

# Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual



Universidad  
del Cauca

Monografía de trabajo de grado para optar por el título de  
*Magister en computación*

Jhon Jairo Alvarez Londoño

**Director: PhD. Julio Ariel Hurtado Alegría**

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Sistemas  
Investigación y Desarrollo en Ingeniería de Software — IDIS  
Ingeniería de Procesos de Software  
Popayán, 6 de mayo de 2020

---

## **Dedicatoria**

*A Dios Todopoderoso, quien nunca me abandonó en momentos difíciles (Jeremías 1:8), que me mantuvo integro a pesar de las dificultades en todo el proceso (Salmos 31:5) y quien me ha permitido cerrar este ciclo y comenzar a hacer lo que más he anhelado estudiar en toda mi vida:*

*la música (Salmos 20:5).*

## **Agradecimientos**

Parte de ésta investigación ha sido apoyada por la Vicerrectoría de Investigaciones de la Universidad del Cauca, principalmente la estancia de investigación en la Universidad de Chile en Santiago de Chile y los artefactos producidos, principalmente algunos avances sobre la herramienta Analysis and Visualization for Software Process Assessment ([AVISPA](#)) y la evaluación [4.2](#) que fue realizada en conjunto con ingenieros expertos en lenguajes de metamodelado de procesos en compañía de expertos en la herramienta [AVISPA](#).

## Resumen

Una necesidad siempre imperante dentro de la industria de software consiste en la ejecución de proyectos con mayor productividad y menor consumo de recursos posibles para disminuir sus costos. Alcanzar este objetivo adecuadamente implica la definición y formalización de un proceso óptimo que permita solventar las variaciones de cada proyecto y, por ende, del producto realizado. Es entonces necesario definir un proceso que sea reutilizable, mantenible y en constante evolución en base a las necesidades particulares de la industria evitando factores indeseables durante su ejecución. La especificación de un modelo de proceso con estas características deseables debe contar con una cantidad considerable de recursos y esfuerzos dirigidos a su creación y formalización con resultados positivos a mediano y largo plazo, sin contar las escasas técnicas y herramientas que lo soporten. Por otro lado, los lenguajes de metamodelado de proceso permiten una definición de modelos de proceso con la suficiente flexibilidad para ser definidas pero con reglas, relaciones, estructuras y restricciones necesarias en un intento por mejorar la documentación y, por lo tanto, la ejecución del mismo. Diversas aproximaciones han sido realizadas con el objetivo de evitar errores de ejecución o factores indeseables del proceso. **AVISPA**, es una herramienta de análisis visual de modelos de proceso que permite la generación de planos a partir de elementos de un modelo de proceso y métricas generadas en base a los mismos elementos, resaltando de forma visual posibles problemas en la su documentación conocido como patrones de error. **AVISPA**, interpreta modelos de proceso documentados con el lenguaje de metamodelado Software and Systems Process Engineering Metamodel (**SPEM**) 2.0 por medio de la herramienta de documentación de procesos Eclipse Process Framework Composer (**EPF**). El análisis con **AVISPA** permite obtener información adicional sobre la construcción del modelo de proceso previamente a su prueba piloto o ejecución. Sin embargo, diversos reportes han indicado que **AVISPA** presenta problemas en la representación de sus planos y patrones de error. Como clasificador de tipo binario, **AVISPA** representa los elementos de procesos en 4 categorías esenciales que son: verdaderos positivos, verdaderos negativos (categorías deseadas), falsos positivos y falsos negativos (categorías indeseadas) cuya causa principal de clasificación de elementos son las métricas de la herramienta de clasificación. Este trabajo presenta los avances realizados para la depuración de **AVISPA**, aumentando los elementos de las categorías deseadas y disminuyendo los elementos de las categorías indeseadas en base a métricas halladas por revisión sistemática de la literatura y usando el método de análisis de varianza ANOVA.

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

# Índice general

<b>Índice general</b>	<b>V</b>
<b>Índice de figuras</b>	<b>XI</b>
<b>Índice de tablas</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema	1
1.2. Pregunta de investigación	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Método de exploración y evaluación de trabajos relacionados	4
1.4.1. Preguntas de investigación	4
1.4.2. Fuentes y estrategia de búsqueda	5
1.4.3. Clasificación por sección y criterios de inclusión y exclusión de artículos	7
1.4.4. Proceso de ejecución	8
1.5. Resultados	8
1.5.1. Referencias encontradas clasificadas	8
1.5.2. Filtrado de hallazgos	8
1.5.2.1. Métricas por cada año	10
1.5.2.2. Métricas halladas por artículo y criterios de inclusión y exclusión de métricas	10
1.5.3. Conclusiones	11
1.6. Desarrollo y validación de la propuesta	12
1.6.1. Fase exploratoria	13
1.6.2. Fase de refactorización de AVISPA	13
1.6.2.1. Primera evaluación de AVISPA (Sección 4.2)	15
1.6.2.2. Segunda evaluación de AVISPA (Sección 4.3)	15
1.6.2.3. Tercera evaluación de AVISPA (Sección 4.4)	16
1.6.2.4. Cuarta evaluación de AVISPA (Sección 4.5)	16
1.6.2.5. Quinta evaluación de AVISPA (Sección 4.6)	16
1.7. Estructura de la investigación	17

<b>2. Estado del arte y trabajos relacionados</b>	<b>19</b>
2.1. Marco conceptual	19
2.2. Procesos Software	21
2.2.1. Introducción y Conceptos	21
2.2.2. Lenguajes de Especificación de Modelos de Proceso	23
2.2.2.1. <b>SPEM 2.0</b>	23
2.2.3. Métodos de Validación y Verificación (V & V)	25
2.2.3.1. Métricas	25
2.2.3.2. Simulaciones	25
2.2.3.3. Verificación formal	25
2.2.3.4. Visualización / Recuperación	26
2.2.4. Atributos de Calidad de Producto / Proceso de Software	26
2.2.4.1. Modularidad	26
2.2.4.2. Variabilidad	27
2.2.4.3. Modificabilidad	27
2.3. <b>AVISPA</b>	28
2.3.1. Introducción y Conceptos	28
2.3.1.1. Métricas establecidas antes de la refactorización	28
2.3.2. Filtros de Validación y Verificación Aplicados	29
2.3.3. Metodología de evaluación de Clasificadores	30
2.3.3.1. Precisión, Recuerdo y F-measure	30
2.4. Variabilidad en <b>SPEM 2.0</b> instanciado en <b>AVISPA</b>	30
2.4.1. Relaciones de variabilidad	30
2.4.1.1. Contribución	31
2.4.1.2. Extensión	31
2.4.1.3. Reemplazo	32
2.4.1.4. Extensión-Reemplazo	32
2.4.2. Consideraciones adicionales	34
2.4.2.1. Definición de entradas y salidas	34
2.4.2.2. Relación del elemento base a los elementos específicos de variabilidad	34
2.4.2.3. Planos afectados y tipos de variabilidad soportados en <b>AVISPA</b> por ocultamiento de elementos variables	36
2.5. Métricas de modelos de proceso	36
2.5.1. Trabajos encontrados en la revisión sistemática	36
2.5.1.1. Métricas de Procesos Software (C1)	37
2.5.1.2. Evaluación de Métricas de Modelos de Procesos Software (C2)	39
2.5.2. Otros trabajos relacionados	40
2.5.2.1. Métricas de Procesos Software	40
2.5.2.2. Evaluación de métricas de procesos de software	41
2.5.2.3. Visualización de modelos de proceso	41

<b>3. Análisis y Visualizaciones de Modelos de Proceso Software con AVISPA</b>	<b>43</b>
3.1. Introducción	43
3.2. Planos / Patrones de Error de Modelos de Proceso	44
3.3. Proceso de elaboración de planos de AVISPA	45
3.4. La herramienta AVISPA	46
3.4.1. Arquitectura	46
3.4.2. Modificaciones respecto a la anterior versión de AVISPA	47
3.4.2.1. Modificaciones generales en la visualización y organización de los planos	47
3.4.2.2. Respecto a las conclusiones y datos hallados de M. Bastarrica et al. [41]	48
3.4.2.3. Mejoras específicas en la visualización y organización de los planos	49
3.4.3. Modificaciones propuestas fuera del ámbito del presente trabajo	51
3.4.4. Descripción de las modificaciones	52
3.4.4.1. Patrón de rol sobrecargado	52
3.4.4.2. Resolución de la variabilidad sobre tareas	53
3.4.4.3. Inicio de navegación por medio de los procesos de entrega	54
3.4.4.4. Factor de acoplamiento de actividades por otras actividades contenedoras y por tareas	54
3.4.4.5. Roles responsables de productos de trabajo	57
3.4.4.6. Artefactos de trabajo desperdiciados	59
3.4.5. Soporte de la variabilidad en AVISPA	59
3.4.5.1. Adiciones de planos de artefactos y roles con resolución de variabilidad	60
3.4.5.2. Resolución de variabilidad en todos los planos de roles, tareas y artefactos	60
<b>4. Evaluación de AVISPA</b>	<b>63</b>
4.1. Introducción	63
4.2. Evaluación 1	65
4.2.1. Descripción del estudio empírico	66
4.2.2. Diseño del estudio	66
4.2.2.1. La entrevista semi-estructurada	67
4.2.2.2. El método think-aloud	67
4.2.2.3. Alcance	68
4.2.2.4. Ejecución	68
4.2.3. Resultados	68
4.2.3.1. Respuestas del cuestionario y calificaciones	68
4.2.3.2. Resultados del método think-aloud	70
4.3. Evaluación 2	71
4.4. Evaluación 3	71
4.4.1. Justificación de la evaluación	71
4.4.2. Diseño de la evaluación	71
4.4.3. Ejecución sobre el patrón de roles responsables	72

4.4.3.1. Resultados . . . . .	75
4.5. Evaluación 4 . . . . .	75
4.5.1. Justificación de la evaluación . . . . .	75
4.5.2. Diseño de la evaluación . . . . .	77
4.5.3. Resultados . . . . .	77
4.5.3.1. Casos de uso . . . . .	77
4.5.3.2. Correcciones realizadas en <a href="#">AVISPA</a> . . . . .	81
4.6. Evaluación 5 . . . . .	81
4.6.1. Introducción . . . . .	81
4.6.2. Concepción . . . . .	83
4.6.3. Diseño . . . . .	83
4.6.3.1. Formulación de Hipótesis . . . . .	83
4.6.3.2. Matriz de Confusión . . . . .	84
4.6.3.3. F-score y ANOVA . . . . .	85
4.6.4. Preparación . . . . .	85
4.6.4.1. ANOVA instanciado al diseño del experimento en <a href="#">AVISPA</a> . . . . .	85
4.6.4.2. Gestión de la aleatoriedad (matriz de grupos vs modelos de proceso) . . . . .	87
4.6.4.3. Modelo de Proceso de Ejemplo de Variabilidad con casos básicos (MP-2) . . . . .	88
4.6.5. Ejecución . . . . .	89
4.6.5.1. Resultados . . . . .	90
4.6.6. Análisis . . . . .	90
4.6.7. Diseminación y toma de decisiones . . . . .	94
4.6.7.1. Estrategia de búsqueda de inconsistencias de clasificación . . . . .	94
4.6.7.2. Ámbito de búsqueda de inconsistencias de clasificación . . . . .	95
4.6.7.3. Plantilla de evaluación . . . . .	97
4.7. Lecciones aprendidas . . . . .	98
4.7.1. Evaluación 1 . . . . .	98
4.7.2. Evaluación 2 . . . . .	99
4.7.3. Evaluación 3 . . . . .	99
4.7.4. Evaluación 4 . . . . .	100
4.7.5. Evaluación 5 . . . . .	100
4.7.6. Heurísticas de documentación de modelos de proceso en <a href="#">EPF</a> orientados a <a href="#">AVISPA</a> . . . . .	100
4.7.7. Heurísticas de evaluación / seguimiento de planos y patrones en <a href="#">AVISPA</a> . . . . .	102
<b>5. Conclusiones, limitaciones, trabajo futuro y aportes adicionales</b> . . . . .	<b>105</b>
5.1. Conclusiones . . . . .	106
5.2. Limitaciones . . . . .	109
5.3. Trabajos futuros . . . . .	109
5.4. Aportes adicionales . . . . .	110
<b>Bibliografía</b> . . . . .	<b>113</b>

<b>A. Códigos de planos y funcionalidades de AVISPA</b>	<b>121</b>
A.1. Introducción . . . . .	121
A.2. Códigos . . . . .	121
A.2.1. Plano de rol sobrecargado . . . . .	121
<b>B. Encuestas realizadas en la Universidad de Chile</b>	<b>123</b>
<b>C. Resultados del test ANOVA</b>	<b>137</b>
<b>D. Ejecutables de AVISPA</b>	<b>141</b>
<b>E. Evaluación 2 sobre código de AVISPA</b>	<b>143</b>
E.0.1. Justificación de la evaluación . . . . .	143
E.0.2. Diseño de la evaluación . . . . .	143
E.0.3. Ejecución sobre el patrón de roles responsables . . . . .	145
E.0.3.1. Resultados del algoritmo importador / de resolución de relaciones . . . . .	146
E.0.3.2. Resultados del algoritmo de visualización del patrón de roles responsables . . . . .	149

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

## Índice de figuras

1.1. Proceso seguido para la captura de trabajos relacionados . . . . .	9
1.2. Artículos encontrados por cada año de la ventana de tiempo . . . . .	11
1.3. Métricas por número de apariciones de los artículos encontrados en la revisión sistemática . . . . .	12
1.4. Fase Exploratoria . . . . .	14
1.5. Fase de refactor de <b>AVISPA</b> . . . . .	14
2.1. Marco conceptual, organización de las temáticas de la investigación . . .	20
2.2. Concepto básico para el modelado de procesos en <b>SPEM 2.0</b> [21] . . . .	24
2.3. Diagrama de variabilidad del metamodelo <b>SPEM 2.0</b> . . . . .	30
2.4. Resolución de variabilidad de tipo <i>contribución</i> . . . . .	31
2.5. Resolución de variabilidad de tipo <i>extensión</i> . . . . .	32
2.6. Resolución de variabilidad de tipo <i>reemplazo</i> . . . . .	33
2.7. Resolución de variabilidad de tipo <i>extensión - reemplazo</i> . . . . .	35
2.8. Artículos seleccionados de la revisión sistemática . . . . .	37
3.1. Generación de planos y patrones de error <b>AVISPA</b> . . . . .	46
3.2. Arquitectura tecnológica de implementación de <b>AVISPA</b> . . . . .	47
3.3. Patrón de roles sobrecargados de la nueva versión de <b>AVISPA</b> . . . . .	53
3.4. Patrón de tareas con variabilidad <b>AVISPA</b> . . . . .	54
3.5. Navegación por medio de procesos de entrega <b>AVISPA</b> . . . . .	55
3.6. Patrón de factor de acoplamiento de actividades de la nueva versión de <b>AVISPA</b> . . . . .	57
3.7. Patrón de factor de acoplamiento de tareas de la nueva versión de <b>AVISPA</b>	58
3.8. Patrón de roles responsables de artefactos de la nueva versión de <b>AVISPA</b>	59
3.9. Patrón de roles responsables de artefactos de la nueva versión de <b>AVISPA</b>	60
3.10. Plano de Variabilidad de Artefactos en <b>AVISPA</b> . . . . .	61
3.11. Plano de Variabilidad de Roles en <b>AVISPA</b> . . . . .	61
3.12. Plano de Subproyectos Independientes en <b>AVISPA</b> . . . . .	62
4.1. Ubicación de evaluaciones entre características a evaluar de <b>AVISPA</b> . . .	64
4.2. Flujograma de pruebas unitarias por cada plano de <b>AVISPA</b> . . . . .	73
4.3. Organización del entorno para pruebas en <b>AVISPA</b> . . . . .	74
4.4. Código del método del modelo de proceso de prueba para el patrón de roles responsables . . . . .	75

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

4.5. Código de pruebas de las relaciones generadas del proceso XML para el patrón de roles responsables . . . . .	76
4.6. Código de pruebas del patrón de roles responsables . . . . .	77
4.7. Flujograma de pruebas unitarias por cada plano de AVISPA . . . . .	78
4.8. Diferencias entre versiones de AVISPA del plano de variabilidad de tareas . . . . .	81
4.9. Proceso de evaluación de AVISPA. . . . .	82
4.10. Relaciones de variabilidad, casos básicos. . . . .	89
4.11. Gráficos de cajas. . . . .	93
5.1. Relación de elementos de AVISPA respecto a <i>Precision y Recall</i> . . . . .	108
A.1. Código de plano de rol sobrecargado . . . . .	122
D.1. Pantalla principal de AVISPA 1.0 . . . . .	142
D.2. Pantalla principal de AVISPA 2.0 . . . . .	142
E.1. Flujograma de pruebas de caja blanca por cada plano de AVISPA . . . . .	144
E.2. Código de importación de elementos de proceso para el patrón roles responsables . . . . .	147
E.3. Grafo de control de flujo del algoritmo de importación del patrón de roles responsables. . . . .	148
E.4. Código de visualización del patrón de roles responsables . . . . .	151
E.5. Grafo de control de flujo del algoritmo de visualización del patrón de roles responsables. . . . .	152

# Índice de tablas

1.1. Preguntas de investigación del estudio . . . . .	5
1.2. Cadena de búsqueda . . . . .	6
1.3. Estrategia de búsqueda . . . . .	6
1.4. Preguntas de evaluación de los trabajos . . . . .	7
1.5. Clasificación de referencias pertinentes encontradas en la revisión de la literatura. . . . .	10
2.1. Relaciones de entrada y salida por elemento variable de proceso . . . . .	34
3.1. Correlación de hipótesis con los patrones de error . . . . .	49
3.2. Resultados de correlación de las hipótesis del trabajo de M. Bastarrica et al. [41] . . . . .	50
4.1. Métricas del estudio . . . . .	67
4.2. Clasificación del nivel de comprensión de los patrones de error por parte de los ingenieros de proceso participantes . . . . .	70
4.3. Caso de Uso de resolución de la contribución en AVISPA. . . . .	79
4.4. Caso de Uso de resolución de la extensión en AVISPA. . . . .	79
4.5. Caso de Uso de resolución del remplazo en AVISPA. . . . .	80
4.6. Caso de Uso de resolución de extensión-reemplazo en AVISPA. . . . .	80
4.7. Categorías de clasificación de elementos de clasificadores binarios. . . . .	84
4.8. Procedimientos matemáticos para el análisis de varianza de ANOVA de un factor. . . . .	86
4.9. Plantilla de datos de ANOVA de un factor . . . . .	87
4.10. Grupos vs Modelos de proceso . . . . .	87
4.11. Resultados de $F_1Score$ . . . . .	90
4.12. Resultados de <i>Precision</i> . . . . .	91
4.13. Resultados de <i>Recall</i> . . . . .	91
4.14. Evaluación ANOVA de la herramienta AVISPA según la tabla 4.9 . . . . .	92
4.15. Datos del diagrama de cajas de $F_1Score$ . . . . .	92
4.16. Datos del diagrama de cajas de <i>Precision</i> . . . . .	94
4.17. Datos del diagrama de cajas de <i>Recall</i> . . . . .	94
4.18. Heurísticas de documentación de modelos de proceso en EPF orientados a AVISPA. . . . .	101
4.19. Heurísticas de evaluación / seguimiento de planos y patrones en AVISPA. . . . .	103

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

E.1. Depuración de variables del código de importación para la visualización del patrón de roles responsables. . . . .	149
E.2. Depuración de variables del código de la visualización del patrón de roles responsables. . . . .	152

## Listado de Acrónimos

**AVISPA** Analysis and Visualization for Software Process Assessment

**BPMN** Business Process Model and Notation

**EPF** Eclipse Process Framework Composer

**FP** Falso Positivo

**FN** Falso Negativo

**PSHN** Prueba de Significación de la Hipótesis Nula

**SEI** Software Engineering Institute (Carnegie Mellon)

**SPEM** Software and Systems Process Engineering Metamodel

**UMA** Unified Method Architecture

**UP-VSE** Unified Process for Very Small Entities

**OpenUP** Proceso Unificado libre de la Fundación Eclipse

**VP** Verdadero Positivo

**VN** Verdadero Negativo

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

# Capítulo 1

## Introducción

La calidad de los productos y servicios es un atributo deseable en cualquier industria en general, así mismo como de la industria de software en particular, puesto que este atributo permite el incremento de su competitividad [82], sin embargo, ésta requiere la especialización de las actividades organizacionales y de modelos de procesos aptos para su respectivo contexto [36]. Debido a esto, se han generado alianzas industria – academia para afrontar los retos que conlleva el desarrollo de un producto software para generar productos de software con más calidad, a menor tiempo, así como un mejor aprovechamiento de recursos. Para tratar de alcanzar esta meta, la industria software ha optado por la mejora sus procesos internos para la producción de software a tal punto que optó por la formalización de sus propios modelos procesos [9]. A partir de la necesidad de la industria para la formalización de sus procesos se han generado diversos lenguajes que le permiten la construcción de sus modelos y la promulgación de los mismos. Pero con cada nuevo avance que se obtiene, se generan nuevos retos para la industria, uno de estos retos ha sido la validación de los procesos, por lo cual los investigadores se han enfocado en estrategias de validación y verificación que le ayuden a generar modelos de proceso de mayor calidad, lo que permitirá un mejor desempeño organizacional para la obtención de productos de calidad [10, 11, 18].

### 1.1. Planteamiento del problema

La industria ha considerado que es necesario invertir esfuerzos para la definición formal de modelos de procesos. Sin embargo, alcanzar esta meta conlleva una gran cantidad de esfuerzo y unos costos asociados [27].

La lógica de negocio de la empresa juega un papel crucial en la definición de modelos de proceso, principalmente por la metodología de desarrollo, el contexto de los proyectos y de los negocios. Además de los problemas para comunicar y ejecutar un proceso, existe una alta complejidad de su diseño ya que debe cubrir muchas aristas: organización, estándares, cultura, dominios, proyectos, talento humano, entre otros [12, 15, 82].

Los metamodelos son especificaciones de lenguajes para la definición y manipulación de modelos dentro del proceso de software, los cuales están enfocados a conseguir cierto grado de estandarización y automatización en la industria de software [32]. En el dominio de procesos existen lenguajes de propósito específico tales como SPEM en su versión 2.0 [24], Business Process Model and Notation (BPMN) [22], BPMN 2.0 [23], XPD L [31], LittleJil [7], vSPEM [89], PROMENADE [25], entre otros. Algunos de estos lenguajes son de propósito específico para el modelado de procesos de software, como es el caso de SPEM 2.0, PROMENADE y vSPEM [27]. Adicionalmente, las organizaciones tienden a evolucionar y con ella también evolucionan sus procesos, principalmente en el crecimiento y profundización de su definición y de sus elementos.

La gran cantidad de factores involucrados en la definición de los modelos de proceso de software brinda una idea de la dimensión del esfuerzo involucrado por parte de las organizaciones para la definición de procesos de software [9]. Así, la construcción de modelos de proceso más grandes desde la práctica, incrementa las posibilidades de generar más defectos, por ejemplo los cuellos de botella, en contraste con modelos más pequeños [38]. Lo anterior evidencia la necesidad de generar indicadores que soporten la toma de decisiones, principalmente sobre la definición, modificación, extensión y/o interacción de los elementos de proceso como parte de la validación y verificación durante su construcción.

Teniendo en cuenta la premisa de que los procesos de software también son software [7], se pueden establecer indicadores a los modelos de proceso software de manera similar a cómo son definidos para los productos de software. Por ejemplo Cánfora et. al. [6] analizan la usabilidad y mantenibilidad de los modelos de procesos de software. Por lo tanto, es deseable contar con indicadores que permitan evaluar los modelos de proceso, antes de que éstos se prueben en la ejecución. Esto evitaría sobrecostos, retrasos y mal uso de recursos [6]. Así, se requiere de nuevos esfuerzos y mayor colaboración entre la industria y la academia con el fin de generar estrategias que le permitan a las organizaciones elaborar modelos de proceso con mayor calidad que les permita alcanzar una real efectividad de sus procesos [5, 27].

Existen actualmente diversas estrategias de validación y verificación de modelos de proceso de software entre los cuales encontramos: el uso de métricas [6], la aplicación en pruebas piloto [30, 35], la simulación [14], el chequeo formal [13] y el análisis visual [4, 5].

La visualización de modelos de proceso ha mostrado ser una estrategia de verificación y validación viable y efectiva para el análisis de modelos de procesos, como se presenta en el caso de los Blueprints [2] y los patrones de errores sugeridos por la herramienta para visualización de modelos de proceso AVISPA [1]. Los Blueprints son representaciones gráficas de modelos de proceso de software que le proporciona a los diseñadores de procesos, pistas sobre las inconsistencias en un modelo de proceso de software. Los patrones de errores de AVISPA permiten identificar de manera más concreta problemas específicos en el modelo de proceso, resaltándolos y diferenciándolos de los demás.

Las visualizaciones que AVISPA presenta siguen se basan en el concepto de Vis-

*ta Polimétrica* [40]. Una vista polimétrica es un grafo donde los nodos y enlaces, que representan elementos y conexiones entre elementos de un dominio, son dimensionados y decorados de acuerdo a métricas establecidas y calculadas para estos elementos y conexiones. Así que el poder y confiabilidad de las visualizaciones de AVISPA depende en gran medida de la especificación de las métricas subyacentes.

La definición de las métricas en AVISPA se encuentra en un estado primitivo, en gran parte por la misma capacidad del lenguaje de modelado para representar todos los aspectos deseables de medir y porque AVISPA no recupera toda la información presente en un modelo SPEM 2.0. Por lo que hace falta un soporte técnico y científico en la definición de sus métricas, lo que lleva a que en el caso de AVISPA se tenga aún, un número no pequeño de falsos positivos en sus patrones de proceso [39].

La determinación de errores de un modelo de proceso de software es crucial para lograr su adopción en las organizaciones. En efecto, los beneficios de un modelo de proceso no pueden ser completamente logrados si los participantes del mismo no lo adoptan. Existen diversos motivos por los que la calidad de un producto puede verse afectada desde la formalización de los modelos de proceso y, por lo tanto, se soportan en indicadores o métricas que ilustran su diseño [41]. Algunos beneficios que se pueden obtener tras la formalización de procesos de software consisten en la resolución de la ambigüedad y la posibilidad de implementar herramientas de soporte para la evolución y el análisis automatizado de los modelos de proceso.

Las métricas para modelos de proceso de software conllevan un gran esfuerzo en su definición y, posteriormente, su validación de efectividad y pertinencia para un modelo de proceso software. Canfora et. al. realizan un modelo de filtro para definición de métricas sobre diferentes atributos de calidad de producto en el marco de la mantenibilidad de software. Lo anterior ilustra una dificultad para establecer una trazabilidad sobre que se mide, cómo se mide, a que afecta, de cuales factores depende su efectividad y si los beneficios se manifiestan a corto, mediano o largo plazo [6].

La definición adecuada de las métricas permite la determinación eficaz y veraz de problemas en la definición de los procesos desde épocas tempranas o anteriores a su implementación. Sin embargo, la definición adecuada de las métricas conlleva un gran esfuerzo y depende de diversos factores desde su definición hasta su experimentación y liberación. La herramienta AVISPA permite una construcción visual de diversos elementos de procesos construidos bajo el lenguaje de especificación SPEM 2.0. Pese a que AVISPA genera una visualización entre elementos de proceso, carece de una base más sólida en la definición de sus métricas.

## 1.2. Pregunta de investigación

Por lo anterior este trabajo construye una base para resolver 2 componentes principales:

- *Filtrado y construcción de métricas de modelos de proceso: ¿Cómo definir e implementar un conjunto de métricas de modelos de proceso en el contexto de*

### AVISPA?

- *Modelo de validación y verificación de la propuesta: ¿Como construir y ejecutar un modelo de medición para la evaluación del contraste del impacto de las métricas implementadas entre la propuesta y la anterior implementación de AVISPA?*

## 1.3. Objetivos

### 1.3.1. Objetivo general

Fundamentar un conjunto de métricas para soportar con mayor confiabilidad AVISPA para el análisis visual de los modelos de proceso de software en SPEM 2.0.

### 1.3.2. Objetivos específicos

- Catalogar, describir y analizar las métricas de modelos de procesos y su potencial para el soporte de la visualización en AVISPA a partir de una revisión de la literatura.
- Especificar un conjunto de métricas bien fundamentadas de modelos de proceso de software en SPEM 2.0 que sustenten el análisis visual provisto AVISPA en forma confiable.
- Implementar las métricas y actualizar los blueprints y patrones de error previamente estructurados en AVISPA.
- Validar la propuesta en un contexto real a través de un estudio de caso donde se analice un modelo de proceso con AVISPA.

## 1.4. Método de exploración y evaluación de trabajos relacionados

Para la realización del presente trabajo, se ha optado por la realización de una revisión sistemática dado el rigor necesario para soportar la elaboración de las métricas, así como el refinamiento adecuado de los factores que deben tomarse en cuenta al momento de considerarlas aptas para su incorporación en el análisis visual con AVISPA.

Para la revisión sistemática, se siguió la organización del trabajo de Pedreira et al. [44]:

### 1.4.1. Preguntas de investigación

Dentro de las preguntas de investigación se considera la existencia de trabajos relacionados sobre la validación y verificación de métricas de modelos de proceso

Id	Descripción
PI1	¿Qué métricas han sido propuestas para el análisis / evaluación de modelos de proceso de software?
PI2	¿Cuál es el nivel de validez de cada una de las métricas encontradas que sean potencialmente representadas en SPEM 2.0 y AVISPA?
PI3	¿Que métodos de validación y verificación de métricas de modelos de proceso de software existen?
PI4	¿Cual es el nivel de pertinencia de los modelos encontrados tanto en SPEM 2.0 y/o como en AVISPA?

**Tabla 1.1:** Preguntas de investigación del estudio

SPEM 2.0 que sean representables sobre AVISPA, la creación o estructuración de métricas orientados a la representación sobre vistas polimétricas y/o características que ayuden a lograrla y trabajos existentes en las mismas áreas. La tabla 1.1 ilustra el conjunto de preguntas de investigación asociadas al estudio.

#### 1.4.2. Fuentes y estrategia de búsqueda

Se identificaron términos claves principales para la búsqueda de artículos relacionados: “Software Process Metamodels and Tools” y “Software Process Metrics Validation”. La primera palabra principal contiene las herramientas y el lenguaje de especificación de modelos de proceso considerados para la revisión del estado del arte. La segunda palabra principal cubre la validación de las herramientas consideradas en el primer termino de este trabajo con el fin de hallar las métricas adecuadas con el fin de extender y/o corregir las métricas existentes de AVISPA. La tabla 1.2 ilustra la cadena de búsqueda para la revisión sistemática como aproximación inicial. Las palabras clave pueden cambiar en relación al estudio de Brereton et al. [45].

Los términos claves principales se relacionan con el operador booleano AND, mientras que los términos alternativos (sinónimos, subcategorías, etc.) se ligan mediante el operador lógico OR. Las palabras con \* al final indican todas las combinaciones de dicha palabra a buscar, por ejemplo: “measur\*”, tendrá como resultado “measurable”, “measurement”, etc. Este tipo de combinatoria abarca más posibilidades de trabajos realizados que las palabras puntuales de búsqueda.

Las fuentes de búsqueda de trabajos relacionados se consultaron principalmente en bases de datos académicas. La inclusión de recursos no académicos requirió de criterios detallados para determinar si una fuente de información fue catalogada como confiable o no. Sumado a ello, el problema de la cantidad de resultados obtenidos de un buscador de propósito general por lo que se limitó la exploración a las primeras 2 páginas de resultados de la búsqueda. No se tomarán en cuenta artículos de los cuales no se obtengan una versión completa del trabajo encontrado. La tabla 1.3 ilustra la estrategia de búsqueda del presente trabajo.

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

<b>Términos principales</b>	<b>Términos alternativos</b>
Software Process Metamodels and Tools	(SPEM OR AVISPA OR EPF OR (Eclipse Process Framework) (Polimetric View*) OR (Software Process Model*)) AND
Software Process Metrics Validation	(Metri* OR Measur* OR (Metrics Scope) OR (Metrics Validation) OR (Metrics Set) OR Confiab* OR Valid* )

**Tabla 1.2:** Cadena de búsqueda

<b>Tipo de búsqueda</b>	<b>Base de datos de trabajos</b>
Bases de datos Académicas	- Engineering Village - Science Direct - IEEEExplore - Librería Digital ACM - Base de Datos Springer
Fuentes no académicas	- Google - Academia.edu - ResearchGate - Página web del SEI
Tipos de trabajo	- Journals - Workshops - Artículos en Conferencias - Contribuyentes en conferencias de industria y/ó profesional - Publicaciones en línea no académicas
Búsqueda aplicada a	- Título - Resumen - Palabras Clave
Idiomas	- Inglés y español
Ventana de tiempo	- Artículos desde 2014 a 2018

**Tabla 1.3:** Estrategia de búsqueda

Id	Descripción
<b>Exploración de métricas de procesos software (C1)</b>	
PE1	¿El estudio presenta mención de alguna métrica de modelo de proceso de software que pueda ser potencialmente representada a partir de <b>SPEM 2.0</b> ?
<b>Exploración de evaluaciones sobre métricas de procesos software (C2)</b>	
PE2	¿El estudio realiza una evaluación adecuada de métricas de proceso software que pueda ser instanciada al contexto de <b>SPEM 2.0</b> ?

**Tabla 1.4:** Preguntas de evaluación de los trabajos

### 1.4.3. Clasificación por sección y criterios de inclusión y exclusión de artículos

Para la evaluación de cada uno de los trabajos se han diseñado unas preguntas de evaluación como filtro de aceptación. Cada pregunta de evaluación se compone de una escala de calificación compuesta de un rango numérico desde cero (0.0: No estoy de acuerdo) a cinco (5.0: Totalmente de acuerdo). Estos rangos ilustran el nivel de pertinencia de un trabajo para el revisor de la revisión sistemática.

El nivel de calificación otorgado para cada artículo ilustra el grado de pertenencia de los mismos al trabajo. Dicha calificación consiste en el promedio de cada una de las calificaciones otorgadas en cada pregunta de investigación.

Los artículos escogidos tienen 3 calificaciones dependiendo de unas reglas específicas:

- **Métricas de procesos software (C1):** Un trabajo pertenece a la sección de métricas de procesos software (**C1**) si obtiene una calificación igual o superior a 3.0 sobre la pregunta PE1 ilustrada en la tabla 1.4 (Si un trabajo es clasificado como primario, no es necesario calcular el resto de promedios de los trabajos secundarios).
- **Evaluación de métricas de procesos software (C2):** Un trabajo pertenece a la sección de evaluación de métricas de procesos software (**C2**) si obtiene una calificación promediada igual o superior a 3.0 sobre la pregunta PE2 definida en la tabla 1.4.
- **No pertinente (NP):** Un trabajo no es pertinente si obtiene una calificación promediada inferior a 3.0 sobre cada una de las preguntas de evaluación de estudios secundarios definidas en la tabla 1.4.

Cada artículo filtrado obedece a los siguientes criterios:

- **Inclusión:**

- Estudios sobre medición y comprensión de elaboraciones de modelos de procesos en [SPEM](#) u otros lenguajes de especificación de modelos de proceso aplicables a [SPEM](#) tanto en métricas como en problemas de visualización / organización de elementos de proceso.
- Estudios sobre revisiones sistemáticas realizadas para [SPEM](#) u otros lenguajes de especificación de modelos de proceso aplicables a [SPEM](#) tanto en métricas como en problemas de visualización / organización de elementos de proceso.
- Estudios sobre la comprensión de modelos de proceso en base a los elementos de taxonomía de los lenguajes de especificación y su distribución visual.

■ **Exclusión:**

- Estudios que involucren métricas solamente del contexto de la ejecución de modelos de proceso, por ejemplo: producción de líneas de código o tiempo de ejecución de un proceso entre otros.
- Estudios sobre [AVISPA](#), ya son parte tema sobre el cual trata la presente investigación, y además, es necesario basar la investigación en otros trabajos que permitan establecer criterios de inclusión y exclusión de métricas actualmente implementadas en [AVISPA](#) que puedan ser soportadas bajo el metamodelo [SPEM 2.0](#).

#### **1.4.4. Proceso de ejecución**

La revisión sistemática cuenta con un flujo de ejecución donde se ilustra la gestión de los estudios hallados basándose en la estrategia de Pino et. al. [43]. La figura 1.1 ilustra el flujo de evaluación de los trabajos<sup>1</sup>.

### **1.5. Resultados**

#### **1.5.1. Referencias encontradas clasificadas**

La tabla 1.5 ilustra la clasificación de las referencias encontradas tras el diseño del método de investigación ilustrado anteriormente. Solamente se incluyen los trabajos que han sido seleccionados en este trabajo.

#### **1.5.2. Filtrado de hallazgos**

Los resultados obtenidos de la literatura ilustran un esfuerzo por el refinamiento de métricas de [AVISPA](#).

---

<sup>1</sup>Se ha definido un máximo número de 10 artículos por fuente para recorrer todas las fuentes encontradas

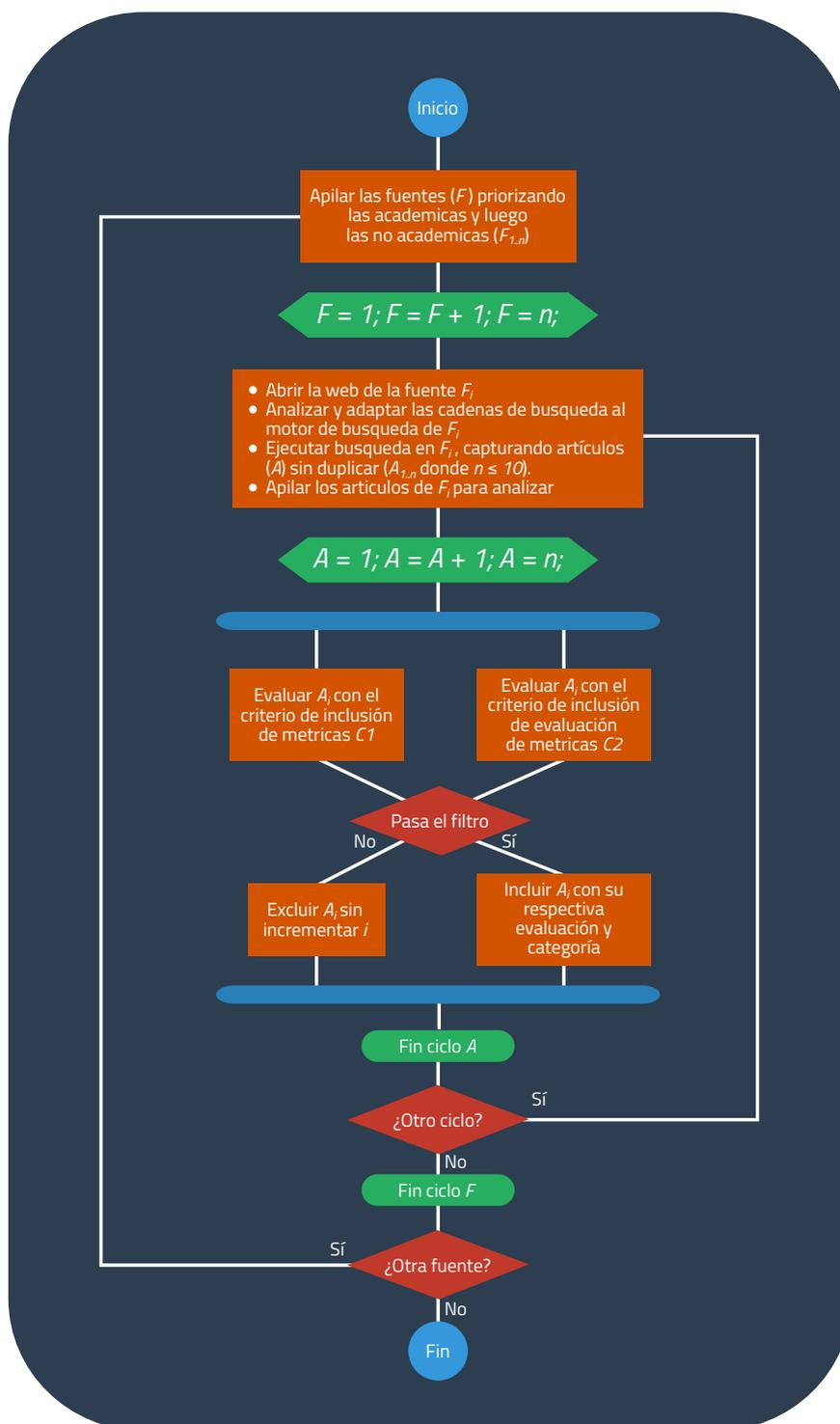


Figura 1.1: Proceso seguido para la captura de trabajos relacionados

Trabajo	Calificación		Categoría
	PE1	PE2	
Primera iteración de artículos			
K. Kluza et al. [50]	5.0	5.0	C1
A. Burattin [72]	3.0	5.0	C1 & C2
K. Kluza [73]	3.0	5.0	C1 & C2
K. Figl et al. [75]	0.0	5.0	C2
J. Pavlicek et al. [46]	3.0	5.0	C1 & C2
J. Pflug et al. [74]	0.0	5.0	C2
del Rio Ortega et al. [51]	5.0	0.0	C1
J. Geisel et al. [52]	5.0	0.0	C1
M. Smatti et al. [53]	2.0	4.0	C2
G. Kaur et al. [54]	5.0	0.0	C1

**Tabla 1.5:** Clasificación de referencias pertinentes encontradas en la revisión de la literatura.

#### 1.5.2.1. Métricas por cada año

Los artículos encontrados fueron categorizados por cada año en el rango de ventana de tiempo (2014 a 2018), por lo cual se ilustra en la figura 1.2.

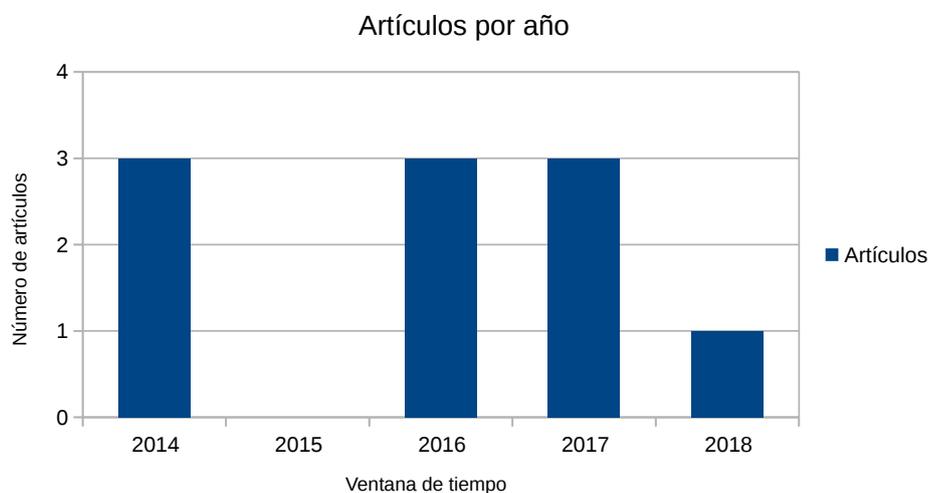
#### 1.5.2.2. Métricas halladas por artículo y criterios de inclusión y exclusión de métricas

El número de coincidencias o apariciones de una métrica por artículo de la revisión sistemática ha sido generado como punto de guía para la refactorización de la herramienta AVISPA, ilustrando el impacto / relevancia de cada métrica encontrada, tal como se encuentra ilustrado en la figura 1.3.

Así mismo, han sido determinados factores adicionales que determinaron la inclusión y exclusión de métricas de la herramienta de visualización de modelos de proceso AVISPA:

- **Interdependencia de métricas:** Es necesario conservar las métricas que generan interdependencia entre sí. Por ejemplo: para calcular el acoplamiento, es necesario calcular algunas métricas que han quedado por fuera de la revisión, tal como el número total de actividades, el número de entradas y salidas, entre otros. En este caso, serán incluidas / conservadas para la modificación de AVISPA.
- **Métricas nuevas:** Se incorporan nuevas métricas, resultantes de la revisión de la literatura, con el objetivo de proveer soporte de acuerdo a los resultados de la misma.

Finalmente, las modificaciones realizadas a cada uno de los blueprints de AVISPA,



**Figura 1.2:** Artículos encontrados por cada año de la ventana de tiempo

con base a los criterios enunciados a lo largo del presente capítulo, se ilustran en la tabla 3.2.

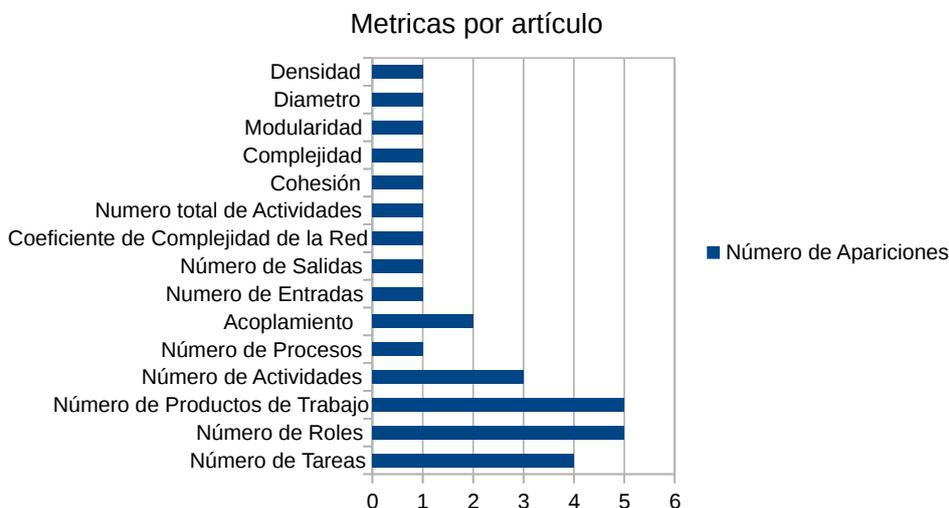
### 1.5.3. Conclusiones

La revisión sistemática conllevó un enorme esfuerzo de búsqueda para conservar el formalismo del proceso de hallazgo de los artículos de éste ciclo de búsqueda. Algunos factores que dificultaron el hallazgo son el foco específico de la temática del presente trabajo y la poca información obtenida de AVISPA y SPEM 2.0 en el rango de la ventana de tiempo.

Dado el enorme esfuerzo invertido en el hallazgo de métricas para SPEM 2.0 es necesario realizar un mayor esfuerzo de búsqueda para lograr una revisión mas profunda, factor por el cual se ha decidido realizar un solo ciclo de búsqueda.

Respecto a las métricas halladas en cada artículo, AVISPA conservará las métricas que tengan 2 o más coincidencias en cada artículo de la revisión. Debido a lo anterior, las siguientes métricas serán implementadas / conservadas en AVISPA:

- Acoplamiento
- Número de Actividades
- Número de Productos de Trabajo
- Número de Roles
- Número de Tareas



**Figura 1.3:** Métricas por número de apariciones de los artículos encontrados en la revisión sistemática

El capítulo 2 ilustra a mayor profundidad el contenido de los artículos de la revisión sistemática de la propuesta.

## 1.6. Desarrollo y validación de la propuesta

La propuesta de tesis consiste en la validación de la construcción de la herramienta en 2 grandes fases:

- **Fase exploratoria:** La fase exploratoria consiste en la validación inicial de expertos para evaluar el nivel de entendimiento de los planos creados. El modelo de implementación de el cual se ha llevado a cabo en 3 actividades:
  - **Modificación / refactorización de AVISPA:** En esta actividad se modifica AVISPA desde la concepción teórica de la métricas halladas y su incorporación a la herramienta de visualización de procesos, también se modifica respecto a los resultados hallados tras un ciclo de resultados de la misma prueba con diferentes expertos en procesos.
  - **Evaluación de AVISPA:** En esta actividad se evalúa AVISPA con los diversos ingenieros de proceso por medio del método Think Aloud<sup>1</sup>. Se realiza una prueba con diversos expertos y diversos modelos de proceso densos relevantes para la prueba como OpenUP y Tutelkan-TRP. También se capturan las ideas de la experiencia de usuario de los blueprints por medio de encuestas.

<sup>1</sup>En el método Think Aloud, mientras el participante está realizando una tarea como parte de un escenario de usuario, expresa en voz alta sus pensamientos, sensaciones y opiniones mientras interactúa con un producto.

- **Resultados:** En esta actividad se discuten los resultados de la prueba de [AVISPA](#), se contrastan conforme a los modelos de proceso en relación con lo presentado por la herramienta de visualización, se actualizan los cuestionarios para próximas evaluaciones y se preparan los resultados para la modificación de [AVISPA](#) en el siguiente ciclo de evaluación.
- **Fase de refactorización de [AVISPA](#):** Esta fase consiste en la evaluación de la herramienta [AVISPA](#) comparada con otro clasificador, en este caso con un experto de la herramienta y en las métricas mostrando el caso ideal del clasificador [AVISPA](#) y el estado actual de la misma. Esta fase se compone de 4 actividades:
  - **Modificación / refactorización de [AVISPA](#):** En esta actividad se modifica [AVISPA](#) desde la concepción teórica de la métricas halladas e incorporadas a la herramienta de visualización de procesos, también se modifica respecto a los resultados hallados tras un ciclo de resultados de la prueba basado en los resultados del ciclo de la evaluación con F-Score.
  - **Clasificador de [AVISPA](#):** En esta actividad se capturan los resultados de la clasificación de los elementos de proceso por [AVISPA](#) así como sus métricas.
  - **Clasificador Humano:** En esta actividad se realiza el mismo ejercicio de blueprints de [AVISPA](#) pero por medio de un experto en procesos capturando métricas y clasificando sus respectivos elementos de proceso.
  - **Comparación y resultados:** En esta actividad se realiza una comparación entre la clasificación de [AVISPA](#) y el experto en procesos.
  - **Resultados:** En esta actividad se discuten los resultados de la prueba, se contrastan conforme a los modelos de proceso en relación con lo presentado por la herramienta de visualización y se preparan y procesan los resultados para la modificación de [AVISPA](#).

### 1.6.1. Fase exploratoria

La fase exploratoria tiene como objetivo ver la percepción de la herramienta [AVISPA](#) por parte de expertos de modelos de proceso pero que no han tenido un acercamiento previo con la herramienta. Con esta fase, los resultados concluyeron también posibles falencias en la construcción de las métricas de [AVISPA](#). La figura 1.4 ilustra el flujo seguido para ésta fase.

### 1.6.2. Fase de refactorización de [AVISPA](#)

La fase de refactor tiene como objetivo hallar las diferencias de clasificadores entre la antigua y la nueva versión de [AVISPA](#) así como entre la nueva versión y la versión de la clasificación de los elementos de proceso por parte de expertos. La figura 1.5 ilustra el flujo seguido para ésta fase.

A continuación, se describen los trabajos relacionados para realizar cada una de las evaluaciones definidas en ésta fase de la investigación.



Figura 1.4: Fase Exploratoria



Figura 1.5: Fase de refactor de AVISPA

#### 1.6.2.1. Primera evaluación de AVISPA (Sección 4.2)

Al ser una evaluación exploratoria, los métodos de captura de la información han sido diseñados para efectos de captura rápida de la información, haciendo uso de dos métodos: *Think-aloud* y la *entrevista semiestructurada*:

El trabajo de A. Donker [83] ilustra cómo el método think-aloud puede ser utilizado para evaluar la usabilidad de sistemas interactivos utilizados por niños. El documento informa sobre una evaluación comparativa de 3 métodos de prueba de usabilidad (UTM) con 45 niños de 8 a 14 años como usuarios de prueba. Los 3 métodos fueron concurrentes de pensamiento en voz alta, entrevista y cuestionario. Estos 3 UTM requieren diferentes niveles de verbalización de los niños que estaban realizando la evaluación. Se planteó la hipótesis de que la edad de los niños, su género, su competencia verbal y el nivel de extroversión influirían en qué método funciona mejor. Los resultados de este estudio muestran que el protocolo de pensar en voz alta ayuda a identificar la mayoría de los problemas de usabilidad y sugiere que las niñas que piensan en voz alta informan más problemas de usabilidad que los chicos.

El trabajo de P Finkbeiner [84] ilustra elementos para llevar a cabo la captura de información por medio de entrevista semiestructurada, donde cada una de las preguntas tiene un propósito específico a evaluar dentro de la investigación. Así, se pueden obtener elementos adecuados de los participantes de las entrevistas.

Algunos de estos elementos han sido utilizados para la implementación de la encuesta de evaluación de la versión inicial de AVISPA como la metodología de la captura por herramientas como la grabación de audio y video de la interacción de las personas con la herramienta.

#### 1.6.2.2. Segunda evaluación de AVISPA (Sección 4.3)

Una de las evaluaciones internas de AVISPA, consiste en la evaluación 2 que lleva a cabo la ejecución de ruta básica por caja blanca, con el propósito de evaluar la consistencia de rutas de los algoritmos de AVISPA, los contenidos de las colecciones, la importación de los elementos de proceso en XML y de las visualizaciones de planos y patrones de la herramienta de visualización.

El trabajo de N. Srinivas et al. [80] presenta una revisión de la literatura de las técnicas para pruebas tanto de caja negra como de caja blanca. El trabajo también enuncia diversas pruebas de software existentes con ejemplos de cada una de las pruebas halladas. Métodos tales como pruebas de bifurcación, pruebas de estados de cuenta, pruebas de cobertura de condición, pruebas de cobertura de condición múltiple, de la misma manera derivan los casos de prueba y los datos de prueba para los métodos de prueba de caja negra como: partición de equivalencia y análisis de valor de límite.

Para éste trabajo se incluyen las pruebas de caja blanca de ruta básica como método para mejorar la programación dentro de AVISPA.

### 1.6.2.3. Tercera evaluación de AVISPA (Sección 4.4)

Con miras de llevar una evaluación más transversal de los planos / patrones, se realizaron las pruebas unitarias de software de AVISPA, con el objetivo de cubrir todo el proceso desde la importación (mapeo) de elementos de proceso en elementos entendibles para AVISPA hasta cada una de las visualizaciones de software de AVISPA.

El trabajo de J. Dooley [85] explica de forma detallada la importancia de las pruebas unitarias en el desarrollo, los tipos de pruebas unitarias, cuándo y cómo aplicarlo definiendo pruebas de desarrollo antes de la realización del código siguiendo metodologías de desarrollo ágil y cómo escribir casos de prueba en Java con JUnit.

La plataforma Moose, donde AVISPA se basa, también soporta pruebas unitarias. AVISPA organiza su código en dos grandes componentes que es el núcleo de la implementación (Core) y las pruebas asociadas (Tests) que implementan las pruebas unitarias de la herramienta de visualización. Así, las pruebas unitarias, han sido elementos cruciales para la implementación adecuada de las visualizaciones de la herramienta, ya que permitieron un entendimiento más amplio del código, y a su vez, de la arquitectura de las extensiones de código de Moose.

### 1.6.2.4. Cuarta evaluación de AVISPA (Sección 4.5)

La variabilidad de procesos de software ha sido un elemento crítico en la disminución de errores de clasificación de AVISPA así también de una alta complejidad de entendimiento al modelo de negocio. Como estrategia para abordar pruebas sobre los planos y patrones de error que necesitan soportar la resolución de la variabilidad, se ha definido la ejecución de pruebas de caja negra por casos de uso donde definen en detalle los flujos de las reglas de resolución de variabilidad propuestas por SPEM 2.0

El trabajo de [86] ilustra la elaboración, prueba y ejecución de casos de uso utilizando lenguaje formal para lograr evitar ambigüedades en la elaboración de los mismos. Así, este trabajo utiliza elementos de lenguaje formal para la definición de reglas de variabilidad en relación con la resolución de los elementos de entrada y salida de elementos de proceso en SPEM 2.0 instanciadas los requerimientos de AVISPA (Ver sección 2.4) para probar las resoluciones de las relaciones de variabilidad por medio de casos de uso.

### 1.6.2.5. Quinta evaluación de AVISPA (Sección 4.6)

Para contrastar la diferencia entre versiones, es necesario realizar una evaluación que permita verificar si existen diferencias significativas entre la antigua y la nueva versión de AVISPA. Así, verificar que sí existe una mejora de la clasificación de los elementos de proceso de la nueva AVISPA. Para lo cual se ha optado por la implementación de los procedimientos estadísticos para el análisis de varianza ANOVA.

El trabajo de F. Sanchez et al. [87] ilustra la implementación de un diseño de experimentos para mejorar una red neuronal para predecir el grosor de la capa de cromo

en un proceso de cromado duro. Para el presente trabajo utilizan el análisis de varianza ANOVA aplicada a las redes neuronales para contrastar mejoras de clasificación de las redes neuronales.

En el presente trabajo se incorporan elementos de F. Sanchez et al. [87] para la elaboración del diseño del experimento e implementación de análisis de varianza ANOVA.

## 1.7. Estructura de la investigación

El presente trabajo está organizado en la siguiente estructura:

- **Capítulo 1. Introducción:** Contiene una explicación inicial del contenido de la tesis. También presenta los resultados de la revisión sistemática aplicada al contexto de métricas de [AVISPA](#) y la estructura general de la metodología para abordar la solución de los objetivos
- **Capítulo 2. Estado del Arte:** Ilustra los artículos de la revisión sistemática de la literatura seleccionados y mencionados en la sección 1.4. También contiene otros artículos relacionados sobre los cuales también se basa la investigación.
- **Capítulo 3. Análisis y Visualizaciones de Modelos de Proceso Software con AVISPA:** Este capítulo contiene la estructura tecnológica y los procedimientos que aplica a los modelos de proceso en [SPEM 2.0](#) para el cálculo de métricas y la construcción de las visualizaciones.
- **Capítulo 4. Evaluación de AVISPA:** Contiene el sistema para la verificación y validación de la nueva construcción de [AVISPA](#), cuenta también con la descripción del modelo de las dos fases ilustradas en la sección 1.6.
- **Capítulo 5. Conclusiones, limitaciones, trabajo futuro y aportes adicionales:** Ilustra los principales hallazgos de la tesis, mencionando los elementos más relevantes y determinantes del direccionamiento de la investigación. Así mismo, se exponen los resultados obtenidos dentro y fuera del ámbito de la tesis.

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

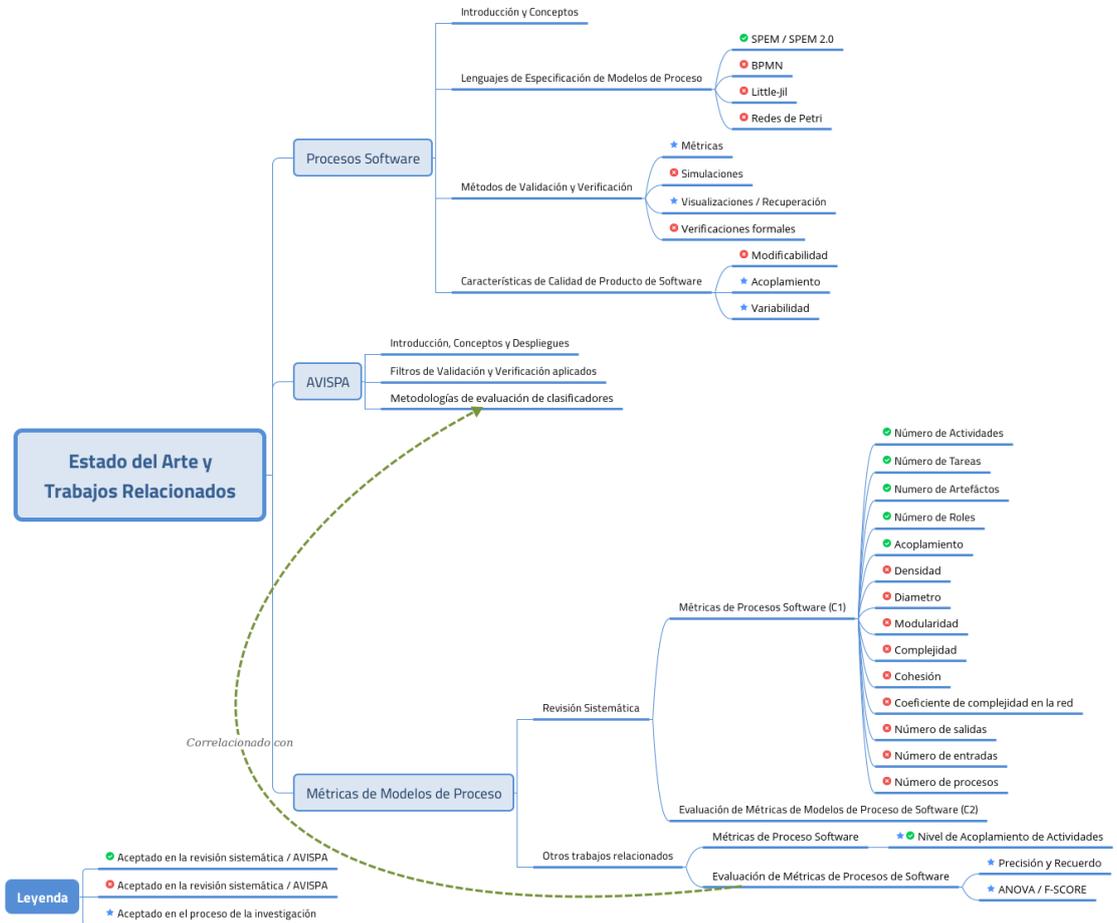
## Capítulo 2

# Estado del arte y trabajos relacionados

### 2.1. Marco conceptual

La estructura de la base conceptual del presente proyecto de investigación gira en torno a 3 grandes temas, (i) las métricas de modelos de procesos de software, (ii) [AVISPA](#), conceptos y validaciones y (iii) precisión, recuerdo y comparaciones entre clasificadores. Estos ejes temáticos están orientados a evaluar y a direccionar el proceso de la tesis respecto a los objetivos y las hipótesis planteadas. El marco conceptual se distribuye como ilustra en la figura [2.1](#). A continuación se describen los ejes temáticos de la investigación:

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual



**Figura 2.1:** Marco conceptual, organización de las temáticas de la investigación

- Métricas de modelos de proceso de software:** Las métricas, son los elementos que **AVISPA** calcula y en los cuales se basa para la visualización en sus diferentes vistas polimórficas. Las métricas definen las principales características de los elementos de las visualizaciones de la herramienta, tal como el color y dimensiones de los nodos, el color de las aristas dependiendo de la intención de la visualización, entre otros.

La importancia de las métricas radica en la información presentada dado que las métricas son el principal objeto de estudio para lograr las clasificaciones y visualizaciones de los elementos de proceso en **AVISPA**. En efecto las métricas son elementos principales de depuración de la herramienta de visualizaciones.

- AVISPA, conceptos y validaciones:** A través de diversos trabajos, la herramienta **AVISPA** ha sido depurada y bifurcada para distintas funcionalidades como soporte al análisis de la variabilidad y la modularidad. Sin embargo, diversos trabajos ilustran problemas en la confiabilidad de las visualizaciones de la herra-

mienta. Esta sección recopila parte de la historia de los diferentes trabajos que se han implementado y principalmente las validaciones y problemas hallados de la herramienta.

- **Precisión, recuerdo y comparaciones entre clasificadores:** Esta sección recopila algunos modelos de evaluación que pueden estar orientados a la validación de [AVISPA](#), es decir, principalmente evalúa la manera en que determina cuando incluir o excluir un elemento de proceso o bajo cuales condiciones la herramienta indica un patrón de error en las visualizaciones.

Esta parte de la investigación ilustra los trabajos involucrados en los métodos para realizar validaciones de clasificadores de distintos grados de libertad desde binarios hasta n-arios. Principalmente se hace énfasis en el método de análisis de varianzas ANOVA [49] donde se determina un factor diferencial de diversos métodos de clasificación de los elementos de proceso construidos en [SPEM 2.0](#), tal como [AVISPA](#) o un ingeniero de proceso experto, para determinar cuan diferente es una distribución de objetos clasificados de otra distribución.

Como insumo de ANOVA para el análisis de la clasificación de elementos de proceso por medio de [AVISPA](#), se implementará el calculo de F-measure [33], el cual determina la precisión de un clasificador de tipo binario y es definido como la media armónica entre la precisión (P) y el recuerdo (R).

Los siguientes trabajos relacionados ofrecen una vista de las generalidades del estado de la investigación actual, así cómo los trabajos en los cuales se basará la investigación para la implementación de su proceso de depuración y refactor de [AVISPA](#).

## 2.2. Procesos Software

### 2.2.1. Introducción y Conceptos

Un proceso es un “conjunto de actividades relacionadas que conducen a la creación de un producto software” [28, 29]. Introduce el concepto de proceso software como un método para determinar de manera sistemática los pasos necesarios para soportar todo lo relacionado con la creación del producto software, además, expone que todos los procesos software deben incluir 4 actividades que son fundamentales para la ingeniería del software: *Especificación del software*, *Diseño e implementación del software*, *Validación del software* y *Evolución del software*. Especifica también que tanto las actividades, como los procesos, deben incluir productos, roles y pre y post condiciones los cuales son sentencias que deben ser verdaderas antes y después de una actividad del proceso [28]. Según I. Jacobson et al. “Un proceso define el quién está haciendo qué, cuándo y cómo construir un producto software o mejorar uno existente” [17].

Feiler P.H. et al. [56] exponen el modelo de procesos como “una representación abstracta de una arquitectura de proceso, de un diseño o definición” para el enten-

dimiento, toma de decisiones y representaciones de los distintos procesos a través de un ciclo de vida. También añade que los modelos de proceso son elementos, cuya abstracción captura estos aspectos de un proceso relevantes para el modelado. Estos elementos de relevancia para la representación del proceso hacen parte del ámbito de la investigación del presente trabajo, pues cada elemento representado es procesado en [AVISPA](#). Así, los modelos de proceso deben contar con la suficiente rigurosidad como para cubrir los casos de ejecución, pero también debe contar con la suficiente flexibilidad para lograr una fácil edición, ejecución, mantenimiento y entendimiento [17].

Modelar el proceso de software es una forma para mejorar el desarrollo y la calidad de las aplicaciones resultantes [42]. La Ingeniería de Procesos de Software (SPE) consiste en modelar, diseñar, mejorar y aplicar procesos utilizando lenguajes de modelado de procesos (PML) [60]. De todos los lenguajes para el modelado de procesos existentes, aquellos basados en productos de trabajo son los más adecuados para modelar los procesos de desarrollo de software [61].

El trabajo de U. Becker realiza una clasificación de los modelos de proceso en descriptivos (que describen la manera de cómo algo realiza en realidad) y prescriptivos (que describen como algo debería realizarse) [57]. O. Armbrust et al. señalan una diferencia característica entre estos 2 tipos de proceso, mientras que un modelo de proceso descriptivo es creado por la observación de los procesos ejecutados actualmente, un modelo de proceso prescriptivo por lo general señala la dirección de todos los temas relevantes para el desarrollo de una pieza de software, otra característica indicada por el mismo autor menciona que un modelo de proceso descriptivo muestra lo que la gente realiza día a día, lo que significa que deja al descubierto las prácticas que se llevan a cabo mientras que un modelo de proceso prescriptivo le indica a la gente el que hacer, lo que implica que debe realizar algunas cosas diferentes de como lo hacen regularmente y cambiar sus hábitos lo que en ingeniería de software es una de las metas más difíciles de lograr. Un despliegue exitoso de una perspectiva de proceso significa un cambio en los hábitos de las personas [20].

Los modelos de proceso deben contar con disponibilidad completa para los roles del proceso, sin embargo, los procesos prescriptivos por lo general cuentan con una característica importante dentro de las organizaciones de software y es que cuentan con la capacidad de ser una guía para personas de cualquier rol de la organización [17]. A diferencia de los modelos descriptivos que ofrecen un mayor campo de acción al no ser tan rígidos y detallados en la documentación.

Por otra parte, este trabajo incluirá únicamente modelos de proceso prescriptivos completos<sup>1</sup> documentados en [SPEM 2.0](#). Esta limitación de la investigación está ligada a la capacidad de éste tipo de modelos para contener información detallada

---

<sup>1</sup>En el ámbito de este trabajo, se considera completo el modelo de proceso que cuente con una estructura de composición adecuada. Una estructura de composición adecuada de elementos de proceso por composición de izquierda (más grande) a derecha (mas atómico) consiste en: procesos de entrega, fases, iteraciones, actividades y, por último, elementos de proceso atómicos como tareas, roles, productos de trabajo, guías, plantillas entre otros. Se sugiere al ingeniero de procesos documentar dicha estructura para que analista de procesos de [AVISPA](#) genere visualizaciones mas adecuadas del proceso en vistas que ilustran la estructura de los elementos de proceso por composición.

que permite representar procesos más informativos y, por lo tanto, permite visualizaciones más completas y focalizadas sobre elementos agrupados por composición en AVISPA [16].

### 2.2.2. Lenguajes de Especificación de Modelos de Proceso

Los lenguajes de especificación de modelos de proceso cumplen el rol de organizar los elementos de un proceso y delimitar sus respectivas interacciones y/o características. Estos elementos a su vez son organizados en ontologías [37] que proveen una descripción formal de los componentes y sus relaciones.

Object Management Group (OMG) es un consorcio dedicado a la creación y mantenimiento de diversos estándares de tecnologías orientadas a objetos. SPEM es un estándar de orientación de objetos creado por la OMG como lenguaje de especificación para generar modelos de proceso. Así mismo, existen múltiples lenguajes de especificación de procesos como BPMN, Little-Jil, UMA, Redes de Petri entre otros.

En el 2002, OMG propuso el metamodelo SPEM en su primera versión de lanzamiento. Con el transcurso del tiempo, los ingenieros de proceso observaron falencias en la construcción del metamodelo. SPEM fue precursor de Unified Method Architecture (UMA) que fue desarrollada por IBM y direccionó varias de sus falencias. OMG, por lo tanto, desarrollo un segundo lanzamiento de la especificación SPEM. IBM ha desarrollado parte de la versión gratuita de la herramienta de modelado para UMA llamada Eclipse Process Framework EPF. Posteriormente IBM implementó la versión comercial llamada Rational Method Composer. A raíz de lo anterior, EPF produce modelos conforme a UMA, sin embargo, no soporta SPEM en su totalidad [58].

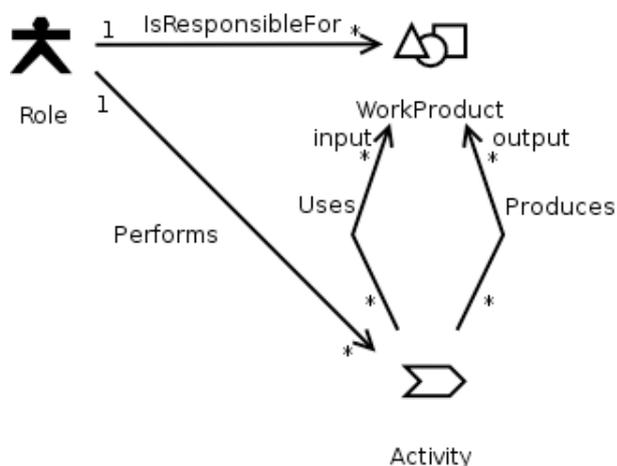
La herramienta de visualización AVISPA utiliza el metamodelo SPEM 2.0 como referencia para la implementación de sus vistas del modelos de proceso documentados en EPF. A continuación se ilustrarán los trabajos involucrados en la incorporación de SPEM / EPF como base de los insumos para las visualizaciones de AVISPA.

#### 2.2.2.1. SPEM 2.0

El trabajo de J. Simmonds et al. [58] ilustra que una de las características AVISPA es la separación de la definición de un proceso y de la entrega final. También contiene una breve historia de la evolución de SPEM 2.0 y de la creación de EPF. Así mismo SPEM es una especificación que está basado en MetaObject Facility (MOF) además de estar definido como un perfil de Unified Modeling Language (UML) [59].

Los elementos de proceso como tareas, roles y productos de trabajo son definidos y almacenados en la Librería de Método. Estas definiciones pueden ser reutilizadas después en definiciones múltiples de proceso. En SPEM, un proceso es una colección de actividades, donde una actividad agrupa roles, productos de trabajo y tareas en uso. Los roles realizan tareas, y los productos de trabajo se usan como artefactos de entrada / salida para las actividades.

Aunque la notación SPEM se parece al UML, tienen estructuras bastante dife-



**Figura 2.2:** Concepto básico para el modelado de procesos en SPEM 2.0 [21]

rentes. En el nivel conceptual, la idea principal del proceso basado en SPEM es la interacción de tres elementos básicos: roles de procesos que son responsables de y ejecutan actividades que consumen y producen productos de trabajo. Roles, productos de trabajo y actividades son todos elementos de definición de proceso. Estos 3 tipos de elementos son los elementos más fundamentales y atómicos para la construcción de procesos. El modelo básico de SPEM para la elaboración de modelos de proceso se ilustra en la figura 2.2.

SPEM también define elementos de ciclo de vida. Las fases e iteraciones son usadas para modelar la estructura dinámica de un proceso. Un ciclo de vida define el orden de las fases, las cuales a su vez puede contener iteraciones. Así, un proceso contiene exactamente un único ciclo de vida.

SPEM también define elementos llamados paquetes, cuya intención consiste en organizar otros elementos de proceso desde la vista de la autoría de procesos, del ensamblaje y de la reusabilidad. El propósito de los paquetes es dividir las descripciones del proceso en partes autocontenidas. Estas partes pueden ser ubicadas bajo la configuración y gestionado de versiones y usado para el ensamblaje y seguimiento de modelos de proceso de software.

Los componentes de proceso son especializaciones de los paquetes. Un componente de proceso es una sección de descripción de procesos internamente consistentes y autocontenidos que pueden ser reusados con otros componentes de proceso para ensamblar el proceso completo. Los componetes de proceso pueden importar un conjunto no arbitrario de elementos de definición. La disciplina es una especialización de los componentes de proceso y es usado para representar actividades entre un área de proceso común, tal como diseño, implementación, pruebas, entre otros. El proceso en sí mismo es una especialización de un componente de proceso, que está destinado a ser únicamente un proceso completo. Así, el ensamblaje de los procesos es realizado a través de componentes de proceso. [61].

### 2.2.3. Métodos de Validación y Verificación (V & V)

Existen diversos métodos de V&V, sin embargo, se mencionarán algunos de los más relevantes para la investigación.

El trabajo de Ruiz et al. [63] realiza una revisión sistemática de los trabajos relacionados con los usos y aplicaciones de modelos de proceso realizados en SPEM 2.0. Este estudio realiza una categorización de los trabajos encontrados por versión del lenguaje de modelado, evaluación de procesos, configuración de proceso y elicitación, adaptabilidad de procesos, modelado de procesos y métodos de V&V entre otros ámbitos.

#### 2.2.3.1. Métricas

El trabajo de Laucerra et al. [64] propone un metamodelo que resulta de la integración entre un metamodelo de métricas y la ejecución de metamodelos de procesos de software y de negocio. Por lo que, al ejecutar los procesos de software de los metamodelos, las métricas deben estar en todo momento bajo control al momento de la creación de un modelo de proceso. Este trabajo también involucra características o atributos de calidad de producto aplicados al proceso como el acoplamiento, entre otros.

#### 2.2.3.2. Simulaciones

El trabajo de S. Park et al. [65] ilustra una propuesta de implementación para realizar simulaciones de software en modelos de procesos basados en SPEM aplicando reglas de transformación de modelos. Uno de los principales resultados de la investigación es la resolución de 3 problemas: (i) reducir el costo de desarrollar un modelo de simulación, (ii) reducir la complejidad de un modelo de simulación y (iii) incrementar la modularidad de un modelo de simulación.

#### 2.2.3.3. Verificación formal

Con el objetivo de disminuir la cantidad de errores de ejecución de proceso de software, el trabajo de E. Pereira [62] ilustra una serie de reglas bien definidas para verificar la consistencia de procesos software basados en SPEM 2.0. Este trabajo incluye una implementación adicional al lenguaje de metamodelado MOF para incluir sSPEM 2.0, una especificación que implementa una librería de consistencia de método que contiene instancias concretas de los elementos y relaciones de SPEM 2.0 que han sido chequeadas usando reglas de consistencia de sSPEM 2.0. Este trabajo es relevante debido a que sus reglas de construcción adecuada de procesos pueden ser de utilidad en el ejercicio de la documentación de modelos de proceso y, por lo tanto, en la correcta generación de las vistas en AVISPA.

El trabajo de V. Gruhn [14] detalla la forma de la realización de la verificación formal de modelos de proceso de software de modelos de proceso realizados en un lenguaje

de especificación de modelos de procesos llamado redes de FUNSOFT. Este trabajo menciona como se puede llevar a cabo la validación y verificación de modelos de proceso por medio de este método y, así mismo, realiza un ejemplo de su ejecución con varias redes de Petri.

### **2.2.3.4. Visualización / Recuperación**

El trabajo de M. Reichert [67] aborda el problema de la visualización de grandes modelos de negocio, los retos, las técnicas y las aplicaciones. En particular, el trabajo expone las dificultades de visualización refiriéndose a la sobrecarga visual que pueden contener los modelos visuales de proceso de software. Este trabajo también expone que la reducción u omisión de elementos de proceso en modelos grandes que no son relevantes sobre un contexto dado puede contribuir a una mejor comprensión e inclusive, disminuye el nivel de complejidad en su ejecución. Así mismo ilustra que los nodos (elementos de proceso) de las visualizaciones deben soportar personalización por parte del usuario. Por último, los formatos de las visualizaciones deben contar tanto con tablas que ilustren los elementos visualizados como con grafos que ilustren las interacciones de los elementos.

El trabajo anterior es de importancia para la visualización de modelos de proceso en AVISPA debido a que ofrece sugerencias previamente probadas por otros investigadores para la implementación de visualizaciones más completas y comprensibles.

### **2.2.4. Atributos de Calidad de Producto / Proceso de Software**

#### **2.2.4.1. Modularidad**

El trabajo de H. Reijers et al. [68] ilustran las ventajas de tener un modelo de proceso de software modularizado. La modularización de modelos de proceso de software facilita la reutilización, la escalabilidad de los modelos de proceso documentado y, al igual que el trabajo de visualización de M. Reichert [67], facilitan el entendimiento de los modelos. En este trabajo proponen diferentes tipos de modularización de modelos de proceso (i) modularización de operaciones que consiste en la selección de partes del proceso para visualización de diagramas de flujo, (ii) modularización de prerequisites del proceso, que son las condiciones que deben ser tomadas en cuenta al momento de modelar procesos (p. ej. Si un modelo de proceso cuenta con mas de 50 elementos debería ser descompuesto en submodulos) y (iii) modularización a selección, que significa la selección de partes de modelos de proceso para la modularización.

El trabajo de F. Cardenas et al. [70] expone el análisis visual de la modularidad de modelos de proceso de software por medio de la herramienta AVIMO-PS que a su vez es una versión de AVISPA. La propuesta incluye un conjunto de métricas y vistas polimétricas para la visualización de modelos de proceso según 3 tipos de elementos de modelos de proceso de software: Roles, tareas y productos de trabajo. La propuesta, tal como AVISPA, ilustra patrones de error sobre las visualizaciones

para indicar el nivel de acoplamiento / cohesión de los elementos de los modelos de proceso evaluados.

### 2.2.4.2. Variabilidad

El trabajo de T. Martínez-Ruiz et al. [66] ilustran un avance en la definición de mecanismos de variabilidad en líneas de procesos de software en SPEM 2.0. Este trabajo indica que los actuales mecanismos de variabilidad que contiene SPEM 2.0 no soportan adecuadamente la variabilidad de un modelo de proceso como se esperaría. Para abordar el problema, el trabajo propone mecanismos para la definición de elementos de variabilidad de modelos de proceso de software dada la variabilidad de flujos de ejecución inherente en las líneas de proceso de software. El trabajo finalmente propone elementos taxonómicos de variabilidad para SPEM 2.0 como puntos de variabilidad y variantes. Posteriormente proponen la creación de una extensión de SPEM 2.0 llamada vSPEM [19] donde ilustran la implementación de la propuesta de los elementos de variabilidad en el modelo de proceso de software COMPETISOFT.

Tal como en la investigación de F. Cardenas et al. [70], el trabajo de J. Salgado et al. [69] propone un avance en el análisis de la variabilidad de modelos de proceso de software sobre la herramienta vAVISPA, que a su vez es una versión de AVISPA. Dado a los elementos inherentes del lenguaje de especificación de modelos de proceso SPEM 2.0, vAVISPA incorpora la resolución de la variabilidad de los elementos de proceso que han sido documentados con ésta característica. Los resultados de la resolución de variabilidad de vAVISPA ilustran una mejoría de la representación de las visualizaciones. Por ejemplo, los planos del modelo de proceso OpenUP desde la herramienta AVISPA ilustran que existen nodos desconectados en ciertas visualizaciones de tareas. Sin embargo, desde la herramienta vAVISPA, dichos nodos no se encuentran aislados en las mismas construcciones visuales por medio de la resolución de variabilidad que la herramienta posee, lo que representa una mejora notable en la confiabilidad de la herramienta y, por lo tanto, en la representación visual de AVISPA de los modelos de proceso que se documenten en SPEM 2.0.

### 2.2.4.3. Modificabilidad

El trabajo de F. Bachmann et al. [55] ilustra los mecanismos para alcanzar la modificabilidad dentro del ámbito de los productos de ingeniería de software. Principalmente ilustra una serie de tácticas a tomar en cuenta para incrementar la modificabilidad de los productos, principalmente incrementar la cohesión y disminuir el acoplamiento como factores esenciales de la modificación de componentes de software que también son aplicables a los componentes de procesos software según L. Osterweil [42]. Así mismo disminuye el nivel de abstracción de las tácticas incluyendo sugerencias sobre la distribución adecuada de responsabilidades, encapsulamiento de objetos, mantenimiento de la coherencia semántica entre otros elementos.

## 2.3. AVISPA

La visualización de los modelos de proceso es considerado un método de validación y verificación dentro del ámbito de los procesos. Las visualizaciones de AVISPA han sido usadas en múltiples trabajos relacionados con la implementación de modelos de proceso elaborados en SPEM 2.0 [3, 5, 71].

### 2.3.1. Introducción y Conceptos

El trabajo de J. Hurtado et al. [16] ilustra los conceptos básicos sobre la visualización de modelos de procesos SPEM 2.0 con AVISPA que consiste en una herramienta que construye planos y resalta sobre ellos posibles patrones de error. Los patrones de error son anomalías recurrentes en la documentación de modelos de proceso y son gráficamente resaltados de los demás elementos representados. El capítulo 3 ilustra a profundidad la construcción y avances de la herramienta de visualización de procesos.

#### 2.3.1.1. Métricas establecidas antes de la refactorización

A continuación, se mencionarán las métricas más relevantes para la investigación. Aquellas que han presentado problemas de inconsistencia o relevancia en la clasificación (Falso Positivo (FP) y PF! (PF!)) de elementos de proceso:

- **Número de roles de entrada y de salida:** Los modelos de proceso no contaban con resolución de variabilidad de manera adecuada para este tipo de elemento de proceso. Por lo cual, se ha definido que esta métrica no contenía el número de roles que debería tener para resolver adecuadamente la variabilidad de este tipo de elementos de proceso.
- **Número de productos de trabajo de entrada y de salida:** Al igual que el anterior punto, la herramienta no contaba con la resolución de variabilidad para este tipo de elementos de proceso, por lo que se determinó inconsistencias en los cálculos de los elementos de proceso de este tipo.
- **Número de tareas de entrada y de salida:** En el plano de resolución de variabilidad de tareas han sido detectados algunos problemas en el cálculo de las relaciones de variabilidad de tipo composición. Por lo cual se ha determinado que en algunos nodos de los elementos representados en este patrón de error presentaban inconsistencias.
- **Número de roles aislados:** Por resolución de variabilidad, también es común encontrar clasificaciones erróneas en la implementación de esta métrica, dado que la resolución de variabilidad de roles no ha sido tomada en cuenta en la elaboración de patrones de error, presentando falsos roles aislados.

- **Número de tareas aisladas:** Debido a la misma resolución de variabilidad, los elementos de proceso de software como tareas aisladas, también presentaban falsos positivos, esto era debido a que existían tareas que solamente estaban conectadas por medio del tipo de relación de variabilidad: "contribución".
- **Número de productos de trabajo aislados:** La variabilidad al interior de productos de trabajo no estaba presente en anteriores versiones de AVISPA, presentando un falso número de productos de trabajo aislados.
- **Número de plantillas asociadas a elementos de trabajo:** Bajo consideración del trabajo de M. Bastarrica et. al. [41] esta métrica carece de suficiente soporte para determinar una decisión adecuada sobre diferentes modelos de proceso de software evaluados.
- **Número de artefactos desperdiciados:** Como en el anterior punto, el trabajo de M. Bastarrica et. al. [41] ha determinado que esta métrica no es un factor decisivo para generar cuellos de botella la ejecución de modelos de procesos.
- **Número de roles sobrecargados:** Según el trabajo de M. Bastarrica [41] era necesario establecer una diferencia entre elementos como roles que producen muchos artefactos, y roles que cumplen esta condición y a su vez son responsables de artefactos, para evaluar diferentes modelos de proceso de empresas chilenas de desarrollos de software. Esta última no ha sido observada en los patrones de error de la versión anterior, por lo que es necesario implementarla. Respecto al número de roles sobrecargados, se estableció el mismo umbral especificado por dicho trabajo en la nueva versión de AVISPA ya que se determinó que sí era un factor numérico determinante en la ejecución de los modelos de proceso hallados.

### 2.3.2. Filtros de Validación y Verificación Aplicados

El trabajo de M. Bastarrica et al. [41] ilustra cómo la calidad de modelos de proceso formalizados afecta su adopción en las organizaciones. El trabajo propone varias hipótesis validadas por medio de AVISPA las cuales, indirectamente, validan las visualizaciones de la herramienta al usarlas para validar su utilidad en diversos modelos de procesos de empresas de Chile. Las hipótesis de validación afirman diversos motivos de construcción de modelos de proceso que pueden desembocar en retrasos en la ejecución del proceso apoyándose en la herramienta de visualizaciones AVISPA. Las conclusiones de algunas hipótesis indican que algunos de los resultados de la investigación no son suficientemente determinantes para indicar si existe o no una correlación con el atraso en la ejecución del proceso para algunos patrones de error de la herramienta y, que a su vez, es necesario realizar más investigación al rededor de éstos. Lo anterior sugiere, de manera indirecta, que es necesario considerar la construcción / utilidad de algunos de los planos generados por la herramienta de visualizaciones de AVISPA en el proceso de V & V de modelos de proceso de software definidos en SPEM 2.0.

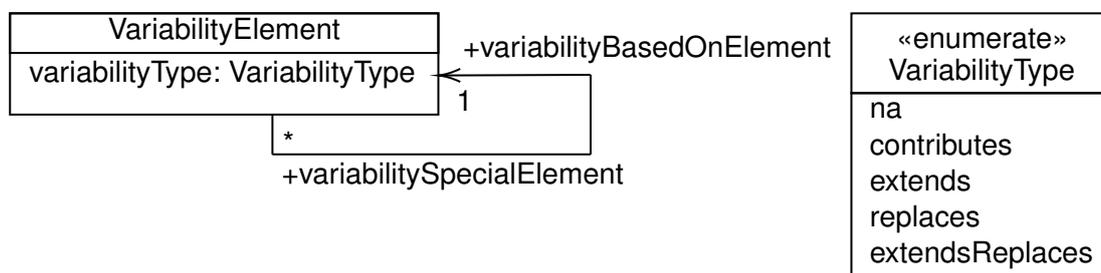


Figura 2.3: Diagrama de variabilidad del metamodelo SPEM 2.0

### 2.3.3. Metodología de evaluación de Clasificadores

AVISPA, al ser un tipo de clasificador, es necesario descartar el nivel de correctitud con el que ordena los elementos de proceso en cada una de las vistas generadas por la herramienta de visualización de modelos de proceso de software. En el capítulo 4 se ilustra mejor el proceso de evaluación, principalmente alrededor de los temas que se indiquen a continuación.

#### 2.3.3.1. Precisión, Recuerdo y F-measure

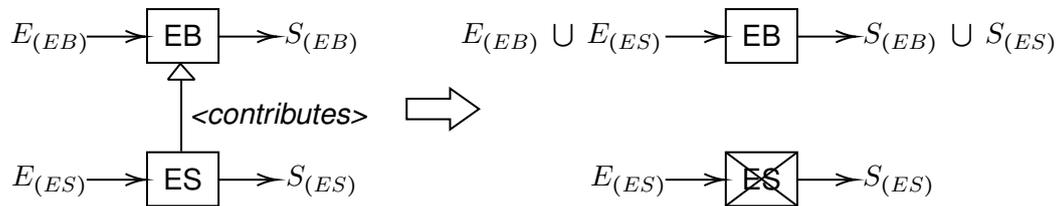
La herramienta AVISPA ha sido calibrada anteriormente con la aplicación de la métrica F-measure bajo el trabajo de J. Hurtado et al. [16] para la evaluación de los patrones: “proyectos independientes” y “tareas multipropósito”. Este trabajo es relevante debido a que ilustra una manera adecuada de aplicar el método, identificar anomalías y recuperar información, más específicamente, recuperación de elementos de proceso, donde las métricas son un factor determinante para la correcta generación de patrones de error por parte de AVISPA.

## 2.4. Variabilidad en SPEM 2.0 instanciado en AVISPA

Parte de la investigación incluye la instanciación de las reglas de las relaciones de variabilidad de SPEM 2.0 instanciadas a los elementos que AVISPA captura de los modelos de proceso para su respectiva visualización en los planos y patrones de error. Los 4 tipos de relaciones de variabilidad en SPEM 2.0 son conocidos como *contribución*, *extensión*, *reemplazo* y *extensión-reemplazo*, tal como se ilustra en la figura 2.3.

### 2.4.1. Relaciones de variabilidad

Para disminuir la complejidad de interpretación, se omiten los atributos internos de los elementos y las relaciones de variabilidad se reducen de ejemplos de contexto, tal como se aplican en el informe técnico del OMG de SPEM 2.0 [21] Anexo B, a visualizaciones concretas de las relaciones de entrada y de salida de relaciones de variabilidad uno a uno y uno a muchos entre elementos de proceso no concretos, ya



Donde:

- $E_{(EB)}$ : elementos de entrada de  $EB$ .
- $S_{(EB)}$ : elementos de salida de  $EB$ .
- $E_{(ES)}$ : elementos de entrada de  $ES$ .
- $S_{(ES)}$ : elementos de salida de  $ES$ .

**Figura 2.4:** Resolución de variabilidad de tipo *contribución*

que las reglas de resolución por cada tipo de relación de variabilidad son las mismas para toda clase de elementos de proceso:

Sea:

- **EB** el elemento base de la relación de variabilidad
- **ES** el elemento especial de variabilidad

Cada una de las reglas sobre los tipos de relaciones de variabilidad de [SPEM 2.0](#) instanciadas a [AVISPA](#) serán expuestas a continuación:

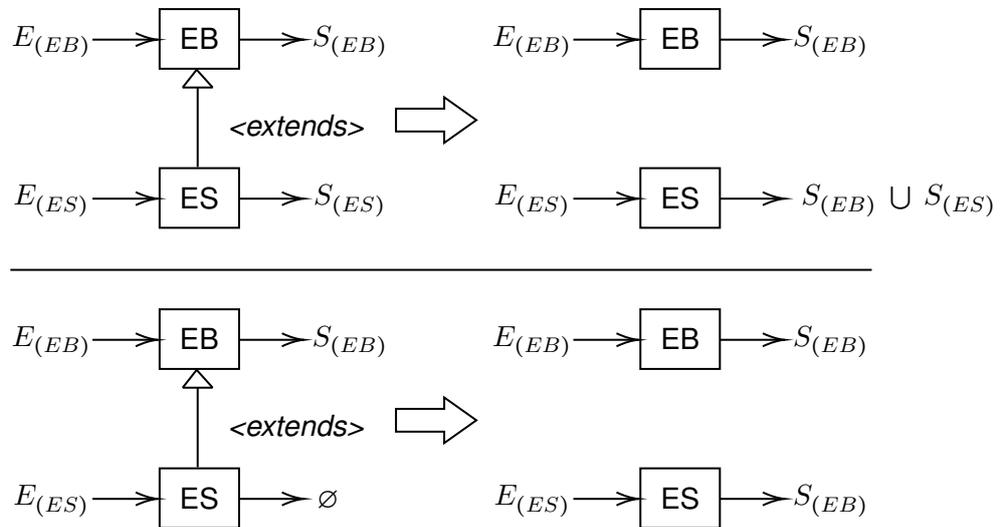
#### 2.4.1.1. Contribución

- Las relaciones de entrada y salida del elemento **ES** se añaden a **EB**.
- Si una relación de entrada o salida del elemento **ES** ya se encuentra en **EB** se ignora, es decir, no se resuelve.
- Este tipo de variabilidad permite que múltiples **ES** puedan ser relacionados con un **EB**.
- **ES** se oculta

La figura [2.4](#) ilustra la resolución de éste tipo de variabilidad.

#### 2.4.1.2. Extensión

- Las salidas de **EB** se añaden a **ES** solo si **ES** no tiene salidas.
- Las instancias de asociación de salida con cardinalidad “muchos” de **EB** se añaden a **ES**.



Donde:

- $E_{(EB)}$ : elementos de entrada de  $EB$ .
- $S_{(EB)}$ : elementos de salida de  $EB$ .
- $E_{(ES)}$ : elementos de entrada de  $ES$ .
- $S_{(ES)}$ : elementos de salida de  $ES$ .

**Figura 2.5:** Resolución de variabilidad de tipo *extensión*

- Los elementos **EB** y **ES** no se ocultan

La figura 2.5 ilustra la resolución de éste tipo de variabilidad.

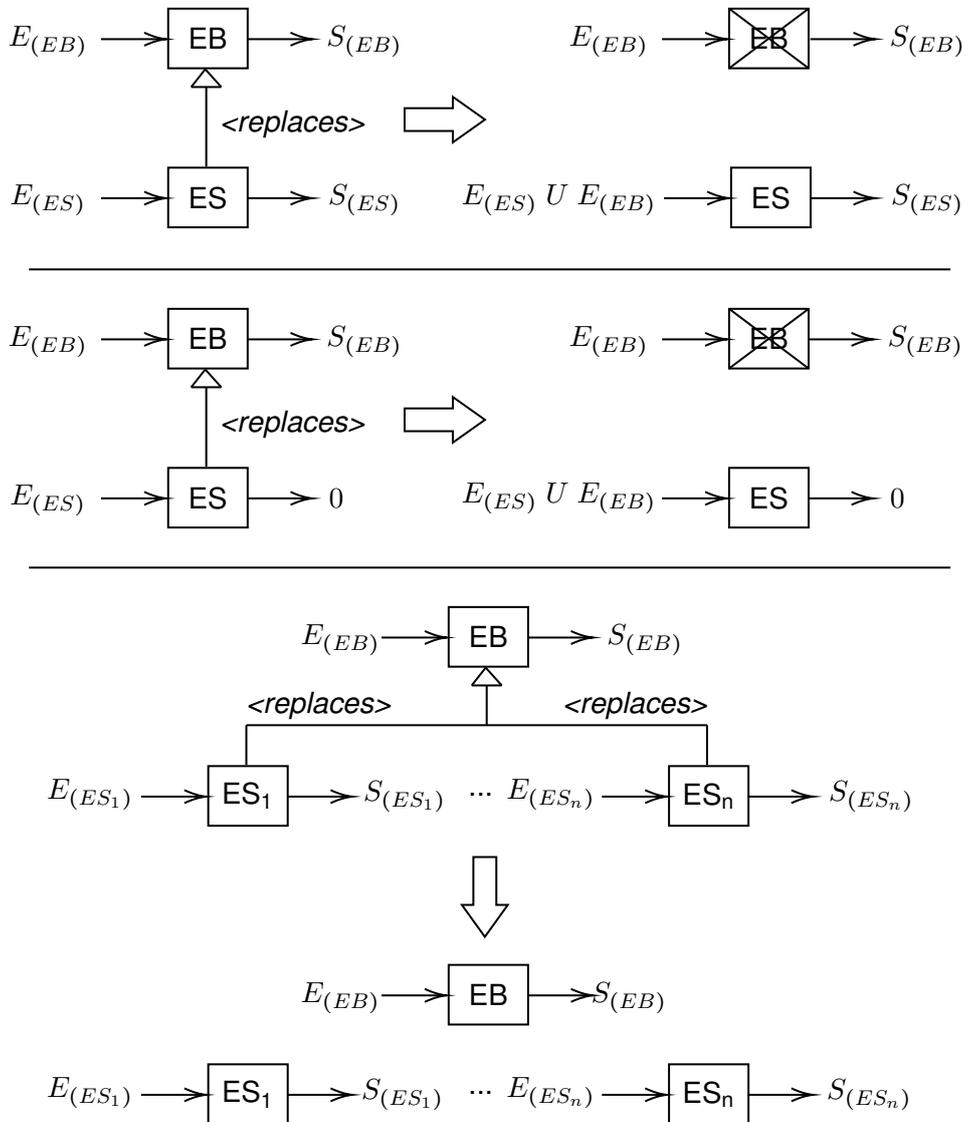
#### 2.4.1.3. Remplazo

- Las salidas de **ES** sobrescribirán todas las relaciones de salida del elemento resultante independientemente si **ES** posee o no salidas.
- Las entradas de **ES** se unen con las de **EB** en el elemento resultante.
- **EB** se oculta, en caso de resolver la relación de variabilidad.
- Si existe mas de un **ES** para un **EB** la resolución de variabilidad no se lleva a cabo.

La figura 2.6 ilustra la resolución de éste tipo de variabilidad.

#### 2.4.1.4. Extensión-Remplazo

- Si **ES** define asociaciones de salida, éstas reemplazarán todas las asociaciones de salida de **EB**.



Donde:

- $E_{(EB)}$ : elementos de entrada de  $EB$ .
- $S_{(EB)}$ : elementos de salida de  $EB$ .
- $E_{(ES)}$ : elementos de entrada de  $ES$ .
- $S_{(ES)}$ : elementos de salida de  $ES$ .
- $E_{(ES_1)}$ : elementos de entrada de  $ES_1$ .
- $S_{(ES_1)}$ : elementos de salida de  $ES_1$ .
- $E_{(ES_n)}$ : elementos de entrada de  $ES_n$ .
- $S_{(ES_n)}$ : elementos de salida de  $ES_n$ .

**Figura 2.6:** Resolución de variabilidad de tipo *reemplazo*

Elementos de proceso	Entrada	Salida
Roles	primary, performs (tarea) performs, additional (tarea)	responsible (artefacto)
Tareas	mandatory, output (artefacto)	input (artefacto) primary, performs (rol) performs, additional (rol)
Artefactos	primary, performs (tarea) performs, additional (tarea)	responsible (artefacto)

**Tabla 2.1:** Relaciones de entrada y salida por elemento variable de proceso

- Si **ES** no define asociaciones de salida, se conservarán todas las asociaciones de salida de **EB**.
- Las asociaciones de entrada de **EB**, se añaden a las de **ES**.
- **EB** se oculta, en caso de resolver la relación de variabilidad.
- Si existe mas de un **ES** para un **EB** la resolución de variabilidad no se lleva a cabo.

La figura 2.7 ilustra la resolución de éste tipo de variabilidad.

Por último, si existen diversas relaciones de diferente tipo de múltiples **ES** a un solo **EB**, primero siempre se resolverán los elementos de tipo *contributes*, y luego las relaciones restantes. La variabilidad siempre se resolverá comenzando desde el tope de los **EB**.

## 2.4.2. Consideraciones adicionales

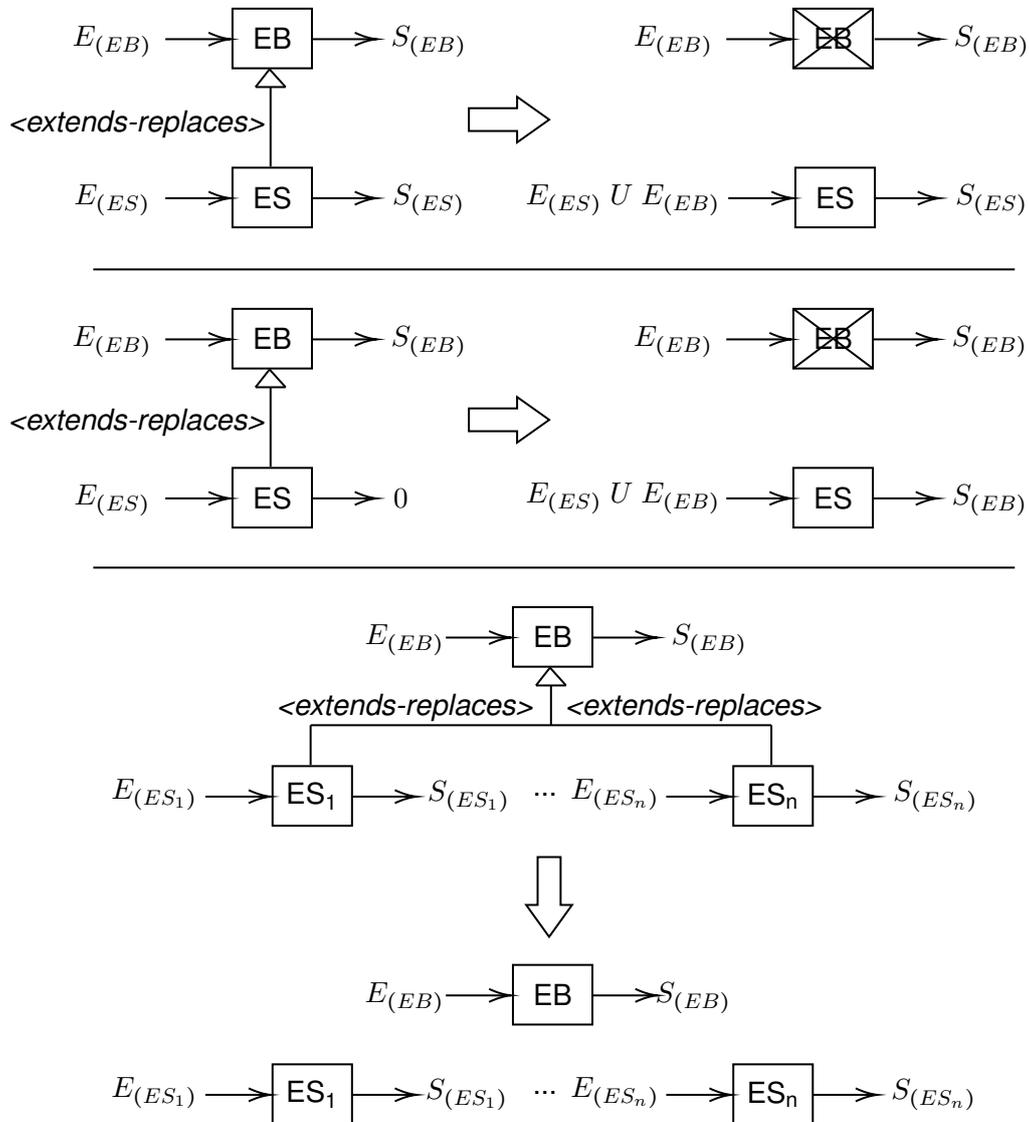
En la anteriores secciones se han establecido las reglas de **SPEM 2.0** instanciadas a los planos y patrones de **AVISPA**. Sin embargo, estas consideraciones no son suficientes para lograr una resolución de la variabilidad en los planos de forma satisfactoria. A continuación se enuncian algunos elementos tomados en cuenta para lograr los elementos mínimos de una resolución adecuada de la variabilidad:

### 2.4.2.1. Definición de entradas y salidas

La definición de las reglas de resolución de modelos de proceso por si mismas son insuficientes, ya que es necesario definir cuales relaciones entre elementos de proceso son entradas y salidas. Es necesaria la clasificación dado que las reglas de resolución de modelos de proceso establecen modificaciones de las relaciones de entradas y salidas de los elementos de proceso. La tabla 2.1 ilustra cada una de las relaciones clasificadas en entradas y salidas.

### 2.4.2.2. Relación del elemento base a los elementos específicos de variabilidad

El metamodelo **SPEM 2.0** incluye elementos de variabilidad dentro de si mismos para tener trazabilidad de los elementos variables en composición. En la figura 2.3



Donde:

- $E_{(EB)}$ : elementos de entrada de *EB*.
- $S_{(EB)}$ : elementos de salida de *EB*.
- $E_{(ES)}$ : elementos de entrada de *ES*.
- $S_{(ES)}$ : elementos de salida de *ES*.
- $E_{(ES_1)}$ : elementos de entrada de  $ES_1$ .
- $S_{(ES_1)}$ : elementos de salida de  $ES_1$ .
- $E_{(ES_n)}$ : elementos de entrada de  $ES_n$ .
- $S_{(ES_n)}$ : elementos de salida de  $ES_n$ .

**Figura 2.7:** Resolución de variabilidad de tipo *extensión - remplazo*

se ilustra detalladamente la relación de composición tanto hacia los elementos base superiores como a los elementos específicos inferiores.

Este tipo de relación debe ser tomada en cuenta para solucionar restricciones específicas sobre las reglas de resolución de 2 tipos de variabilidad: *reemplazo* y *extensión-reemplazo* los cuales especifican que si hay más de 1 elemento de sus respectivos tipos compartiendo un elemento base, la resolución de la variabilidad no se puede llevar a cabo. Esta regla adicional presupone un esfuerzo sumado para a resolución de las reglas de variabilidad de [AVISPA](#).

### 2.4.2.3. Planos afectados y tipos de variabilidad soportados en [AVISPA](#) por ocultamiento de elementos variables

Al tener en cuenta que la contribución es el único tipo de variabilidad que, en todos los caso, oculta uno de los elementos involucrados en la relación de variabilidad, específicamente los **ES** (ver sección 2.4.1), los demás tipos de variabilidad siempre deben ser resueltos en todos los planos y patrones de [AVISPA](#) cuyos elementos de proceso la soporten a pesar de estar ocultos en por las reglas de variabilidad.

Esto implica que, la resolución de los tipos de variabilidad *extensión*, *reemplazo* y *extensión-reemplazo* deben resolverse en todos los planos y patrones de [AVISPA](#), para los elementos que los soporten que son tareas, roles y artefactos. Esto se debe a las reglas de los tipos:

- **Extensión:** ni el **EB** ni el / los **ES** se ocultan.
- **Reemplazo y extensión-reemplazo:** **EB** no se oculta en caso de que existan múltiples **ES** apuntándolo.

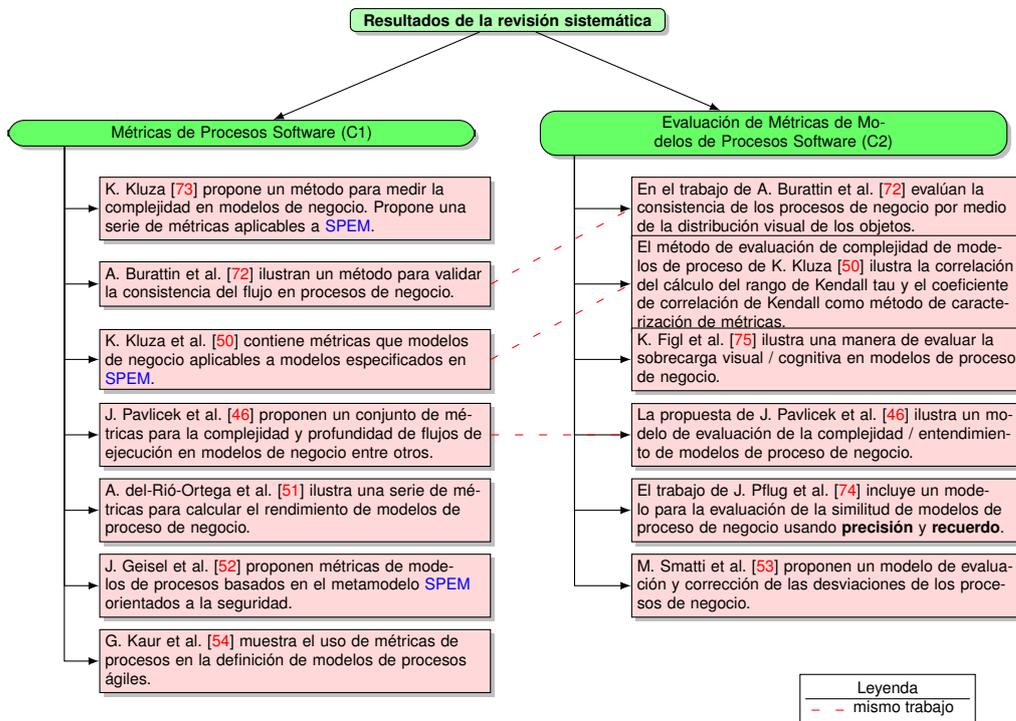
## 2.5. Métricas de modelos de proceso

### 2.5.1. Trabajos encontrados en la revisión sistemática

En esta sección se ilustran los diferentes trabajos que han clasificado en la revisión sistemática realizada en el capítulo 1. Las referencias encontradas en la sección 1.5.1 se detallan a profundidad en la presente sección. Cada trabajo de la clasificación pertenece a una subcategoría del conocimiento que influye en el rumbo de la investigación. Dichas categorías se pueden consultar en la sección 1.4.3.

Cada uno de los artículos de la revisión sistemática ha pasado por un exhaustivo proceso de búsqueda. La gran cantidad de esfuerzo invertido para encontrar diversos artículos que aporten a los trabajos relacionados se debe a la escasez de los mismos en las bases de datos del protocolo de la revisión, lo cual sugiere un mayor nivel de esfuerzo para encontrar más artículos que clasifiquen en el protocolo.

La principal dificultad hallada consistió en la definición del alcance de las métricas de modelos de proceso en la búsqueda. Algunos artículos encontrados se refieren



**Figura 2.8:** Artículos seleccionados de la revisión sistemática

a métricas de rendimiento o de ejecución de los procesos como líneas de código producidas o nivel de eficiencia de la ejecución del mismo.

Otra dificultad hallada consiste en el hallazgo de métricas de proceso directamente de SPEM. Los artículos hallados son asociaciones directas de modelos de negocio aplicables a modelos realizados con SPEM. Así mismo, otra dificultad en la búsqueda de artículos consistió en que las métricas están orientadas más a BPMN que a SPEM.

Así mismo, el alcance de la investigación, en efecto, ha sido acotado a una temática de búsqueda muy específica, por lo que la dificultad en el hallazgo de trabajos ha sido incrementada.

La figura 2.8 ilustra una clasificación de los artículos incluyendo la justificación de su inclusión en el proceso de la revisión sistemática.

Las siguientes secciones describen los artículos seleccionados a profundidad, principalmente los más involucrados en el desarrollo del presente trabajo.

### 2.5.1.1. Métricas de Procesos Software (C1)

En ésta sección se clasificaron los artículos que contienen información sobre las métricas usadas para modelos de proceso documentados en el lenguaje de especificación de procesos SPEM o en otros lenguajes donde las métricas también pueden

ser aplicadas en [SPEM](#).

El trabajo de K. Kluza [73] propone un método para medir la complejidad en modelos de negocio involucrando una serie de métricas aplicables a [SPEM](#). El lenguaje de especificación de procesos BPMN contiene el concepto de reglas donde restringe los flujos de ejecución de los modelos de proceso para obtener un comportamiento deseado. Las métricas definidas en el trabajo de Kluza involucran el uso de métricas relacionadas con las reglas de BPMN que no son soportadas por [SPEM](#) por lo que la validación de métricas propuesto no son compatibles con el ámbito del problema del presente trabajo, sin embargo, se incluyen diversas métricas que son aplicables a [SPEM](#) que demuestra su relevancia en los modelos de procesos de software.

A. Burattin et al. [72] ilustran un método para validar la consistencia del flujo en procesos de negocio. Este trabajo expone diferentes métricas de procesos que están relacionadas al entendimiento y distribución de la visualización de elementos de procesos de negocio. Para la validación de la propuesta (descrita en la siguiente sección), se basan en el uso de las métricas M-E1 y M-E2 y M-BP propuestas como coeficientes de entendimiento de modelos de procesos y calculadas dependiendo de la dirección y ángulo de las aristas en los procesos de negocio. Así mismo, estas métricas fueron contrastadas con un promedio de evaluación por parte de expertos.

El trabajo de K. Kluza et al. [50] contiene métricas que modelos de negocio aplicables a modelos especificados en [SPEM](#). Los modelos de correlación se aplicaron a un conjunto de métricas seleccionadas de modelos de proceso de negocio. También indican que la correlación de métricas es un componente crucial en la toma de decisiones de los procesos debido a la poca información que pueden ofrecer las métricas en sí mismas o a la misma dificultad para interpretarlas.

J. Pavlicek et al. [46], similar al trabajo de K. Kluza [73], ilustran una serie de métricas aplicables a modelos de negocio para medir su complejidad y profundidad en diferentes lenguajes de especificación de modelos de procesos como BPMN, Redes de Petri, Cadena de Procesos dirigida a Eventos (EPC), entre otros. Algunas de las métricas aportadas están orientadas principalmente a la construcción de grafos que a los modelos de proceso de un lenguaje de especificación en particular.

A. del-Ri3-Ortega et al. [51] proponen una notaci3n gr3fica para indicadores de rendimiento de procesos que proveen informaci3n esencial para la toma de decisiones. Las métricas usadas en el trabajo incluye el análisis de los modelos de proceso por medio del número de actividades y número de procesos, 2 métricas que pueden ser aplicadas a los modelos de proceso de [SPEM 2.0](#).

J. Geisel et al. [52] realizan una propuesta que consiste en una herramienta de soporte para procesos orientados a la seguridad elaborados en [SPEM 2.0](#). La propuesta involucra diversas métricas orientadas a la seguridad como tamaño del proceso definido como el número de nodos y/o arcos, el diámetro, la profundidad y otras medidas.

El trabajo de G. Kaur et al. [54] ilustran un resumen de métricas aplicables a procesos ágiles, así mismo expande el ámbito de su trabajo a métricas de hardware, de producto software, de proyecto y de contexto del proceso como disponibilidad, tolerancia a fallos entre otros. Así mismo este trabajo también incluye recomendaciones

sobre en qué parte del ciclo de vida del software pueden ser aplicadas.

### 2.5.1.2. Evaluación de Métricas de Modelos de Procesos Software (C2)

El trabajo de A. Buratting et al. [72], mencionado en la anterior sección, ilustran también un modelo de evaluación para detectar los factores por los cuales se ve afectado el entendimiento de los modelos de proceso de negocio, principalmente se ve afectado por elementos visuales como la longitud y estilo de las aristas entre tareas, la distribución de las tareas en el plano, los ángulos de los conectores, la dirección (consistencia) del flujo entre otros. Los resultados indican que la distribución de los elementos de proceso entre las métricas propuestas y las calificaciones por parte de los expertos son similares. La correlación indica que el método puede ser incluido como propuesta para la implementación de elementos de evaluación de modelos de proceso elaborados bajo el lenguaje **SPEM**.

El trabajo de K. Kluza et al. [50] citado anteriormente, propone un modelo de evaluación de las métricas presentadas para conocer el nivel de correlación entre las mismas por medio de 2 métodos: el coeficiente de correlación de rango de Kendall tau y de Spearman. Los resultados indican que los métodos para la extracción del coeficiente de correlación arrojan resultados similares por lo que los análisis de los resultados aumentan en el nivel de confiabilidad entre las medidas presentadas de la correlación de métricas similares.

El trabajo de K. Figl [75] contiene una revisión sistemática de los modelos de proceso de negocio que favorecen la comprensión de procesos, principalmente durante su ejecución. Los resultados ilustran el uso de diversos métodos de comparación de aproximaciones de las soluciones encontradas como el uso de ANOVA, análisis de regresión, factor de correlación de Spearman y Pearson. Así mismo, diversos lenguajes de especificación de modelos de proceso como BPMN y sus derivados como EPC, vBPMN, YAWL, entre otros.

J. Pavlicek [46] involucra dentro de su modelo de evaluación un conjunto de directivas para la medición del entendimiento, claridad y orientación de modelos de proceso. La evaluación de estas 3 dimensiones han sido realizadas de forma cualitativa bajo opinión de expertos a través del método de prueba de usabilidad por pares. Una de las conclusiones del estudio determinó que es importante conocer el usuario final del proceso pues el contexto es importante para una óptima ejecución del mismo.

El trabajo J. Pflug et al. [74] incluyen un modelo para la evaluación de la similitud de modelos de proceso de negocio usando **precisión** y **recuerdo**. La evaluación se realiza sobre la similitud entre elementos de procesos de software para lo cual se propusieron diversas métricas tanto para instancia de proceso como para su ejecución basadas de lógica de conjuntos que fueron evaluadas entre sí. Finalmente el trabajo evalúa la correcta asignación de los atributos de proceso basándose en las métricas de similitud por medio de la **precisión** y el **recuerdo**.

El trabajo de M. Smatti et al. [53] propone una serie de evaluaciones para la detección y corrección de desviaciones en la ejecución de modelos de proceso de ma-

nera cuantitativa. El trabajo está orientado a hallar los subprocesos que más tienen desviación de la ejecución de proceso y como aplicar unas reglas de transformación propuestas por ellos mismos para la corrección de los modelos de proceso por medio de un patrón de adaptación que indica cómo detectar y aplicar la corrección deseada al modelo.

### 2.5.2. Otros trabajos relacionados

#### 2.5.2.1. Métricas de Procesos Software

El trabajo de Canfora et. al [6] proponen una serie de métricas para modelos de proceso que son aplicables en el marco de [SPEM 2.0](#). Así mismo realizan una familia de experimentos para validar las métricas propuestas. Así mismo utilizan la notación [SPEM 2.0](#). El trabajo también ilustra que las métricas propuestas son fácilmente replicables a otros meta-lenguajes para especificación de procesos. Este trabajo también propone la inclusión de métricas relacionadas con la complejidad como parte de las propiedades de la estructura de los modelos de proceso. Este trabajo también ilustra una comparativa de los atributos de calidad de los productos software del Software Engineering Institute (Carnegie Mellon) ([SEI](#)) a la luz de los modelos de proceso, específicamente con [SPEM](#). Respecto al proceso de experimentación de las métricas, se ha realizado una evaluación de 18 modelos de proceso de software, los sujetos son estudiantes y profesionales en el área de ingeniería de software evaluando los atributos **entendibilidad**, **modificabilidad** y **analizabilidad** de forma subjetiva (2 experimentos) y de forma objetiva (3 experimentos). La validación de las métricas ilustra que la métrica NCA (acoplamiento de actividades en el modelo de proceso, que se define como la relación entre el número de actividades sobre el número de dependencias de precedencia entre actividades) no tiene impacto en los experimentos con análisis subjetivo, sin embargo, los resultados de los experimentos objetivos reflejan una correlación de la presente métrica con el tiempo de entendimiento en 2 de los 3 experimentos con análisis objetivo.

El trabajo anterior es relevante debido a la inclusión en [AVISPA](#) de la métrica NCA, sin embargo, respecto a las métricas relacionadas con la complejidad, al no pasar el filtro propuesto dentro de la revisión sistemática (ver figura 1.3) no ha sido incluido en [AVISPA](#) como parte de sus visualizaciones.

El trabajo de J. Hurtado et al. [16] ilustran la primera aproximación de la herramienta de visualización de modelos de proceso [AVISPA](#). Los investigadores ilustran los primeros patrones de error que agrupan elementos de modelos de proceso de manera global tomando todo el modelo de proceso realizado aplicando la solución en diversos modelos de proceso de software como el Proceso Unificado libre de la Fundación Eclipse ([OpenUP](#)) y en modelos de proceso reales de Chile como Tutelkan-TRP. Así mismo, se ilustran las características básicas, entre ellas algunas métricas y algo de la codificación realizada para la creación de las visualizaciones como número de tareas, de roles, de productos de trabajo, tanto de entrada como de salida para cada una de ellas, entre otras.

### 2.5.2.2. Evaluación de métricas de procesos de software

El trabajo de M. Bastarrica et al. [41] expone un marco de trabajo para la evaluación indirecta de la confiabilidad de los datos presentados por AVISPA al momento de evaluar el impacto de la rigurosidad de la documentación de los modelos de proceso. Para ello se realizaron una serie de hipótesis que direccionaron la ejecución de modelo de procesos bajo hechos supuestos orientados a la determinación de su efecto en el tiempo de ejecución de los mismos y su relación con la configuración / documentación de los elementos de proceso. Para la evaluación de cada una de las hipótesis, los evaluadores se apoyaron en la herramienta de visualizaciones AVISPA para examinar la composición de los procesos elaborados en diferentes industrias de Chile. El trabajo indica que algunas de las hipótesis no pueden determinarse o son falsas respecto a la correlación de hechos de construcción del modelo con tiempo de atraso de la ejecución del proceso. Lo anterior sugiere que **algunos planos generados por AVISPA no ofrecen información suficiente para predecir un posible tiempo de atraso del proceso previo a su ejecución**. Este trabajo es de relevancia dado que indica una brecha relevante en la elaboración de planos generados por AVISPA, pues revela un nivel de utilidad implícito de los planos producidos por la herramienta.

### 2.5.2.3. Visualización de modelos de proceso

El trabajo de M. Reichert [67] aborda el problema de la sobrecarga visual de los elementos de proceso y su comprensión. Principalmente los modelos de proceso de negocio, por lo general, contienen gran cantidad de elementos de proceso que no involucran adecuadamente elementos de distribución de la información para facilitar la comprensión por parte de usuarios del mismo. Este trabajo ilustra ejemplos de modelos de proceso de negocio de gran tamaño y realiza una discusión sobre algunos de los retos de la representación y abstracción de los mismos.

El trabajo de R. Bobrik et al. [76] también es usado en el modelo de validación de M. Reichert [67] e ilustra una herramienta de visualización de modelos de procesos de software facilitando su modificación. La herramienta Proviado, provee un modo de visualización flexible y adaptable de modelos de proceso de negocio que pueden ser agrupados en diversos sistemas de información.

El trabajo de M. Reichert et al. [77] ilustran una manera de visualización de modelos de proceso de gran tamaño por medio de vistas parametrizables. El concepto de vista permite la visualización más adecuada y focalizada de los modelos de proceso y sus elementos de interés, para lo cual se implementan las visualizaciones de subprocesos por medio de lógica de conjuntos. Los autores del trabajo también se apoyan bajo la herramienta Proviado, que permite definir composiciones flexibles de las vistas para realizar abstracciones de subprocesos más sofisticadas. El trabajo aporta reglas de abstracción para la generar vistas de subprocesos de modelos de proceso de negocio en 5 diversas categorías.

Los anteriores trabajos presentan diversas herramientas conceptuales para la implementación de las visualizaciones que AVISPA genera. Como ha sido mencionado

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

anteriormente, las visualizaciones se han realizado conforme a una sola categoría de elementos de proceso por lo que las visualizaciones no poseen el adecuado nivel de claridad para los encargados del proceso. La abstracción adecuada de categorías es crucial para lograr un mayor nivel de comprensibilidad.

## Capítulo 3

# Análisis y Visualizaciones de Modelos de Proceso Software con AVISPA

### 3.1. Introducción

Los modelos de proceso son construcciones complejas orientadas a la organización y administración de la construcción de software. Muchos de los modelos de proceso demandan una gran cantidad de esfuerzo y, una vez especificados, resultan cada vez menos herramientas y aproximaciones para analizar su calidad.

Los planos, en el ámbito de AVISPA, también conocidos como “Blueprints”, son construcciones arquitectónicas de los modelos de proceso de software que hacen uso de métricas calculadas a partir de la información de los mismos y cuyo fin es el entendimiento y evaluación de aspectos que no están directamente disponibles en SPEM 2.0 y sus herramientas asociadas. Así, usando ésta herramienta, el ingeniero de procesos únicamente necesita analizar los elementos que están señalados por medio de los patrones de error que resaltan elementos de posible importancia en los planos producidos.

Existen diversos métodos de validación y verificación (V & V) de calidad de modelos de proceso de software. Uno de ellos es el método de análisis visual que consiste en la evaluación de elementos de proceso por medio de visualizaciones. Una aproximación de V & V de análisis visual ha sido la construcción de blueprints [16] los cuales ofrecen diferentes perspectivas de modelos de proceso elaborados en SPEM 2.0 utilizando los datos cuantitativos recolectados del mismo.

Los planos han demostrado utilidad en la representación de datos y visualizaciones holísticas de los elementos de un modelo de proceso de software. También han demostrado utilidad en la evaluación de modelos de proceso software en la industria de software y a ingenieros de proceso que deseen realizar análisis de estos modelos con la información obtenida en estos planos visuales.

Algunos de éstos planos cuentan con patrones de error. Estos patrones de error son indicadores visuales dentro de los blueprints que resaltan posibles errores recurrentes en la construcción y ejecución de los modelos de proceso de software, permitiendo un menor tiempo en la detección de errores reales por ingenieros de proceso de diferente nivel de experiencia.

Para asistir la creación de estos planos, la herramienta **AVISPA** ha sido desarrollada. **AVISPA** es una herramienta de renderización gráfica de los elementos de proceso de software que genera los blueprints con sus respectivos patrones de error.

Una de las funciones de **AVISPA** es clasificar los elementos de proceso de software con el fin de resaltar patrones de error: los que pertenecen al patrón de error y los que no.

**AVISPA** es una herramienta que está construida sobre la plataforma de visualización y análisis de datos Moose<sup>1</sup> y el sistema Mondrian de visualización y análisis de datos, y es de libre acceso en diversos repositorios<sup>2</sup>.

### 3.2. Planos / Patrones de Error de Modelos de Proceso

Una construcción bien definida de los modelos de proceso de software permite lograr una sistematización del desarrollo de software. Para ello, esfuerzos entre la industria y la academia han sido realizados en orden de lograr la madurez y robustez de los modelos y estándares de calidad. La capacidad humana para la interpretación componentes visuales es inherente y necesaria con el fin de llegar a una interpretación coherente de la estructura del proceso, su contexto entre otras características.

Tomando como base que los procesos de software son software también [42], y que múltiples vistas de una arquitectura de software permiten obtener diferente tipo de información para la evaluación de diferentes actores interesados en el sistema [47], las diferentes vistas arquitectónicas de un proceso software sirven para evaluar diferentes partes y/o características del mismo. Además, contienen diferente información específica para la evaluación por parte de los interesados.

Los planos de modelos de proceso han permitido identificar diversos problemas en procesos de diferentes compañías y los interesados concuerdan en que los problemas identificados por la herramienta han sido problemas que, en efecto, se encontraban presentes [3, 69–71].

Los blueprints han sido de utilidad para detectar diversos problemas en la construcción de los procesos así como antes de su ejecución, sin embargo, al ser un clasificador, el proceso de elaboración de planos de **AVISPA** puede presentar errores en la representación de los elementos de proceso.

Por otra parte, las métricas utilizadas en la representación de los blueprints no han pasado por un respectivo filtro de validación y verificación, por lo que ha sido

---

<sup>1</sup> Disponible en <http://www.moosetechnology.org/>

<sup>2</sup> Disponible en <https://github.com/jjalvarez/AVISPA> y en <http://smalltalkhub.com/#!/~abergel/Avisp>

necesaria una búsqueda detallada de las mismas (ver sec. 1.4 y cap. 2) con el objetivo de implementar las más adecuadas y maduras para las visualizaciones.

Teniendo en cuenta que por medio de la agrupación de elementos visuales disminuye la sobrecarga visual y por lo tanto cognitiva [48], los blueprints, tanto los existentes en las anteriores versiones de AVISPA como los nuevos planos resultantes de la investigación, cuentan con vistas de agrupación de elementos de proceso en sí mismos.

La versión previa de AVISPA ha sido mejorada con diversos elementos que han aportado positivamente en la comprensibilidad y madurez de los planos producidos por la herramienta. Las siguientes secciones y capítulos de este trabajo ilustran las mejoras de los blueprints en detalle y procedimientos llevados a cabo para realizar mejoras y depuración de AVISPA.

### 3.3. Proceso de elaboración de planos de AVISPA

AVISPA cuenta con dos tipos de visualizaciones de modelos de proceso de software, los planos y los patrones de error.

Los planos son visualizaciones que ofrecen información de la construcción del proyecto, nos indica la construcción de los elementos de un proceso.

Los planos a su vez pueden ir acompañados de patrones de error que ilustran posibles errores recurrentes probados por experiencias previas en la industria durante la ejecución de un modelo de proceso de software.

La figura 3.1 ilustra cada uno de los pasos realizados para llevar a cabo la implementación de planos y patrones de error:

1. **Abrir proceso en AVISPA:** En la visualización de los planos, ésta tarea consiste en la interacción del usuario con el sistema para incorporar el modelo de proceso exportado en EPF en la herramienta AVISPA.
2. **Extracción de elementos de proceso:** Esta tarea consiste en el mapeo de los elementos de proceso junto con sus conexiones / dependencias del proceso XML exportado de EPF en objetos propios reconocibles dentro de Moose en base a las clases que los recolectan en AVISPA.
3. **Extracción de elementos de proceso:** Esta tarea consiste en el mapeo de los elementos de proceso junto con sus conexiones / dependencias del proceso XML exportado de EPF en objetos propios reconocibles dentro de Moose en base a las clases que los recolectan en AVISPA.
4. **Cálculo de métricas a partir de los elementos de proceso:** La información otorgada por los elementos de proceso extraídos es utilizada para realizar la generación de valores a partir de métricas establecidas y filtradas en el presente estudio.

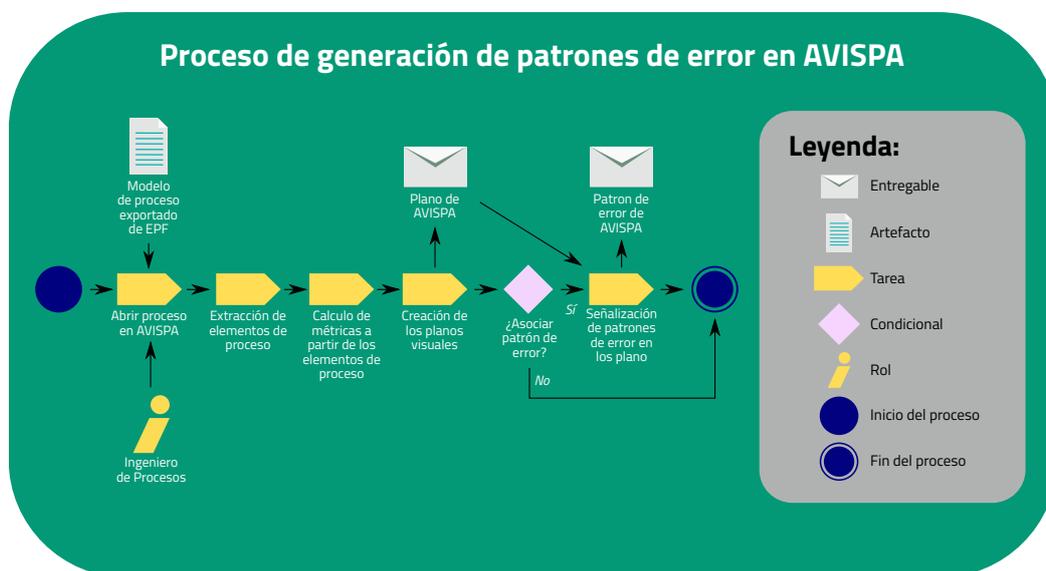


Figura 3.1: Generación de planos y patrones de error AVISPA

5. **Creación de los planos visuales:** Una vez los elementos estén cargados y calculados, AVISPA puede tener la posibilidad de generar los planos respectivos. Por lo general los patrones de error finalmente son incorporados a los planos, puesto que sin los patrones, los planos carecerían de información valiosa para la toma de decisiones.
6. **Señalización de los patrones de error en los planos:** Finalmente, los patrones de error, se señalan dentro del plano y la leyenda respectiva del patrón de error es generada.

### 3.4. La herramienta AVISPA

#### 3.4.1. Arquitectura

Moose es una plataforma de código abierto para soporte de análisis de datos. La herramienta es una bifurcación de la plataforma Pharo cuya construcción ha sido llevada a cabo en el lenguaje de programación Smalltalk. Moose utiliza diferentes librerías para la visualización de datos donde AVISPA hace uso de la librería Mondrian para la representación de sus planos. La figura 3.2 ilustra la pila de tecnologías utilizadas para la implementación de AVISPA que serán descritas a continuación:

- **Moose:** Es una plataforma para desarrollo de software y análisis de datos, está basado en Pharo y provee diversas herramientas y utilidades como edición y depuración de código en tiempo de ejecución y total orientación a objetos gracias al lenguaje Smalltalk. AVISPA realiza el cálculo de las métricas de sus planos en Pharo.

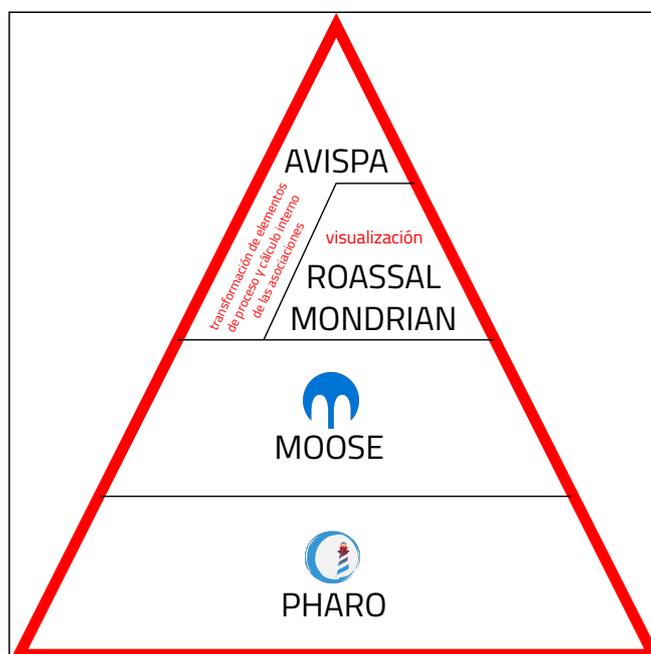


Figura 3.2: Arquitectura tecnológica de implementación de AVISPA

- **Mondrian:** Es una librería que hace parte del conjunto de extensiones de Pharo, utilizada para el diseño gráfico de los planos de AVISPA.
- **Monticello:** Es un explorador de repositorios de Pharo compatible con Small-talkHub cuya función consiste en ser un sistema de gestión de versionado el cual posee múltiples opciones de exportación del código de los proyectos.

La versión inicial de AVISPA ha sido descrita en otros trabajos que ofrecen una perspectiva más amplia de su contenido [1, 5, 16, 69, 70]. Por tal motivo las especificaciones de los planos realizados de dicha versión no serán detallados en el presente trabajo.

### 3.4.2. Modificaciones respecto a la anterior versión de AVISPA

Se han realizado mejoras respecto a las anteriores versiones de AVISPA en relación a las visualizaciones, indicadores de los planos y ayudas visuales, a continuación se enunciarán las diferencias:

#### 3.4.2.1. Modificaciones generales en la visualización y organización de los planos

En algunos de los planos de AVISPA existen algunas mejoras visuales en la representación de los planos, esto es:

- **Información de un elemento:** cuando se coloca el puntero sobre un elemento de un plano se puede obtener información extra. Por ejemplo, en un plano de productos de trabajo, al colocar el puntero en un vértice del nodo (producto de trabajo) se resalta de un color distinto los nodos de los cuales el producto de trabajo depende.
- **Leyendas:** Las leyendas indican como interpretar el plano o patrón de error presentado, indican el significado de las representaciones en cada plano y las restricciones en las visualizaciones que puedan presentarse.
- **División de los planos por composición:** Dada la naturaleza de interpretación de los elementos de proceso por composición por encima de su visualización holística, la herramienta provee una visualización de la composición de los elementos de proceso de uno en particular. Así por ejemplo, si se visualiza una fase, se visualizan todos los elementos de proceso por tipo, bien sea iteraciones, actividades, tareas y demás, de dicha fase. Aquí también se aplican los planos y patrones de error acotados a los elementos de proceso que contiene.
- **Adición de nuevos tipos de elementos de proceso:** Esencialmente estos nuevos elementos de proceso permiten una visualización más organizada de la construcción de un modelo de proceso software. Al ser una representación por composición, se añaden los procesos de entrega (Delivery Process en [SPEM](#)) como el elemento base de la composición, pues de este tipo de elemento de proceso se procede a visualizar las fases, de las fases las iteraciones y así hasta el nivel más atómico de las visualizaciones (tareas, roles, productos de trabajo entre otros modelos atómicos).
- **Resolución de variabilidad de modelos de proceso en [SPEM](#):** Los mecanismos de variabilidad hacen parte del conjunto de elementos de metamodelo de [SPEM 2.0](#) y que soporta [EPF](#). La versión actual de [AVISPA](#) soporta dicha resolución bajo las reglas propuestas por la OMG. El trabajo de S. Salgado [69] ilustra a profundidad las modificaciones llevadas a cabo para realizar la resolución de variabilidad con éxito, incrementando la confiabilidad de las visualizaciones.

#### 3.4.2.2. Respecto a las conclusiones y datos hallados de M. Bastarrica et al. [41]

Los patrones de error, al ser representaciones de posibles errores en la ejecución del proceso, dependen en gran medida de la información de las experiencias de las ejecuciones de los modelos de proceso de software en orden de brindar información más exacta de lo que es error y de lo que no lo es antes de la ejecución del proceso.

Según el trabajo de M. Bastarrica et al. [41] se realizaron seis hipótesis basadas en los patrones de error de [AVISPA](#) sobre diversos proyectos usando un modelo de proceso de software, donde cada una demuestra o refuta la utilidad de los modelos generados en [AVISPA](#).

Hipótesis	Patrones de Error
H1	Productos de Trabajo sin Roles
H2	Productos de Trabajo Demandados
H3	N.A.
H4	Rol Sobrecargado
H5	Sin Guía Asociada
H6	Productos de Trabajo Desperdiciados

**Tabla 3.1:** Correlación de hipótesis con los patrones de error

Las hipótesis determinan la veracidad de los patrones de error de **AVISPA** en el retraso de la entrega de un proyecto tomando como referencia los artefactos producidos y sus tiempos de entrega tres estados: entregado, retrasado y no entregado. Las hipótesis son:

- Ningún rol es responsable del producto de trabajo (H1).
- Dos o más roles son responsables del producto de trabajo (H2).
- El producto de trabajo está asignado a demasiados roles (H3).
- Un rol sobrecargado es responsable del producto de trabajo (H4).
- No hay plantillas disponibles para el producto de trabajo (H5).
- Producto de trabajo potencialmente desperdiciado (H6).

La tabla 3.1 correlaciona cada una de las hipótesis del estudio con cada uno de los patrones de error de **AVISPA** del trabajo de J. Hurtado et al. [16].

Los resultados de éste trabajo ilustran que existen diversos patrones de error en cuyas hipótesis no se pueden establecer decisiones sobre el nivel de pertinencia de las visualizaciones de modelos de proceso de **AVISPA**. Sin embargo, dada la evidencia en relación con el nivel de pertinencia de estos resultados en la exactitud de las visualizaciones generadas por la herramienta, se definieron modificaciones de acuerdo a los resultados del estudio (ver tabla 3.2).

### 3.4.2.3. Mejoras específicas en la visualización y organización de los planos

Tras la serie de mejoras establecidas tras el análisis del trabajo de M. Bastarrica et al. [41] se han establecido una serie de cambios específicos en determinados planos y patrones de error con el objetivo de incrementar la exactitud e información de los datos presentados en los planos de **AVISPA**:

- **Patrón de rol sobrecargado:** En el patrón de error de rol sobrecargado únicamente se indicaban los roles que tienen muchos artefactos por implementar.

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Hipótesis	Trabajo M. Bastarrica et al.	Edición en AVISPA
H1 y H2	El estudio demuestra que las hipótesis pocas veces representan retrasos en la entrega de los productos de trabajo en el estudio debido la poca frecuencia de con la que aparecen (producto de la buena documentación del proceso en este sentido) y a la baja tasa de casos donde las hipótesis son verdaderas.	∅
H3	Esta hipótesis indica que muchos roles asignados a un producto de trabajo es un factor relevante para el tiempo de entrega de los productos. Sin embargo, se especifica que, para considerar que un producto de trabajo esté sobrecargado, el umbral consiste en 4 o más roles asociados a dicho artefacto.	<input checked="" type="checkbox"/> Se crea el patrón de error.
H4	Indica que un rol sobrecargado responsable de un producto de trabajo puede ser responsable de retrasos en su entrega.	<input checked="" type="checkbox"/> Se edita el plano de rol sobrecargado para que también indique los roles que tienen uno o más artefactos de responsabilidad (artefactos de salida de tareas que son los que requieren un esfuerzo al realizar / actualizar).
H5	No existe soporte de ésta hipótesis, los casos donde los productos de trabajo no tienen plantilla son numerosos, pero el retraso en su entrega se cumple solo en uno	∅
H6	Esta hipótesis tiene una relación débil con el retraso de proyectos debido principalmente a que los planos no arrojan ninguna diferencia entre un artefacto y un entregable. Así se señalan como desperdicio todos los productos de trabajo	<input checked="" type="checkbox"/> Se excluyen de este patrón de error los entregables.

Los iconos en la casilla AVISPA son modificaciones de los patrones de error relacionados con sus respectivas hipótesis:

- Aprobación / creación, donde se someterá a la siguiente fase (ver capítulo 4).
- Modificación, en cuyo caso se especifica la edición a realizar para aprobar.
- Eliminación.

**Tabla 3.2:** Resultados de correlación de las hipótesis del trabajo de M. Bastarrica et al. [41]

En esta nueva versión, aparte de indicar los roles que están sobrecargados, también señala los roles bajo ésta misma condición y que además sean responsables de algún producto de trabajo.

- **Patrón de productos de trabajo desperdiciados:** Este plano se modifica, dado que un entregable en efecto no es un producto de trabajo desperdiciado así se generen para no ser consumidos en ninguna otra parte del flujo del proceso, por lo cual los entregables se excluyen del patrón de error. Así, el patrón señala exclusivamente los artefactos que no son entregables y que son desperdiciados o generados sin una posible razón.
- **Patrones productos de trabajo sin plantilla y productos de trabajo sin roles y Productos de trabajo demandados:** Al no tener el suficiente soporte empírico, los productos de trabajo sin plantilla no entran en los planos de detección de errores de [AVISPA](#).

Así mismo, diferentes esfuerzos realizados en conjunto con el presente trabajo han incluido diversas modificaciones sobre la antigua versión de [AVISPA](#):

- **Correcciones sobre resolución de la variabilidad en tareas [69]:** [SPEM 2.0](#) posee mecanismos de variabilidad de tareas, roles y artefactos de los modelos de proceso (ver sección 2.4) los cuales [AVISPA](#) no consideraba en su versión anterior, excluyendo modelos de proceso con resolución de variabilidad tales como [OpenUP](#). La nueva versión de [AVISPA](#) contempla la resolución de variabilidad de las tareas, disminuyendo los errores de la representación de los elementos de modelos de proceso en los planos de [AVISPA](#). El aporte de éste trabajo incluye la corrección de las reglas de resolución de variabilidad, la corrección de las asociaciones de *contribución* entre otros.
- **Soporte de resolución de variabilidad en roles y productos de trabajo:** El soporte parcial de la variabilidad de [SPEM 2.0](#) [69] no amplía su funcionalidad hacia la resolución de variabilidad de roles y productos de trabajo. Este trabajo soporta estos dos tipos de elementos con verificación de reglas con ejemplos sencillos ilustrados en la sección 4.5.

### 3.4.3. Modificaciones propuestas fuera del ámbito del presente trabajo

Algunas modificaciones adicionales han sido realizadas en un intento de mejorar el uso de la herramienta así como propuestas de métricas, más exactamente, tomando en cuenta el trabajo de Canfora et al. [6]:

- **Vistas por composición de elementos de proceso:** Una de las características de [SPEM](#) es que cuenta con la posibilidad de documentar elementos de procesos dentro de si mismos. La nueva versión de [AVISPA](#) cuenta con ésta funcionalidad, permitiendo navegar entre elementos de proceso que contienen a su vez otros.

- **Inicio de navegación por medio del “Delivery Process”**: Los modelos de proceso bien documentados por lo general utilizan los procesos de entrega (delivery process). Como parte del punto inmediatamente anterior. Bajo el mismo concepto composicional, los modelos de proceso que inician con los procesos de entrega, tienen acceso a visualizaciones de todo el modelo de proceso empezando por los elementos mas grandes (procesos de entrega, fases, iteraciones, ...) hasta los elementos de proceso más atómicos (... , actividades, tareas, roles y productos de trabajo).
- **Planos de Factor de Acoplamiento**: El artículo de Canfora et al. [6] propone una métrica que consiste en el factor de acoplamiento de actividades realizando calculos sobre las mismas subactividades contenidas. Sin embargo, en SPEM 2.0, el nivel de acoplamiento de actividades puede desembocar finalmente en tareas como elementos atómicos del proceso, por lo se incluyen dos nuevos planos que permiten indicar el nivel de acoplamiento en elementos de proceso que pueden contener a su vez otros elementos de proceso, estos dos planos: son **Factor de acoplamiento por actividades** y **Factor de acoplamiento por tareas**
- **Plano de responsabilidad de roles sobre tareas**: El trabajo de M. Bastarrica et al. [41] indica que, bajo el soporte empírico realizado respecto a los roles en relación con los artefactos, si existen más de 4 roles que producen o modifican artefactos, estos pueden representar posibles cuellos de botella. El plano de roles responsables de productos de trabajo indican diversos casos entre roles responsables de artefactos del modelo de proceso y roles que generan dichos artefactos.

Cada una de estas modificaciones es explicada a continuación

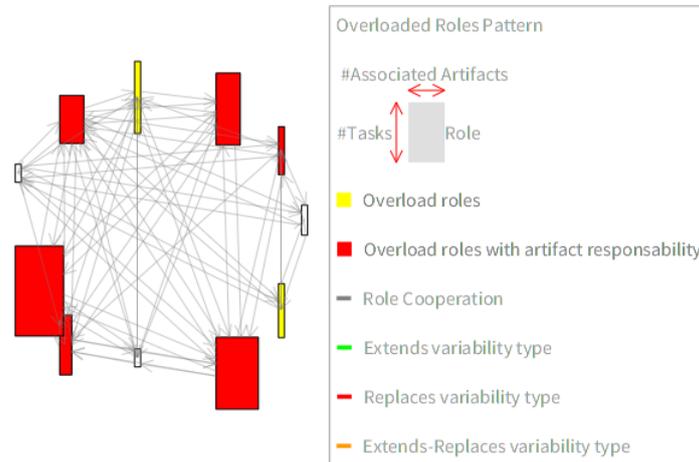
#### 3.4.4. Descripción de las modificaciones

Cada uno de los códigos de AVISPA se encuentra documentado en el apéndice A. A continuación se describen las modificaciones realizadas bajo el soporte empírico de M. Bastarrica et al. [41].

##### 3.4.4.1. Patrón de rol sobrecargado

Este patrón fue modificado debido a que la evidencia empírica de M. Bastarrica et al. [41] ilustra que los roles que están sobrecargados y los que están sobrecargados y además, tienen artefactos de responsabilidad a cargo pueden realizar una diferencia en la ejecución de los modelos de proceso de dicho trabajo. Este patrón de error ha sido modificado para diferenciar entre estas dos clases de roles.

La figura 3.3 ilustra el patrón de roles sobrecargados del modelo de proceso OpenUP. Los roles sobrecargados se ilustran en amarillo, mientras que los roles



**Figura 3.3:** Patrón de roles sobrecargados de la nueva versión de AVISPA

sobrecargados responsables de artefactos están ilustrados en rojo. Las líneas que interconectan los roles son cooperaciones entre ellos.

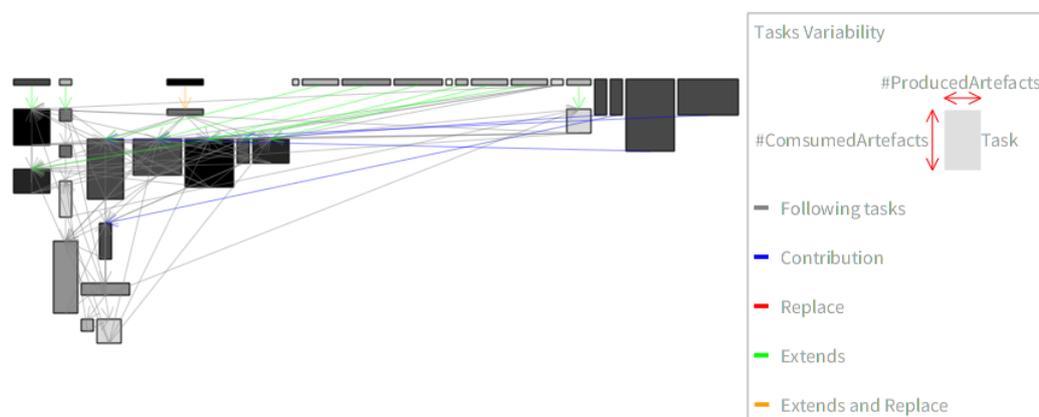
El código respectivo del plano ha sido ilustrado en la figura A.1.

### 3.4.4.2. Resolución de la variabilidad sobre tareas

En la versión anterior de AVISPA, los planos de tareas no tomaban en cuenta la resolución de variabilidad presente en las tareas de los modelos de proceso de software, por lo que se visualizaban las tareas sin ninguna clase de modificación al momento de la representación sin tomar en cuenta los tipos de resolución de variabilidad. Posteriormente se han realizado esfuerzos para definir una versión que soportaba resolución de variabilidad según SPEM 2.0, sobre el Blueprint de tareas, sin embargo, tenía errores en la resolución de variabilidad de tipo “contribución”. La nueva versión de AVISPA soporta esta funcionalidad en los planos de tareas y roles, mejorando la resolución de variabilidad en las tareas, así como las visualizaciones de los planos y patrones de este tipo de elementos de proceso de software. La figure 3.4 ilustra la implementación de esta funcionalidad.

Uno de los primeros avances en la resolución de variabilidad se encuentran en el trabajo de J. Salgado et al. [69]. Sin embargo, realizando un análisis de los resultados y procedimientos del estudio, se encontraron diversas falencias que, como resultado final, producen relaciones y, por lo tanto, visualizaciones, que no corresponden con la realidad de las reglas del informe técnico de SPEM 2.0 [21]. Entre éstas se incluyen:

- **Resolución de la contribución:** Este tipo de relación no se estaba llevando a cabo adecuadamente en el plano de variabilidad de tareas. Una depuración del sistema AVISPA revela que la relación estaba llevándose a cabo, sin embargo, no se visualizaba en el plano
- **Resolución de la relación *reemplazo* y *extensión-reemplazo*:** La regla ignorada



**Figura 3.4:** Patrón de tareas con variabilidad AVISPA

consiste en no resolver estas relaciones en caso de que existan 2 o más relaciones del mismo tipo apuntando al mismo elemento base **EB**. Para lo cual se añade el campo respectivo en el elemento correspondiente para obtener dichas relaciones.

El presente trabajo aporta a la dinámica de resolución de la variabilidad resolviendo estas falencias previamente encontradas.

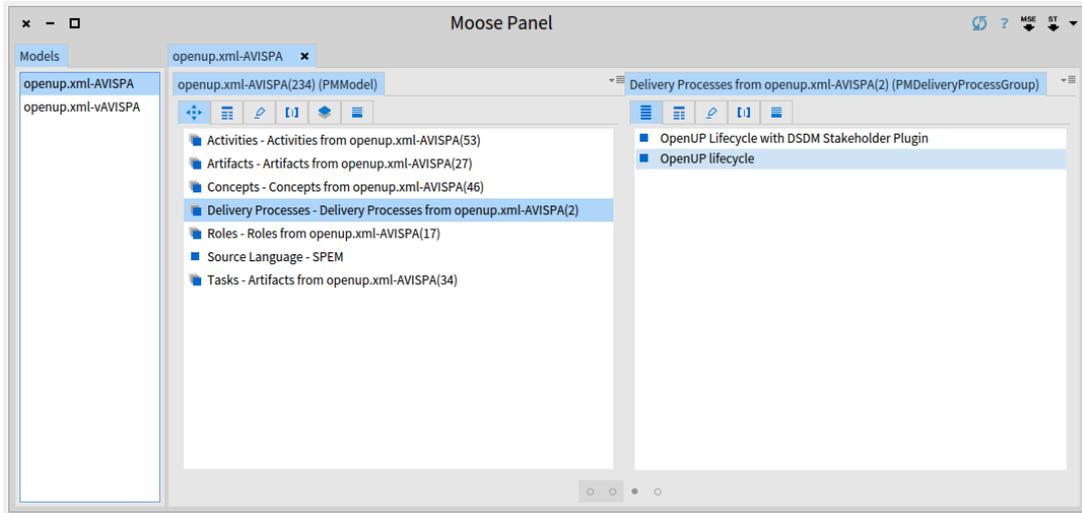
#### 3.4.4.3. Inicio de navegación por medio de los procesos de entrega

Los procesos de entrega en SPEM 2.0 tienen la funcionalidad de empaquetar la publicación de modelos de proceso que extienden su funcionalidad para múltiples proyectos de software. Así, los procesos de entrega, son procesos que cubren la ejecución del ciclo de vida de software de inicio a fin.

Teniendo en cuenta que los procesos de entrega ofrecen una visualización más “natural” y, por lo tanto, orientada al negocio de los procesos, AVISPA cuenta con la posibilidad de una navegación más orientada hacia el entendimiento del proceso por medio de la inclusión de los procesos de entrega, pues ofrecen una vista de los elementos de proceso desde el principio de la ejecución. Así, el usuario de AVISPA aparte de visualizar los patrones en cualquier parte o subconjunto del proceso, posibilita la ubicación del usuario en la ejecución del proceso ofreciendo una idea más exacta del entendimiento tanto de los modelos de proceso y su ejecución como de los patrones.

#### 3.4.4.4. Factor de acoplamiento de actividades por otras actividades contenedoras y por tareas

El factor de acoplamiento que tiene cada actividad puede ofrecer una mayor posibilidad de entendimiento tanto del modelo de proceso como de la dificultad que puede conllevar cambiar los elementos de proceso al interior de una actividad.



**Figura 3.5:** Navegación por medio de procesos de entrega AVISPA

El factor de acoplamiento ha sido validada en el trabajo de Canfora et al. [6] por lo que se propone como un patrón dentro de AVISPA.

El factor de acoplamiento por actividades (NCA), en el trabajo de Canfora et al. [6] es definido por la ecuación 3.1.

$$NCA = \frac{MP(\sum_{i=1}^n A_i)}{MP(\sum_{i=1}^n P(A_i))} \quad (3.1)$$

Donde:

$MP$  = Modelo de proceso

$MP(\sum_{i=1}^n A_i)$  = Total de actividades del modelo de proceso

$MP(\sum_{i=1}^n P(A_i))$  = Total de precedencias entre actividades del modelo de proceso

Sin embargo, la ecuación 3.1 tiene dos limitantes que pueden dificultar la interpretación de la métrica y su respectiva implementación en AVISPA: (i) la métrica NCA está definida como factor inversamente proporcional al nivel de acoplamiento del modelo de proceso y (ii), está definida para todas las actividades de todo el modelo de proceso y AVISPA interpreta el factor por actividad individual. Bajo estas limitantes, para obtener una representación más adecuada dentro de los patrones de error de

AVISPA, se ha realizado la ecuación 3.2 que define el factor de acoplamiento de una actividad individual por sus actividades contenidas conservando la intención de la fórmula 3.1 e indicando el factor de cada actividad individual representada por AVISPA, además de reescribir la fórmula de manera directamente proporcional al nivel de acoplamiento de las actividades.

$$NCA_{A_i} = \frac{A(\sum_{j=1}^m P(A_j))_i}{A(\sum_{j=1}^m A_j)_i} \quad (3.2)$$

Donde:

$NCA_{A_i}$  = Factor de acoplamiento de actividades de la actividad  $i$

$A(\sum_{k=1}^l P(A_k))_i$  = Total de relaciones de precedencia entre actividades de la actividad  $i$

$A(\sum_{j=1}^m A_j)_i$  = Total de actividades de la actividad  $i$

Sin embargo, tras un análisis de los planos generados por AVISPA bajo la fórmula 3.2, éstos no ofrecen la información suficiente para determinar si una actividad tiene un alto factor de acoplamiento, puesto que se omiten las relaciones entre las tareas contenidas en la actividad a analizar. Por lo que se ha indicado un nuevo factor de acoplamiento de actividades, ya no por las actividades contenidas dentro de la actividad, sino por el total de tareas dentro de dicha actividad. La ecuación 3.3 Lo ilustra.

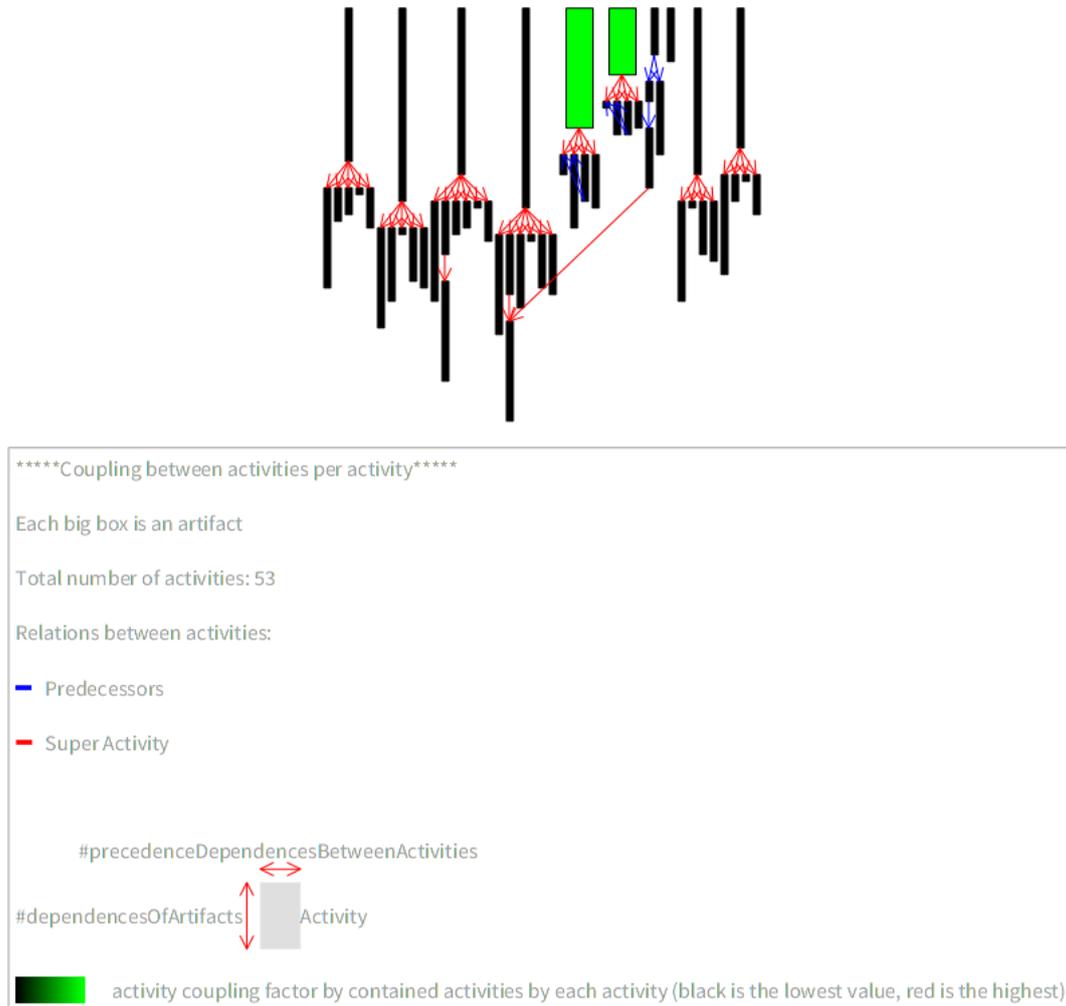
$$NCA_{A_i} = \frac{A(\sum_{j=1}^m P(T_j))_i}{A(\sum_{j=1}^m T_j)_i} \quad (3.3)$$

Donde:

$NCA_{A_i}$  = Factor de acoplamiento de actividades por tareas de la actividad  $i$

$A(\sum_{j=1}^m P(T_k))_i$  = Total de relaciones de precedencia entre tareas de la actividad  $i$

$A(\sum_{j=1}^m T_j)_i$  = Total de tareas de la actividad  $i$



**Figura 3.6:** Patrón de factor de acoplamiento de actividades de la nueva versión de AVISPA

Las gráficas de factor de acoplamiento de actividades por actividades y por tareas están ilustradas en las figuras 3.6 y 3.7 respectivamente.

### 3.4.4.5. Roles responsables de productos de trabajo

AVISPA cuenta con la capacidad de ilustrar que roles son responsables sobre los productos de trabajo y cuáles de ellos los producen / modifican. Como ha sido mencionado anteriormente, el trabajo de M. Bastarrica et. al. [41] demuestra por validación de hipótesis que 4 o más roles que producen / modifican artefactos puede representar posibles cuellos de botella en la ejecución de modelos de proceso por lo que ahora AVISPA soporta la representación visual para ésta hipótesis.

En relación del artículo de M. Bastarrica et al. [41] los roles responsables son

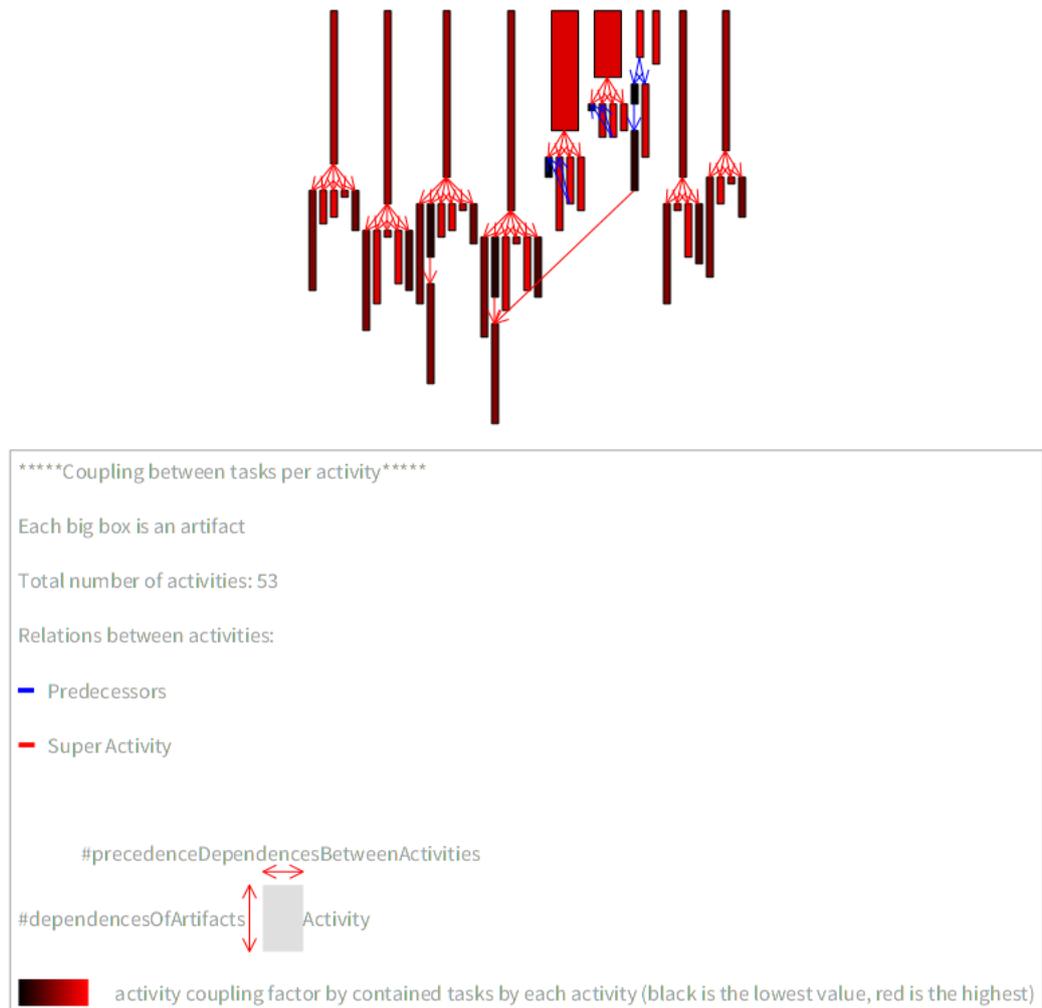
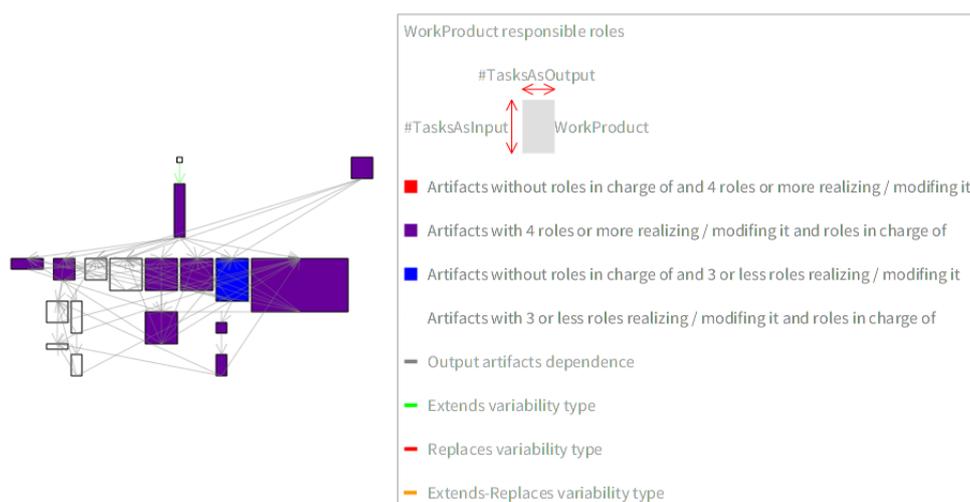


Figura 3.7: Patrón de factor de acoplamiento de tareas de la nueva versión de AVISPA



**Figura 3.8:** Patrón de roles responsables de artefactos de la nueva versión de AVISPA

incluidos bajo la premisa la hipótesis H3 y H4 del trabajo (ver tabla 3.2).

Este patrón clasifica los roles responsables de los que producen o modifican artefactos. Así, este patrón cumple con varias funcionalidades:

- Especificar cuando hay roles responsables en un artefacto.
- Señalar cuando hay 4 o más roles que producen o modifican dicho artefacto (H3).
- Resaltar cuando hay 4 o más roles que producen o modifican dicho artefacto (H3) y al mismo tiempo es responsable de otros artefactos (H4).

La figura 3.8 ilustra este nuevo patrón conforme a las nuevas modificaciones de AVISPA según lo indicado en el trabajo de M. Bastarrica et. al. [41].

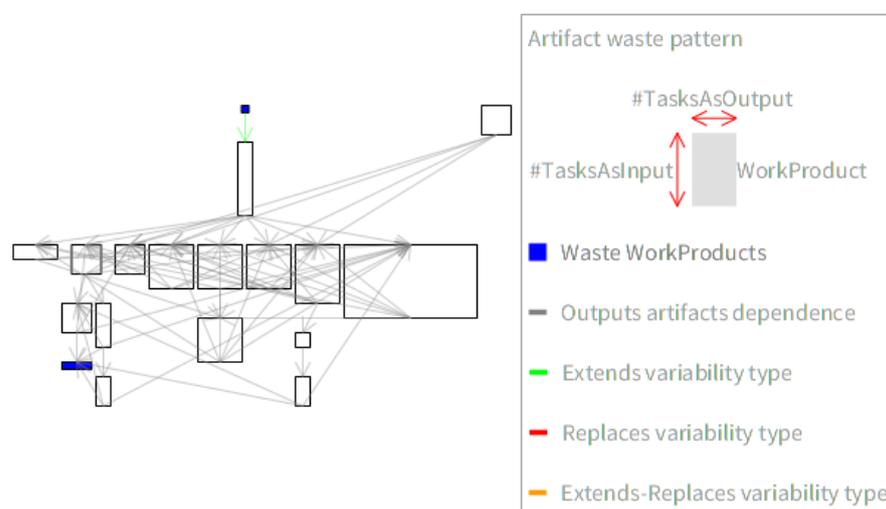
#### 3.4.4.6. Artefactos de trabajo desperdiciados

El patrón de error de trabajos desperdiciados ha sido modificados respecto a la hipótesis H3 y H4 del trabajo del artículo de M. Bastarrica et al. [41] (ver tabla 3.2). Ahora excluye entregables del patrón y ha sido trasladado a la subsección de pruebas desde la sección de patrones de error de la herramienta AVISPA.

La figura 3.9 ilustra éste patrón de error

#### 3.4.5. Soporte de la variabilidad en AVISPA

En la sección 2.4.2 se han indicado una serie de consideraciones adicionales las cuales se implementaron en cada uno de los planos y patrones de error de AVISPA.



**Figura 3.9:** Patrón de roles responsables de artefactos de la nueva versión de AVISPA

Por lo tanto, se han realizado las siguientes modificaciones a la herramienta de visualización:

### 3.4.5.1. Adiciones de planos de artefactos y roles con resolución de variabilidad

La resolución de la variabilidad se ha definido como necesaria puesto que SPEM 2.0 dispone de relaciones de variabilidad de elementos, principalmente de tareas, roles y productos de trabajo. De esta manera, los modelos de proceso documentados en SPEM 2.0 pueden contener relaciones que AVISPA debe contemplar en sus planos y patrones de error.

Los modelos de proceso en AVISPA ahora contienen resolución de variabilidad de modelos de proceso de software extendido a roles y productos de trabajo. Así, permite ilustrar planos y patrones más cercanos a la realidad del modelo. Las figuras 3.10 y 3.11 ilustran la resolución de la variabilidad en artefactos y en patrones respectivamente.

### 3.4.5.2. Resolución de variabilidad en todos los planos de roles, tareas y artefactos

En la sección 2.4.2.3 se expone la variabilidad de elementos de proceso de software y cómo afectan la visualización en todos los planos y patrones de error restantes en AVISPA.

Como ha sido mencionado en dicha sección, los planos y patrones de error que involucren visualización de elementos de proceso como tareas, roles y productos de trabajo soportarán la visualización y resolución de variabilidad. Ofreciendo así infor-

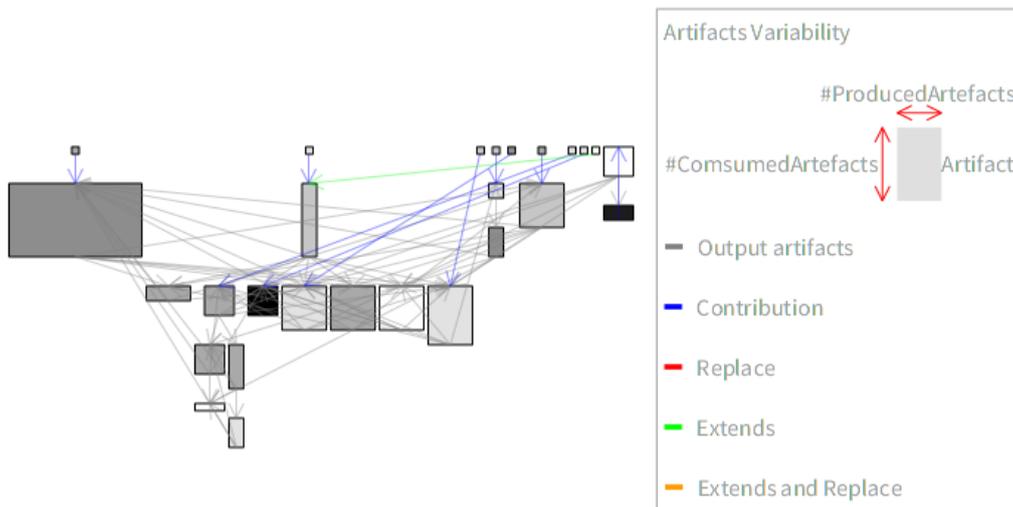


Figura 3.10: Plano de Variabilidad de Artefactos en AVISPA

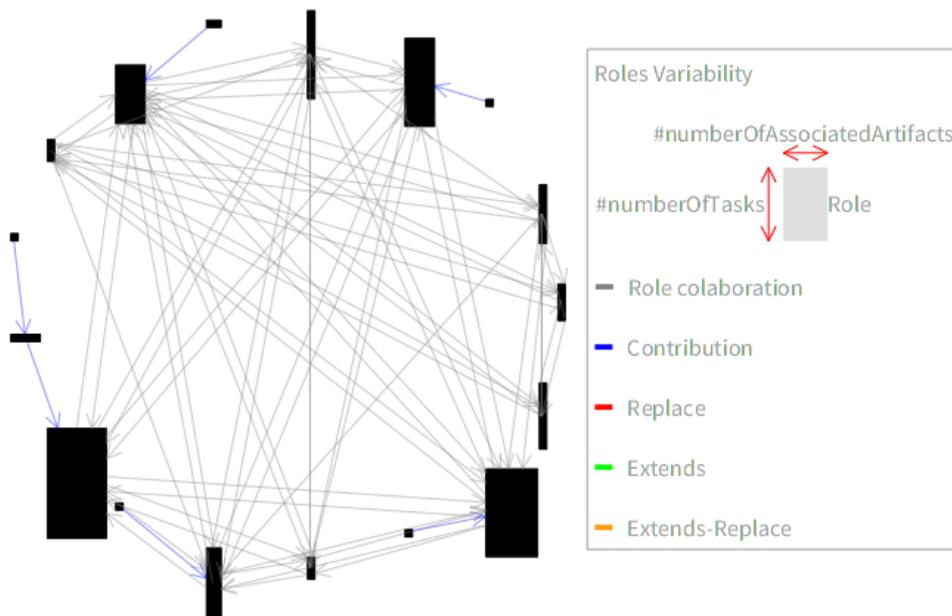
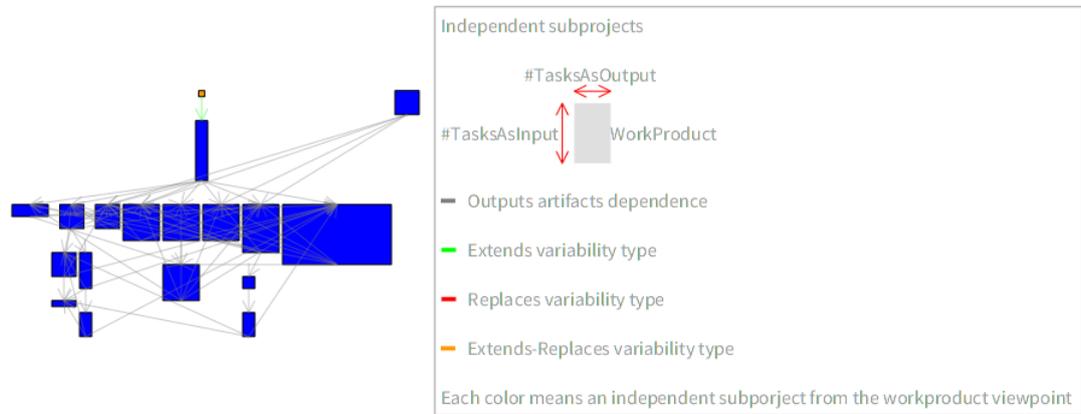


Figura 3.11: Plano de Variabilidad de Roles en AVISPA



**Figura 3.12:** Plano de Subproyectos Independientes en AVISPA

mación más completa en todos los elementos de proceso aproximando los planos a la realidad concreta de la ejecución del modelo de proceso, tal como lo ilustra la figura 3.12.

## Capítulo 4

# Evaluación de AVISPA

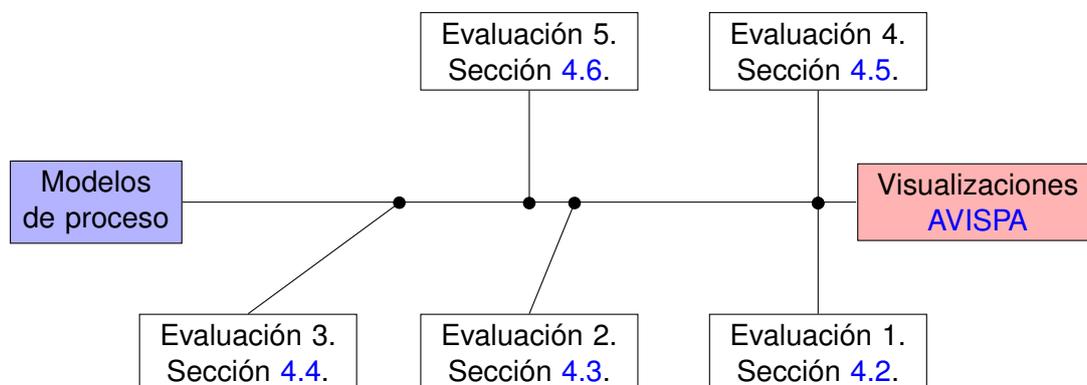
### 4.1. Introducción

El refinamiento de AVISPA ha sido realizado bajo diferentes estrategias de evaluación que han permitido descubrir, desde aspectos de entendimiento de la herramienta y de los modelos de proceso evaluados, hasta las respectivas correcciones del mapeo, relaciones y visualizaciones de planos y patrones generados.

Para poder llevar a cabo la evaluación de la herramienta han sido definidas algunas características básicas que AVISPA debe cubrir antes de su respectiva evaluación:

1. **Entendimiento de los modelos de proceso a analizar:** Para descartar si es necesario conocer el modelo de proceso antes de usar AVISPA como instrumento de toma de decisiones sobre el modelo de proceso, algunas evaluaciones se han llevado a cabo para descartar y / o reforzar éste factor en el refinamiento de las evaluaciones y, por lo tanto, como insumo para la comprensión de la herramienta.
2. **Entendimiento / confiabilidad de los planos y patrones de AVISPA:** Este estudio se ha enfatizado en el refinamiento de las visualizaciones de AVISPA con el ánimo de brindar utilidad de la información presentada para las decisiones de los modelos de proceso documentados. Este factor se ha refinado por medio del trabajo de M. Bastarrica et al. [41], y de evaluaciones de refinamiento del mapeo de los modelos de proceso, sus relaciones y métricas de los planos y patrones de modelos de proceso así como sus respectivas visualizaciones.

Las anteriores características deben ser evaluadas con el objetivo de lograr criterios y bases suficientes para la toma de decisiones frente a la corrección de las visualizaciones, patrones y métricas generadas por la herramienta AVISPA. Para ello se han generado diversas evaluaciones con el objetivo de hallar puntos de alta relevancia para su respectiva mejora.



**Figura 4.1:** Ubicación de evaluaciones entre características a evaluar de AVISPA.

Para esto se han realizado un total de 4 evaluaciones que varían entre las 2 características anteriormente mencionadas que serán detalladas a continuación:

1. **Evaluación 1, entendibilidad de AVISPA (Sección 4.2):** Esta evaluación se realizó sobre una versión preliminar de AVISPA contando con un total de 4 ingenieros de procesos expertos en definición de procesos pero sin conocimiento de los modelos de procesos a evaluar, permitiendo hallar diversas conclusiones sobre la ejecución de pruebas y evaluaciones sobre AVISPA.
2. **Evaluación 2, procesos minimalistas evaluados en AVISPA (Sección 4.3):** Dada la experiencia de documentación de modelos de proceso y planos, patrones y métricas de la anterior evaluación, se crean modelos de proceso mínimos de ejemplo a propósito con el fin ser evaluados con AVISPA donde se evalúan de manera aislada los mapeos de los elementos de proceso, las relaciones y representaciones entre los mismos en AVISPA.
3. **Evaluación 3, pruebas unitarias para validar planos de AVISPA (Sección 4.4):** La flexibilidad de Pharo como entorno para la depuración de objetos y su poderosa sintaxis que facilita la programación y extracción de objetos en tiempo de ejecución han facilitado las pruebas de tal manera que se pueden extraer y comparar los elementos de las visualizaciones, comparar métricas y hasta depurar los componentes visuales de los planos. Esta característica facilita la implementación de pruebas unitarias y, por lo tanto, consistencia en la información de los planos generados por AVISPA en relación con la información de los modelos de proceso documentados en EPF.
4. **Evaluación 4, evaluación de la variabilidad en AVISPA por pruebas de caja negra (Sección 4.5):** La evaluación de la variabilidad de AVISPA se ha realizado utilizando el método de prueba de caso de uso de caja negra. Así, el enfoque de la prueba se realiza más en la evaluación del modelo de negocio (contexto de los planos y patrones) que en el contenido de los datos que sin la apropiada profundización del modelo de proceso no permite brindar la información de

contexto suficientemente valiosa para la toma de decisiones sobre el código de AVISPA.

5. **Evaluación 5, diseño de experimento para la evaluación de AVISPA (Sección 4.6):** La evaluación de AVISPA finalmente se realizó con un diseño experimental para determinar las mejoras de la implementación de esta nueva versión.

La figura 4.1 ilustra la ubicación de cada una de las evaluaciones llevadas a cabo en este trabajo entre estas 2 características.

Algunos de los aportes de valor realizados sobre las pruebas a AVISPA consisten en:

- **Incorporación de pruebas de planos y patrones de error:** Las evaluaciones de las secciones 4.3 y 4.4, incorporan evaluaciones de planos y patrones que no se han intentado anteriormente evaluando la integridad de los colores de los patrones y resaltos de los elementos de proceso respecto al objetivo del plano / patrón.
- **Verificación exhaustiva de las relaciones de variabilidad de elementos de proceso que la soportan:** El trabajo de J. Salgado [69] presenta limitaciones de representación de los elementos de proceso, principalmente en la organización de código para la resolución de dependencias, resolución de la contribución y extensión de la variabilidad a los demás planos y patrones de error soportados por AVISPA.
- **Documentación para replicación de pruebas:** La documentación de las pruebas se realiza con el objetivo de replicarlas, validando el procedimiento realizado para en la presente investigación.

## 4.2. Evaluación 1

Como primera aproximación, a manera de exploración del modelo de negocio de AVISPA, y así mismo conocer problemas en la interacción de los usuarios con los planos generados por la herramienta, se ha implementado una evaluación del nivel de entendimiento por expertos de procesos en diferentes metamodelos.

Además, hasta donde se conoce, no es posible determinar si los planos y patrones son realmente entendibles para cualquier ingeniero de procesos trabajando por sí mismo sin acompañamiento de un experto en la herramienta.

Esta evaluación presenta un estudio empírico cualitativo ha sido propuesto y ejecutado en el cual 4 ingenieros de procesos evalúan 2 modelos de procesos diferentes. La evaluación a su vez usa 2 técnicas de extracción de información: la entrevista semi-estructurada [78] y el método think-aloud [79].

#### 4.2.1. Descripción del estudio empírico

El objetivo de esta evaluación es determinar qué tan comprensibles son los patrones de error para un ingeniero de procesos que no ha tenido experiencia previa en el trabajo con [AVISPA](#).

Este estudio considera dos modelos de procesos de software que serán evaluados por 4 ingenieros de procesos diferentes sin ninguna asistencia en [AVISPA](#). La intervención, que no tenía un límite de tiempo, constaba de 2 secciones, la primera parte dedicada a la comprensión de los planos y patrones de error de [AVISPA](#) y la segunda parte dedicada a la evaluación respectiva.

La evaluación de la herramienta por medio de encuestas fue llevada a cabo como prueba piloto para evaluar el impacto de la nueva versión de [AVISPA](#) en una comunidad de ingenieros en procesos de la Universidad de Chile.

Los evaluados realizaron sus prácticas con 2 modelos de proceso: Tutelkan y OpenUp, cada uno de ellos realizando un análisis de diversos puntos de vista y concluyendo cambios y mejoras desde la experiencia en modelos de proceso.

Los resultados varían significativamente con diversas opiniones encontradas entre los evaluadores, que van desde el entendimiento de los planos hasta la estructura de elementos visuales de la herramienta (estas últimas observaciones no son tomadas en cuenta debido a que se requiere evaluar únicamente la confiabilidad de la información presentada por las visualizaciones generadas por [AVISPA](#)).

Un análisis de la evaluación revela que existe mucha variación sobre los resultados en base a la opinión de los encuestados. Por lo tanto, no hay un control para determinar la eficacia de la herramienta sobre los datos de un proceso software. En efecto, se optó por realizar un análisis a través de Prueba de Significación de la Hipótesis Nula ([PSHN](#)) para determinar la calidad de la información presentada por la herramienta en base a procesos en [SPEM](#).

Para mayor información del contenido de las encuestas ver el anexo [B](#).

#### 4.2.2. Diseño del estudio

De acuerdo al objetivo de esta evaluación, la pregunta de investigación de esta evaluación es: ¿cual es el nivel de entendimiento de los planos y patrones por los ingenieros de proceso?. La hipótesis propuesta cuando se evalúa el entendimiento de un modelo de proceso con conocimiento de él previamente a la evaluación por medio de [AVISPA](#) es que existen dificultades para entender adecuadamente los patrones de error y visualizaciones de [AVISPA](#). La tabla [4.1](#) ilustra la construcción de las métricas asociadas a la presente evaluación.

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Indicador	Escala	Descripción	Información fuentes y Herramientas
NC: Nivel de Comprensión.	<i>Alto (A):</i>	El ingeniero de procesos entiende completamente la estructura del patrón de error / plano indicado en el cuestionario y su función respectiva en el modelo de proceso.	Pruebas en audio, video y entrevistas.
	<i>Medio (M):</i>	El ingeniero comprende parcialmente el plano y / o patrón de error indicado en la pregunta del cuestionario o entiende parcialmente su utilidad en el modelo de proceso.	
	<i>Bajo (B):</i>	Poca o ninguna comprensión del modelo y / o patrón de error indicado en la pregunta del cuestionario o de su utilidad en el modelo de proceso.	
	<i>No sabe / No responde (NS/NR):</i>	No responde la pregunta específica justificando su respuesta o no sabe cómo responder.	

**Tabla 4.1:** Métricas del estudio

### 4.2.2.1. La entrevista semi-estructurada

Las preguntas / actividades del cuestionario realizado para cada uno de los ingenieros de proceso que evaluaron [AVISPA](#) fueron:

- Q-1. Elija una actividad del proceso, la que parezca más compleja, e indique qué roles están sobrecargados, donde hay subproyectos independientes y productos de trabajo demandados, justifique las razones de su apreciación.
- Q-2. Indique en el plano de acoplamiento cuáles son las actividades cuyos elementos de proceso (tareas) están más acoplados.
- Q-3. Indique qué mejoras o correcciones haría al modelo de proceso que ha evaluado.
- Q-4. ¿Encuentra alguna diferencia entre los planos de todos los elementos de proceso del modelo agrupados en un único plano y los planos que visualizan los elementos del modelo de proceso agrupados por actividades? ¿Cuál de estos dos tipos de visualizaciones es más fácil de entender para usted?

### 4.2.2.2. El método think-aloud

El método think-aloud es un protocolo utilizado para recopilar información adicional que otros métodos no pueden, principalmente en el tiempo de ejecución. Este método se utilizó para recopilar información sobre lo que los ingenieros de procesos pueden percibir de [AVISPA](#). El método think-aloud se llevó a cabo de manera transversal en todas las fases de ejecución que involucran la participación de los ingenieros de procesos de software. La evidencia se ha capturado en audio y video que reúne todas las acciones que los ingenieros de procesos realizaron en el software y en la prueba.

#### 4.2.2.3. Alcance

Los resultados de esta evaluación indican algunos aspectos de usabilidad de sugeridos por los entrevistados en este estudio, por lo que, para efectos de redireccionamiento de la ejecución de la presente evaluación, fué necesario filtrar los resultados obtenidos por la pertinencia de los mismos hacia los planos, patrones de error y métricas de AVISPA, y no de otros elementos de usabilidad como la distribución de los elementos de Pharo entre otros.

#### 4.2.2.4. Ejecución

De acuerdo con los comentarios de retroalimentación de los experimentos, el contenido del cuestionario de la introducción a AVISPA así como las preguntas respectivas de evaluación por parte de los ingenieros de proceso pueden variar en la información de contexto para la prueba siguiendo un enfoque iterativo e incremental para cada evaluación realizada.

La evaluación de comprensión de AVISPA se ejecuta en las siguientes etapas:

- **Ruta de Software:** En esta etapa, los ingenieros de proceso realizaron una lectura inicial en la que se presentó la herramienta AVISPA en la que cada uno de los patrones de error se describe de manera conceptual. Además, se realiza un breve recorrido por algunos de los patrones de error de la herramienta, y se hicieron algunas preguntas de contexto sobre la experiencia en la definición de modelos de procesos e idiomas para especificar modelos de procesos. Esta etapa se llevó a cabo con un tiempo programado de 12 minutos.
- **Entrevista:** Luego ejecutaron una serie de pruebas en varios modelos de proceso de software a partir de los cuales identificaron los elementos del modelo de proceso que corresponden a la descripción en cada pregunta correspondiente y llevaron a cabo el análisis respectivo de estos elementos. Esta etapa se llevó a cabo en un tiempo promedio de 15 a 20 minutos.
- **Finalización:** Una vez que se completa la experiencia de interactuar con el sistema AVISPA, los participantes responden preguntas de contexto, principalmente sobre la toma de decisiones del modelo de proceso según lo analizado por AVISPA. Esta etapa se llevó a cabo en un tiempo promedio de 5 a 10 minutos.

### 4.2.3. Resultados

#### 4.2.3.1. Respuestas del cuestionario y calificaciones

- **Ingeniero de procesos 1 (Tutelkan-TRP):**

R-1.1. Actividad: control de proyecto y monitoreo. **Rol sobrecargado:** gestor del proyecto en el gráfico de “contained overloaded roles”, este rol aparece en

rojo y la leyenda indica que clasifica en los “roles sobrecargados”. **Subproyectos independientes:** “Revisar el producto y aseguramiento de la calidad del proceso”, esto es porque en el grafo es el único elemento de color azul. **Productos demandados:** “plan de proyecto” debido a que es el elemento más largo.

- R-1.2. Comprender los requisitos, analizar y validar los requisitos y definir los objetivos de medición.
- R-1.3. Al comenzar a evaluar el inicio, desde cero, y otras herramientas como Newrelic (para evaluar el código), es preferible que comience con un panel de indicadores relevantes para la industria.
- R-1.4. Modelar los procesos de software por actividades es más comprensible, por lo que ver mis actividades es más natural para obtener una perspectiva más amplia.

■ **Ingeniero de procesos 2 (OpenUP):**

- R-2.1. Iteración de Elaboración: Desarrollador (**sobrecargado**) desde que él realiza más tareas. **Subproyectos independientes:** 5. **Artefactos demandados:** vision (entrada para mas tareas).
- R-2.2. Inicio, mas claro, mayor número de precedencias.
- R-2.3. *No responde*, ya que considera que una mejora en un modelo de proceso requiere ciclos de mejora de procesos y un trabajo más elaborado para considerar una mejora.
- R-2.4. Encuentro que las visualizaciones son útiles o fáciles dependiendo de si está analizando todo el esfuerzo del proceso o por actividad.

■ **Ingeniero de procesos 3 (Tutelkan-TRP):**

- R-3.1 Patrón de error: Plano de Tareas en Actividades, Rol sobrecargado: “Gestor del proyecto”, subproyectos independientes: “Revisar informes de aseguramiento de calidad de productos y procesos”, Artefactos capturados: “Plan de proyecto”
- R-3.2 “Analizar y validar los requisitos, comprender los requisitos, definir los objetivos de medición, describir los requisitos, planificar la gestión de riesgos, monitorear los riesgos”, “plano: acoplamiento de actividades por tareas contenidas”
- R-3.3 *No responde*
- R-3.4 *No responde*

■ **Process engineer 4 (OpenUP):**

- R-4.1 De acuerdo con el plan de actividad y tareas, puedo reconocer más fácilmente el grado de criticidad de estas actividades. La actividad más acoplada es: desarrollo de la arquitectura. En los roles del plano sobrecargados por actividad, encontramos a las partes interesadas del analista probador

<b>Plano / Patron</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Roles sobrecargados	M	B	M	M
Subproyectos independientes	M	B	M	B
Producto demandados	M	B	M	B
Acoplamiento de actividades por actividades	M	M	M	M
Acoplamiento de actividades por actividades	M	M	M	B

**Tabla 4.2:** Clasificación del nivel de comprensión de los patrones de error por parte de los ingenieros de proceso participantes

presentes en la iteración de Elaboración. No encontré un plan que indique subproyectos independientes. Tampoco encontré productos de trabajo demandados.

R-4.2 La iteración de inicio tiene más tareas de acoplamiento.

R-4.3 *No responde*

R-4.4 “Los modelos agrupados por actividades”

Sobre la base de los resultados anteriores, los resultados se han catalogado en relación con cada plano / patrón evaluado de [AVISPA](#) que se ilustra en la tabla 4.2

#### 4.2.3.2. Resultados del método think-aloud

- **Ingeniero de procesos 1 (Tutelkan-TRP):** expone los beneficios de [AVISPA](#) en el momento de realizar la extracción de metadatos del proceso de software. Los planos se consideraron bien organizados y comprensibles, pero los resultados del cuestionario para este ingeniero de proceso no reflejaron ese hecho. Sin embargo, la codificación de colores de la herramienta [AVISPA](#) fue esencial para determinar algunos patrones de error del proceso de software y comprenderlos.
- **Ingeniero de procesos 2 (OpenUP):** también considera que la distribución de los elementos de la interfaz gráfica de usuario de [AVISPA](#) es desordenada, algunos de los planos producen ruido visual y, por lo tanto, confusión en el momento de encontrar los planos que responden a las preguntas formuladas en el cuestionario. Al analizar los videos de los investigadores del trabajo, el ingeniero de procesos dedicó demasiado tiempo a los planos que involucran gradientes de color. Parece que los gradientes de color no son representaciones visuales eficientes para los metadatos de planos de proceso de modelo, principalmente el factor de acoplamiento de actividad por el plano de tarea contenida porque el gradiente de color tiende a ser más confuso al representar cada factor de acoplamiento de tarea. Sin embargo, el ingeniero de procesos 2 tuvo experiencias y enfoques anteriores con [AVISPA](#) y considera que los elementos de proceso agrupados por actividades son mucho más comprensibles que las versiones anteriores que clasificar los elementos de proceso por todo el proceso.

- **Ingeniero de procesos 3 (Tutelkan-TRP):** tiene otra representación del proceso de software debido a que se convierte en el metalenguaje del proceso de software BPMN. Algunas de las preguntas del cuestionario se convierten en una alta complejidad para obtener una respuesta adecuada y la evidencia en video ilustra las razones, principalmente relacionadas con la dificultad de interpretación alta de los planos y los errores de redacción del cuestionario.
- **Ingeniero de procesos 4 (OpenUP):** no lee completamente la descripción del cuestionario, por lo que un cierto nivel de negligencia está implícito en las respuestas y decisiones tanto en el cuestionario como en la herramienta.

### 4.3. Evaluación 2

La evaluación obedece a una depuración profunda acerca de la organización de la información en el código de [AVISPA](#) en el cual era necesario visualizar la forma de cómo los elementos de proceso eran administrados y almacenados en las diferentes variables del código de la herramienta por medio de grafos de control organizados por bloques de código y el contenido de las variables en transcurso del flujo de ejecución del mismo.

El nivel de granularidad de la división de los bloques de código del grafo de control de flujo obedece a indicios de mal manejo de información por parte del código escrito en ciertos puntos críticos de clasificación de la información dentro de los códigos analizados en esta evaluación. La selección de los códigos a ser evaluados se realizó tomando en cuenta su relevancia para la adecuada organización de elementos de proceso, con el objetivo de generar adecuadamente patrones de error de alta relevancia y complejidad para [AVISPA](#). El anexo [E](#) ilustra el contenido detallado esta evaluación.

### 4.4. Evaluación 3

#### 4.4.1. Justificación de la evaluación

En orden de asegurar que los datos importados y visualizados concuerden con los objetivos tanto del plano / patrón como del XML exportado, se realizan las pruebas unitarias que responden a comparar cada elemento finalmente con el plano / patrón deseado, y con el modelo de proceso documentado en [EPF](#). En el presente estudio se ilustrará la técnica aplicada de un solo plano.

#### 4.4.2. Diseño de la evaluación

Las pruebas unitarias deben cumplir con tres componentes esenciales para tener trazabilidad en todas las secciones que dependen principalmente de la programación de los planos / patrones y no de los componentes del core de Pharo:

1. Los elementos de proceso importados deben corresponder con los elementos de proceso de los modelos de proceso documentados en [EPF](#).
2. Las relaciones entre los elementos de proceso importados deben corresponder con las relaciones entre elementos de proceso del modelo de proceso documentado en [EPF](#).
3. Las visualizaciones y métricas de los modelos de proceso deben corresponder con los cálculos y decisiones realizadas por [AVISPA](#) y por la realidad del estado de los modelos de proceso documentados [EPF](#).

La figura [4.2](#) ilustra el flujo de las pruebas unitarias realizadas para [AVISPA](#).

#### 4.4.3. Ejecución sobre el patrón de roles responsables

Este estudio se realiza con el patrón de roles responsables indicado en la sección [3.4.4.5](#) del capítulo anterior, reutilizando el ejemplo del modelo de proceso generado en la sección [4.3](#). Partiendo del flujograma de la figura [4.2](#) las pruebas respectivas se generan de la siguiente manera:

1. **Organización de las pruebas:** Tal como está indicado en el flujograma [4.2](#), las pruebas unitarias deben estar organizadas primero por tipo de elemento de proceso, en este caso, los artefactos serán el objeto de análisis de estas pruebas, por lo que se realiza el caso de prueba con este tipo de elemento de proceso. Luego se categorizan los métodos dentro del plano / patrón a evaluar.
2. **Definición del método de retorno del modelo en XML:** el método del modelo de proceso es una cadena de caracteres cuyo contenido es el mismo del archivo de exportación del modelo de proceso documentado en [EPF](#).
3. **Prueba sobre el código de importación:** El método `importXMLDocument: document` se encarga de procesar el documento XML exportado de [EPF](#) y lo transforma en elementos y relaciones entendibles para Pharo / Moose / Mondrian. Esta función a su vez llama a otra llamada `correlationIOArtifactsAndPerformerRoles: tasks to: pmTasks and: pmArtifacts and: pmRoles` que, como su nombre lo indica, se encarga de la importación y resolución de relaciones entre los artefactos y los roles responsables. Sin embargo, las pruebas llaman al primer método mencionado, ya que éste se encarga de llamar al segundo método que finalmente resuelve los elementos y relaciones para el presente plano / patrón de error. Las pruebas finalmente se encargan de verificar dichas relaciones.
4. **Prueba sobre el patrón de error:** la función `viewArtifactResponsibleRolesOn: aView` se encarga de implementar la visualización del patrón con los datos ya cargados del punto anterior usando Mondrian. Para estas pruebas unitarias se llama tanto a la función de importación como la función de visualización, verificando que cada uno de los componentes visualizados estén correctamente resaltados según su finalidad.

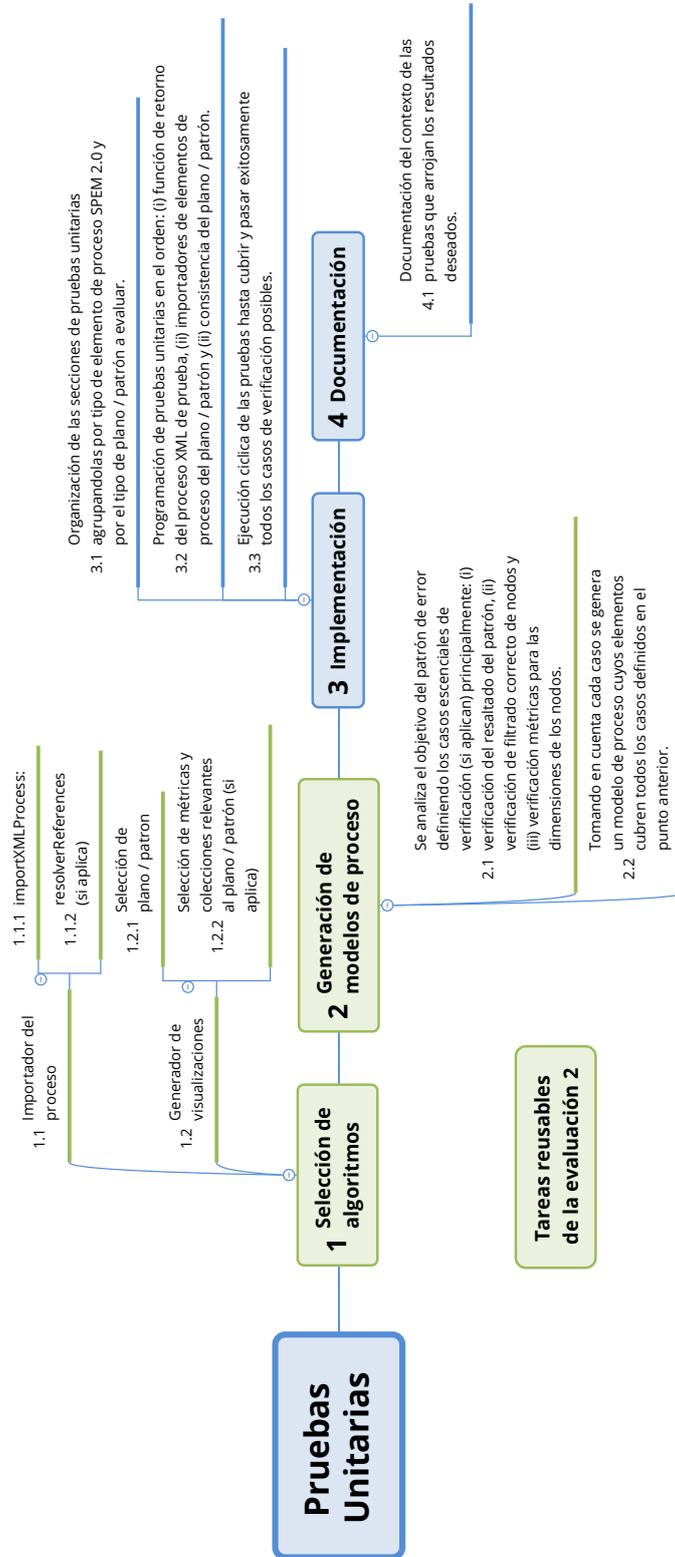


Figura 4.2: Flujo de pruebas unitarias por cada plano de AVISPA

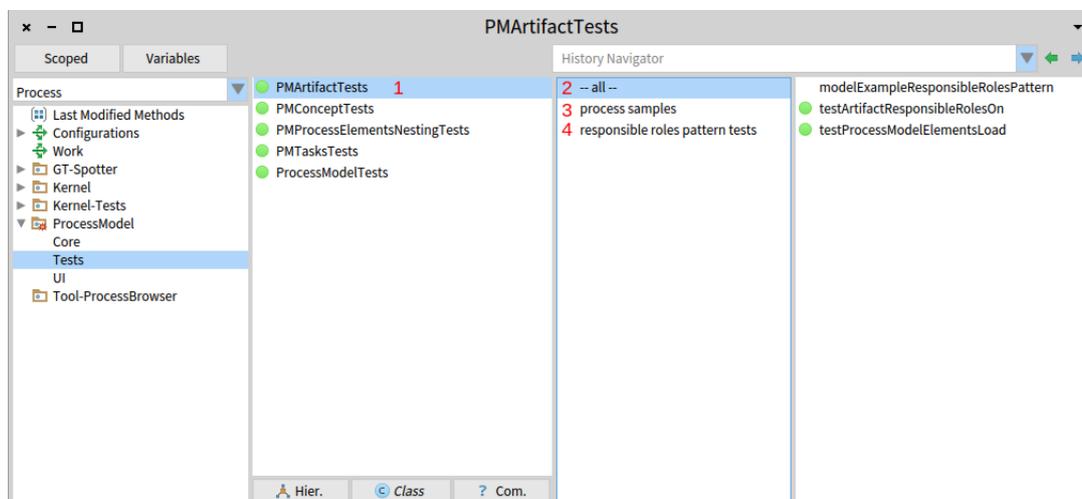


Figura 4.3: Organización del entorno para pruebas en AVISPA

Como se ilustra en la figura 4.3, la organización de las pruebas en Pharo se organizan en:

1. **El tipo de elemento de proceso:** seguido de la palabra `Tests`, en este caso el caso de prueba `PMArtifactTests`.
2. **Categoría de todos los métodos creados para pruebas:** Cada opción dentro de ésta categoría se conoce como “protocolos” que son aquellos de agrupar los métodos dentro del caso de prueba de `PMArtifactTests` en categorías, en ésta opción se ilustran todos los métodos creados.
3. **Categoría de modelos de procesos de prueba:** Este protocolo se encarga de organizar todos los métodos que retornan el modelo de proceso en XML.
4. **Categoría de pruebas por plano / patrón:** Esta categoría se encarga de colocar todos los métodos que realizan las pruebas de los planos patrones con sus respectivas evaluaciones de las importaciones y relaciones de elementos de proceso, para este item, la categoría encierra al patrón de roles responsables. Es importante colocar en el método el prefijo `test` para que Pharo lo reconozca como prueba

Así, todos los métodos de la figura 4.3 se ubican en el último segmento de la derecha de la figura:

- **Método** `modelExampleResponsibleRolesPattern`: contiene el modelo de proceso de prueba en XML tal como se ilustra en la figura 4.4.
- **Método** `testProcessModelElementsLoad`: contiene la prueba unitaria de elementos importados con sus respectivas relaciones tal como se ilustra en la figura 4.5.

```
1 modelExampleResponsibleRolesPattern:  
2 ^ '<?xml version="1.0" ... </uma:MethodLibrary>'
```

**Figura 4.4:** Código del método del modelo de proceso de prueba para el patrón de roles responsables

- **Método** `testArtifactResponsibleRolesOn`: contiene la prueba unitaria de los elementos visualizados en el patrón de roles responsables tal como se ilustra en la figura 4.6. Cabe resaltar que los casos de las líneas 13 a 20 evalúan cada uno de los elementos de proceso y su respectivo color dentro del patrón. De esta manera se evalúa si los elementos de proceso están correctamente representados.

### 4.4.3.1. Resultados

En la figura 4.3 se pueden visualizar unos círculos rellenos con verde. Estos son botones que ejecutan cada una de las pruebas que contiene la categoría o sencillamente un único método. A continuación se explica el código de colores para estas pruebas:

- **Verde:** La(s) prueba(s) o el / los caso(s) se ha(n) ejecutado sin problemas
- **Amarillo:** El / los caso(s) de la(s) prueba(s) o se ha(n) ejecutado satisfactoriamente, la(s) salida(s) esperada(s) es /son distinta(s) a la(s) obtenida(s).
- **Rojo:** La prueba esta escrita incorrectamente, moose / pharo no puede ejecutarla.

En la misma figura mencionada anteriormente todas las pruebas unitarias se han ejecutado satisfactoriamente.

## 4.5. Evaluación 4

### 4.5.1. Justificación de la evaluación

La variabilidad de los modelos de proceso de software es un concepto de alta complejidad donde primero es necesario lograr un nivel de entendimiento adecuado (negocio). Para ello se procede a usar las pruebas de casos de uso de caja negra [81] con el fin de verificar el funcionamiento adecuado de la resolución de la variabilidad de los elementos de proceso en [SPEM 2.0](#).

```
1 testProcessModelElementsLoad
2   | importer model |
3
4 "Initialization of the model"
5 importer := ProcessModelImporter new.
6 model := importer importFromStream: self modelExampleResponsibleRolesPattern readStream.
7
8 "Activities, roles, and artifacts load tests"
9 self assert: model roles size = 5.
10 self assert: model artifacts size = 4.
11 self assert: model tasks size = 2.
12
13 "Responsible roles tests"
14 self assert: model artifacts first responsibleRoles first equals: model roles first.
15 self assert: model artifacts fourth responsibleRoles first equals: model roles fifth.
16
17 "Task outputs tests"
18 self assert: model tasks first artifactOutputs size = 2.
19 self assert: model tasks first artifactOutputs first equals: model artifacts first.
20 self assert: model tasks first artifactOutputs second equals: model artifacts second.
21 self assert: model tasks second artifactOutputs size = 1.
22 self assert: model tasks second artifactOutputs first equals: model artifacts third.
23
24 "Responsible roles tests"
25 self assert: model artifacts first responsibleRoles first equals: model roles first.
26 self assert: model artifacts fourth responsibleRoles first equals: model roles fifth.
27
28 "Roles that perform artifacts"
29 self assert: model artifacts first performerRoles size = 4.
30 self assert: model artifacts first performerRoles first = model roles first.
31 self assert: model artifacts first performerRoles second = model roles second.
32 self assert: model artifacts first performerRoles third = model roles third.
33 self assert: model artifacts first performerRoles fourth = model roles fourth.
34 self assert: model artifacts second performerRoles size = 4.
35 self assert: model artifacts second performerRoles first = model roles first.
36 self assert: model artifacts second performerRoles second = model roles second.
37 self assert: model artifacts second performerRoles third = model roles third.
38 self assert: model artifacts second performerRoles fourth = model roles fourth.
39 self assert: model artifacts third performerRoles size = 3.
40 self assert: model artifacts third performerRoles first = model roles first.
41 self assert: model artifacts third performerRoles second = model roles second.
42 self assert: model artifacts third performerRoles third = model roles third.
43 self assert: model artifacts fourth performerRoles size = 0.
```

**Figura 4.5:** Código de pruebas de las relaciones generadas del proceso XML para el patrón de roles responsables

```
1 testArtifactResponsibleRolesOn
2 | importer model v artifacts |
3
4 importer := ProcessModelImporter new.
5 model := importer importFromStream: self modelExampleResponsibleRolesPattern readStream.
6 v := RTView new.
7 model artifactsGroup viewArtifactResponsibleRolesOn: v.
8
9 artifacts := OrderedCollection new.
10
11 "Testing that model artifacts has correct pattern colors."
12 v elements do:[ :element |
13     (element model = model artifacts first)
14     ifTrue: [ self assert: element trachelShape color = Color purple ].
15     (element model = model artifacts second)
16     ifTrue: [ self assert: element trachelShape color = Color red ].
17     (element model = model artifacts third)
18     ifTrue: [ self assert: element trachelShape color = Color blue ].
19     (element model = model artifacts fourth)
20     ifTrue: [ self assert: element trachelShape color = Color white ].
21 ].
```

Figura 4.6: Código de pruebas del patrón de roles responsables

### 4.5.2. Diseño de la evaluación

Para la presente evaluación se utiliza la plantilla de A. Cockburn [82] adaptada para la realización de los casos de uso involucrados en la evaluación de caja negra. La figura 4.7 ilustra el proceso seguido para la evaluación de caja negra por medio de casos de uso. Así, se realizan todos los flujos de prueba correspondientes a la resolución de las relaciones de variabilidad en los planos y patrones de AVISPA.

Las pruebas se realizan a nivel de todos los planos, sin embargo, únicamente se ilustra el cambio entre la versión de J. Salgado et al. [69] y la nueva versión de AVISPA del plano de variabilidad de tareas.

### 4.5.3. Resultados

#### 4.5.3.1. Casos de uso

Cada caso de uso documentado se realiza de manera general ya que las relaciones de variabilidad se cumplen indistintamente del tipo de elemento de proceso que las implementa.

Los casos de uso están orientados a la comprensión y pruebas de la resolución de las relaciones de variabilidad. Así, la plantilla de A. Cockburn [82] se ha reducido a los siguientes campos que, en este punto del presente trabajo, presentan valor al negocio (pruebas de software en AVISPA):

- ID.

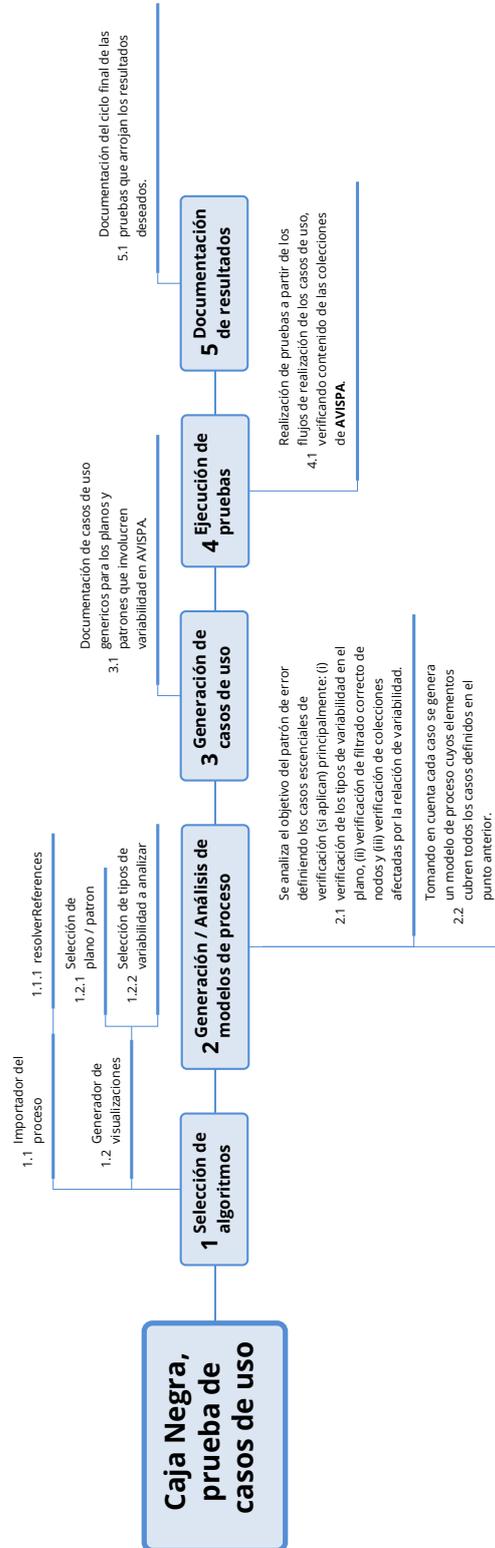


Figura 4.7: Flujograma de pruebas unitarias por cada plano de AVISPA

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

<b>Caso de Uso</b>	1
<b>Nombre</b>	Resolución Contribución
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Ya que <b>EPF</b> no permite asignar el nombre de presentación a los <b>ES</b>, es necesario que el nombre de presentación sea el mismo identificador en <b>AVISPA</b> para tener forma de identificarlo visualmente.</li> <li>2. Las relaciones de entrada y salida de los elementos <b>ES</b> se unen con las de <b>EB</b>.</li> <li>3. Se ocultan los <b>ES</b>.</li> </ol>
<b>Extensiones y/o excepciones</b>	Ninguna

**Tabla 4.3:** Caso de Uso de resolución de la contribución en **AVISPA**.

<b>Caso de Uso</b>	2
<b>Nombre</b>	Resolución Extensión
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Las salidas de <b>EB</b> se añaden a los <b>ES</b> solo si el <b>ES</b> de turno no tiene salidas.</li> <li>2. Los elementos <b>EB</b> y <b>ES</b> no se ocultan.</li> </ol>
<b>Extensiones y/o excepciones</b>	<ol style="list-style-type: none"> <li>1. Si el <b>ES</b> de turno ya posee relaciones de salida, la resolución de la variabilidad se ignora, es decir, no se resuelve.</li> </ol>

**Tabla 4.4:** Caso de Uso de resolución de la extensión en **AVISPA**.

- **Nombre.**
- **Descripción:** describe el flujo normal / exitoso y esperado de eventos.
- **Extensiones:** describe comportamientos adicionales que pueden ser definidos al momento de la ejecución del caso de uso.
- **Extensiones y/o excepciones:** describe comportamientos adicionales y errores esperados que pueden ser ejecutados al momento de la ejecución del caso de uso.

Las tablas 4.3, 4.4, 4.5 y 4.6 ilustran los casos de uso de **AVISPA** que cubren todas las relaciones de variabilidad presente en los elementos de un modelo de proceso en **SPEM 2.0**. También es necesario entender la taxonomía usada en la sección 2.4.1 para entender la especificación de los casos de uso.

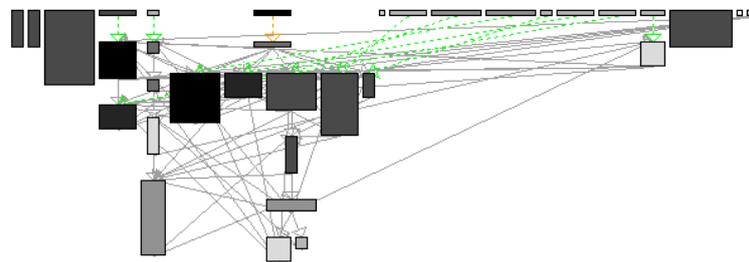
Es necesario tener en cuenta que la unión estándar, estrictamente en lógica de conjuntos, a diferencia de la unión disyunta, consiste en unir los elementos de dos subconjuntos sin repeticiones de elementos en el resultado final. Así, los casos de uso, al utilizar la unión de conjuntos de relaciones entre elementos de procesos, se referirán siempre a la unión estándar.

<b>Caso de Uso</b>	3
<b>Nombre</b>	Resolución Reemplazo
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Las salidas de <b>ES</b> sobrescribirán todas las relaciones de salida del elemento resultante independientemente si <b>ES</b> posee o no salidas.</li> <li>2. Las entradas de <b>ES</b> se unen con las de <b>EB</b> en el elemento resultante.</li> <li>3. <b>EB</b> se oculta, en caso de resolver la relación de variabilidad.</li> </ol>
<b>Extensiones y/o excepciones</b>	<ol style="list-style-type: none"> <li>1 Si existe mas de un <b>ES</b> para un <b>EB</b> la resolución de variabilidad no se lleva a cabo.</li> </ol>

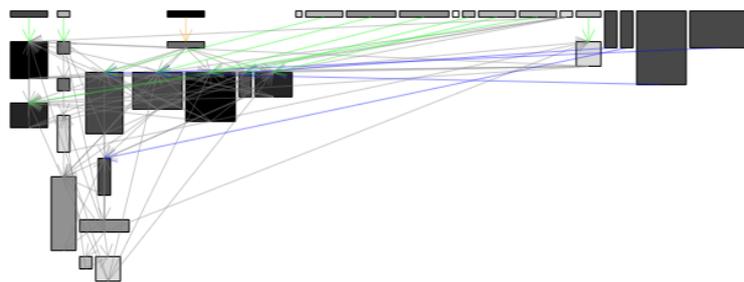
**Tabla 4.5:** Caso de Uso de resolución del reemplazo en [AVISPA](#).

<b>Caso de Uso</b>	4
<b>Nombre</b>	Resolución Extensión-Reemplazo
<b>Descripción</b>	<ol style="list-style-type: none"> <li>1. Si <b>ES</b> define asociaciones de salida, éstas reemplazarán todas las asociaciones de salida de <b>EB</b>.</li> <li>2. Si <b>ES</b> no define asociaciones de salida, se conservarán todas las asociaciones de salida de <b>EB</b>.</li> <li>3. Las asociaciones de entrada de <b>EB</b>, se unen con las de <b>ES</b>.</li> <li>4. <b>EB</b> se oculta, en caso de resolver la relación de variabilidad.</li> </ol>
<b>Extensiones y/o excepciones</b>	<ol style="list-style-type: none"> <li>1 Si existe más de un <b>ES</b> para un <b>EB</b> la resolución de variabilidad no se lleva a cabo.</li> </ol>

**Tabla 4.6:** Caso de Uso de resolución de extensión-reemplazo en [AVISPA](#).



(a) Antiguo AVISPA.



(b) Nuevo AVISPA.

Figura 4.8: Diferencias entre versiones de AVISPA del plano de variabilidad de tareas

#### 4.5.3.2. Correcciones realizadas en AVISPA

- Los elementos **ES** en la contribución poseen como nombre el identificador.
- La resolución de la contribución en AVISPA ha sido arreglada.
- Las restricciones de 2 o mas **ES** a un **EB** en los tipos **reemplazo** y **extensión-reemplazo** ya están incluidas.

La figura 4.8 se visualizan las diferencias entre las versiones anterior y nueva de AVISPA tras la presente evaluación.

## 4.6. Evaluación 5

### 4.6.1. Introducción

La evaluación incremental de fue la estrategia seleccionada en esta tesis para determinar, refinar y evaluar los modelos producidos en AVISPA.

La herramienta AVISPA ha sido evaluada siguiendo el proceso:

1. Inicialmente se han definido una serie de encuestas dirigidas a la entendibilidad exclusivamente de los planos generados por la herramienta

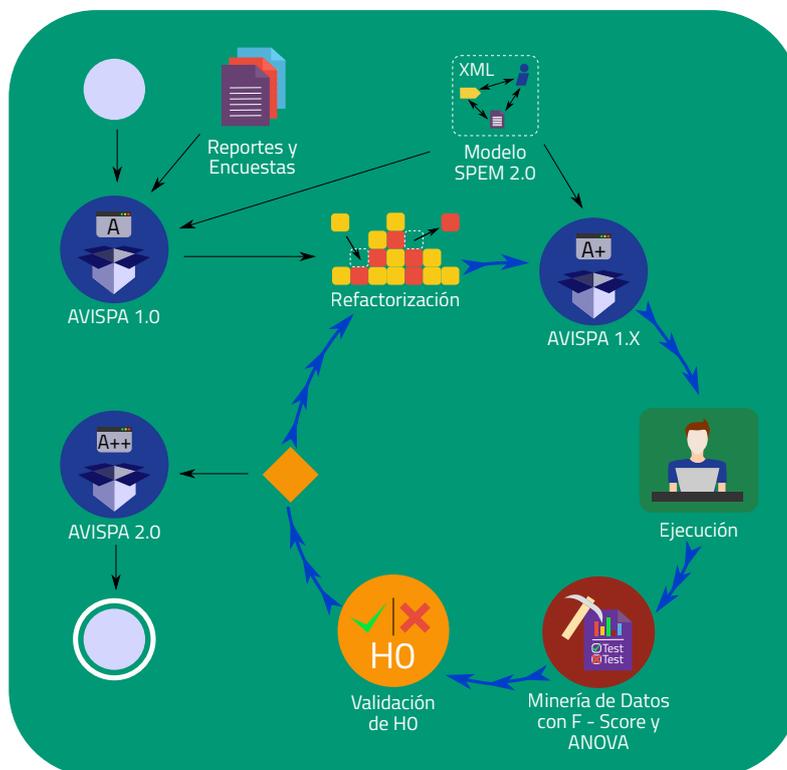


Figura 4.9: Proceso de evaluación de AVISPA.

2. Una vez entendido el contexto y necesidades de los ingenieros de proceso respecto a la herramienta AVISPA se realizan correcciones de la misma utilizando diferentes técnicas de evaluación / corrección de código.
3. Luego, se definió un diseño de experimentos para validar la implementación de los planos realizados en AVISPA por medio del análisis ANOVA.

La figura 4.9 ilustra el proceso seguido para el refinamiento de la herramienta.

El trabajo de L. Pfleeger [26] ilustra una guía para el diseño de experimentos de la ingeniería de software definiendo actividades clave para su correcta ejecución. También ilustra un conjunto de prácticas para lograr un adecuado control de las variables.

En el diseño de experimentos existe un procedimiento que determina una serie ordenada de pasos a seguir para su ejecución formal:

- **Concepción:** Definición de las necesidades de aprendizaje y objetivos del experimento.
- **Diseño:** Descender los objetivos del experimento a hipótesis formales (alternativa y nula). Definiendo para ello instrumentos de medición, factores de influencia en el experimento, técnicas para pruebas entre otros.

- **Preparación:** Involucra organización de los sujetos para aplicación del experimento (configuración software y hardware, instrucciones escritas y grabadas adecuadamente). Aquí pueden realizarse pruebas pilotos con grupos pequeños para asegurarse de que el plan está completo y las instrucciones se entienden.
- **Ejecución:** Se ejecuta el experimento conforme a los pasos definidos en el plan. Es necesario ser cuidadoso en que las medidas y tratamientos son aplicados consistentemente.
- **Análisis:** Se filtran los resultados para determinar si son útiles y validos, se interpretan los resultados de acuerdo a los objetivos de la investigación y la evaluación de las hipótesis
- **Diseminación y toma de decisiones:** Se ilustran detalladamente los cambios en el experimento como los elementos o herramientas han sido usados adicionalmente al momento de realizar el experimento entre otros con el objetivo de replicarlo.

#### 4.6.2. Concepción

Para efectos de la investigación, el objetivo es determinar la diferencia de la veracidad y calidad de la información de los planos generados entre las 2 versiones de AVISPA.

Este objetivo del experimento ha sido un elemento crítico, pues permitió validar el objetivo general del presente trabajo al someter a prueba el uso de AVISPA respecto a sus métricas para la generación de los blueprints.

#### 4.6.3. Diseño

##### 4.6.3.1. Formulación de Hipótesis

Para cumplir con el objetivo de la investigación se formulan las siguientes hipótesis para efectos de dirigir el experimento a su resultado final:

- **Hipótesis Alternativa  $H_1$ :** La información generada la herramienta [AVISPA 2.0](#) representa una diferencia significativa respecto a su anterior versión, para detectar patrones de error recurrentes en base a la información obtenida de modelos de proceso elaborados en [SPEM 2.0](#), en por medio del *F-Score* y de ANOVA.
- **Hipótesis Nula  $H_0$ :** La información presentada por la herramienta AVISPA 2.0 carece de una diferencia significativa respecto a su versión anterior. Debido a lo anterior es necesario reevaluar las interpretaciones de los usuarios de [AVISPA 2.0](#) en relación a los modelos de proceso elaborados en SPEM 2.0, así como la eficiencia de la clasificación de [AVISPA 2.0](#).

Así  $H_0$  es verdadera hasta que los datos demuestren lo contrario.

		Clase observada	
		Verdadero	Falso
Clase Predecida	Verdadero	Verdadero Positivo (VP)	Falso Positivo (FP)
	Falso	Falso Negativo (FN)	Verdadero Negativo (VN)

Un buen clasificador ubica sus elementos en las secciones verdes, los elementos mal ubicados se clasifican en las categorías de rojo.

**Tabla 4.7:** Categorías de clasificación de elementos de clasificadores binarios.

#### 4.6.3.2. Matriz de Confusión

La matriz de confusión es una representación del desempeño de la clasificación correcta e incorrecta de elementos de un modelo.

La matriz de confusión más simple es la binaria o de dos grados de libertad (clases) *positivo/negativo, blanco/negro, verdadero/falso*.

Así pues, el desempeño de un clasificador binario se resume a una matriz de confusión 2 x 2 que tabula la información cruzando las muestras predichas de las observadas en cuatro opciones:

- **Verdadero Positivo (VP):** Predicción correcta de una clasificación positiva (Predicción: “positivo”, realidad: “positivo”),
- **Verdadero Negativo (VN):** Predicción correcta de una clasificación negativa (Predicción: “negativo”, realidad: “negativo”),
- **FP:** Falsa predicción de una clasificación positiva (Predicción: “positivo”, realidad: “negativo”),
- **Falso Negativo (FN):** Falsa predicción de una clasificación negativa (Predicción: “negativo”, realidad: “positivo”).

Así entonces, para que un clasificador ordene los elementos correctamente es necesario que todos sean VP y VN.

La matriz de confusión, en efecto, se representa en la figura 4.7. La matriz de confusión es un organizador cuya funcionalidad consiste en evaluar el desempeño de la clasificación de AVISPA a través de la prueba F-score.

#### 4.6.3.3. F-score y ANOVA

ANOVA es una técnica estadística para el análisis de varianzas que permite definir si existe similitud o diferencia significativa entre dos muestras de datos.

F-Score, es la media armónica entre la precisión y la exhaustividad / recuerdo, es una medida de precisión utilizada para calcular la precisión de algoritmos de búsqueda y recuperación de la información.

Mientras F-score se utiliza para hallar la precisión  $p$  y el recuerdo  $r$ , la prueba ANOVA se realiza para hallar el índice de varianza  $\alpha$  respecto a un único valor F-Score.

Así, se utiliza el análisis de varianza ANOVA de un factor con una única muestra por grupo, pues la prueba F-Score únicamente produce una única calificación de la clasificación por modelo de proceso en [AVISPA](#).

#### 4.6.4. Preparación

##### 4.6.4.1. ANOVA instanciado al diseño del experimento en [AVISPA](#)

**Variables Independientes** Las variables independientes, al ser los tratamientos, factores o condiciones que se controlan para probar la variable independiente, serían los modelos de procesos de software junto con los grupos de evaluación del modelo por medio de [AVISPA](#).

**Variables Dependientes** Las variable dependiente sería en este caso la que refleja los resultados de un estudio de investigación, en este caso serían las 2 versiones de [AVISPA](#), es necesario determinar entonces la varianza de los resultados más los planos que están en la nueva versión más las justificaciones de los planos que se eliminaron en esta nueva versión.

**Fórmulas** Debido a lo anterior, es necesario definir el conjunto de fórmulas que se involucran en la evaluación por medio de ANOVA:

- Establecer las formulas necesarias para evaluar los clasificadores de los elementos de modelo de proceso, en este caso la precisión, el recuerdo y F-Score que finalmente será un único valor que correlaciona las primeras 2 métricas.
- Las fórmulas de implementación del análisis de varianza ANOVA a partir de cada una de las evaluaciones F-Score de los modelos de proceso evaluados

**Precisión, Recuerdo y F-Score** La ecuación [4.1](#) y [4.2](#) ilustran las fórmulas de precisión y recuerdo respectivamente para la calificación de clasificadores.

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Formula	Descripción
$T_{A_j} = \sum_i^{N_j} Y_{ij}$	$T_{A_j}$ es la suma de los puntajes del grupo $A_j$ . Se suman los valores de las observaciones de cada uno de los grupos en $A$ y a veces es llamado $T_j$ .
$\psi = \frac{\left(\sum_j \sum_i y_{ij}\right)^2}{N} = N\bar{Y}^2$	Sumar todas las observaciones. elevar al cuadrado el resultado y dividirlo por el número total de observaciones.
$\lambda = \sum_j \sum_i y_{ij}^2$	Elevar al cuadrado cada observación y sumar cada resultado.
$\epsilon = \sum_j \frac{T_{A_j}^2}{N_{A_j}}$	Elevar al cuadrado $T_{A_j}$ y dividirlo por $N_{A_j}$ para todos los grupos $J$ y sumar los resultados.
$SS_{Total} = \lambda - \psi$ . O, si los efectos del tratamiento han sido computados usar $\sum_j \sum_i (\bar{y}_j - \bar{y})^2 = \sum_j \sum_i \hat{\tau}_j^2 = \sum_j N_j \hat{\tau}_j^2$	Suma de cuadrados entre. Es también llamado $SS_A$ , $SS_{Treatment}$ , ó $SS_{Explicado}$ .
$SS_{Within} = \lambda - \epsilon$	Suma de cuadrados dentro. Llamado también $SS_{Error}$ ó $SS_{Residual}$ .
$MS_{Total} = \frac{SS_{Total}}{(N-1)}$	Media del cuadrado total, lo mismo que $s^2$ , la varianza de muestra.
$MS_{Between} = \frac{SS_{Between}}{(J-1)}$	Media del cuadrado entre, también llamado $MS_A$ , $MS_{Treatment}$ ó $MS_{Explicado}$ .
$MS_{Within} = \frac{SS_{Within}}{(N-J)}$	Media del cuadrado dentro, también llamado $MS_{Error}$ , ó $MS_{Residual}$ .
$F = \frac{MS_{Between}}{MS_{Within}}$	Resultado final a comparar con el valor crítico $F$ a descartar o aceptar $H_0$ con $F_{critica}$ con $\alpha = 0.05$ .

**Tabla 4.8:** Procedimientos matemáticos para el análisis de varianza de ANOVA de un factor.

$$precision = \frac{|VP|}{|VP| + |FP|} \tag{4.1}$$

$$recall = \frac{|VP|}{|VP| + |FN|} \tag{4.2}$$

Así mismo, la ecuación de 4.3 ilustra la correlación final entre la precisión y el recuerdo para evaluar con ANOVA.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.3}$$

**Prueba ANOVA de un factor** En la prueba de ANOVA de un factor es necesario definir operaciones de procedimiento paso a paso que serán seguidos para hacer el análisis de varianza. La tabla 4.8 ilustra las fórmulas utilizadas paso a paso. Finalmente, la plantilla de la tabla 4.9 ilustra los datos que finalmente serán organizados y visualizados para el análisis de contraste de las hipótesis  $H_0$  y  $H_1$ .

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Fuente	SS	Grados	Media de cuadrados	F
$F_1$ Score	$SS_{Between}$	$J - 1$	$\frac{SS_{Between}}{J-1}$	$\frac{MS_{Between}}{MS_{Within}}$
Error / Residual	$SS_{Within}$	$N - J$	$\frac{SS_{Within}}{N-J}$	
Total	$SS_{Total}$	$N - 1$	$\frac{SS_{Total}}{N-1}$	

**Tabla 4.9:** Plantilla de datos de ANOVA de un factor

	AVISPA 1.0	AVISPA 2.0
MP-1	$F_{1(AVISPA\_1.0,MP-1)}$ ó $Precision_{(AVISPA\_1.0,MP-1)}$ ó $Recall_{(AVISPA\_1.0,MP-1)}$	$F_{1(AVISPA\_2.0,MP-1)}$ ó $Precision_{(AVISPA\_2.0,MP-1)}$ ó $Recall_{(AVISPA\_2.0,MP-1)}$
⋮	⋮	⋮
MP-18	$F_{1(AVISPA\_1.0,MP-18)}$ ó $Precision_{(AVISPA\_1.0,MP-18)}$ ó $Recall_{(AVISPA\_1.0,MP-18)}$	$F_{1(AVISPA\_2.0,MP-18)}$ ó $Precision_{(AVISPA\_2.0,MP-18)}$ ó $Recall_{(AVISPA\_2.0,MP-18)}$

**Tabla 4.10:** Grupos vs Modelos de proceso

### 4.6.4.2. Gestión de la aleatoriedad (matriz de grupos vs modelos de proceso)

La tabla 4.10 ilustra la manera cómo se organizaran los grupos con respecto a los modelos de proceso obtenidos.

Cada uno de los modelos a proceso evaluados con AVISPA en orden de realizar el análisis de varianza con ANOVA será descrito a continuación:

- MP-1.** OpenUP
- MP-2.** Ejemplo variabilidad con todos los casos
- MP-3.** APF
- MP-4.** UP-VSE [71]
- MP-5.** Scrum
- MP-6.** XP
- MP-7.** Tutelkan
- MP-8.** Mobius
- MP-9.** Rhiscom
- MP-10.** BBR
- MP-11.** OpenUP - SPL Family
- MP-12.** EPF Practices
- MP-13.** Modelo de proceso DTS

**MP-14.** OpenUP / Desarrollo de Software Dirigido por Modelos - DSDM 1.0

**MP-15.** ISPW 6 [88]

**MP-16.** REuse and Migration of Legacy Applications to Interoperable Cloud Services -REMICS

**MP-17.** PRODIGIA [34]

**MP-18.** OpenUP / Basic

#### **4.6.4.3. Modelo de Proceso de Ejemplo de Variabilidad con casos básicos (MP-2)**

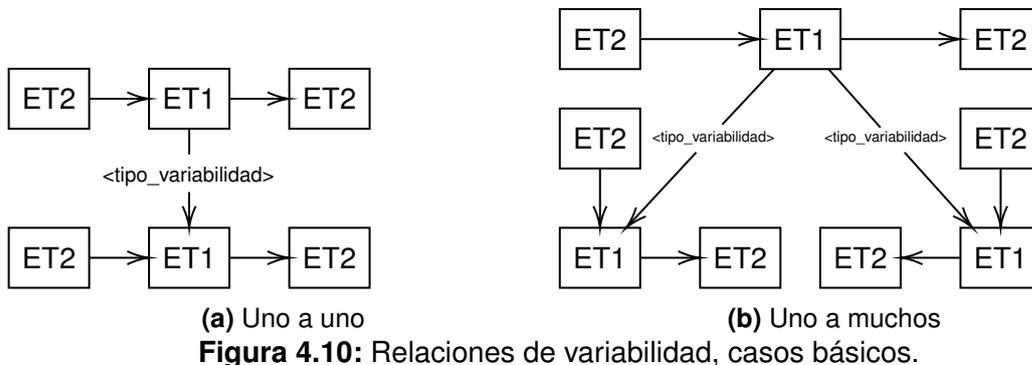
La justificación para la realización de este modelo de proceso, fuera de la industria de software, consiste en la necesidad de depuración de [AVISPA](#) y las reglas de resolución de variabilidad de [SPEM 2.0](#).

Esta sección está dedicada a explicar el modelo de proceso de software que cubre todos los casos de variabilidad de software que son soportados por [AVISPA](#), para lo cual se han tomado los siguientes elementos a consideración para su elaboración respectiva:

- Los elementos de proceso que se incluyen en la resolución de variabilidad son *tareas, roles y productos de trabajo*. Las actividades quedan fuera ya que la nueva versión de AVISPA no soporta la resolución de variabilidad de las mismas.
- Los elementos de proceso que soportan variabilidad se componen de una a dos relaciones del mismo tipo de variabilidad con otros elementos de proceso del mismo tipo.
- Cada elemento afectado por una relación de variabilidad, contiene una relación de entrada y otra de salida diferentes de la misma relación de variabilidad con el objetivo de verificar el comportamiento de AVISPA y la resolución de relaciones de los elementos de proceso variables que se afectan por la resolución de variabilidad.

La figura [4.10](#) ilustra los casos básicos de variabilidad para la construcción del modelo de proceso **MP-2** para la evaluación de AVISPA teniendo en cuenta los siguientes aspectos:

- Para depurar el caso más básico de variabilidad, se ha generado un único caso por cada tipo de elemento de proceso que soporta variabilidad (roles, tareas y productos de trabajo).
- Así mismo, en cada uno de los elementos de proceso que soportan variabilidad se han generado dos clases de relación: de uno a uno (ver figura [4.10a](#)) y de uno a muchos (ver figura [4.10b](#)).



- En cada tipo de relación se incluyen otros tipos de elementos de proceso que sirven como entrada y/o salida de los elementos de proceso con relaciones de variabilidad para evaluar las relaciones subsecuentes de los elementos **ET1**. Así, en la figura 4.10, los **ET1** son los elementos de proceso un único tipo afectados por un único tipo de variabilidad, mientras que los **ET2** son elementos cuyas relaciones pueden ser de entrada o salida de un elemento **ET1**. De forma más concreta, los elementos **ET1** pueden ser tareas, mientras los **ET2** son productos de trabajo.

Este modelo de proceso ha sido definido también como insumo de evaluación de las demás evaluaciones del presente capítulo de este trabajo.

#### 4.6.5. Ejecución

Para esta sección de la evaluación, se tomaron en cuenta los siguientes elementos de búsqueda de datos:

- Características sobre visualizaciones:** Añadirle algunas características sobre las visualizaciones permitieron un mejor estudio en **AVISPA** tal como soportar arrastre de todo el plano o patrón de proceso que no afecta de ninguna manera los resultados.
- Reordenamiento de visualizaciones:** el enfoque general de las calificaciones de cada una de las vistas permitió definir un rango de clasificación de cada uno de los planos involucrados en la calificación por ANOVA
- Cambios de código:** Cada evaluación determinó un subconjunto de pasos detallados de cada elemento de las visualizaciones. Durante la evaluación se realizaron mejoras a las visualizaciones, principalmente sus colecciones. Para lo anterior se volvieron a ejecutar las evaluaciones de F-Score de cada uno de los planos y patrones.

	AVISPA 1.0	AVISPA 2.0
MP-1	0.587155963302752	0.947289532850362
MP-2	0.602294455066922	0.893564356435644
MP-3	0.646193771626298	0.886506935687263
MP-4	0.643320363164721	0.857142857142857
MP-5	0.837092731829574	0.913319238900634
MP-6	0.743155149934811	0.893877551020408
MP-7	0.63677130044843	0.884404474665497
MP-8	0.621739130434783	0.874669487043892
MP-9	0.636609558160505	0.875834445927904
MP-10	0.637647058823529	0.872088271352677
MP-11	0.59709911361805	0.942387840156901
MP-12	0.636048526863085	0.883747831116252
MP-13	0.633555420956996	0.880942364324779
MP-14	0.587155963302752	0.947289532850362
MP-15	0.675409836065574	0.854320987654321
MP-16	0.630728513720574	0.889279094722466
MP-17	0.632575757575758	0.884287454323995
MP-18	0.610619469026549	0.846481876332623

**Tabla 4.11:** Resultados de  $F_1Score$

#### 4.6.5.1. Resultados

Como se ha indicado en la sección 4.6.4.2, en la gestión de la aleatoriedad se organiza las calificaciones de  $F_1Score$  que involucra la precisión y el recuerdo de los clasificadores binarios AVISPA 1.0 y AVISPA 2.0 por medio de 18 modelos de proceso identificados desde MP-1 a MP-18. A continuación se presenta la tabla 4.11 que contiene los resultados de la investigación con la metodología de análisis de varianza ANOVA cuya plantilla se encuentra en la tabla 4.10.

Así mismo, las calificaciones de precisión y recuerdo han sido organizadas utilizando la misma plantilla de la tabla 4.10. Las tablas 4.12 y 4.13 contienen los resultados de la precisión y el recuerdo de cada modelo de proceso por cada clasificador evaluado.

La ejecución de ANOVA dentro de la evaluación de AVISPA finalmente se han obtenido los resultados documentados en la tabla 4.14.

#### 4.6.6. Análisis

Tas una revisión adecuada de los resultados y revisar su consistencia respectiva, se procede a calcular los valores  $F$  y  $F-Critico$  que son los resultados que finalmente determinan la dirección de la investigación.

Para hallar el valor  $F-Critico$  se compara la tabla de distribución F con los grados de libertad del numerador (1) y del denominador (34) con un área de 0.05 hacia la

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

	AVISPA 1.0	AVISPA 2.0
MP-1	0.571892770105605	0.942814520139234
MP-2	0.589887640449438	0.888068880688807
MP-3	0.647313691507799	0.885390428211587
MP-4	0.634271099744246	0.858585858585859
MP-5	0.856410256410256	0.919148936170213
MP-6	0.762032085561497	0.888438133874239
MP-7	0.638872225554889	0.884598508117595
MP-8	0.626277372262774	0.876059322033898
MP-9	0.637184115523466	0.877005347593583
MP-10	0.625866050808314	0.889908256880734
MP-11	0.574864235841738	0.947264662395269
MP-12	0.624149659863946	0.880184331797235
MP-13	0.624850657108722	0.881313131313131
MP-14	0.571892770105605	0.942814520139234
MP-15	0.695945945945946	0.896373056994819
MP-16	0.630640668523677	0.890582258694803
MP-17	0.676113360323887	0.896296296296296
MP-18	0.636923076923077	0.834033613445378

**Tabla 4.12:** Resultados de *Precision*

	AVISPA 1.0	AVISPA 2.0
MP-1	0.603256212510711	0.951807228915663
MP-2	0.615234375	0.899128268991283
MP-3	0.645077720207254	0.887626262626263
MP-4	0.652631578947368	0.855704697986577
MP-5	0.818627450980392	0.907563025210084
MP-6	0.725190839694657	0.899383983572895
MP-7	0.634684147794994	0.884210526315789
MP-8	0.61726618705036	0.873284054910243
MP-9	0.636036036036036	0.874666666666667
MP-10	0.649880095923261	0.854967948717949
MP-11	0.621123218776194	0.937560975609756
MP-12	0.648409893992933	0.887340301974448
MP-13	0.642506142506142	0.880571909167368
MP-14	0.603256212510711	0.951807228915663
MP-15	0.656050955414013	0.816037735849057
MP-16	0.630816383393703	0.887979738944087
MP-17	0.594306049822064	0.872596153846154
MP-18	0.586402266288952	0.859307359307359

**Tabla 4.13:** Resultados de *Recall*

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Fuente	SS	Grados	Media de cuadrados	F
$F_1$ Score	0,54569	1	0,54569	244,695
Error / Residual	0,07582	34	0,00223	
Total	0,62151	35	0,01775	

**Tabla 4.14:** Evaluación ANOVA de la herramienta AVISPA según la tabla 4.9

Datos	AVISPA 1.0	AVISPA 2.0
Mínimo	0,587155963302752	0,846481876332623
1er Cuartil	0,613399384378608	0,874960726764895
Mediana (2do Cuartil)	0,634801973910041	0,874960726764895
3er Cuartil	0,641902037079423	0,893799252374217
Máximo	0,837092731829574	0,947289532850362

**Tabla 4.15:** Datos del diagrama de cajas de  $F_1$ Score.

derecha (valor  $\alpha$ ).

Así, el valor  $F - Critico$  finalmente es: 4,13001774565202.

Los resultados indican que el valor  $F$  es mayor que  $F - Critico$  por lo que  $H_1$  es la hipótesis que la evaluación ANOVA arroja rechazando finalmente  $H_0$ .

Para más información respecto a los resultados de ANOVA ver el Anexo C.

A partir de los resultados se generaron los gráficos de barras de error (ver figura 4.11) donde se visualizan las diferencias entre las formulas  $F_1$ Score, Precision y Recall y sus respectivas desviaciones estándar.

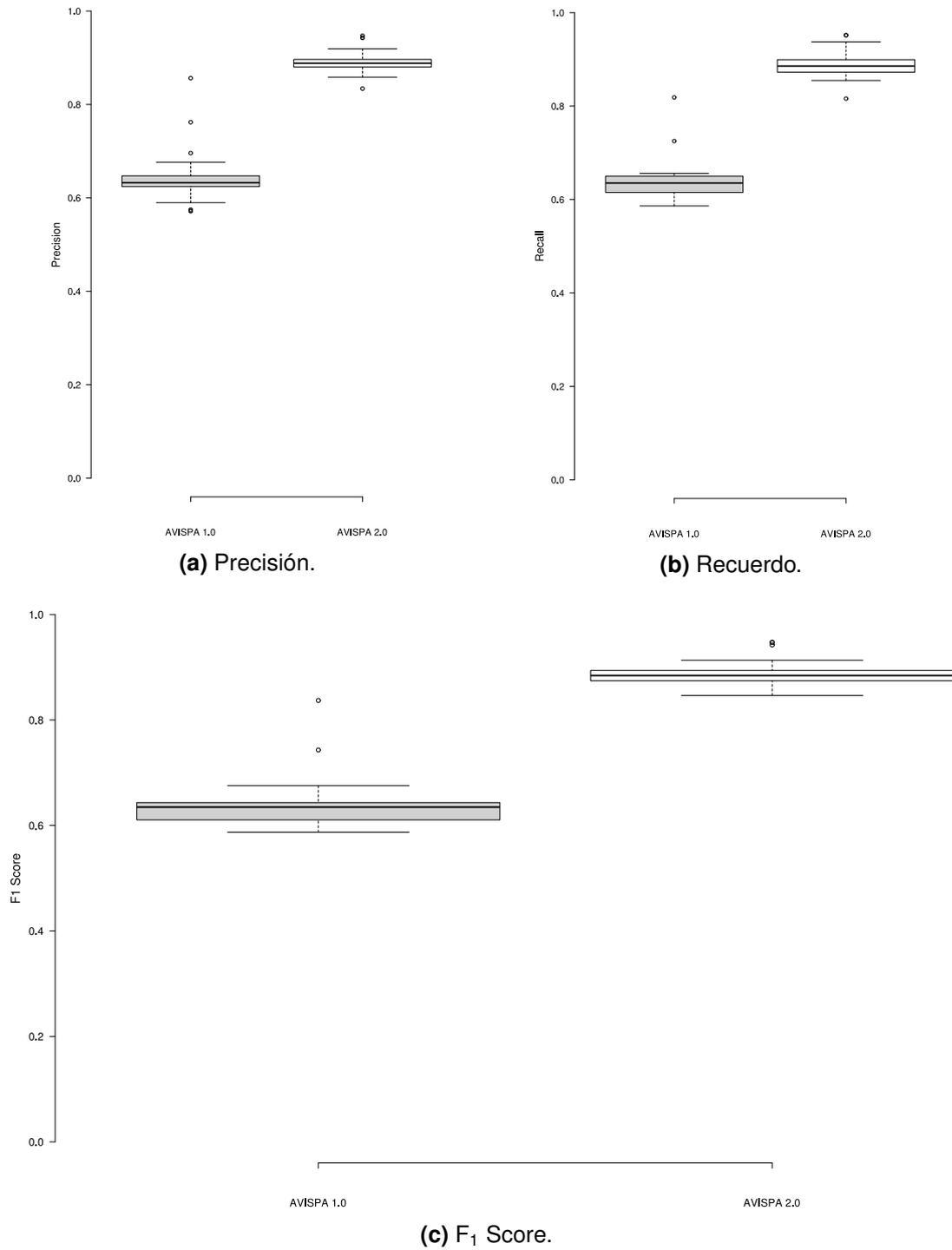
La relación entre clasificadores, tomando el mismo porcentaje de significancia entre precision (50%) y recall (50%) en la fórmula de  $F_1$ Score, y los resultados de distribución de error de la figura 4.11c ilustran que hubo diferencias significativas entre versiones de AVISPA. Así mismo, la distribución estándar en cada métrica indica que las calificaciones de las 3 métricas de AVISPA han tenido una disminución significativa en la dispersión de las mismas por cada modelo de proceso.

Sin embargo, al realizar una observación entre las 3 métricas no existe una diferencia significativa entre puntajes de las métricas, esto quiere decir que el porcentaje de elementos relevantes en la clasificación, así como el número de elementos relevantes seleccionados ha incrementado con un alto nivel de simetría.

De igual manera, los datos de los gráficos también proveen elementos sobre predicciones de clasificación de nuevos elementos por AVISPA. Los elementos de un nuevo modelo de proceso tienen alrededor de un (90%) de probabilidad de ser correctamente clasificados bajo los resultados de 18 elementos (modelos de proceso) como grupo de muestra con baja dispersión de datos (alrededor de un 10%).

Las tablas 4.15, 4.16 y 4.17 ilustran los datos tomados para la generación de los diagramas de caja de  $F_1$ Score, Precision y Recall respectivamente.

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual



**Figura 4.11:** Gráficos de cajas.

Datos	AVISPA 1.0	AVISPA 2.0
Mínimo	0,571892770105605	0,834033613445378
1er Cuartil	0,62432490917514	0,880466531676209
Mediana (2do Cuartil)	0,632455884133962	0,888253507281523
3er Cuartil	0,645203325019572	0,896353866820188
Máximo	0,856410256410256	0,947264662395269

**Tabla 4.16:** Datos del diagrama de cajas de *Precision*.

Datos	AVISPA 1.0	AVISPA 2.0
Mínimo	0,586402266288952	0,816037735849057
1er Cuartil	0,61574232801259	0,872768129112176
Mediana (2do Cuartil)	0,635360091915515	0,885775414145119
3er Cuartil	0,649512545440679	0,899320054927492
Máximo	0,818627450980392	0,951807228915663

**Tabla 4.17:** Datos del diagrama de cajas de *Recall*.

#### 4.6.7. Diseminación y toma de decisiones

##### 4.6.7.1. Estrategia de búsqueda de inconsistencias de clasificación

En el orden de establecer un protocolo de búsqueda de errores de clasificación tanto de [AVISPA 1.0](#) como de [AVISPA 2.0](#) se ilustran los puntos estratégicos definidos en orden de verificar adecuadamente las importaciones y visualizaciones respectivas:

1. **Métodos de las importaciones y resolucionador de relaciones:** Los métodos que importan los modelos de proceso incorporan la información de los archivos XML a estructuras de datos entendibles para [AVISPA](#). Además, también existen métodos que resuelven las relaciones entre las instancias de las estructuras de datos. Estos dos tipos de método representan la parte más importante de la clasificación de los elementos de proceso de software en [AVISPA](#).
2. **Elementos de las visualizaciones:** Los planos y patrones deben estar correctamente asociados a las métricas y, especialmente, a los mecanismos y decisiones de resaltado que se realizan en [AVISPA](#), así, es un foco de alta importancia en la clasificación adecuada de elementos de proceso de la presente herramienta de visualización de modelos de proceso.
3. **Plugin de método:** El plugin de método, directamente desde [EPF](#) contiene todos los elementos, relaciones y resoluciones de variabilidad ejecutadas correspondientes para verificar si las visualizaciones de [AVISPA](#) efectivamente corresponden al plugin de método.
4. **XML exportado:** Cada uno de los elementos exportados obedecen a una organización única del XML exportado de EPF. En la importación de los elementos del XML, la herramienta [AVISPA](#) lee la estructura interna del archivo y construye

las colecciones y, posteriormente, las visualizaciones respectivas. Así, es necesario verificar que las estructuras de elementos del XML estén correctamente importadas.

Así, para depurar la estructura interna de **AVISPA** para los puntos 1 y 2, se utiliza la herramienta de visualización de elementos internos de Moose del modelo de proceso que consiste en una entrada de texto para procesamiento sintáctico de las colecciones. En los puntos donde no es posible tener ésta herramienta, como en el punto 4, para el cual existen herramientas de detención forzosa de la ejecución del flujo de importación como el comando `self halt`.

### 4.6.7.2. **Ámbito de búsqueda de inconsistencias de clasificación**

**Acerca de los elementos de proceso:** Para reconocer el alcance de la búsqueda de inconsistencias de la clasificación, es necesario descubrir el alcance de las métricas relacionadas con la verificación de inconsistencias de clasificadores, en este caso, las formulas 4.1, 4.2 y 4.3.

El análisis de cada una de las éstas ecuaciones revela que, para la evaluación de la clasificación, los elementos involucrados no solamente son los errores hallados, sino también los elementos correctamente clasificados ignorando los verdaderos negativos según las fórmulas de F-Score. Este ámbito también pertenece entonces al ámbito de la búsqueda.

**Acerca de los planos / patrones a evaluar:** Así mismo y, sumado al actual ámbito encontrado de las métricas, es necesario especificar adecuadamente el ámbito de los planos presentes en **AVISPA**. A pesar de la aparente redundancia de esta acción, es necesario identificar los planos y patrones que finalmente han sido seleccionados de este estudio de los planos y patrones que se ofrecen como información adicional y/o poco relevante para la toma de decisiones de los ingenieros de proceso, o que son inestables en la presentación de la información. Así, los planos seleccionados para la presente evaluación son:

- **AVISPA 1.0:**
  - *Roles:*
    - Isolated Role Pattern
    - Overload Role Pattern
    - Role Blueprint
  - *Artefactos:*
    - Workproduct Blueprint
    - Workproduct Guidance Blueprint
    - Workproduct Waste Pattern
    - Workproduct Without Roles

- *Tareas:*
  - Independent Subproject Pattern
  - Multipurpose Tasks Pattern
  - Task Blueprint
  - Task Guidelines Pattern
  - Task Variability Blueprint

■ **AVISPA 2.0:**

- *Roles:*
  - Isolated Role Pattern
  - Overload Role Pattern
  - Role Blueprint
  - Role Variability Blueprint
- *Artefactos:*
  - Artifact Blueprint
  - Artifact Variability Blueprint
  - Artifact Responsible Roles
  - Demanded Artifacts Pattern
  - Independent Suproject Pattern
- *Tareas:*
  - Task Blueprint
  - Task Method Content vs Process
  - Task Variability Blueprint
  - Independent Subproject Pattern
  - Multipurpose Tasks Pattern
- *Actividades:*
  - Activities Coupling Factor by Contained Activities
  - Activities Coupling Factor by Contained Tasks

**Acerca de las estructuras de datos relevantes en la clasificación:** Finalmente, existen ciertas estructuras de datos que contienen y organizan toda la información relevante sobre la cual se basan los planos / patrones para generar sus visualizaciones. Estas estructuras relevantes para la visualización son:

- **Colecciones:** Las colecciones son los elementos principales para las visualizaciones de los elementos de proceso. Son el principal foco de visualización de los elementos de proceso. También pueden verificarse las relaciones entre los elementos de las visualizaciones de **AVISPA** por medio de su contenido.

- **Atributos:** A nivel de clase, los atributos consisten tipos de dato, primitivos o compuestos, que contienen información atómica de una clase. Específicamente, existen atributos en las clases más específicas que representan un único elemento de proceso en [AVISPA](#) y cuyos atributos contienen información relevante para el análisis de las visualizaciones que no están siendo llenadas correctamente en versiones anteriores de AVISPA, principalmente sobre la resolución de la variabilidad.
- **Relaciones de variabilidad:** Para la variabilidad, existen tipos reservados instanciados del reporte de [SPEM 2.0](#) en [AVISPA 2.0](#) que permite la trazabilidad bidireccional de elementos **EB** y **ES** que sirven como elementos cruciales para la resolución de algunas relaciones de variabilidad en [AVISPA](#).

Al definir estos 3 tipos de elementos, todas las clasificaciones finalmente se reducen a su respectivo análisis por cada plano / patrón de [AVISPA](#).

#### 4.6.7.3. Plantilla de evaluación

**Descripción:** La plantilla de evaluación de clasificadores se ha organizado obedeciendo los criterios de evaluación previamente especificados. Así, la plantilla de hoja de cálculo se compone de la siguiente estructura de pestañas:

- **Principal:** Es la vista principal de la evaluación, contiene una tabla con la estructura de pestañas, fecha, responsable, modelo de proceso evaluado y si ha sido usado [AVISPA 1.0](#) o [2.0](#). También contiene una tabla con la calificación F-Score final de cada uno de los modelos de proceso por cada uno de los clasificadores. Así mismo contiene una clasificación de planos y patrones relevantes para la evaluación. Por último, contiene una tabla del método ANOVA instanciado las clasificaciones F-Score y un contador de inconsistencias de todos los elementos de proceso.
- **Gráficos:** Pestaña que contiene datos de relevancia para la generación de los diagramas de caja de la figura [4.11](#).
- **Pestañas de calificación de cada clasificador por modelo de proceso:** Contienen la calificación de métricas de precisión, recuerdo, F-Score, número de falsos positivos, número de positivos falsos, número de verdaderos positivos y un contador de inconsistencias que verifica la igualdad de los elementos clasificados por patrón / plano entre clasificadores y por tipo de elementos de proceso.
- **Ejemplo de aplicación:** Contiene un ejemplo de aplicación de ANOVA según la fuente documental oficial de Libreoffice Calc.

**Verificadores de consistencia de la plantilla de evaluación:** La justificación para incluir el contador de inconsistencias es que, por cada plano y tipo de elemento de proceso, las clasificaciones entre verdaderos negativos y todos los demás elementos

clasificados tendrán la misma cantidad de elementos de proceso, por lo que se incluyó éste pequeño verificador que ayuda a que las métricas traten de clasificarse de manera consistente donde todos los valores de los contadores de inconsistencias deberían estar en “Si” o 0 (cero) a excepción del tercer verificador que cuenta el número total de verdaderos positivos, falsos positivos y falsos negativos. Los verificadores se describen a continuación:

**V1.** Verificador de consistencia a nivel del plano / patrón por tipo de elemento de proceso. El número de elementos de proceso deben ser los mismos en cada uno de los planos por tipo de elemento de proceso. Esto es:

$$|EP_1| = |EP_2| = \dots = |EP_i| = \dots = |EP_N|, \forall EP \exists! T \wedge \exists! C : T \in EP \wedge C_j \in EP \wedge T = (Tareas \vee Roles \vee Artefactos \vee Actividades) \wedge EP_i = FNP_i \cup VPP_i \cup FPP_i \wedge C = (AVISPA_{1.0} \vee AVISPA_{2.0})$$

**V1.** Verificador de consistencia a nivel de clasificadores por tipo de elemento de proceso. El número de elementos de proceso deben ser los mismos en cada clasificador de todos los planos por tipo de elemento de proceso. Esto es:

$$|EP_{1,1}| = |EP_{2,1}| = |EP_{1,2}| = |EP_{2,2}| = |EP_{3,1}| = \dots = |EP_{i,j}| = \dots = |EP_{N,M}|, \forall EP \exists! T \wedge \exists! C_j : T \in EP \wedge C_j \in EP \wedge T = (Tareas \vee Roles \vee Artefactos \vee Actividades) \wedge EP_{i,j} = (FNP_i \in C_j) \cup (VPP_i \in C_j) \cup (FPP_i \in C_j) \wedge C = \{AVISPA_{1.0}, AVISPA_{2.0}\}$$

**V3.** Contadores  $|EP|$ . Consiste en sumar los valores absolutos de los elementos de verdaderos positivos, falsos negativos y falsos positivos por cada patrón del clasificador para la construcción de los validadores **V1** y **V2**.

A continuación se especifican las variables de cada una de las verificaciones:

$EP_i$ . Elementos del  $i$ -ésimo proceso.

$T_i$ . Tipo de elementos del  $i$ -ésimo proceso.

$C_i$ . Clasificador del  $i$ -ésimo proceso.

$FNP_i$ . Falsos negativos del  $i$ -ésimo proceso.

$VPP_i$ . Verdaderos positivos del  $i$ -ésimo proceso.

$FPP_i$ . Falsos positivos del  $i$ -ésimo proceso.

$EP_{i,j}$ . Elementos del  $i$ -ésimo proceso del  $j$ -ésimo clasificador

## 4.7. Lecciones aprendidas

### 4.7.1. Evaluación 1

- La evaluación de modelos de procesos tan extensos como [OpenUP](#) y Tutelkan-TRP no resultó ser una aproximación adecuada para el acercamiento de los

ingenieros de proceso a la herramienta [AVISPA](#). Esto debido a que los participantes no conocieron de antemano la elaboración del proceso por lo que [AVISPA](#) generó planos y patrones de error con mayor abundancia de información lo que generó mayor complejidad en la comprensión de los elementos de los modelos de proceso a evaluar. En efecto, es preciso definir modelos de proceso más pequeños con el ánimo de reducir la carga cognitiva y el ruido visual de los planos y patrones generados por [AVISPA](#).

- Debido a lo anterior, una conclusión de la presente evaluación consiste en que es necesario conocer el modelo de proceso de antemano para lograr una mejor comprensión de los modelos de proceso, puesto que el usuario de la aplicación puede tener la posibilidad de contrastar que la información presentada por [AVISPA](#) corresponde con la información que el usuario tiene del modelo de proceso. Por lo tanto, un experto del proceso creado será quien acompañe el mapa del modelo de proceso para que ilustre rápidamente el contenido del mismo.
- Tras un estudio más profundo de los modelos de proceso evaluados y de las representaciones de [AVISPA](#), los modelos de proceso que hacen un uso más extenso del metamodelo [SPEM 2.0](#) por lo general ofrecen visualizaciones más modulares y, por lo tanto, más entendibles tanto en la documentación en [EPF](#) como en [AVISPA](#).

### 4.7.2. Evaluación 2

- Los modelos de proceso más pequeños resultaron una mejor aproximación para la evaluación los patrones / visualizaciones debido a que no se requiere un gran esfuerzo para su documentación así como su respectiva evaluación, lo cual permitió definir evaluaciones más completas para depurar la herramienta [AVISPA](#).
- Dentro de la misma evaluación, se descubrieron elementos que no permitían una adecuada visualización en el patrón de roles responsables, la resolución de referencias ha sido corregida y fue detectada gracias a ésta evaluación.

### 4.7.3. Evaluación 3

- Las pruebas unitarias permitieron realizar una depuración más profunda de todos los elementos de proceso involucrados en la resolución de relaciones y visualizaciones de [AVISPA](#).
- Las pruebas unitarias se posesionan como un intento de conservar la funcionalidad y finalidad adecuadas de los patrones de error definidos en [AVISPA](#).
- Las pruebas unitarias también se han implementado como método para entender a nivel de depuración las visualizaciones de [AVISPA](#), donde se definen los elementos y sus respectivos colores y como se pueden obtener de los elementos ya visualizados en los planos / patrones.

#### 4.7.4. Evaluación 4

- Cada uno de los elementos evaluados por medio de casos de uso permitió definir un entendimiento adecuado de los elementos de software involucrados en la resolución de relaciones de [AVISPA](#).
- Los casos de uso no proveen mayor información de los componentes internos de software, sin embargo, proveen amplia información acerca de los elementos de negocio y sus flujos para ser programados adecuadamente.
- Así mismo, para el caso de [AVISPA](#), parece ser una buena elección al momento de tratar de evaluar la resolución de relaciones de variabilidad para cualquier plano de la herramienta de análisis visual de procesos software, ya que provee elementos de fácil comprensión de cada una de las relaciones involucradas.

#### 4.7.5. Evaluación 5

- Las validaciones de cada una de las evaluaciones que se han llevado a cabo respecto a la evaluación de procesos de software han servido como puntos de verificación sobre el total de elementos de proceso evaluados incluyendo elementos duplicados dentro de las colecciones que se colocan como falsos positivos dentro de la evaluación.
- Los elementos duplicados o mal clasificados por [AVISPA](#), sencillamente pasan de estar clasificados de una sección a otra dentro de las posibilidades de clasificadores binarios (VP, VN, FP, FN). Así, los elementos clasificados **deben ser los mismos en cada plano de un solo tipo de elemento de proceso**.
- La plantilla de errores de [AVISPA](#) no ofrece los suficientes detalles para la evaluación con ANOVA ya que los errores de clasificación en la plantilla inicial son más cualitativos, omitiendo detalles de las clasificaciones de los clasificadores binarios, sin contar que, clasificar los errores de esta manera hace que sea más difícil y desordenado al momento de documentar los errores. Así que se omiten los detalles cualitativos, mejorando la organización agrupando errores de clasificación por planos / patrones y disminuyendo la entropía de documentar errores de manera cualitativa.

#### 4.7.6. Heurísticas de documentación de modelos de proceso en EPF orientados a [AVISPA](#)

En la tabla [4.18](#) se proponen algunas heurísticas para la elaboración de modelos de proceso en [EPF](#) que pretenden ser utilizados con la herramienta [AVISPA](#).

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Heurística	Descripción
Inicio desde la configuración del modelo de proceso	Los elementos de proceso que no están documentados bajo una configuración específica no pueden ser visualizados para ser usados patrones de capacidad y menos en procesos de entrega. Es preferible que los modelos de proceso que apenas inician la documentación inicien realizando una nueva configuración. Si es un modelo de proceso existente es necesario que todos los elementos de proceso se documenten bajo al menos una configuración.
Organización del trabajo	La documentación de un modelo de proceso EPF puede ser mejor representada en AVISPA si se ordenan los elementos de trabajo de manera modular. Una de las prácticas que más ha funcionado es documentar librería de método por actividad, sin importar si existen elementos de proceso que se usan en múltiples actividades. Esto mejora el entendimiento del proceso documentado.
Uso adecuado de los componentes de SPEM 2.0	La documentación del modelo de proceso en EPF debería contener 4 componentes esenciales: librerías de método (con los elementos de proceso más atómicos, como tareas, roles, productos de trabajo, guías etc.), patrones de capacidad, configuraciones y procesos de entrega. Una configuración adecuada de la documentación del proceso puede permitir mejores visualizaciones y ordenamiento de los elementos del modelo en AVISPA.
Diseño del modelo de proceso centrado en los procesos de entrega	Los procesos de entrega, como componente de SPEM 2.0 son elementos que presentan una descripción completa de todos los elementos de procesos ordenados. Así, los procesos de entrega sirven para tener una visualización de todos los elementos de proceso en una vista de gestión que ilustra en orden de composición dichos elementos. En AVISPA esto significa el estudio de los patrones y procesos de entrega mientras se visualizan los elementos de proceso que pertenecen únicamente al elemento padre, es decir, que en AVISPA es posible obtener información parcial del orden de ejecución de los procesos documentados esencial para la comprensión tanto de AVISPA como del modelo de proceso.
Elementos de proceso en composición representados usando patrones de capacidad	Como es sabido, los patrones de capacidad pueden mutar en 3 tipos de elementos: actividades, iteraciones y fases. Para obtener consistencia entre lo que se documenta y lo que se expone es necesario que estos tipos de elementos de proceso sean representados con patrones de capacidad.
Mismo contenido de método, mismos procesos de entrega	Como es sabido, los patrones de capacidad pueden mutar en 3 tipos de elementos: actividades, iteraciones y fases. Para obtener consistencia entre lo que se documenta y lo que se expone es necesario que estos tipos de elementos de proceso sean representados con patrones de capacidad. Si un patrón de capacidad necesita, o es necesitado por otro patrón de capacidad, o necesita elementos documentados las librerías de método, entonces es necesario extenderlo (recomendado) o copiarlo (no recomendado), no documentar elementos nuevos dentro de los patrones ni dentro de los procesos de entrega (A excepción de los hitos que sí se pueden ser creados).

**Tabla 4.18:** Heurísticas de documentación de modelos de proceso en EPF orientados a AVISPA.

#### **4.7.7. Heurísticas de evaluación / seguimiento de planos y patrones en AVISPA**

En la tabla 4.19 se proponen algunas heurísticas para la evaluación de planos y patrones AVISPA desde la importación de todos los elementos de proceso, pasando por la resolución de sus relaciones, hasta las visualizaciones de planos y patrones de AVISPA.

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

Heurística	Descripción
Proceso mínimos de cobertura completa	Los procesos reales son demasiado complejos y dispendiosos de analizar y algunos no exploran todo el potencial de los patrones. Crear procesos mínimos realizados con el propósito de explorar todas las variantes que los patrones, pueden ilustrar es una aproximación adecuada para la corrección respectiva de los algoritmos responsables de la visualización adecuada que no están correctamente programados, ya que otras maneras de documentación, y por tanto, de la representación en el XML, que pueden presentar otros modelos de proceso se materializan menos que los casos más comunes y naturales de documentación, logrando así aislarlos en casos especiales para un tratamiento más adecuado.
Costos de procesamiento de modelos de proceso	Para la creación de los procesos mínimos de cobertura en <a href="#">AVISPA</a> , los elementos de proceso debe tener un propósito debidamente justificado para la implementación de las visualizaciones. Esto con el objetivo de reducir los costos de creación y seguimiento de los elementos de proceso en el código de <a href="#">AVISPA</a> , sobrecarga de la plataforma Pharo para abrir, editar y procesar modelos de proceso demasiado grandes en las pruebas, entre otros.
Experiencia en sintaxis Smalltalk y de depuración en Pharo Moose	La poderosa sintaxis orientada a objetos de Smalltalk y las funcionalidades implementadas en Pharo permite un acceso a cada objeto instanciado del metamodelo de software FAMIX de Pharo. Así, los cada uno de los elementos del modelo importado, de sus relaciones, y de las visualizaciones realizadas en Mondrian son de fácil acceso. Además, que permiten la edición y actualización de código en tiempo de ejecución. Sin embargo, es necesario conocer los espacios del entorno de Moose donde se pueden depurar dichos elementos. Conocer esto es esencial para lograr calidad en las visualizaciones e importaciones de <a href="#">AVISPA</a> .
Pruebas de depuración	Las pruebas de depuración de código de software permiten un seguimiento detallado de los tipos de dato y sus contenidos en tiempo de ejecución, las pruebas de caja blanca permiten el seguimiento a profundidad del código de software.
Pruebas de consistencia	Las pruebas de consistencia permiten determinar que las salidas generadas del código son realmente las salidas esperadas del mismo. Las pruebas unitarias permiten determinar precisamente esto. Para realizar una prueba unitaria adecuadamente, por organización se genera una clase con el elemento de proceso a probar. Luego, se genera un grupo de protocolo dentro de la clase de pruebas creada para organizar en él todas las funciones correspondientes a la evaluación de un único plano o patrón de <a href="#">AVISPA</a> , por último, se generan 3 tipos de métodos: (i) el método que retorna un texto en formato XML que es el archivo XML generado por EPF, (ii) los métodos que evalúan la importación de elementos de proceso del modelo y (iii) los métodos que evalúan la visualización de elementos del plano o patrón en relación con los elementos del plano / patrón importado.
Trazabilidad de los elementos de proceso	Es necesario verificar en todo momento que los elementos de proceso deben corresponder con la clasificación de objetos generados en <a href="#">AVISPA</a> . Así, es posible asegurar que los elementos de proceso están correctamente importados y clasificados en los planos y patrones de error.

**Tabla 4.19:** Heurísticas de evaluación / seguimiento de planos y patrones en [AVISPA](#).

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

## Capítulo 5

# Conclusiones, limitaciones, trabajo futuro y aportes adicionales

Esta tesis presenta una propuesta de refactorización y mejora de clasificación de elementos de modelos de proceso de [AVISPA](#) por medio del análisis de su estructura interna y de las métricas asociadas.

Para verificar las mejoras realizadas en [AVISPA](#), el trabajo utiliza una combinación de la técnica estadística  $F_1$ -Score para evaluar la eficacia de clasificadores binarios y el análisis de varianza ANOVA que verifica la correlación entre 2 variables (independiente y dependiente) basándose en la distribución F como parte del contraste de hipótesis realizadas en el diseño de la investigación.

Para la fase de validación de la investigación, se definieron diversas evaluaciones de los contenidos de la herramienta [AVISPA](#). De todas las anteriores, la evaluación 5 (ver sección 4.6), que consiste en la evaluación de ANOVA, se realizó utilizando diseño de experimentos.

Cada una de las evaluaciones realizadas sobre la herramienta definió elementos críticos de búsqueda de errores de clasificación de elementos de proceso de [AVISPA](#). Lo que permitió hallar errores de una manera sistemática y ordenada de cada uno de los planos y patrones de software.

Las relaciones de variabilidad de [AVISPA](#) han sido verificadas y corregidas gracias a cada una de las evaluaciones realizadas, lo cual permitió un mejor ordenamiento de los planos y patrones. Así mismo, las correcciones y relaciones de variabilidad se han extendido hacia todos los demás planos y patrones de error de [AVISPA](#) lo que permitió una disminución de errores de clasificación por parte de la herramienta.

Cada uno de los modelos de proceso de software creado en [SPEM 2.0](#) y evaluado con [AVISPA](#) han sido procesos creados y algunos de ellos probados dentro de la industria de software.

## 5.1. Conclusiones

Las secciones que aportan a la investigación consisten en la revisión sistemática de la literatura y cada una de las evaluaciones realizadas en el capítulo 4.

La revisión sistemática de la literatura permitió encontrar elementos cruciales de direccionamiento de la investigación:

- Definir el subconjunto de métricas de modelos de proceso en [SPEM 2.0](#) que [AVISPA](#) podría incorporar dado el impacto de las mismas en la literatura.
- Definir nuevos planos y / o patrones que ilustren mejor la construcción de los mismos debido a reportes de investigación que sugieren el uso de métricas que aportan a la evaluación visual de los elementos de proceso de software.
- Eliminar planos y / o patrones existentes que, dado el filtro de inclusión de métricas debido a el número de coincidencias en los artículos encontrados, no son relevantes en la construcción de las visualizaciones de [AVISPA](#).

Respecto a cada una de las evaluaciones se han podido detectar algunos elementos esenciales en la ejecución dentro de [AVISPA](#):

- **Evaluación 1 (sección 4.2):**
  - Cada uno de los participantes incluyeron experiencias previas para la definición de planos y patrones de modelos de proceso de [AVISPA](#) para lo cual resaltan la evolución de la misma herramienta. Sin embargo, también recalcan la definición del concepto de patrón de error en el contexto del plano y de la evaluación del modelo de proceso.
  - Algunos usuarios con experiencias previas en [AVISPA](#) tenían una sobreexpectación sobre la nueva versión de la herramienta, sin embargo, los elementos visuales de la herramienta no se ajustan al modelo de experiencias previas que generalmente los usuarios han tenido en la interacción con otros productos software por lo que se resaltan bastante en la evaluación.
- **Evaluación 2 (sección 4.3):**
  - Cada proceso se compone de elementos y relaciones mínimas para evaluar la clasificación de cada uno de los elementos de proceso de software en [SPEM](#). Por lo que la implementación de las visualizaciones de los planos se pueden evaluar más fácilmente a profundidad, permitiendo mejorar el análisis por caja blanca del código de [AVISPA](#).
  - Dentro de las evaluaciones mínimas se evaluaron elementos de variabilidad y relaciones por medio de las colecciones, que han sido los elementos más cruciales para la implementación adecuada de las visualizaciones de software.

■ **Evaluación 3 (sección 4.4):**

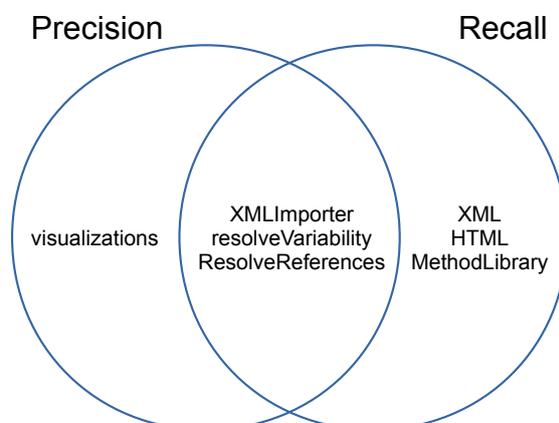
- Las pruebas unitarias permitieron la incorporación de pruebas detalladas de los componentes de software donde se encontraron elementos que definieron una correcta visualización de cada uno de los elementos de proceso basándose en XML de ejemplos cargados en las pruebas y ejecutados conforme al orden establecido de la evaluación.
- El lenguaje Smalltalk y la construcción de la máquina virtual de Pharo permitieron un avance a profundidad de cada uno de los elementos, permitiendo hacer depuraciones en cualquier punto del código con comandos simples para su uso.

■ **Evaluación 4 (sección 4.5):**

- Los casos de uso permitieron determinar con mayor profundidad los casos y las relaciones resultantes de cada uno de los tipos de variabilidad en los que AVISPA debía resolver tanto sus colecciones como las duplicidades
- Esta evaluación permitió corregir las variabilidades, principalmente las relaciones de tipo contribución que no estaban siendo visualizadas correctamente y extender todas las variabilidades a todos los planos de tareas.
- Así mismo, se extendió la funcionalidad de resolución de la variabilidad a todos los demás planos de todos los otros tipos de elementos de proceso que soportan tipos de variabilidad. Y, por lo tanto, disminuir los errores en la representación de los planos y patrones de error de la herramienta.

■ **Evaluación 5 (sección 4.6):**

- La organización de los elementos de proceso en los XML importados principalmente se ve afectada por las diversas versiones de los XML y, por lo tanto, de las versiones de EPF. Por lo tanto, en la evaluación con ANOVA, los elementos de proceso que no están correctamente clasificados, obedecen principalmente a éste problema de versionado de la herramienta de documentación de modelos de proceso de software.
- Los datos obtenidos en esta evaluación para los diagramas de caja (ver figura 4.11) ilustran la distribución de calificaciones de AVISPA. Así, se puede obtener una percepción de la clasificación de elementos de proceso de las diversas versiones de AVISPA. La dispersión de datos disminuyó en la calificación por  $F_1Score$ , por lo tanto, disminuye la entropía de clasificación en AVISPA 2.0
- Los cuartiles 1, 2 y 3 en cada uno de los diagramas indican que las calificaciones de modelos de proceso en industria permiten indicar que, con otros modelos de proceso, existe un 75% de probabilidad de que la calificación en cada una de las métricas ( $F_1Score$ ,  $Precision$  y  $Recall$ ), esté entre un 6% o menos de estar en el respectivo promedio. Esto implica mejora en la clasificación por medio de una reducción significativa de la dispersión de las calificaciones y, por lo tanto, de los elementos de los modelos de proceso en AVISPA.



**Figura 5.1:** Relación de elementos de AVISPA respecto a *Precision* y *Recall*.

- Un detalle importante a resaltar es el área entre los cuartiles de Q1 a Q2 y de Q4 a Q5. Aún sigue conservándose sin cambios significativos entre las dos versiones de AVISPA para las 3 métricas involucradas en la evaluación de ANOVA. Principalmente en la gráfica de *Recall* (ver figura 4.11b) donde hay una distancia mayor de estos cuartiles que en las otras métricas, lo cual indica que es necesario invertir mayor esfuerzos en ésta métrica.
- Según un análisis entre las gráficas de *Precision* y *Recall*, la clasificación de elementos de proceso tiene una leve mejora en la precisión que en el recuerdo.
- La figura 5.1 ilustra la relación entre la precisión y recuerdo de cada una de las funciones más relevantes de AVISPA y cómo influyen en la calificación de éstas métricas.

Las métricas halladas finalmente han sido incorporadas, así como las respectivas visualizaciones. Sin embargo, para que arrojen valores más exactos, es necesario focalizar esfuerzos en otros elementos como las colecciones, la correcta importación y clasificación de los elementos modelos de proceso en AVISPA, entre otros.

Adicionalmente, las métricas, por si mismas, no ofrecen mayor información, ni siquiera sus valores concretos, debido a que no hay forma de contrastarlas con información que ofrezca un soporte verídico de las mismas, por lo que fue necesario mejorar otros elementos de contexto de la herramienta para mejorar la confiabilidad de los datos y de las métricas calculadas.

Así mismo, AVISPA 2.0 ofrece una capa de abstracción de las métricas, al ofrecer visualizaciones más confiables sobre las mismas, y una actualización respecto a la revisión sistemática. Por tanto, la herramienta permite una implementación confiable para la toma de decisiones de los modelos de proceso elaborados en SPEM 2.0 previa a su ejecución en base a la información de los patrones de error.

AVISPA 2.0, en base a las visualizaciones de OpenUP con las que se entrega esta herramienta, puede sugerir un estilo de documentación para la definición de

modelos de proceso en SPEM 2.0, con tal de aprovechar al máximo las bondades y herramientas del metamodelo, principalmente el soporte de relaciones de variabilidad de modelos de proceso.

### 5.2. Limitaciones

Parte de las limitaciones de la investigación consistió en que las bases de datos especializadas en artículos de ingeniería de software, no cuentan con una cantidad adecuada de artículos específicos en el tema de métricas de modelos de proceso en [SPEM 2.0](#). Las métricas halladas en su gran mayoría son tomadas de modelos de proceso generales o de otros metamodelos de proceso de software.

La evaluación exploratoria de [AVISPA](#) no permitió obtener mayores resultados respecto a la comprensión de los modelos de proceso debido a que los usuarios se fijaban principalmente en los detalles de usabilidad de la herramienta como distribución de las opciones de menú. Sin embargo, los resultados incluyeron otros elementos que permitieron encontrar otros detalles de la implementación para abordar elementos que son críticos dentro del análisis de los modelos de proceso de software como la descripción de las leyendas, de las métricas de los nodos, entre otros.

La evaluación 5 (ver sección [4.6](#)) se realizó de manera aislada, es decir, sin participantes que evaluaran cada modelo de proceso respecto a las métricas de evaluación de clasificadores. Esto posiblemente pueda incluir ruido en la clasificación, el protocolo seguido para evaluar la clasificación consistió en la visualización y descarte de elementos del XML, del modelo de proceso exportado a HTML y de las colecciones de Smalltalk, por lo que es necesario definir un proceso más riguroso para establecer la clasificación de elementos de proceso en la matriz de confusión (ver tabla [4.7](#)).

### 5.3. Trabajos futuros

Como base del avance de [AVISPA](#), la inclusión de un árbol de decisiones representaría un avance crucial donde la misma herramienta sugiera modificaciones de los elementos de proceso documentados de [AVISPA](#), pues podría presentar elementos que, a simple vista, no se podrían determinar, apoyando la toma de decisiones de los cambios necesarios en los modelos de proceso.

Así mismo, otros elementos de la evaluación [4.2](#) indican que puede ser necesario tener una tabla donde indiquen los elementos de proceso que se visualizan en el mismo plano.

Así mismo, las pruebas de resolución de variabilidad (ver evaluación [4.5](#)) se realizaron principalmente para cumplir la necesidad de entender las reglas de resolución. Es necesario realizar otra evaluación, preferiblemente de caja blanca, donde se es-

tablezca adecuadamente la resolución de variabilidad de elementos de proceso de SPEM 2.0 que permita confirmar o descartar si están correctamente clasificados en las colecciones y, por lo tanto, en AVISPA.

Como consecuencia de lo anterior, es necesario mejorar los modelos de proceso del estado actual de AVISPA respecto a los resultados hallados con ANOVA. Algunos errores están en la importación de elementos de proceso, principalmente en la resolución de sus relaciones y en el soporte de las diversas versiones de EPF.

Es posible representar más elementos de proceso y sus respectivas relaciones de variabilidad en AVISPA como en el caso de las actividades. Así mismo, es necesario verificar como se realiza la resolución de variabilidad de tipo muchos a muchos para depuración completa de las relaciones de variabilidad.

Volviendo al análisis de los diagramas de caja, al ser la métrica  $F_1Score$  una correlación entre las métricas *Precision* y *Recall*, es necesaria otra refactorización de AVISPA:

- Las visualizaciones, tienen un filtro adicional, que permite seleccionar elementos de proceso que si puedan ser visualizados con AVISPA, principalmente utilizado por resolución de variabilidad. Sin embargo, para la resolución de variabilidad, existen algunos elementos por mejorar. Por ejemplo: (i) resolución de variabilidad de muchos a muchos, (ii) implementación de variabilidad cuando un elemento variable **ES** tiene múltiples elementos base **EB** con múltiples relaciones con diversos tipos de variabilidad. Estas mejoras pueden mejorar drásticamente la representación de los elementos de AVISPA y, por lo tanto,  $F_1Score$ .
- Por lo anterior, es necesario evaluar de nuevo la herramienta con empresas donde pueda ser corroborada su confiabilidad de manera práctica, así como el costo de comprensión de los planos y patrones generados con las características desarrolladas y depuradas del presente trabajo.

### 5.4. Aportes adicionales

Para hallar las métricas que están involucradas en la evaluación de los modelos de proceso se ha realizado una revisión sistemática para el filtrado de las mismas por su nivel de relevancia para la investigación.

El trabajo también propone una serie de heurísticas para la documentación de modelos de proceso en EPF así como para la evaluación de modelos de proceso de AVISPA para que futuras implementaciones y evaluaciones puedan realizarse respecto a planos y patrones de error.

Así mismo, el trabajo propone una serie de reglas de variabilidad abstraídas del informe técnico de la OMG sobre SPEM 2.0 por medio de ejemplos para cualquier clase de tipo de elemento de proceso de software en el metamodelo de proceso.

Se han diseñado y ejecutado diferentes evaluaciones para la corrección de los elementos de [AVISPA](#) en cada una de las visualizaciones de los modelos de proceso. El mayor peso de aportes está en la evaluación de [AVISPA](#), ver capítulo 4. Cada una de las evaluaciones, a excepción de la evaluación [4.6](#), son aportes adicionales fuera del flujo regular de la investigación, las cuales cumplieron con 2 elementos de relevancia para la mejora sustancial de la herramienta:

- Tener soporte de las acciones de edición realizadas sobre la herramienta [AVISPA](#).
- Tener una guía formal de técnicas / procedimientos de prueba, que ya han sido probadas en academia / industria, con resultados exitosos y con guías y documentación adecuadas.

Esto con el fin de evitar realizar correcciones de código sin una organización ni justificación adecuadas. Además del registro adecuado de las experiencias de modificación con el fin de seguir avanzando en la mejora del clasificador [AVISPA](#) en base a los casos más exitosos de corrección de código de [AVISPA](#).

Cada una de las evaluaciones adicionales será descrita a continuación:

- La evaluación [4.2](#) se realizó como una aproximación del estado de [AVISPA](#) 1.0. Así mismo, permitió un mejor entendimiento por parte del investigador de la construcción de la herramienta.
- La evaluación [4.3](#) permitió una depuración a profundidad de los elementos y de los datos de cada sección de una parte específica de código que necesitaba ser depurado con el objetivo de verificar la correcta distribución de elementos de proceso dentro de la colecciones.
- La evaluación [4.4](#) aprovecha las herramientas de Pharo para realizar pruebas unitarias de código de software, permitiendo la evaluación tanto de funciones de back como la misma lógica de front de las visualizaciones.
- La evaluación [4.5](#) permitió definir un set de pruebas de contexto para la resolución de variabilidad donde el fuerte principal fue esclarecer la instanciación de las reglas de variabilidad de [SPEM](#) 2.0 dentro de [AVISPA](#).

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

## Bibliografía

- [1] J. A. H. Alegria, M. C. Bastarrica, and A. Bergel, “Analyzing the Scrum Process Model with AVISPA,” in *Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the*, nov 2010, pp. 60–65. [2](#), [47](#)
- [2] J. A. H. Alegría, A. Lagos, A. Bergel, and M. C. Bastarrica, “New Modeling Concepts for Today’s Software Processes: International Conference on Software Process, ICSP 2010, Paderborn, Germany, July 8-9, 2010. Proceedings,” J. Münch, Y. Yang, and W. Schäfer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. Software P, pp. 273–284. [2](#)
- [3] J. Alvarez and J. Hurtado, “Implementing the software requirements engineering practices of the ISO 29110-5-1-1 standard with the unified process,” *Computing Colombian Conference (9CCC), 2014 9th*, pp. 175–183, 2014. [28](#), [44](#)
- [4] R. CONRADI, M. N. NGUYEN, A. I. WANG, and C. LIU, “Planning support to software process evolution,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 10, no. 01, pp. 31–47, 2000. [2](#)
- [5] J. A. H. Ariel, C. María Bastarrica, and A. Bergel, “Analyzing software process models with AVISPA,” in *Proceedings of the 2011 International Conference on Software and Systems Process*. ACM, 2011, pp. 23–32. [2](#), [28](#), [47](#)
- [6] G. Canfora, F. García, M. Piattini, F. Ruiz, and C. Visaggio, “A family of experiments to validate metrics for software process models,” *Journal of Systems and Software*, vol. 77, no. 2, pp. 113–129, aug 2005. [2](#), [3](#), [40](#), [51](#), [52](#), [55](#)
- [7] A. G. Cass, A. S. Lerner, E. K. McCall, L. J. Osterweil, S. M. Sutton, and A. Wise, “Little-JIL/Juliette: a process definition language and interpreter,” in *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, 2000, pp. 754–757. [2](#)
- [8] A. Cockburn, “Selecting a Project’s Methodology,” *IEEE Softw.*, vol. 17, no. 4, pp. 64–71, 2000.
- [9] R. Conradi, C. Fernström, and A. Fuggetta, “A Conceptual Framework for Evolving Software Processes,” *SIGSOFT Softw. Eng. Notes*, vol. 18, no. 4, pp. 26–35, 1993. [1](#), [2](#)

- [10] P. B. Crosby, *Quality is Free: The Art of Making Quality Certain*, ser. Mentor book. McGraw-Hill, 1979. [1](#)
- [11] W. E. Deming, *Out of the crisis : quality, productivity, and competitive position*. Cambridge [Cambridgeshire]; New York: Cambridge University Press, 1988. [1](#)
- [12] N. Emilio, D. Ferreyra, H. P. Leone, and S. Gonnet, “OVM2PN : configuraciones en Líneas de Productos de Software,” pp. 427–437, 2012. [1](#)
- [13] J. Ge, H. Hu, Q. Gu, and J. Lu, “Modeling Multi-View Software Process with Object Petri Nets,” in *Software Engineering Advances, International Conference on*, oct 2006, p. 41. [2](#)
- [14] V. Gruhn, “Validation and Verification of Software Process Models,” in *Proceedings of the European Symposium on Software Development Environments and CASE Technology*. New York, NY, USA: Springer-Verlag New York, Inc., 1991, pp. 271–286. [2](#), [25](#)
- [15] J. A. Hurtado Alegría, M. C. Bastarrica, A. Quispe, and S. F. Ochoa, “An MDE Approach to Software Process Tailoring,” in *Proceedings of the 2011 International Conference on Software and Systems Process*, ser. ICSSP '11. New York, NY, USA: ACM, 2011, pp. 43–52. [1](#)
- [16] J. A. Hurtado Alegría, M. C. Bastarrica, and A. Bergel, “Avispa: a tool for analyzing software process models,” *Journal of Software: Evolution and Process*, vol. 26, no. 4, pp. 434–450, apr 2014. [23](#), [28](#), [30](#), [40](#), [43](#), [47](#), [49](#)
- [17] I. Jacobson, G. Booch, and J. Rumbaugh, *The unified software development process*. Addison-Wesley Longman Publishing Co., Inc., 1999. [21](#), [22](#)
- [18] J. M. Juran and A. B. Godfrey, *Juran's Quality Handbook*, ser. JURAN'S QUALITY HANDBOOK. McGraw Hill, 1999. [1](#)
- [19] T. Martinez-Ruiz, F. Garcia, M. Piattini, and J. Munch, “Modelling software process variability: an empirical study,” *IET Software*, vol. 5, no. 2, pp. 172–187, 2011. [27](#)
- [20] J. Münch, O. Armbrust, M. Kowalczyk, and M. Soto, “Software Process Definition and Management.” Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Prescripti, pp. 19–77. [22](#)
- [21] Object Management Group (OMG), “SPEM 2.0,” OMG, Tech. Rep., 2008. [xi](#), [24](#), [30](#), [53](#)
- [22] OMG, “Business Process Model and Notation (BPMN),” Object Management Group (OMG), Tech. Rep., 2011. [2](#)
- [23] —, “Business Process Model and Notation (BPMN) 2.0,” Object Management Group (OMG), Tech. Rep., 2013. [2](#)

- [24] —, “Software & Systems Process Engineering Meta-Model Specification,” Object Management Group (OMG), Tech. Rep., 2008. [2](#)
- [25] J. M. Ribó and X. Franch, “PROMENADE: A PML Intended to Enhance Standardization, Expressiveness and Modularity in Software Process Modelling,” Tech. Rep., 2000. [2](#)
- [26] S. L. Pfleeger, “Experimental design and analysis in software engineering,” *Annals of Software Engineering*, vol. 1, no. 1, pp. 219–253, dec 1995. [82](#)
- [27] J. Simmonds, M. Bastarrica, L. Silvestre, and A. Quispe, “Analyzing methodologies and tools for specifying variability in software processes,” *Universidad de Chile, Santiago, Chile*, 2011. [1](#), [2](#)
- [28] I. Sommerville, *Software Engineering*, 9th ed. Addison-Wesley Publishing Company, 2007. [21](#)
- [29] —, *Software Engineering*, 7th ed. Pearson Education, 2004. [21](#)
- [30] C. P. Team, “CMMI for Development, Version 1.3,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-2010-TR-033, 2010. [2](#)
- [31] WfMC, “XML Process Definition Language,” Workflow Management Coalition (WfMC), Tech. Rep., 2001. [2](#)
- [32] M. Wu, G. Li, J. Ying, and H. Yan, “A Metamodel Approach to Software Process Modeling Based on UML Extension,” in *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 6, oct 2006, pp. 4508–4512. [2](#)
- [33] Y. Sasaki *et al.*, “The truth of the f-measure. 2007,” 2007. [21](#)
- [34] “Proceso de Desarrollo Industrial del Grupo de Ingeniería Automática (PRODIGIA).” [88](#)
- [35] “ISO/IEC 15504: Software Process Improvement and Capability Determination,” 2007. [2](#)
- [36] P. B. Petersen, “Total quality management and the Deming approach to quality management,” *Journal of Management History (Archive)*, vol. 5, no. 8, pp. 468–488, 1999. [1](#)
- [37] International Organization for Standardization (ISO), “ISO 18629-1:2004. Industrial automation systems and integration – Process specification language – Part 1: Overview and basic principles,” Tech. Rep. April, 2014. [23](#)
- [38] J. Mendling, G. Neumann, and W. van der Aalst, “On the correlation between process model metrics and errors,” in *Tutorials, Posters, Panels and Industrial Contributions at the 26th International Conference on Conceptual Modeling - Volume 83*, ser. ER '07. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2007, pp. 173–178. [2](#)

- [39] J. A. H. Alegria, "A meta-process for defining adaptable software processes," Ph.D. dissertation, University of Chile, 2012. [3](#)
- [40] M. Lanza and S. Ducasse, "Polymetric views-a lightweight visual approach to reverse engineering," *IEEE Trans. Softw. Eng.*, vol. 29, no. 9, pp. 782–795, Sep. 2003. [3](#)
- [41] M. C. Bastarrica, G. Maturro, R. Robbes, L. Silvestre, and R. Vidal, *How does Quality of Formalized Software Processes Affect Adoption?* Cham: Springer International Publishing, 2014, pp. 226–240. [VII](#), [XIII](#), [3](#), [29](#), [41](#), [48](#), [49](#), [50](#), [52](#), [57](#), [59](#), [63](#)
- [42] L. Osterweil, "Software processes are software too," in *Proceedings of the 9th International Conference on Software Engineering*, ser. ICSE '87. Los Alamitos, CA, USA: IEEE Computer Society Press, 1987, pp. 2–13. [Online]. Available: <http://dl.acm.org/citation.cfm?id=41765.41766> [22](#), [27](#), [44](#)
- [43] F. J. Pino, F. García, and M. Piattini, "Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas," *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 2, pp. 6–23, 2006. [8](#)
- [44] O. Pedreira, F. García, N. Brisaboa, and M. Piattini, "Gamification in software engineering – a systematic mapping," *Information and Software Technology*, vol. 57, pp. 157–168, jan 2015. [4](#)
- [45] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571 – 583, 2007, software Performance. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S016412120600197X> [5](#)
- [46] J. Pavlicek, R. Hronza, P. Pavlickova, and K. Jelinkova, "The business process model quality metrics," in *Enterprise and Organizational Modeling and Simulation*, R. Pergl, R. Lock, E. Babkin, and M. Molhanec, Eds. Cham: Springer International Publishing, 2017, pp. 134–148. [10](#), [37](#), [38](#), [39](#)
- [47] V. Omrani and S. A. R. Ebrahimi, "Software architecture viewpoint models: A short survey," *Advances in Computer Science: an International Journal*, vol. 2, no. 5, pp. 55–62, 2013. [44](#)
- [48] S. Harper, E. Michailidou, and R. Stevens, "Toward a definition of visual complexity as an implicit measure of cognitive load," *ACM Trans. Appl. Percept.*, vol. 6, no. 2, pp. 10:1–10:18, Mar. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1498700.1498704> [45](#)
- [49] T. K. Kim, "Understanding one-way anova using conceptual figures," *Korean journal of anesthesiology*, vol. 70, no. 1, p. 22, 2017. [21](#)
- [50] K. Kluza, G. J. Nalepa, and J. Lisiecki, *Square Complexity Metrics for Business Process Models*. Cham: Springer International Publishing, 2014, pp. 89–107. [Online]. Available: [https://doi.org/10.1007/978-3-319-03677-9\\_6](https://doi.org/10.1007/978-3-319-03677-9_6) [10](#), [37](#), [38](#), [39](#)

- [51] A. del Río-Ortega, M. Resinas, A. Durán, B. Bernárdez, A. Ruiz-Cortés, and M. Toro, "Visual ppinot: A graphical notation for process performance indicators," *Business & Information Systems Engineering*, Jun 2017. [Online]. Available: <https://doi.org/10.1007/s12599-017-0483-3> 10, 37, 38
- [52] J. Geisel, B. Hamid, D. Gonzales, and J.-M. Briel, "Towards a methodological tool support for modeling security-oriented processes," in *Model and Data Engineering*, L. Bellatreche, Ó. Pastor, J. M. Almendros Jiménez, and Y. Aït-Ameur, Eds. Cham: Springer International Publishing, 2016, pp. 31–41. 10, 37, 38
- [53] M. Smatti, M. Oussalah, and M. Ahmed Nacer, "Supporting deviations on software processes: A literature overview," in *Software Technologies*, P. Lorenz, J. Cardoso, L. A. Maciaszek, and M. van Sinderen, Eds. Cham: Springer International Publishing, 2016, pp. 191–209. 10, 37, 39
- [54] G. Kaur and K. B. Bahl, "Software reliability , metrics , reliability improvement using agile process," 2014. 10, 37, 38
- [55] F. Bachmann, L. Bass, and R. Nord, "Modifiability Tactics," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-2007-TR-002, 2007. 27
- [56] P. H. Feiler and W. S. Humphrey, "Software process development and enactment: concepts and definitions," in *Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the*. IEEE, Feb. 1993, pp. 28–40. [Online]. Available: <http://dx.doi.org/10.1109/spcon.1993.236824> 21
- [57] U. Becker-Kornstaedt, "Descriptive software process modeling - how to deal with sensitive process information," *Empirical Software Engineering*, vol. 6, no. 4, pp. 353–367, 2001. [Online]. Available: <http://dx.doi.org/10.1023/A%3A1011986902298> 22
- [58] J. Simmonds, D. Perovich, M. C. Bastarrica, and L. Silvestre, "A megamodel for software process line modeling and evolution," in *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, Sept 2015, pp. 406–415. 23
- [59] V. H. M. Domínguez and M. E. C. Bolaños, "SPEM: Software process engineering metamodel," *Revista Latinoamericana de Ingeniería de Software*, vol. 3, no. 2, p. 92, may 2015. 23
- [60] A. Järvi and T. Mäkilä, "Observations on modeling software processes with spem process components," 2005. 22
- [61] T. Mäkilä and A. Järvi, "Spemmet—a tool for modeling software processes with spem," in *Proceedings of the 9th International Conference on Information Systems Implementation and Modelling, ISIM*, vol. 6, 2006. 22, 24

- [62] E. B. Pereira, R. M. Bastos, T. C. Oliveira, and M. C. Móra, “A set of well-formedness rules to checking the consistency of the software processes based on spem 2.0,” in *Enterprise Information Systems*, R. Zhang, J. Zhang, Z. Zhang, J. Filipe, and J. Cordeiro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 284–299. 25
- [63] I. Ruiz-Rube, J. M. Doderó, M. Palomo-Duarte, M. Ruiz, and D. Gawn, “Uses and applications of spem process models . a systematic mapping study,” 2012. 25
- [64] X. Larrucea and E. Iturbe, “A metamodel integration for metrics and processes correlation.” in *ICSOFT (1)*, 2010, pp. 63–68. 25
- [65] S. Park, K. Choi, K. Yoon, and D. Bae, “Deriving software process simulation model from spem-based software process model,” in *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, Dec 2007, pp. 382–389. 25
- [66] T. Martínez-Ruiz, F. García, and M. Piattini, *Towards a SPEM v2.0 Extension to Define Process Lines Variability Mechanisms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 115–130. 27
- [67] M. Reichert, “Visualizing large business process models: Challenges, techniques, applications,” in *Business Process Management Workshops*, M. La Rosa and P. Soffer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 725–736. 26, 41
- [68] H. Reijers and J. Mendling, “Modularity in process models: Review and effects,” in *Business Process Management*, M. Dumas, M. Reichert, and M.-C. Shan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 20–35. 26
- [69] J. S. Salgado, J. J. Alvarez, and J. A. Hurtado, “Análisis visual de la variabilidad de modelos de procesos de software,” in *Computing Colombian Conference (11CCC), 2016 11th*, 2011. 27, 44, 47, 48, 51, 53, 65, 77
- [70] F. B. Cardenas, J. S. Martinez, M. C. Camacho, and J. A. Hurtado, “Análisis visual de la modularidad de modelos de procesos de software avimo-ps,” in *Computing Colombian Conference (8CCC), 2013 8th*, 2013. 26, 27, 44, 47
- [71] J. Alvarez and J. Hurtado, “Up-vse: A unified process - based lifecycle model for very small entities,” in *Advances in Computing*, J. E. Serrano C. and J. C. Martínez-Santos, Eds. Cham: Springer International Publishing, 2018, pp. 260–275. 28, 44, 87
- [72] A. Burattin, V. Bernstein, M. Neuraüter, P. Soffer, and B. Weber, “Detection and quantification of flow consistency in business process models,” *Software & Systems Modeling*, vol. 17, no. 2, pp. 633–654, May 2018. [Online]. Available: <https://doi.org/10.1007/s10270-017-0576-y> 10, 37, 38, 39
- [73] K. Kluza, “Measuring complexity of business process models integrated with rules,” in *Artificial Intelligence and Soft Computing*, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Cham: Springer International Publishing, 2015, pp. 649–659. 10, 37, 38

- [74] J. Pflug and S. Rinderle-Ma, "Process instance similarity: Potentials, metrics, applications," in *On the Move to Meaningful Internet Systems: OTM 2016 Conferences*, C. Debruyne, H. Panetto, R. Meersman, T. Dillon, e. Kühn, D. O'Sullivan, and C. A. Ardagna, Eds. Cham: Springer International Publishing, 2016, pp. 136–154. [10](#), [37](#), [39](#)
- [75] K. Figl, "Comprehension of procedural visual business process models," *Business & Information Systems Engineering*, vol. 59, no. 1, pp. 41–67, Feb 2017. [Online]. Available: <https://doi.org/10.1007/s12599-016-0460-2> [10](#), [37](#), [39](#)
- [76] R. Bobrik, T. Bauer, and M. Reichert, "Proviado – personalized and configurable visualizations of business processes," in *E-Commerce and Web Technologies*, K. Bauknecht, B. Pröll, and H. Werthner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 61–71. [41](#)
- [77] M. Reichert, J. Kolb, R. Bobrik, and T. Bauer, "Enabling personalized visualization of large business processes through parameterizable views," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 1653–1660. [Online]. Available: <http://doi.acm.org/10.1145/2245276.2232043> [41](#)
- [78] P.-M. Daigneault, D. Duval, and L. M. Imbeau, "Supervised scaling of semi-structured interview transcripts to characterize the ideology of a social policy reform," *Quality & Quantity*, vol. 52, no. 5, pp. 2151–2162, Sep 2018. [Online]. Available: <https://doi.org/10.1007/s11135-017-0650-0> [65](#)
- [79] J.-L. Padilla and J. P. Leighton, *Cognitive Interviewing and Think Aloud Methods*. Cham: Springer International Publishing, 2017, pp. 211–228. [Online]. Available: [https://doi.org/10.1007/978-3-319-56129-5\\_12](https://doi.org/10.1007/978-3-319-56129-5_12) [65](#)
- [80] N. Srinivas and D. Jagruthi, "Black box and white box testing techniques - a literature review," *International Journal of Embedded Systems and Applications*, vol. 2, pp. 29–50, 06 2012. [15](#), [143](#)
- [81] A. Roman, *Black-Box Testing Techniques*. Cham: Springer International Publishing, 2018, pp. 25–60. [Online]. Available: [https://doi.org/10.1007/978-3-319-98740-8\\_4](https://doi.org/10.1007/978-3-319-98740-8_4) [75](#)
- [82] A. Cockburn, *Writing Effective Use Cases*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000. [1](#), [77](#)
- [83] A. Donker and P. Markopoulos, "A Comparison of Think-aloud, Questionnaires and Interviews for Testing Usability with Children," in *People and Computers XVI - Memorable Yet Invisible*, X. Faulkner, J. Finlay, and F. Détienne, Eds. London: Springer London, 2002, pp. 305–316. [15](#)
- [84] P. Finkbeiner, *Qualitative Research: Semi-structured Expert Interview*. Cham: Springer International Publishing, 2017, pp. 141–181. [Online]. Available: [https://doi.org/10.1007/978-3-319-48544-7\\_6](https://doi.org/10.1007/978-3-319-48544-7_6) [15](#)

- [85] J. Dooley, *Unit Testing*. Berkeley, CA: Apress, 2011, pp. 193–208. [Online]. Available: [https://doi.org/10.1007/978-1-4302-3802-7\\_14](https://doi.org/10.1007/978-1-4302-3802-7_14) 16
- [86] W. Shen and S. Liu, “Formalization, testing and execution of a use case diagram,” in *Formal Methods and Software Engineering*, J. S. Dong and J. Woodcock, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 68–85. 16
- [87] F. S. Lasheras, J. V. Vilán, P. G. Nieto, and J. del Coz Díaz, “The use of design of experiments to improve a neural network model in order to predict the thickness of the chromium layer in a hard chromium plating process,” *Mathematical and Computer Modelling*, vol. 52, no. 7, pp. 1169 – 1176, 2010, mathematical Models in Medicine, Business & Engineering 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895717710001251> 16, 17
- [88] M. I. Kellner, P. H. Feiler, A. Finkelstein, T. Katayama, L. J. Osterweil, M. H. Penedo, and H. D. Rombach, “Ispw-6 software process example,” in *Proceedings. First International Conference on the Software Process*, Oct 1991, pp. 176–186. 88
- [89] T. Martinez-Ruiz, F. Garcia, M. Piattini, and J. Münch, “Modelling software process variability: an empirical study,” *IET Software*, vol. 5, no. 2, pp. 172–187, April 2011. 2

## **Anexo A**

# **Códigos de planos y funcionalidades de AVISPA**

### **A.1. Introducción**

Los códigos de cada uno de los planos y funcionalidades de la herramienta [AVISPA](#) están incluidos en el presente documento. A continuación se ilustran cada uno de ellos.

### **A.2. Códigos**

#### **A.2.1. Plano de rol sobrecargado**

La figura [A.1](#) ilustra el código del plano de rol sobrecargado de la nueva versión de [AVISPA](#).

```
viewOverloadPatternOn: mondrian
| sum average sumstandard standardsigma view legend |
sum := 0.
self myModel roles do:[each| sum:= sum + each numberOfTasks ].
average:= sum/(self size).
sumstandard:=0.
self myModel roles do:[each | sumstandard:= sumstandard + ((average – (each numberOfTasks ))
squared)].
standardsigma := ((sumstandard/(self size))) sqrt .
view:= RTMondrian new.
view view: mondrian.
view interaction highlightNeighbors: #relatedRoles color: Color orange.
view shape rectangle
width: [each | each numberOfAssociatedArtifacts * 2];
height: [each | each numberOfTasks * 2];
borderColor: Color black;
fillColor:[each | ((each numberOfTasks ) > standardsigma)
ifTrue: [
((each responsibleArtifacts size) > 0)
ifTrue: [ Color red ]
ifFalse: [ Color yellow ].
]
ifFalse: [Color white].
].
"text: [each | each presentationName ]."
view nodes: self.
view shape arrowedLine color: Color gray trans; withShorterDistanceAttachPoint.
view edges source: self connectFrom: #yourself toAll: #relatedRoles.
view layout circle.
view build.
legend := RTLegendBuilder new.
legend view: view view.
legend addText: 'Overloaded Roles Pattern'.
legend addRectanglePolymetricWidth: '#Associated Artifacts' height: '#Tasks' box: 'Role'.
legend addColor: Color yellow text:'Overload roles'.
legend addColor: Color red text:'Overload roles with artifact responsibility'.
legend addLineColor: Color gray text:'Role Cooperation'.
legend build
```

Figura A.1: Código de plano de rol sobrecargado

## Anexo B

# Encuestas realizadas en la Universidad de Chile

Se realizaron 2 entrevistas en total luego de la refactorización y mejoramiento de la herramienta [AVISPA](#).

Cada una de las pruebas contiene una descripción general de la herramienta, los planos que puede visualizar y una serie de preguntas sobre cada modelo de proceso de software para cada estudiante.

La primera prueba fue realizada a Daniel Perovich Gerosa de la Universidad de Chile que tiene conocimiento de Unified Method Architecture ([UMA](#)) así como [SPEM](#) sin conocimientos técnicos del entorno Moose.

La segunda prueba fue realizada a Alvaro José Peralta estudiante de la Universidad de Chile con conocimientos de [BPMN](#) y de la plataforma Moose.

Las dos pruebas realizadas se anexan a continuación.

**Lugar:** Departamento de Ciencias de la Computación - Universidad de Chile

**Nombre:** Alvaro Jose Peralta

**Fecha:** Septiembre de 2017

## AVISPA

### ¿Que es un modelo de proceso?

Un modelo de proceso describe una secuencia de fases para un ciclo de vida o producto. Por lo tanto, algunas veces es llamado también Ciclo de Vida de Producto, y cubre todo desde la idea comercial inicial hasta la desinstalación final de un producto después de su uso.

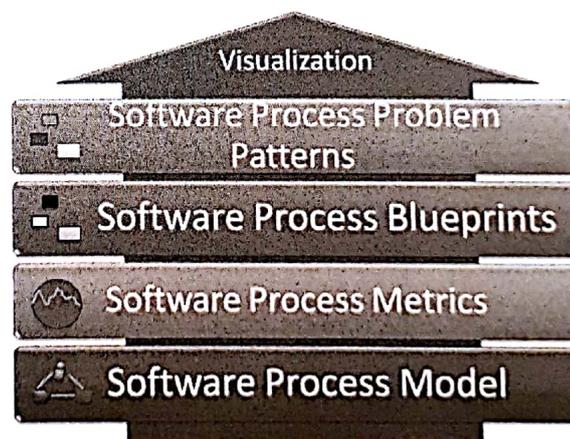
Cada fase del ciclo de vida tiene actividades que a su vez son subprocesos del mismo. Finalmente, el nivel de granularidad se reduce a contener tareas, roles y productos de trabajo.

- Las tareas son esfuerzos realizados, dirigidos por las organizaciones que cumplen un fin determinado. Así mismo, son suficientemente específicas como para ser ejecutadas.
- Los roles son los ejecutores de las tareas, cada rol en el modelo de proceso incluye una descripción de sus responsabilidades y sus habilidades requeridas.
- Los productos de trabajo son insumos o resultados de las tareas. Cada tarea como resultado establece o actualiza productos de trabajo.

Los meta-modelos de proceso contienen reglas que organizan la complejidad de los modelos de proceso ofreciendo suficiente flexibilidad a la organizaciones para la documentación de sus procesos pero la suficiente rigurosidad para permitir cumplir sus productos o servicios con calidad.

### ¿Que es AVISPA?

AVISPA es una herramienta que permite el análisis visual de modelos de proceso de software. Produce blueprints (planos visuales) que son construidos con la información del contenido de los modelos de proceso documentados en Eclipse Process Framework que están basados en el lenguaje de especificación de metamodelos SPEM 2.0. Estos contienen elementos visuales cuyas dimensiones varían con la información de los elementos de proceso y sus respectivas métricas.



La figura anterior ilustra cómo AVISPA permite construir las visualizaciones:

1. Se documenta el modelo de proceso de software
2. Se extraen métricas que serán utilizadas en la generación de los blueprints.
3. Se construyen los blueprints.
4. Se extraen patrones de error de los blueprints.

Un patrón de error son elementos identificados por AVISPA que son graficamente como elementos “anormalmente diferentes” del resto.

Existen pues diferentes patrones de error que pueden ser visualizados en los diversos planos de AVISPA:

- **Roles aislados.** Pueden existir ciertas tareas que un unico rol ejecuta por si mismo pero no es muy frecuente tener un rol que nunca colabora en ninguna tarea con otros roles. Por lo general este tipo de error es mas que todo un error en la especificacion del modelo de proceso de software. Este patrón de error se encuentra en AVISPA como “ROLE BLUEPRINT”.
- **Subproyectos independientes.** En un “TASK BLUEPRINT” y en un “WORK PRODUCT BLUEPRINT”, las tareas y los productos de trabajo estan relacionados con vertices que indican precedencia. El modelo de proceso especifica la manera de proceder cuando se trabaja en un unico proyecto. Entonces esto es conceptualmente irregular tener grafos desconectados en ambos blueprints.
- **Roles sobrecargados.** Si un rol esta involucrado en un largo numero de tareas, se convierte en un riesgo: si un rol falla, todas sus tareas asociadas también lo harán. Es un posible error en la concepción del modelo de proceso. Podríamos decir que un rol está sobrecargado si es mas de una desviación estándar mas larga que el tamaño promedio. Este patron de error es ilustrado como “ROLE BLUEPRINT”.
- **Tareas multipropósito.** Un proceso cuyas tareas tengan demasiados productos de entrada y demasiados productos de salida revelan que dichas tareas no han sido especificadas con la suficiente granularidad. Este patron es visto en el “TASK BLUEPRINT”. Se considera una tarea demasiado compleja si es mas ancha que una desviación estandar del ancho promedio de tareas.
- **Productos de trabajo demandados.** Los productos de trabajo requeridos para un alto numero de tareas puede causar serios cuellos de botella que no están disponibles y, por lo tanto, podrían revelar una posible falla de documentación del modelo. Esta situación es vista en el “WORK PRODUCT BLUEPRINT”. Se consideran como nodos cuyo ancho es mas que una desviación estándar del promedio.
- **Roles, tareas y productos de trabajo sin guía.** Si un rol, tarea o producto de trabajo no tiene una guía asociada, dicho nodo se resaltarán con azul aquellos elementos que no tengan dicha guía
- **Productos de trabajo desperdiciados.** Los entregables son esos artefactos que necesitan ser entregados al cliente como parte del producto final. Si hay productos de trabajo que no son prerequisites para otros productos ni entregables o tampoco son una entrada a ninguna tarea del proceso entonces son desperdicio dentro del modelos de proceso. AVISPA resalta éste tipo de nodos dentro del “WORK PRODUCT BLUEPRINT”.

Las anteriores figuras ilustran algunos de estos modelos de proceso mencionados anteriormente.

AVISPA agrupa los elementos de proceso en diferentes tipos de planos:

- Los que ilustran la interacción de todos los elementos de proceso de un solo tipo (tareas, roles y/o productos de trabajo) por todo el modelo de proceso.
- Lo mismo que lo anterior pero agrupando estos elementos por fases / actividades. Ya que bien sabemos que una actividad puede ser entendida como un subproceso o un subconjunto de elementos del modelo de proceso.

De todas maneras, los planos enunciados anteriormente (y por lo tanto los patrones de error asociados)

pueden ser identificados en estos dos tipos de agrupamiento de elementos de proceso.

Sin embargo existen planos por cada actividad, y gracias al trabajo de Canfora et al. que ilustran como algunas métricas asociadas a las actividades y las tareas permiten tener una noción del acoplamiento de actividades entre si.

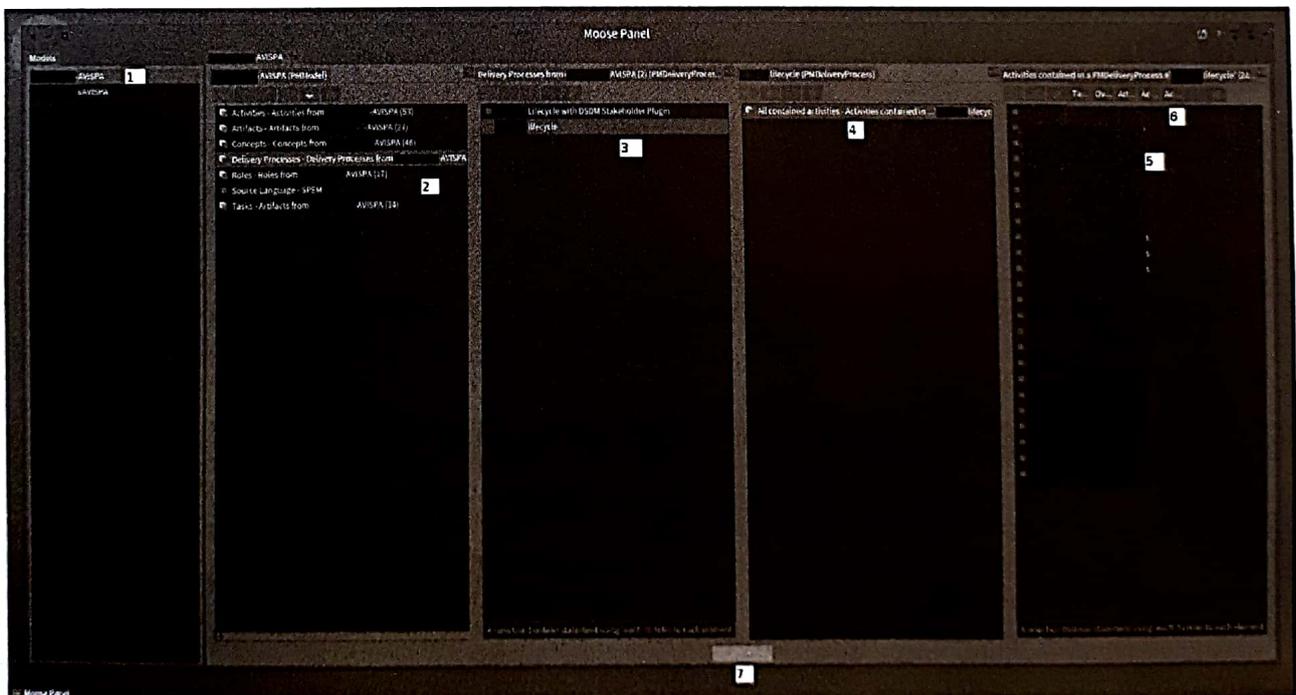
De aquí nacen 2 modelos de proceso nuevos (que se muestran al momento de seleccionar una actividad dentro del Moose Panel):

- **Activity coupling by tasks:** Las actividades al ser un subproceso, también contienen tareas dentro de si mismas con las cuales se puede calcular un indice de acoplamiento. Sin embargo, no están basadas en las métricas de Canfora et al. Pero ofrecen una visualización del acoplamiento entre tareas contenidas en una actividad. Se visualizan con un gradiente de negro (menos acoplado) a rojo (mas acoplado) por cada nodo dibujado
- **Activity coupling by contained activities:** Es la misma que la anterior pero basandose completamente en las métricas de Canfora. et al. Ofreciendo una visualización del acoplamiento entre las actividades contenidas en una actividad. Se visualizan con un gradiente de negro (menos acoplado) a verde (mas acoplado) por cada nodo dibujado.

¿Como usar AVISPA?

Avispa importa un XML construido a partir de la información de la herramienta de modelado Eclipse Process Framework e importa los elementos de proceso en objetos que son finalmente utilizados para generar sus visualizaciones.

A continuación se ilustra en una figura la interfaz principal de la herramienta:



Se enuncia cada ítem de la anterior gráfica:

1. Aquí se ilustran los modelos de procesos cargados en la herramienta.

### Preguntas antes de ejecutar la prueba:

- Por favor indique cuantos años de experiencia tiene en modelamiento de procesos:

5 años

- Indique cual meta-modelo de proceso usa habitualmente para documentar sus procesos de software. Justifique el por qué su preferencia.

BPMN 2.0 principalmente por la gran cantidad de herramientas que existen en el mercado y la promesa de poder ejecutar el proceso usando un motor que lo automatice.

- ¿Alguna vez ha utilizado Eclipse Process Framework para documentar sus procesos?.

Si, dos veces para comparar y tambien para editar modelos ya diseñados por ejemplo SCRUM. y así adaptarlo a la empresa donde trabajo.

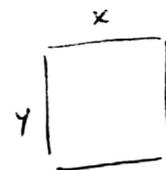
### Ejercicios de la prueba:

- Elija una actividad, la que parezca mas compleja, e indique que roles están sobrecargados, donde hay subproyectos independientes y productos de trabajo demandados, justifique su respuesta:

Actividad: Monitorear y controlar el proyecto

rol sobrecargado: jefe de proyecto

en el grafico contained overload roles este rol aparece en rojo y la leyenda indica "overload roles"



Actividad mas grande en  
x = artefactos producidos  
y = artefactos consumidos

subproyecto independiente: Revisar Reportes de Aseguramiento de calidad de producto y proceso, esto porque en la grafica es el unico de color azul.

producto demandado: plan de proyecto, porque es el mas grande.

- Indicar cuales son las actividades cuyos elementos de proceso (tareas) son mas acoplados, justifique su respuesta:

comprar requerimientos  
 Analizar y validar requerimientos

es porque en el grafico son los que mas resaltan por color

Definir objetivos de medicón (tengo la duda porque no tiene relación, pero si tiene el mismo color que los otros 2)

- Indique que mejoras usted sugiere para el modelo de proceso que ha evaluado:

al empezar a evaluar empiezo desde cero y en otras herramientas como newrelic (para evaluar código) comienzo con un dashboard de indicadores relevantes para la industria.

- ¿Encuentra usted alguna diferencia entre los planos de todos los elementos de proceso del modelo y los planos que visualizan los modelos de proceso agrupados por actividades? ¿Cual de estos dos tipos de visualizaciones es mas fácil de entender para usted?

yo modelo en base a actividades por lo que ver mas actividades es mas natural para obtener una perspectiva mas amplia.

2 preguntas muy poco espacio respondo solo 1.

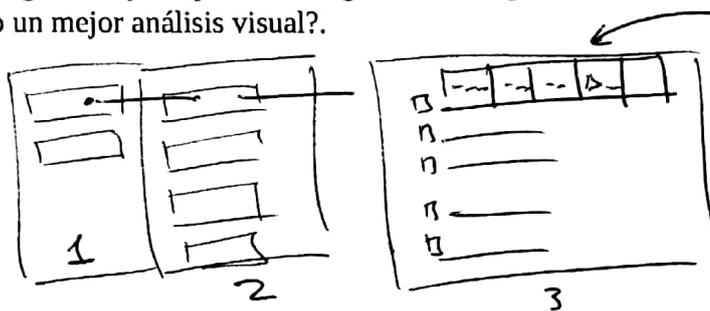
Preguntas después de la prueba:

- ¿Que opinión tiene de la prueba que acaba de realizar?.
- es una prueba para alguien que tenga al menos 5 diseños, asi sean simples de procesos.
- la explicación inicial es buena para entender la herramienta, no se si sea apropiado agregar contexto de que se espera con la prueba al principio.

- ¿Es para usted intuitiva la manera de como se visualizan los modelos de proceso de software en la herramienta AVISPA?

Al principio si ya que al ver el proceso de alto nivel es facil iniciar la navegacion, pero al entrar por ejemplo a las actividades cuesta decidir por donde seguir si visitando los menus superiores o seguir navegando.

- ¿Hay alguna mejora que usted sugeriría a la representación de modelos de procesos para llevar a cabo un mejor análisis visual?



este tab es muy incomodo y no deja leer bien el contenido de lo puedo esperar.

- ¿Es confiable para usted la manera de como se representan los modelos de proceso de software en la herramienta AVISPA?

lo primero hay coherencia entre graficos y eso es muy importante, ademas si yo conosco el modelo tambien se de sus relaciones y es como estar leyendo una critica de un escrito, si no le encontrara sentido dudaria al instante.

AVISPA

11-09-2017.  
OpenUp- Daniel P.  
UCHILE

¿Que es un modelo de proceso?

Un modelo de proceso describe una secuencia de fases para un ciclo de vida o producto. Por lo tanto, algunas veces es llamado también Ciclo de Vida de Producto, y cubre todo desde la idea comercial inicial hasta la desinstalación final de un producto después de su uso.

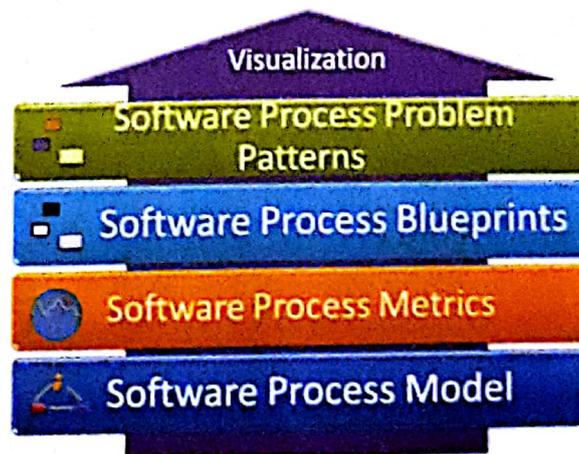
Cada fase del ciclo de vida tiene actividades que a su vez son subprocesos del mismo. Finalmente, el nivel de granularidad se reduce a contener tareas, roles y productos de trabajo.

- Las tareas son una actividad suficientemente específica como para ser ejecutada y cumple un fin específico dentro del modelo de proceso.
- Los roles son los ejecutores de las tareas, cada rol en el modelo de proceso incluye una descripción de sus responsabilidades y sus habilidades requeridas.
- Los productos de trabajo son insumos o resultados de las tareas. Cada tarea como resultado establece o actualiza productos de trabajo.

Los meta-modelos de proceso contienen reglas que organizan la complejidad de los modelos de proceso ofreciendo suficiente flexibilidad a la organizaciones para la documentación de sus procesos pero la suficiente rigurosidad para permitir cumplir sus productos o servicios con calidad.

¿Que es AVISPA?

AVISPA es una herramienta que permite el análisis visual de modelos de proceso de software. Produce blueprints (planos visuales) que son construidos con la información del contenido de los modelos de proceso documentados en Eclipse Process Framework que están basados en el lenguaje de especificación de metamodelos SPEM 2.0. Estos contienen elementos visuales cuyas dimensiones varían con la información de los elementos de proceso y sus respectivas métricas.



La figura anterior ilustra cómo AVISPA permite construir las visualizaciones:

1. Se documenta el modelo de proceso de software
2. Se extraen métricas que serán utilizadas en la generación de los blueprints.
3. Se construyen los blueprints.
4. Se extraen patrones de error de los blueprints.

Un patrón de error son elementos identificados por AVISPA que son graficamente como elementos "anormalmente diferentes" del resto.

Existen pues diferentes patrones de error que pueden ser visualizados en los diversos planos de AVISPA:

- **Roles aislados.** Pueden existir ciertas tareas que un único rol ejecuta por sí mismo pero no es muy frecuente tener un rol que nunca colabora en ninguna tarea con otros roles. Por lo general este tipo de error es más que todo un error en la especificación del modelo de proceso de software. Este patrón de error se encuentra en AVISPA como "ROLE BLUEPRINT".
- **Subproyectos independientes.** En un "TASK BLUEPRINT" y en un "WORK PRODUCT BLUEPRINT", las tareas y los productos de trabajo están relacionados con vértices que indican precedencia. El modelo de proceso especifica la manera de proceder cuando se trabaja en un único proyecto. Entonces esto es conceptualmente irregular tener grafos desconectados en ambos blueprints.
- **Roles sobrecargados.** Si un rol esta involucrado en un largo número de tareas, se convierte en un riesgo: si un rol falla, todas sus tareas asociadas también lo harán. Es un posible error en la concepción del modelo de proceso. Podríamos decir que un rol está sobrecargado si es más de una desviación estándar más larga que el tamaño promedio. Este patrón de error es ilustrado como "ROLE BLUEPRINT".
- **Tareas multipropósito.** Un proceso cuyas tareas tengan demasiados productos de entrada y demasiados productos de salida revelan que dichas tareas no han sido especificadas con la suficiente granularidad. Este patrón es visto en el "TASK BLUEPRINT". Se considera una tarea demasiado compleja si es más ancha que una desviación estandar del ancho promedio de tareas.
- **Productos de trabajo demandados.** Los productos de trabajo requeridos para un alto número de tareas puede causar serios cuellos de botella que no están disponibles y, por lo tanto, podrían revelar una posible falla de documentación del modelo. Esta situación es vista en el "WORK PRODUCT BLUEPRINT". Se consideran como nodos cuyo ancho es más que una desviación estándar del promedio.
- **Roles, tareas y productos de trabajo sin guía.** Si un rol, tarea o producto de trabajo no tiene una guía asociada, dicho nodo se resaltaran con azul aquellos elementos que no tengan dicha guía
- **Productos de trabajo desperdiciados.** Los entregables son esos artefactos que necesitan ser entregados al cliente como parte del producto final. Si hay productos de trabajo que no son prerrequisitos para otros productos ni entregables o tampoco son una entrada a ninguna tarea del proceso entonces son desperdicio dentro del modelos de proceso. AVISPA resalta éste tipo de nodos dentro del "WORK PRODUCT BLUEPRINT".

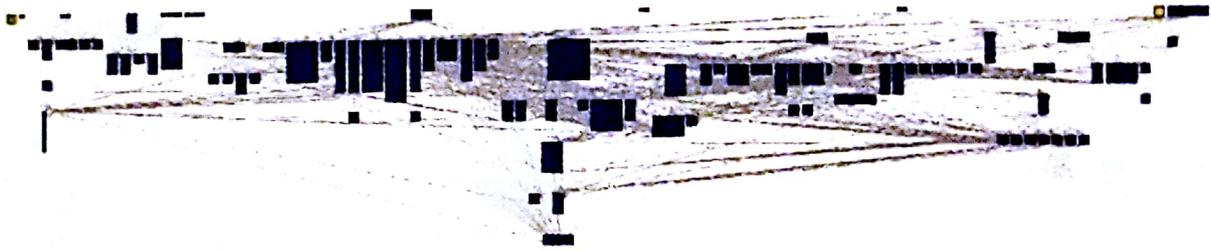


Figure 11. TASK BLUEPRINT for localizing disconnected subgraphs in Amisoft.

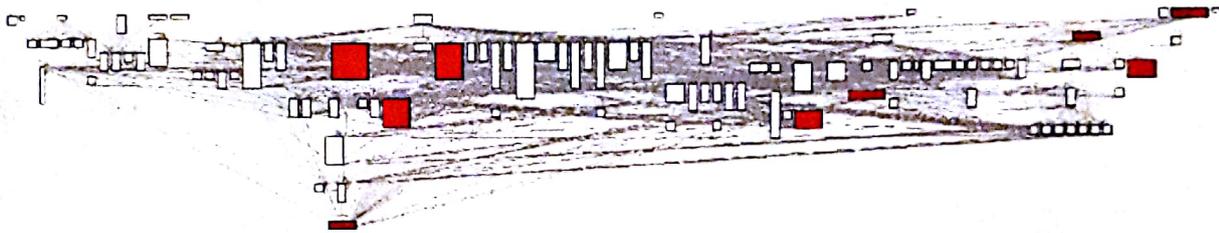


Figure 12. TASK BLUEPRINT for localizing multiple purpose tasks in Amisoft.

Las anteriores figuras ilustran algunos de estos modelos de proceso mencionados anteriormente.

AVISPA agrupa los elementos de proceso en diferentes tipos de planos:

- Los que ilustran la interacción de todos los elementos de proceso de un solo tipo (tareas, roles y/o productos de trabajo) por todo el modelo de proceso.
- Lo mismo que lo anterior pero agrupando estos elementos por fases / actividades. Ya que bien sabemos que una actividad puede ser entendida como un subproceso o un subconjunto de elementos del modelo de proceso.

De todas maneras, los planos enunciados anteriormente (y por lo tanto los patrones de error asociados) pueden ser identificados en estos dos tipos de agrupamiento de elementos de proceso.

Sin embargo existen planos por cada actividad, y gracias al trabajo de Canfora et al. que ilustran cómo algunas métricas asociadas a las actividades y las tareas permiten tener una noción del acoplamiento de actividades entre sí.

De aquí nacen 2 modelos de proceso nuevos (que se muestran al momento de seleccionar una actividad dentro del Moose Panel):

- **Activity coupling by tasks:** Las actividades al ser un subproceso, también contienen tareas dentro de sí mismas con las cuales se puede calcular un índice de acoplamiento. Sin embargo, no están basadas en las métricas de Canfora et al. Pero ofrecen una visualización del acoplamiento entre tareas contenidas en una actividad. Se visualizan con un gradiente de negro a rojo por cada nodo dibujado
- **Activity coupling by contained activities:** Es la misma que la anterior pero basándose completamente en las métricas de Canfora. et al. Ofreciendo una visualización del acoplamiento entre las actividades contenidas en una actividad. Se visualizan con un gradiente de negro a verde por cada nodo dibujado.

Preguntas antes de ejecutar la prueba:

- Por favor indique cuantos años de experiencia tiene en modelamiento de procesos:

8 años .

- Indique cuál meta-modelo de proceso usa habitualmente para documentar sus procesos de software. Justifique el por qué su preferencia.

UMA (SPEN) por el soporte gratuito por EEF Composer.

- ¿Alguna vez ha utilizado Eclipse Process Framework para documentar sus procesos?.

Si .

Ejercicios de la prueba:

- Elija una actividad, la que parezca más compleja, e indique que roles están sobrecargados, dónde hay subproyectos independientes y productos de trabajo demandados, justifique su respuesta:

Elaboration iteration → developer (sobrecargado) ; - realiza más tareas.  
↓  
subproyectos: 5 subproyectos.  
artefacto demandado: visión (entrada de más tareas).

- Indicar cuáles son las actividades cuyos elementos de proceso (tareas) son más acoplados, justifique su respuesta:

Inception. más clara, mejor precedencia.

- ~~Indique que mejoras usted sugiere para el modelo de proceso que ha evaluado.~~

- ¿Encuentra usted alguna diferencia entre los planos de todos los elementos de proceso del modelo y los planos que visualizan los modelos de proceso agrupados por actividades? ¿Cuál de estos dos tipos de visualizaciones es más fácil de entender para usted?

Encuentro que las visualizaciones ~~de~~ son fáciles o fáciles en función de qué esté analizando el mismo proceso.

Todos (los planos) o por actividad (cuando se quiere analizar algo concreto).

#### Preguntas después de la prueba:

- ¿Que opinión tiene de la prueba que acaba de realizar?.

El proceso es complejo, sugiero utilizar uno más simple.  
Debe proveerse ejemplos concretos.

- ¿Es para usted intuitiva la manera de como se visualizan los modelos de proceso de software en la herramienta AVISPA?.

No .

- ¿Hay alguna mejora que usted sugeriría a la representación de modelos de procesos para llevar a cabo un mejor análisis visual?.

El análisis visual debe estar acompañado de datos de datos .

- ¿Es confiable para usted la manera de como se representan los modelos de proceso de software en la herramienta AVISPA?

Aunque que es confiable, pero no tanto como demostrarlo  
datos que no conozca el proceso original en EFPC .

Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

## Anexo C

# Resultados del test ANOVA

Los resultados de la prueba de ANOVA se han documentado utilizando 2 herramientas, SPSS y LibreOffice Calc.

Se usaron las 2 herramientas para poder obtener diferentes datos de las evaluaciones ANOVA y realizar validación & verificación de la evaluación y sus resultados respecto a la evaluación 5 (ver sección 4.6).

Para el sistema LibreOffice Calc, el archivo que contiene tanto la evaluación de  $F_1$ -Score como la evaluación por ANOVA se encuentra en el archivo `./Anexos/AnovaAVISPA/LibreOffice Calc/Evaluación con ANOVA.ods`.

Para el software SPSS, cuyos archivos se ubican en `./Anexos/AnovaAVISPA/SPSS/` se documentan 2 archivos:

- Datos de entrada: El archivo `Datos AVISPA F1Score.sav` contiene los datos de la evaluación, organizados de tal manera que sean entendidos por el software SPSS.
- Resultados de la evaluación de AVISPA: El archivo `Resultado ANOVA.spv` contiene los resultados obtenidos por ANOVA, las siguientes páginas ilustrarán los resultados obtenidos por el sistema SPSS.

```

ONEWAY F1Score BY Avispa
  /STATISTICS DESCRIPTIVES EFFECTS HOMOGENEITY BROWNFORSYTHE WELCH
  /PLOT MEANS
  /MISSING ANALYSIS.

```

## ANOVA de un factor

### Notas

Resultados creados		30-MAR-2019 15:06:40
Comentarios		
Entrada	Datos	F:\Trabajos académicos\Maestría\Tesis\Monografía\Otros\Anova AVISPA\SPSS\Datos AVISPA F1Score.sav
	Conjunto de datos activo	Conjunto_de_datos0
	Filtro	<ninguno>
	Peso	<ninguno>
	Dividir archivo	<ninguno>
	Núm. de filas del archivo de trabajo	36
Tratamiento de los valores perdidos	Definición de los valores perdidos	Los valores perdidos definidos por el usuario serán tratados como perdidos.
	Casos utilizados	Los estadísticos de cada análisis se basan en los casos sin datos perdidos para cualquier variable en el análisis.
Sintaxis		ONEWAY F1Score BY Avispa /STATISTICS DESCRIPTIVES EFFECTS HOMOGENEITY BROWNFORSYTHE WELCH /PLOT MEANS...
Recursos	Tiempo de procesador	00:00:00,17
	Tiempo transcurrido	00:00:00,15

[Conjunto\_de\_datos0] F:\Trabajos académicos\Maestría\Tesis\Monografía\Otros\Anova AVISPA\SPSS\Datos AVISPA F1Score.sav

**Descriptivos**

F1 Score

	N	Media	Desviación típica	Error típico
1	18	,6441762269	,0597767289	,0140895101
2	18	,8904130074	,0297814062	,0070195448
Total	36	,7672946171	,1332576655	,0222096109
Modelo				
Efectos fijos			,0472238789	,0078706465
Efectos aleatorios				,1231183902

**Descriptivos**

F1 Score

	Intervalo de confianza para la media al 95%		Mínimo	Máximo
	Límite inferior	Límite superior		
1	,6144499590	,6739024948	,5871559633	,8370927318
2	,8756030625	,9052229522	,8464818763	,9472895329
Total	,7222067099	,8123825243	,5871559633	,9472895329
Modelo				
Efectos fijos	,7512995390	,7832896952		
Efectos aleatorios	-,7970728560	2,331662090		

**Descriptivos**

F1 Score

	Varianza entre componentes
1	
2	
Total	
Modelo	
Efectos fijos	
Efectos aleatorios	,0301923819

**Prueba de homogeneidad de varianzas**

F1 Score

Estadístico de Levene	gl1	gl2	Sig.
1,453	1	34	,236

### ANOVA de un factor

F1 Score

	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Inter-grupos	,546	1	,546	244,695	,000
Intra-grupos	,076	34	,002		
Total	,622	35			

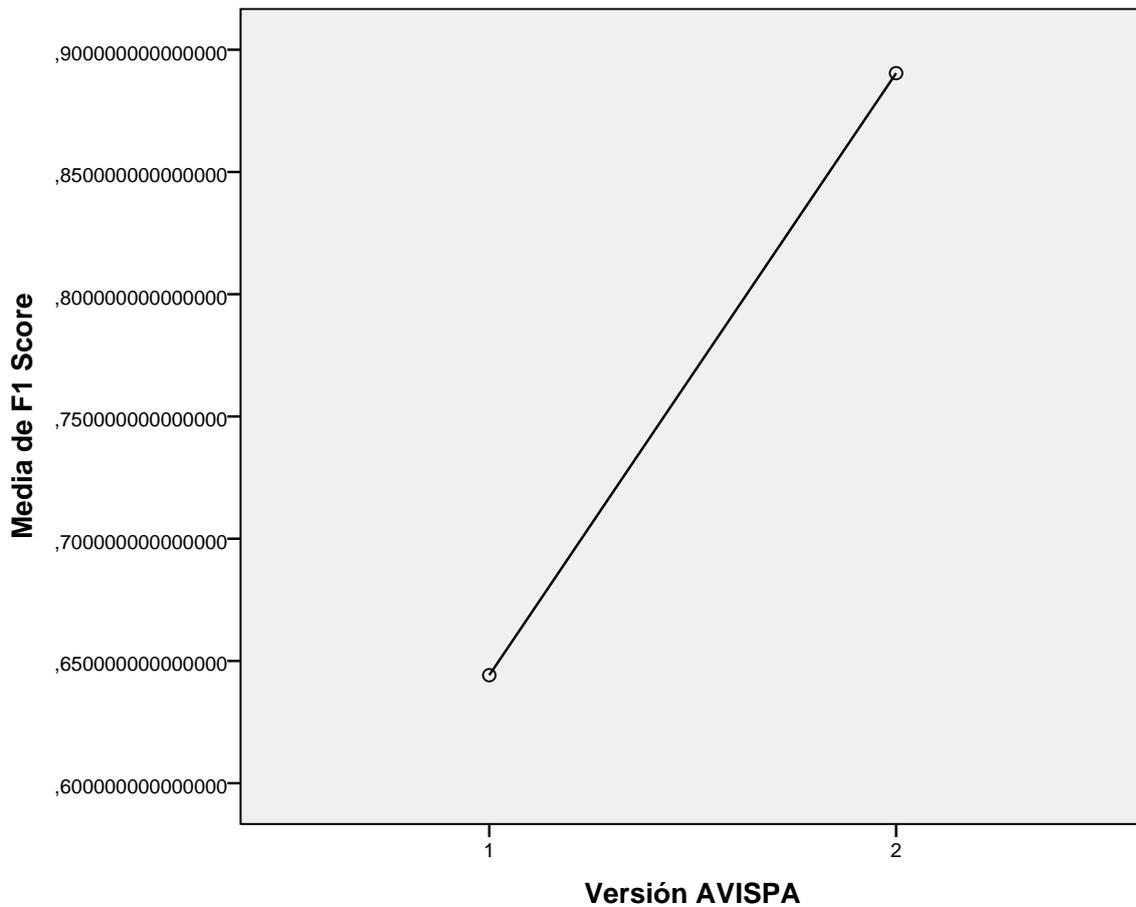
### Pruebas robustas de igualdad de las medias

F1 Score

	Estadístico <sup>a</sup>	gl1	gl2	Sig.
Welch	244,695	1	24,950	,000
Brown-Forsythe	244,695	1	24,950	,000

a. Distribuidos en F asintóticamente.

### Gráfico de las medias



## Anexo D

# Ejecutables de AVISPA

Los ejecutables de [AVISPA](#), que se ubican dentro de la carpeta `.Anexos/Software AVISPA/`, se ejecutan de la siguiente manera:

- Para [AVISPA](#) 1.0 se ejecuta con el archivo `v1.0/Moose.sh` (ver figura [D.1](#)).
- Para [AVISPA](#) 2.0 se ejecuta con el archivo `v2.0/Pharo` (ver figura [D.2](#)).

Ambos software se ejecutan en sistemas operativos linux, y como pantalla inicial tienen el explorador de Moose con todos los modelos de procesos de software en [SPEM](#) 2.0 que fueron evaluados con ANOVA (ver sección [4.6](#) que ilustra los procesos evaluados).

## Métricas de Modelos de Procesos SPEM 2.0 en la Confiabilidad de su Análisis Visual

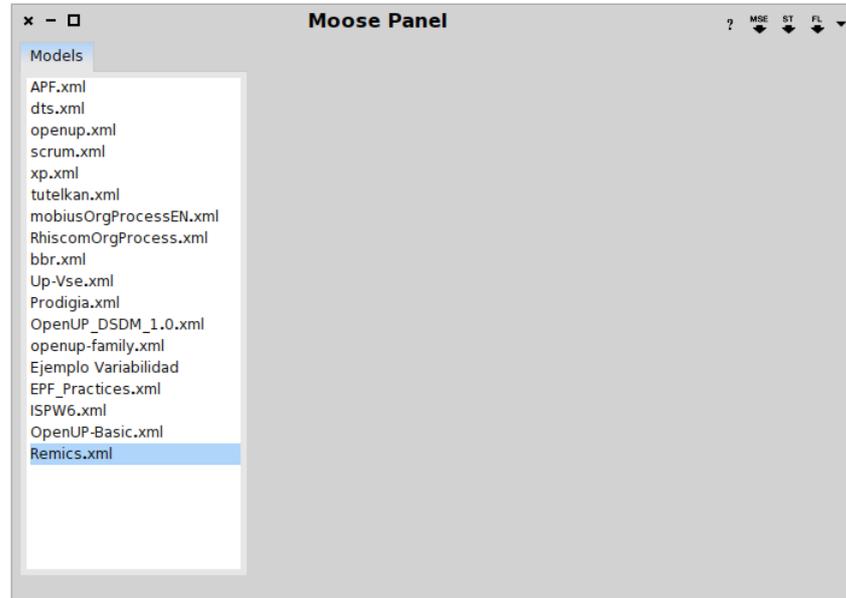


Figura D.1: Pantalla principal de AVISPA 1.0

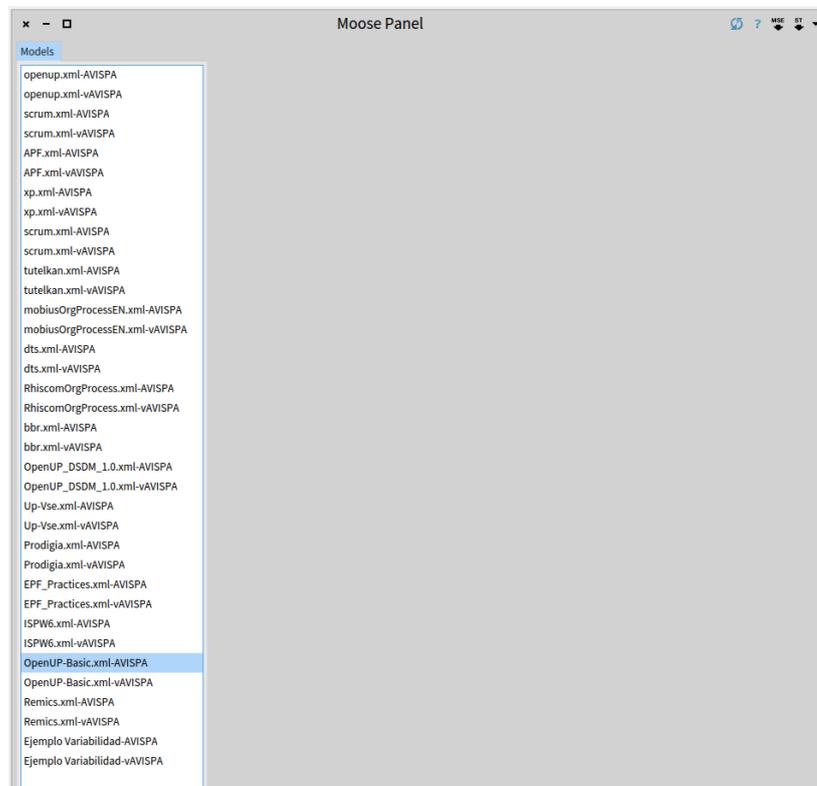


Figura D.2: Pantalla principal de AVISPA 2.0

## Anexo E

# Evaluación 2 sobre código de AVISPA

### E.0.1. Justificación de la evaluación

Para mejorar la confiabilidad de los datos ilustrados en [AVISPA](#) se definen unos casos de prueba por medio de la técnica de caja blanca de ruta básica [80] que permiten hacer una depuración más adecuada de los datos de los planos, patrones y métricas de [AVISPA](#) que no pueden ser visualizados fácilmente en ciertas partes de código de [AVISPA](#). En el presente estudio se ilustrará la técnica aplicada de un solo plano.

### E.0.2. Diseño de la evaluación

Para cubrir todos los casos que deben ilustrar los planos de [AVISPA](#), es necesario realizar ejemplos mínimos que cubran todas las rutas del algoritmo desde su importación hasta sus visualizaciones. Por lo tanto, las pruebas de caja blanca de ruta básica entonces deben cubrir 2 partes del algoritmo esenciales para asegurar, o por lo menos mejorar considerablemente, la calidad de las visualizaciones: (i) la sección del algoritmo que importa y mapea los datos del XML en datos que Pharo / Moose / Mondrian pueden procesar y (ii) las decisiones sobre los elementos de proceso que Pharo / Moose / Mondrian realizan para resaltar visualmente los patrones de error [AVISPA](#) genera.

La figura [E.1](#) ilustra el flujo de las pruebas de caja blanca realizadas para [AVISPA](#).

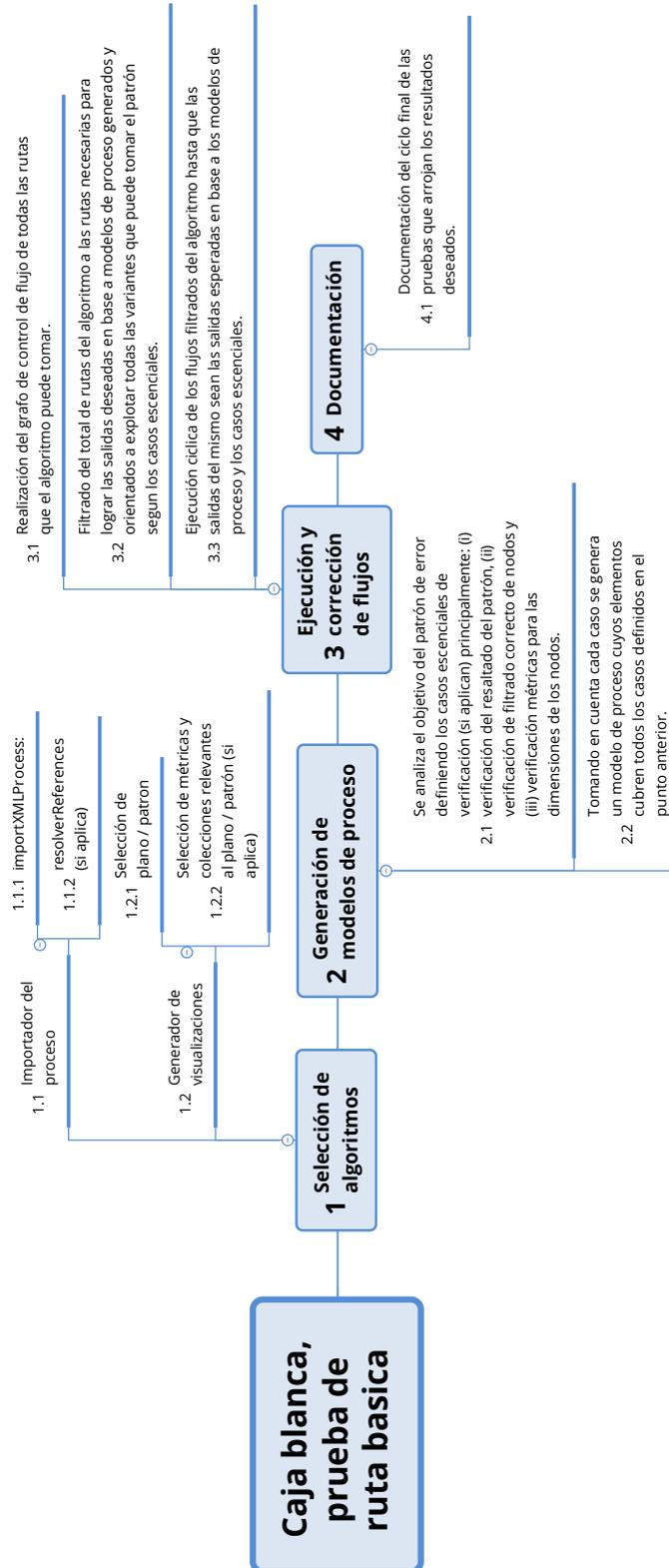


Figura E.1: Flujograma de pruebas de caja blanca por cada plano de AVISPA

### E.0.3. Ejecución sobre el patrón de roles responsables

Este estudio se realiza con el patrón de roles responsables indicado en la sección 3.4.4.5 del capítulo anterior. Partiendo del flujograma de la figura E.1 las pruebas respectivas se generan de la siguiente manera:

1. **Prueba sobre el código de importación:** El método `importXMLDocument: document` se encarga de procesar el documento XML exportado de EPF y lo transforma en elementos y relaciones entendibles para Pharo / Moose / Mondrian. Esta función a su vez llama a otra llamada `correlationIOArtifactsAndPerformerRoles: tasks to: pmTasks and: pmArtifacts and: pmRoles` que, como su nombre lo indica, se encarga de la importación y resolución de relaciones entre los artefactos y los roles responsables. Es a éste algoritmo que se le aplica la prueba de caja blanca.
2. **Prueba sobre el patrón de error:** la función `viewArtifactResponsibleRolesOn: aView` se encarga de implementar la visualización del patrón con los datos ya cargados del punto anterior usando Mondrian. A éste segundo algoritmo también se le aplica la prueba de caja blanca.

Siguiendo del orden estipulado en el proceso de evaluación se procede a instanciar el proceso de validación en el presente patrón de roles responsables (únicamente puntos 1 y 2 del proceso de evaluación):

#### 1. Selección de algoritmos:

##### 1.1. Importador del proceso:

1.1.1. **Importador:** función `correlationIOArtifactsAndPerformerRoles: tasks to: pmTasks and: pmArtifacts and: pmRoles`

1.1.2. **Solucionador de relaciones:** No aplica, el importador las resuelve

##### 1.2. Generador de visualizaciones:

1.2.1. **Plano / patrón seleccionado:** función `viewArtifactResponsibleRolesOn: aView`

##### 1.2.2. Factores relevantes para V&V:

###### 1.2.2.1. Importador:

- Los roles que producen artefactos deben ser los mismos del modelo de proceso.
- Los roles responsables de artefactos deben ser los mismos del modelo de proceso.
- Los artefactos de salida de una tarea deben ser los mismos del modelo de proceso.

###### 1.2.2.2. Plano / patrón:

- Los colores de cada artefacto de salida corresponden al objetivo del patrón y sus decisiones de resaltado.

1.2.2.3. **Métricas:**

- Número de roles que producen un artefacto por cada uno de estos.

2. **Generación de modelos de proceso:**

2.1. **Análisis del patrón:** El patrón de error solo es aplicable a artefactos de salida ya que su producción conlleva un esfuerzo. Este patrón de error diferencia roles responsables de artefactos de aquellos que los producen cumpliendo las siguientes reglas:

- Resaltar con azul los artefactos que no tengan roles responsables ó (exclusivo) 3 roles o menos produciéndolos
- Resaltar con purpura los artefactos con 4 o más roles produciéndolos y con roles responsables
- Resaltar con rojo los artefactos con 4 o más roles produciéndolos y sin roles responsables
- Resaltar con blanco los artefactos con 3 roles o menos produciéndolos y con roles responsables

2.2. **Descripción del modelo de proceso de prueba:** El modelo de proceso cumple con las siguientes características:

- **Roles:** 5
- **Tareas:** 2
- **Artefactos:** 4
- **Relaciones:**
  - Roles 1 a 4 ejecutan tarea 1.
  - Tarea 1 produce artefactos 1 y 2 (artefacto 2 se debe resaltar de rojo).
  - Rol 1 es responsable de artefacto 1 (artefacto 1 se debe resaltar de purpura).
  - Roles 1 a 3 producen artefacto 3 (artefacto 3 se debe resaltar de azul así no tenga roles produciéndolo).
  - Rol 5 es responsable de artefacto 4 (artefacto 4 se debe resaltar de blanco).

**E.0.3.1. Resultados del algoritmo importador / de resolución de relaciones**

El código de importación que se encarga de la definición y resolución de elementos de proceso y sus respectivas relaciones está ilustrado en la figura E.2. Para continuar con el protocolo de implementación de prueba de caja blanca por ruta básica se divide el código por bloques de ejecución tal como se ilustra a continuación:

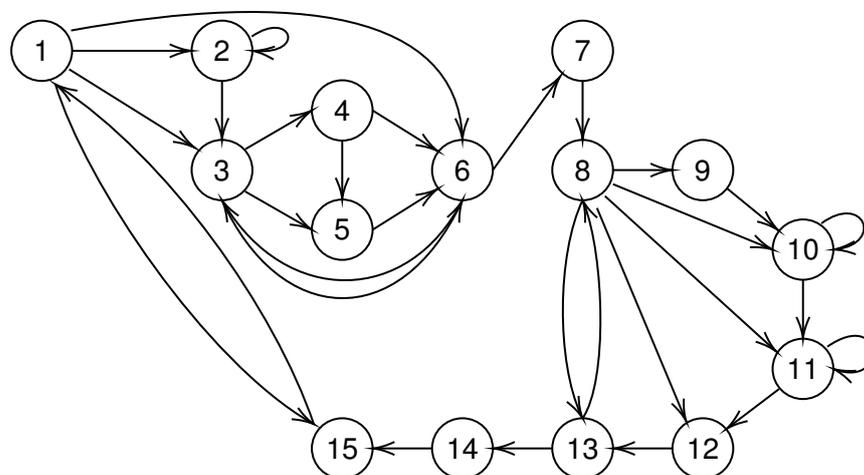
1. Líneas 3-6

```

1 correlationIOArtifactsAndPerformerRoles: tasks to: pmTasks and: pmArtifacts and: pmRoles
2 | tempRoles tempInputArtifacts tempOutputArtifacts tempPMTask |
3 tempRoles := OrderedCollection new.
4 tempInputArtifacts := OrderedCollection new.
5 tempOutputArtifacts := OrderedCollection new.
6 tasks do: [ :task |
7     "Extraccion de roles, tareas de entrada y salida del XML"
8     "Extraccion de los roles que ejecutan la tarea task del ciclo"
9     pmRoles do: [ :tempPMRole |
10         (task / #PerformedBy) do: [ :role |
11             ((tempPMRole id asString) = (role nodes first asString))
12             ifTrue: [tempRoles add: tempPMRole].
13         ].
14     ].
15     "Extraccion de los artefactos de entrada y de salida de la tarea task del ciclo"
16     pmArtifacts do: [ :tempPMArtifact |
17         (task / #MandatoryInput) do: [ :inputArtifact |
18             ((tempPMArtifact id asString) = (inputArtifact nodes first asString))
19             ifTrue: [tempInputArtifacts add: tempPMArtifact].
20         ].
21         (task / #Output) do: [ :outputArtifact |
22             ((tempPMArtifact id asString) = (outputArtifact nodes first asString))
23             ifTrue: [tempOutputArtifacts add: tempPMArtifact].
24         ].
25     ].
26     tempPMTask := PMTask fromXMLDescription: task.
27     "Resolucion de relaciones de las extracciones anteriores"
28     pmTasks do: [ :pmTask |
29         (pmTask id asString = tempPMTask id asString)
30         ifTrue: [
31             "Asignando roles que ejecutan a la tarea"
32             pmTask roles: tempRoles.
33             "Asignando artefactos de entrada a la tarea"
34             tempInputArtifacts do: [ :tempInputArtifact |
35                 tempInputArtifact inputTasks add: pmTask.
36             ].
37             "Asignando artefactos de salida a la tarea"
38             tempOutputArtifacts do: [ :tempOutputArtifact |
39                 tempOutputArtifact outputTasks add: pmTask.
40                 tempOutputArtifact performerRoles: tempRoles.
41             ].
42             pmTask artifactMandatoryInputs: tempInputArtifacts.
43             pmTask artifactOutputs: tempOutputArtifacts.
44         ]
45     ].
46     tempRoles := OrderedCollection new.
47     tempInputArtifacts := OrderedCollection new.
48     tempOutputArtifacts := OrderedCollection new.
49 ].
50 "pmTasks." "Mensajes de depuracion."
51 "self halt."

```

**Figura E.2:** Código de importación de elementos de proceso para el patrón roles responsables



**Figura E.3:** Grafo de control de flujo del algoritmo de importación del patrón de roles responsables.

2. Líneas 9-14
3. Línea 16
4. Líneas 17-20
5. Líneas 21-24
6. Línea 25
7. Línea 26
8. Líneas 28-30
9. Línea 32
10. Líneas 34-36
11. Líneas 38-41
12. Líneas 42-43
13. Líneas 44-45
14. Líneas 46-48
15. Línea 49

Cada uno de estos bloques es correlacionado en un grafo de control de flujo con el objetivo de conocer todas las posibles rutas por las cuales el algoritmo puede ejecutarse. La figura E.3 ilustra el grafo de ejecución por bloques del algoritmo.

Debido a lo anterior, se filtran las rutas a probar bajo las siguientes razones:

Variable	Valores
pmTasks	{Tarea 1, Tarea 2}
pmArtifacts	{Artefacto 1, Artefacto 2, Artefacto 3, Artefacto 4}
pmRoles	{Rol 1, Rol 2, Rol 3, Rol 4, Rol 5}
pmTasks	{Tarea 1, Tarea 2}
pmTasks first artifactOutputs	{Artefacto 1, Artefacto 2}
pmTasks second artifactOutputs	{Artefacto 3}
pmArtifacts first performerRoles	{Rol 1, Rol 2, Rol 3, Rol 4}
pmArtifacts first responsibleRoles	{Rol 1}
pmArtifacts second performerRoles	{Rol 1, Rol 2, Rol 3, Rol 4}
pmArtifacts second responsibleRoles	{∅}
pmArtifacts third performerRoles	{Rol 1, Rol 2, Rol 3}
pmArtifacts third responsibleRoles	{∅}
pmArtifacts fourth performerRoles	{∅}
pmArtifacts fourth responsibleRoles	{ Rol 5 }
pmTasks first artifactOutputs first performerRoles	{Rol 1, Rol 2, Rol 3, Rol 4}
pmTasks first artifactOutputs first responsibleRoles	{Rol 1}
pmTasks first artifactOutputs second performerRoles	{Rol 1, Rol 2, Rol 3, Rol 4}
pmTasks first artifactOutputs second responsibleRoles	{∅}
pmTasks second artifactOutputs first performerRoles	{Rol 1, Rol 2, Rol 3}
pmTasks second artifactOutputs first responsibleRoles	{∅}

**Tabla E.1:** Depuración de variables del código de importación para la visualización del patrón de roles responsables.

- **Ignorar rutas con colecciones nulas o vacías:** no es del interés de la prueba tener en cuenta colecciones vacías para descartar el comportamiento del algoritmo debido a que los ciclos se encargan de ignorar los bloques de código de las colecciones que cumplen esta característica
- **Ignorar rutas que no aportan a los casos de prueba:** no es del interés de la prueba ejecutar rutas que no cumplan con la verificación de todos los casos que puede tomar el patrón / plano ni evaluar modelos de proceso que no tengan este mismo objetivo.

Así, las rutas filtradas quedan de la siguiente manera:

R1. 1, 2, 3, 5, 6, 7, [8, 10, 11, 12, 13], 14, 15

Las variables relevantes para la visualización y resultados finales del código para el ejemplo escogido están reflejadas en la tabla E.1.

Los resultados del algoritmo demuestran la correcta organización de las relaciones y de la extracción de todos los elementos de proceso que están en el modelo de proceso de ejemplo documentado en EPF.

### E.0.3.2. Resultados del algoritmo de visualización del patrón de roles responsables

Una vez realizada la importación y resolución de relaciones de los elementos de proceso involucrados en la adecuada visualización del rol se procede a depurar

el algoritmo de visualización respectivo (función `viewArtifactResponsibleRolesOn: aView`).

Así mismo, se generan los bloques de código respectivos para generar el grafo de control de flujo:

1. Líneas 3-9
2. Líneas 10-11
3. Línea 12
4. Líneas 13-14
5. Línea 15
6. Líneas 16-17
7. Línea 18
8. Línea 19
9. Líneas 20-40

La figura [E.5](#) ilustra el grafo de control de flujo de la visualización del patrón de roles responsables.

Las rutas respectivas de los grafos donde el ejemplo de modelo de proceso se ejecuta en el código de visualización de roles responsables son:

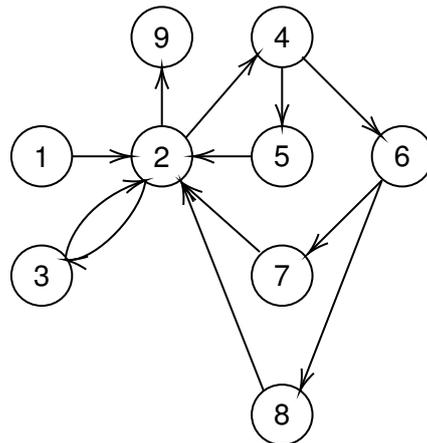
- R.1. 1, 2, 3, 2, 9
- R.2. 1, 2, 4, 5, 2, 9
- R.3. 1, 2, 4, 6, 7, 2, 9
- R.4. 1, 2, 4, 6, 8, 2, 9

Tras la ejecución de las rutas respectivas, el contenido de cada una de las variables de interés para la visualización en Mondrian se visualiza en la tabla [E.2](#).

Los resultados del algoritmo de visualización del patrón de roles responsables demuestran la correcta distribución de colores de cada elemento del modelo de proceso que está involucrado en el patrón de roles responsables.

```
1 viewArtifactResponsibleRolesOn: aView
2 | coll mondrian legend |
3 mondrian := RTMondrian new.
4 mondrian view: aView.
5 mondrian interaction highlightNeighbors: #outputArtifacts color: Color orange.
6 coll := (self sorted: [:m1 :m2 | m1 numberOfInputTasks < m2 numberOfInputTasks]) asArray.
7 mondrian shape rectangle
8   borderColor: Color black;
9   borderWidth: 1;
10  fillColor: [: each |
11    ((each responsibleRoles size = 0) & (each performerRoles size >= 4))
12    ifTrue: [Color red]
13    ifFalse: [
14      ((each performerRoles size) >= 4)
15      ifTrue: [Color purple ]
16      ifFalse: [
17        ((each responsibleRoles size) = 0)
18        ifTrue: [Color blue ]
19        ifFalse: [Color white ].
20      ].
21    ].
22 ];
23 width: [:each | each numberOfOutputTasks * 10 ];
24 height: [:each | each numberOfInputTasks * 10 ].
25 "text: [:each | each presentationName ]."
26 mondrian nodes: coll.
27 mondrian shape arrowedLine color: Color gray trans; withShorterDistanceAttachPoint.
28 mondrian edges source: self connectFrom: #yourself toAll: #outputArtifacts.
29 mondrian layout tree.
30 mondrian build.
31 legend := RTLegendBuilder new.
32 legend view: mondrian view.
33 legend addText: 'WorkProduct responsible roles'.
34 legend addRectanglePolymetricWidth: '#TasksAsOutput' height: '#TasksAsInput' box: 'WorkProduct'.
35 legend addColor: Color red text:'WorkProducts without roles in charge of and 4 roles or more
36 realizing / modifying it'.
37 legend addColor: Color purple text:'WorkProducts with 4 roles or more realizing / modifying it and
38 roles in charge of'.
39 legend addColor: Color blue text:'WorkProducts without roles in charge of and 3 or less roles
40 realizing / modifying it'.
41 legend addColor: Color white text:'WorkProducts with 3 or less roles realizing / modifying it and roles
42 in charge of'.
43 legend addLineColor: Color gray text:'Artefact Dependence'.
44 legend build
```

Figura E.4: Código de visualización del patrón de roles responsables



**Figura E.5:** Grafo de control de flujo del algoritmo de visualización del patrón de roles responsables.

Ruta	Variable	Valores
R.2.	each	{Artefacto 1}
	fillColor:	Color purple
R.1.	each	{Artefacto 2}
	fillColor:	Color red
R.3.	each	{Artefacto 3}
	fillColor:	Color blue
R.4.	each	{Artefacto 4}
	fillColor:	Color white

**Tabla E.2:** Depuración de variables del código de la visualización del patrón de roles responsables.