de valores de los parámetros durante su ejecución el usuario



puede establecer o modificar la cantidad de integrantes del Team que abandonan el proyecto, clasificar al Team mediante la asociación de estos a su experiencia en Scrum en Juniors o Expertos, cambiar la cantidad de horas estimadas de duración del proyecto, generar horas extras e inasistencia de los integrantes de manera determinística o pseudoaleatoria, entre otros.

3. De esta forma, luego de validar y experimentar con el modelo, se ha llegado a la conclusión de que el mismo puede ser utilizado como herramienta para evaluar el impacto de diferentes decisiones de gestión como la aparición tareas extras.

❖ Ciurariu & Risu, (2016), en su artículo titulada como: “USAGE OF SCRUM METHODOLOGY FOR IMPLEMENTING SOFTWARE PROJECT”.

Presentada en Rumania; las principales conclusiones descritas son las siguientes:

1. A lo largo de la sección Resultados, se presentan partes débiles de esta metodología y soluciones que podrían mejorar el uso de Scrum. Se muestra la necesidad de incluir las partes de análisis y diseño dentro del proceso, para tener especificaciones al menos para el caso en que sea necesario incluir a otras personas en el equipo, para ajustar la forma en que se organizan las reuniones, porque generalmente el equipo Contiene especialistas con diferentes experiencias y capacidad para comprender de forma ágil el negocio de las aplicaciones.
2. El método Scrum se puede usar en una variedad de proyectos de software, pero sin una experiencia en el uso de este método, todas las ventajas descritas por este método pueden resultar en desventaja. Un porcentaje muy bajo de aplicaciones que usan Scrum, respeta los principios definidos por el manifiesto ágil. Entre los proyectos que pueden permitirse el lujo de usar Scrum puro



están las nuevas empresas y las pequeñas aplicaciones donde la rapidez para lanzar periódicamente nuevas versiones es muy importante. Es obvio que este método con algunos ajustes se puede utilizar en un área más grande de proyectos de software.

3. La novedad está representada por el análisis crítico y por las soluciones propuestas para mejorar el uso de este método en la gestión de proyectos de software.

❖ White, (2014), en su artículo titulada “AN AGILE PROJECT SYSTEM DYNAMICS SIMULATION MODEL”, desarrollada en el School of Science and Technology, Middlesex University, London; las conclusiones son las siguientes:

1. Se creó un nuevo modelo de proyecto ágil SIMULINK TM mínimo y se comparó con un modelo de cascada de un proyecto de la NASA. Los resultados se presentan para permitir a los gerentes de proyectos predecir el desempeño de futuros procesos ágiles de proyectos.
2. El modelo incluye el tiempo para generar los requisitos y una función para proporcionar la volatilidad de los requisitos. Estos modelos muestran que para la misma productividad y fracción de errores que son satisfactorios, los costos generales del personal de desarrollo son similares para los proyectos ágiles y en cascada y el reproceso no descubierto es menor para el proyecto ágil.
3. Este modelo generalmente admite observaciones empíricas de que las iteraciones más cortas de la caja de tiempo producen una finalización general del proyecto más corta.

❖ Cao, Ramesh, & Abdel-Hamid, (2010), en su paper titulada “MODELING DYNAMICS IN AGILE SOFTWARE DEVELOPMENT”, describe las siguientes conclusiones principales:



1. Un experimento con el modelo sugiere que la refactorización afecta el costo de realizar cambios en el sistema. El costo cambia cíclicamente y aumenta con el tiempo. Este patrón es el resultado del comportamiento de refactorización que es crítico para el éxito del desarrollo ágil. Además, en un proyecto con un cronograma fijo, los retrasos con una refactorización importante conducen a un mayor costo del proyecto y menos historias entregadas.
2. Se estudiaron los impactos de la programación de pares en la productividad, retrabajo, refactorización y pruebas unitarias. Los resultados muestran que, sin programación de pares, se entregan menos tareas que con la programación de pares. El costo de cada tarea entregada y el costo de retrabajo y cambio sin programación de pares son mucho más altos que con la programación de pares. Con la programación de pares, la necesidad de una refactorización mayor es menor y el costo total de la refactorización se reduce.
3. El modelo puede ayudar a los profesionales a mejorar la práctica de desarrollo de software al facilitar la comprensión de la dinámica ágil del proyecto. Primero, el modelo puede usarse para investigar los impactos de prácticas específicas como el nivel de participación del cliente en el desempeño del proyecto. En segundo lugar, el modelo se puede utilizar para diseñar y analizar políticas de gestión de proyectos. Por ejemplo, la asignación de esfuerzo entre refactorización y desarrollo se puede explorar para encontrar el nivel óptimo de esfuerzo de refactorización. Tercero, el modelo puede usarse para examinar la sensibilidad del proceso de desarrollo ágil a una variedad de factores internos y externos. Por ejemplo, la sensibilidad del proceso ágil al tamaño del equipo, la duración de la iteración y la volatilidad de los requisitos se pueden examinar utilizando el modelo. En resumen, esta investigación proporciona a



los investigadores una herramienta para estudiar el desarrollo ágil y ayuda a los profesionales a tomar mejores decisiones y formular procesos de software adecuados.

2.1.1. Utilidad de Antecedentes

El primer antecedente nos recalca la utilidad e influencia de usar modelos de simulación enfatizando en las siguientes: planificación, control y gestión operativa, mejora de procesos y adopción de tecnología, capacitación y aprendizaje. Nos deja una base a tomar en cuenta para iniciar con la identificación de los diagramas de flujo dentro de los procesos como son la planificación, diseño, codificación, prueba entre otros. Al igual que el presente proyecto utilizan un modelo de dinámica de sistemas, basado en las relaciones entre variables del sistema, para evaluar los beneficios relativos de los enfoques estudiados. Tomaremos en cuenta también las recomendaciones de que el modelo debe ser elaborado y validado, ya que incluso si los resultados del modelo de simulación dentro de este antecedente muestran un comportamiento real, en el trabajo se proponen validar empíricamente en un proyecto real, dicho punto que dentro de nuestro proyecto se prioriza.

El segundo antecedente ilustra claramente cómo utilizar un modelo de simulación dinámica como herramienta de apoyo a la gestión de proyectos de desarrollo de software Scrum. Para ello, presenta un experimento y un estudio de caso que implican el uso de un modelo de simulación dinámica, que sirve de guía a través de la desafiante etapa de validación del método dinámico. Los diferentes diagramas que se construyó también son un apoyo y punto de partida para construir los nuestros con mayor asertividad al modelar el marco de trabajo



Scrum. La adaptación de importantes coeficientes necesarios para las pruebas de escenario fueron también aportes relevantes por parte de este trabajo.

El tercer antecedente nos demuestra el correcto modelado tomando mucho en cuenta la literatura, la experiencia adquirida en la implementación de proyectos de TI con Scrum y estudios de caso relacionados con este marco de trabajo.

Dentro del cuarto antecedente hacen la comparación de los modelos establecidos de Dinámica de Sistemas (SD) de proyectos de software con modelos de desarrollo ágil, creando un nuevo modelo de proyecto ágil SIMULINK™ mínimo, siendo también una valiosa referencia y apoyo para el presente proyecto con sus diferentes diagramas y funcionalidad del mismo. Nos aclaran algo muy importante, que hay pocos modelos de simulación de procesos ágiles y muchos todavía son bastante complejos y requieren un experto en SD para usarlos.

El quinto antecedente nos ayuda con la comprensión de la dinámica de gestión de cambios, planificación, controles ágiles y otras prácticas dentro del marco de trabajo Scrum. La forma en la que se hace la validación del modelo dinámico tanto de la estructura y comportamiento también es una guía para el propósito de nuestro proyecto, ya que también lo usan en un estudio de caso de un proyecto real. Es complejamente bueno y tan bien interpretado dinámicamente el marco de trabajo Scrum que, por ello el modelo puede ser considerado como una "teoría" del desarrollo ágil de software y por ello es que hacemos el uso de esta para nuestra investigación.

2.2. BASE TEÓRICA

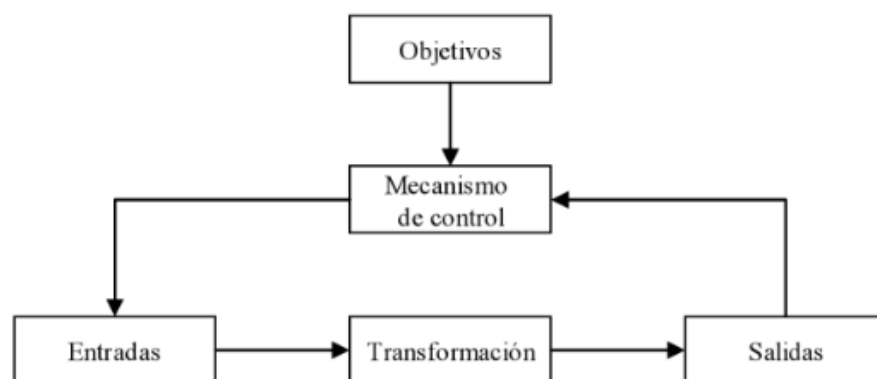
2.2.1. Sistema

El sistema está conformado por diferentes partes coordinadas entre sí, unidas entre ellas que tienen por finalidad lograr un objetivo. Alarcón (2010) afirma:

Un sistema se compone de varias partes que trabajan juntas para lograr un objetivo común. A pesar de la amplia gama de sistemas disponibles, la mayoría de ellos pueden modelarse utilizando cinco bloques fundamentales: objetivos, sección de transformación, elementos de entrada y salida, mecanismos de control y sección de transformación. Los elementos de entrada que deben modificarse en la sección de transformación proporcionan recursos con acceso al sistema. El mecanismo de control regula este proceso para cumplir el objetivo predeterminado. El resultado sale del sistema a través de los elementos de salida una vez completada la transformación. (p.11).

Figura 1

Modelo general de un sistema



Nota: fuente (Alarcón, 2010)

Se entiende que las partes del sistema interactúan y se influyen mutuamente. Debido a estas diversas interrelaciones, el sistema parece único



en su entorno y puede mantener esta singularidad a lo largo del tiempo y en otros contextos. El termino sistema se puede aplicar en diferentes aspectos y escenarios de la vida, por ejemplo, el sistema solar o planetario que interactúan entre si gracias a la fuerza gravitacional y sus múltiples componentes. Se puede también mencionar a una población o estado que también es un sistema ya que está conformada por diferentes componentes, así como el transporte, el comercio, la política, las instituciones y tantos otros que interactúan entre si conllevando la vida con sus diferentes objetivos, pero dinámicamente relacionadas ya que si una de estas no actuara tendríamos un sistema diferente o menos llevadera para el ser humano. Los diferentes departamentos de una empresa serían los componentes del sistema, ya que coordinan entre ellas, para producir el bien o servicio ofrecido, encaminados a lograr el objetivo de la empresa. La lista de ejemplos es interminable, ya que casi de todo lo que nos rodea se puede decir que es un sistema.

Un sistema está formado por varios subsistemas, o sistemas más pequeños, que interactúan para intercambiar energía y convertirla (alcanzar un objetivo).

Pensemos en la estructura familiar. Entre otras cosas, está formada por padres e hijos, que son sistemas separados en sí mismos. Por otra parte, cuando las piezas que componen un fenómeno interactúan entre sí, también se consideran sistemas (Carmona, 2011, p.12).

Los componentes de un sistema se logran caracterizan a través de sus atributos, de modo que las influencias entre las partes se convierten en relaciones entre los atributos característicos.



2.2.2. Modelo

El modelo es una maqueta que representa un aspecto determinado de la realidad. Alfonso U., Carla M. (2013) afirman “Un modelo es una ilustración de un sistema creada con un objetivo concreto.” Líneas más abajo nos dicen también “en lugar de experimentar con el sistema real puede realizarse un modelo de la operación de cada una de las configuraciones y comparar el comportamiento de los modelos” (Sección introducción, párrafo 15). El modelo es una herramienta construida a partir de un sistema real con objetivos de estudio y análisis en sus diferentes aspectos funcionales.

Hay formas más rápidas y menos costosas de lograr el mismo objetivo, por lo que no es necesario construir un modelo. Existen esencialmente otros dos métodos, que son la intuición y la estadística.

- En situaciones en las que hay abundancia de datos históricos y además podemos presumir que la realidad no va a cambiar, la estadística, o las técnicas de cálculo numérico, son muy útiles para resolver muchos problemas. Por ejemplo, basta con tener suficientes datos del pasado para estimar el número de coches que pasarán hoy por delante de la puerta de tu casa, suponiendo que la calle no ha cambiado.
- Has llegado hasta aquí gracias a tu intuición, así que no la infravalores. Gracias a nuestra pericia o experiencia, ya podemos intuir con seguridad la solución a muchas dificultades (García, 2017a).

Pero si es primordial una mayor confianza, el cual se requiere en este proyecto, será necesario desarrollar un modelo de simulación, tomando como base



lo mencionado por García en los anteriores párrafos, ya que es importante los datos estadísticos y la experiencia. La construcción de esta tiene que ser una imagen aceptable del sistema y que posibilite cumplir el objetivo planteado. Hablamos entonces de los modelos matemáticos que dentro de ella se tiene una variedad de clasificaciones como son los cualitativos, cuantitativos, estáticos, dinámicos entre otros, pero la que más nos interesa dentro de esta investigación es el modelo de simulación.

2.2.3. Simulación

Una manera de visualizar comportamientos de un determinado sistema dentro de un computador, sin afectar directamente el sistema real. Belda y Grande (s. f.), dentro de sus escritos relacionado a los modelos de simulación nos dice lo siguiente:

Mediante el uso de la simulación, es posible planificar, evaluar y considerar tanto los procesos nuevos como los actuales sin asumir el riesgo que conlleva hacerlo en un sistema real. Dicho de otro modo, esto permite a las empresas examinar sus procedimientos desde un punto de vista metódico, con el objetivo de obtener una comprensión más profunda de la relación entre ellos y hacer predicciones más precisas sobre determinadas circunstancias. La aplicación de la teoría de la simulación permite evaluar, reconsiderar y medir diversos aspectos de un nuevo proceso, como la satisfacción del cliente, la utilización de recursos y las medidas de ahorro de tiempo. Por todas estas opciones, la simulación es un instrumento perfecto para el proceso de replanteamiento de la empresa (p.2).

La simulación entonces nos dará diferentes respuestas a lo que es dudoso como respuesta del - ¿qué sucedería sí? - y con el beneficio de no

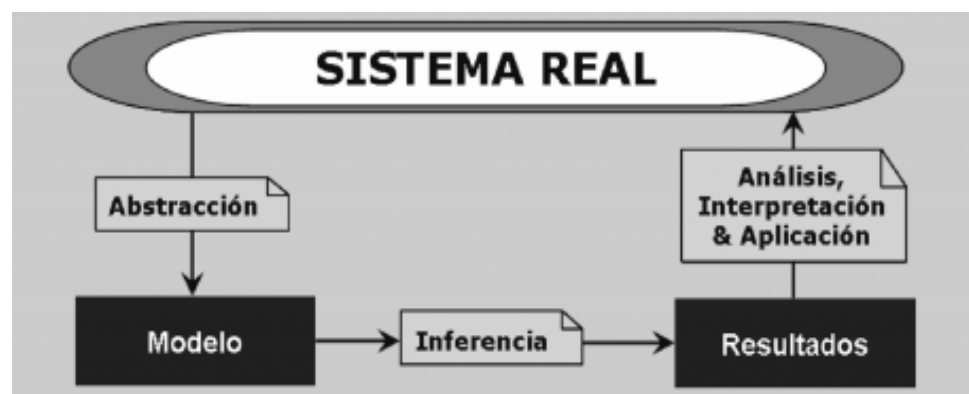
tocar o afectar el sistema real, si no que se ejecuta digitalmente con un modelo construido dentro del computador.

En sus diversos enfoques (niveles de agregación), la simulación es una herramienta crucial para apoyar el proceso de construcción de modelos, ya que ofrece imágenes (comportamientos de cada variable) que muestran al modelizador la precisión con la que representa matemáticamente la realidad. Además, la simulación puede utilizarse para el análisis de sensibilidad, la validación final del modelo y la experimentación en general para la obtención de conocimientos o para la definición de estrategias de control o intervención sobre la realidad. (Modelo de Simulación, s. f.).

En la siguiente figura se muestra el esquema que generalmente representa el proceso de modelado.

Figura 2

Proceso de modelado



Nota: fuente (Izquierdo, Galán Ordax, Santos, & Del Olmo Martínez, 2008)

2.2.3.1. Clasificación de la simulación

- **Estática o Dinámica**

La simulación es estática si dentro del modelo las variables de estado no son influenciadas por el tiempo, ya que el número de



ecuaciones con la cual trabaja es en general una sola, mientras que es dinámica cuando el tiempo es una de las variables más importantes del modelo, que una y otra vez influye en sus diferentes ecuaciones en diferentes tiempos.

La simulación dinámica está en constante evolución ya que actúa con constantes cambios con el pasar del tiempo, por esto lo más normal en este tipo de simulación es el análisis frecuente o en cada momento del tiempo. Dentro de la simulación dinámica existen dos tipos la simulación continua y discreta, según Abad (2002) afirma:

Tanto la simulación continua, en la que se supone que el sistema cambia continuamente de estado, como la discreta, en la que los cambios se producen en momentos concretos y puntuales. Sus nombres se deben a que el conjunto de estados es discreto en el segundo caso y continuo en el primero (p.102).

- **Simulación por cuantos y por eventos**

En la simulación por eventos, se está al pendiente de cada suceso que ocurre y ocurrirá anticipando un suceso posterior. Así también nos dice Abad (2002) lo siguiente:

El tipo de simulación dinámica discreta en la que la variable temporal se controla moviéndola hasta la ocurrencia del siguiente suceso (o evento) se conoce como simulación dirigida por eventos o asíncrona. Esto sugiere que es necesario controlar cuidadosamente el siguiente suceso, es



decir, determinar cuál de los sucesos potenciales en un futuro próximo tiene más probabilidades de ocurrir (p.102).

A continuación, se actualiza el sistema identificando el nuevo estado que ha producido este suceso y generando aleatoriamente la cantidad de tiempo hasta el siguiente suceso de cualquier tipo que puede ocurrir mientras se está en este estado. Además, se anotan los datos deseados sobre el comportamiento del sistema

Por otro lado se tiene la simulación por cuantos que sería bastante diferente a la anterior descrita, así pues nos lo dice Abad (2002) con los siguientes párrafos:

La simulación asíncrona, o de cuantos, responde a una filosofía totalmente distinta. Para estudiar el sistema, que cambia a lo largo del tiempo, es necesario prever cuantos -pequeños intervalos de tiempo de longitud fija- durante los cuales, como máximo, se considera posible un acontecimiento.

En general, la simulación por eventos es mucho más rápida que la simulación por cuantos, pero también más precisa y difícil de construir. No obstante, en la simulación dinámica continua, esta última suele ser la única opción práctica (p.103).

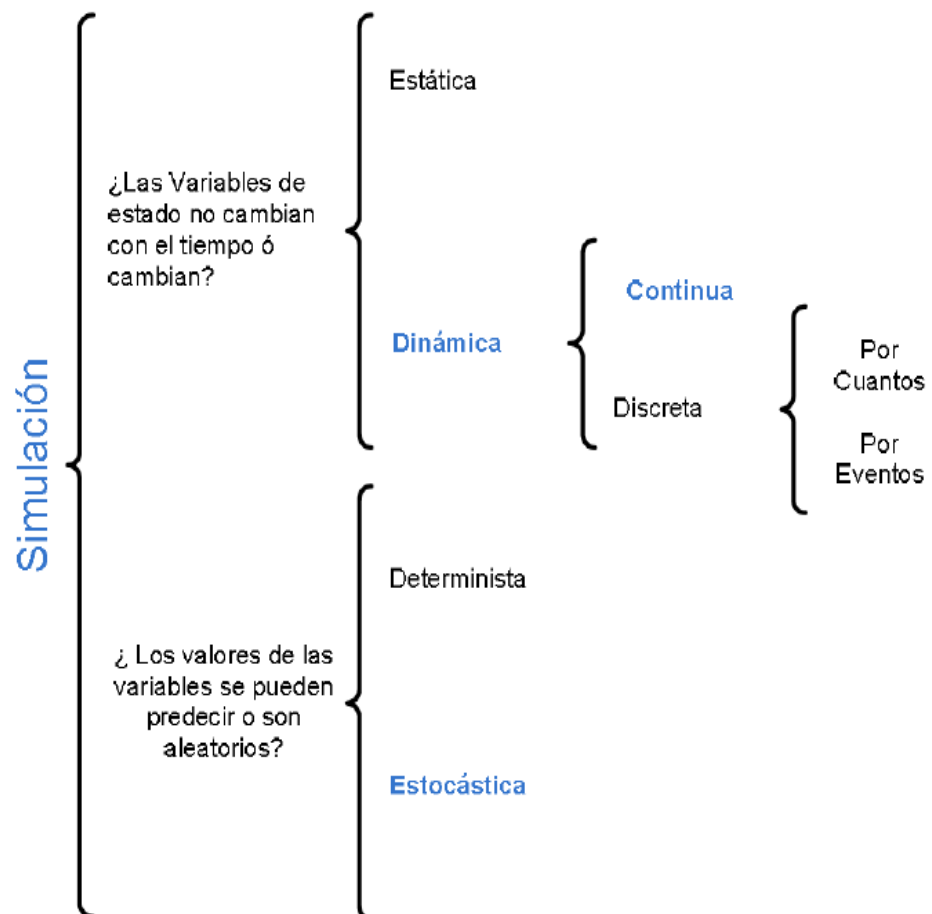
- **Estocástica o Determinística**

Dentro de esta clasificación se diferencian por la procedencia de las variables de valor que trabajan dentro de la simulación o incluso su salida, siendo estocástica si su variable procede de una aleatorización y determinística si los datos son claramente conocidos con precisión.

En la posterior figura se visualiza el resumen de algunos tipos de simulación.

Figura 3

Resumen de los tipos de simulación



Nota: fuente (Barceló, 1996)



2.2.3.2. Metodología de Simulación

Las fases de la simulación son las siguientes que se describirán brevemente:

- **Definición y Formulación del Problema**

Es muy importante la definición y formulación del problema y la definición exacta del objetivo que tendrá el proyecto, ya que esta enrumbará el camino a seguir para lograr el objetivo planteado y dar solución al problema del proyecto. Se debe definir claramente el logro que se pretende obtener, así también las hipótesis que se probaran o refutaran según lo trazado. “Existe el riesgo de que el proyecto de simulación no resuelva con precisión la cuestión para la que se ha buscado y no cumpla las expectativas fijadas si los objetivos son ambiguos u oscuros.” (Petit, 2009, p.18).

- **Conceptualización del Sistema**

Una vez desarrollada la definición y formulación del problema, se tiene que continuar con la segunda etapa, la cual es el proceso de abstracción de la realidad para conceptualizar la porción de ésta, como un sistema. Se tendrá que tomar mucho en cuenta la literatura y el estado de arte de la línea de investigación del proyecto conllevándonos a una comprensión altamente precisa de los procesos y mecanismos básicos de funcionamiento del sistema como tal.

- **Formulación de un Modelo Matemático**

Tras la definición de los objetivos experimentales y el conocimiento del sistema, el siguiente paso consiste en formular un modelo matemático que ilustre las relaciones entre las variables



endógenas y exógenas del sistema. Cabe señalar, como las variables exógenas son fuerzas externas que influyen en el sistema, pueden ser de tipo aleatorio. El proceso de definición de estas variables será considerablemente más sencillo si la fase anterior se ha desarrollado adecuadamente.

- **Estimación de Parámetros**

Siguiendo a la formulación del modelo matemático viene la estimación de parámetros, esto con la verificación e importancia de la estadística de estas estimaciones, para este propósito se toma lo obtenido de la realidad.

- **Evaluación de Modelo**

Al llegar hasta esa fase se continua con la evaluación del modelo. La práctica es la ejecución del modelo y la comparación de un cálculo manual, o para mayor precisión, también se hace la verificación con la comparación de los datos históricos y resultados del modelo ejecutado. Si el modelo no supera satisfactoriamente esta evaluación, se tendrá que retomar el proceso desde su primera etapa, analizando las posibles variantes que no lograron conllevar al modelo a los resultados deseados.

- **Formulación del Programa de Computadoras**

Para desarrollar un programa informático para experimentos de simulación se necesitan básicamente tres elementos: 1) Diagrama de flujo: una de las características del programa informático es un diagrama de flujo en el que se esboza el orden lógico de los acontecimientos que deben desarrollarse. 2) Datos de entrada y



condiciones iniciales: aquí es importante averiguar qué valores deben asignarse inicialmente a las variables y parámetros del modelo. Para ello, hay que utilizar técnicas de ensayo y error con el fin de obtener resultados objetivos. Y, por último, 3) El problema que hay que abordar en la generación de datos es la creación de procedimientos numéricos para la generación de datos, que pueden generarse internamente mediante subrutinas o introducirse desde fuentes externas.

- **Validación**

La validación de modelos de simulación dará al modelo una mayor confiabilidad acerca de los resultados que genere al ser ejecutada, esta validación también ayudará con la detección de posibles errores, que diferencien grandemente el sistema real de lo modelado, y claro lo que se precisa es lo contrario. Para esta práctica se tienen diferentes metodologías, formas y maneras de validar, pero se tomarán como base los siguientes dos puntos:

En primer lugar, hay que contrastar los valores simulados de la variable de salida o endógena con los datos históricos disponibles.

El segundo paso consiste en evaluar hasta qué punto el modelo de simulación predice el comportamiento del sistema real en distintos momentos.



2.2.4. Dinámica de Sistemas

Por otra parte, dado que el sistema está formado por varios componentes, la dinámica del sistema describe cómo estos componentes se alteran o cambian con el tiempo bajo el efecto de influencias externas o mutuas. Esta práctica se ejecuta dentro del computador como la simulación de un modelo construido a partir de una realidad determinada. (García, 2017a) afirma:

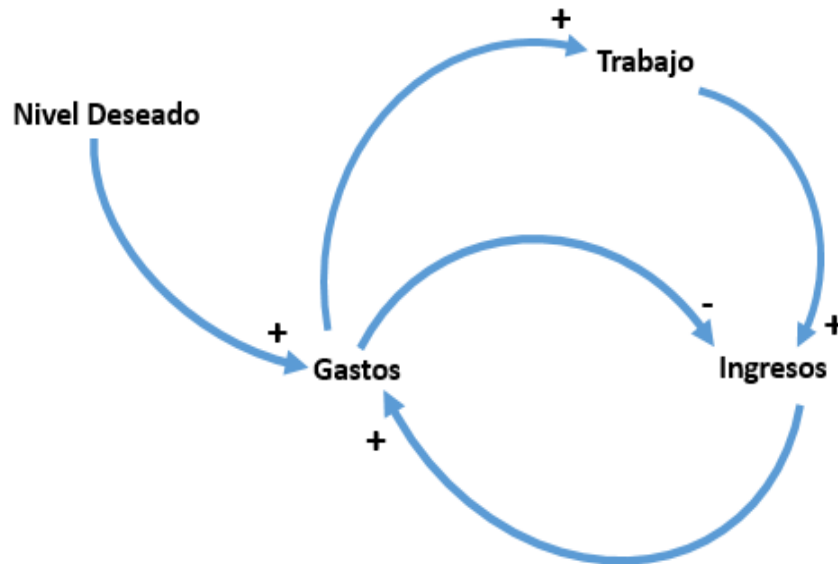
La dinámica de sistemas es la metodología que nos permite introducir sistemáticamente nuestra percepción de la realidad en un ordenador y construir modelos de simulación para verificar la coherencia de esta percepción de la realidad, observar lo sensibles que son sus componentes a diversos cambios posibles y desarrollar sugerencias de actuación plausibles (p.10).

Cada componente del sistema y sus respectivas relaciones se pueden representar con grafos orientados, un grafo está compuesto por nodos y aristas, las mismas que representan componentes y relaciones respectivamente, las aristas en formas de flechas son acompañadas con dos tipos de símbolos (+) y (-), que serían la representación del tipo de relación que se da entre un componente y otro según la orientación de la flecha, las que son acompañadas con el símbolo (+), representan la relación del tipo positivo, es decir, la influencia de un nodo X que incrementa también incrementa el nodo Y con el cual se relaciona, tenemos también la relación con el símbolo (-), que representa que si el nodo X disminuye Y hará lo opuesto en este caso incrementara y si el caso es que X incrementa entonces Y disminuye, es decir este símbolo representa que si el primer nodo varia en cualquier sentido el nodo influenciado hará lo contrario.

Un diagrama de influencias básico se observa en la Figura 4.

Figura 4

Diagrama influencia básico



Nota: elaboración propia

Bucle de Realimentación Negativa

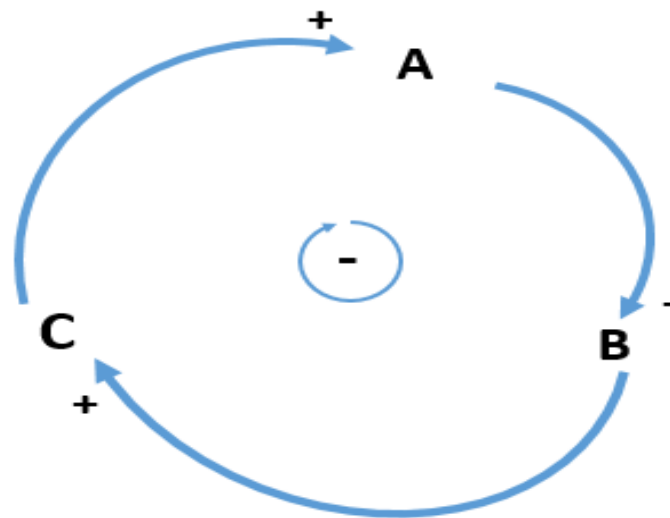
El bucle de realimentación negativa es considerado como la estructura estabilizadora del sistema, puesto que el objetivo principal de este bucle es mantener la estructura que tiene, esto ante alguna perturbación.

En la Figura 5 se muestra un bucle de retroalimentación negativa con los componentes A, B y C, en ella A influye negativamente a B, y a su vez, B influye positivamente a C y este influye positivamente sobre A. Lo que ocurre realmente dentro de este bucle es que se da un incremento en A y por el tipo de relación que se visualiza con B es del tipo negativo esto hace que B disminuya, influyendo esto sobre C que por el tipo de relación que B tiene sobre C esta disminuye y por ultimo C tiene la relación del tipo positivo sobre A por ello este último también disminuye, haciendo que el incremento

de A en el bucle de realimentación negativo se contrarreste, esto hace que el bucle sea conocido como el bucle regulador. Una forma práctica de identificar a un Bucle de Realimentación Negativa, es la verificación contabilizada de todas las influencias negativas nos resulte impar.

Figura 5

Estructura de realimentación negativa



Nota: elaboración propia

Características:

- Son denominadas como bucles reguladores o estabilizadores.
- El bucle de retroalimentación negativa se llega a dar si conforman un número impar de influencias negativas.
- Estas tienden a resistirse al cambio tratando de mantener su objetivo.
- Es necesario su identificación ya que dentro del modelo pueden tener un comportamiento complejo y problemático.



Bucle de Realimentación Positiva

Los bucles de realimentación positiva existen en oposición a los bucles de realimentación negativa. su sección explicará su estructura y cómo se comporta.

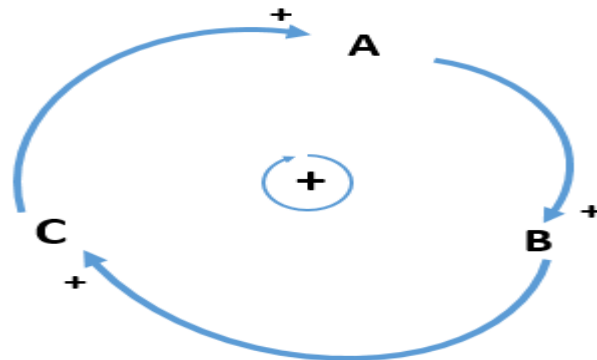
La figura 6 ilustra gráficamente un bucle de realimentación positiva. Se representa la relación entre las variables A, B y C, en la que A tiene una influencia positiva sobre B, B tiene una influencia positiva sobre C y C tiene una influencia positiva sobre A.

Si se examina más detenidamente el conjunto de impactos, se observa que, en el primer caso, un aumento de la variable A provocará un aumento de la variable B porque la primera tiene un efecto positivo sobre la segunda. Cuando B suba, C se encontrará en una circunstancia similar. Por último, un aumento de C provocará un aumento de A.

Debido a las diversas influencias positivas entre todas las variables, el sistema crece infinitamente, creando una situación inestable en este caso. Sin embargo, todo sistema tiene un límite o regulador de crecimiento, por lo que, si la modelización se realiza correctamente, el caso de crecimiento infinito se producirá de forma irregular.

Figura 6

Estructura de realimentación positiva



Nota: elaboración propia.

Características

- En el caso de que la totalidad de las influencias negativas es par o todas las influencias son positivas.
- Una determinada acción hace que las siguientes según la influencia crezcan o disminuyan constantemente.
- Conocido también como círculo vicioso o bola de nieve.
- El efecto de este tipo de bucles es prácticamente lo opuesto al Bucle de retroalimentación negativa, puesto que este es un desestabilizante del sistema.

2.2.5. Metodología Dinámica de Sistemas

La dinámica de sistemas hace referencia a una metodología presentada por Jay W. Forrester.

Como ingeniero eléctrico que empezó en el campo de los servomecanismos y el diseño informático, Forrester fue capaz de comprender cómo un sistema oscilante tendería hacia el resultado deseado debido a los retrasos en la transmisión de información. Desarrolló lo que primero denominó dinámica industrial -que más tarde se conocería como dinámica de sistemas- comparando



la mecánica de los servomecanismos y llegando a la conclusión de que, puesto que eran similares, podían representarse de forma parecida.

2.2.5.1. Fases de la Metodología Dinámica de Sistemas

Las distintas fases que componen la metodología, son: Conceptualización, Formulación y Análisis-Evaluación según (Aracil, 1995), las mismas que a continuación se describen:

a) La Fase de Conceptualización es la primera fase

Obtener una visión y comprensión de un determinado acontecimiento del mundo real. Consiste en:

- Familiarización con el problema.
- Tratamiento de literatura al respecto.
- Opiniones de expertos.
- Experiencias propias.

Una vez hecho esto, es necesario definir adecuadamente los componentes del problema y explicarlos de forma clara, sucinta y precisa.

La descripción del comportamiento dinámico objeto de estudio puede formar parte de este paso. Esta descripción se utilizará para trazar el comportamiento temporal de las magnitudes primarias de interés, lo que se conoce como Modo de Referencia y que proporciona una aproximación de los trazados que se generarán a partir del modelo original.

Al modelizar un fenómeno histórico, este modo de referencia representará el comportamiento histórico que se ha documentado y que debe reproducirse en el modelo.



Cuando se modelan escenarios futuros, el modo de referencia se vuelve más ambiguo, pero aun así debe ser capaz de capturar la gama de diversos tipos, modos o patrones de desarrollo a través de las variaciones apropiadas de los parámetros.

El horizonte temporal del modelo se establece al establecer el modo de referencia.

b) La Fase de Formulación es la segunda fase

En base al Diagrama Causal se procede a la formulación del sistema. Los pasos a seguir son:

- Establecer el diagrama de Forrester
- Partiendo del diagrama de Forrester, escribir las ecuaciones del modelo.
- Asignar valores a los parámetros.

c) La Fase de Evaluación es la tercera fase

Las simulaciones sirven para evaluar la coherencia del modelo y de las hipótesis sobre las que se ha construido.

En esta fase, se realiza un análisis de sensibilidad, lo que significa que se investiga la dependencia de las conclusiones de posibles variaciones en los valores de los parámetros.

Se dan recomendaciones tras analizar el comportamiento del modelo con varias políticas alternativas.

Hay múltiples repeticiones de algunos pasos en este proceso no lineal.



2.2.5.2. Utilizando la Metodología de Dinámica de Sistemas

La metodología presenta una serie de componentes cruciales, como los diagramas de influencia, los diagramas de Forrester y las múltiples clases de variables. Se describe, en general, cómo crear modelos de simulación utilizando este paradigma.

Se pasa a la primera aproximación del diseño del modelo mediante la construcción de los Diagramas de Influencia utilizando los datos y la información recopilados en la fase anterior y que son apropiados para el dominio del problema a resolver. Este tipo de imagen permite identificar los arquetipos sistémicos que componen el sistema y comprender sus fuerzas y estructuras.



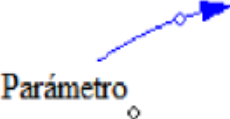
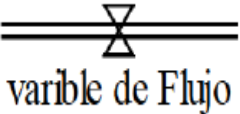
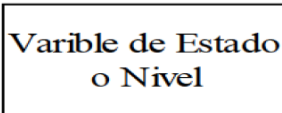



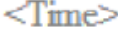
El procedimiento de convertir el Diagrama Causal en Diagramas Forrester debe completarse en la segunda fase. Esta fase también implica identificar las variables y clasificarlas en "Niveles", "Flujos" y "Auxiliares". Es crucial asignar los valores y unidades de medida adecuados a las distintas constantes y variables que posteriormente permitirán simular el sistema.

En principio, Forrester identificó tres categorías de variables: variables de estado, de flujo y auxiliares. Las magnitudes que han experimentado una evolución especialmente considerable están representadas por las variables de estado. La relación de las variables de estado con las variables de flujo controla su evolución en el tiempo. Las variables auxiliares, en definitiva, sirven de trampolín para determinar las variables de estado a partir de las variables de flujo relacionadas. Los

componentes de un diagrama Forrester se muestran en la Tabla 1 junto con la forma en que Vensim PLE los representa.

Tabla 1

Elementos del diagrama de forrester de Vensim.

SÍMBOLO	DESCRIPCIÓN
	Canales de información: sirven para transferir datos entre las múltiples variables sin que se queden en el propio modelo.
	Canales materiales: facilitan el diálogo entre niveles y flujos. Se supone que comunican valores que corresponden a magnitudes físicas a través de estos canales.
	Constantes y Parámetros: Los valores utilizados en los cálculos por las variables auxiliares o en los flujos se almacenan en constantes y parámetros. Las constantes y los parámetros sirven de base para todos los canales y flujos de información.
	Variables de flujo: Son funciones del tiempo. Expresan las acciones derivadas de las decisiones tomadas en el sistema que hacen que éste varíe relacionando unas variables con otras y representando los vínculos causales que existen entre ellas.
	Variables de nivel: son las variables cuya evolución es importante para el estudio del sistema, y se correlacionan con las variables de estado de la teoría de sistemas.
	Nube: una nube puede considerarse como un sumidero, un pozo o una fuente. Puede entenderse como un estado esencialmente interminable y carente de interés.
	Retardos: se emplean para imitar desfases temporales en las vías de transmisión de materiales o información.
	Variables Auxiliares: son aquellas que utilizan otras variables y que se emplean en cálculos intermedios. Reciben canales de información todo el tiempo.
	Shadow: referirse a una variable existente o a una variable del sistema (variable de control de la simulación) en varias vistas, este componente es único.

Nota: adaptación de (García, 2017b).



Una vez terminado el diagrama de Forrester, se utiliza una herramienta computacional para programar las fórmulas o ecuaciones, se hace uso del Vensim PLE 6.3 Versión Académica.

2.2.6. Proyecto

El proyecto es la planificación de varias actividades relacionadas y orientadas a uno o varios objetivos trazados, nos dice Talledo (2013), que “ Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único”.

Mencionándonos también el Project Management Institute (2008) en su cuarta edición lo siguiente:

Un proyecto es un esfuerzo breve iniciado con la intención de producir un bien, servicio o resultado especial. El carácter transitorio de los proyectos denota un comienzo y un final claros. El proyecto llega a su fin cuando se cumplen sus objetivos, cuando se cancela porque sus objetivos no se alcanzarán o no pueden alcanzarse, o cuando ya no existe la necesidad que lo motivó. Ser temporal no siempre implica ser breve. Dado que la mayoría de los proyectos se llevan a cabo para producir un efecto duradero, este atributo no suele aplicarse al producto, servicio o resultado producido por el proyecto. Por ejemplo, se prevé que los resultados de un proyecto de construcción de un monumento nacional perduren durante décadas. Sin embargo, las iniciativas también pueden tener efectos duraderos en el medio ambiente, la economía y la sociedad.

Cada proyecto produce un bien, servicio o resultado distinto. Aunque algunos productos del proyecto puedan tener elementos que se repiten, esta repetición no cambia la originalidad esencial del trabajo del



proyecto. Los edificios de oficinas, por ejemplo, se construyen con el mismo equipo y los mismos materiales, pero cada ubicación es diferente, con un diseño distinto, condiciones distintas, contratistas distintos, etc.

El proyecto tiene varias naturalezas, es por ello que tienen entonces aspectos formales y teóricos diferentes, haciendo que sean diferentes los objetivos de estudio según cada proyecto.

A grandes rasgos, un proyecto es el empeño por encontrar una solución inteligente a un problema que pretende atender, entre otras cosas, una necesidad que tienen las personas.

Así pues, aunque los conceptos, las inversiones, las tecnologías y la metodología pueden variar, todos ellos están dirigidos a atender necesidades humanas en todas sus formas, incluidas las relacionadas con la educación, la alimentación, la salud, el medio ambiente, la cultura, etc. (Urbina, p.2).

2.2.6.1. Fases de un Proyecto

Un proyecto es un conjunto de tareas conectadas entre sí para alcanzar un objetivo común. Los proyectos son distintos, a diferencia de los procedimientos o procesos. Cada proyecto consta de fases y etapas, requisitos, determinadas tareas, un equipo de trabajo y un periodo de ejecución. Dentro del ciclo de vida de un proyecto, hay fechas fijas de inicio y finalización para cada proyecto. Lo siguiente es lo que nos dice Hernández Sampieri, Fernández Collado, Baptista Lucio, Méndez Valencia, & Mendoza Torres (2014) del inicio del proyecto:

Un proyecto comienza con la formulación de preguntas basadas en la observación; éstas surgen en conferencias, al leer



periódicos o en la vida cotidiana. Estas preguntas deben ser verificadas por expertos en la materia para asegurarse de que son pertinentes, útiles para llevar a cabo la investigación y, si tienen éxito, harán avanzar el campo o abordarán un problema acuciante (p.31).

El proyecto cuenta con diferentes fases, la misma que procura aligerar la carga total de trabajo, ya que se divide en componentes menores facilitando el desarrollo y seguimiento de las actividades.

Nos propone Talledo (2013), dividir un proyecto en las siguientes cinco fases:

- Iniciación
- Planificación
- Ejecución
- Seguimiento y control
- Cierre

Existen enfoques para proyectos de software que requieren fases más especializadas y específicas para obtener el resultado deseado.

La identificación de un problema, que se define como la incapacidad de modificar las condiciones organizativas para alcanzar los objetivos previstos, requiere la creación y ejecución de proyectos que terminan con la instalación de nuevas capacidades dentro de la organización. Dicho de otro modo, la iniciativa pretende cerrar la brecha que existe entre la situación prevista y la actual, ya que esta brecha se considera un problema. Dado que un proyecto es especial por ser único en su género, requiere una metodología o disciplina rigurosa para ayudar a la



empresa a abordar una tarea que se hace por primera vez, como la introducción de un nuevo producto o la instalación de un nuevo sistema. Luego, los proyectos son el medio esencial por el que las empresas llevan a cabo reformas organizativas. En ellos deben participar profesionales y técnicos que, con frecuencia, carecen de conocimientos suficientes sobre gestión de proyectos.

2.2.6.2. Gestión de Riesgos

Según el PMI (Project Management Institute), afirma:

Los acontecimientos que, en caso de producirse, podrían tener un impacto favorable o desfavorable en los objetivos del proyecto son la fuente de los riesgos del proyecto. Los riesgos tienen un origen y un efecto si se materializan. El riesgo está asociado tanto a las oportunidades de mejorar los objetivos del proyecto como a las amenazas para su consecución.

Según este punto de vista, los riesgos son fuentes de incertidumbre y no siempre cosas malas, por lo que identificarlos, evaluarlos y gestionarlos se convierte en algo esencial.

La autoridad de gestión de proyectos PMI define seis procesos esenciales de gestión de riesgos:

- Planificación de Riesgos: Esbozar cómo se gestionarán y procesarán los riesgos a lo largo del proyecto.
- Identificación de Riesgos: El proceso de identificar peligros potenciales para el proyecto y registrar sus características.
- Evaluación de los Riesgos: Para determinar la importancia relativa de cada uno, debe realizarse un análisis cualitativo.



- **Cuantificación de Riesgos:** Evaluación cuantitativa de cada riesgo para determinar los posibles resultados. Dado que ayudará a establecer prioridades, este punto debería derivarse del anterior.
- **Definición de Respuestas ante Riesgos:** Describe las reacciones potenciales a los peligros y oportunidades que se han reconocido.
- **Seguimiento y Control de Riesgos.** control de todas las variaciones de riesgo durante el transcurso de un proyecto.

Aunque una metodología orientada al riesgo se esforzará por proporcionar las herramientas necesarias para tener en cuenta los riesgos, gran parte del cómo y de las directrices para la toma de decisiones procederán principalmente de las lecciones aprendidas en proyectos anteriores, que ofrecen un conocimiento de primera mano de los problemas encontrados, sus efectos y los diversos enfoques utilizados para abordar adecuadamente esos efectos.

2.2.7. Métodos Ágiles

En la década de 1990, varios desarrolladores de software propugnaron nuevos "métodos ágiles" que permitían al equipo de desarrollo concentrarse en el software en lugar de en el diseño y la documentación. El enfoque incremental de la especificación, el desarrollo y la entrega del software es un componente fundamental de todos los enfoques ágiles. Funcionan mejor en el diseño de aplicaciones, porque las necesidades del sistema suelen cambiar rápidamente a medida que avanza el desarrollo. Su objetivo es proporcionar a los clientes un software rápidamente utilizable para que puedan sugerir nuevas características y modificaciones para futuras revisiones del sistema. Su objetivo es agilizar el



proceso burocrático eliminando papeleo que quizá nunca se necesite y suprimiendo tareas de dudosa relevancia a largo plazo.

Como veremos más adelante en este capítulo, la programación extrema (Beck, 1999; 2000) es posiblemente la metodología ágil más conocida. El desarrollo adaptativo de software (Highsmith, 2000), Scrum (Cohn, 2009; Schwaber, 2004; Schwaber y Beedle, 2001), Crystal (Cockburn, 2001; Cockburn, 2004), DSDM (Stapleton, 1997; Stapleton, 2003) y el desarrollo basado en características (Palmer y Felsing, 2002) son algunas otras metodologías ágiles. La eficacia de estas técnicas fomentó cierta integración con técnicas de desarrollo más convencionales basadas en el modelado de sistemas, dando lugar a los conceptos de modelado ágil (Ambler y Jeffries, 2002) y ejemplificaciones ágiles del Proceso Racional Unificado (Larman, 2002).

Aunque el desarrollo y la entrega incrementales son la base de todas estas técnicas ágiles, sugieren métodos distintos para conseguirlo. No obstante, el manifiesto ágil esboza algunos principios a los que todas ellas se adhieren, lo que las hace bastante similares (Sommerville, 2011).

Tabla 2

Los principios de los métodos ágiles

PRINCIPIO	DESCRIPCIÓN
Participación del cliente	Los clientes deben participar activamente en el proceso de desarrollo. Es su responsabilidad formular, clasificar y evaluar las nuevas necesidades del sistema, así como sus interacciones.
Entrega incremental	El cliente determina lo que debe incluirse en cada incremento de software que se desarrolle.
Personas, no proceso	Hay que reconocer y utilizar las habilidades del equipo de desarrollo. Sin procedimientos establecidos, los miembros del equipo deben tener libertad para crear sus propios métodos de trabajo.
Adoptar el cambio	Espere a que se produzcan modificaciones en los requisitos del sistema antes de diseñarlo para adaptarse a esos cambios.
Mantener simplicidad	Asegúrese de que tanto el software que se va a producir como el proceso de desarrollo son sencillos. Esfuérzate por reducir la complejidad del sistema siempre que sea posible.

Nota: adaptación de (Sommerville, 2011, p.60)

A continuación, se enumeran los enfoques más pertinentes que ahora se categorizan como "Ágiles":

- Programación Extrema, o XP. Con el énfasis puesto en el desarrollo de fuertes vínculos interpersonales como secreto del éxito, fomenta la colaboración, presta atención a la educación de los desarrolladores y crea un ambiente de trabajo positivo. La base de XP es la retroalimentación constante del cliente al equipo de desarrollo, las líneas abiertas de comunicación entre todas las partes, las soluciones directas y la capacidad de adaptarse a los cambios. Funciona bien en



proyectos en los que existe un nivel significativo de riesgo técnico y en los que los requisitos son imprecisos y están sujetos a cambios.

- Crystal Methodologies. Se trata de una colección de enfoques de desarrollo de software que ha presentado Alistair Cockburn. Su atención se dirige a los miembros del equipo y a minimizar la cantidad de artefactos generados. El desarrollo de software se considera un juego cooperativo de comunicación y creatividad limitado por los recursos disponibles. Dado que el equipo de desarrollo es un componente crucial, hay que esforzarse por mejorar sus conocimientos y experiencia, además de establecer directrices claras para el trabajo en equipo. El tamaño del equipo determina las políticas que deben aplicarse, que incluyen la creación de categorías codificadas por colores, como Crystal Clear (de 3 a 8 miembros) y Crystal Orange (de 25 a 50 miembros).
- ASD (Adaptative Software Development). Fue propuesto por Jim Highsmith y posee las siguientes características clave: es tolerante a los cambios, iterativo y se centra en componentes de software más que en tareas. El ciclo consta de tres fases clave: aprendizaje, cooperación y especulación. En la primera fase se inicia el proyecto y se planifican las características del software. Las funciones se crean en la segunda fase, mientras que el control de calidad y la entrega al cliente tienen lugar en la tercera. Reiniciar el proceso y aprender de los errores se consigue mediante el examen de los componentes.
- FDD (Feature Driven Development). Jeff De Luca y Peter Coad proponen un procedimiento iterativo que consta de un conjunto de



fases e iteraciones breves (hasta dos semanas). Partiendo de un conjunto de requisitos que debe cumplir el programa, se centra en las fases de diseño e implantación del sistema.

- DSDM (Dynamic System Development Method). Esboza la estructura necesaria para crear un proceso de producción de software. El objetivo de su creación es establecer un enfoque coherente para el desarrollo rápido de aplicaciones, o RAD. Entre sus rasgos destacan la colaboración entre el usuario y el equipo de desarrollo y un proceso progresivo e iterativo. La investigación de viabilidad, el estudio de negocio, el modelado funcional, el diseño y la construcción, y la implementación son las cinco etapas que componen este proceso. Hay retroalimentación para cada una de las tres últimas etapas de iteración.
- SCRUM. El sistema de gestión de proyectos creado por Ken Schwaber, Jeff Sutherland y Mike Beedle se ha utilizado eficazmente en los últimos años. Está pensado para proyectos cuyos requisitos cambian con frecuencia.

2.2.8. Marco de Trabajo Scrum

Scrum es un marco que, visto desde una perspectiva metodológica, aplica una serie de métodos y herramientas para trabajar en equipos cooperativos y autodirigidos con el fin de maximizar los resultados de los proyectos. El estudio sobre "nuevas prácticas de producción" que Hirotaka Takeuchi e Ikujiro Nonaka realizaron y publicaron en su ensayo "El juego del desarrollo de nuevos productos" a mediados de la década de 1980 es donde se seleccionaron por primera vez estos métodos.



A principios de los 90, Jeff Sutherland y Ken Schwaber lo pusieron en práctica por primera vez, y escribieron sobre ello en su libro "Agile Software Development with Scrum".

Bloques de tiempo, artefactos, reglas y equipos scrum, junto con sus correspondientes roles, son algunos de los componentes que conforman scrum.

Scrum es un marco que permite a los equipos multifuncionales trabajar de forma incremental e iterativa en proyectos o productos. Los sprints, comúnmente denominados iteraciones, son ciclos de trabajo utilizados en la estructura del desarrollo. Estas iteraciones deben producirse consecutivamente sin pausa intermedia, con un periodo máximo de cuatro semanas (dos semanas es lo más típico). Los sprints tienen una duración determinada; nunca duran más de la fecha de finalización programada, independientemente de si el trabajo se realiza en su totalidad. Por lo general, los equipos Scrum seleccionan una duración de Sprint y se adhieren a ella para cada Sprint hasta que mejoran y pueden trabajar con ciclos más cortos. Un equipo multifuncional formado por unos siete miembros elige cosas (peticiones de los clientes) de una lista priorizada al inicio de cada sprint. El grupo decide sobre un objetivo compartido para lo que esperan lograr al final del Sprint: un producto concreto que esté "terminado" en su totalidad.

Aunque Scrum se ajusta a los cambios en el Sprint siguiente, no se pueden añadir nuevos elementos durante el Sprint. En su lugar, el pequeño Sprint actual está destinado a concentrarse en un único objetivo, modesto, bien definido y en gran medida coherente.

La Reunión de Planificación de la Entrega, Reunión de Planificación del Sprint, Scrum Diario, Sprint, Revisión del Sprint y Retrospectiva del Sprint son los componentes de Scrum que se basan en bloques de tiempo. Scrum utiliza cuatro partes principales en su metodología. Durante la duración de un sprint, el Product Backlog, Sprint Backlog, Version Burndown, y Sprint Burndown se muestran.

Figura 7

Marco de Trabajo Scrum



Nota: fuente (Deemer, Benefield, Larman, & Vodde, 2009, p.5)

2.2.8.1 Roles

Responsable-Facilitador (Scrum Master): Sus principales objetivos son asegurarse de que la estructura de trabajo se sigue correctamente, de que el ambiente del equipo es el adecuado y de que no hay influencias externas que perturben al equipo. Como portavoz del equipo, se encargará de todas las comunicaciones externas y retransmitirá todas las comunicaciones internas. Esta persona puede desempeñar otras funciones en el equipo, pero nunca puede ser el Scrum Master y el Propietario del



Producto simultáneamente. La persona encargada de organizar la planificación es el Scrum Master.

Proporciona:

- Orientar e instruir al grupo sobre cómo funcionar como una unidad cohesionada y asumir responsabilidades de equipo.
- Examinar y confirmar la pila de productos.
- Moderación de la reunión.
- Resolución de obstáculos en el sprint que puedan impedir que se completen las tareas.
- Manejar la "dinámica de grupo" del equipo.
- Establecimiento, diseño y mejora de los procedimientos de scrum en la empresa. Cortesía hacia el equipo y todas las partes implicadas, respetando las normas de tiempo y forma de scrum.

Propietario del Producto (Product Owner): Toma decisiones en nombre del cliente. Es responsable de la valía del producto. Es imprescindible que esta función la desempeñe una sola persona para agilizar la comunicación y la toma de decisiones. El cliente puede elegir el método de comunicación interna que más le convenga si se trata de una gran organización con varios departamentos. Sin embargo, en el equipo de desarrollo, el cliente está representado por una sola persona que debe conocer bien el producto y tener autoridad para tomar decisiones en su nombre.

Es necesario que el propietario de producto:

- Utilizar scrum para completar las tareas que se le asignen:
 - La creación y administración de la pila de productos.



- Participar en la reunión de planificación de cada sprint y compartir la visión y las historias de usuario.
- Estar familiarizado y tener experiencia trabajando con el mismo equipo

La empresa debe respetar sus decisiones y abstenerse de alterar los componentes o la prioridad de la pila de productos.

Equipo (Team): El desarrollo de las funcionalidades del software es su responsabilidad. Una de las ventajas de Scrum sobre otros enfoques es que da autonomía al Equipo en la gestión del Sprint Backlog y la selección de lo que hay que hacer en el siguiente Sprint. Sin embargo, el ProductOwner es el encargado de supervisar su creación. Un equipo multidisciplinar formado por 7 ± 2 miembros conforman el conjunto de roles del equipo porque cada iteración pasa por todas las fases de desarrollo de un proyecto. Durante la duración del Sprint, la composición del equipo no cambiará, y no se realizarán más cambios hasta que comience el siguiente Sprint.

El grupo trabaja bien en equipo y se esfuerza al máximo por hacer realidad la visión del cliente.

En el equipo:

- Todos son conscientes de la visión del propietario del producto.
- Trabajan junto con el propietario del producto para diseñar la pila del producto.
- comparten el objetivo y la responsabilidad de su consecución en cada sprint.
- Cada miembro participa en la toma de decisiones.



- Se valoran todos los puntos de vista y contribuciones.
- Todos están familiarizados con el modelo de trabajo scrum.

2.2.8.2 Bloques de Tiempo

- **Planificación del Sprint (Sprint Planning)**

Establecer un plan y unos objetivos que los Equipos Scrum y las demás empresas puedan comprender y con los que puedan comunicarse es el objetivo de la planificación de la entrega. Se divide en dos secciones:

Determinar qué tareas de la Lista de Pendientes del Producto se completarán durante el Sprint es el objetivo de la primera fase. El Propietario del Producto y el Equipo, asistidos por el Scrum Master, repasan durante esta reunión los elementos de alta prioridad del Backlog del Producto que más interesan al Propietario y que deberían implementarse este Sprint.

En la segunda fase se decide la estrategia para alcanzar los objetivos del Sprint. Durante esta fase se establece el Backlog del Sprint, que enumera las tareas que el equipo tiene que completar para alcanzar estos objetivos. Se centra en la preparación meticulosa de las tareas para saber cómo llevar a cabo las decisiones tomadas por el equipo. Empezando por la parte superior de la Pila de Productos, el equipo elige los componentes de la pila con los que comprometerse al concluir el Sprint.



- **Reunión Diaria de Sincronización del Equipo (Scrum Daily Meeting)**

Estas reuniones tienen por objeto fomentar la cooperación y el intercambio de información. No duran más de quince minutos y se celebran siempre en el mismo lugar.

- **Ejecución de la Iteración (Sprint)**

Un proyecto se lleva a cabo en bloques de tiempo condensados y predeterminados utilizando Scrum. Suelen durar entre una y cuatro semanas, en función de lo que decida el equipo. Debido a su naturaleza iterativa, cada nuevo Sprint comienza justo después de que finalice el anterior. Al final de cada iteración, se debe proporcionar un resultado global, es decir, una parte del producto final que se puede entregar al cliente con el menor esfuerzo posible. El plan del proyecto, o lista priorizada de necesidades y objetivos del producto, sirve como punto de partida del proceso. El cliente clasifica los objetivos de esta lista según su importancia, sopesando los beneficios que ofrecen frente a sus gastos, y los distribuye entre las iteraciones y la entrega. No es posible añadir nuevos requisitos, hacer cambios en los existentes o alterar los existentes mientras se desarrolla el sprint, aunque el cliente puede maximizar regularmente la utilidad de lo desarrollado y el rendimiento de la inversión replanificando los objetivos al principio de cada iteración. Dicho de otro modo, después de que el equipo se comprometa, cualquier alteración o adición tiene que esperar hasta el siguiente sprint.



Esto hace que un sprint se resume fácilmente como un bloque iterativo de tiempo en el que se producen o actualizan nuevos incrementos de la funcionalidad de un sistema.

Un producto puede planificarse, codificarse, probarse y su arquitectura y diseño pueden cambiar a medida que se desarrolla.

- **Demostración de los Requisitos Completados (Sprint Review)**

Al final de cada Sprint, la revisión del Sprint ofrece un punto de inspección del progreso del proyecto. A la luz de esta inspección, pueden introducirse cambios en el proyecto.

El equipo entrega el incremento del producto al final del sprint. El propósito de la presentación del incremento es promover la cooperación y el intercambio de información. No pueden transcurrir más de dos horas.

- **Retrospectiva (Sprint Retrospective)**

Los objetivos principales de este bloque de tiempo son evaluar el rendimiento del Sprint anterior en términos de personas, relaciones, procesos y herramientas, así como priorizar y clasificar los componentes más cruciales y las posibles mejoras.

Tras la Revisión del Sprint, esta reunión se celebra para debatir la incorporación de comentarios y el cumplimiento de las expectativas. Para los Sprints de un mes, la reunión se limita a un bloque de tiempo de tres horas. Para sprints más cortos se asigna menos tiempo en proporción.



2.2.8.3 Artefactos Principales de Scrum

- **El Tablero de Tareas (Scrum Taskboard)**

Con un Tablero de Tareas Scrum, se puede supervisar la lista de objetivos que deben alcanzarse durante la iteración. Las tareas necesarias para lograr cada objetivo se escriben en post-its junto a él, y se pueden mover a la derecha para cambiar su estado (iniciado, en curso o completado).

- **Lista de Objetivos o Lista de Requisitos Priorizada (Product Backlog)**

Es una lista de los requisitos y objetivos priorizados que reflejan las expectativas y la visión del propietario del proyecto o producto. Es el único lugar donde se pueden encontrar los requisitos para cualquier alteración del producto.

Se trata de una lista dinámica que siempre cambia en respuesta al entorno para determinar qué necesita el producto para ser suficiente, competitivo y útil. Se enumeran las características, funcionalidades, especificaciones, mejoras y correcciones que deben realizarse en el producto para que pueda entregarse en el futuro. Los elementos del Product Backlog tienen características como el pedido, la estimación y la descripción.

- **Lista de Tareas de la Iteración (Sprint Backlog)**

En el Sprint Backlog se incluye un plan para entregar el incremento de producto y alcanzar el objetivo del sprint, junto con una selección de elementos del Product Backlog elegidos para el sprint. El trabajo realizado por el Equipo durante el Sprint es



visible y se actualiza en tiempo real en el Sprint Backlog. Puedes tomar decisiones con respecto a las tareas de esta lista viendo con cuál el equipo está teniendo problemas y con cuáles no está avanzando. Es crucial destacar que el Equipo se organiza para cumplir el objetivo, actualizando las estimaciones cada vez que se dispone de nueva información y teniendo la flexibilidad de añadir o quitar actividades según sea necesario para cumplir el objetivo previsto. En resumen, sólo el equipo tiene la capacidad de realizar cambios en el Sprint Backlog.

Se muestra la auto-asignación de los miembros del equipo y la cantidad de trabajo que queda para realizar cada trabajo asociado con los objetivos/requisitos. El trabajo restante necesario para producir el producto se programa y se mide en Scrum. Dado que el camino crítico en el desarrollo ágil siempre está cambiando, este gráfico se utiliza con frecuencia en lugar de los gráficos PERT porque el gráfico PERT quedaría obsoleto todos los días. Por esta razón, es inútil emplear una tecnología que cree un modelo de camino crítico a partir de actividades. Utilizar un método para medir la rapidez con que se desarrollan las actividades a medida que avanza el proyecto es la respuesta. El plan y el ritmo de trabajo se recalculan a diario en función de la ruta crítica, que se determina según los criterios del equipo. Para alcanzar la fecha de entrega, el equipo puede proceder a acelerar o ralentizar el ritmo de trabajo.



- **BurdownChart**

BurdownChart es un cuadro de mandos que le permite realizar varios tipos de simulaciones. Puede observar cómo la adición de requisitos hace que se retrasen las fechas de entrega, cómo la eliminación de requisitos hace que se adelanten, cómo la adición de otro equipo hace que se adelanten, etc.

2.2.8.4 Ciclo de Vida

- **Pregame Phase**

Esta fase tiene las siguientes subfases:

- **Planning:** Incluye una descripción del sistema que hay que construir. Para ello, se genera la lista Product Backlog a partir de la información existente del sistema. En ella se expresan los requisitos priorizados y a partir de ella se estima el esfuerzo necesario. Continuamente se añaden al Product Backlog nuevos elementos más completos, con estimaciones más precisas y prioridades de elementos cambiantes.
- **Architecture / High level Design:** Los componentes que figuran actualmente en la Lista de productos pendientes se utilizan para planificar el diseño de alto nivel del sistema. En caso de que el producto que se vaya a crear sea una actualización de un sistema ya existente, se anotan las modificaciones necesarias para llevar a cabo los elementos de la Lista de productos pendientes y sus posibles efectos. Para evaluar los objetivos de implantación y tomar decisiones basadas en la revisión, se convoca una reunión de revisión del



diseño. Para cada versión, se crean borradores de planes para su contenido.

- **Development o Game Phase**

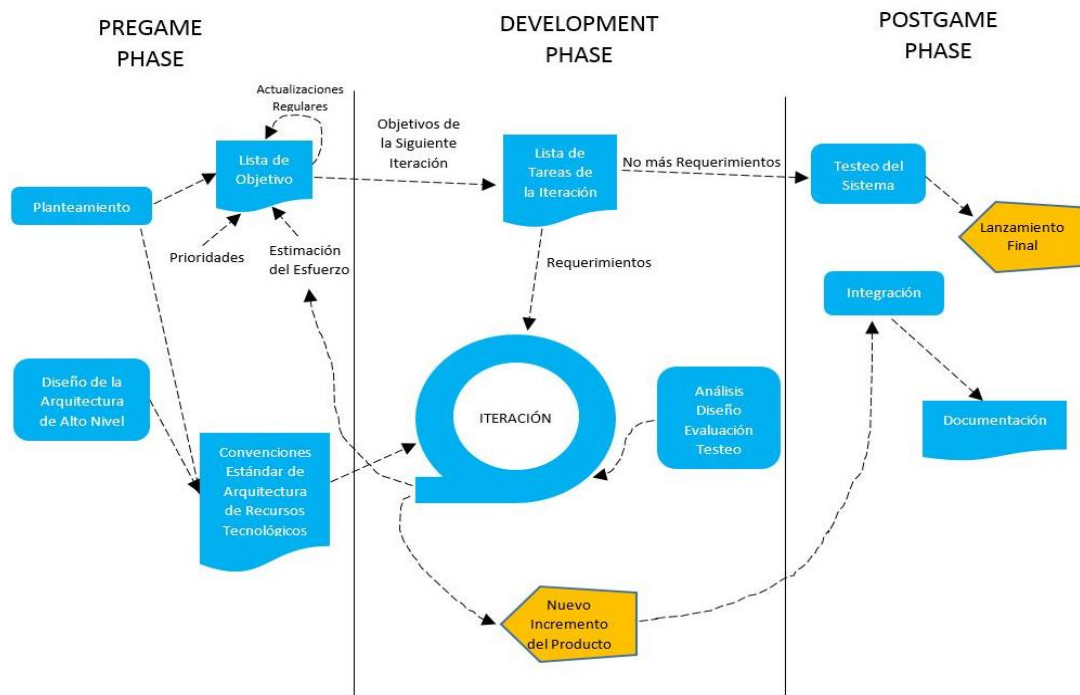
Esta es la parte ágil del proceso Scrum, cuando se anticipan los acontecimientos imprevistos. Scrum describe los procedimientos para supervisar y controlar los factores ambientales y tecnológicos, así como el enfoque de desarrollo que se ha elegido y está sujeto a cambios con el fin de evitar el caos. Para este control se utilizan sprints. El tiempo, la calidad, las necesidades, los recursos, la tecnología y las herramientas de implementación son ejemplos de variables del entorno. Scrum sugiere controlarlas continuamente para permitir una adaptación flexible a los cambios, en lugar de considerarlas al inicio del desarrollo.

- **Postgame Phase**

incluye el cierre de la liberación. Para poder avanzar en esta fase es necesario conocer las variables del entorno, como que se han cumplido los requisitos. La integración, las pruebas del sistema y la documentación se realizan ahora que el sistema está preparado para su liberación.

Figura 8

Fases Scrum



Nota: fuente (Peralta, 2003)

2.3. MARCO CONCEPTUAL

Sistema

Es el conjunto resultante de distintos componentes que trabajan juntos para guiar al sistema hacia su objetivo; un conjunto de sistemas o subsistemas más pequeños que intercambian energía para cambiarlo o lograr un objetivo.

Pensemos en la estructura familiar. Está formada, entre otras cosas, por padres e hijos, que son sistemas separados en sí mismos. Sin embargo, cuando los componentes de un fenómeno interactúan entre sí y con otros sistemas, ese fenómeno se considera un sistema (Carmona, 2011).

La conducta global del sistema se da por sub conductas que ejecutan las sus partes o sub sistemas que a su vez interactúan entre ellas.



Simulación

Es una importante herramienta dentro de la ingeniería industrial, experimentando con esta, operaciones diferentes, computacionalmente.

Dado que la simulación proporciona elementos visuales (comportamiento de cada variable) que muestran al modelizador hasta qué punto es preciso en la descripción matemática de la realidad, constituye un apoyo vital para el proceso de construcción del modelo en sus distintos métodos (etapas agregadas). Del mismo modo, la simulación es una herramienta para el análisis de sensibilidad, la validación final del modelo y la experimentación en general para la obtención de conocimientos o para la definición de técnicas de control o intervención basadas en la realidad (Parra et al., 2005, p.9).

Modelo

Un modelo es, a grandes rasgos, una representación de un sistema creada con un objetivo concreto (URQUÍA MORALED A & MARTÍN VILLALBA, 2016).

Los modelos siempre han sido aceptados por ingenieros, artistas y gestores como una técnica inestimable para presentar ideas, ayudar a la comprensión, inculcar intuición e incluso predecir nuevas formas de hacer cosas. En efecto, las teorías científicas son modelos y no hechos difíciles diseñados sobre la base de la evidencia disponible cuando se expusieron. Muchos modelos se perfeccionan a la vista de hechos adicionales y algunos quedan finalmente descartados cuando se descubren sus limitaciones.

Se puede argumentar que nuestra comprensión del mundo es un modelo complejo formado en nuestra mente por los datos que hemos acumulado. A continuación, lo guardamos de forma que se pueda utilizar para ayudarnos a



explicar y a superar las nuevas situaciones. Nuestras imaginaciones y pensamientos creativos utilizan por completo nuestro modelo intelectual del mundo (Barker, 1994).

Modelado

El modelado es una representación física y matemática de un determinado sistema, con el objetivo de obtener una similitud en su representación lo más parecido posible de sus diferentes componentes, “El modelado (...) una forma de adquirir conocimiento acerca del comportamiento de los sistemas” (URQUÍA & MARTÍN, 2016).

Maldonado & Gómez-Cruz (2010) proponen el modelado como una programación computacional con vista a aplicaciones prácticas, mientras que consideran a la simulación como un método para la implementación de un modelo en un lapso de tiempo definido. Con ayuda del modelo, es posible construir escenarios virtuales que pongan a prueba teorías sobre el funcionamiento del sistema social objetivo. También pueden tener validez cuantitativa y ser útiles como instrumentos de previsión en circunstancias específicas.

Dentro del diccionario de la Real Academia Española (RAE) el término “Modelado” lo define como “la acción y efecto de modelar.”.

Dinámica de Sistemas

La dinámica de sistemas se utiliza para investigar procesos en diversos ámbitos sociales y económicos. Gracias a su gran poder descriptivo y a su capacidad para convertirse en un modelo matemático, facilita el análisis y permite la simulación informática de los resultados. La dinámica de sistemas fue desarrollada en los años 60 por Jay W. Forrester en el Instituto Tecnológico de Massachusetts. Desde entonces, sus aplicaciones han crecido exponencialmente



en el sector empresarial, abarcando campos como la gestión de activos, los servicios financieros, la simulación de procesos, la defensa, la logística y la consultoría.

La modelización de la dinámica de sistemas es una profesión interesante que, por sus ventajas, debería ser de lectura obligatoria en todos los centros de enseñanza superior. Las competencias necesarias para ser modelador de dinámica de sistemas incluyen el pensamiento dinámico, causal, perspectivo, operativo, cíclico, cuantitativo y científico.

Las decisiones a tomar en la vida cotidiana son constantes, en las áreas de desarrollo de proyectos no es diferente, dentro de cada elemento que conforma el proyecto, el encargado o supervisor toman decisiones para el buen encaminamiento al objetivo, pero no siempre serán las más adecuadas o correctas, siendo incomprensibles los nuevos problemas que no registran antecedentes, y la falta de visión o anticipación de las consecuencias de una decisión, hacen vulnerables la correcta culminación del proyecto.

Aquí es donde entra en juego la dinámica de sistemas, que examina el sistema y utiliza un modelo matemático para explicar las causas y efectos que dan lugar a su comportamiento (Amézquita, Vergara, & Maza, 2008).

Scrum

Scrum es un marco de trabajo que facilita enormemente la creación de software complejo y su entrega en el plazo previsto. Aunque Scrum es bastante sencillo de aprender, puede llevar varios años llegar a dominarlo. Aun así, no hay por qué perder el interés porque las ventajas son considerablemente mayores que la curva de aprendizaje. Desde su introducción a principios de la década de 1990, Scrum ha crecido significativamente en popularidad. Scrum es un marco dentro



del proceso ágil de desarrollo de software que, cuando se aplica con un conjunto de reglas que deben seguir los equipos de trabajo y la utilización de roles designados, le ayudará a producir un buen software. No es una metodología en sí misma.

Los principios de la metodología de desarrollo ágil de software también se aplican al marco Scrum, que es un subconjunto de ella. Inicialmente, se establece una lista priorizada de características o funcionalidades que debe tener el producto, conocida como product backlog, y se recopila de personas relevantes como compañeros de trabajo y futuros usuarios. Aunque Scrum se utiliza sobre todo en el sector del desarrollo de software, muchas de sus ideas pueden aplicarse a la producción de cualquier otro tipo de producto o incluso a las tareas habituales del trabajo. Scrum crea procesos realmente sostenibles teniendo en cuenta la naturaleza humana (Dimes, 2015).

2.4. HIPÓTESIS Y VARIABLES

2.4.1. Hipótesis General

El modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum, mejora significativamente la planificación y desarrollo de dichos proyectos en la región de Puno.

2.4.2. Operacionalización de Variables

(X) Variable Independiente

-Modelo de Dinámica de Sistemas

(Y) Variable Dependiente

-Planificación y desarrollo de proyectos de software que aplican el marco de trabajo Scrum.

Tabla 3

Operacionalización de Variables

VARIABLE INDEPENDIENTE	DIMENSIONES	INDICADORES	ÍTEMS
Modelo de dinámica de sistemas	Conceptualización del sistema marco de trabajo Scrum, usando diagramas causales.	Diagramas causales del sistema.	Número de diagramas causales.
	Construcción del modelo mediante diagramas Forrester.	Diagramas Forrester.	Número de diagramas Forrester.
VARIABLE DEPENDIENTE	DIMENSIONES	INDICADORES	INSTRUMENTO
Planificación y Desarrollo de proyectos de software que aplican Scrum.	Planificación	-Toma de decisiones.	Pre Test y Post Test
		-Experiencia	
		-Estimación	
	Desarrollo	-Predictibilidad	
		-Propuesta	
		-Marco de trabajo Scrum.	
		-Dominio	

Nota: elaboración propia



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. METODOLOGÍA DE LA INVESTIGACIÓN

De las dos metodologías de investigación cualitativa y cuantitativa, direccionalaremos el presente proyecto con la cuantitativa, puesto que esta metodología busca cuantificar los datos y aplicar alguna forma de análisis estadístico, usando magnitudes numéricas que favorecen a la posibilidad de ser tratadas mediante herramientas del campo de la estadística.

La investigación cuantitativa nos proporciona control sobre los fenómenos, un punto de vista sobre el recuento y sus magnitudes, y la capacidad de generalizar los resultados de forma más amplia. Además, nos brinda una fantástica oportunidad de replicar el fenómeno, se centra en aspectos concretos del mismo y facilita la comparación de las conclusiones de estudios relacionados (Hernández, Fernández, Baptista, Méndez, & Mendoza, 2014b).

Dado que el presente proyecto está diseñado para mostrar y crecer de forma objetiva, se ajusta a la técnica de investigación cuantitativa.

3.2. DISEÑO DE LA INVESTIGACIÓN

El Diseño de investigación es cuasi experimental porque permite establecer una relación causal entre una o más variables denominada dependiente (Y) y otras variables independientes (X) en una situación estrictamente controlada.

Los diseños cuasi experimentales son esquemas de investigación no aleatorios. Dado la no aleatorización, el establecer de forma exacta la equivalencia inicial del grupo de investigación, es posible mediante la selección y autoselección, siendo también posible el diseño de comparación de grupos.



Dado que la palabra «cuasi» significa «casi», podríamos definir un diseño cuasi experimental como aquel que no es totalmente experimental. Este tipo de ensayo no alcanza el umbral experimental porque no puede garantizar la equivalencia inicial de los grupos experimental y de control, o aleatorización. Dado que en este tipo de estudios se seleccionan dos grupos que ya están integrados, ni las unidades de análisis ni el grupo experimental se asignan al azar (Serrano et al., 2011 p.24).

Dentro del diseño cuasiexperimental los sujetos no son asignados al azar a los grupos, estos grupos ya están formados previos al experimento, sin que esta haya influido de ninguna manera en su formación.

3.3. POBLACIÓN Y MUESTRA

3.3.1. Población

La población que se ha de considerar para la presente investigación está conformada por el personal desarrollador de un total de tres proyectos de desarrollo de software que aplican el marco de trabajo Scrum en la región de Puno.

El total de proyectos que se desarrolló aplicando el marco de trabajo Scrum, esto para inicios del año 2017, fueron un total de tres proyectos, los mismos que se consideran como población en el presente proyecto.

Verificando los diferentes antecedentes, la gran mayoría de estos hacen la validación de sus modelos propuestos tomando como muestra mínima 01 proyecto real, logrando satisfactoriamente los objetivos propuestos, esto nos indica que, si se toma más de 01 muestra en específico para este tipo de proyectos, la confianza del modelo diseñado estará lejos de la imperfección.



3.3.2. Muestra

La muestra que se tomó para la presente investigación es el total de la población.

Esta muestra se presenta a continuación:

- SISTEMA DE PLANIFICACIÓN DE RECURSOS, Para la empresa PERUVIAN COMPUTER TECHNOLOGIES E.I.R.L. El equipo de Trabajo: 3 personas, 1 Scrum Master y 2 Team, 1 Nivel Junior y 2 Nivel Senior.
- APLICACIÓN MULTIPLATAFORMA PANDA COMPUTER, desarrollado para la empresa PANDA COMPUTER S.R.L. El equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 1 Nivel señor y 1 Nivel Junior.
- APLICATIVO WEB ALPAMAYO CLIENT. Proyecto ejecutado para la empresa PRODUCTOS Y SERVICIOS INDUSTRIALES ALPAMAYO S.A.C. El equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 2 Nivel Junior.

3.4. MÉTODO DE RECOPIACIÓN DE DATOS

Se hizo uso de la siguiente:

3.4.1. Encuesta

El investigador elaboró un conjunto de preguntas para las encuestas previas y posteriores a la prueba, que se distribuyeron a los participantes en el estudio y permitieron poner a prueba el modelo dinámico para la validación correspondiente. El instrumento de encuesta utilizado para estas evaluaciones se muestra en el «Anexo A», y se ofrecerá una explicación más detallada del mismo en la prueba de hipótesis.



3.5. MÉTODO DE TRATAMIENTO DE DATOS

Para verificar y comprobar la hipótesis, se aplicó la investigación cuasi experimental para medir el efecto de la variable independiente sobre la dependiente, cumpliendo las siguientes tareas:

- Recopilación de datos
- Tabulación de Datos
- Análisis de Datos
- Consistencia de Datos
- La interpretación de los datos
- Por último, validación de la Hipótesis mediante la prueba de Hipótesis.

3.6. MATERIAL EXPERIMENTAL

El modelado se llevó a cabo utilizando Vensim PLE® 6.3, que permite el desarrollo de modelos de simulación de Ventana Systems, así como la conceptualización, construcción, simulación, análisis, optimización y desarrollo de modelos de sistemas dinámicos complejos. Vensim es increíblemente rápido y eficaz. La pantalla de trabajo de Vensim presenta un elemento visual relativamente sencillo que simplifica el uso de las herramientas para la construcción de modelos, la simulación y el análisis de resultados, a la vez que produce rápidamente resultados de alta calidad.

Con Vensim, crear modelos de simulación es sencillo. Se parte de un diagrama causal, que se convierte en un diagrama de flujo. Observando la evolución gráfica de las variables del sistema, se puede observar lo que ocurre en el modelo y entenderlo como un reflejo de la realidad en cada momento.



3.7. DISEÑO ESTADÍSTICO PARA LA PRUEBA DE HIPÓTESIS

Se utilizaron los cinco procedimientos siguientes para utilizar el estadístico de distribución t de Student para analizar los resultados de la encuesta de satisfacción antes y después de la prueba con el fin de comprobar la hipótesis:

Paso 1: Plantear Hipótesis Nula (Ho) e Hipótesis Alternativa (Hi).

La Hipótesis Alternativa plantea matemáticamente lo que se quiere demostrar y la Hipótesis Nula plantea exactamente lo contrario.

Paso 2: Determinar Nivel de Significancia. (Rango de aceptación de hipótesis alternativa) α

$$\alpha = 0.05$$

En la investigación se utilizó el nivel confiable, que equivale al 95 % de aceptación y 5 % de error.

Paso 3: Se calcula la diferencia de media y la desviación estándar para grupos relacionados a partir de la muestra.

$$\bar{X}_d = \sum_1^n \frac{X_{i1} - X_{i2}}{n} \dots\dots\dots (Ec. 1)$$

$$S_d = \sqrt{\frac{\sum_1^n (d_i - \bar{X}_d)^2}{n-1}} \dots\dots\dots (Ec. 2)$$

Paso 4: Se aplica la distribución T de Student para calcular la probabilidad de error (P) por medio de la fórmula:

$$t = \frac{\bar{X}_d}{\frac{S_d}{\sqrt{n}}} \dots\dots\dots (Ec. 3)$$

También se determina grado de libertad:

$$Gl = n - 1 \dots\dots\dots (Ec. 4)$$



Media aritmética de las diferencias: \bar{X}_d

Desviación estándar de las diferencias: S_d

Numero de sujetos de la muestra: n

$t =$ Distribución t

Paso 5: Regla de decisión

En base a la evidencia disponible se acepta o se rechaza la hipótesis alternativa.

Si la probabilidad de error (P) es mayor que el nivel de significancia: Se rechaza la hipótesis alterna.

Si la probabilidad de error (P) es menor que el nivel de significancia: se acepta hipótesis alternativa.



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Las distintas fases a seguir de la Metodología de Dinámica de Sistemas son tres, las mismas que con mayor detalle se nos muestra en la Base Teórica del presente proyecto, en el cual se nos indica que la Conceptualización en la primera fase, Formulación es la segunda fase y como la tercera y última fase tenemos el Análisis-Evaluación, en los siguientes se nos muestra el desarrollo de cada una de estas fases.

Es importante recordar que este modelo está pensado para facilitar la toma de decisiones y no para ser utilizado como estimación. La aplicación de la Dinámica de Sistemas como herramienta de predicción precisa está fuera de lugar. Los resultados de un modelo de dinámica de sistemas no pueden juzgarse en función de lo bien que predice el futuro, sino que el valor de la modelización de dinámica de sistemas reside en el examen de los comportamientos de retroalimentación de un sistema. En lugar de utilizarse para la estimación directa en esta situación, el modelo se entiende mejor como una herramienta de aprendizaje y política.

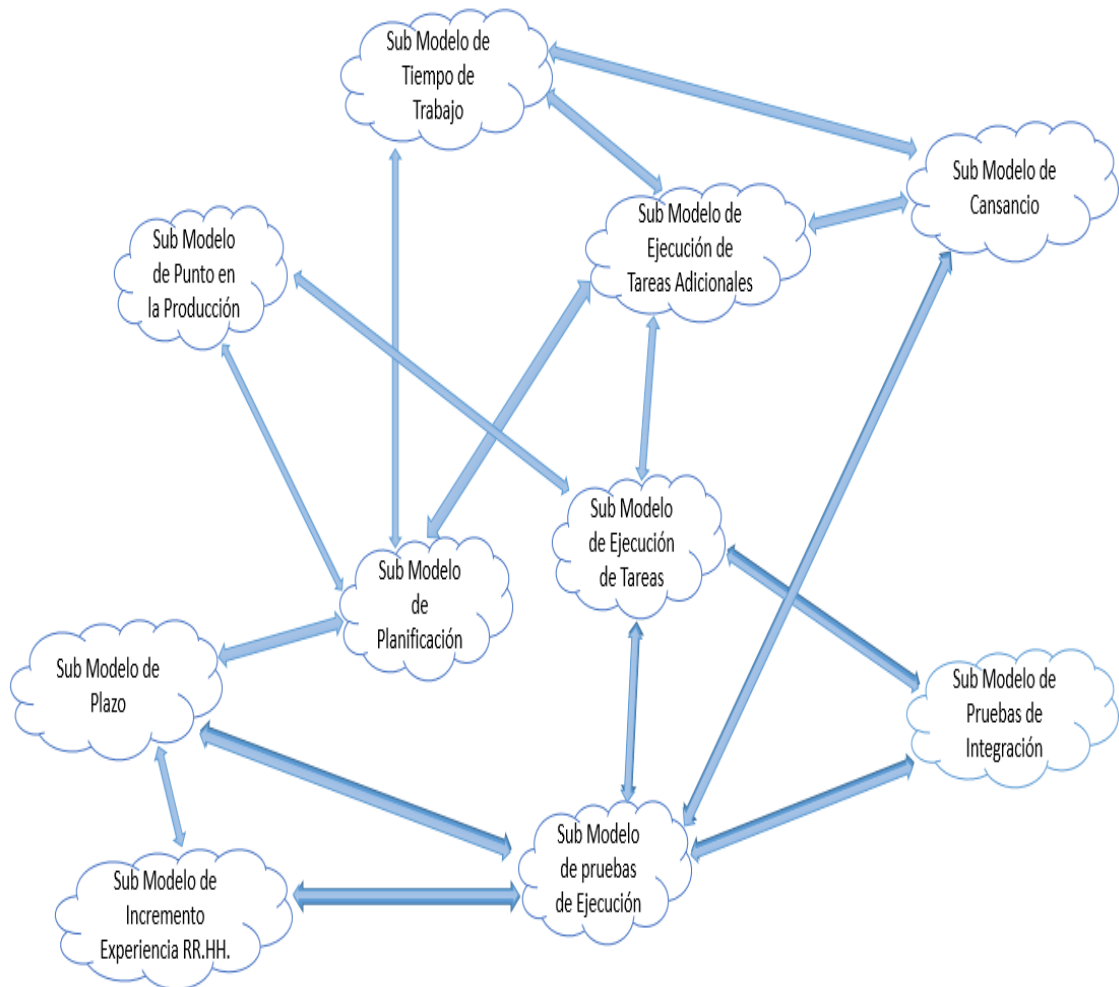
4.1. FASE DE CONCEPTUALIZACIÓN

Se trata del método inicial de diseño del modelo mediante la creación de diagramas de influencia, que permiten identificar los arquetipos sistémicos que componen el sistema y comprender sus estructuras y las fuerzas que actúan en su interior.

Por el mismo hecho de que el modelo que se diseñó tiene un tamaño considerable, se muestra en la siguiente figura dividida en diez sub diagramas de influencia o sub modelos, además que esto nos facilitará con la lectura del mismo modelo. Los sub modelos no están aisladas unas de otras por ello están asociadas entre las mismas.

Figura 9

Diseño del modelo y sus diez sub modelos



Nota: elaboración propia



Los sub modelos son los siguientes:

4.1.1. Diagrama de Influencia del Sub Modelo de Planificación

En este Sub Modelo se da prácticamente el inicio del proceso de simulación, ya que aquí valores e inicializadores son definidos.

Cada Sprints tiene un momento de inicio y la duración en su ejecución, estas están siendo representadas por variables del tipo Lookup las cuales son TiempoInicioSprint y DuracionPorSprint, y que sus valores se determinan en el Sub Modelo.

También tenemos las variables VelocidadPromedioSprints y PuntosPorSprints del tipo Lookup, la primera toma valores de velocidad ideal con la que se ejecutan los diferentes Sprints esto en función a los puntos respectivos, en el caso de la segunda variable, son puntos asignados a cada Sprint la misma que representa la cantidad que deberá ser desarrollada en cada Sprint.

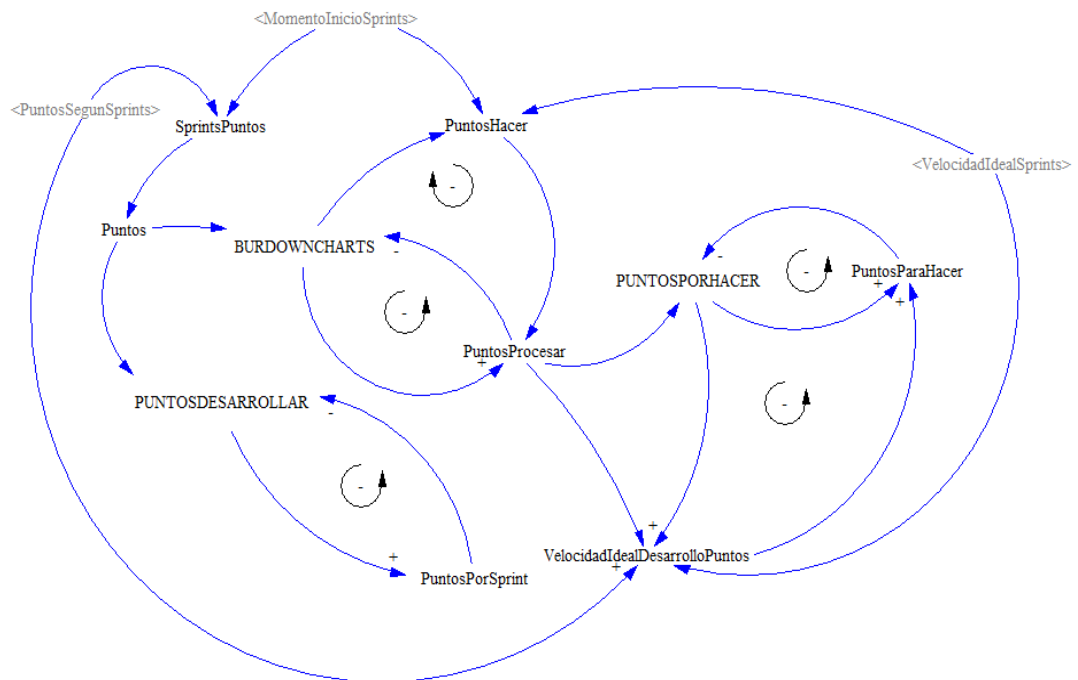
Se definió dos variables de Nivel, el primero SPRINTSPLANIFICADOS que almacena valores del total de tiempo en el que cada Sprint se ejecuta y se muestra la comparación entre ellas, la segunda variable TIEMPOTOTALSPRINTSPLANIFICADOS expresa valores del total de tiempo que se toma cada Sprint en su ejecución consecutivamente uno tras otro.

El Sub Modelo obtiene las variables auxiliares MomentoInicioSprints y VelocidadIdealSprint esto a través de la variable de Nivel TIEMPOTOTALSPRINTSPLANIFICADOS, las que también interactúan con otros Sub Modelos.

Nivel es PUNTOSPORHACER esta almacena valores que muestran puntos que se están desarrollando según el tiempo vaya avanzando.

Figura 11

Diagrama de influencia del sub modelo de puntos en la producción



Nota: elaboración propia

4.1.3. Diagrama de Influencia del Sub Modelo de Ejecución de Tareas

Tenemos la dinámica de ejecución de Tareas planificadas, las que se están desarrollando de acuerdo al tiempo, tareas con errores y tareas codificadas.

El número total de tareas por punto es una de las variables que interactúa en este Sub Modelo, para poder almacenar en otra variable el valor del total de puntos y tareas a desarrollar por cada Sprint. Así pues, tenemos la variable de Nivel TAREASPLANIFICADAS.

La variable de nivel TAREASPENDIENTESCODIF representa las tareas que ya fueron planificadas y según el tiempo están en espera de ser codificadas. Siendo también influido por la variable TareasConErrARecod que desde la

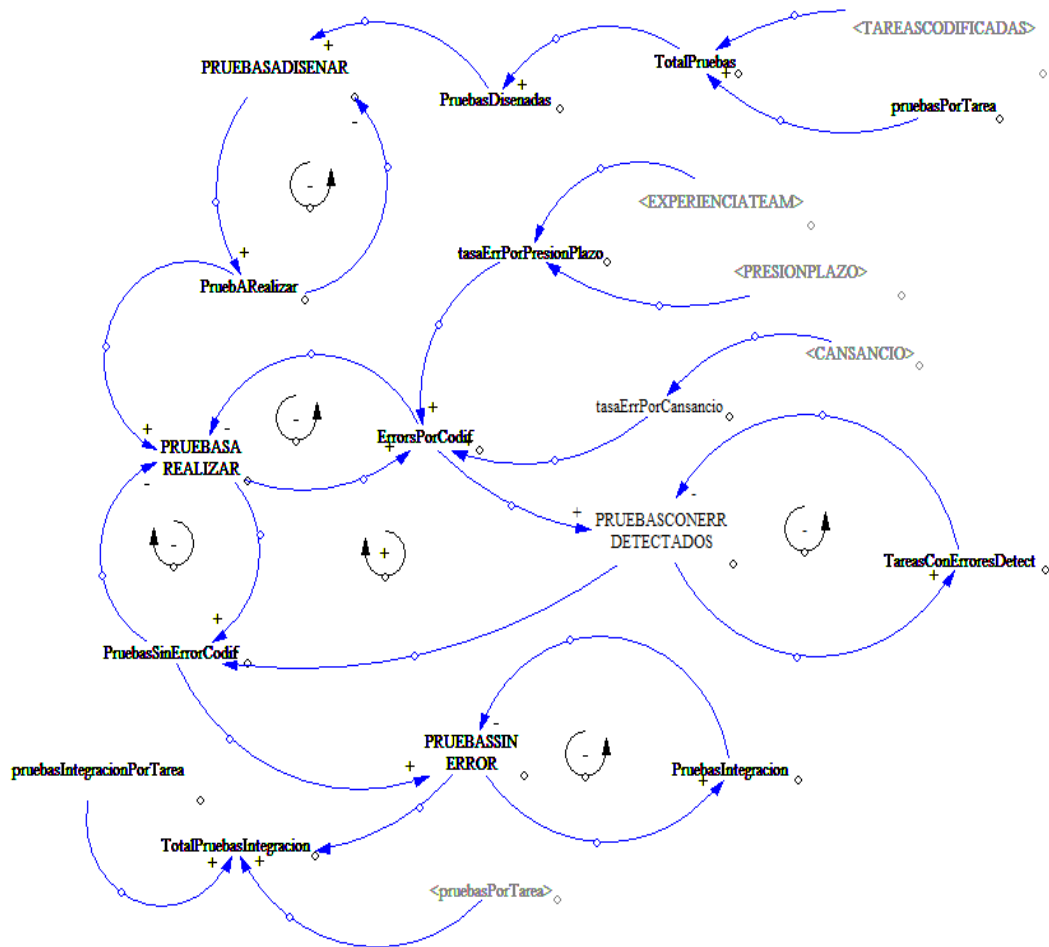


Se inicia con el total de pruebas que se tendrán que hacer representada con la variable de nivel PRUEBASADISENAR, datos que pasan a la siguiente variable de nivel PRUEBASAREALIZAR desde la cual obtenemos valores para las variables definidas como pruebas con error detectados y pruebas sin error.

Las tasas de errores cumplen los roles más importantes en este Sub Modelo ya que interactúa con la variable de flujo ErrorPorCodif para luego obtener los resultados.

Figura 13

Diagrama de influencia del sub modelo de pruebas de tareas



Nota: elaboración propia

4.1.5. Diagrama de Influencia del Sub Modelo de Presión del Equipo

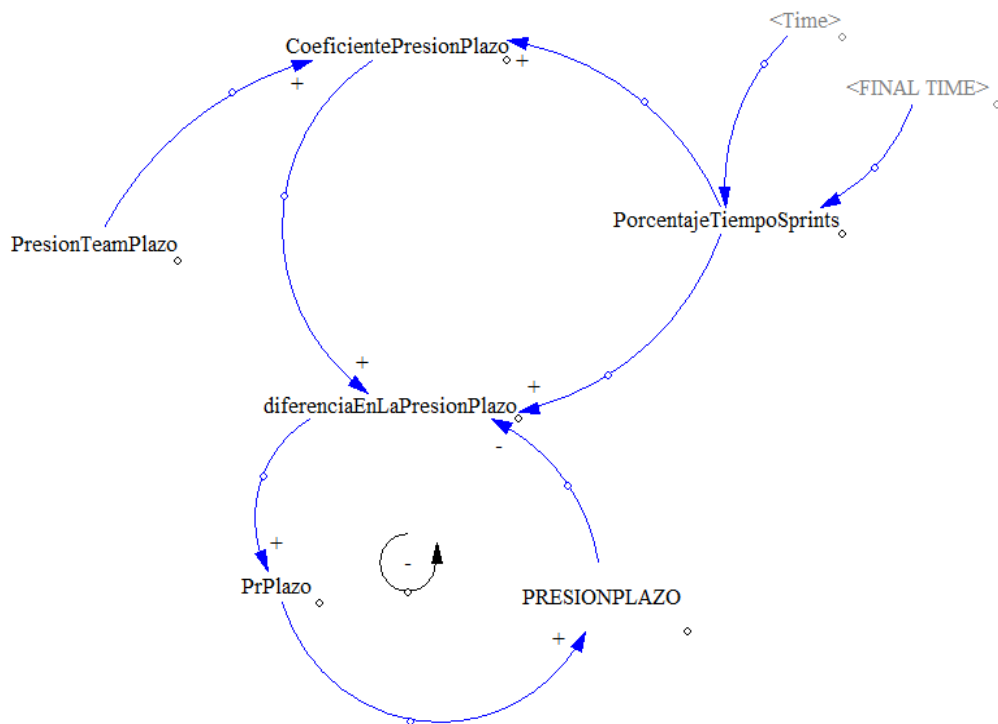
Según el tiempo va transcurriendo y el plazo de entrega final se va acervando, se genera una presión en el equipo de trabajo, es esta dinámica la que el Sub Modelo representa.

PresionTeamPlazo es la variable de tipo lookup, que da inicio al Sub Modelo conjuntamente con las variables de FINAL TIME y Time.

La única variable de Nivel que se tiene es PRESIONPLAZO que almacena información de la presión en el equipo esto según el tiempo este avanzando.

Figura 14

Diagrama de influencia del sub modelo de presión del equipo



Nota: Elaboración propia

4.1.6. Diagrama de Influencia del Sub Modelo de Tareas Adicionales

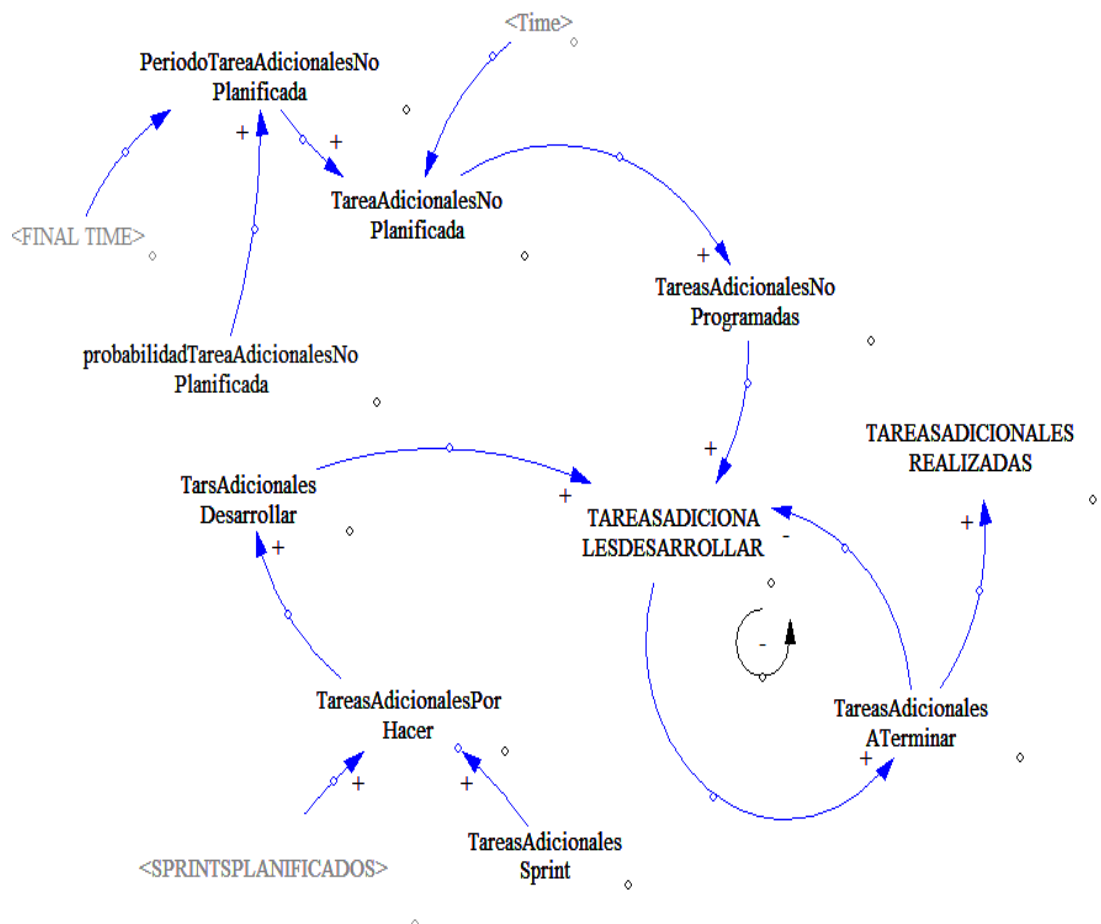
Al ejecutar el desarrollo de los Sprints, tenemos dos principales casos en las que las tareas aumentan o se adicionan, el primero por error que al ser detectada se adiciona como una nueva tarea a rehacer, y la segunda es la falta de tareas no contempladas en la planificación y que se detecta en plena ejecución de los Sprints. Estas tareas por error o las no planificadas serán las Tareas Adicionales.

Damos inicio este Sub Modelo con las tareas que se adicionaran las que no se planificaron por Sprint, la que contendrá esta información será la variable de tipo Lookup TareasAdicionalesSprint. La probabilidad de tareas que no fueron planificadas también da inicio al Sub Modelo, con un valor en porcentaje que es necesario definir según experiencias o datos precisos.

Para el proceso y almacenamiento de la información a lo largo del proyecto, tenemos las variables de nivel TAREASADICIONALESDESARROLLAR Y TAREASADICIONALESREALIZADAS.

Figura 15

Diagrama de influencia del sub modelo de tareas adicionales



Nota: elaboración propia

4.1.7. Diagrama de Influencia del Sub Modelo de Pruebas de Integración

Después de las pruebas que resultaron sin error es que tenemos el total de pruebas de integración, la misma que da inicio como TotalPruebasIntegracion al presente sub modelo.

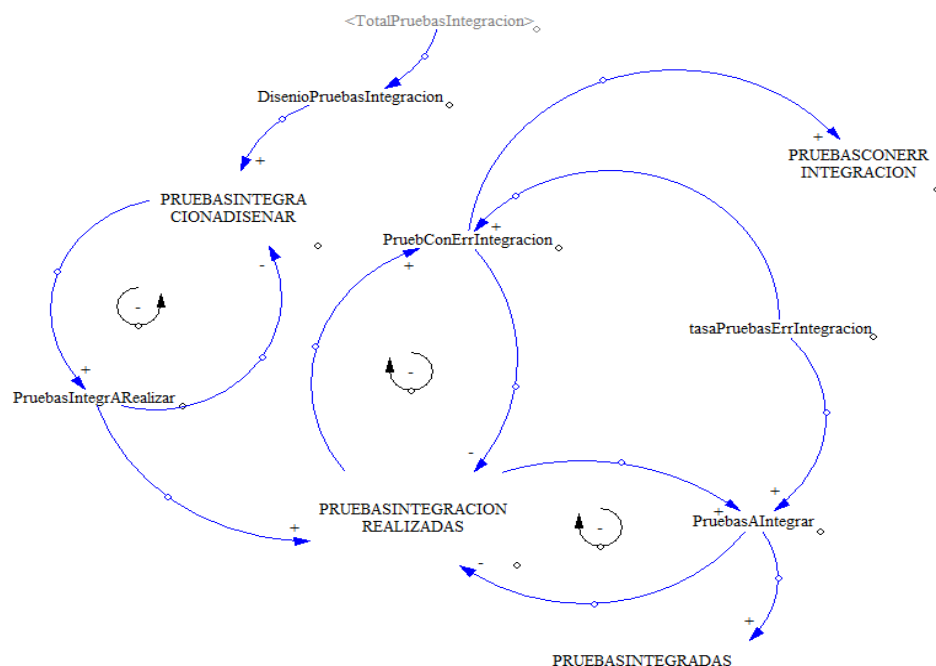
La variable encargada de hacer la planificación de las Pruebas de Integración que se realizaran es la variable de Nivel

PRUEBASINTEGRACIONADISENAR, posteriormente las pruebas con errores y las correctas son obtenidas a partir de la variable PRUEBASINTEGRACIONREALIZADAS que interactúa con la una tasa de error ya definida.

La variable PRUEBASINTEGRADAS es la cual almacena información de las pruebas que dieron como resultado correcto sin algún error, por otro lado, la variable PRUEBASCONERRINTEGRACION almacena información de las pruebas que resultaron con error, la misma que se reprogramara y pasara al Sub Modelo de Tareas Adicionales.

Figura 16

Diagrama de influencia del sub modelo de pruebas de integración



Nota: elaboración propia

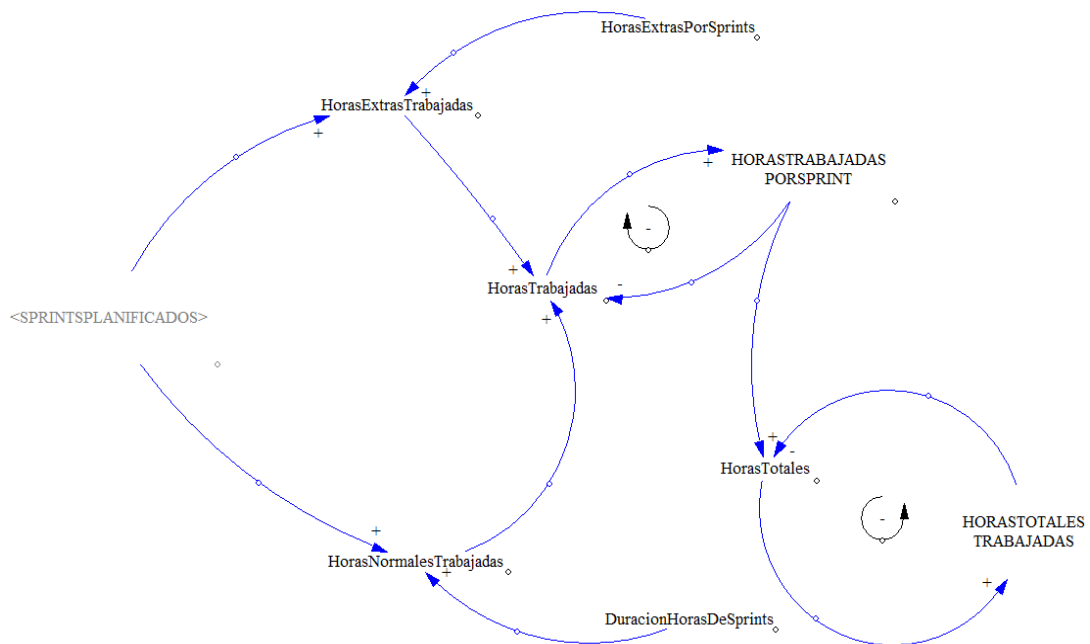
4.1.8. Diagrama de Influencia del Sub Modelo de Tiempo de Trabajo

Según el total de tareas y Sprint planificados que fueron trabajados, tendremos el tiempo total en la que se desarrolla el proyecto.

Las variables del tipo Lookup HorasExtrasPorSprints y DuracionHorasDeSprints son las que dan inicio a este sub Modelo, interactuando con SPRINTSPLANIFICADOS, de lo cual se obtendrán datos para la variable de nivel HORASTRABAJADASPORSPRINT y HORASTOTALESTRABAJADAS.

Figura 17

Diagrama de influencia del sub modelo de tiempo de trabajo



Nota: elaboración propia

4.1.9. Diagrama de Influencia del Sub Modelo de Cansancio

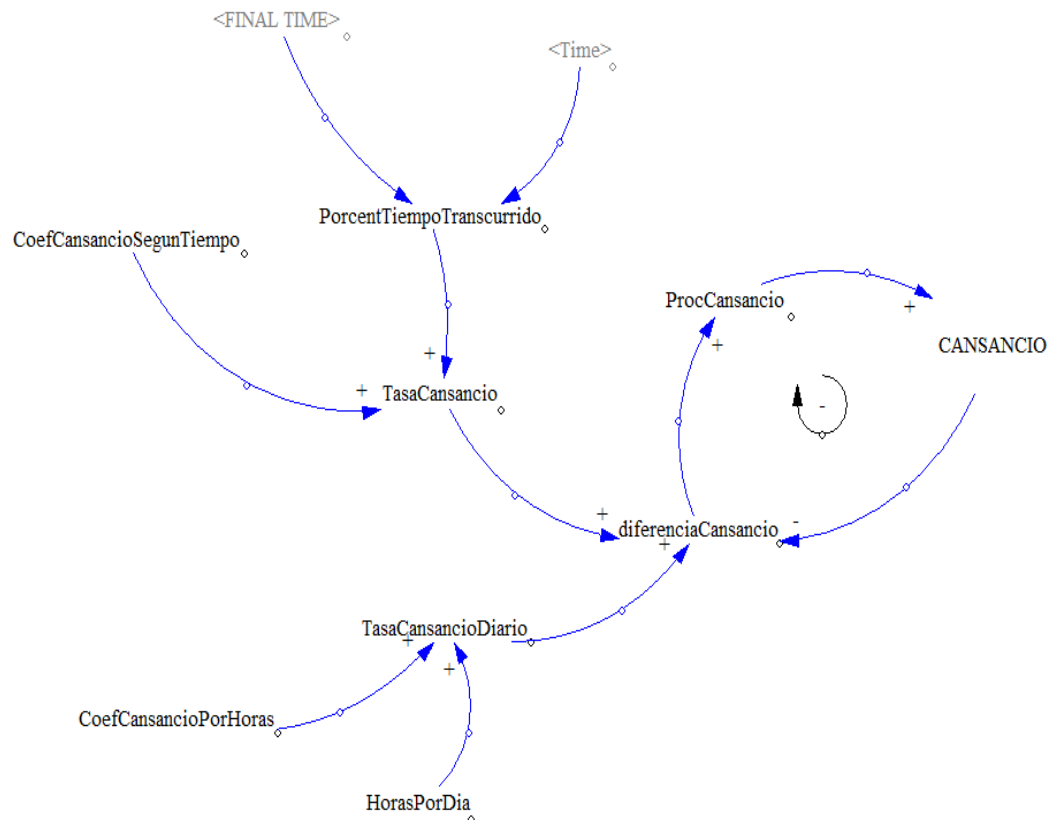
El equipo de trabajo a lo largo del proyecto acumula cansancio y es justamente ello lo que simula el Sub Modelo.

Para iniciar el sub modelo, tenemos dos variables del tipo Lookup las cuales son CoefCansancioSegunTiempo que según pasa el tiempo se eleva el porcentaje de cansancio, y CoefCansancioPorHoras que según las horas trabajadas por día toma el valor correspondiente, estas variables posteriormente generan la Tasa de Cansancio y la Tasa de Cansancio Diario.

El cansancio acumulado de grupo de trabajo teniendo en cuenta el tiempo se almacena en la variable de Nivel CANSANCIO.

Figura 18

Diagrama de influencia del sub modelo de cansancio



Nota: elaboración propia

4.1.10. Diagrama de Influencia del Sub Modelo de Incremento de Experiencia de Equipo

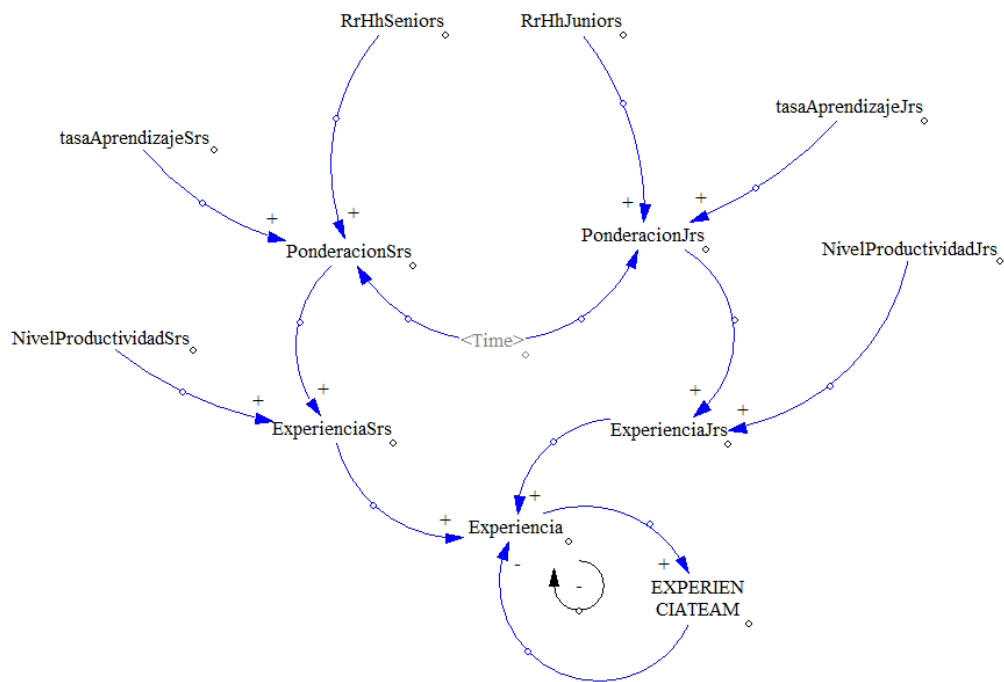
Recursos Humanos es el mismo equipo de trabajo conformada por Recursos Humanos Seniors y Recursos Humanos Juniors, estos según el proyecto van ganando experiencia y es justo esto lo que modela el Sub Modelo.

Para poder dar inicio en este Sub modelo se tiene las Tasa de Aprendizaje Señor y Tasa de Aprendizaje Junior, conllevando los datos obtenidos a la Ponderación Señor y Junior.

Los niveles de productividad tanto Señor y Junior son también variables parte del Sub Modelo, que hacen que la variable de Nivel EXPERIENCIATEAM obtenga y almacene datos del cansancio del grupo de trabajo, esto contemplando todo el tiempo en el que se ejecutan lo Sprints.

Figura 19

Diagrama de influencia del sub modelo de incremento de experiencia de equipo



Nota: elaboración propia

4.2. FASE DE FORMULACIÓN

Desarrollado los diagramas causales en la fase anterior, transformaremos estas en Diagramas Forrester con el software Vensim® PLE apoyándonos con la creación de modelos de simulación, en esta fase también se identifica las variables, la definición de su clasificación en Nivel, Flujo y Auxiliares, se continua con la definición de las ecuaciones con las que trabaja el modelo.

Al igual que con los Diagramas de Influencia, tenemos sub modelos de cada uno de los Diagramas de Forrester y estas serán descritas y son las siguientes:

4.2.1. Sub Modelo de Planificación

4.2.1.1. Clasificación de Variables

En este Sub Modelo tenemos diferentes variables que hacen la dinámica, las cuales serán brevemente descritas a continuación según el tipo de variable:

- **LookUp**

DuracionPorSprints: Es el tiempo total que cada Sprint lleva su desarrollo. Como entrada X tenemos el tiempo total requerido por Sprint, y como salida Y el número de Sprint al cual corresponde el tiempo requerido.

TiempoInicioSprints: Momento del tiempo en el que da inicio un Sprint determinado. Como entrada X el número de Sprint, y como salida Y el instante exacto en el tiempo que da inicio el Sprint X.

TiempoFinSprint: Momento del tiempo en el que finaliza el desarrollo de un Sprint determinado. Como entrada X tenemos el momento en el tiempo donde finaliza el Sprint en su ejecución, y como salida Y el número de Sprint correspondiente a X.

PuntosPorSprints: puntos las cuales tienen que ser completadas al culminar el desarrollo de cada Sprint al cual fue asignado. Como entrada X tenemos el número de Sprint al cual se asignan los puntos, y como salida Y tenemos los puntos.

VelocidadPromedioSprints: La velocidad en promedio a la cual se desarrolla un determinado Sprint. Como entrada X tenemos el



número de Sprint, y como salida Y tenemos el promedio de velocidad.

- **Auxiliar**

MomentoInicioSprints: Almacena el momento en el tiempo que da inicio cada Sprint, esto con ceros y unos según la información que la variable TiempoInicioSprints y la variable de Nivel TIEMPOTOTALSPRINTS proporcionen, aplicando dos funciones la primera es lookup backward que busca el valor de TIEMPOTOTALSPRINT y la busca en TiempoInicioSprints al encontrarla le agrega 2 y hace la comparación con el tiempo en ese instante, para tener resultados tenemos la función if then else que devuelve 1 si la comparación es correcta caso contrario cero.

SprintsVigente: Almacena la información de la comparación de tiempo de duración de cada Sprint con los otros Sprints. La función Lookup Forward es la que nos permite hacer esta dinámica, busca con el tiempo en curso en la variable DuracionPorSprints el Sprint que se está trabajado.

PuntosSegunSprints: Representa el total de puntos que en un lapso de tiempo determinado por los Sprint deberán ser completados. La función que se hace uso es Lookup Forward, trabajando con las variables TIEMPOTOTALSPRINTS con la cual buscamos el número de Sprint en curso para obtener de la variable PuntosPorSprints los puntos que por todo el lapso de tiempo el Sprint al completar su desarrollo deberá tener.



VelocidadIdealSprints: Almacena la información de la velocidad ideal que cada Sprint en su tiempo de desarrollo tiene. La función Lookup Forward tomara los datos de TIEMPOTOTALSPRINTS para buscarla en VelocidadPromedioSprints la velocidad promedio que fue definido.

- **Flujo**

TiempoTotal: Esta variable regula la variable de Nivel TIEMPOTOTALSPRINTS disminuyendo el valor que asume con el valor anterior con respecto a tiempo.

SprintsATerminar: Procesa datos que vienen de la variable SprintsVigente sacando la diferencia de este y SPRINTSPLANIFICADOS para luego transferir el resultado a SPRINTSPLANIFICADOS.

- **Nivel**

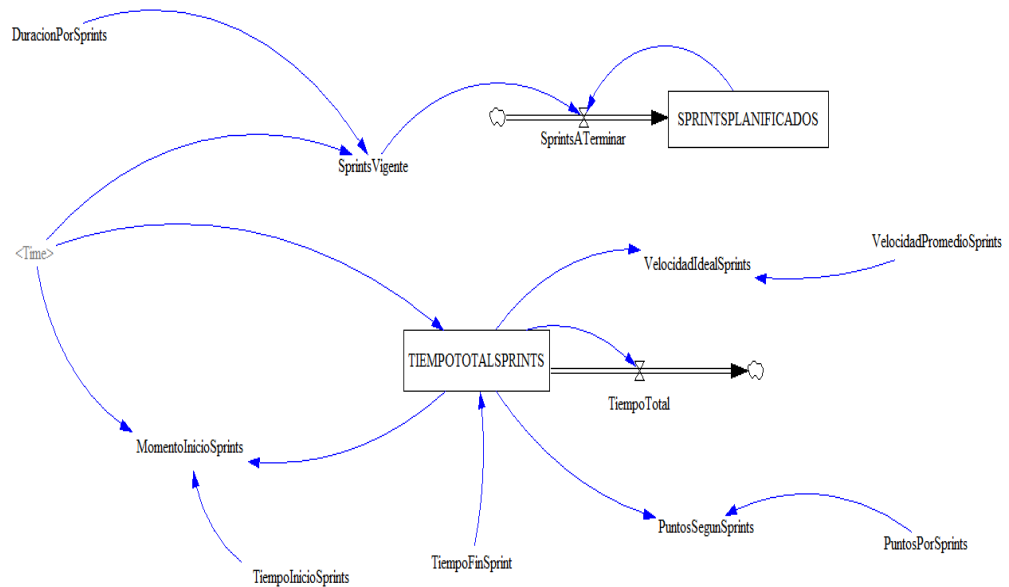
SPRINTSPLANIFICADOS: Toma como valor inicial cero. Su valor viene del flujo SprintsATerminar. Almacena información de los Sprints Planificados, y una comparación entre las mismas según el tiempo.

TIEMPOTOTALSPRINTS: Valor inicial cero. Aquí trabajamos con la función Lookup Forward la cual toma la variable Time para buscar en TiempoFinSprint el tiempo fin de cada Sprint y obtener el número de Sprint, y almacenar la información del número de Sprint y repetirlo tanto tiempo este programado dicho Sprint para su ejecución y culminación.

4.2.1.2. Diagrama de Forrester

Figura 20

Diagrama de forrester del sub modelo de planificación



Nota: elaboración propia

4.2.1.3. Ecuaciones del Sub Modelo

Las ecuaciones con las que el sub modelo trabaja dentro del software Vensim son descritas a continuación:

- **Auxiliar**

MomentoInicioSprints:

MomentoInicioSprints=if then else(lookup
backward(TiempoInicioSprints, TIEMPOTOTALSPRINTS)+2=Time, 1, 0)

Units: sprint

SprintsVigente:

SprintsVigente=

LOOKUP FORWARD(DuracionPorSprints, Time)



Units: sprint [0,80]

PuntosSegunSprints:

PuntosSegunSprints= LOOKUP FORWARD(PuntosPorSprints,
TIEMPOTOTALSPRINTS)

Units: Puntos

VelocidadIdealSprints:

VelocidadIdealSprints= LOOKUP
FORWARD(VelocidadPromedioSprints,
TIEMPOTOTALSPRINTS)

Units: Horas

▪ **Flujo**

TiempoTotal:

TiempoTotal= TIEMPOTOTALSPRINTS

Units: Horas

SprintsATerminar:

SprintsATerminar= SprintsVigente-SPRINTSPLANIFICADOS

Units: sprint

▪ **Nivel**

SPRINTSPLANIFICADOS:

SPRINTSPLANIFICADOS= INTEG (SprintsATerminar,0)

Units: sprint

TIEMPOTOTALSPRINTS:



TIEMPOTOTALSPRINTS=INTEG(LOOKUP

FORWARD(TiempoFinSprint, Time)-TiempoTotal,0)

Units: Horas

4.2.2. Sub Modelo de Puntos en la Producción

4.2.2.1. Clasificación de Variables

En presente Sub Modelo tenemos diferentes variables que hacen la dinámica, las mismas que serán descritas a continuación según su clasificación:

- **Auxiliar**

SprintsPuntos: Lo que almacena es el total de puntos asignados para su desarrollo por cada Sprint. Tenemos la función if then else que verifica que MomentoInicioSprints sea igual a 1, caso sea verdadera esta comparación en uno o varios momentos en el tiempo, almacenará el valor de PuntosSegunSprints, caso contrario cero.

PuntosHacer: Procesa y prepara datos para la variable de Nivel PUNTOSHECHAS, influida por los MomentoInicioSprints, VelocidadIdealSprints y BURDOWNCHARTS, las cuales son trabajadas dentro de la actual variable con tres niveles de la función if then else.

- **Flujo**

Puntos: Toma valores de SprintsPuntos según el tiempo transcurre, y pasara los datos a dos variables de Nivel y las modula.



PuntosProcesar: En esta variable se obtiene la diferencia de la variable de nivel BURDOWNCHARTS y la variable de auxiliar PuntosHacer, y esta es transferida a dos nuevas variables.

PuntosParaHacer: Dentro de ella se hace produce una resta entre la variable PUNTOSHECHAS y VelocidadIdealDesarrolloPuntos, y este resultado según el tiempo modula disminuyendo a PUNTOSHECHAS.

PuntosPorSprint: Toma el valor de la variable PUNTOSDESARROLLAR y la disminuye en el siguiente tiempo con esta.

- **Nivel**

BURDOWNCHARTS: Variable que muestra cómo se debería desarrollar cada punto, esto en cada Sprint en función al tiempo. A esta se le disminuye los valores de la variable PuntosProcesar, y por ello se logra el Burdown Chart.

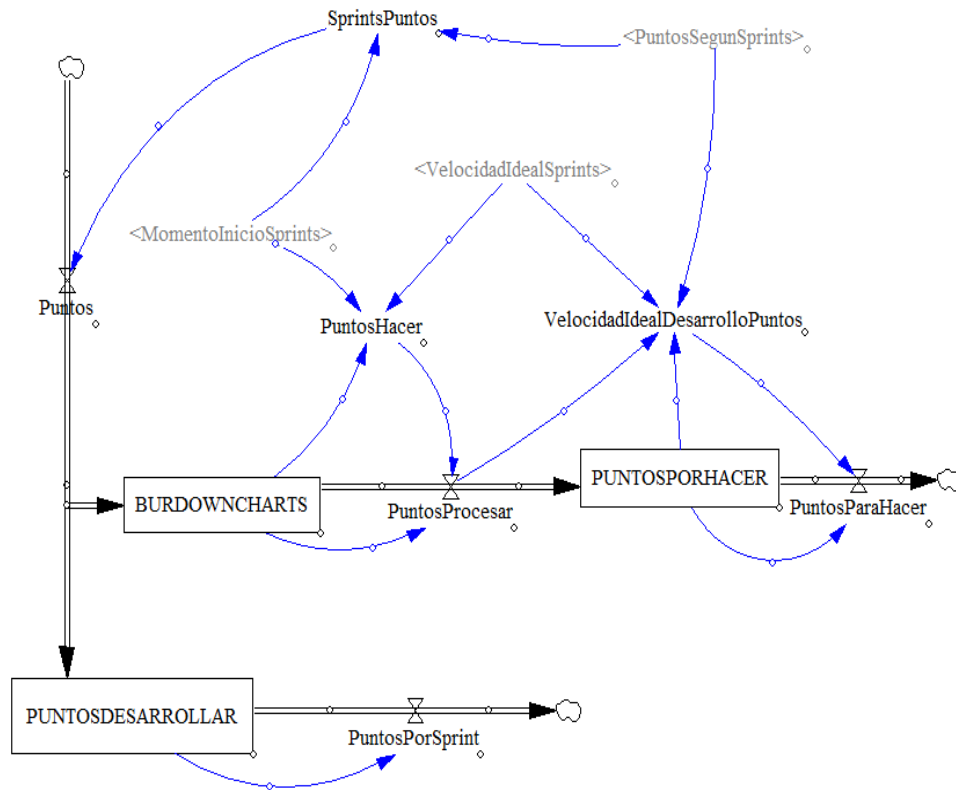
PUNTOSHECHOS: Almacena información de los puntos que progresivamente según el tiempo se desarrolló. La variable PuntosProcesar ingresa los datos y la encargada de modularla es la variable PuntosParaHacer.

PUNTOSDESARROLLAR: Almacena información del total de puntos que por cada Sprint tiene que ser completado, mostrando esto al principio de cada Sprint. La entrada de datos se da por la variable Puntos y PuntosPorSprint es la que actúa sobre la anterior variable para obtener la correcta dinámica.

4.2.2.2. Diagrama de Forrester

Figura 21

Diagrama de forrester del sub modelo de puntos en la producción



Nota: elaboración propia

4.2.2.3. Ecuaciones del Sub Modelo

Las ecuaciones con las que el sub modelo trabaja dentro del software Vensim se muestra a continuación:

- **Auxiliar**

SprintsPuntos:

SprintsPuntos=

if then else(MomentoInicioSprints=1, PuntosSegunSprints , 0)

Units: Puntos

PuntosHacer:



PuntosHacer=

if then else (MomentoInicioSprints =1,0, if then

else((BURNDOWNCHARTS-

VelocidadIdealSprints)>=VelocidadIdealSprints,

BURNDOWNCHARTS-VelocidadIdealSprints, if then else

(BURNDOWNCHARTS=0: OR: BURNDOWNCHARTS=

VelocidadIdealSprints,0, 0)))

Units: Puntos

- **Flujo**

Puntos:

Puntos= SprintsPuntos

Units: Puntos

PuntosProcesar:

PuntosProcesar= BURNDOWNCHARTS-PuntosHacer

Units: Puntos

PuntosParaHacer:

PuntosParaHacer=

PUNTOSHECHAS-VelocidadIdealDesarrolloPuntos

Units: Puntos

PuntosPorSprint:

PuntosPorSprint= PUNTOSDESARROLLAR

Units: Puntos

- **Nivel**



BURDOWNCHARTS:

BURNDOWNCHARTS= INTEG (Puntos-PuntosProcesar, 0)

Units: Puntos

PUNTOSHECHOS:

PUNTOSHECHAS= INTEG (PuntosProcesar-PuntosParaHacer,
0)

Units: Puntos

PUNTOSDESARROLLAR:

PUNTOSDESARROLLAR= INTEG (Puntos-PuntosPorSprint,
0)

Units: Puntos

4.2.3. Sub Modelo de Ejecución de Tareas

4.2.3.1. Clasificación de Variables

Las variables del sub modelo se muestran a continuación respectivamente clasificadas y una breve descripción incluida:

- **Auxiliar**

TareasPorPuntos: El valor que contiene esta variable es el total de tareas que equivale a un punto. Definido por el encargado de la ejecución del modelo.

cantidadTareasPorPuntos: En esta variable se trabaja los datos provenientes de variables como PuntosSegunSprints, TareasPorPuntos y PUNTOSHECHAS. Representa la cantidad de tareas que se debes de desarrollar por cada punto. Aplica la función if then else, la cual verifica que mientras PUNTOSHECHAS sea



menor que `PuntosSegunSprints` guardara el valor del producto entre `PUNTOSHECHAS` con `tareasPorPuntos` y este resultado resta a `PuntosSegunSprints`, en caso que la comparación no cumpla, la presente variable almacenara cero.

TotalErroresDesarrollo: Almacena el total de errores del proyecto, haciendo la suma de la variable `PRUEBASCONERRDETECTADOS` y `PRUEBASCONERRINTEGRACION`.

TareasARecodificar: Obtiene el número total de tareas a ser recodificados. Trabaja con las variables `TAREASCONERRORES` y `TareasConErrDeCodificacion`, las cuales aplicando la función `if then else` define el valor a almacenar ya sea los de `TAREASCONERRORE` o simplemente cero.

tasaAjusteVelocidad: Es la velocidad ideal con la que se desarrollan las tareas, modificable con la finalidad de cumplir los tiempos establecidos por cada Sprint para su desarrollo.

- **Flujo**

TareasConErrDeCodificacion: Esta variable vincula `TotalErroresDesarrollo` con las variables `TAREASCONERRORES` y `TareasARecodificar`. La función `if then else` verifica que `TotalErroresDesarrollo` sea mayor a cero, si se cumple la condición hace la diferencia entre `TotalErroresDesarrollo` y `TAREASCONERRORES`, si la condición no se cumple tomara cero como valor.



TareasAProbar: Es la variable que resta a TAREASPARACODIF, y para obtener su valor, dentro de esta variable se resta TareasProbar de TAREASPARACODIF.

- **Nivel**

TAREASPLANIFICADAS: Tareas que de acuerdo al tiempo serán desarrollados por cada Sprint. A TareasPendientes se le resta PlanificacionACodificacion para definir el valor de la actual variable.

TAREASPENDIENTESCODIFICAR: Son el total de tareas que serán codificados, y el valor será obtenido sumando PlanificacionACodificacion, TareasConErrARecod, TareasExtras y restando TareCodif.

TAREASCONERRORES: Almacena información del total de tareas con errores, el cual es obtenido por la variable TareasConErrDeCodificacion.

TAREASPARACODIF: Almacena información de las tareas para codificar. Para obtener sus valores tenemos TareCodif al cual se le resta TareasAProbar y SiendoCodif.

TAREASCODIFICADAS: Los valores que almacena son la codificación de las tareas de determinados Sprint de acuerdo al tiempo en curso.



TotalErroresDesarrollo:

TotalErroresDesarrollo=INTEGER(PRUEBASCONERRDETEC
TADOS*pruebasPorTarea+PRUEBASCONERRINTEGRACIO
N*pruebasIntegracionPorTarea)

Units: Tareas

TareasARecodificar:

TareasARecodificar= if then
else(TAREASCONERRORES>TareasConErrDeCodificacion,
TAREASCONERRORES, 0)

Units: Tareas

TareasProbar:

TareasProbar= if then
else(MomentoInicioSprints=1:OR:((TAREASPARACODIF)-
(tasaAjusteVelocidad-VelocidadIdealSprints))<0, 0,
(TAREASPARACODIF)-(tasaAjusteVelocidad-
VelocidadIdealSprints))

Units: Tareas

- **Flujo**

TareasConErrDeCodificacion:

TareasConErrDeCodificacion= if then
else(TotalErroresDesarrollo>0, TotalErroresDesarrollo-
TAREASCONERRORES, 0)

Units: Tareas



TareasConErrARecod:

TareasConErrARecod= TareasARecodificar

Units: Tareas

PlanificacionACodificacion:

PlanificacionACodificacion= TAREASPLANIFICADAS

Units: Tareas

TareasPendientes:

TareasPendientes= cantidadTareasPorPuntos

Units: Tareas

TareasExtras:

TareasExtras= TAREASEXTRASDESARROLLADAS

Units: Tareas

TareCodif:

TareCodif= TAREASPENDIENTESCODIFICAR

Units: Tareas

TareasAProbar:

TareasAProbar= TAREASPARACODIF-TareasProbar

Units: Tareas

SiendoCodif:

SiendoCodif= TAREASPARACODIF

Units: Tareas

- **Nivel**

TAREASPLANIFICADAS:

TAREASPLANIFICADAS= INTEG (



TareasPendientes-PlanificacionACodificacion,0)

Units: Tareas

TAREASPENDIENTESCODIFICAR:

TAREASPENDIENTESCODIFICAR= INTEG (

PlanificacionACodificacion+TareasConErrARecod+TareasExtras

-TareCodif, 0)

Units: Tareas

TAREASCONERRORES:

TAREASCONERRORES= INTEG (

TareasConErrDeCodificacion, 0)

Units: Tareas

TAREASPARACODIF:

TAREASPARACODIF= INTEG (

TareCodif-TareasAProbar-SiendoCodif, 0)

Units: Tareas

TAREASCODIFICADAS:

TAREASCODIFICADAS= INTEG (

SiendoCodif-TAREASCODIFICADAS, 0)

Units: Tareas



4.2.4. Sub Modelo de Pruebas de Tareas

4.2.4.1. Clasificación de Variables

Las variables que trabajan en este sub modelo son las siguientes según su clasificación:

- **Auxiliar**

pruebasPorTarea: El número de pruebas por tarea que se tiene que hacer.

pruebasIntegracionPorTarea: Numero de pruebas en la integración de tareas que se tendrá que hacer.

TotalPruebas: Total de tareas que entraran a la prueba. Obtenida por el producto de pruebasPorTarea y TAREASCODIFICADAS.

TotalPruebasIntegracion: Llega el total de tareas que fueron sometidos a la prueba y que no dieron errores, estas se convierten en el total de pruebas de Integración que se aran. La ecuación que se aplica es la siguiente: se hace la división entre la variable PRUEBASSINERROR y pruebasPorTarea, para luego multiplicarlo con pruebasIntegracionPorTarea.

tasaErrPorPresionPlazo: Variable que según transcurre el tiempo toma valores de la división entre PRESIONPLAZO y EXPERIENCIATEAM, para obtener la tasa de error en momentos de presión de plazo de entrega.

tasaErrPorCansancio: Toma el valor de CANSANCIO de team, obteniendo la tasa de error por motivo de cansancio.

- **Flujo**



ErrorsPorCodif: Variable que procesa la información de la división de PRUEBASREALIZAR y la suma de tasaErrPorPresionPlazo, tasaErrPorCansancio y TotalTeam. La cual nos daría información para transferirla a PRUEBASCONERRDETECTADOS.

PruebasSinErrorCodif: En esta variable se hace la resta de PRUEBASREALIZAR y PRUEBASCONERRDETECTADOS, para trabajar posteriormente en la variable PRUEBASSINERROR.

- **Nivel**

PRUEBASREALIZAR: Almacena información proveniente de PruebasDisendas, es el total de prueba a realizar. A PruebasDisenadas se le resta ErrorsPorCodif y PruebasSinErrorCodif, eso para obtener los valores de la actual variable.

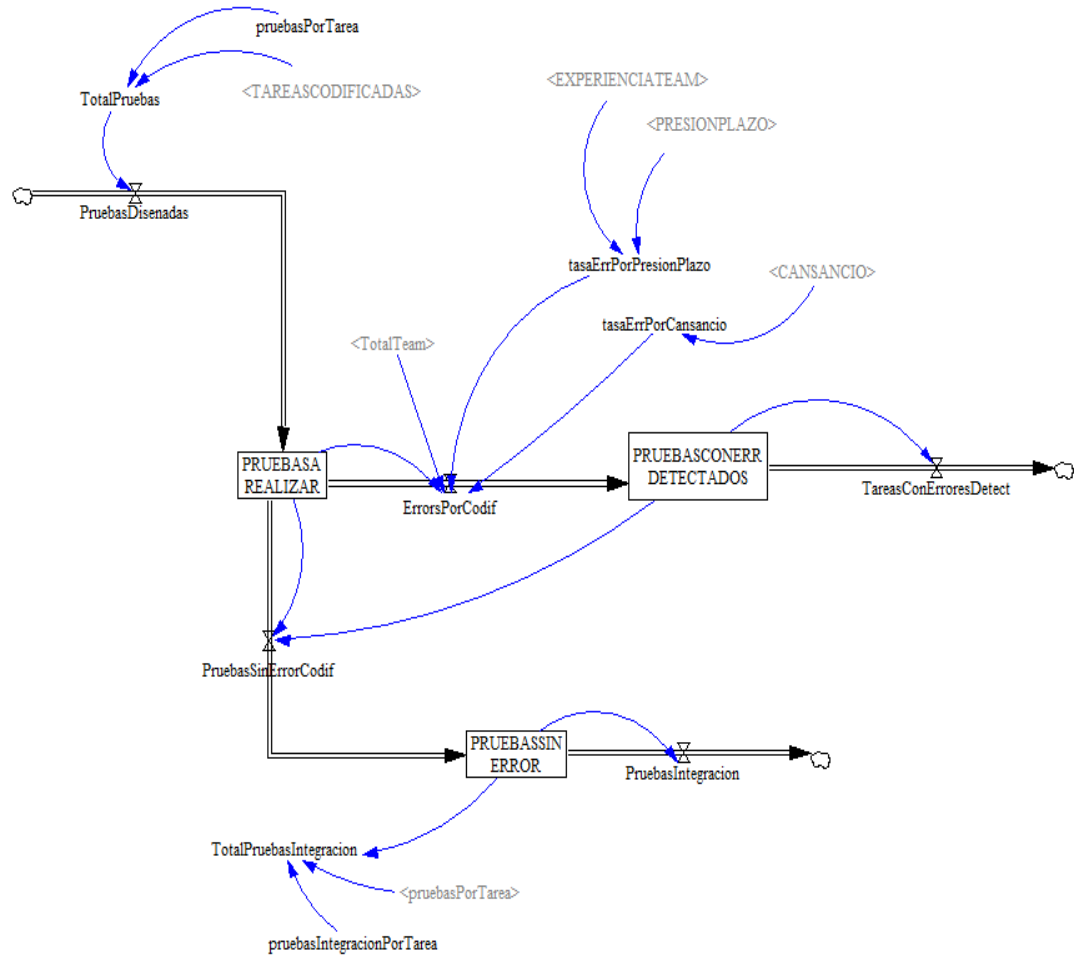
PRUEBASCONERRDETECTADOS: Pruebas que se hicieron a las tareas y que resultaron con error. Ingresan los valores de ErrorsPorCodif y se le disminuye TareasConErroresDetect.

PRUEBASSINERROR: Pruebas que se hicieron a las tareas y que resultaron correctas. Ingresan los valores de PruebasSinErrorCodif y se le disminuye PruebasIntegracion. Estos datos serán útiles también para las pruebas de Integración.

4.2.4.2. Diagrama de Forrester

Figura 23

Diagrama de forrester del sub modelo de pruebas de tareas



Nota: elaboración propia

4.2.4.3. Ecuaciones del Sub Modelo

- **Auxiliar**

TotalPruebas:

$$\text{TotalPruebas} = \text{pruebasPorTarea} * \text{TAREAS CODIFICADAS}$$

Units: Pruebas

TotalPruebasIntegracion:

$$\text{TotalPruebasIntegracion} = ((\text{PRUEBAS SIN ERROR} / \text{pruebasPorTarea}) * \text{pruebasIntegracionPorTarea})$$

Units: Pruebas



tasaErrPorPresionPlazo:

tasaErrPorPresionPlazo= if then else(EXPERIENCIATEAM>0,
(PRESIONPLAZO/EXPERIENCIATEAM), 0)

Units: Tasa

tasaErrPorCansancio:

tasaErrPorCansancio= CANSANCIO

Units: Tasa

- **Flujo**

ErrorsPorCodif:

ErrorsPorCodif= if then
else(tasaErrPorCansancio+tasaErrPorPresionPlazo=0, 0 , if then
else(PRUEBASAREALIZAR/(tasaErrPorPresionPlazo+tasaErrP
orCansancio+TotalTeam)>=0,PRUEBASAREALIZAR/(tasaErrP
orPresionPlazo+tasaErrPorCansancio+TotalTeam) , 0))

Units: Pruebas

PruebasSinErrorCodif:

PruebasSinErrorCodif= if then else(PRUEBASAREALIZAR-
PRUEBASCONERRDETECTADOS>=0,
PRUEBASAREALIZAR-
PRUEBASCONERRDETECTADOS,0)

Units: Pruebas

PruebasDisenadas:

PruebasDisenadas= TotalPruebas

Units: Pruebas

TareasConErroresDetect:



TareasConErroresDetect= PRUEBASCONERRDETECTADOS

Units: Tareas

PruebasIntegracion:

PruebasIntegracion= PRUEBASSINERROR

Units: Pruebas

- **Nivel**

PRUEBASAREALIZAR:

PRUEBASAREALIZAR= INTEG (PruebasDisenadas-

PruebasSinErrorCodif-ErrorsPorCodif, 0)

Units: Pruebas

PRUEBASCONERRDETECTADOS:

PRUEBASCONERRDETECTADOS= INTEG (ErrorsPorCodif-

TareasConErroresDetect,0)

Units: Pruebas

PRUEBASSINERROR:

PRUEBASSINERROR= INTEG (PruebasSinErrorCodif-

PruebasIntegracion,0)

Units: Pruebas

4.2.5. Sub Modelo de Presión en el Equipo

4.2.5.1. Clasificación de Variables

La clasificación de las variables del presente sub modelo se muestra a continuación:

- **Lookup**



PresionTeamPlazo: Define según el tiempo en curso, el porcentaje de presión que sufre el equipo de trabajo. Tiene como entrada X la representación del tiempo completo en porcentajes, y como salida Y el porcentaje de presión.

- **Auxiliar**

CoficientePresionPlazo: Extrae el porcentaje de presión de la variable `PresionTeamPlazo` usando la variable `PorcentajeTiempoSprints`, esto gracias a la función `LOOKUP FORWARD`.

PorcentajeTiempoSprints: Obtiene el porcentaje del tiempo según esta va transcurriendo.

- **Flujo**

PrPlazo: Información que se obtiene de sumar `EXPERIENCIATEAM` y `diferenciaEnLaPresionPlazo`, para luego trabajarla en `PRESIONPLAZO`.

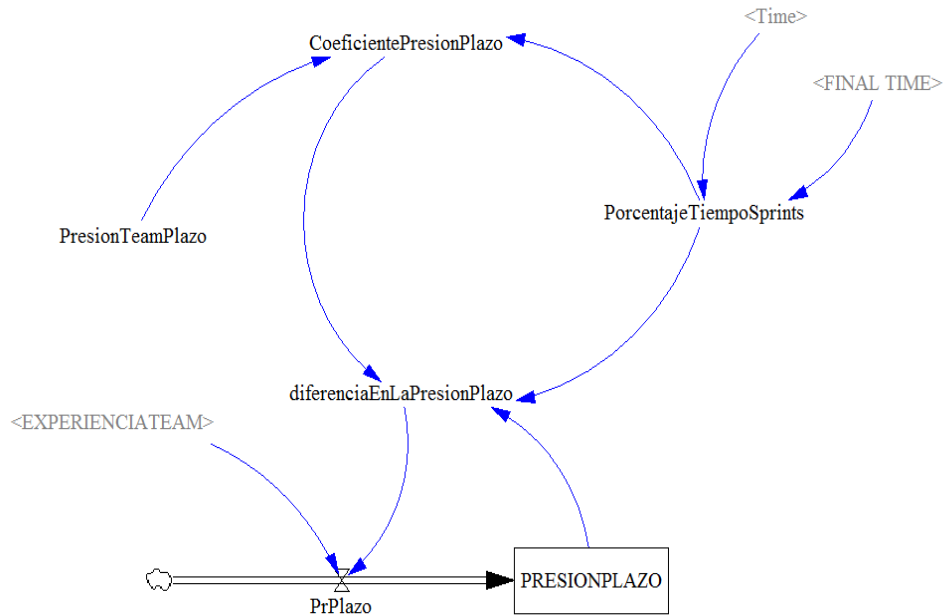
- **Nivel**

PRESIONPLAZO: Almacena información de la presión que el equipo de trabajo tiene de acuerdo al tiempo que transcurre en el proyecto.

4.2.5.2. Diagrama de Forrester

Figura 24

Diagrama de forrester del sub modelo de presión en el plazo



Nota: elaboración propia

4.2.5.3. Ecuaciones del Sub Modelo

- **Auxiliar**

CoeficientePresionPlazo:

CoeficientePresionPlazo=LOOKUP

FORWARD(PresionTeamPlazo, PorcentajeTiempoSprints)

Units: Tasa

PorcentajeTiempoSprints:

PorcentajeTiempoSprints= if then else(FINAL TIME>Time, 1-

((FINAL TIME-Time)/FINAL TIME) , 1)

Units: Horas

diferenciaEnLaPresionPlazo:



$diferenciaEnLaPresionPlazo =$

$PorcentajeTiempoSprints * CoeficientePresionPlazo -$

$PRESIONPLAZO$

Units: Tasa

- **Flujo**

PrPlazo:

$PrPlazo = diferenciaEnLaPresionPlazo + EXPERIENCIATEAM$

Units: Presion

- **Nivel**

PRESIONPLAZO:

$PRESIONPLAZO = INTEG (PrPlazo, 0)$

Units: Presion

4.2.6. Sub Modelo de Tareas Adicionales

4.2.6.1. Clasificación de Variables

Las variables que interactúan en el diagrama de forrester del presente sub modelo son las que se muestran a continuación según su clasificación:

- **Lookup**

TareasExtrasSprint: En esta variable se define el total de tareas adicionales por cada Sprint. Tenemos como entrada X el número de Sprint y la salida Y el número de tareas extras.

- **Constante**



probabilidadTareaExtraNoPlanificada: Es el porcentaje de probabilidad de tareas que no fueron planificados al principio del proyecto.

- **Auxiliar**

PeriodoTareaExtraNoPlanificada: El periodo de tiempo en el cual tenemos la mayor probabilidad de que las tareas no planificadas aparezcan.

TareaExtraNoPlanificada: A través de la función PULSE TRAIN, esta variable obtiene pulsaciones que representan las tareas que no fueron planificadas, vendrían a ser las tareas adicionales.

TareasExtrasPorHacer: Obtiene datos del total de tareas por cada sprint que no fue planificado. Esto con la función if then else que verifica que MomentoInicioSprints sea igual a la unidad, cumpliendo esta condición la función LOOKUP FORWARD obtiene las tareas adicionales haciendo uso de las variables TareasExtrasSprint y TIEMPOTOTALSPRINTS, en caso que la comparación resulte falsa da cero.

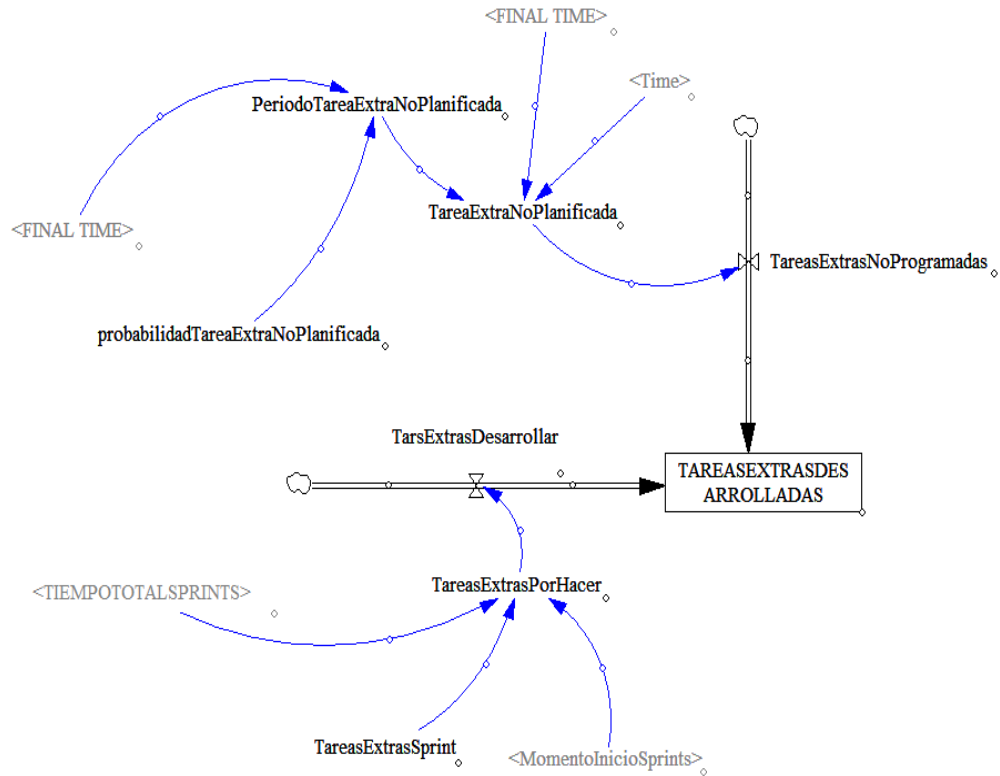
- **Nivel**

TAREASEXTRASDESARROLLADAS: Almacena información de las tareas probables que no fueron considerados en la planificación y las tareas que se adicionaran a cada sprint después de haber dado inicio al proyecto.

4.2.6.2. Diagrama de Forrester

Figura 25

Diagrama de forrester del sub modelo de tareas adicionales



Nota: Elaboración Propia

4.2.6.3. Ecuaciones del Sub Modelo

- **Auxiliar**

PeriodoTareaExtraNoPlanificada:

PeriodoTareaExtraNoPlanificada= if then

else(probabilidadTareaExtraNoPlanificada>0,

(FINALTIME/3)*(1-probabilidadTareaExtraNoPlanificada) , 0)

Units: Horas

TareaExtraNoPlanificada:

TareaExtraNoPlanificada= if then

else(PeriodoTareaExtraNoPlanificada>0:AND:Time>1,PULSE



TRAIN(1, 1 , PeriodoTareaExtraNoPlanificada , FINAL TIME) ,0
)

Units: Tareas

TareasExtrasPorHacer:

TareasExtrasPorHacer=if then else(MomentoInicioSprints=1,
LOOKUP FORWARD(TareasExtrasSprint,
TIEMPOTOTALSPRINTS) , 0)

Units: Tareas

- **Flujo**

TarsExtrasDesarrollar= TareasExtrasPorHacer

Units: Tareas

- **Nivel**

TAREASEXTRASDESARROLLADAS:

TAREASEXTRASDESARROLLADAS= INTEG (
(TarsExtrasDesarrollar+TareasExtrasNoProgramadas)-
TAREASEXTRASDESARROLLADAS, 0)

Units: Tareas

4.2.7. Sub Modelo de Pruebas de Integración

4.2.7.1. Clasificación de Variables

La clasifiacacion de las variables que trabajan en este sub modelo son las que a continuación se muestran:

- **Constante**

tasaPruebasErrIntegracion: Porcentaje de probabilidad de errores de Integración.



- **Flujo**

DisenioPruebasIntegracion: Mediante la cual se transfiere datos de la variable TotalPruebasIntegracion, esto a la variable PRUEBASINTEGRACIONDISENAR.

PruebConErrIntegracion: En esta variable se realiza el producto de PRUEBASINTEGRACIONREALIZADAS con tasaPruebasErrorIntegracion.

PruebasAIntegrar: Esta variable obtiene información a través de la diferencia entre PRUEBASINTEGRACIONREALIZADAS y PRUEBASCONERRINTEGRACION.

- **Nivel**

PRUEBASINTEGRACIONADISENAR: Plasma datos que muestran las pruebas que se realizaran. Obteniendo de la diferencia entre DisenioPruebasIntegracion y PruebasIntegrARealizar.

PRUEBASINTEGRACIONREALIZADAS: Vendría a ser la simulación del total de las pruebas de Integración. Para obtener datos se hace la diferencia de PruebasIntegrARealizar con la suma de PruebasAIntegrar y PruebConErrIntegracion.

PRUEBASCONERRINTEGRACION: Representa las pruebas de integración que dieron como resultado erróneas. De la variable PruebConErrIntegracion le disminuye PRUEBASCONERRINTEGRACION para definir datos.

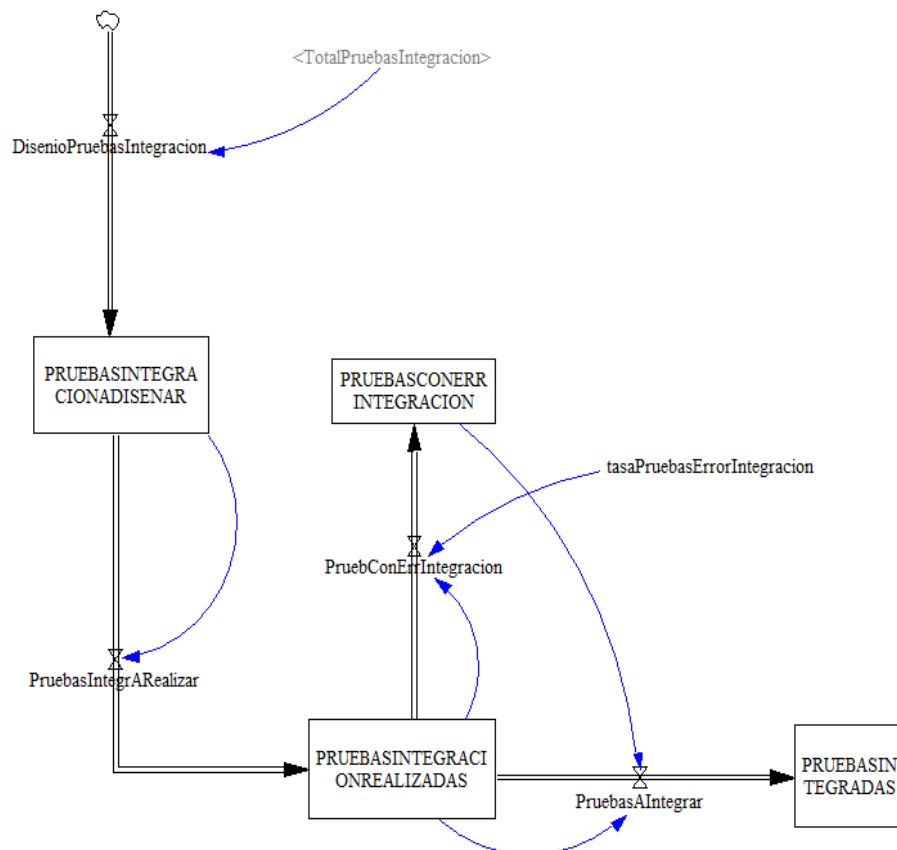
PRUEBASINTEGRADAS: Representa las pruebas de integración que dieron como resultado correctas. Los datos que

almacena resultan de la diferencia de la variable PruebasAIntegrar y PRUEBASINTEGRADAS.

4.2.7.2. Diagrama de Forrester

Figura 26

Diagrama de forrester del sub modelo de pruebas de integración



Nota: elaboración propia

4.2.7.3. Ecuaciones del Sub Modelo

- Flujo

DisenioPruebasIntegracion:

$\text{DisenioPruebasIntegracion} = \text{TotalPruebasIntegracion}$

Units: Pruebas

PruebConErrIntegracion:



$PruebConErrIntegracion = PRUEBASINTEGRACIONREALIZA$

$DAS * tasaPruebasErrorIntegracion$

Units: Pruebas

PruebasAIntegrar:

$PruebasAIntegrar = PRUEBASINTEGRACIONREALIZADAS -$

$PRUEBASCONERRINTEGRACION$

Units: Pruebas

PruebasIntegrARealizar:

$PruebasIntegrARealizar =$

$PRUEBASINTEGRACIONADISENAR$

Units: Pruebas

- **Nivel**

PRUEBASINTEGRACIONADISENAR:

$PRUEBASINTEGRACIONADISENAR = INTEG ($

$DisenioPruebasIntegracion - PruebasIntegrARealizar, 0)$

Units: Pruebas

PRUEBASINTEGRACIONREALIZADAS:

$PRUEBASINTEGRACIONREALIZADAS = INTEG ($

$PruebasIntegrARealizar -$

$(PruebasAIntegrar + PruebConErrIntegracion), 0)$

Units: Pruebas

PRUEBASCONERRINTEGRACION:

$PRUEBASCONERRINTEGRACION = INTEG ($

$PruebConErrIntegracion - PRUEBASCONERRINTEGRACION,$

$0)$



Units: Pruebas

PRUEBASINTEGRADAS:

PRUEBASINTEGRADAS= INTEG (

PruebasAIntegrar-PRUEBASINTEGRADAS, 0)

Units: Pruebas

4.2.8. Sub Modelo de Tiempo de Trabajo

4.2.8.1. Clasificación de Variables

Esta clasificación se muestra a continuación con una breve descripción de las mismas:

- **Lookup**

HorasExtrasPorSprints: Variable que almacena información de las horas extras no planificadas por cada Sprint. Como entrada X tenemos el número de Sprint, y salida Y el total de horas extras.

DuracionHorasDeSprints: El tiempo total que cada Sprint lleva su desarrollo. Como entrada X tenemos el número de Sprint, y como salida Y el tiempo que se toma la ejecución de Sprint.

- **Auxiliar**

HorasExtrasTrabajadas: Dato de las horas totales que se trabaja por cada Sprint. La extrae aplicando la función if then else donde verifica que `MomentoInicioSprints` sea mayor a cero, si la comparación es correcta, entra a la función `lookup backward` donde la variable `TIEMPOTOTALSPRINTS` busca el Sprint que corresponde en `HorasExtrasPorSprints`, y obtiene el total de horas



extras que se trabajó de dicho Sprint, caso contrario en la condición obtendrá cero.

HorasNormalesTrabajadas: Dato de las horas que se trabaja por cada Sprint planificado. La extrae aplicando la función if then else donde verifica que MomentoInicioSprints sea mayor a cero, si la comparación es correcta, entra a la función lookup backward donde la variable TIEMPOTOTALSPRINTS busca el Sprint que corresponde en DuracionHorasDeSprints, y obtiene el total de horas panificadas que se trabaja en un determinado Sprint, caso contrario en la condición obtendrá cero.

- **Flujo**

HorasTrabajadas: Variable que suma HorasExtrasTrabajadas y HorasNormalesTrabajadas, para procesar la información del total de horas trabajadas por Sprint.

- **Nivel**

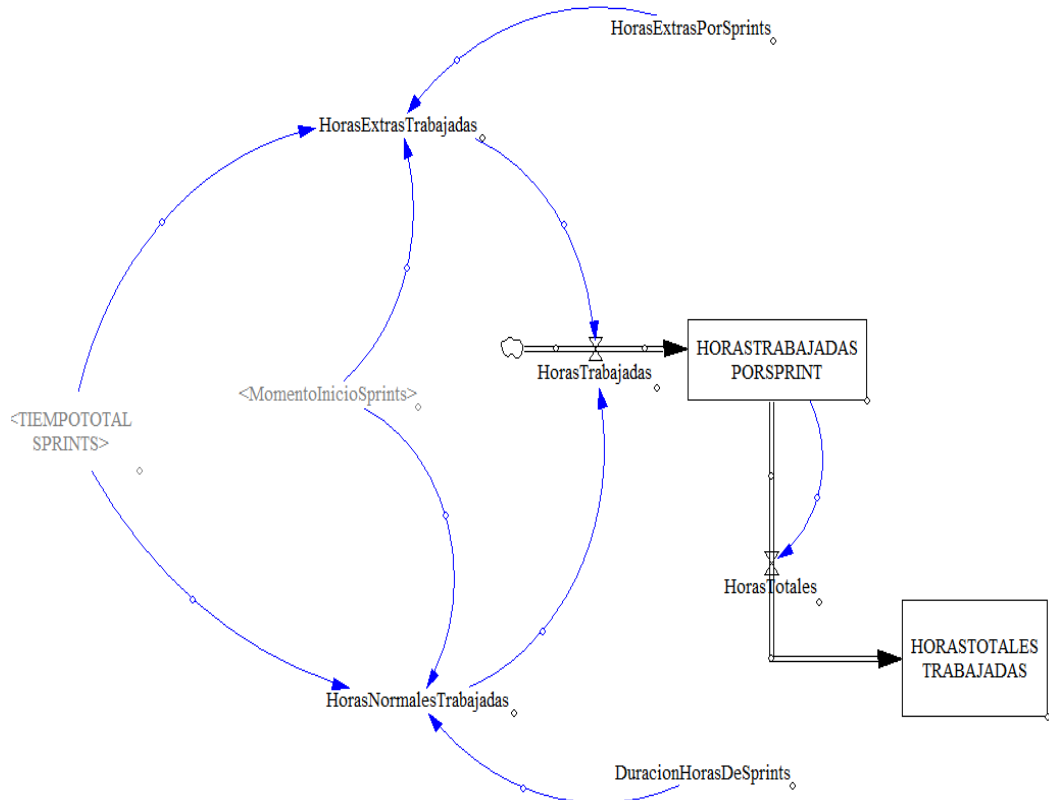
HORASTRABAJADASPORSPRINT: Almacena información de las horas totales trabajadas por cada Sprint, mostrándola como pulsaciones.

HORASTOTALESTRABAJADAS: Almacena información de las horas totales trabajadas por cada Sprint, mostrándola en forma de escalones.

4.2.8.2. Diagrama de Forrester

Figura 27

Diagrama de forrester del sub modelo de tiempo de trabajo



Nota: elaboración propia

4.2.8.3. Ecuaciones del Sub Modelo

- **Auxiliar**

HorasExtrasTrabajadas:

```
HorasExtrasTrabajadas= if  $\langle \text{MomentoInicioSprints} \rangle > 0$  then
else(MomentoInicioSprints>0,lookup
backward(HorasExtrasPorSprints, TIEMPOTOTALSPRINTS),0)
```

Units: Horas

HorasNormalesTrabajadas:



HorasNormalesTrabajadas= if _____ then
else(MomentoInicioSprints>0, lookup backward(
DuracionHorasDeSprints, TIEMPOTOTALSPRINTS), 0)

Units: Horas

- **Flujo**

HorasTrabajadas:

HorasTrabajadas=HorasNormalesTrabajadas+HorasExtrasTrabaj
adas

Units: Horas

HorasTotales:

HorasTotales= HORASTRABAJADASPORSPRINT

Units: Horas

- **Nivel**

HORASTRABAJADASPORSPRINT:

HORASTRABAJADASPORSPRINT= INTEG (
HorasTrabajadas-HorasTotales, 0)

Units: Horas

HORASTOTALESTRABAJADAS:

HORASTOTALESTRABAJADAS= INTEG (
HorasTotales, 0)

Units: Horas



4.2.9. Sub Modelo de Cansancio

4.2.9.1. Clasificación de Variables

Las variables se muestran a continuación según su clasificación y una breve descripción de las mismas:

- **Lookup**

CoefCansancioSegunTiempo: El cansancio crece progresivamente y esta variable la almacena según el transcurre el tiempo. Como entrada X tenemos el tiempo en expresado en determinados periodos, y como salida Y tenemos el coeficiente de cansancio.

CoefCansancioPorHoras: Determina el cansancio por horas trabajadas dentro del día. Tenemos como entrada X las horas, y como salida Y el coeficiente de cansancio.

- **Auxiliar**

TiempoTranscurrido: El tiempo que transcurre asta tiempo fin.

TasaCansancio: Tasa de cansancio según el proyecto avanza. Deriva gracias a la función lookup forward que busca el tiempo en curso con la variable TiempoTranscurrido en CoefCansancioSegunTiempo.

TasaCansancioDiario: Tasa de cansancio total diario dentro del proyecto. Resulta de buscar con la función lookup forward el resultado de la suma de HorasPorDia y TAREASEXTRASDESARROLLADAS en CoefCansancioPorHoras.



DiferenciaCansancio: Variable que almacena información de las altas y bajas en el cansancio. Hace la suma de TasaCansancioDiario y TasaCansancio y le disminuye al resultado el valor de la variable CANSANCIO.

- **Constante**

HorasPorDia: Se define el total de horas trabajadas cada día.

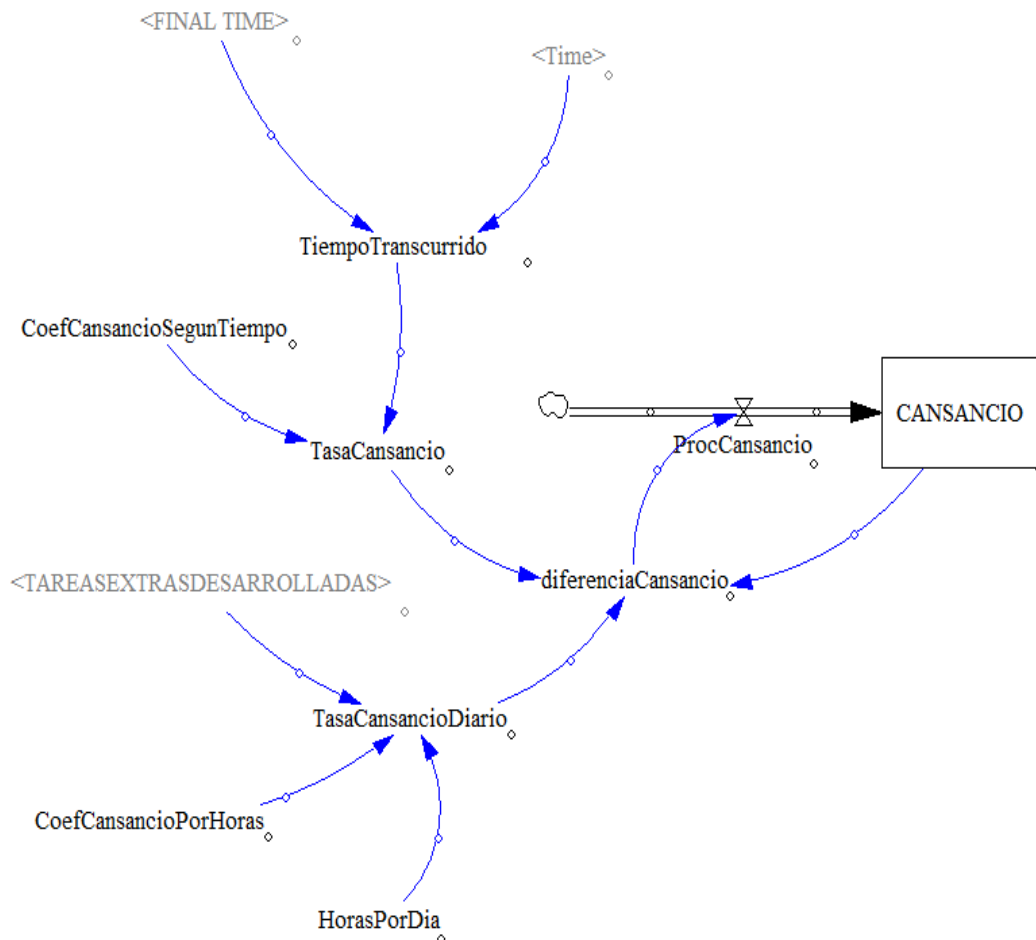
- **Nivel**

CANSANCIO: Almacena información del cansancio del equipo de trabajo de acuerdo al tiempo. Plasmándonos escalones de cansancio.

4.2.9.2. Diagrama de Forrester

Figura 28

Diagrama de forrester del sub modelo de cansancio



Nota: elaboración propia

4.2.9.3. Ecuaciones del Sub Modelo

- **Auxiliar**

TiempoTrascurrido:

TiempoTrascurrido= if then else(Time<FINAL TIME, Time , 0)

Units: Horas

TasaCansancio:



TasaCansancio=LOOKUP

FORWARD(CoefCansancioSegunTiempo, TiempoTranscurrido)

Units: Tasa

TasaCansancioDiario:

TasaCansancioDiario=LOOKUP

FORWARD(CoefCansancioPorHoras,(HorasPorDia+TAREASE
XTRASDESARROLLADAS))

Units: Tasa

DiferenciaCansancio:

diferenciaCansancio= TasaCansancioDiario+TasaCansancio-
CANSANCIO

Units: Cansancio

- **Flujo**

ProcCansancio= diferenciaCansancio

Units: Cansancio

- **Nivel**

CANSANCIO:

CANSANCIO= INTEG (ProcCansancio, 0)

Units: Cansancio

4.2.10. Sub Modelo de Incremento de Experiencia de Equipo

4.2.10.1. Clasificación de Variables

Las que se muestran a continuación son las que interactúan en el sub modelo:

- **Auxiliar**



ExperienciaSrs: Incremento de experiencia de senior. Obtenida del producto de la variable NivelProductividadSrs y PonderacionSrs.

PonderacionSrs: Ponderación del aprendizaje del total de todos los integrantes de nivel senior.

TotalTeam: Representa el total de integrantes del grupo de trabajo. Suma los integrantes de nivel Junior y Senior.

PonderacionJrs: Ponderación del aprendizaje del total de todos los integrantes de nivel junior.

ExperienciaJrs: Incremento de experiencia de senior. Obtenida del producto de la variable NivelProductividadSrs y PonderacionSrs.

- **Constante**

NivelProductividadSrs: Nivel de productividad del tipo de integrante senior.

tasaAprendizajeSrs: Tasa de aprendizaje del tipo de integrante senior.

RrHhSeniors: Total número de integrantes del tipo senior del equipo de trabajo.

RrHhJuniors: Total número de integrantes del tipo junior del equipo de trabajo.

tasaAprendizajeJrs: Tasa de aprendizaje del tipo de integrante junior.

NivelProductividadJrs: Nivel de productividad del tipo de integrante junior.

- **Flujo**

Experiencia: Acumulación de la experiencia de senior y junior.

Suma las variables ExperienciaSrs y ExperienciaJrs.

- **Nivel**

EXPERIENCIATEAM: Almacena información de la

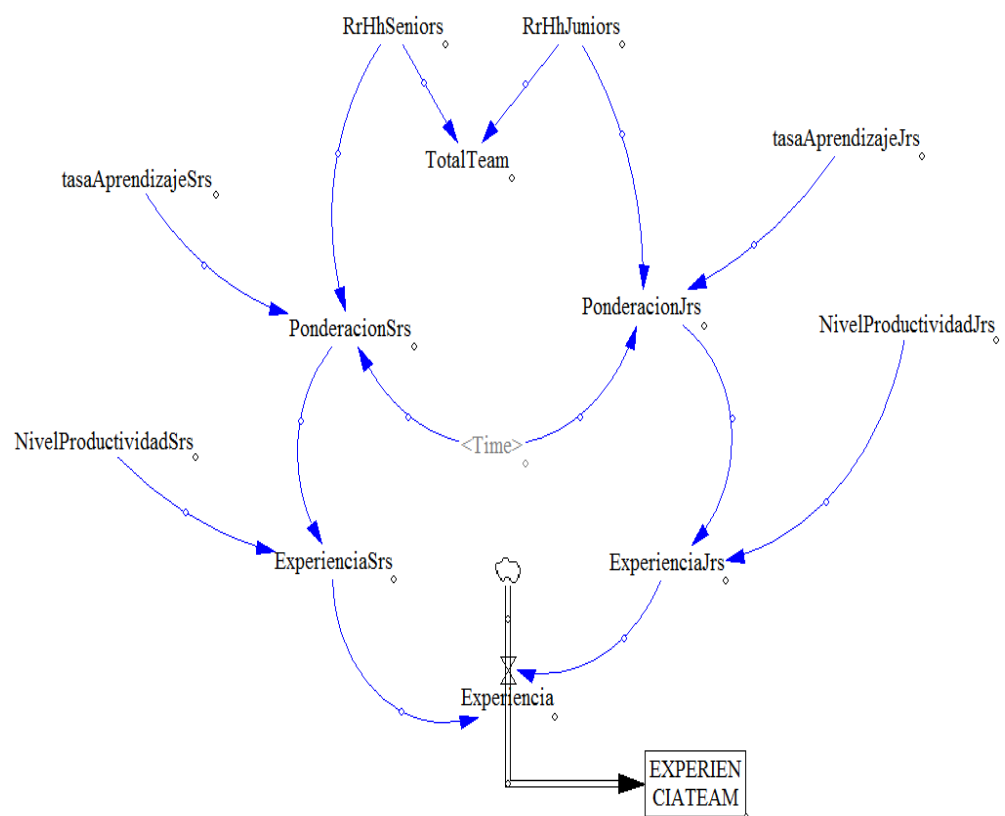
experiencia ganada de todo el equipo de trabajo. Logramos obtener

estos datos disminuyendo EXPERIENCIATEAM de Experiencia.

4.2.10.2. Diagrama de Forrester

Figura 29

Diagrama de forrester del sub modelo de incremento de experiencia de equipo



Nota: elaboración propia



4.2.10.3. Ecuaciones del Sub Modelo

- **Auxiliar**

ExperienciaSrs:

$$\text{ExperienciaSrs} = \text{NivelProductividadSrs} * \text{PonderacionSrs}$$

Units: Experiencia

PonderacionSrs:

$$\text{PonderacionSrs} = \begin{cases} \text{if} & \text{then} & \text{else}(\text{RrHhSeniors} > 0, \\ \text{RrHhSeniors} * (1 - (1.05^{-(\text{tasaAprendizajeSrs} * \text{Time}))}), 0 \end{cases}$$

Units: Tasa

TotalTeam:

$$\text{TotalTeam} = \text{RrHhJuniors} + \text{RrHhSeniors}$$

Units: Personas

PonderacionJrs:

$$\text{PonderacionJrs} = \begin{cases} \text{if} & \text{then} & \text{else}(\text{RrHhJuniors} > 0, \\ \text{RrHhJuniors} * (1 - ((2.712^{-(\text{tasaAprendizajeJrs} * \text{Time}))})), 0 \end{cases}$$

Units: Tasa

ExperienciaJrs:

$$\text{ExperienciaJrs} = \text{NivelProductividadJrs} * \text{PonderacionJrs}$$

Units: Experiencia

- **Flujo**

Experiencia:

$$\text{Experiencia} = (\text{ExperienciaSrs} + \text{ExperienciaJrs}) * 0$$

Units: Experiencia

- **Nivel**



EXPERIENCIATEAM:

EXPERIENCIATEAM= INTEG (Experiencia-

EXPERIENCIATEAM, 0)

Units: Experiencia

4.3. FASE DE EVALUACIÓN

La tercera fase que nos llevara a la verificación de la confiabilidad del modelo, permite comparar el desarrollo de las variables del modelo con el comportamiento de las variables del proyecto real. El mismo sirve para terminar de calibrar el modelo.

Dado que la validación de un modelo dinámico está más estrechamente relacionada con el objetivo del modelo que cualquier método, hemos seguido el camino sugerido por Forrester, Senge y Barlas al desarrollar el conjunto de métodos y la metodología para validar el modelo generado en este proyecto.

Se sugiere validar primero la estructura del modelo; es decir, se realizaría inicialmente una verificación estructural. Una vez confirmado el diseño estructural, se comparan los datos reales del sistema y los procedentes de las simulaciones utilizando la técnica estadística del valor absoluto del error medio para apoyar la validación del comportamiento del modelo.

Las cuales se desarrollan a continuación:

4.3.1. Validación Estructural

4.3.1.1. Verificación de la Estructura

La verificación de la estructura implica comparar explícitamente la estructura del modelo con los modelos actuales y los del mundo real, asegurándose de que ambos no entran en conflicto con la comprensión del sistema.

Para verificar la estructura del modelo propuesto se recurre al conocimiento experto del sistema y a una revisión exhaustiva de la bibliografía pertinente. Examinando el resumen, las conclusiones y las palabras clave, se creó una selección preliminar de más de 15 artículos para garantizar un estudio crítico de la revisión bibliográfica. Posteriormente, se



llevó a cabo un análisis más exhaustivo de cinco artículos; el proceso de selección se centró en las contribuciones, los enfoques, los métodos y las metodologías de los distintos estudiosos.

Como consecuencia de la revisión bibliográfica, se establecieron los puntos de vista actuales sobre el uso de la Dinámica de Sistemas en cuatro áreas cruciales en el campo del desarrollo de software: planificación y control, equipos de trabajo, producción de software y cambios. A su vez, los trabajos académicos aportados por diferentes autores fueron escrutados y evaluados en función de su metodología e implementación.

El libro titulado "Dinámica de Sistemas" de Javier Aracil y Francisco Gordillo, está entre las primeras fuentes más importantes de información estructural para el desarrollo del modelo, ya que representa el origen y las bases para el dominio de un modelo dinámico en materia de: simbología, estructura de relación causal, diagrama Forrester, caracterización de variables, entre otros puntos importantes.

De la estructura comparada con los modelos actuales se desprende claramente la contribución de los modelos creados por Cao, Ramesh y Abdel-Hamid, Godoy y Belloni, Sosa y Kotynski y Benites. Éstos se han concentrado en la metodología y la evaluación de las principales políticas de capacidad dinámica de los marcos ágiles para la construcción de proyectos de software.

Las aportaciones de la revisión bibliográfica sirvieron de base para desarrollar la estructura fundamental del modelo y examinar cómo se integraban las distintas variables asociadas, garantizando que no se pasara

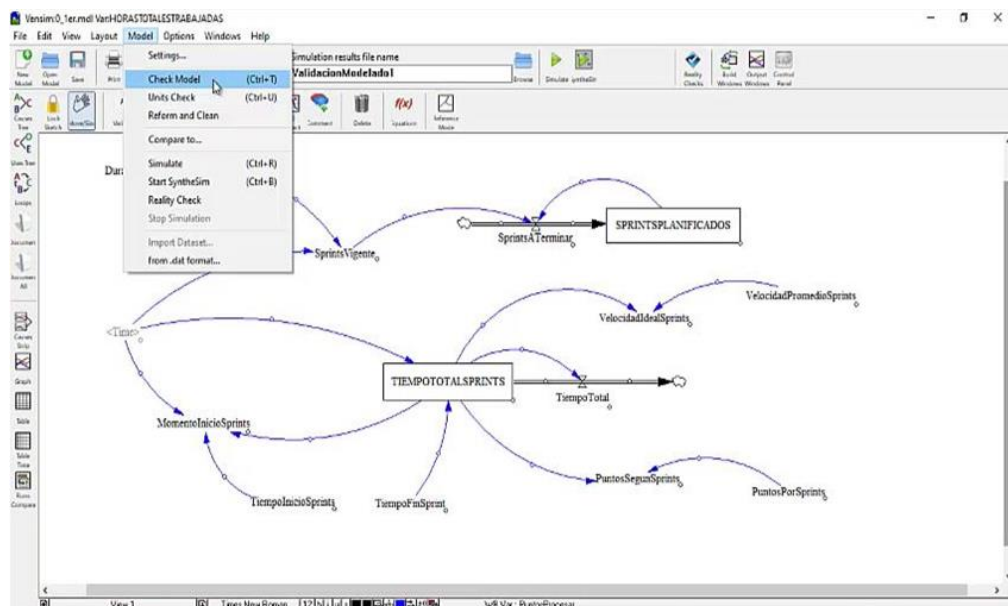
por alto ningún factor que pudiera repercutir en el correcto funcionamiento del modelo.

4.3.1.2. Consistencia Dimensional

Cada ecuación matemática del modelo, así como la adición de parámetros, deben evaluarse y examinarse para superar la prueba de coherencia dimensional. El entorno de simulación de fácil manejo que ofrece el software Vensim Ple® permite construir modelos y reproducir su comportamiento en tiempo real. Este software mejora la capacidad de examinar las unidades de medida y la estructura de las ecuaciones del modelo.

Figura 30

Funciones en el software Vensim (Check Model y Units Check) para revisar sintaxis del modelo



Nota: fuente Software Vensim PLE

El verificador de modelos proporcionado por el programa Vensim® PLE se representa en la Figura 31. Tiene dos funciones: una para



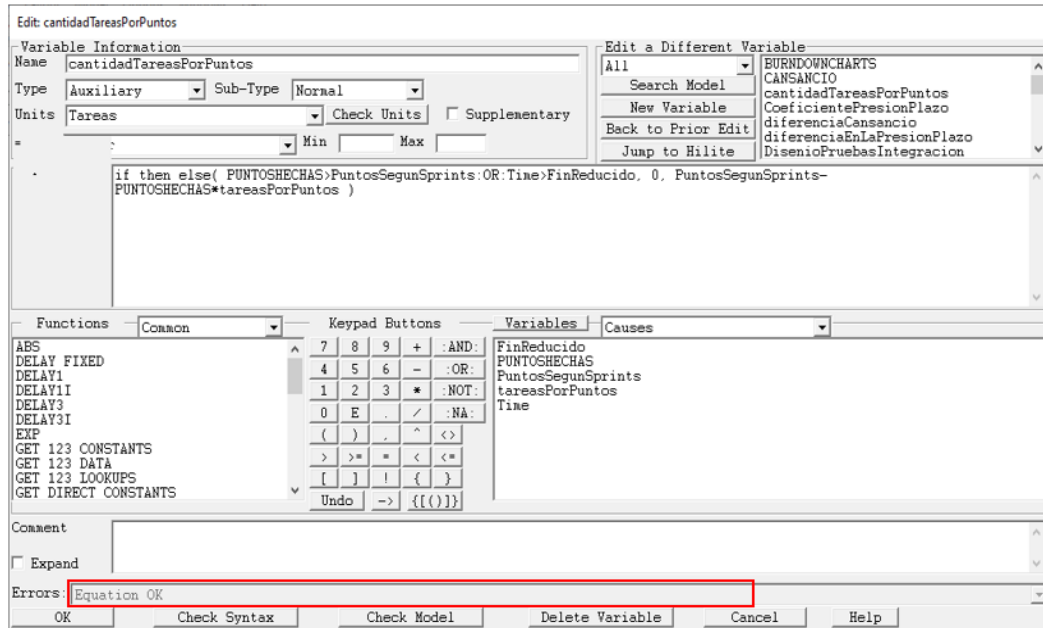
verificar la sintaxis del modelo (Check Model) y otra para buscar inconsistencias en la consistencia de las unidades (Units Check). Esto permite verificar el modelo en paralelo mientras se construye y se añaden los distintos parámetros.

$$\begin{aligned} \text{cantidadTareasPorPuntos} = & \text{if}(\text{PUNTOSHECHAS} \\ & > \text{PuntosSegunSprints}, 0, (\text{PuntosSegunSprints} \\ & - (\text{PUNTOSHECHAS} * \text{tareasPorPuntos}))) \end{aligned}$$

Considere el siguiente ejemplo, se verifica la consistencia de la ecuación de la cantidad de tareas que se deben de desarrollar por cada punto (cantidadTareasPorPuntos), con las herramientas disponibles en el software Vensim PLE. En esta variable se trabaja los datos provenientes de variables como PuntosSegunSprints, TareasPorPuntos y PUNTOSHECHAS. Aplica una condicional, la cual verifica que mientras PUNTOSHECHAS sea menor que PuntosSegunSprints guardara el valor del producto entre PUNTOSHECHAS con tareasPorPuntos y este resultado resta a PuntosSegunSprints, en caso que la comparación no cumpla, la presente variable almacenara cero.

Figura 31

Estado de la ecuación cantidadtareasporpuntos



Nota: fuente software Vensim PLE

El estado de la ecuación mencionada se muestra en la Figura 32 mediante el recuadro rojo. Afirmando que la ecuación se ha generado adecuadamente y de acuerdo con los objetivos del modelo, la sintaxis "Ecuación OK" certifica que la ecuación es correcta.

Cabe indicar que misma al anterior ejemplo es el estado de las ecuaciones con las que se trabaja en cada uno de los sub modelos, con esto el software Vensim PLE nos indica que el proyecto de modelado de DS que se ha trabajado cumple con los requisitos para considerarse que cumple con la consistencia dimensional.

4.3.1.3. Pruebas de Condiciones Extremas

En esta parte del proyecto se desarrolla la experimentación de escenarios de cada uno de los proyectos tomados como muestra, por ello serán un total de tres modelados. Ya que se trabaja con los tres casos como

base, se harán modificaciones de valores y parámetros iniciales a fin de probar el modelo ante situaciones extremas y/o diferentes.

Para estos casos de experimentación de escenario se requieren asignar valores a los coeficientes del tipo de variable LookUp tales como del Coeficiente de Cansancio Según el Tiempo, Coeficiente de Cansancio por Horas y Presión Plazo, es por ello que dichos valores se adaptaron de los antecedentes que trabajaron con este tipo de proyectos, es válido mencionar también a Godoy & Bartó.

Así mismo los resultados de las figuras y cambios generados se muestran haciendo comparaciones.

4.3.1.3.1. Escenario 1: Agotamiento en el Equipo

Los datos iniciales del proyecto “Sistema de Planificación de Recursos” son las siguientes:

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	40	40	8	0.21	8	0
2	40	120	80	14	0.175	14	0
3	120	240	120	18	0.15	18	0

Equipo de Trabajo: 3 personas, 1 Scrum Master y 2 Team, 1 Nivel Junior y 2 Nivel Senior.

Horas Diarias: 8h.

Para realizar el modelado como escenario, los datos que también el modelo tomara son estas:



Número de Tareas que Equivale a Un Punto: 1

Pruebas por tarea: 1

Pruebas de Integración por Tarea: 1

Tasa de Pruebas de Error de Integración: 0

Presión de Plazo de Entrega en el Equipo de Trabajo:

(0,0),(0.1,0),(0.2,0),(0.3,0),(0.4,0),(0.5,0),(0.6,0),(0.7,0),(0.8,0),(0.9,0),(1,0)

Probabilidad de Tareas Extras no Planificadas: 0

Horas Promedio Por Tareas Extras: 0

Tareas Extras por Sprints: (1,0), (2,0), (3,0)

Tasa de Aprendizaje Seniors: 0

Nivel de Productividad Seniors: 0

Tasa de Aprendizaje Juniors: 0

Nivel de Productividad Juniors: 0

Valores adaptados de antecedentes para el modelado de escenario:

Coefficiente de Cansancio Según el Tiempo:

(0,0),(40,0.01),(80,0.015),(120,0.02),(160,0.025),(200,0.03),(240,0.035),(280,0.04),(320,0.045),(360,0.05),(400,0.055),(440,0.06),(480,0.065),(520,0.07),(600,0.075)

Coefficiente de Cansancio por Horas:

(1,0),(2,0.001),(3,0.002),(4,0.003),(5,0.004),(6,0.005),(7,0.006),(8,0.007),(9,0.008),(10,0.009),(11,0.01)

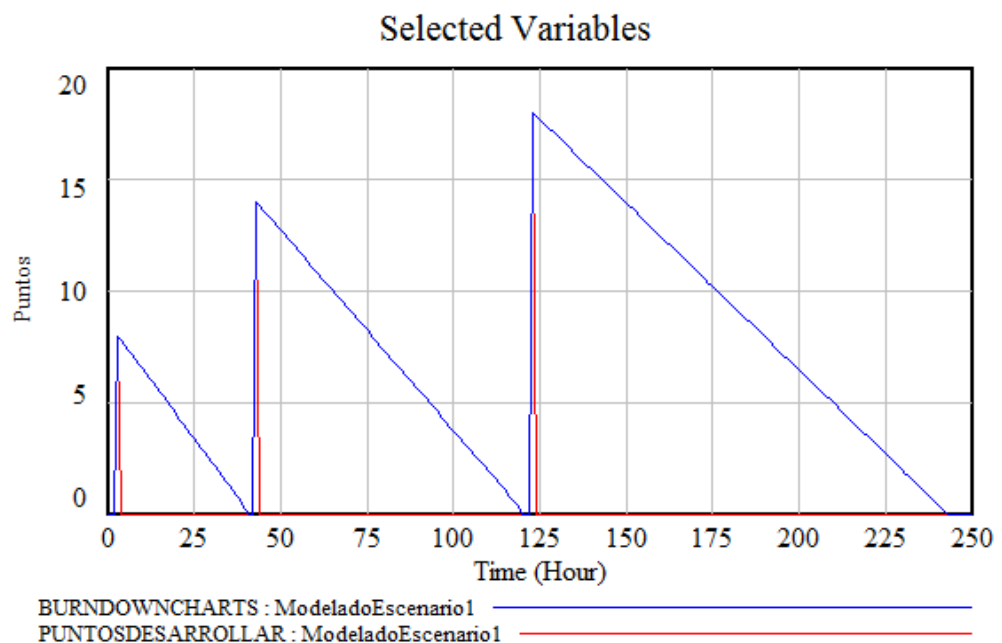
Horas Extras Trabajadas por Sprints: (1,0),(2,0),(3,0)

Resultados del Modelado de Escenario 1

En la siguiente figura se muestra los puntos de desarrollo e inicio de desarrollo de cada Sprint, por ello es que se puede visualizar tres líneas verticales con límites de altura independientes que vendrían a ser los puntos totales designados a cada Sprint, el primero se inicia en el tiempo 0 y con una altura de 8, el segundo se inicia en el tiempo 40 y su altura es de 14, por último el tercero se inicia en 120 y una altura de 18, se visualiza también que las líneas verticales son paralelas entre ellas con ciertas distancias entre las mismas que vendrían a ser el momento de inicio en el tiempo de cada Sprints, también se muestra la programación BURNDOWNCHARTS, que se simboliza con los tres picos cada una vendría a ser un Sprint.

Figura 32

Modelado Escenario1 variables BurndownCharts y puntos desarrollar



Nota: elaboración propia

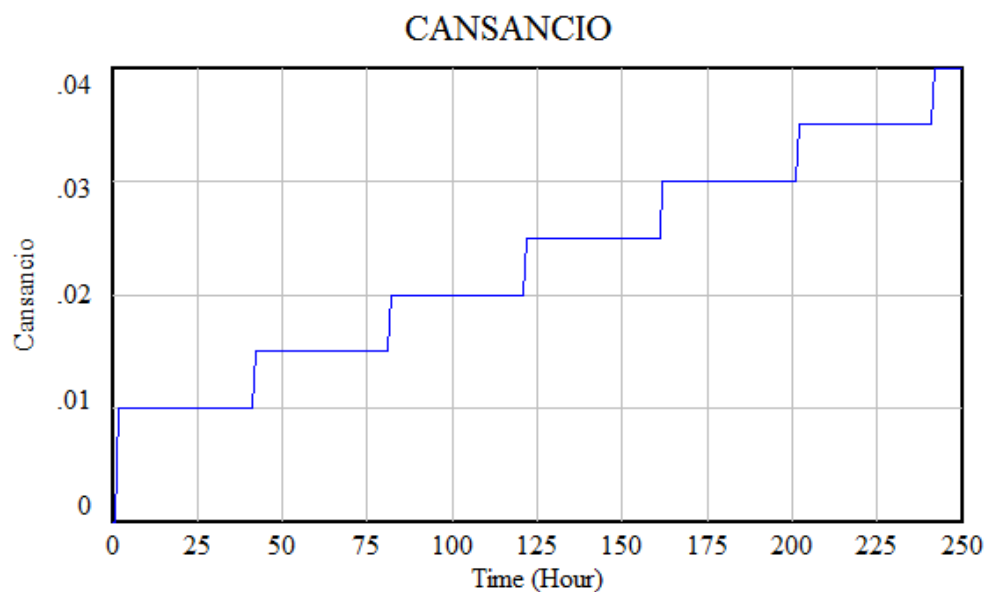
El cansancio que se genera sobre el equipo de trabajo es acumulativo y progresivo. Por los diferentes promedios de tiempos es

que el cansancio se va haciendo cada vez mayor, por ello que hasta la hora 40 de trabajo se acumula un total de 0.01 de cansancio, hasta el momento 80 se incrementa a 0.015 de cansancio y sigue acumulándose progresivamente cada 40 horas de trabajo. El cansancio definitivamente afecta en el proyecto y desarrollo de las diferentes tareas.

La figura de la variable cansancio se muestra a continuación:

Figura 33

ModeladoEscenario1 Variable Cansancio



CANSANCIO : ModeladoEscenario1

Nota: elaboración propia

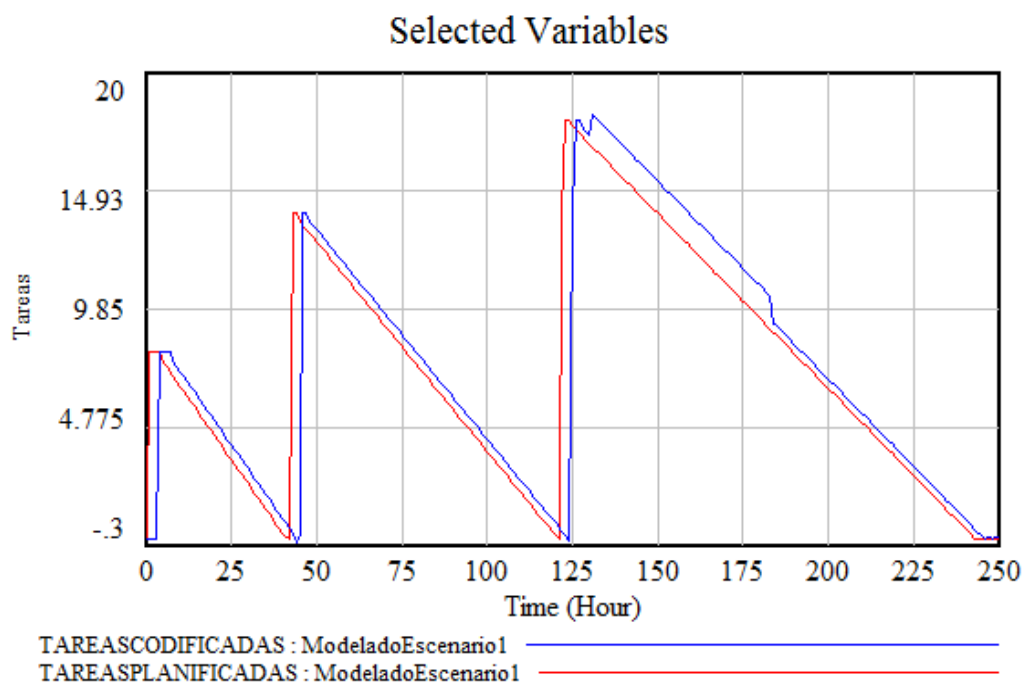
En la siguiente figura se muestra el total de Sprints específicamente las tareas planificadas como la variable TAREASPLANIFICADAS, juntamente con las tareas codificadas como la variable TAREASCODIFICADAS, esto con la influencia del cansancio del equipo de trabajo. Vemos en la figura la magnitud en que afecta al desarrollo de las tareas con el cansancio del Team en el proyecto.

Se aprecian tres picos las cuales indican las tareas planificadas, para el pico 1 que sería el Sprint 1 se tiene programada 8 tareas y se debería terminar en 40 horas, para el pico 2 que representa el Sprint 2 se tiene una programación de 14 tareas y su culminación de estas en 80 horas desde la culminación del primer sprint y por último el pico 3 que representa el sprint 3 con un total de 18 tareas con un total de 120 horas para su culminación, se visualiza estas sin ninguna variación.

Se aprecia también tres picos algo diferentes al anterior descrito estas representan las tareas codificadas, esta variación se da ya que el cansancio afecta en esta magnitud el desarrollo de tareas, puesto que la franja que sobresale en el pico 3 o Sprint 3, son la representación de tareas con errores a recodificar y esta se suma al Sprint al que corresponde.

Figura 34

ModeladoEscenario1 Variables TAREASPLANIFICADAS y TAREASCODIFICADAS



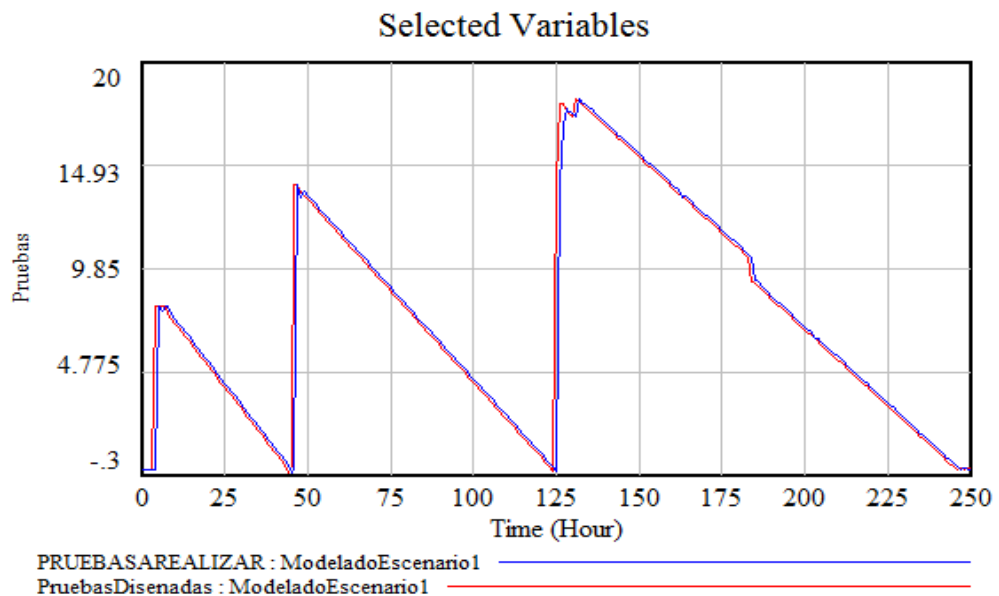
Nota: elaboración propia

Una vez que se tienen simulado el total de tareas codificadas pasan a las pruebas de codificación, en ellas tenemos la variable PruebasDiseñadas que representa el total de tareas codificadas y el total de pruebas por cada Sprint, por cada tarea se tiene una prueba, es por ello que se muestra que para el sprint 1 se programa 8 pruebas, para el sprint 2 se programa 14 pruebas y para el sprint 3 se programa 18 pruebas que al pasar unas horas de trabajo se agrega un promedio de 1 prueba, esto a raíz del total de tareas codificadas descrita de la anterior figura. Se muestra también las pruebas a realizar que dinámicamente es influida por el cansancio del equipo de trabajo. Esto en la figura siguiente se muestra:

Figura 35

ModeladoEscenario1 Variables PRUEBASAREALIZAR y

PruebasDiseñadas



Nota: elaboración propia

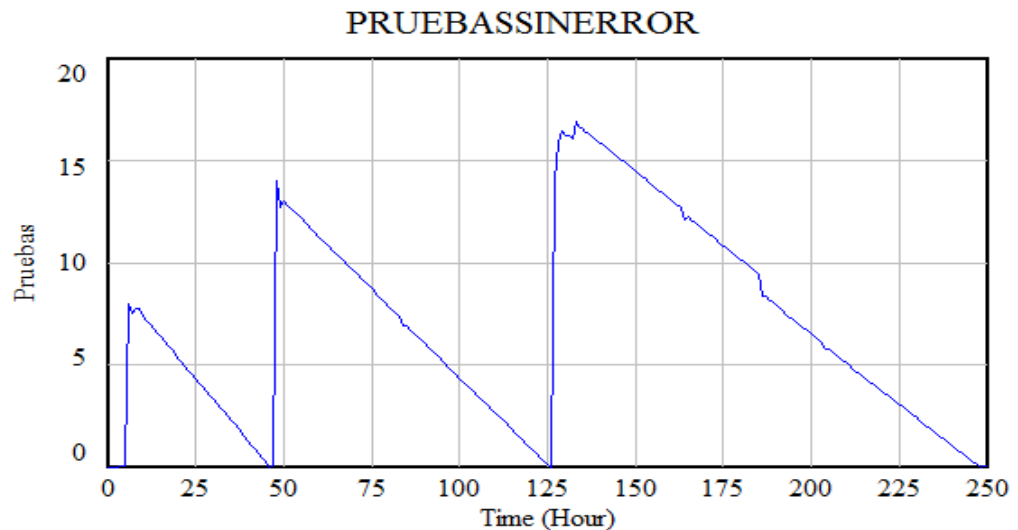
La figura siguiente nos muestra las pruebas de codificación sin errores detectados, se nos muestran los 3 picos o sprints, los dos primeros picos tienen ligeras variaciones, en el pico 1 entre el momento 6 y 8 se

visualiza una caída, el mismo que indica que la diferencia de pruebas que falta son parte de las pruebas con error de un 0.5 o la mitad de una tarea, por otro lado, en el pico 2 se visualiza una caída brusca de 1.3 pruebas y en el tiempo 51 se estabiliza y sigue su descendencia con normalidad hasta finalizar todas las pruebas del sprint, y por último el pico 3 que en la cima de ella se muestra un pico extra, esto se debe a que las pruebas totales de tareas de este sprint no todas resultaron sin error, esta diferencia se muestra en la figura 40 como pruebas con error detectados.

El cansancio del equipo de trabajo influye en estas pruebas y por ello tenemos en cada Sprint una cantidad justificada por el cansancio como errores de codificación.

Figura 36

ModeladoEscenario1 Variable PRUEBASSINERROR

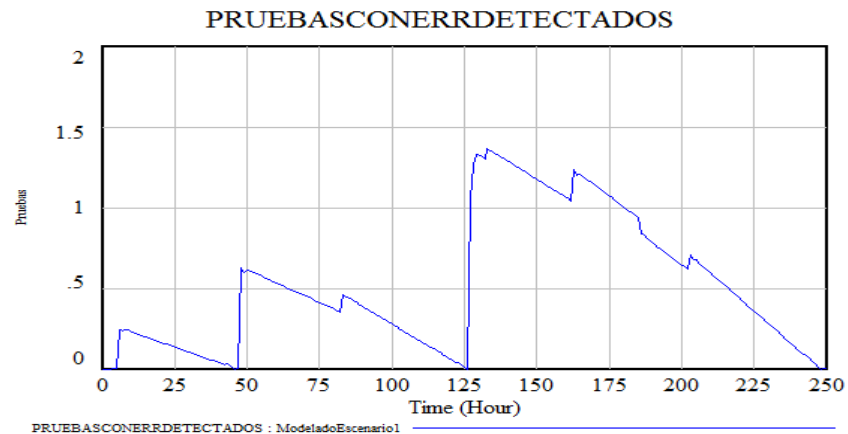


PRUEBASSINERROR : ModeladoEscenario1

Nota: elaboración propia

Figura 37

ModeladoEscenario1 Variables PRUEBASCONERRDETECTADOS



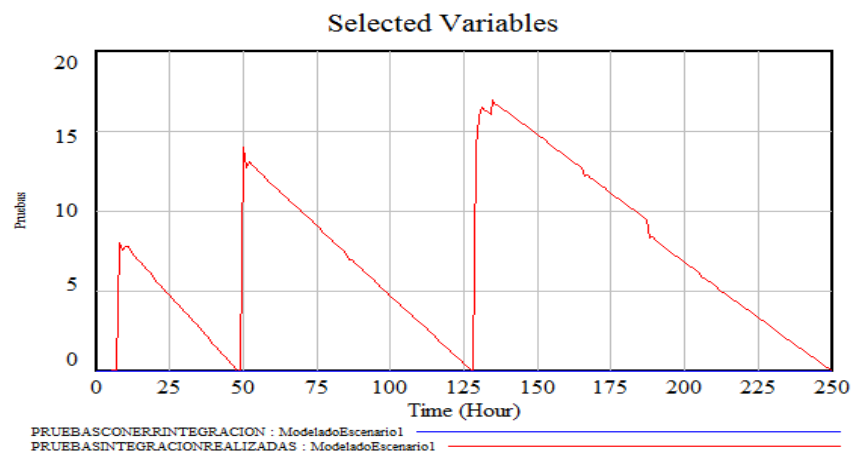
Nota: elaboración propia

Al tener las pruebas de codificación sin error, estas pasan a las pruebas de integración que también separan las pruebas con error detectados y pruebas de integración sin error, los 3 sprints nos muestran que la totalidad de pruebas de ejecución sin error son correctas al momento de hacer las pruebas de integración, así pues, el primer pico llega a un máximo de altura de 7.9, luego baja a 7.5 y se estabiliza posteriormente con 7.7 para continuar su avance de prueba.

Figura 38

ModeladoEscenario1 Variables PRUEBASCONERRINTEGRACION y

PRUEBASINTEGRACIONREALIZADAS



Nota: elaboración propia

En cuanto al segundo y tercer pico se muestran las pruebas con ciertas irregularidades tal y como llegan de las pruebas de ejecución, en cuanto a las pruebas de integración con error la figura 41 nos muestra una línea horizontal en todo momento con una equivalencia de 0 errores de integración a lo largo del proyecto.

4.3.1.3.2. Modelado 2: Presión en el Equipo de Trabajo

Los datos principales del proyecto “APLICACIÓN MULTIPLATAFORMA PANDA COMPUTER” son las siguientes:

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	352	352	10	0.0285	10	0
2	352	504	152	5	0.0328	5	0

Equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 2 Nivel Junior.

Horas Diarias: 8h.

Para realizar el modelado como escenario, los datos que también el modelo tomara son estas:

Número de Tareas que Equivale a Un Punto: 1

Pruebas por tarea: 1

Pruebas de Integración por Tarea: 1

Tasa de Pruebas de Error de Integración: 0

Presión de Plazo de Entrega en el Equipo de Trabajo:

(0,0.1),(0.1,0.1),(0.2,0.1),(0.3,0.15),(0.4,0.2),(0.5,0.25),(0.6,0.3),(0.7,0.4),
(0.8,0.5),(0.9,0.5),(1,0.5)

Probabilidad de Tareas Extras no Planificadas: 0



Horas Promedio Por Tareas Extras: 0

Tareas Extras por Sprints: (1,0), (2,0)

Tasa de Aprendizaje Seniors: 0.8

Nivel de Productividad Seniors:0.9

Tasa de Aprendizaje Juniors: 0.5

Nivel de Productividad Juniors: 0.5

Coefficiente de Cansancio Según el Tiempo:

(0,0),(40,0.01),(80,0.015),(120,0.02),(160,0.025),(200,0.03),(240,0.035),
(280,0.04),(320,0.045),(360,0.05),(400,0.055),(440,0.06),(480,0.065),(520,0.07),(600,0.075)

Coefficiente de Cansancio por Horas:

(1,0),(2,0.001),(3,0.002),(4,0.003),(5,0.004),(6,0.005),(7,0.006),(8,0.007),
(9,0.008),(10,0.009),(11,0.01)

Horas Por Día: 8

Horas Extras Trabajadas por Sprints: (1,0), (2,0)

Los coeficientes que se tomaron para el sub modelo de Cansancio del Equipo de Trabajo, son las mismas del anterior Modelado de Escenario 1 pero el encargado del modelado junto al Scrum Master tiene la posibilidad de tomar otros valores como coeficientes.

Para hacer un modelado con casos algo más complicados, se ejecuta el sub modelo de Experiencia del Equipo de Trabajo con las respectivas tasas de aprendizaje y nivel de productividad para los tipos de integrantes del grupo de trabajo ya sean Senior o Junior, información facilitada y procesada de los tres proyectos que se tomó de muestra para



el presente proyecto, adicionalmente para mayor valides, contrastadas con la información de tasas de aprendizaje y nivel de productividad de los antecedentes, ya que en estos proyectos se hicieron mayor investigación sobre este tipo de información.

También visualizaremos en qué medida el Sub Modelo de Presión en el Plazo de Entrega influye en este Modelado de Escenario.

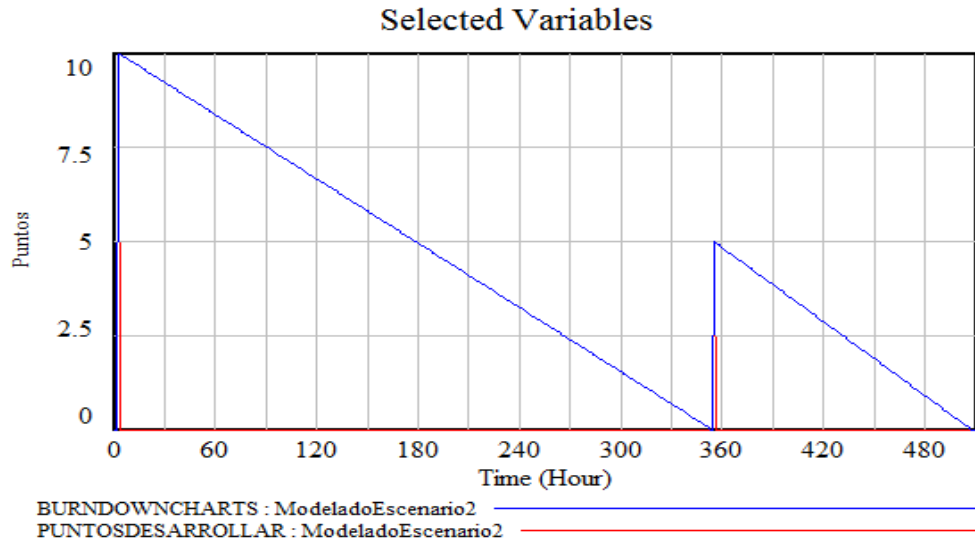
Resultados Modelado de Escenario 2

Ya que en este proyecto las tareas totales fueron desarrolladas dentro de dos Sprints, es por ello que en la siguiente figura los puntos de desarrollo e inicio de desarrollo de cada Sprints se muestra como la programación de BURNDOWNCHARTS, en forma de 2 picos, el pico 1 o sprint 1 se inicia en el tiempo 0, es por ello que a partir de esta se eleva verticalmente una línea hasta llegar al valor 10 del paralelo que mide los puntos a desarrollar y esta desciende progresivamente según el tiempo que tiene dicho sprint para su desarrollo y culminación, el segundo pico o spint 2 inicia con su ejecución en la hora 352 del desarrollo del proyecto, por esto se muestra que en este punto se levanta una línea vertical llegando al nivel de 5 puntos como total a lograr al desarrollar todas las tareas que sería en el momento 504 como se muestra en la figura, todo esto obtenida de la variable PUNTOSDESARROLLAR. Contamos con estos datos como datos base o iniciales.

Figura 39

ModeladoEscenario2 Variables BurndownCharts y Puntos

desarrollar

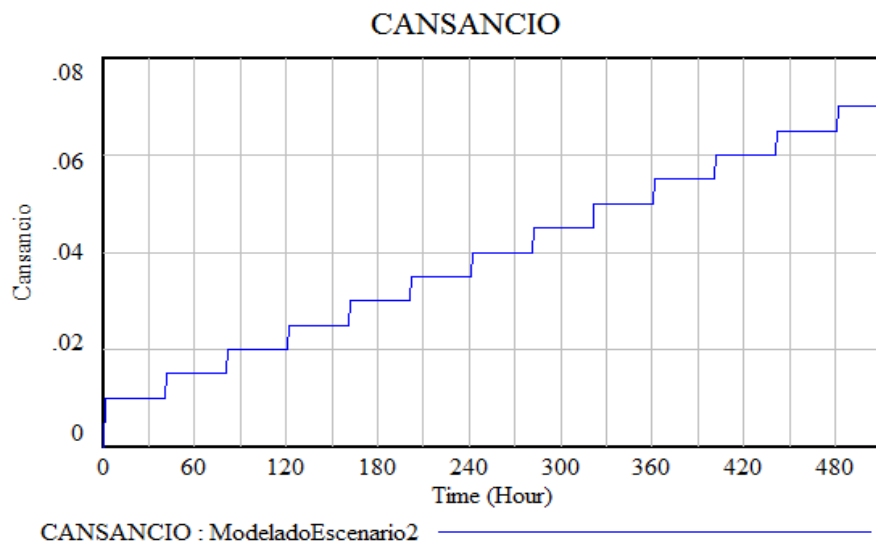


Nota: elaboración propia

Los datos de la variable cansancio son similares al primer modelado de escenario, con la diferencia de que en el actual modelado el tiempo del proyecto es superior y por ello el coeficiente de cansancio sigue subiendo progresivamente. La figura es la siguiente:

Figura 40

ModeladoEscenario2 Variable Cansancio



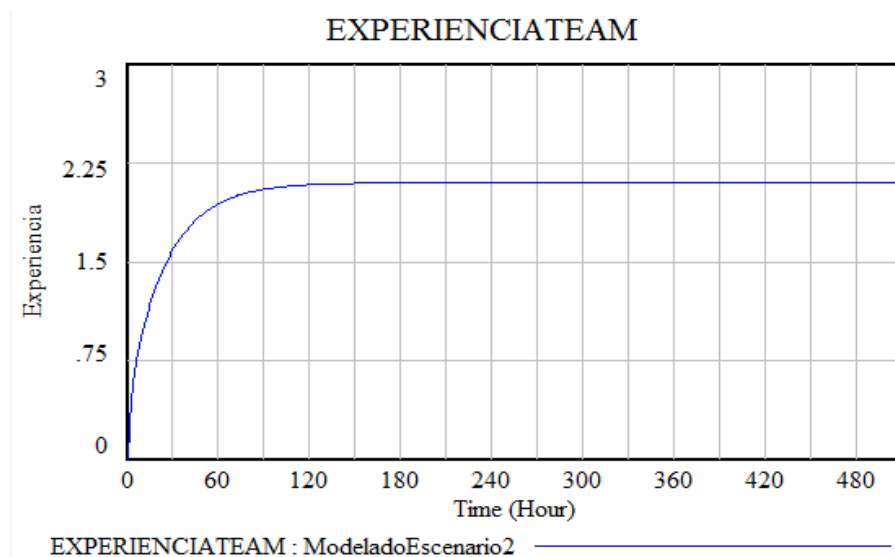
Nota: elaboración propia

Como ya anteriormente describimos que el sub modelo Experiencia del Equipo de Trabajo interactuara en el actual modelado de escenario.

La experiencia del equipo una vez iniciado el proyecto crece progresivamente según las tasas de aprendizaje definida, la experiencia como equipo se inicia en 0 en el siguiente momento que seria 2 la experiencia crece a 0.25 luego 0.43 y así continua creciendo hasta un promedio del momento 60 que llega la experiencia a 1.9, de este punto en posterior el incremento de experiencia es más calmada ya que el equipo está cerca de lograr la consolidación adecuada en el desarrollo de proyectos, no se visualizan oscilaciones en el grafico ya que no se hicieron incremento ni disminución de los integrantes del equipo. Y la figura es la siguiente:

Figura 41

ModeladoEscenario2 Experiencia del Equipo de Trabajo



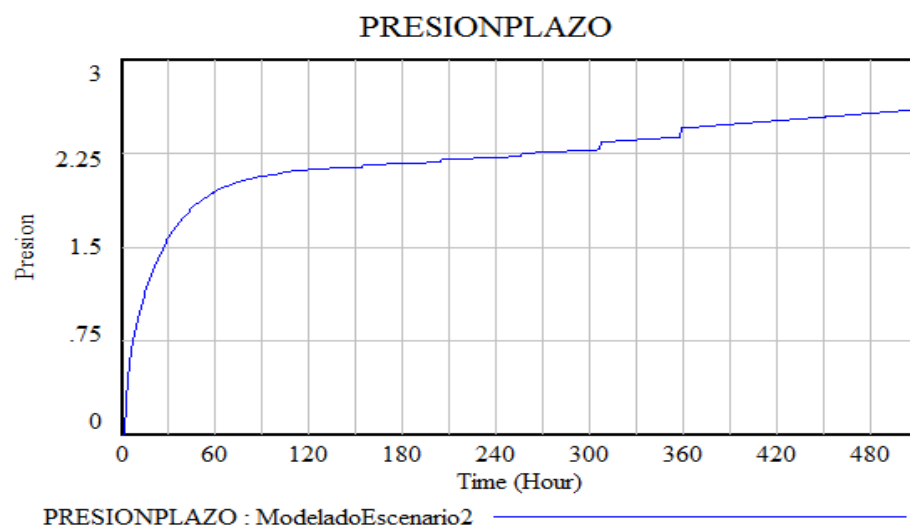
Nota: elaboración propia

La presión del plazo de entrega del proyecto que se produce en el equipo de trabajo es acumulativa esto a lo largo de todo el proyecto. El sub Modelo Presión en el Equipo de Trabajo es influida por los datos resultantes del sub modelo Experiencia del Equipo de Trabajo, dándonos la figura que a continuación se muestra, al principio del proyecto se inicia en 0, entre el tiempo 50 y 60 se visualiza una curvatura que normalmente indica el crecimiento de la presión a un ritmo menos acelerado ya que el equipo se empieza a consolidar, para el momento fin del proyecto la presión plazo logra llegar a 2.59 que equivaldría a algunos errores dentro del proyecto.

Como en todo proyecto mientras mayor sea la presión de obtener resultados, mayor será la probabilidad de que el equipo de trabajo tenga errores al cumplir con el trabajo.

Figura 42

ModeladoEscenario2 Variable PresionPlazo



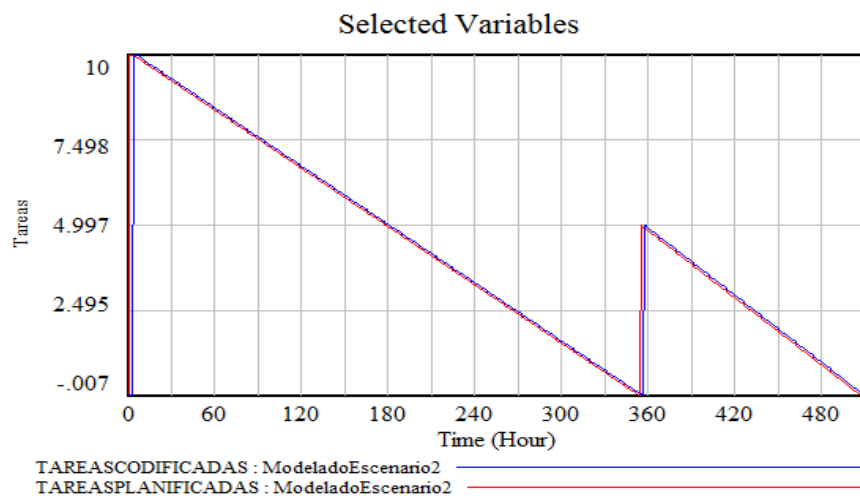
Nota: elaboración propia

Se aprecia en la siguiente figura el total de tareas planificadas como la variable TAREASPLANIFICADAS, juntamente con las tareas

codificadas como la variable TAREASCODIFICADAS, esto con la influencia del Cansancio del Equipo de Trabajo, Experiencia del Equipo de Trabajo y Presión Plazo. Vemos en la figura 2 picos el primero llega a un máximo de 10 tareas y su ejecución es la que se visualiza como la diagonal inclinada que llega o termina en el momento 352, del mismo se da un inicio hasta una altura de 5 tareas esto en el segundo pico culminando en el tiempo 504, estos picos muestran que no tienen muchas varianzas en el desarrollo de las tareas, esto se da por la contraparte de la Experiencia del Equipo de Trabajo al Cansancio y Presión.

Figura 43

ModeladoEscenario2 Variables TAREASPLANIFICADAS y TAREASCODIFICADAS



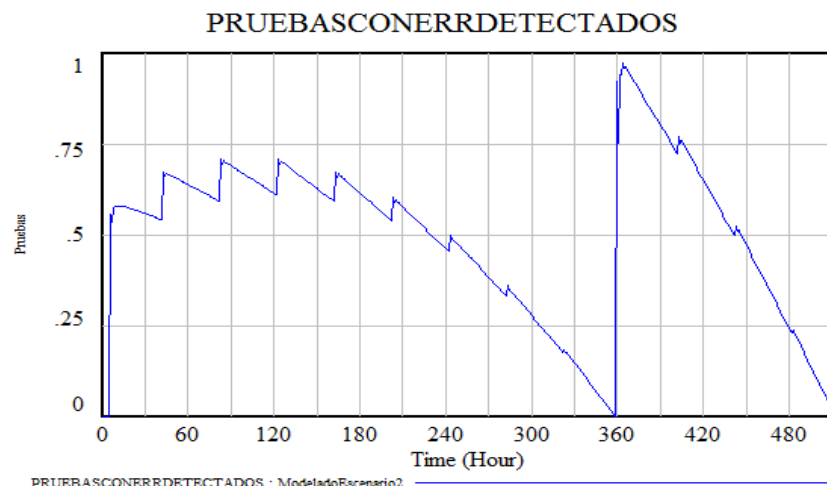
Nota: elaboración propia

En la figura 45 se aprecia la prueba de tareas codificadas con errores detectados, en ella se ven los dos picos como muestra de los 2 sprint, pero con ciertas irregularidades cada una, el primero se muestra con más de 8 picos que sobre salen de ella, esta muestra que según se avanza con las pruebas los errores están siendo detectadas y se representa como cada pico que sobre sale, por ejemplo, en el momento 163 se eleva

uno de los picos que nos indica que un 0.67 del total de tareas se detecta con error de codificación, la misma explicación reciben las irregularidades en cada sprint mostrada en la figura cambiando tiempo y magnitud de error detectado.

Figura 44

ModeladoEscenario2 Variables PRUEBASCONERRDETECTADOS

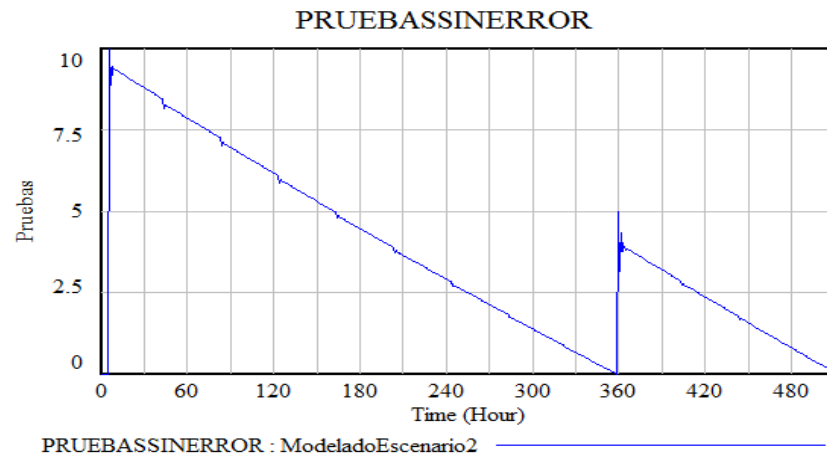


Nota: elaboración propia

En la figura siguiente se muestra las pruebas de codificación sin errores, el pico 1 se inicia en 10 pruebas según se avanza con las pruebas se observan pequeñas caídas que son la selección de las pruebas sin error de las que se detectaron con error, por ejemplo, en el momento 163 se nos muestra la diagonal a la altura de 4.97 de pruebas, indicándonos esta cantidad de pruebas sin error detectadas. Esta figura indica en qué magnitud el Cansancio y Presión en el equipo de trabajo influyen en el desarrollo de las tareas, por ello tenemos en cada Sprint una cierta cantidad de errores de codificación.

Figura 45

ModeladoEscenario2 Variables PRUEBASSINERROR



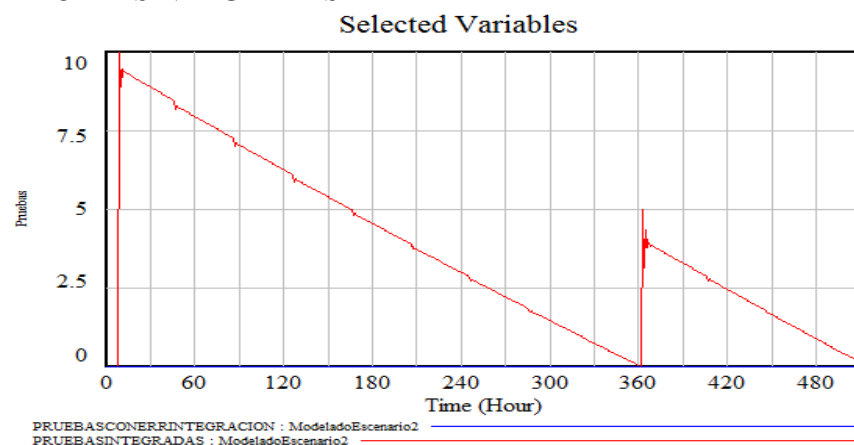
Nota: elaboración propia

Todas las tareas que pasaron como sin error por la prueba de codificación son las que necesariamente entran a la prueba de integración, y es justo el sub modelo de pruebas de integración la que ejecuta este proceso. En la figura siguiente se muestra el resultado de esta prueba, las tareas que se integraron correctamente sin error alguno y las tareas que tuvieron error en la integración, este último como se ve en la figura no presenta ninguna tarea con error de integración.

Figura 46

ModeladoEscenario2 Variables PRUEBASCONERRINTEGRACION y

PRUEBASINTEGRADAS



Nota: elaboración propia



4.3.1.3.3. Modelado 3: Probabilidad de Tareas Adicionales

Datos principales que se tomó del proyecto “APLICATIVO WEB ALPAMAYO CLIENT”, son las siguientes:

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	208	208	20	0.097	10	0
2	208	400	192	16	0.084	8	0
3	400	592	192	16	0.084	8	0
4	592	720	128	10	0.078	5	0

Equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 2 Nivel

Junior.

Horas Diarias: 8h.

Para realizar el modelado como escenario, los datos que también el modelo tomara son estas:

Número de Tareas que Equivale a Un Punto: 1

Pruebas por tarea: 1

Pruebas de Integración por Tarea: 1

Tasa de Pruebas de Error de Integración: 0

Presión de Plazo de Entrega en el Equipo de Trabajo:

(0,0.1),(0.1,0.1),(0.2,0.1),(0.3,0.15),(0.4,0.2),(0.5,0.25),(0.6,0.3),(0.7,0.4)
,(0.8,0.5),(0.9,0.5),(1,0.5)

Probabilidad de Tareas Extras no Planificadas: 0.05

Horas Promedio Por Tareas Extras: 5

Tareas Extras por Sprints: (1,0), (2,0), (3,0)



Tasa de Aprendizaje Seniors: 0.8

Nivel de Productividad Seniors:0.9

Tasa de Aprendizaje Juniors: 0.5

Nivel de Productividad Juniors: 0.5

Coefficiente de Cansancio Según el Tiempo:

(0,0),(40,0.01),(80,0.015),(120,0.02),(160,0.025),(200,0.03),(240,0.035),
(280,0.04),(320,0.045),(360,0.05),(400,0.055),(440,0.06),(480,0.065),(520,0.07),(600,0.075)

Coefficiente de Cansancio por Horas:

(1,0),(2,0.001),(3,0.002),(4,0.003),(5,0.004),(6,0.005),(7,0.006),(8,0.007),
(9,0.008),(10,0.009),(11,0.01)

Horas Por Día: 8

Horas Extras Trabajadas por Sprints: (1,0), (2,0), (3,0)

Los coeficientes que se tomaron para el sub modelo de Cansancio del Equipo de Trabajo, para el sub modelo de Experiencia del Equipo de Trabajo y sub modelo Presión del Plazo de Entrega, son las mismas del anterior Modelado de Escenario 2 pero el encargado del modelado junto al Scrum Master tiene la posibilidad de tomar otros valores como coeficientes.

Para hacer un modelado con casos algo más complicados, se ejecuta el sub modelo de Tareas Adicionales.

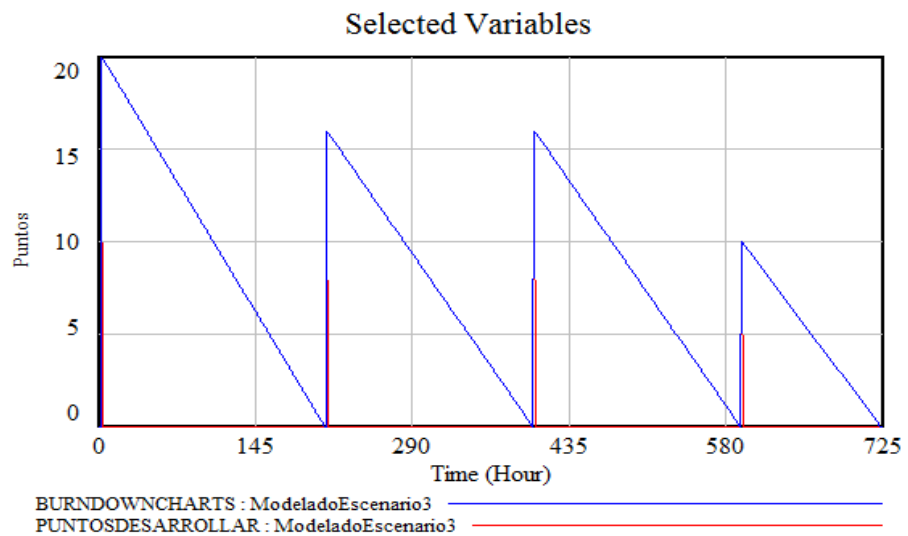
Resultados Modelado de Escenario 3

La siguiente figura nos muestra los puntos de desarrollo e inicio de desarrollo de cada Sprint, esto con las 4 líneas verticales paralelas y separadas entre sí por una representación de tiempo de inicio entre sprint

y sprint, la primera línea vertical se inicia en el tiempo 0 y se eleva hasta el nivel de puntos 20, el segundo se inicia en 208 del tiempo y llega al nivel de puntos 16, el tercero se inicia en 400 del tiempo y llega al nivel de puntos 16 y por último la cuarta línea vertical se inicia en 492 del tiempo y se eleva hasta 10 puntos, esto se muestra con la variable PUNTOSDESARROLLAR. También se muestra la programación BURNDOWNCHARTS, como 4 picos que se muestran una tras otra representando su continuidad como el inicio y fin del desarrollo del sprint. Contamos con estos datos como datos base o iniciales.

Figura 47

ModeladoEscenario3 Variables BurndownCharts y Puntos desarrollar

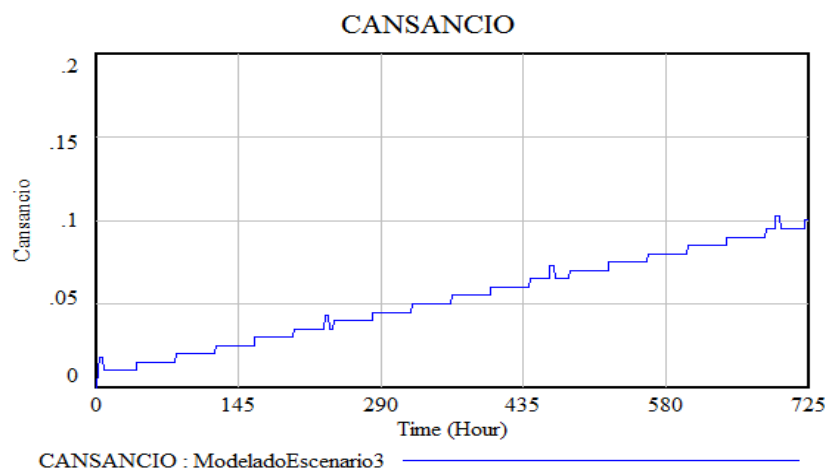


Nota: elaboración propia

Los datos de la variable cansancio en este modelado tiene el tiempo más largo a comparación de los anteriores modelados y por ello el coeficiente de cansancio sigue subiendo progresivamente, se puede visualizar en diferentes momentos del tiempo ligeras elevaciones del cansancio, esto debido a las tareas extras que se desarrollaron en estos momentos donde se las visualiza.

Figura 48

ModeladoEscenario3 Variable CANSANCIO

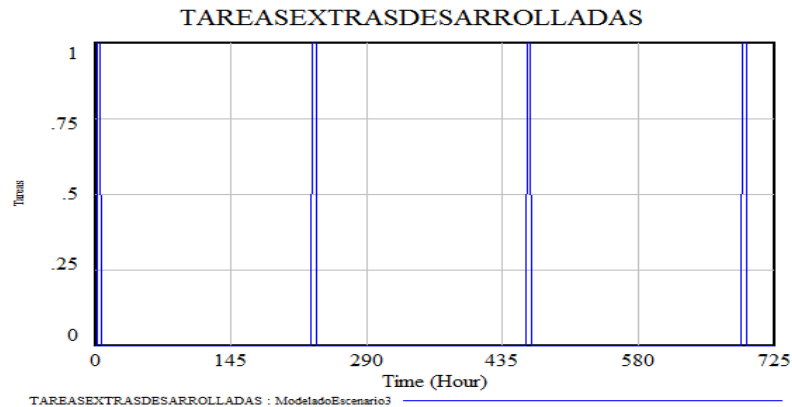


Nota: elaboración propia

El sub Modelo de Tareas Extras a través de la variable probabilidad de Tareas Extras no Planificadas, teniendo el valor de un promedio como posibilidad, es que resulta la siguiente figura que muestra 4 líneas verticales que simbolizan las tareas extras que se tendrán que desarrollar, el primero surge para el primer sprint en el momento 2 a una altura de 1 tareas, el segundo surge en el momento 232 a una altura de 1 tarea y en similar situación surgen las dos últimas tareas extras que se adicionan a los dos últimos sprints.

Figura 49

ModeladoEscenario3 Variable TAREASEXTRASDESARROLLADAS



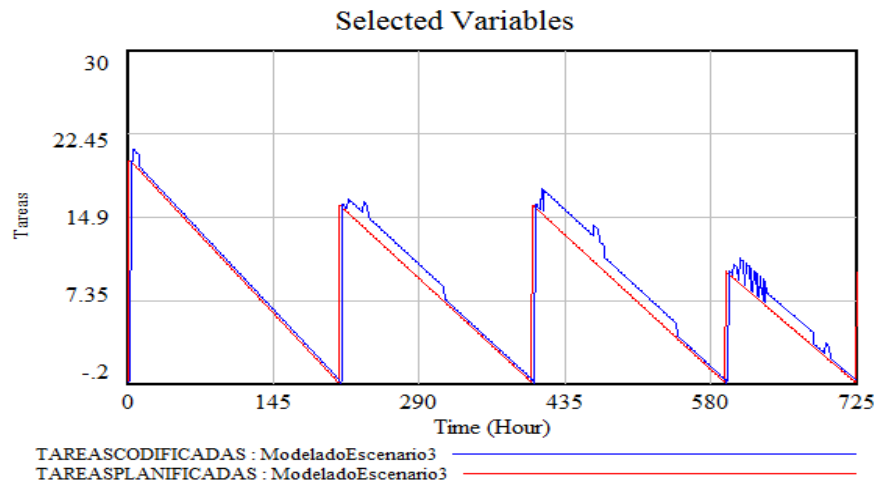
Nota: elaboración propia

Los sub modelos Cansancio, Experiencia, Presión de Plazo y Tareas Extras ejecutadas sobre el caso actual nos da los siguientes resultados en las variables TAREASCODIFICADAS y TAREASPLANIFICADAS.

Las tareas codificadas han incrementado en comparación a las planificadas, y en algunos casos se observa que hay elevaciones que sobresalen al momento de codificar tareas, el cansancio, presión plazo y tareas extras son las que influyeron para estos resultados en el modelado, por ello en el pico 1 se nos muestra un sub pico que sobresale en el momento 6 y llega al nivel de tareas hasta 21 que indica que en el sprint 1 se adiciona 1 tarea, similar caso ocurre en los otros tres picos o sprints.

Figura 50

*ModeladoEscenario3 Variables TAREASCODIFICADAS y
TAREASPLANIFICADAS*



Nota: elaboración propia

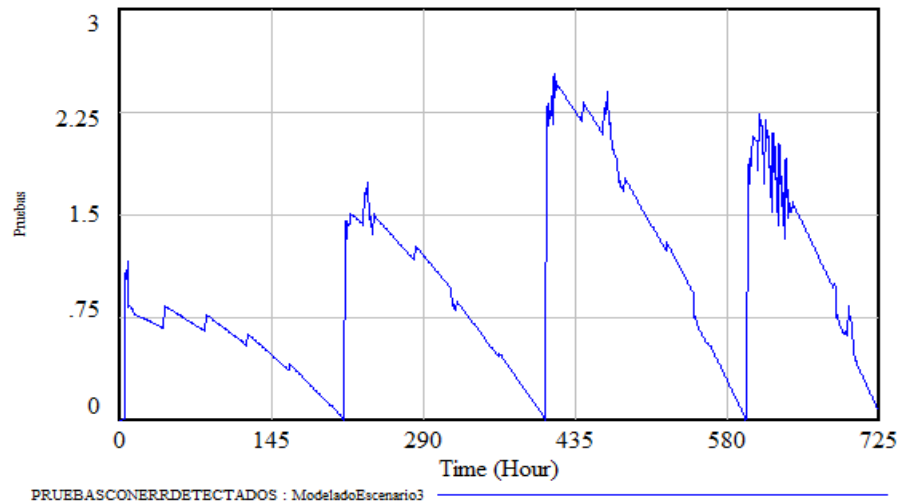
En las dos figuras siguientes se aprecian la prueba de codificación con errores detectados y las pruebas de codificación sin errores. Podemos ver en qué magnitud el Cansancio, Presión en el equipo de trabajo y Tareas extras, influyen en el desarrollo de las tareas, por ello tenemos en cada Sprint una cierta cantidad de errores de codificación, en la figura 54 podemos apreciar el pico 3 con elevaciones y caídas distorsionadas que simbolizan los errores detectados en el tercer sprint, es por ello que en el tiempo 605 se ve una elevación que llega a 2.07 como el total de pruebas con error detectadas, en otro momento como es 637 la elevación llega a 1.92 como máximo de pruebas de tareas con error detectadas, similares casos ocurren en los otros picos que representan a los sprints en desarrollo.

En la figura 55 se nos muestra la diferencia del total de tareas a desarrollar con las pruebas de tareas desarrolladas con error detectadas, como las pruebas de tareas desarrolladas sin error detectadas, al principio

del pico 3 se denota varias altas y bajas que simbolizan que en estos puntos se detectaron errores.

Figura 51

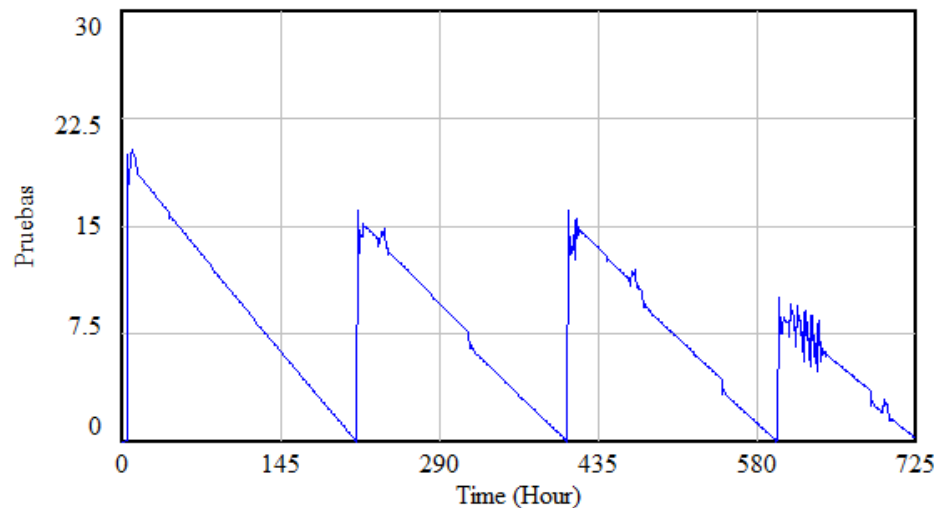
ModeladoEscenario3 Variables PRUEBASCONERRDETECTADOS
PRUEBASCONERRDETECTADOS



PRUEBASCONERRDETECTADOS : ModeladoEscenario3
Nota: elaboración propia

Figura 52

ModeladoEscenario3 Variables PRUEBASSINERROR
PRUEBASSINERROR



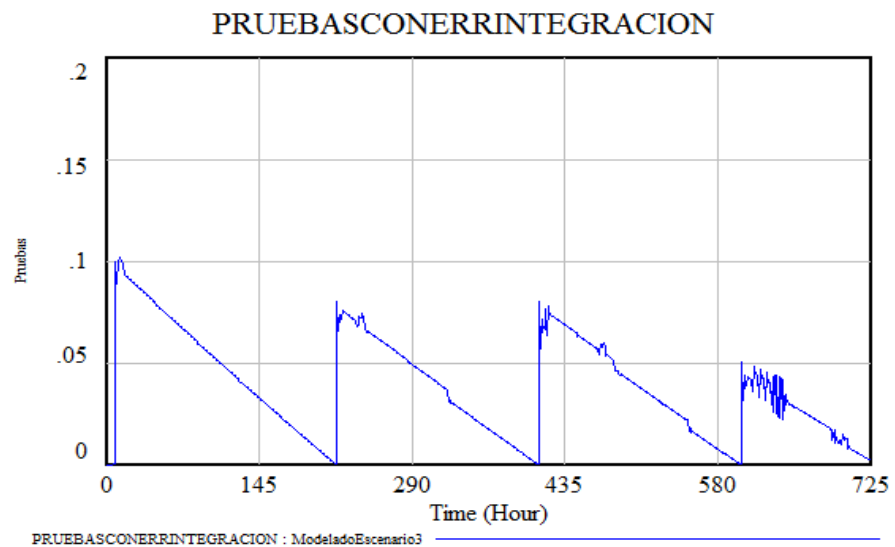
PRUEBASSINERROR : ModeladoEscenario3
Nota: elaboración propia

Tenemos en la siguiente figura resultados de la prueba de integración en el caso de las pruebas que resultaron con error, la tasa de error de integración nos da por cada Sprint una cierta cantidad de pruebas con error, en los 4 sprints se detectaron errores de integración por ello los cuatro picos, el primer pico nos indica que en el momento o tiempo 11 se llega a detectar 0.1 de pruebas con error, en el pico 2 el punto máximo al que llega verticalmente es 0.08 como pruebas de integración con error detectados esto en el tiempo 219, similares son los casos el 2 últimos picos. Todo esto en la figura que se muestra a continuación:

Figura 53

ModeladoEscenario3 Variables

PRUEBASCONERRDEINTEGRACION



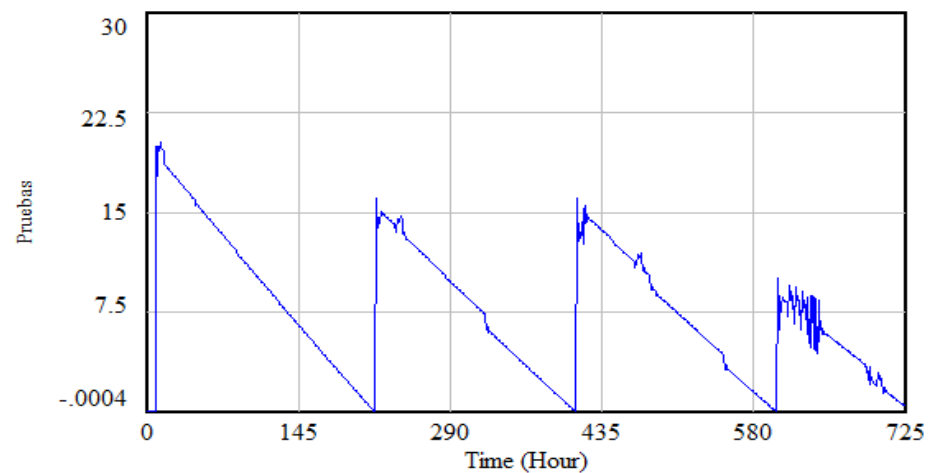
Nota: elaboración propia

Por otro lado, tenemos resultados de las pruebas de integración con resultados correctos sin algún error, son llamadas pruebas integradas, esto se muestra en la figura 55, en la que se visualizan los 4 picos o sprints desarrollados, las mismas que fueron puestas a prueba en la etapa de integración, dentro de estos picos haciendo la interpretación nos muestra

el 3 que en el tiempo 411 se inicia con su prueba por ello se desde este momento se eleva el pico hasta el total de pruebas 16, pero en su decreción como muestra del avance en el tiempo nos muestra altas y bajas que indican que solo en esa magnitud o medida se integraron sin errores.

Figura 54

ModeladoEscenario3 Variables PRUEBASINTEGRADAS
PRUEBASINTEGRADAS



PRUEBASINTEGRADAS : ModeladoEscenario3

Nota: Elaboración: Propia

Hasta este punto se concluye con las pruebas de modelado de escenario, logrando obtener resultados diferentes que afectan principalmente al correcto desarrollo de tareas programadas y mostrándolas en las diferentes pruebas que se les aplica dentro del proyecto, mostrándonos también la dinámica con la que los casos influyen dentro del modelo construido, vemos por ejemplo el escenario del agotamiento del equipo y su influencia mostrada en errores de ejecución de tareas, vemos también la presión en el equipo de trabajo y una probabilidad de tareas adicionales no planificadas. Las figuras que se mostraron en cada caso son resultados de la ejecución de la simulación de modelado de cada escenario debidamente acompañada con una descripción propia.

4.3.2. Validación Estadística

Según las investigaciones de Barlas sobre la validación de modelos de simulación, un modelo se considera legítimo si su tasa de error es inferior al 5%, en consecuencia, para lograr validar el modelo estadísticamente se muestra los resultados de la corrida simulada de los tres proyectos tomadas como muestra.

4.3.2.1. Modelado 1: “SISTEMA DE PLANIFICACIÓN DE RECURSOS”

Proyecto que para su desarrollo se aplica el marco de trabajo Scrum. Su objetivo es optimizar la administración e integración de información entre las diferentes áreas de la empresa Peruvian Computer.

En este proyecto se puede apreciar las características de scrum desarrollado, analizado e implementado. El tiempo estimado para la conclusión del proyecto es de seis (6) semanas, con un total de tres Sprints y sus diferentes cantidades de tareas propias.

Datos Necesarios

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	40	40	8	0.21	8	0
2	40	120	80	14	0.175	14	0
3	120	240	120	18	0.15	18	0

Equipo de Trabajo: 3 personas, 1 Scrum Master y 2 Team, 1 Nivel Junior y 2 Nivel Senior.

Horas Diarias: 8h.

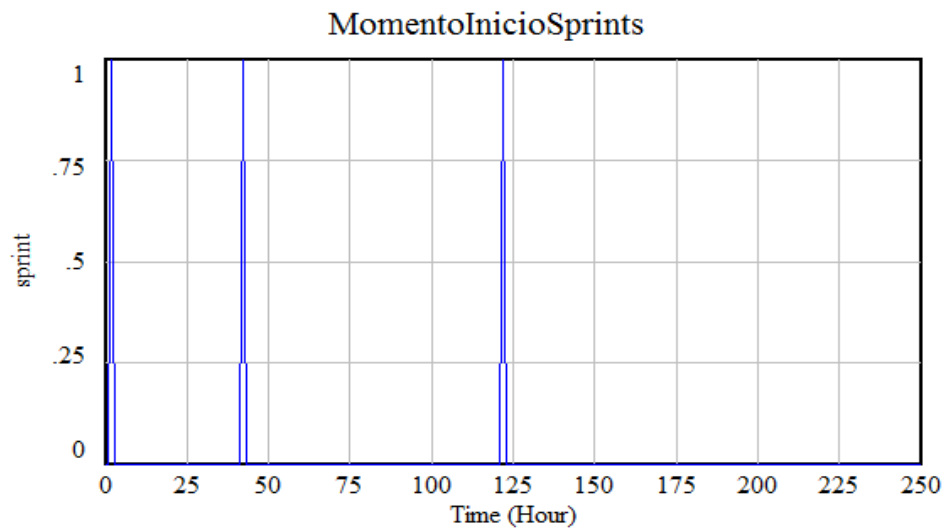
Principales Resultados del Modelado 1

Para dar inicio al modelado tenemos el sub modelo de planificación, donde variables como MomentoInicioSprints, VelocidadIdealSprints, PuntosSegunSprints y algunas otras son las que hacen la dinámica inicial.

El grafico siguiente nos muestra los puntos de inicio para cada Sprint, mostrándonos tres líneas verticales que se inician en diferentes tiempos, como es el primero en el tiempo 0, el segundo en el tiempo 40 y el tercero en el tiempo 120, indicándonos con estos el tiempo en el que se programa el inicio de cada Sprint, siendo estos de suma importancia para los posteriores sub modelos, ya que a partir de estos puntos se dará inicio la ejecución de desarrollo por Sprint.

Figura 55

ValidacionModelado1 Variable Momento Inicio Sprints



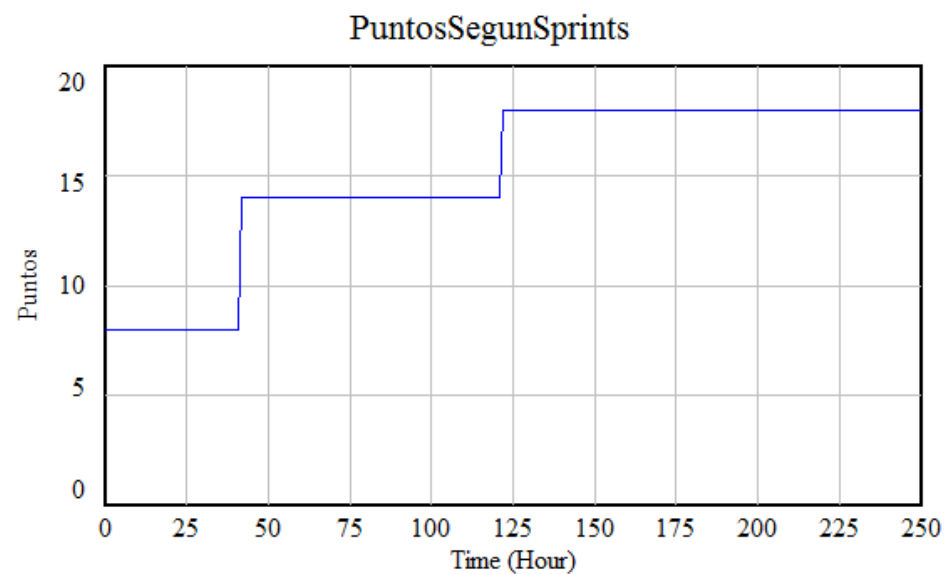
MomentoInicioSprints : ValidacionModelado1

Nota: elaboración propia

La ejecución y desarrollo de cada Sprint, tiene designado un total de puntos que tienen que ser completados, según el equipo team avance con un determinado Sprint, en la figura 57 podemos visualizar tres niveles diferentes, estas representan los 3 sprints, el primero por ejemplo nos indica que se inicia en el tiempo 0 y culmina en el tiempo 40 y el total de puntos por el desarrollo completo de este sprint es de 8, el segundo escalos vendría a ser el segundo sprint que se inicia en 40 y culmina en 120 del tiempo y los puntos asignados por todo el sprint es de 14, y similar es la interpretación del último escalón. La grafica de los puntos por Sprint es la siguiente:

Figura 56

ValidacionModelado1 Variable Puntos Según Sprints



PuntosSegunSprints : ValidacionModelado1

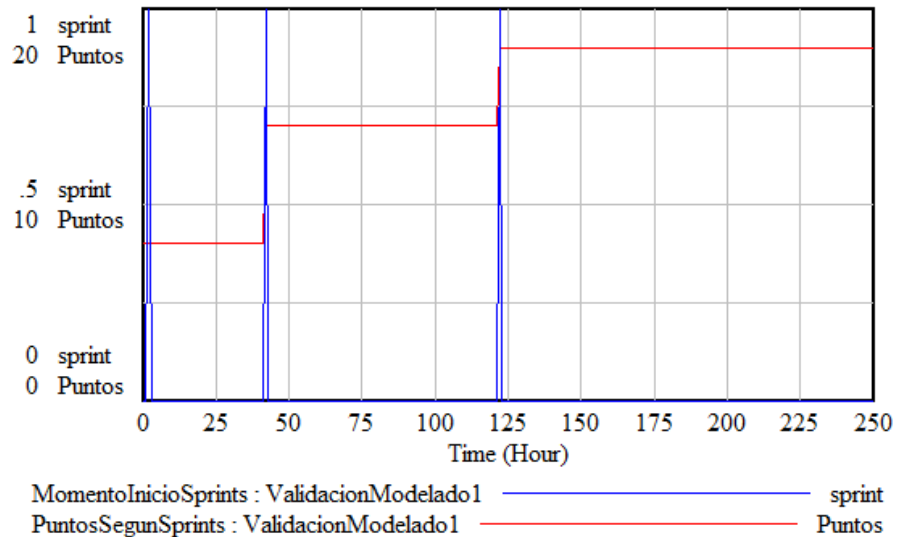
Nota: elaboración propia

En la figura 59 se visualizan tres líneas verticales que representan el momento en el tiempo del inicio y momento en el tiempo de finalización del desarrollo de un determinado sprint, también se visualizan tres líneas horizontales distribuida consecutivamente en diferentes niveles que

simboliza los puntos que le corresponden lograr al equipo por cada sprint culminada.

Figura 57

*ValidacionModelado1 Momentos de Inicio y Puntos por Sprints
Selected Variables*

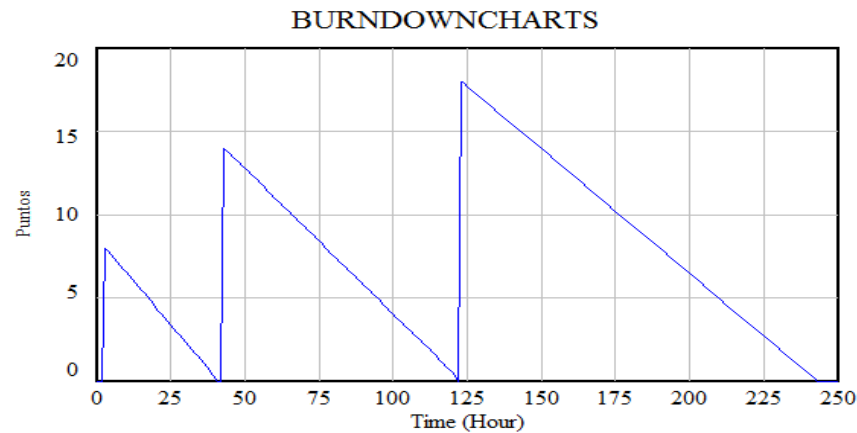


Nota: elaboración propia

El BURNDOWNCHARTS como ya sabemos es el gráfico de trabajo pendiente a lo largo del tiempo, que muestra la velocidad a la que se está completando los objetivos/requisitos, en la figura 59 se nos muestra tres picos, el primero que representa al primer Sprint se levanta como punto de inicio en el nivel tiempo en 0 y alcanza verticalmente al nivel Puntos 8 y una vez allí se visualiza una caída diagonal que simboliza el progreso del desarrollo del sprint 1 con relación al tiempo y puntos logrados, similares son los casos para los consiguientes sprints, con las diferencias de puntos a lograr y el tiempo en el que se desarrolla un respectivo sprint. El grafico siguiente es el correspondiente al actual modelo.

Figura 58

ValidacionModelado1 Variable BURNDOWN CHARTS



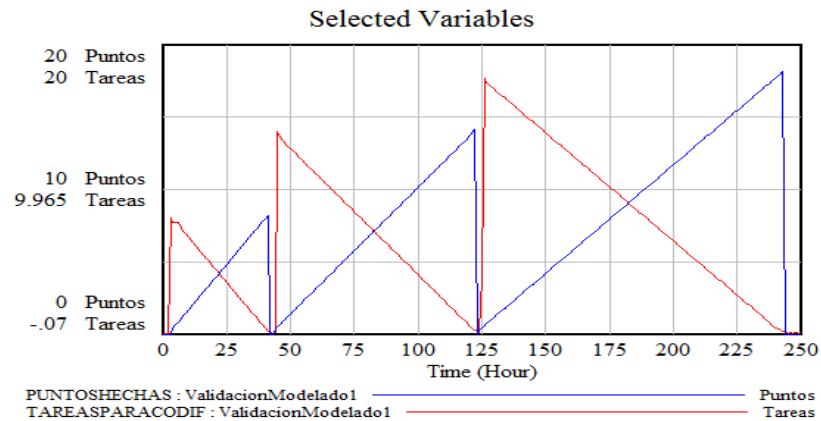
BURNDOWNCHARTS : ValidacionModelado1

Nota: elaboración propia

Cada punto histórico de este modelo tiene una tarea asignada, y no se muestra ninguna circunstancia que impida que las tareas y los puntos asignados progresen al ritmo de crecimiento predeterminado. La figura siguiente nos muestra los picos inclinados para el lado izquierdo son las Tareas para codificar y los picos que se inclinan para el lado derecho son los puntos hechos, si teníamos en el primer sprint 8 tareas entonces se realizó los 8 por ello los picos son similares en alturas y sus diagonales. El avance de las tareas y los puntos planificados pueden observarse en la siguiente figura:

Figura 59

ValidacionModelado1 Puntos y Tareas por Sprints

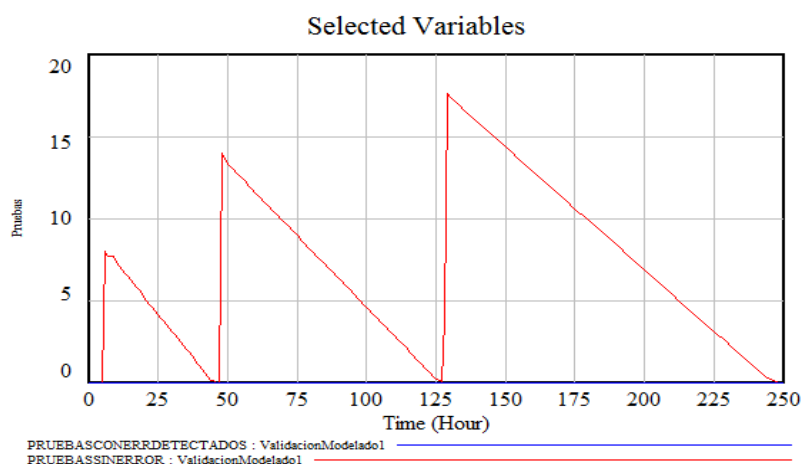


Nota: elaboración propia

El grafico de las pruebas de tareas realizadas con errores detectados y sin errores es la que se muestra debajo en la figura 61. Los tres picos que se nos muestran son la imagen de las pruebas realizadas a cada sprint, en ellas no se aprecian oscilaciones, caídas u otras irregularidades, se interpreta este resultado como correctas todas las tareas sometidas a la prueba. Ya que en este caso no se tuvo ningún tipo de complicaciones ni retrasos, por ello se muestra en cero en todo el proyecto la variable PRUEBASCONERRDETECTADOS.

Figura 60

ValidacionModelado1 Pruebas con Errores y Pruebas sin Error



Nota: elaboración propia

4.3.2.2. Modelado 2: “APLICACIÓN MULTIPLATAFORMA PANDA COMPUTER”

Este proyecto que fue desarrollado para la empresa PANDA COMPUTER S.R.L. fue ejecutado aplicando el marco de trabajo Scrum, orientándola para la administración de las diferentes compras, ventas y manejo de almacén.

El proyecto tiene 5 Sprints, de las cuales el Sprint 0 y Sprint 1 fueron designados para la etapa Análisis, el Sprint 2 para diseño, los Sprints 3 y 4 son las que el presente modelo considerará ya que dichos Sprints fueron destinados para la programación, desarrollo y evaluación de todos los componentes y funcionalidades. Estos dos Sprints tuvieron programados hasta su culminación un total de 3 meses y 3 días. Se ha trabajado 8 horas diarias y en la cual una semana se considera 5 días hábiles. Y el Sprint 5 para la puesta en marcha en las sucursales.

Datos Necesarios

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	352	352	10	0.0285	10	0
2	352	504	152	5	0.0328	5	0

Equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 1 Nivel señor y 1 Nivel Junior.

Horas Diarias: 8h.

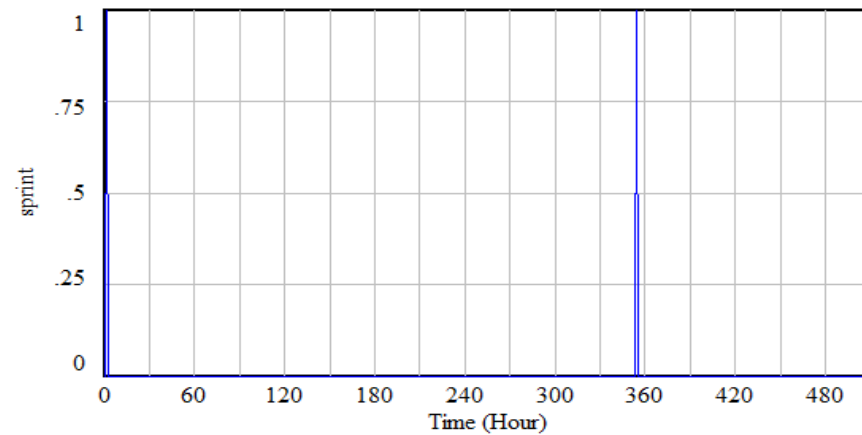
Principales Resultados del Modelado 2

Teniendo dos Sprints con sus respectivos datos necesarios para hacer el modelado, tenemos también las figuras siguientes con las diferentes variables plasmadas.

En la siguiente figura 62 podemos ver la representación del momento de inicio de cada Sprint a lo largo del proyecto, el primero se muestra al principio del cuadro que las contiene, esto como una línea vertical que se inicia en el momento 0, el segundo se visualiza en el momento 352, mostrándonos los momentos en el tiempo en el que se da inicio el desarrollo de cada sprint.

Figura 61

ValidacionModelado2 Variable Momento Inicio Sprints
MomentoInicioSprints



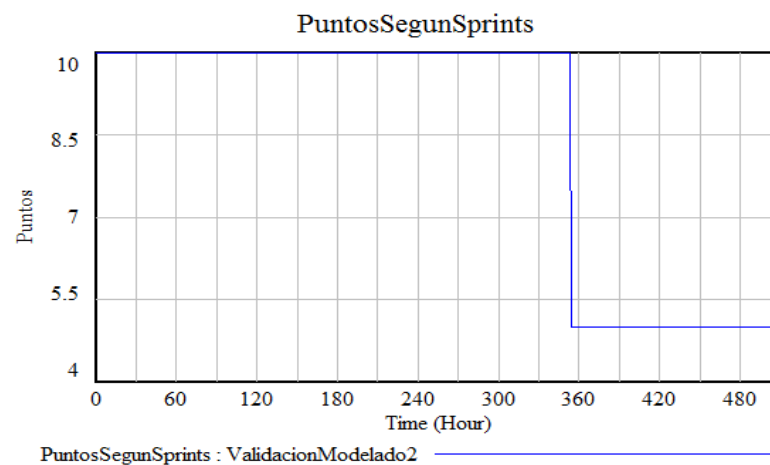
MomentoInicioSprints : ValidacionModelado2

Nota: elaboración propia

Los dos Sprints que este proyecto tiene y que están siendo modelados, tienen asignado una cierta cantidad de puntos que se completan al desarrollar por completo el determinado Sprint, mostrándonos esto en la figura 63, como dos escalones una que se visualiza en lo más alto del cuadro que las contiene y el segundo a una distancia baja de la base del cuadro, el primer escalón indica que dentro del primer sprint se tiene un total de 10 puntos a lograr con su desarrollo completo, el segundo escalón indica que 5 en es el total de puntos a lograr con el desarrollo completo del segundo sprint. La siguiente es la figura que la representa:

Figura 62

ValidacionModelado2 Variable Puntos Según Sprints



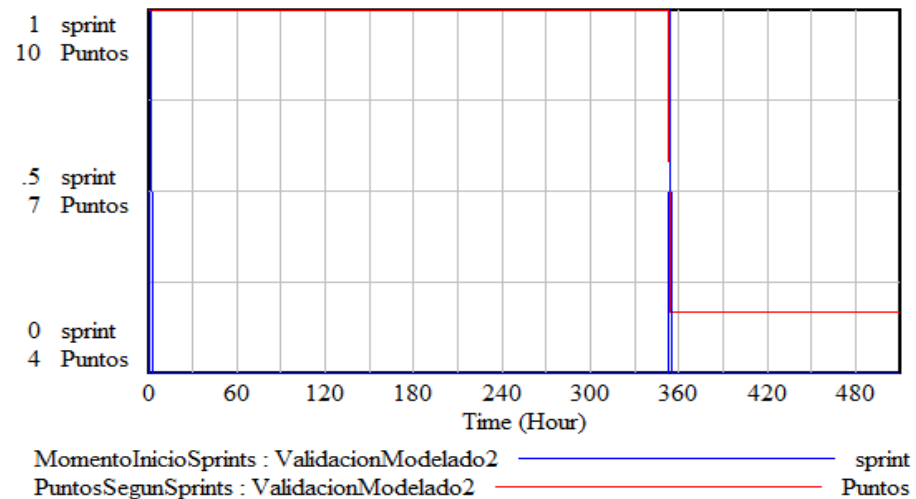
Nota: elaboración propia

En la figura 64 se visualiza dos escalones de diferentes tamaños esto representando por la altura en la que se encuentran, los puntos que le corresponde a un determinado sprint al ser culminado con su desarrollo, lo que separa estas dos representaciones que las separa son líneas que se levantan verticalmente en determinados momentos del tiempo, el primero

se inicia en 0 del tiempo y el segundo en el momento 352 del tiempo, todo esto se muestra en la figura siguiente:

Figura 63

ValidacionModelado2 Momentos de Inicio y Puntos por Sprints
Selected Variables

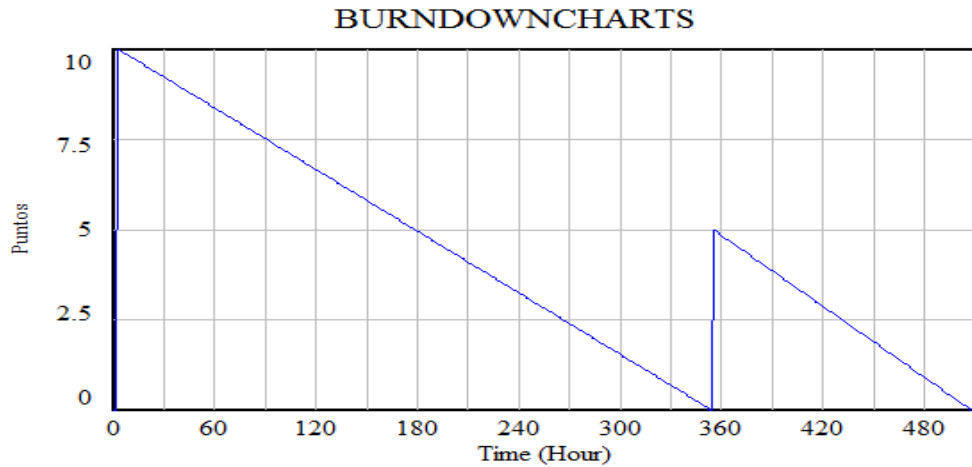


Nota: elaboración propia

La figura siguiente es la de BURNDOWNCHARTS que vendría a ser el gráfico de trabajo pendiente a lo largo del tiempo, que muestra la velocidad a la que se está completando los Sprints, se visualizan dos figuras en forma de picos una más grande que la otra, esto por los puntos programados por cada sprint, por ello el primer pico se eleva a una altura de 10 puntos y el segundo a 5 puntos, iniciando y culminando cada pico como se programó. La figura siguiente es el correspondiente al actual modelo.

Figura 64

ValidacionModelado2 Variable BURNDOWN CHARTS

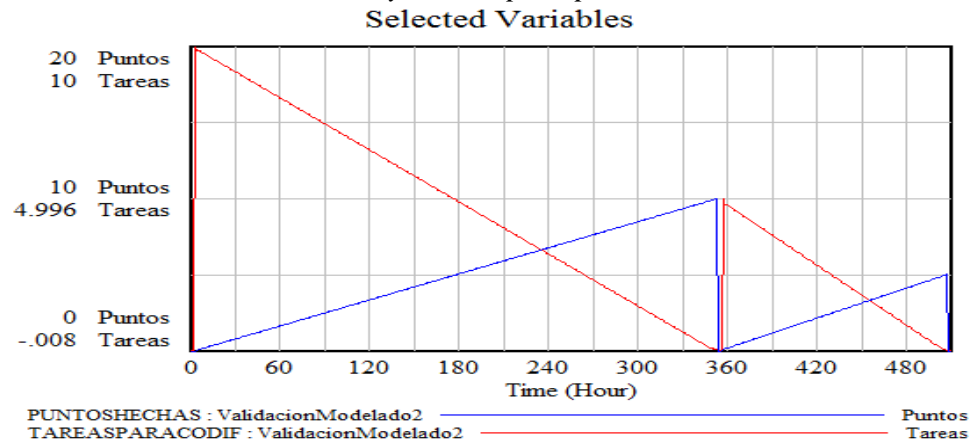


Nota: elaboración propia

A cada punto histórico del modelado actual se le asigna una tarea, y no se presenta ninguna circunstancia que pueda impedir que la codificación de ese punto se desarrolle con normalidad. En su lugar, las tareas y los puntos asignados avanzan gradualmente de acuerdo con la velocidad de desarrollo predeterminedada. Los dos picos recostados para el lado izquierdo son la representación de las tareas para codificar y los dos picos recostados para el lado derecho son la representación de los puntos hechas. Como podemos ver a continuación:

Figura 65

ValidacionModelado2 Puntos y Tareas por Sprints

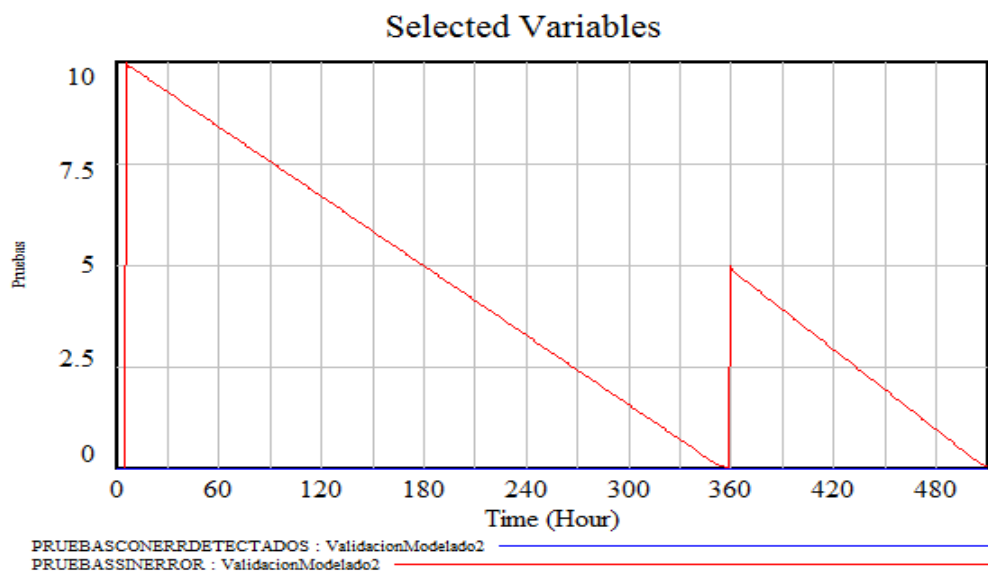


Nota: elaboración propia

Las pruebas de codificación con errores detectados y sin errores es la que se muestra debajo en la figura 67, los dos picos que se muestran son el resultado de las pruebas de tareas desarrolladas sin detección de error alguno, por ello no se denota ninguna irregularidad en ninguno de los dos picos. Ya que en este caso no se tuvo ningún tipo de complicaciones ni retrasos, por ello se muestra en cero en todo el proyecto la variable PRUEBASCONERRDETECTADOS.

Figura 66

ValidacionModelado2 Pruebas con Errores Detectados y Pruebas sin Error



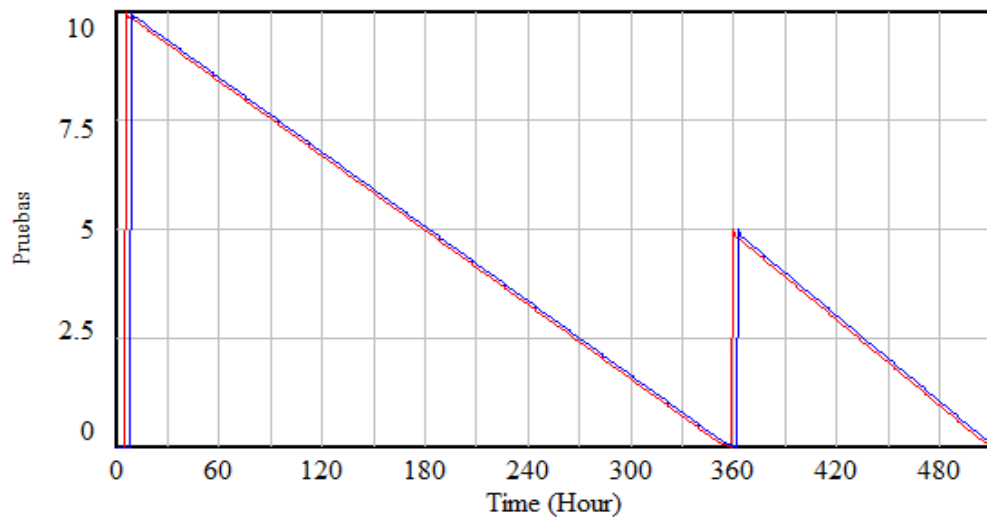
Nota: elaboración propia

Al ya tener el total de tareas por Sprints que pasaron por la prueba sin errores, el siguiente sub modelo a la cual se las deriva es el sub modelo de pruebas de Integración, este resultado se muestra en la figura 68, en esta se muestran las pruebas de integración en forma de dos picos que tienen un punto alto y con una diagonal la caída del mismo en representación del avance de la prueba, estas representaciones no presentan irregularidad

alguna, indicándonos esto que no se tiene detectada errores en la integración. La siguiente es la figura descrita:

Figura 67

ValidacionModelado2 Prueba de Integración
Selected Variables



PRUEBASINTEGRADAS : ValidacionModelado2

PRUEBASINERROR : ValidacionModelado2

Nota: elaboración propia

4.3.2.3. Modelado 3: “APLICATIVO WEB ALPAMAYO

CLIENT.”

Proyecto ejecutado para la empresa PRODUCTOS Y SERVICIOS INDUSTRIALES ALPAMAYO S.A.C. en la cual se aplicó el marco de trabajo Scrum, orientada principalmente para la atención a los clientes que la empresa tiene a través de la plataforma web.

Categorizamos los Sprints y establecimos el número total de Sprints para el proyecto teniendo en cuenta la importancia de cada Sprint según lo indicado por cada relato de usuario. Se utilizaron cuatro sprints para construir la aplicación web, y cada uno de ellos se organizó en función de la duración y la importancia de cada relato de usuario.



El equipo de trabajo tiene asignadas ocho horas diarias a lo largo de la jornada laboral, que es de lunes a viernes, más los sábados durante un periodo de tres meses. Esto nos permite calcular el número de días laborables que se asignan al proyecto para cada Sprint.

Datos Necesarios

Sprint	Tiempo Inicio	Tiempo Fin	Duración Horas	Puntos	Velocidad Promedio	Tareas	Tareas Extras
1	0	208	208	20	0.097	10	0
2	208	400	192	16	0.084	8	0
3	400	592	192	16	0.084	8	0
4	592	720	128	10	0.078	5	0

Equipo de Trabajo: 2 personas, 1 Scrum Master y 1 Team, 2 Nivel Junior.

Horas Diarias: 8h.

Principales Resultados del Modelado 3

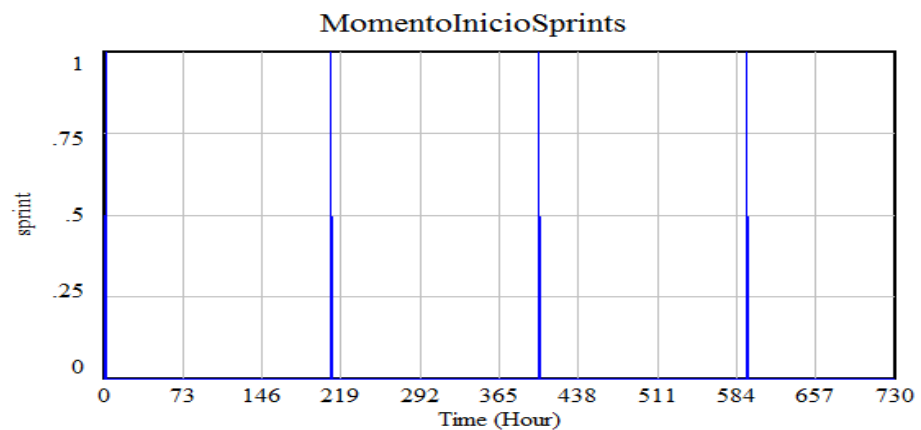
Tenemos un total de 4 Sprints los cuales serán modelados en este apartado con sus respectivos datos necesarios para hacer el modelado. Las figuras siguientes son el resultado de correr el modelo, esto con las diferentes variables plasmadas.

El momento de inicio de cada Sprint esta presentada en la figura 69, como ya se describió anteriormente se tiene un total de 4 sprint desarrolladas en este proyecto de muestra, por ello se nos muestra 4 líneas verticales distribuidas a lo largo del tiempo, estas representan el momento en el tiempo donde se programa el inicio del desarrollo de un determinado

sprint, por ejemplo la primera línea vertical se nos muestra en el momento del tiempo 0, la segunda se ubica en el momento del tiempo 208, las últimas dos se ubican en 400 y 592. La figura es la siguiente:

Figura 68

ValidacionModelado1 Variable Momento Inicio Sprints



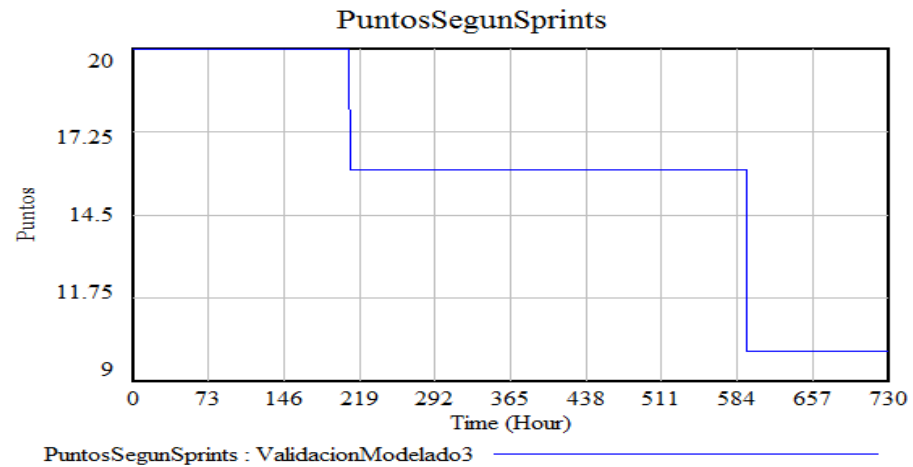
MomentoInicioSprints : ValidacionModelado3

Nota: elaboración propia

Los puntos de historia asignados a cada Sprint son también correspondientes al número de tareas, en la figura 70 se muestra tres escalones estas son los puntos asignados por cada sprint completado, el primer escalón iniciando del lado izquierdo ubicado a un nivel de 20 puntos es perteneciente al sprint 1, el segundo que se ubica a un nivel de 16 puntos pertenece al sprint 2 y sprint 3, el tercer escalón representa 10 puntos para el sprint 4. Este modelado se muestra en la siguiente figura:

Figura 69

ValidacionModelado3 Variable Puntos Según Sprints

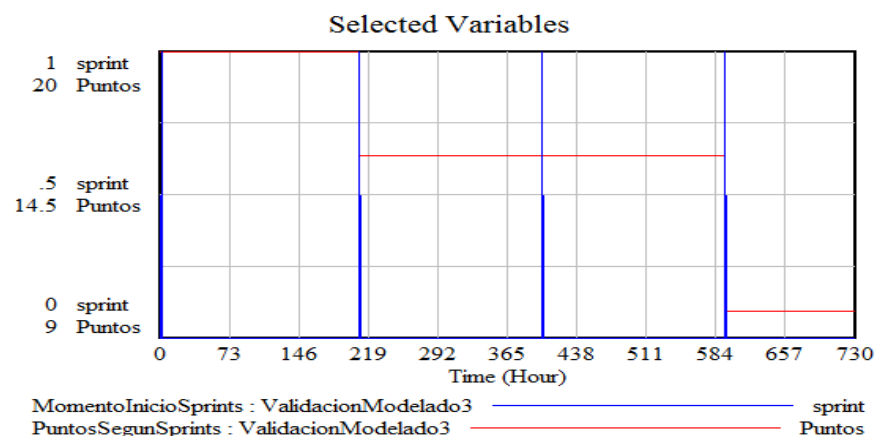


Nota: elaboración propia

En la anterior grafica se muestra tres escalones, esto no implica que solo existan puntos para tres Sprints, resulta que el Sprint 2 y 3 tienen la misma cantidad de puntos asignados y por ello en la figura se muestra como uno solo, para mayor entendimiento en la figura 71 se muestran las separaciones de cada sprint con sus respectivos momentos de inicio. La figura siguiente muestra lo anterior descrito.

Figura 70

ValidacionModelado3 Momentos de Inicio y Puntos por Sprints



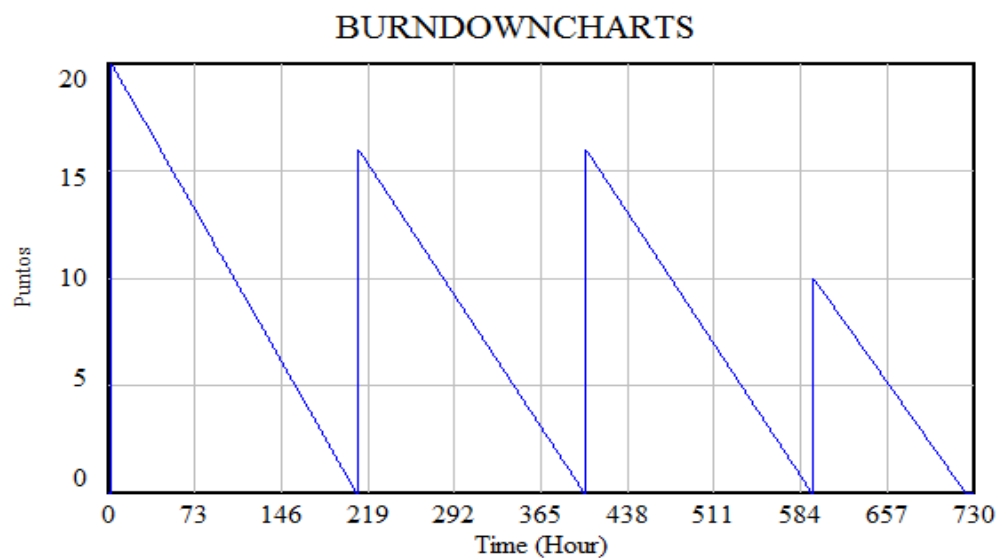
Nota: elaboración propia

La figura de BURNDOWNCHARTS se muestra a continuación, esto por cada Sprint, los 4 picos representan cada sprint, el primero con

una altura máxima de 20 al nivel de puntos, la diagonal que desciende es el progreso de puntos logrados en función al tiempo hasta su culminación que se muestra como 208 el punto más bajo en el tiempo con 0 puntos pendientes, veremos que Sprint 2 y 3 tienen el mismo gráfico, esto se debe a que tienen la misma cantidad de tareas planificadas, misma cantidad de puntos asignados y mismo tiempo de desarrollo, para el último sprint con representación del último pico, tiene una altura de 10 en el nivel de puntos. La figura 71 muestra lo descrito y se muestra a continuación:

Figura 71

ValidacionModelado3 Variable BURNDOWN CHARTS



BURNDOWNCHARTS : ValidacionModelado3

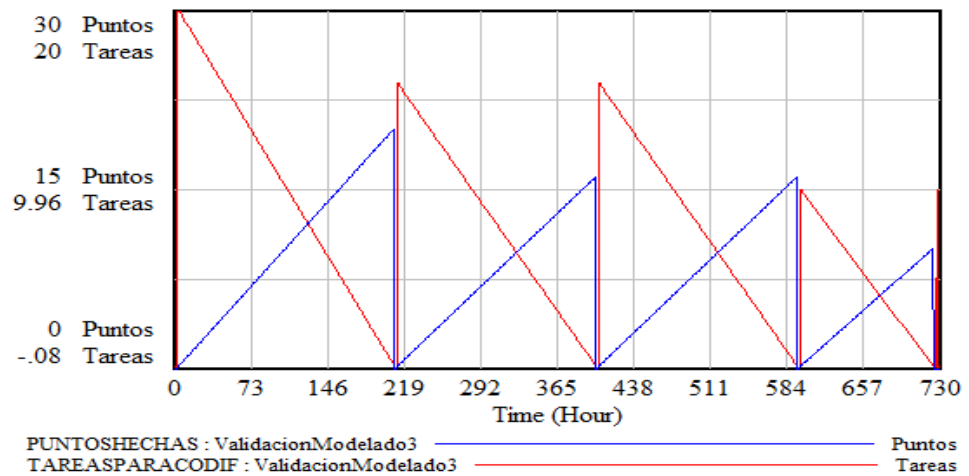
Nota: elaboración propia

Se designó dos puntos por cada tarea completada por ello se muestra en la figura 73 la medida de los puntos como picos inclinados para la izquierda que son el doble en altura de las tareas que están representadas como picos inclinados para la derecha.

Puesto que en la descripción y documentación mencionan que no hubo ninguna clase de retrasos o errores con las tareas y así mismo con el proyecto, no se visualizan irregularidades en ninguno de los picos.

Figura 72

ValidacionModelado3 Puntos y Tareas por Sprints
Selected Variables



Nota: elaboración propia

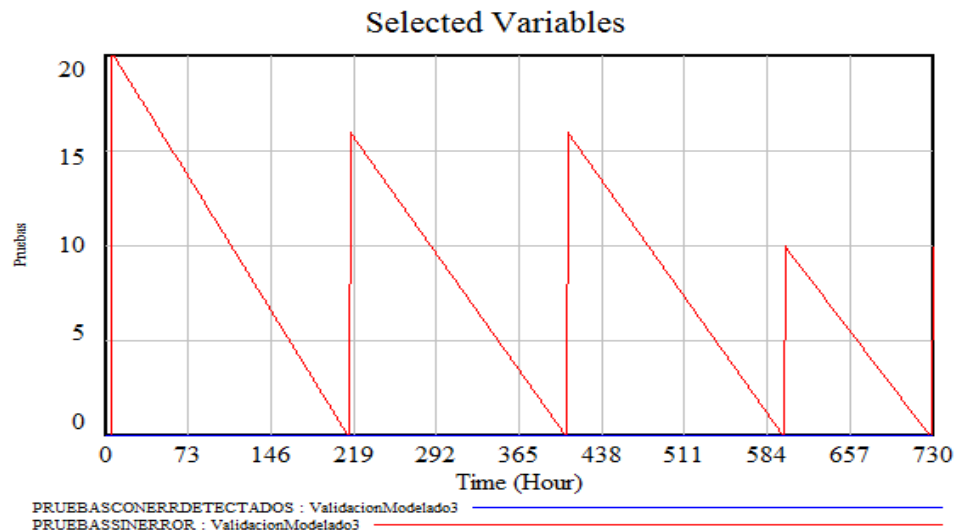
Las pruebas de desarrollo nos plasman la situación del total de pruebas con resultados correctos e incorrectos, pero como se describió anteriormente no se tuvo errores en la codificación esto basándonos en el proyecto descrito.

La figura siguiente nos muestra en su totalidad las tareas como pruebas sin errores, esto se visualiza como los 4 picos sin alguna irregularidad, y por otro lado las pruebas con errores muestran cero o ninguna a lo largo del proyecto, esto se visualiza con una línea horizontal al nivel de 0 puntos a lo largo de todo el tiempo.

Figura 73

ValidacionModelado3 Pruebas con Errores Detectados y Pruebas sin

Error

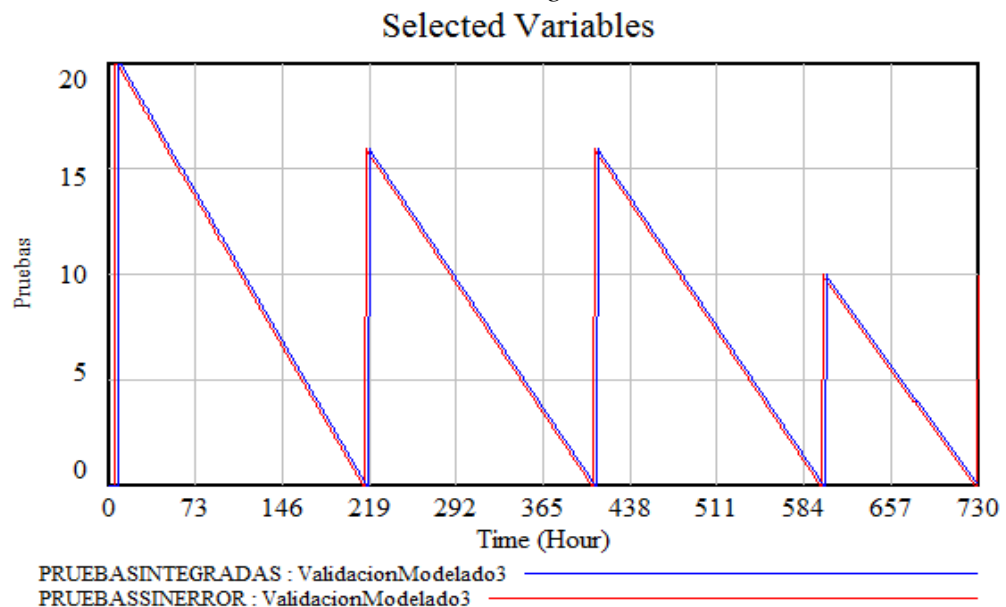


Nota: elaboración propia

La figura 75 nos muestra las pruebas de integración sin errores y como la totalidad de las tareas y Sprints fueron correctas, por ello todas estas pasan a ser evaluados con la prueba de integración, y como el proyecto lo dice todas son correctas.

Figura 74

ValidacionModelado3 Prueba de Integración



Nota: elaboración propia



4.3.2.4. Análisis de Validación Estadística

Para este análisis es que se realizó la simulación de los tres proyectos que se tomaron de muestra, inicializadas con los diferentes datos reales proporcionadas por los responsables encargados de cada proyecto.

El porcentaje de error producido por los datos reales y simulados se examinó mediante la técnica estadística conocida como error relativo. Barlas afirma que un modelo se considera válido si su porcentaje de error es inferior al 5% en sus investigaciones sobre la validación de modelos de simulación. La fiabilidad del modelo se evaluó teniendo en cuenta el porcentaje de error relativo absoluto. La siguiente ecuación establece su expresión matemática:

$$\% \text{ error relativo} = \left| \frac{\text{dato simulado} - \text{dato real}}{\text{dato real}} \right| * 100$$

Los datos seleccionados para realizar el análisis de validación son representativos del sistema y resumen en gran medida el comportamiento de la simulación de cada proyecto estudiado. Para tal efecto se consideraron: Total de Tareas Codificadas, Total de Tareas Integradas y el progreso de desarrollo del proyecto en tres diferentes momentos del tiempo total del cual dispone un determinado proyecto.

En la tabla 3 se muestra el resultado estadístico de los datos analizados, de este análisis se determina que, el porcentaje del valor absoluto de la media del error obtenido de cada uno de los datos analizados es menor al 5%, esto indica que la representación del modelo propuesto es confiable.



Tabla 4

Análisis Estadístico de Datos Reales y Datos Simulados

	% DE AVANCE EN EL TIEMPO			TAREAS CODIFICADAS	TAREAS INTEGRADAS
	1/4	2/4	3/4		
DATOS REALES					
Muestra 1	28	52	78	38	38
Muestra 2	23	50	73	15	15
Muestra 3	25	55	75	62	62
DATOS SIMULADOS					
Muestra 1	28,3552632	55,9868789	74,8683684	38,09	38,24
Muestra 2	22,9432	46,8832667	71,1546667	14,993	14,993
Muestra 3	27,3080968	52,1450806	76,3257258	61,92	61,92
% DE ERROR DATOS REALES/SIMULADOS	3,42968412	1,2641871	1,61559252	0,0026087	0,13304348

Nota: elaboración propia

4.4. PRUEBA DE HIPÓTESIS DE LA INVESTIGACIÓN

4.4.1. Planteamiento de la Hipótesis

Se plantea las siguientes Hipótesis las mismas que serán comprobadas en aplicación y prueba de resultados.

- **Hipótesis Nula**

Ho: El modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum, no mejora la planificación y desarrollo de dichos proyectos en la región de Puno.

- **Hipótesis Alterna**

Hi: El modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum, mejora la planificación y desarrollo de dichos proyectos en la región de Puno.

4.4.2. Nivel de Significancia

El nivel de significancia escogido para la prueba de hipótesis es de 5%, es decir, $\alpha=0.05$ (nivel de significancia).

El Grado de Libertad para el caso será:

$$GL = 7 - 1 = 6$$

Se tiene por lo tanto como Valor Critico el siguiente:

$$t_c = \pm 2,446$$

Región de rechazo, valores t debe rebasar t_c .

Si la probabilidad obtenida $P \leq \alpha$, se rechaza H_0 (se acepta H_1).

Resultados de las Encuestas Pre Test y Post Test

Se realizó las encuestas Pre-test y Post-test dando la siguiente significancia

de las alternativas:

Tabla 5

Significado de las Alternativas

Alternativa	Significado
A	5
B	4
C	3
D	2
E	1

Nota: elaboración propia

El resultado de la encuesta realizada a los tres grupos de trabajo dio los siguientes resultados:

Tabla 6

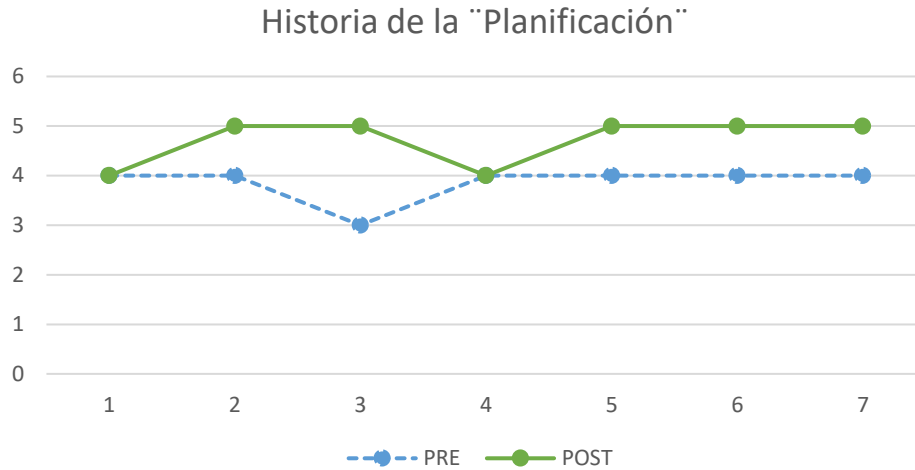
Resultados de la Encuesta Pre-Test y Post-Test

Nº	PRE TEST	POST TEST
1	23	27
2	19	27
3	19	32
4	18	26
5	20	25
6	20	26
7	19	26

Nota: elaboración propia

Figura 75

Resultados Pre y Post de la Planificación

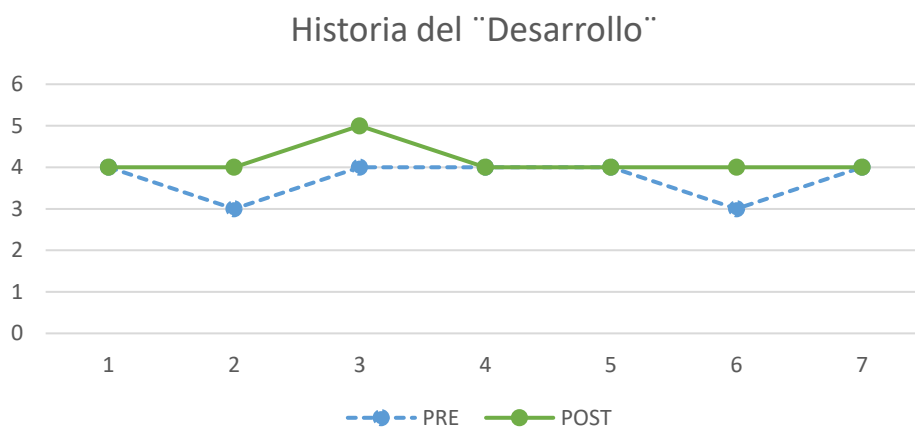


Nota: elaboración propia

La diferencia de la "Planificación" con un antes de(pre) y un después de(post) se visualiza en la Figura 76, si lo representamos en porcentajes para mayor entendimiento se puede describir que en respecto al pre el post se incrementa en un 22,22%, lo cual nos indica una clara mejora en el rubro ya mencionado.

Figura 76

Resultados Pre y Post del Desarrollo



Nota: elaboración propia

En cuanto a la Figura 77, podemos visualizar un antes de(pre) y después de(post) del "Desarrollo", observando diferencias positivas para el proyecto, si lo representamos en porcentajes se puede describir que en respecto al pre el post se incrementa en un 11,54%, indicándonos una buena mejora.

Procesamiento de Datos

Al aplicar el software de análisis estadístico, los resultados se muestran a continuación:

Tabla 7

Resultados de Medias Prueba T-Student.

		Media	N	Desviación típ.	Error típ. de la media
Par 1	PRE_TEST	19,71	7	1,604	0,606
	POST_TEST	27,00	7	2,309	0,873

Nota: elaboración propia

Tabla 8

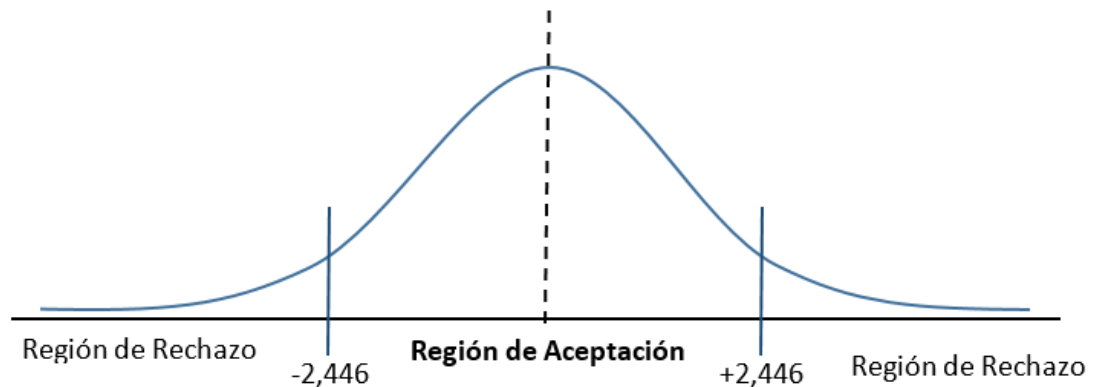
Resultados de la Prueba T-Student

Prueba de muestras relacionadas									
Diferencias relacionadas									
		Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia		T (student)	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	PRE_TEST - POST_TEST	-7,286	2,928	1,107	-9,993	-4,578	-6,584	6	0,0005

Nota: elaboración propia

Figura 77

Región de Aceptación Hipótesis



Nota: elaboración propia

4.4.3. Interpretación de Resultados

Observamos en la Tabla 4 los resultados de las medias correspondientes a datos obtenidos en el Pre-Test y Post-Test, dichos resultados tienen un notorio y significativa diferencia, mostrándonos que la media de aprobación antes de tener la experiencia de hacer uso del Modelo diseñado era de 19.71 y posterior a ella se incrementa la media de aprobación a 27.00, por lo tanto, se puede decir que el Modelo Diseñado, si tiene efectos positivos sobre los proyectos de software que aplican el marco de trabajo Scrum. Como punto importante también se muestran resultados obtenidos del antes de(Pre) y después de(Post) de la planificación y desarrollo, esto en la Figura 78 y Figura 79, obteniendo resultados descritos en porcentajes del Post en base al Pre del proyecto, los mismos que son de un 22,2% y 11,5% de mejora respectivamente.

En la Tabla 5 se nos muestra el resultado de la prueba T Student, el valor que nos resulta para t es de $-6,584$ y al hacer la verificación en la Figura 80, denotamos que t queda en la Región de Rechazo de la Alternativa Nula. Al verificar la tabla mencionada, también podemos apreciar que la Significancia



(Probabilidad P) es igual a 0,0005 la cual es menor que α que es de 0,05 (5%), por lo tanto, se Rechaza la Hipótesis Nula y se acepta la Hipótesis Alternativa.

Se rechaza H_0 .

Es decir, “El modelo de dinámica de sistemas en los proyectos de software que aplican el marco de trabajo Scrum, mejora la planificación y desarrollo de dichos proyectos en la región de Puno”, se acepta.

4.5. DISCUSIÓN

El objetivo de este proyecto fue: Determinar que el modelo de dinámica de sistemas mejora la planificación y desarrollo de proyectos de software que aplican el marco de trabajo Scrum en la región de Puno. Para este fin se inició con la construcción y validación del modelo dinámico, siguiendo las fases de la metodología de dinámica de sistemas, conllevándonos al logro de, tener la confiabilidad en la representación del modelo diseñado, ya que en la fase de validación el porcentaje del valor absoluto de la media del error obtenido de cada uno de los datos analizados nos dio resultados menores al 5%, que es el máximo porcentaje permitido como tasa de error.

La encuesta pre-test y pos-test como método de investigación aplicada a la muestra seleccionada nos da resultados de un logro de mejora en la planificación y desarrollo, de un 22,2% y 11,5% respectivamente, evidenciando el logro del objetivo trazado.

De acuerdo con las conclusiones de (Godoy, Belloni, Sosa, Kotynski, & Benitez, 2014), el comportamiento del modelo puede ser efectivamente replicado por el modelo a través del ajuste, gracias a la validación. Además, afirma que el modelo puede ser utilizado como una herramienta para evaluar los efectos de políticas de gestión alternativas e identificar cuellos de botella en el desarrollo de proyectos gestionados con la metodología Scrum después de la validación y experimentación. Con respecto a estas conclusiones se concuerda en que la validación es una etapa extremadamente importante en este tipo de proyectos ya que, con un cuidadoso y correcto desarrollo de esta etapa, al momento de simular escenarios más complejos se visualiza la magnitud potencial de alcance que tiene el modelo dinámico construido. Así también lo hace notar (Lan, Balasubramaniam, & Abdel-Hamid, 2010), quienes desarrollaron un modelo de dinámica de sistema integrador de desarrollo de software ágil, con la correcta validación de su



estructura y comportamiento, utilizando un estudio de caso de un proyecto real. El modelo integrado describe el comportamiento generado por la interacción de prácticas ágiles y gestión de proyectos, incluido el alcance del proyecto, horario y costo. Es este paper nuevamente se resalta la validación del modelo construido, pero a diferencia de nuestro proyecto no solo se hizo el estudio de 01 proyecto para su validación, nosotros hicimos el estudio de 03 proyectos reales llevados a la simulación. La más grande diferencia que se puede denotar entre el antecedente anterior mente mencionado y el presente proyecto que desarrollamos es la ausencia de la construcción de un sub modelo de costo en nuestro modelo dinámico.

Un punto importante a considerar es que en sus recomendaciones para futuros proyectos (Godoy & Bartó, 2002), afirman que para obtener datos post mortem de proyectos gestionados con métodos ágiles, se gestione bases de datos que contengan datos de proyectos de software que aplicaron scrum reales que se hayan completado.



V. CONCLUSIONES

PRIMERO: Con un estudio de análisis previo sobre el marco de trabajo Scrum, se logró con éxito cumplir con el objetivo de conceptualizar y formular el modelo dinámico, esto siguiendo las fases de la metodología de dinámica de sistemas, en su primera fase que es la fase de conceptualización, se construyeron los diagramas causales logrando un total de 10 diagramas, la segunda fase a desarrollar fue la fase de formulación, dentro de esta se establecieron los diagramas de Forrester y la definición de ecuaciones haciendo uso del programa Vensim PLE, conllevándonos a la obtención del Modelo de Dinámica de Sistemas para proyectos de Desarrollo de Software que aplican el marco de trabajo Scrum.

SEGUNDO: La evaluación del modelo de dinámica de sistemas para proyectos de desarrollo de software que en su elaboración hacen uso del marco de trabajo Scrum, fue un éxito. Se dio paso a la fase de evaluación, en el cual se valida el modelo dinámico construido, para ello se siguió el camino propuesto por Forrester, Senge y Barlas, que nos indica trabajar dentro de dos puntos importantes, la validación estructural y validación estadística, dentro del primer punto se trabajó en la verificación de la estructura, en el cual se hizo la comparación del modelo construido con modelos existentes y la estructura real del sistema, también se hizo la consistencia dimensional apoyándonos en el software Vensim PLE, puesto que dicho software cuenta con las herramientas necesarias para evaluar la consistencia de cada ecuación matemática y parámetros que trabajan dentro del modelo propuesto, por último las pruebas de condiciones extremas o escenarios de



pruebas, en ella se simuló verificando la flexibilidad del modelo trabajando con información proveniente de dos fuentes, la primera datos reales obtenida de los 3 proyectos de muestra las mismas que se usaron como base de inicialización para el modelo y el segundo obtenida de los diferentes antecedentes que se consideró apropiadas para el proyecto, haciendo con todo ello 3 diferentes escenarios, resultando en las simulaciones comportamientos propios a los escenarios que se las sometió. Dentro del segundo punto que fue la validación estadística, se simuló los 3 proyectos con sus datos reales y se hizo la comparación de los resultados simulados con los resultados reales de cada proyecto, aplicando la ecuación propuesta por Barlas para la validación de modelos, se obtuvieron resultados menores del 5% en cada tasa de error, el cual es el máximo posible para la validación de un modelo.

TERCERO: Se afirma que el modelo cumple con el objetivo de ser una herramienta útil para el equipo de desarrolladores que trabajan haciendo uso del marco de trabajo Scrum, demostrando dentro del proyecto que mejora satisfactoriamente la planificación y desarrollo con un 22,22% y 11,5% respectivamente, y en la prueba de hipótesis, aplicando la prueba T Student tenemos que la significancia es igual a 0,0005% el cual se encuentra dentro del rango (<5%) de aceptación de la hipótesis planteada en este proyecto. Adicionalmente se puede decir que dada las características con las que cuenta el modelo desarrollado, aportando escenarios variantes, el modelo simula estas proporcionando datos e información para su análisis y apoyo en reuniones de planificación entre otras, influyendo en los riesgos e



implicancias de las diferentes decisiones que se pudieran tomar en el transcurso del desarrollo del software.



VI. RECOMENDACIONES

PRIMERO: Siendo una de las más conocidas y claramente eficientes en el desarrollo de software, se recomienda aplicar el marco de trabajo Scrum en proyectos que requieran un progreso incremental.

SEGUNDO: Se recomienda a los desarrolladores de software y aplicaciones, hacer uso del presente modelo dinámico trabajado, como una herramienta de gestión y entrenamiento de habilidades y conocimientos orientado al marco de trabajo Scrum.

TERCERO: Se recomienda aplicar el modelado y simulación como herramienta altamente influyente en los diferentes proyectos de desarrollo de software, que apliquen Scrum, Extreme Programming, Crystal Methods, Kanban, entre otros marcos de trabajo y metodologías.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Abad, R. C. (2002). *Introducción a la Simulación y a la Teoría de Colas*. Netbiblo.
- Alarcón, V. F. (2010). *Desarrollo de Sistemas de Información una Metodología Basada en el Modelado*. Univ. Politèc. de Catalunya.
- Alfonso, U. M., & Carla, M. V. (2013). *MODELADO Y SIMULACIÓN DE EVENTOS DISCRETOS*. Editorial UNED.
- Amézquita, J. A., Vergara, J. C., & Maza, F. J. (2008). *Modelamiento de Cadenas Agroindustriales mediante Simulación de Redes*. Juan C Vergara Schmalbach.
- Aracil, J. (1995). *Dinámica de sistemas*. Madrid: ISDEFE.
- Aracil, J., & Gordillo, F. (1997). *Dinámica de sistemas*. Madrid: Alianza Editorial S.A.
- Belda, C. F., & Grande, E. U. (s. f.). *LOS MODELOS DE SIMULACIÓN: UNA HERRAMIENTA MULTIDISCIPLINAR DE INVESTIGACIÓN*. 11.
- Barlas, Y. (1994). *Model Validation in System Dynamics*. Scotland: Clare Monaghan and Eric Wolstenholme.
- Bertalanffy, L. v. (1976). *Teoría general de los sistemas: fundamentos, desarrollo, aplicaciones*. Nueva York: Fondo de Cultura Economica.
- Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems*, 1(1), 1-26. <https://doi.org/10.1145/1877725.1877730>
- Carmona, D. H. (2011). *Teoria General de Sistemas: Un Enfoque Hacia La Ingenieria de Sistemas 2ed*. Lulu.com.
- Ciurariu, B., & Risu, C. (2016). *USAGE OF SCRUM METHODOLOGY FOR IMPLEMENTING SOFTWARE PROJECT*. 9.
- Cocco, L., Mannaro, K., Concas, G., & Marchesi, M. (2011). Simulating Kanban and Scrum vs. Waterfall with System Dynamics. En A. Sillitti, O. Hazzan, E. Bache, & X. Albaladejo (Eds.), *Agile Processes in Software Engineering and Extreme*



Programming (Vol. 77, pp. 117-131). https://doi.org/10.1007/978-3-642-20677-1_9

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *Información Básica de SCRUM*. San Francisco, California.

Dimes, T. (2015). *Conceptos Básicos De Scrum: Desarrollo De Software Agile Y Manejo De Proyectos Agile*. Babelcube Inc.

Forrester, J. W., & Senge, P. M. (1980). Tests for building confidence in system dynamics models. *TIME studios in the management science* 14, 204-228.

García, J. M. (2017a). *Conceptos de Dinámica de Sistemas: Y System Thinking*. Juan Martín García.

García, J. M. (2017b). *Teoría y ejercicios prácticos de Dinámica de Sistemas: Dinámica de Sistemas con VENSIM PLE*. Juan Martín García.

Godoy, D. A., Belloni, E., Sosa, E. O., & Kotynski, H. (2014). *Evaluación de Alternativas de Gestión en Proyectos de Software Desarrollados con Scrum utilizando Dinámica de Sistemas*. 10.

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., Méndez Valencia, S., & Mendoza Torres, C. P. (2014a). *Metodología de la investigación*. Mexico, D.F.: McGrawHill.

Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, P., Méndez Valencia, S., & Mendoza Torres, C. P. (2014b). *Metodología de la investigación*. Mexico, D.F.: McGrawHill.

Izquierdo, L. R., Galán Ordax, J. M., Santos, J. I., & Del Olmo Martínez, R. (2008). Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas. *Empiria. Revista de metodología de ciencias sociales*, 0(16), 85. <https://doi.org/10.5944/empiria.16.2008.1391>

Maldonado, C., & Gómez-Cruz, N. (2010). Modelamiento y simulación de sistemas complejos. *Documentos de Investigación, Escuela de Administración*, 66.



- Parra, J. L., Rodríguez, A., Beltrán, D. M., Onofre, H. G., Bueno, G. A., & Uribe, N. (2005). *Modelo de Simulación*. Villavicencia, Colombia: Corpoica.
- Peralta, A. (2003). *Metodología SCRUM*. Montevideo: Universidad ORT Uruguay.
- Petit, A. G. (2009). *Modelado y simulación. Aplicación a procesos logísticos de fabricación y servicios*. Univ. Politèc. de Catalunya.
- Project Management Institute. (2008). *Guía de los fundamentos para dirección de proyectos: Guía del PMBOK cuarta edición*. Newtown Square: Project Management Institute.
- Serrano, A. A., Sanz, L. G., Rodrigo, I. L., Gordo, E. G., Álvaro, B. G., & Brea, L. R. (2011). *MÉTODOS DE INVESTIGACIÓN DE ENFOQUE EXPERIMENTAL*. 33.
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE*. Mexico: Pearson Educación S.A.
- Talledo Jimenez, M. (2013). *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)-Quinta Edición*. Pennsylvania, EEUU: Newtown Square.
- Urbína, G. B. (s. f.). *EVALUACION DE PROYECTOS*. 409.
- URQUÍA, A., & MARTÍN, C. (2016). *MÉTODOS DE SIMULACIÓN Y MODELADO*. Editorial UNED.
- White, A. S. (2014). An Agile Project System Dynamics Simulation Model: *International Journal of Information Technologies and Systems Approach*, 7(1), 55-79. <https://doi.org/10.4018/ijitsa.2014010104>



ANEXOS

ANEXO 1: Encuesta para la prueba de hipótesis

ENCUESTA PRE-TEST Y POST-TEST

INTRODUCCION:

El presente instrumento forma parte del trabajo de investigación titulada: “MODELO DE DINÁMICA DE SISTEMAS PARA LA MEJORA DE LA PLANIFICACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE QUE APLICAN EL MARCO DE TRABAJO SCRUM”.

La información es de carácter confidencial y reservado; ya que los resultados serán manejados solo para la investigación. Agradezco anticipadamente su valiosa colaboración.

INSTRUCCIONES:

A continuación, se le presenta 7 preguntas, le solicito que frente a ellas exprese su opinión personal considerando que no existe preguntas correctas ni incorrectas, marcando con un aspa (X) una de las alternativas que cree conveniente de acuerdo a su criterio y experiencia.

1. En cuanto al desarrollo del proyecto de software aplicando el marco de trabajo Scrum, ¿qué nivel de complejidad considera usted que tiene?
a) Muy baja b) Baja c) Regular d) Alta e) Muy alta

2. ¿En qué medida afecta los cambios en el Product Backlog (lista de deseos) en cuanto al avance del proyecto?
a) No afecta b) Poco c) Regular d) Bastante
e) Significativamente



3. ¿Qué nivel de confianza considera usted, que tiene la predictibilidad del avance del desarrollo del proyecto?
a) Muy bueno b) Bueno c) Regular d) Malo e) Muy malo
4. Al tener la planificación del modo de desarrollo del proyecto, ¿cuán seguro es que se seguirá lo propuesto?
a) Muy seguro b) Seguro c) Tal vez d) No e) Definitivamente no
5. ¿Económicamente que nivel de impacto tiene una mala decisión en cuanto al desarrollo del proyecto?
a) No afecta b) Poco c) Regular d) Bastante e) Significativamente
6. ¿Considera precisa la estimación del tiempo de liberación de un determinado Spint?
a) Completamente de acuerdo b) De acuerdo c) Indiferente d) En desacuerdo e) Muy en desacuerdo
7. ¿Cómo considera en general el desarrollo del software aplicando el marco de trabajo Scrum?
a) Muy buena b) Buena c) Regular d) Mala e) Muy mala

Autor: Verardo Greta Roque



ANEXO 2: Declaración jurada de autenticidad de tesis



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo VERARDO BRESTA ROQUE,
identificado con DNI 73745283 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado
INGENIERIA DE SISTEMAS

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:
"MODELO DE DINAMICA DE SISTEMAS PARA LA MEJORA DE LA
PLANIFICACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE QUE
APLICAN EL METODO DE TRABAJO SERV EN LA REGION DE PUNO"

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y no existe plagio/copia de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 16 de agosto del 2024


FIRMA (obligatoria)



Huella



ANEXO 3: Autorización para el depósito de tesis en el Repositorio Institucional



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo VERARDO GRETA ROQUE,
identificado con DNI 73745283 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

INGENIERIA DE SISTEMAS,
informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

"MODELO DE DINÁMICA DE SISTEMAS PARA LA MEJORA DE LA
PLANIFICACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE QUE APLICAN
EL MARCO DE TRABAJO SCRUM EN LA REGIÓN DE PUNO"

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 16 de agosto del 2024


FIRMA (obligatoria)



Huella