

MODELO DE ARQUITECTURA EMPRESARIAL PARA
LA EMPRESA MINSTITUTO.COM UTILIZANDO
ARCHIMATE

José Daniel Peña

febrero 20, 2016
Creatics

Universidad Distrital Francisco José de Caldas



Facultad de Ingeniería
Ingeniería de Sistemas

Trabajo de Grado

**MODELO DE ARQUITECTURA EMPRESARIAL PARA LA EMPRESA
MINSTITUTO.COM UTILIZANDO ARCHIMATE**

José Daniel Peña
20001020109

<i>Director</i>	Sandro Bolaños Castro Universidad Distrital Francisco José de Caldas Ingeniería de Sistemas
<i>Revisor</i>	Julio Barón Velandia Universidad Distrital Francisco José de Caldas Ingeniería de Sistemas
<i>Jurados</i>	Jurado 1 y Jurado 2

febrero 20, 2016

José Daniel Peña

MODELO DE ARQUITECTURA EMPRESARIAL PARA LA EMPRESA MINSTITUTO.COM UTILIZANDO ARCHIMATE

Trabajo de Grado, febrero 20, 2016

Director: Sandro Bolaños Castroph.D.

Revisor: Julio Barón Velandia

Jurados: Jurado 1 y Jurado 2

Universidad Distrital Francisco José de Caldas

Facultad de Ingeniería

Ingeniería de Sistemas

Cra. 8 # 40-62

Bogotá D.C.

110231

Dedicado a María P.

AGRADECIMIENTOS

RESUMEN

El objetivo del trabajo de grado es el desarrollo de la arquitectura empresarial para la empresa Creatics y su producto Minstituto. La arquitectura empresarial se define como una metodología que a través de una visión integral de la organización realiza la alineación de la filosofía organizacional con los diferentes procesos al interior de la empresa, identificando la interrelaciones entre los mismos, teniendo en cuenta la estructura de ADM (Architecture Development Method), permitiendo que los procesos estratégicos y de apoyo tengan un horizonte definido formalmente.

Un instrumento fundamental para el desarrollo del trabajo de grado fue Archimate, que es un lenguaje gráfico por medio del cual se representa la arquitectura empresarial que soporta la operación de la empresa. Con esta herramienta se realizó la descripción de los diferentes puntos de vista que abarcan toda la operación generando la visión global de la misma y su respectiva alineación con los objetivos estratégicos.

Palabras Clave: Archimate, Arquitectura, Patrones, Colosoft, Creatics

ABSTRACT

The aim of this thesis is the development of enterprise architecture for the Creatics company and its product Minstituto. Enterprise Architecture is defined as a methodology in which through a holistic view of the organization the alignment of organizational philosophy with different processes within the company is performed, identifying the relationships between them, taking into account the ADM structure (Architecture Development Method), enabling that strategic and support processes have a formally defined horizon.

An essential tool for the development of the thesis was ArchiMate, which is a graphical language whereby enterprise architecture that supports the operation of the company is represented. With this tool the different views covering the entire operation were described, generating a global view of the organization and its alignment with the strategic objectives.

Índice general

I	Contextualización	3
1	Descripción del proyecto	5
1.1	Definición del problema	6
1.2	Formulación del problema	6
1.3	Objetivos	6
1.3.1	Objetivo General	6
1.3.2	Objetivos Específicos	6
1.4	Alcance	6
2	Presentación de la organización	7
2.1	Filosofía Organizacional	8
2.1.1	Misión	8
2.1.2	Visión	8
2.1.3	Objetivos estratégicos	8
2.1.4	Principios	8
2.1.5	Valores	9
2.2	Estructura Organizacional	9
2.3	Mapa de Procesos	9
3	Metodología y Cronograma de trabajo	11
3.1	Metodología SCRUM	12
3.1.1	Visión General	12
3.1.2	Teoría de Scrum	13
3.1.3	Eventos de Scrum	14
3.1.4	El equipo Scrum	15
3.2	Cronograma	16
4	Arquitectura Empresarial	17
4.1	Arquitectura empresarial	18
4.2	Componentes de una Arquitectura Empresarial	19
4.2.1	Arquitectura de negocio	19
4.2.2	Arquitectura de información	19
4.2.3	Arquitectura de sistemas de información o aplicaciones	20
4.2.4	Arquitectura tecnológica	20

5	Framework y Modelado	21
5.1	The Open Group	22
5.2	TOGAF	22
5.2.1	¿Qué es Arquitectura para TOGAF?	23
5.2.2	Arquitectura soportada por TOGAF	23
5.2.3	Método de desarrollo de la Arquitectura (ADM)	23
5.3	Archimate 2.1	27
5.3.1	Versiones	27
5.3.2	Conceptos centrales	29
5.3.3	Colaboración e interacción	30
5.3.4	Relaciones	31
5.3.5	Capas	31
5.3.6	Marco de referencia	32
5.3.7	Motivación	33
5.3.8	Implementación y Migración	34
5.3.9	Archimate: Su relación con TOGAF	35
6	Patrones	37
6.1	Programación Orientada a objetos	38
6.2	Patrones de software	38
6.2.1	Definiciones	39
6.2.2	Clases de patrones software	39
6.3	Patrones de Diseño	40
6.3.1	Clasificación de los patrones de diseño	40
7	Colosoft	45
7.1	Descripción	46
II	Arquitectura Empresarial	47
8	Modelo de Negocio	49
8.1	Punto de Vista de la Organización	50
8.1.1	Metamodelo	50
8.1.2	Modelo mInstituto	51
8.2	Punto de Vista Cooperación de Actor	51
8.2.1	Metamodelo	52
8.2.2	Modelo mInstituto	53
8.3	Punto de Vista Función de Negocio	54
8.3.1	Metamodelo	54
8.3.2	Modelo mInstituto	55
8.4	Punto de Vista Proceso de Negocio	55
8.4.1	Metamodelo	56

8.4.2	Modelo mInstituto	57
8.5	Punto de Vista Cooperación de Proceso de Negocio	58
8.5.1	Metamodelo	58
8.5.2	Modelo mInstituto	59
8.6	Punto de Vista de Producto	60
8.6.1	Metamodelo	61
8.6.2	Modelo mInstituto	62
9	Capa de Aplicación	63
9.1	Punto de Vista Comportamiento de Aplicación	64
9.1.1	Metamodelo	64
9.1.2	Modelo mInstituto	65
9.2	Punto de Vista Cooperación de Aplicación	66
9.2.1	Metamodelo	67
9.2.2	Modelo mInstituto	68
9.3	Punto de Vista Estructura de Aplicación	68
9.3.1	Metamodelo	69
9.3.2	Modelo mInstituto	70
9.4	Punto de Vista de Uso de Aplicación	70
9.4.1	Metamodelo	71
9.4.2	Modelo mInstituto	72
10	Capa de Infraestructura	75
10.1	Punto de Vista de Infraestructura	76
10.1.1	Metamodelo	76
10.1.2	Modelo mInstituto	77
10.2	Punto de Vista Uso de Infraestructura	78
10.2.1	Metamodelo	79
10.2.2	Modelo mInstituto	79
10.3	Punto de Vista de Organización e Implementación	80
10.3.1	Metamodelo	81
10.3.2	Modelo mInstituto	82
10.4	Punto de Vista de Estructura de Información	83
10.4.1	Metamodelo	84
10.4.2	Modelo mInstituto	84
10.5	Punto de vista de Realización del Servicio	85
10.5.1	Metamodelo	86
10.5.2	Modelo mInstituto	86
10.6	Punto de Vista de Capas	87
10.6.1	Metamodelo	88
11	Capa Motivacional	91
11.1	Punto de Vista de Stakeholder	92

11.1.1	Metamodelo	92
11.1.2	Modelo mInstituto	93
11.2	Punto de Vista de Realización de Objetivos	93
11.2.1	Metamodelo	94
11.2.2	Modelo mInstituto	95
11.3	Punto de Vista de Contribución	96
11.3.1	Metamodelo	97
11.3.2	Modelo mInstituto	97
11.4	Punto de Vista de Principios	98
11.4.1	Metamodelo	99
11.4.2	Modelo mInstituto	99
11.5	Punto de Vista de Realización de Requerimientos	100
11.5.1	Metamodelo	101
11.5.2	Modelo mInstituto	101
11.6	Punto de Vista de Motivación	102
11.6.1	Metamodelo	103
11.6.2	Modelo mInstituto	103
12	Implementación y Migración	105
12.1	Punto de Vista de Proyecto	106
12.1.1	Metamodelo	107
12.1.2	Modelo mInstituto	107
12.2	Punto de Vista de Migración	108
12.2.1	Metamodelo	108
12.2.2	Modelo mInstituto	109
12.3	Punto de Vista de Migración e Implementación	109
12.3.1	Metamodelo	110
12.3.2	Modelo mInstituto	111
III	Conclusiones	113
	Conclusiones	115
	Trabajos Futuros	117
	Bibliografía	119
IV	Anexos	121
A	Patrones	123
A.1	Cadena de Responsabilidad	124
A.1.1	Patrón cadena de responsabilidad mInstituto	124

A.1.2	Clases cadena de responsabilidad	125
A.2	Fabrica Abstracta	126
A.2.1	Patrón Fabrica Abstracta mInstituto	126
A.2.2	Clases fabrica abstracta	127
A.3	Decorador	128
A.3.1	Patrón Decorador mInstituto	129
A.3.2	Clases Decorador	129
B	Resumen Simbología Archimate 2.1	131
B.1	Capa de Negocios	132
B.2	Capa de Aplicación	134
B.3	Capa de Tecnología	135
B.4	Relaciones	136
B.4.1	Estructurales	136
B.4.2	Dinámicas	137
B.4.3	Otras	137
B.5	Motivación	138
B.6	Implementación y Migración	139
C	A Software Architecture Proposal Artistic Engineering Environment -AEE	141

Índice de figuras

2.1	Organigrama Creatics	9
2.2	Mapa de Procesos	10
3.1	Visión general de Scrum	12
3.2	Cronograma	16
4.1	Componentes de Arquitectura Empresarial	19
5.1	El ciclo del Modelo de Desarrollo de la Arquitectura	25
5.2	Metamodelos y diferentes niveles de especificación	29
5.3	Conceptos básicos de ArchiMate	30
5.4	Colaboración e interacción	31
5.5	Estructura de la Arquitectura	32
5.6	Relación entre los elementos básicos de motivación en ArchiMate . . .	34
5.7	Relaciones entre la motivación, Core, Implementación y Migración . .	35
5.8	Correspondencia entre ArchiMate (incluyendo extensiones) y TOGAF .	36
7.1	Interfaz Coloso	46
8.1	Posición del punto de vista de organización conceptualmente y marco del punto de vista	50
8.2	Metamodelo Punto de vista organización	51
8.3	Modelo Punto de vista organización: minstituto	51
8.4	Posición del punto de vista cooperación de actor conceptualmente y marco del punto de vista	52
8.5	Metamodelo Punto de Vista Cooperación de Actor	53
8.6	Modelo Punto de Vista Cooperación de Actor: minstituto	53
8.7	Posición del punto de vista función de negocio conceptualmente y marco del punto de vista	54
8.8	Metamodelo Punto de Vista Función de Negocio	54
8.9	Modelo Punto de Vista Función de Negocio: minstituto	55
8.10	Posición del punto de vista proceso de negocio conceptualmente y marco del punto de vista	56
8.11	Metamodelo Punto de Vista Proceso de Negocio	57
8.12	Modelo Punto de Vista Proceso de Negocio: minstituto	57

8.13	Posición del punto de vista de cooperación de proceso conceptualmente y marco del punto de vista	58
8.14	Metamodelo Punto de Vista de Producto	59
8.15	Modelo Punto de Vista de Producto: minstituto	60
8.16	Posición del Punto de Vista de Producto conceptualmente y marco del punto de vista	61
8.17	Metamodelo Punto de Vista de Producto	61
8.18	Modelo Punto de Vista de Producto: minstituto	62
9.1	Posición del punto de vista comportamiento de aplicación conceptualmente y marco del punto de vista	64
9.2	Metamodelo punto de vista comportamiento de aplicación	65
9.3	Modelo punto de vista comportamiento de aplicación: minstituto	66
9.4	Posición del punto de vista cooperación de aplicación conceptualmente y marco del punto de vista	67
9.5	Metamodelo punto de vista cooperación de aplicación	67
9.6	Modelo Punto de Vista Cooperación de Aplicación: minstituto	68
9.7	Posición del punto de vista estructura de aplicación conceptualmente y marco del punto de vista	69
9.8	Metamodelo punto de vista estructura de aplicación	69
9.9	Modelo punto de vista estructura de aplicación: minstituto	70
9.10	Posición del punto de vista de Uso de Aplicación conceptualmente y marco del punto de vista	71
9.11	Metamodelo punto de vista de Uso de Aplicación	72
9.12	Modelo punto de vista de Uso de Aplicación: minstituto	73
10.1	Posición del punto de vista de infraestructura conceptual y marco del punto de vista	76
10.2	Metamodelo punto de vista de infraestructura	77
10.3	Modelo punto de vista de infraestructura: minstituto	78
10.4	Posición del punto de vista uso de infraestructura conceptualmente y marco del punto de vista	79
10.5	Metamodelo punto de vista uso de infraestructura	79
10.6	Modelo punto de vista uso de infraestructura: minstituto	80
10.7	Posición de la organización e implementación conceptualmente y marco del punto de vista	81
10.8	Metamodelo punto de vista de organización e implementación	82
10.9	Modelo punto de vista de organización e implementación: minstituto	83
10.10	Posición del punto de vista de estructura de información conceptualmente y marco del punto de vista	84
10.11	Metamodelo punto de vista de estructura de información	84
10.12	Modelo punto de vista de estructura de información: minstituto	85

10.13	Posición del punto de vista de realización del servicio conceptualmente y marco del punto de vista	85
10.14	Metamodelo punto de vista de estructura de realización del servicio . .	86
10.15	Modelo punto de vista de estructura de realización del servicio: minstituto	87
10.16	Posición del punto de vista de capas conceptualmente y marco del punto de vista	88
10.17	Metamodelo punto de vista de capas	89
11.1	Posición del punto de vista de stakeholders conceptualmente y marco del punto de vista	92
11.2	Metamodelo punto de vista de stakeholders	93
11.3	Modelo punto de vista de stakeholders: minstituto	93
11.4	Posición del punto de vista de la realización de objetivos conceptualmente y marco del punto de vista	94
11.5	Metamodelo punto de vista de la realización de objetivos	95
11.6	Modelo punto de vista de la realización de objetivos: minstituto	96
11.7	Posición del punto de vista de la contribución conceptualmente y marco del punto de vista	97
11.8	Metamodelo punto de vista de la contribución	97
11.9	Modelo punto de vista de la contribución: minstituto	98
11.10	Posición del punto de vista de los principios conceptualmente y marco del punto de vista	99
11.11	Metamodelo punto de vista de los principios	99
11.12	Modelo punto de vista de los principios: minstituto	100
11.13	Posición del punto de vista de realización de requerimientos conceptualmente y marco del punto de vista	101
11.14	Metamodelo punto de vista de realización de requerimientos	101
11.15	Modelo punto de vista de realización de requerimientos: minstituto . .	102
11.16	Posición del punto de vista de la motivación conceptualmente y marco del punto de vista	103
11.17	Metamodelo punto de vista de la motivación	103
11.18	Modelo punto de vista de la motivación: minstituto	104
12.1	Posición del punto de vista del proyecto conceptualmente y marco del punto de vista	106
12.2	Metamodelo punto de vista del proyecto	107
12.3	Modelo punto de vista del proyecto: minstituto	107
12.4	Posición del punto de vista de migración conceptualmente y marco del punto de vista	108
12.5	Metamodelo punto de vista de migración	109
12.6	Modelo punto de vista de migración: minstituto	109

12.7	Posición del punto de vista de la implementación y migración conceptualmente y marco del punto de vista	110
12.8	Metamodelo punto de vista de la implementación y migración	110
12.9	Modelo punto de vista de la implementación y migración: minstituto	111
A.1	Patrón cadena de responsabilidad mInstituto	124
A.2	Patrón fabrica abstracta mInstituto	126
A.3	Patrón decorador mInstituto	128

Índice de cuadros

5.1	Tipos de la arquitectura soportados por TOGAF	23
5.2	Actividades del Método de Desarrollo de la arquitectura por Fase . . .	26
6.1	Catalogo de Patrones de Diseño	43
8.1	Descripción Punto de Vista de la Organización	50
8.2	Descripción Punto de Vista Cooperación de Actor	52
8.3	Descripción Punto de Vista Función de Negocio	54
8.4	Descripción punto de vista proceso de negocio	56
8.5	Descripción punto de vista de cooperación de proceso	58
8.6	Descripción Punto de Vista de Producto	60
9.1	Descripción punto de vista comportamiento de aplicación	64
9.2	Descripción punto de vista cooperación de aplicación	67
9.3	Descripción punto de vista estructura de aplicación	69
9.4	Descripción punto de vista de Uso de Aplicación	71
10.1	Descripción punto de vista de Infraestructura	76
10.2	Descripción punto de vista uso de infraestructura	78
10.3	Descripción punto de vista de organización e implementación	81
10.4	Descripción punto de vista de estructura de información	83
10.5	Descripción punto de vista de realización del servicio	85
10.6	Descripción punto de vista de capas	88
11.1	Descripción punto de vista de stakeholders	92
11.2	Descripción punto de vista de la realización de objetivos	94
11.3	Descripción punto de vista de la contribución	96
11.4	Descripción punto de vista de los principios	98
11.5	Descripción punto de vista de realización de requerimientos	100
11.6	Descripción punto de vista de la motivación	102
12.1	Descripción punto de vista del proyecto	106
12.2	Descripción punto de vista de migración	108
12.3	Descripción punto de vista de la implementación y migración	110
B.1	Simbología capa de Negocios	133
B.2	Simbología capa de Aplicación	134

B.3	Simbología capa de Tecnología	136
B.4	Simbología de Relaciones Estructurales	137
B.5	Simbología de Relaciones Dinámicas	137
B.6	Simbología de otras Relaciones	138
B.7	Simbología capa de Motivación	138
B.8	Simbología capa de Implementación y Migración	139

INTRODUCCIÓN

Con el trabajo desarrollado se elaboró una propuesta de Arquitectura empresarial para el producto mistituto.com de la empresa Creatics la cual se dedica al desarrollo de software. Este proceso se generó a través de la identificación de las características organizacionales y la representación de sus componentes y sus relaciones de forma integral utilizando el lenguaje Archimate.

La Arquitectura Empresarial es una metodología que describe formalmente el sistema visualizando de forma global los elementos de las organizaciones, su relación en la consecución de los objetivos estratégicos y su evolución en el tiempo. Archimate es un lenguaje de modelamiento de Arquitectura empresarial que mediante un conjunto de símbolos y estructuras gráficas permite representar la arquitectura empresarial.

El documento se encuentra dividido en cuatro partes; la primera parte contiene la contextualización, la cual incluye la descripción y filosofía organizacional de la empresa y la conceptualización. En la segunda parte se desarrolla la arquitectura empresarial. En la tercera parte se presentan las conclusiones y por último la cuarta parte relaciona referencias bibliográficas.

Parte I

Contextualización

Descripción del proyecto

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

1.1	Definición del problema	6
1.2	Formulación del problema	6
1.3	Objetivos	6
1.3.1	Objetivo General	6
1.3.2	Objetivos Específicos	6
1.4	Alcance	6

1.1 Definición del problema

Una de las mayores ventajas competitivas en una empresa es que la gestión de todos sus procesos y componentes estén alineados con su estrategia empresarial y su filosofía corporativa, sin embargo, realizar este proceso requiere una metodología que asegure que su planteamiento es efectivo.

Creatics es una empresa en proceso de creación que requiere establecer estrategias de competitividad y sostenibilidad que la hagan desde el inicio una empresa sólida.

1.2 Formulación del problema

¿Es la Arquitectura Empresarial la metodología que permita alinear los procesos y componentes del producto mistituto.com de la empresa Creatics con el cumplimiento de sus objetivos estratégicos?

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar una Arquitectura Empresarial para el producto mistituto.com de la empresa Creatics utilizando como herramienta el lenguaje de Arquitectura Archimate.

1.3.2 Objetivos Específicos

- Identificar la filosofía y características organizacionales de Creatics.
- Establecer actores, roles y stakeholders dentro de los diferentes procesos.
- Determinar los procesos, datos, aplicaciones, infraestructura tecnológica y demás componentes para la modelación.

1.4 Alcance

Se desarrollará la Arquitectura Empresarial para el producto minstituto.com de la empresa Creatics utilizando el lenguaje de arquitectura Archimate.

Presentación de la organización

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

2.1	Filosofía Organizacional	8
2.1.1	Misión	8
2.1.2	Visión	8
2.1.3	Objetivos estratégicos	8
2.1.4	Principios	8
2.1.5	Valores	9
2.2	Estructura Organizacional	9
2.3	Mapa de Procesos	9

2.1 Filosofía Organizacional

2.1.1 Misión

Generamos soluciones tecnológicas que promueven el progreso de nuestros clientes contando con un equipo de trabajo altamente competitivo.

2.1.2 Visión

Ser reconocidos por nuestros productos innovadores principalmente en el sector de instituciones educativas.

2.1.3 Objetivos estratégicos

- Proveer soluciones tecnológicas que permitan optimizar la gestión de la información en las instituciones educativas, generando renta para nuestros socios.
- Generar reconocimiento en el sector de instituciones educativas.

2.1.4 Principios

Compromiso Orientamos nuestros esfuerzos al cumplimiento de las metas establecidas.

Innovación Buscamos generar valor agregado a nuestros productos que permitan incrementar los beneficios a nuestros clientes.

Enfoque al cliente Nuestros procesos se orientan en ofrecer el mejor servicio siendo respaldo para nuestros clientes.

Confidencialidad Nuestros clientes cuentan con la garantía de que su información no será divulgada y tendrán un manejo apropiado de la información de su organización.

Responsabilidad social y ambiental Buscamos beneficiar a nuestros colaboradores y a la sociedad al igual que el medio ambiente por medio de nuestros procesos internos.

2.1.5 Valores

Respeto Consiste en reconocer, aceptar y comprender los intereses y necesidades de todos los integrantes de la organización y de nuestros clientes.

Transparencia Generar procesos donde predomine la comunicación, la claridad y la honestidad.

Trabajo en equipo Buscamos integrar las habilidades y destrezas del equipo de profesionales en beneficio de nuestros clientes.

2.2 Estructura Organizacional

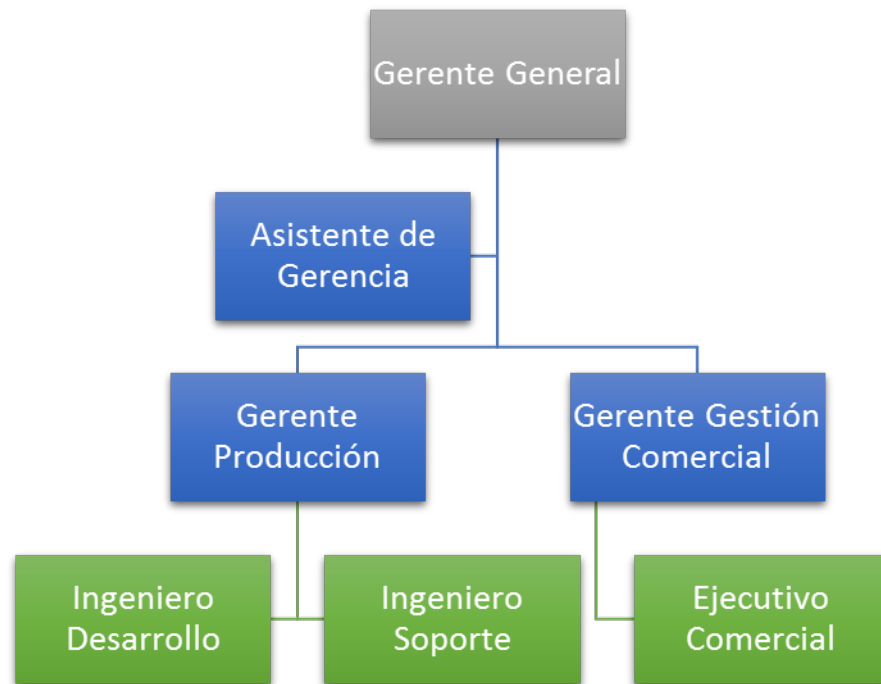


Fig. 2.1.: Organigrama Creatics

2.3 Mapa de Procesos

La organización con el fin de satisfacer las necesidades del cliente apoya su operación en tres procesos fundamentales:

Estratégicos La gestión de dirección y la mejora continua dentro de su funcionamiento establecen las políticas y lineamientos que permiten llevar a cabo la misión.

Misionales Estos procesos llevan a cabo las operaciones orientadas a la satisfacción de las necesidades de los clientes.

Apoyo Son los procesos que brindan soporte a la gestión de los procesos estratégicos y misionales.



Fig. 2.2.: Mapa de Procesos

Metodología y Cronograma de trabajo

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

3.1	Metodología SCRUM	12
3.1.1	Visión General	12
3.1.2	Teoría de Scrum	13
3.1.3	Eventos de Scrum	14
3.1.4	El equipo Scrum	15
3.2	Cronograma	16

3.1 Metodología SCRUM

3.1.1 Visión General

Scrum es un marco de trabajo en el que equipos cross-funcionales pueden crear productos o desarrollar proyectos de una forma iterativa e incremental. El desarrollo se estructura en ciclos de trabajo llamados Sprints (también conocidos como iteraciones). Estas iteraciones no deben durar más de cuatro semanas cada una (siendo dos semanas la duración más habitual) y tienen lugar una tras otra sin pausa entre ellas. Los Sprints están acotados en el tiempo – finalizan en una fecha determinada independientemente de si el trabajo ha finalizado por completo o no, y jamás se prorrogan. Normalmente los equipos Scrum escogen una duración de Sprint y la mantienen para todos sus Sprints hasta que mejoran y pueden emplear ciclos más cortos. Al principio de cada Sprint, un Equipo cross-funcional (de en torno a siete personas) selecciona elementos (peticiones del cliente) de una lista priorizada. El equipo acuerda un objetivo colectivo respecto a lo que creen que podrán entregar al final del Sprint, algo que sea tangible y que estará “terminado” por completo. Durante el Sprint no se podrán añadir nuevos elementos; Scrum se adapta a los cambios en el siguiente Sprint, pero el pequeño Sprint actual está pensado para concentrarnos en un objetivo pequeño, claro y relativamente estable. Todos los días el Equipo se reúne brevemente para inspeccionar su progreso y ajustar los siguientes pasos necesarios para completar el trabajo pendiente. Al final del Sprint, el Equipo revisa el Sprint con los diferentes Stakeholders (interesados e involucrados en el producto) y realiza una demostración de lo que han desarrollado. Se obtiene feedback que podrá ser incorporado en el siguiente Sprint. Scrum enfatiza un producto “funcionando” al final del Sprint que esté realmente “terminado”. En el caso del software, esto significa un sistema que está integrado, testado, con la documentación de usuario generada y potencialmente entregable.

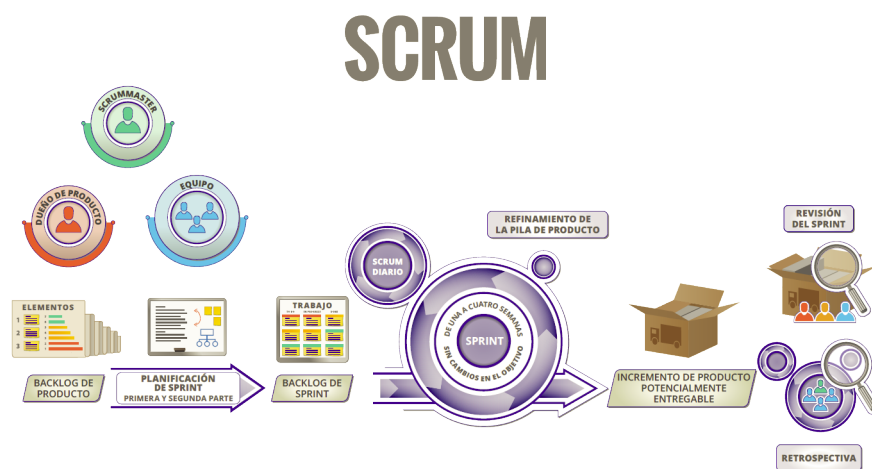


Fig. 3.1.: Visión general de Scrum

3.1.2 Teoría de Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce, Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

Transparencia Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.

Inspección Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

Adaptación Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación y son los siguientes.

- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

3.1.3 Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar las vitales transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse.

El Sprint

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Revisión de Sprint

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Lista de Producto durante el Sprint.

Retrospectiva de Sprint

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint. La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes.

3.1.4 El equipo Scrum

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son autoorganizados y multifuncionales. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto terminado aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

El Dueño de Producto

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Equipo de Desarrollo

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto terminado, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento. Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

3.2 Cronograma

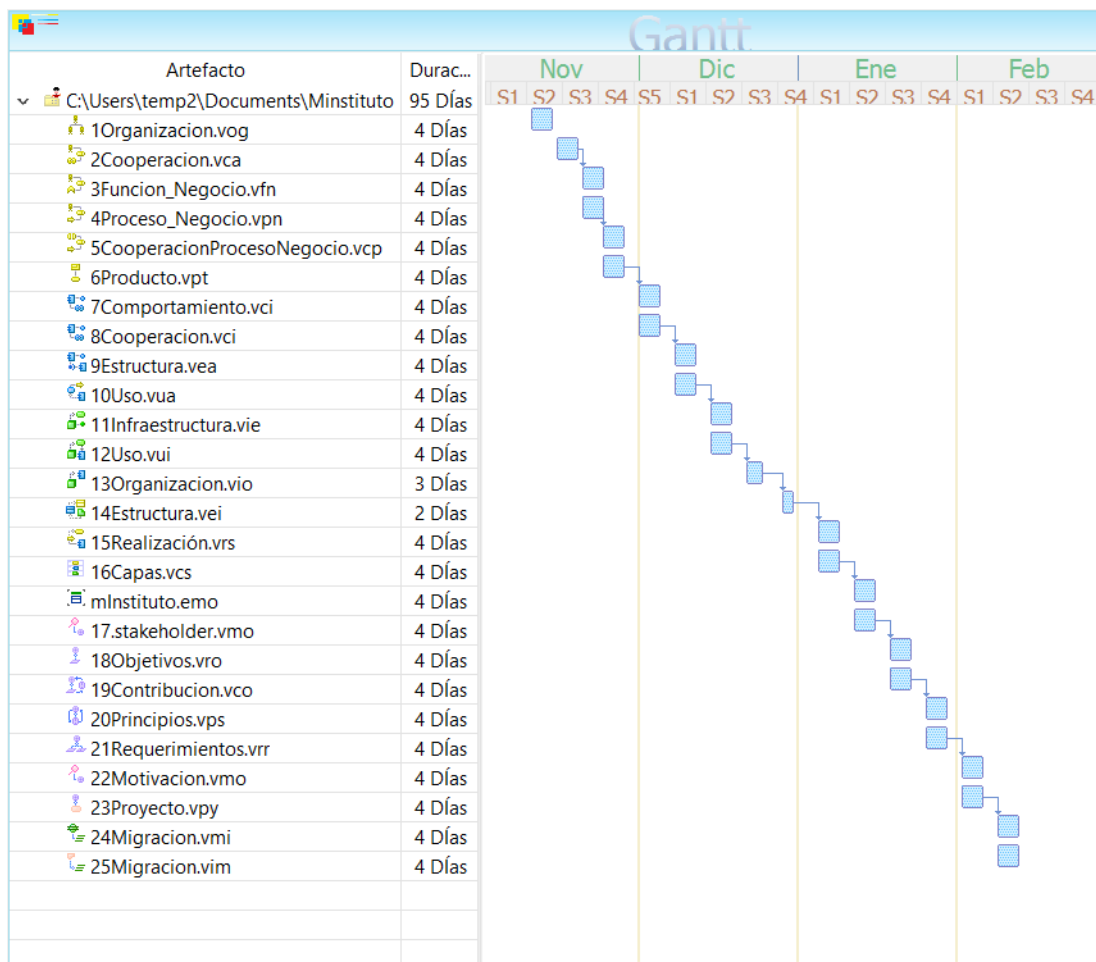


Fig. 3.2.: Cronograma

Arquitectura Empresarial

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

4.1	Arquitectura empresarial	18
4.2	Componentes de una Arquitectura Empresarial	19
4.2.1	Arquitectura de negocio	19
4.2.2	Arquitectura de información	19
4.2.3	Arquitectura de sistemas de información o aplicaciones .	20
4.2.4	Arquitectura tecnológica	20

4.1 Arquitectura empresarial

La definición de arquitectura empresarial surge frente a la necesidad de alinear las tecnologías de información a los objetivos estratégicos del negocio. Algunas definiciones de arquitectura empresarial se presentan a continuación:

- *IEEE Std. 1471-2000:*
“... organización fundamental de un sistema, compuesta por sus componentes, las relaciones entre ellos y su ambiente y los principios que gobiernan su diseño y evolución”.
- *The Open Group Architecture Framework:*
“... la arquitectura empresarial se puede definir de dos posibles formas dependiendo del contexto en que se utilice 1) una descripción formal de un sistema o un plan detallado de un sistema a nivel de sus componentes para guiar su implementación; o 2) una estructura de componentes, sus interrelaciones, y los principios y guías que gobiernan su diseño y evolución en el tiempo”.
- *International Enterprise Architecture Institute:*
“El análisis y documentación de una organización en su estado actual y futuro desde las perspectivas de negocio, tecnología y estrategias integradas”.
- *Federal Enterprise Architecture Framework, 1ra versión – 1999:* “... las arquitecturas empresariales son modelos que se aplican de manera sistemática y completa para definir el ámbito presente o futuro de una organización. Arquitecturas empresariales son esenciales para la evolución y desarrollo de nuevos sistemas de información que optimicen el valor de la misión de una organización...”
- *Gartner Research:*
“Una arquitectura empresarial es un proceso de planeamiento estratégico que traduce la visión y estrategias de negocio de una organización en un efectivo plan de cambio empresarial”.

En conclusión arquitectura empresarial es alinear los objetivos estratégicos de una organización con tecnología, describiendo el estado actual de la organización y proyectándonos a una visión futura con la implementación de nuevas tecnologías.

4.2 Componentes de una Arquitectura Empresarial

Los diferentes framework de arquitectura empresarial realizan un planteamiento de los componentes o dominios de arquitectura que son los elementos que definen el funcionamiento de una empresa. En la figura 1.5 se presentan los componentes de arquitectura empresarial: Arquitectura de negocio, arquitectura de información, arquitectura de sistemas de información y arquitectura tecnológica.



Fig. 4.1.: Componentes de Arquitectura Empresarial

4.2.1 Arquitectura de negocio

Para Ralph Whittle, “La primera vista representa la arquitectura de negocio, la cual se encarga de la descripción de la estructura organizacional, de los procesos de negocio, los sistemas de planeación y control, los mecanismos de gobierno y administración de políticas y procedimientos en el entorno empresarial. Esta vista de arquitectura es la que refleja el valor del negocio obtenido de las sinergias y resultados que se producen desde las otras vistas de arquitectura que le preceden. La arquitectura de negocio recibe como insumo principal el plan estratégico de la empresa, los lineamientos corporativos, los indicadores de gestión, y se nutre de la misión, la visión, las estrategias y los objetivos corporativos. Las estrategias y objetivos de alto nivel los traducen en requerimientos que son relevantes para el negocio”.

4.2.2 Arquitectura de información

Para Richard Wurman, “La segunda vista representa la arquitectura de información, la cual describe los activos lógicos y físicos de los datos como un activo de la empresa, y la administración de los recursos de información; esta perspectiva muestra cómo

los recursos de información están siendo administrados, compartidos y utilizados por la organización.

La arquitectura de información es una disciplina que organiza conjuntos de información, permitiendo que cualquier persona los entienda y los integre a su propio conocimiento de manera simple. La construcción de una arquitectura de información requiere el levantamiento de un inventario de los objetos de negocio que representan los activos de información que están disponibles y que son utilizados por la organización. La información juega un rol fundamental para el funcionamiento de los sistemas de información y de los proceso de negocio”

4.2.3 Arquitectura de sistemas de información o aplicaciones

Para Richard Wurman, “La tercera vista representa la Arquitectura de sistemas de información que incorpora soluciones aplicativos que apoyan al negocio basadas en las capacidades funcionales requeridas y las estrategias de tecnología definidas, e identifica componentes y servicios que den respuesta a necesidades comunes de las áreas de negocio. La arquitectura aplicativa define qué clase de aplicaciones son relevantes para la empresa y lo que estas aplicaciones necesitan para gestionar los datos y presentar la información”.

4.2.4 Arquitectura tecnológica

Según Jaap Schekkerman, “La arquitectura técnica define la estrategia y arquitectura tecnológica en la infraestructura de TI, y el marco tecnológico de las plataformas computacionales y bases de datos que deben soportar las distintas soluciones del negocio, así como los mecanismos de almacenamiento de los datos e información, las redes de datos, los centros de procesamiento de datos y los servicios integrados de tecnología”.

Framework y Modelado

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

5.1	The Open Group	22
5.2	TOGAF	22
5.2.1	¿Qué es Arquitectura para TOGAF?	23
5.2.2	Arquitectura soportada por TOGAF	23
5.2.3	Método de desarrollo de la Arquitectura (ADM)	23
5.3	Archimate 2.1	27
5.3.1	Versiones	27
5.3.2	Conceptos centrales	29
5.3.3	Colaboración e interacción	30
5.3.4	Relaciones	31
5.3.5	Capas	31
5.3.6	Marco de referencia	32
5.3.7	Motivación	33
5.3.8	Implementación y Migración	34
5.3.9	Archimate: Su relación con TOGAF	35

5.1 The Open Group

The Open Group es un consorcio de la industria del software que provee estándares abiertos neutrales para la infraestructura de la informática. Fue formado a partir de la fusión de X/Open con OSF en 1996. The Open Group es muy famoso por sus sistemas de certificación de la marca UNIX; en el pasado el grupo fue reconocido por publicar el artículo Single UNIX Specification, el cual extiende los estándares de POSIX y es la definición oficial del sistema operativo conocido como UNIX. Sus miembros incluyen un conjunto de empresas y agencias gubernamentales, como por ejemplo Capgemini, Fujitsu, Hitachi, HP, IBM, NEC, Departamento de Defensa de Estados Unidos, NASA y otros.

5.2 TOGAF

TOGAF es un marco de referencia de arquitectura. En términos simples, TOGAF es una herramienta para asistir en la aceptación, creación, uso, y mantenimiento de arquitecturas. Está basado en un modelo iterativo de procesos apoyado por las mejores prácticas y un conjunto reutilizable de activos arquitectónicos existentes.

TOGAF es desarrollado y mantenido por el Foro de Arquitectura de The Open Group. La primera versión de TOGAF, desarrollada en 1995, se basó en el Marco de Referencia de Arquitectura Técnica para la Gestión de la Información del Ministerio de Defensa Estadounidense (TAFIM por sus siglas en inglés). Comenzando con esta sólida fundación, el Foro de Arquitectura de The Open Group ha desarrollado versiones sucesivas de TOGAF con regularidad y ha publicado cada una en el sitio web público de The Open Group.

Según The Open Group, el 80 % de las grandes organizaciones a nivel mundial ha adoptado TOGAF como marco de referencia para sus Arquitecturas Empresariales; así mismo, decenas de miles de personas en todo el mundo han recibido formación y certificación en el marco del programa 'Open CA'.

Por otro lado, TOGAF se basa en modelos descriptivos que permiten definir la arquitectura desde diversos y estratégicos puntos de vista, para establecer a nivel general las necesidades, limitaciones y oportunidades del negocio.

5.2.1 ¿Qué es Arquitectura para TOGAF?

Uno de los conceptos clave para cualquier proceso de arquitectura empresarial es la comprensión misma de la Arquitectura, ya que ésta determina, en parte, el enfoque desde el cual se adoptará el modelo. En TOGAF, “Arquitectura” tiene dos significados según el contexto:

1. Una descripción formal de un sistema o un plano detallado del sistema al nivel de sus componentes para orientar su implementación.
2. La estructura de componentes, sus interrelaciones y los principios y guías que gobiernan su diseño y evolución a través del tiempo.

5.2.2 Arquitectura soportada por TOGAF

TOGAF cubre el desarrollo de cuatro tipos relacionados en la arquitectura. Estos cuatro tipos de arquitectura son comúnmente aceptados como subconjuntos de una arquitectura empresarial, los cuales TOGAF esta diseñado para soportar.

Tipo de Arquitectura	Descripción
Arquitectura de Negocio	La estrategia de negocio, gobierno, organización y procesos clave de la organización.
Arquitectura de datos	La estructura de datos lógicos y físicos que posee una organización y sus recursos de gestión de datos.
Arquitectura de aplicación	Un plano (blueprint en inglés) de las aplicaciones individuales a implementar, sus interacciones y sus relaciones con los procesos de negocio principales de la organización.
Arquitectura tecnológica	Las capacidades de software y hardware que se requieren para apoyar la implementación de servicios de negocio, datos y aplicación. Esto incluye infraestructura de IT, capa de mediación (middleware en ingles), redes, comunicaciones, procesamiento y estándares.

Tab. 5.1.: Tipos de la arquitectura soportados por TOGAF

5.2.3 Método de desarrollo de la Arquitectura (ADM)

El **ADM** describe cómo obtener una Arquitectura Empresarial que sea específica para la organización y para responder a los requerimientos del negocio. El ADM es el

componente principal de TOGAF y proporciona dirección a los arquitectos en varios niveles:

- Proporciona varias fases de desarrollo de arquitectura (Arquitectura de Negocio, Arquitecturas de Sistemas de Información, Arquitectura Tecnológica) en un ciclo, que sirve como una plantilla general de procesos para la actividad de desarrollo de la arquitectura.
- Proporciona una narrativa de cada fase de la arquitectura, describiendo la fase en términos de objetivos, enfoque, entradas, pasos a seguir, y salidas. Las secciones de entradas y salidas proporcionan una definición de la estructura del contenido de arquitectura y entregables (una descripción detallada de las entradas de la fase y las salidas de la fase se da en el Marco de Referencia del Contenido Arquitectónico).
- Proporciona resúmenes multi-fase que abordan también la Gestión de Requerimientos.

El ADM es el resultado de las contribuciones de numerosos profesionales de la arquitectura y constituye el núcleo de TOGAF. Es un método para obtener Arquitecturas Empresariales que son específicas para la organización, y está especialmente diseñado para responder a los requerimientos del negocio. El ADM describe:

- Un modo confiable y probado para desarrollar y utilizar una Arquitectura Empresarial
- Un método para desarrollar arquitecturas en diferentes niveles¹ (negocio, aplicaciones, datos, tecnología) que permiten al arquitecto asegurar que un conjunto complejo de requerimientos se aborden adecuadamente
- Un conjunto de guías y técnicas para el desarrollo de arquitectura

Fases del ADM

El ADM consiste en varias Fases que se desplazan cíclicamente a través de una serie de Dominios de Arquitectura y permiten al arquitecto asegurar que un conjunto complejo de requerimientos se aborden adecuadamente. La estructura básica del ADM se muestra en la Figura 2.

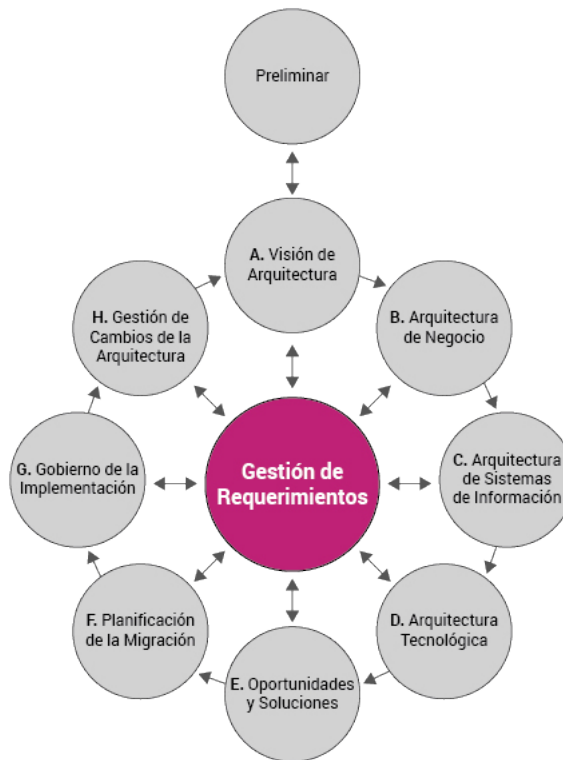


Fig. 5.1.: El ciclo del Modelo de Desarrollo de la Arquitectura

El ADM se aplica iterativamente durante todo el proceso, entre las diferentes Fases, y dentro de ellas. Durante todo el ciclo del ADM se debe realizar una validación frecuente de los resultados respecto a los requerimientos originales, tanto aquellos del ciclo completo del ADM como los de la Fase particular del proceso. Esta validación debe reconsiderar el alcance, los detalles, el plan y los hitos. Cada Fase debe considerar los activos producidos a partir de las iteraciones anteriores del proceso y los activos externos de mercado, así como otros marcos de referencia o modelos.

El ADM apoya el concepto de iteración en tres niveles:

- **Ciclo alrededor del ADM:** El ADM se presenta de manera circular indicando que la finalización de una Fase de trabajo en la arquitectura alimenta directamente las Fases subsecuentes de trabajo en la arquitectura.
- **Iteración entre Fases:** TOGAF describe el concepto de la iteración a través de Fases (por ejemplo, volviendo a la Arquitectura de Negocio posteriormente a la finalización de la Arquitectura Tecnológica).

- **Ciclo alrededor de una Fase individual:** TOGAF apoya la ejecución repetida de las actividades dentro de una Fase individual del ADM como una técnica para elaborar contenido arquitectónico.

Fase de ADM	Actividad
Gestión de Requerimientos	Cada etapa de un proyecto de TOGAF está basada en los requerimientos del negocio, incluyendo su validación. Los requerimientos se identifican, se almacenan y se gestionan al ingreso y egreso de las fases relevantes del ADM, las cuales eliminan, abordan, y priorizan los requerimientos.
A. Visión de arquitectura	Establece el alcance, las limitaciones y expectativas de un proyecto de TOGAF. Crea la Visión de la Arquitectura. Identifica a los Interesados. Valida el contexto de negocio y crea la Declaración de Trabajo de Arquitectura. Obtiene aprobaciones.
B. Arquitectura de Negocio C. Arquitecturas de sistemas de información D. Arquitectura tecnológica	Desarrolla arquitecturas en cuatro dominios: <ol style="list-style-type: none"> 1. Negocio 2. Sistemas de Información - Aplicaciones 3. Sistemas de Información - Datos 4. Tecnología <p>En cada caso, desarrolla la Arquitectura de la línea de base y de destino y analiza las brechas entre ambas.</p>
E. Oportunidades y soluciones	Realiza la planificación de la implementación inicial y la identificación de medios de entrega para los Bloques de Construcción identificados en las Fases anteriores. Determina si se requiere un enfoque incremental, y si así fuera, identifica las Arquitecturas de Transición.
F. Planificación de la migración	Desarrolla el Plan detallado de Implementación y Migración que aborda cómo moverse de la Arquitectura de la Línea de Base a la Arquitectura de Destino.
G. Gobierno de la Implementación	Proporciona supervisión arquitectónica para la implementación. Prepara y publica Contratos de Arquitectura. Asegura que el proyecto de implementación esté en conformidad con la arquitectura.
H. Gestión de cambios de la Arquitectura	Proporciona seguimiento continuo y un proceso de gestión de cambios para asegurar que la arquitectura responda a las necesidades de la empresa y que se maximice el valor de la arquitectura para el negocio.

Tab. 5.2.: Actividades del Método de Desarrollo de la arquitectura por Fase

5.3 Archimate 2.1

ArchiMate nace como un lenguaje de modelado de arquitecturas empresariales el cual tiene como objetivo proveer una representación uniforme de los diagramas que describen la arquitectura empresarial de una organización, permitiendo comprender las diferentes áreas o capas empresariales: estrategia, negocio, información, aplicaciones e infraestructura tecnológica; describiendo los diferentes dominios, relaciones, dependencias e incorporando el concepto de orientación a servicios.

El principal elemento en esta metodología es el servicio, que está definido como una unidad de funcionalidad que un actor (sistema o una organización) pone a disposición del ambiente de trabajo. Se adopta otro concepto ya existente: el concepto de arquitectura en capas. En una arquitectura orientada a servicios, cada capa proporciona servicios que pueden ser consumidos por capas de nivel superior, y cada capa utiliza los servicios proporcionados por capas de nivel inferior.

5.3.1 Versiones

Desde el año 2008 que la propiedad y los derechos de la arquitectura ArchiMate fueron transferidos al Open Group, por parte del consorcio de Universidades, empresas y el Gobierno holandés, se han publicado tres versiones de la arquitectura: ArchiMate 1.0, lanzada el año 2008 como un estándar técnico; ArchiMate 2.0, lanzada el año 2012 como un estándar; ArchiMate 2.1, en el año 2013, constituye la última versión lanzada como actualización a la versión 2.0, donde se considerando comentarios de la comunidad que aplicaban el estándar.

La primera versión, ArchiMate 1.0, fue considerada como ya un estándar formal técnico teniendo como base lo realizado por el equipo de desarrollo del estándar. Cabe recalcar que este equipo tomó como referencia a otro estándar el de la IEEE 1471, el mismo que es utilizado para describir el diseño de una arquitectura de software. Este estándar técnico describe ArchiMate como un lenguaje que complementa al marco de trabajo (framework) TOGAF, proporcionado un juego de conceptos y definiciones que permiten representar a través de un lenguaje unificado y de representación gráfica un diseño de arquitectura empresarial basado en el framework TOGAF. Se plantea el lenguaje ArchiMate como una correspondencia a las principales vistas que son definidas en la arquitectura TOGAF ADM (ADM - Architecture Development Method), específicamente no se hace uso de todas las vistas que menciona TOGAF, como se

ilustra en la Figura 1, éstas son alineadas a las tres capas que establece el estándar ArchiMate: Business, Application, Technology. La no coincidencia de todas las capas de ArchiMate con TOGAF denota que el lenguaje está basado en el framework, el cual sí amplía y considera aspectos mucho más profundos concernientes a la arquitectura empresarial, decisiones estratégicas y de dirección. ArchiMate ofrece más un estándar formal donde se aterrizan en un diseño y un lenguaje esquemas donde se permite desarrollar la arquitectura empresarial del cual se derivarán procedimientos más operativos.

ArchiMate 2.0, es lanzada en el año 2012 oficialmente como un estándar que conserva gran parte de su versión anterior (1.0) pero en esta ya se incluyen retroalimentaciones por parte de los usuarios del estándar, agregando así nuevas características. Esta nueva versión aparecen dos extensiones: Motivación, e Implementación y Migración. La primera considera aspectos de modelamiento de los interesados, manejos de cambios, objetivos del negocio, principios y requerimientos, se alinea principalmente a la fase inicial de las vistas de TOGAF. La extensión de Implementación y Migración, está diseñada para el manejo del portafolio del proyecto, análisis de brecha, transiciones y planes de migración, principalmente está enfocada a las fases finales de TOGAF. Ambas extensiones incluyen nuevos conceptos que dan atención a las inquietudes y experiencias de los practicantes del estándar en su versión anterior. Así tenemos los conceptos de Ubicación, que brinda un punto de vista conceptual que puede ser asignado a los elementos estructurales e indirectamente también determina sus comportamientos; y la de Función de Infraestructura, que modela el comportamiento interno de un nodo dentro de la capa de tecnología, esto permite mejorar la consistencia de la capa de tecnología con las otras dos capas (Negocios y Aplicación).

La más reciente versión de ArchiMate es la 2.1, lanzada el año 2013, esta conserva la gran mayoría de características de su versión anterior (2.0), el motivo de su actualización se debe al direccionamiento de los comentarios de los usuarios que por años han venido usando haciendo que los mismos sean considerados y han constituido de aportes para brindar de mayores detalles y aclaraciones a los aspectos definidos por el estándar.

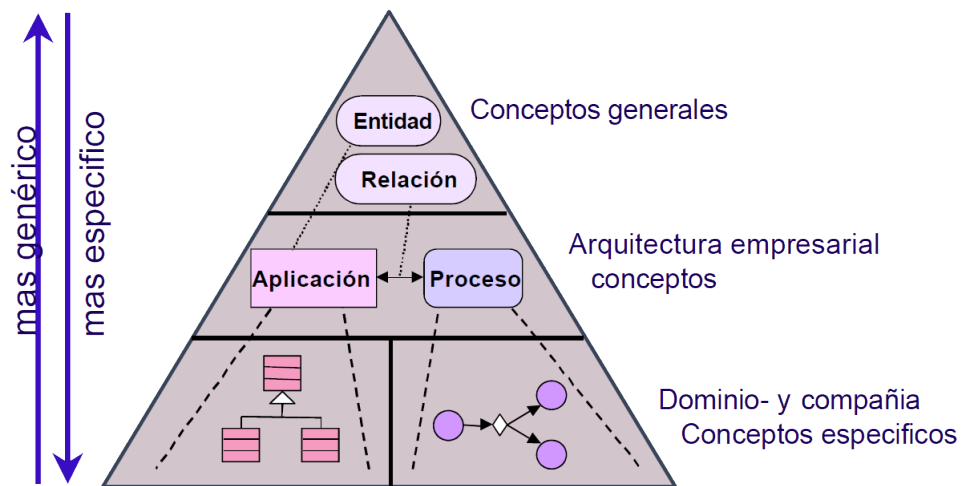


Fig. 5.2.: Metamodelos y diferentes niveles de especificación

5.3.2 Conceptos centrales

El lenguaje central consiste de tres tipos de elementos:

1. **Elementos de estructura activa** Es una entidad capaz de ejercer comportamientos, tales como los actores del negocio, componentes de aplicación
2. **Elementos de comportamiento** Es una unidad de actividad ejecutada por uno o más elementos de estructura activa.
3. **Elementos de estructura pasiva** Es un objeto sobre el cual se ejecuta un comportamiento.
4. **Servicio** Es una unidad de funcionalidad que el sistema provee.
5. **Interfaz** Es el punto de acceso donde uno o más servicios son hechos disponibles al entorno. Provee una vista externa sobre el proveedor de servicio y oculta su estructura interna.
6. Para nombrar los roles de las interrelaciones, se utiliza una convención similar a la de UML(pero usando verbos en lugar de sustantivos).
7. Si no se muestra cardinalidad alguna al final de un interrelación, se asume una 0..* (cero o más).

Estos elementos tienen como inspiración el lenguaje natural donde una sentencia tiene un sujeto (estructura activa), un verbo (comportamiento) y un predicado (estructura pasiva)

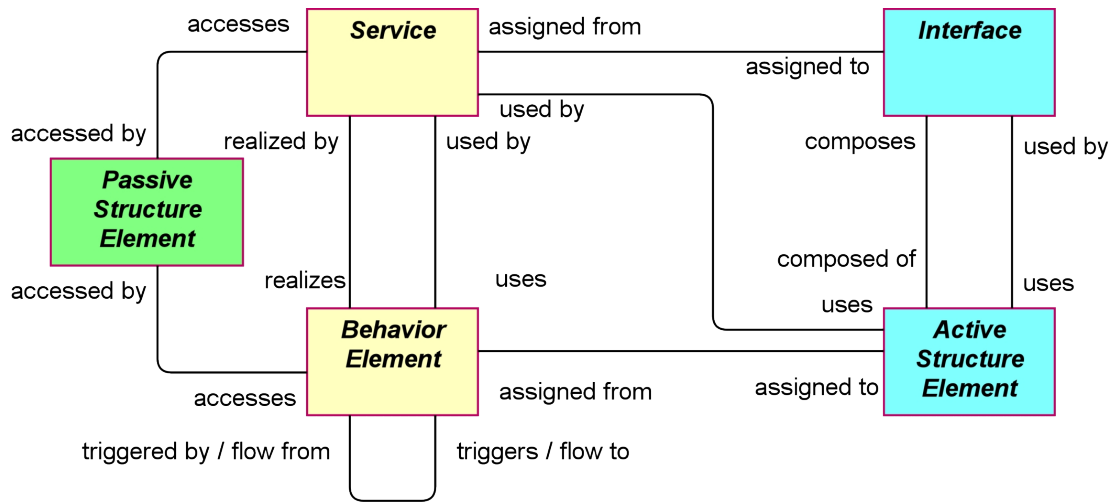


Fig. 5.3.: Conceptos básicos de ArchiMate

5.3.3 Colaboración e interacción

Si vamos a un nivel más profundo en la estructura del lenguaje, se distingue entre el comportamiento que es realizado por un elemento de estructura única (Ejemplo: actor, rol, componentes, etc.), o comportamiento colectivo (Interacción) que se realiza por una colaboración de varios elementos de la estructura.

Colaboración Es una (temporal) agrupación (o agregación) de dos o más elementos de estructura, trabajando juntos para realizar algún comportamiento colectivo. Este comportamiento colectivo puede ser modelado como una interacción.

Interacción Es una unidad de comportamiento llevada a cabo por una colaboración de dos o más elementos de estructura.

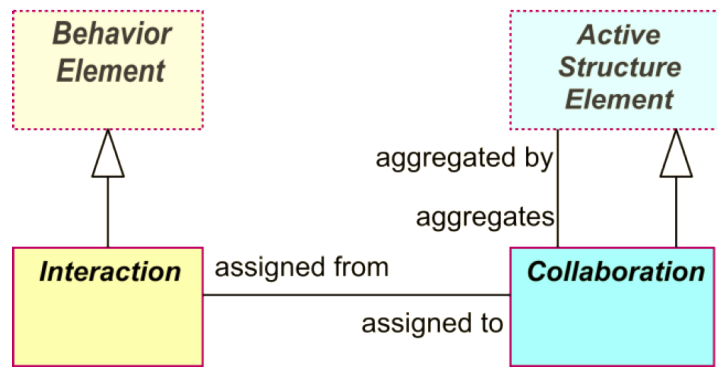


Fig. 5.4.: Colaboración e interacción

5.3.4 Relaciones

Al lado de los conceptos fundamentales antes reseñados, ArchiMate contiene un conjunto básico de las relaciones. Varias de estas relaciones han sido adoptadas a partir de conceptos de relación correspondientes que se producen en normas existentes. Relaciones tales como la composición, la agregación, asociación y especialización se toman de UML 2.0, mientras que la activación se utiliza en muchos procesos de negocio lenguajes de modelado.

5.3.5 Capas

En ArchiMate, hay tres diferentes capas para tres niveles diferentes en una arquitectura empresarial:

1. **Capa empresarial** La capa de negocio ofrece productos y servicios para clientes externos. Estos servicios están implementados internamente por los procesos de negocios y ejecutados por actores de negocio.
2. **Capa de aplicación** La capa de aplicación es compatible con la capa de negocio por servicios de aplicaciones implementados por software.
3. **Capa de Tecnología** La capa de tecnología proporciona los servicios de infraestructura que son necesarios para ejecutar aplicaciones (software). Son implementados por computadores y la comunicación del hardware y software del sistema.

5.3.6 Marco de referencia

Los aspectos y capas identificadas en los apartados anteriores se pueden organizar como un marco de nueve celdas, es importante darse cuenta de que la clasificación de conceptos basados en aspectos y capas es sólo uno y es imposible definir un límite estricto entre los aspecto capas, porque los conceptos que enlazan los diferentes aspectos y capas juegan un papel central en una arquitectura coherente. El trabajo de un arquitecto de la empresa toca varios aspectos, no expresamente contemplados en el marco ArchiMate. Dominios conceptuales, son:

- Objetivos, principios y requisitos
- Riesgos y seguridad
- Gobierno
- Políticas y reglas de negocio
- Costos
- Rendimiento
- Planificación y evolución

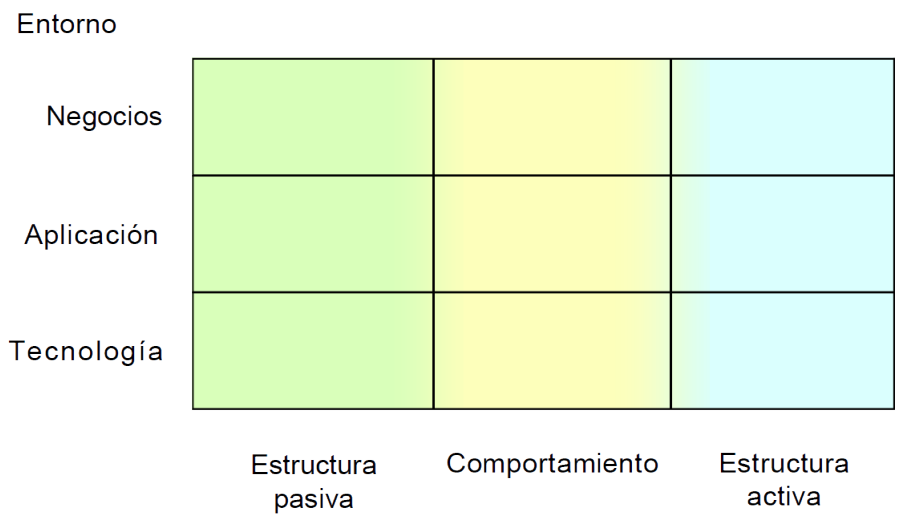


Fig. 5.5.: Estructura de la Arquitectura

5.3.7 Motivación

Archimate añade los conceptos de motivación, como objetivo, principio y requisito. Un elemento de motivación se define como un elemento que proporciona el contexto o la razón que está detrás de la arquitectura de la empresa.

La gestión de requisitos es una actividad importante en el proceso de diseño y gestión de las arquitecturas empresariales. Las metas de las diversas partes interesadas son la base de cualquier cambio en una organización. Estos objetivos deben ser traducidos a los requisitos sobre la arquitectura de la organización. Esta arquitectura debería reflejar cómo los requisitos son realizados por los servicios, procesos y aplicaciones de software en las operaciones del día a día. Por lo tanto, la calidad de la arquitectura está determinada en gran medida por la capacidad de capturar y analizar los objetivos y requisitos pertinentes, el grado en que puede ser realizado por la arquitectura, y la facilidad con que objetivo y los requisitos se puede cambiar.

Principios y requerimientos están fuertemente relacionados. Los principios son normas y directrices generales que ayudan a informar y apoyar la forma en que una organización se marca sobre el cumplimiento de su misión. En contraste, las limitaciones de los requisitos dan forma a un diseño específico de la arquitectura empresarial. Esto corresponde a la distinción entre dos interpretaciones generales:

- Como la estructura de una organización en términos de sus componentes y sus relaciones
- Como un conjunto de principios que se deben aplicar a dicha estructura.

El alcance de la primera interpretación se refiere a un solo diseño de la organización, mientras que el segundo se refiere a cualquier diseño posible. Los requisitos están asociados con la primera interpretación. En su lugar, los principios son independientes de un diseño específico y tienen que especializarse en las necesidades y en el proceso de diseño de la arquitectura de la organización.

La gestión de requisitos inadecuada es una de las principales causas de daños y fallas en los proyectos informáticos, por exceder los presupuestos o plazos, por no entregar los resultados esperados. Por lo tanto, el proceso de gestión de requisitos y el proceso de desarrollo de la arquitectura necesitan estar bien alineados, y la trazabilidad

deben mantenerse entre las necesidades de los elementos arquitectónicos que dan cuenta a estos requisitos.

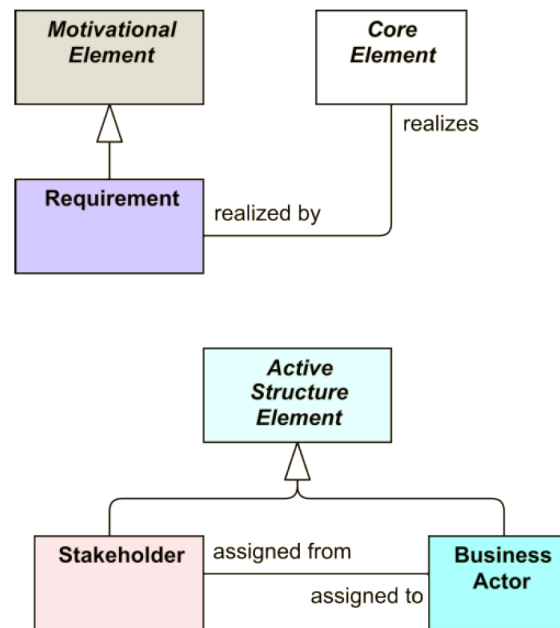


Fig. 5.6.: Relación entre los elementos básicos de motivación en ArchiMate

5.3.8 Implementación y Migración

La extensión de implantación y migración de ArchiMate añade conceptos para apoyar las fases finales de ADM, relacionados con la implementación y migración de arquitecturas: Fase E (Oportunidades y Soluciones), Fase F (planeamiento de migración), y la Fase G (Gobierno de Aplicación).

Esta extensión incluye conceptos para el modelado de los programas y proyectos de aplicación que soportan los programas, el portafolio y la gestión de proyectos. Conceptos que son específicos para uno de estos métodos no son parte de la extensión, pero se pueden definir como la especialización de los conceptos genéricos. De esta manera, el conjunto de conceptos y relaciones que se define en la extensión se mantiene a un mínimo.

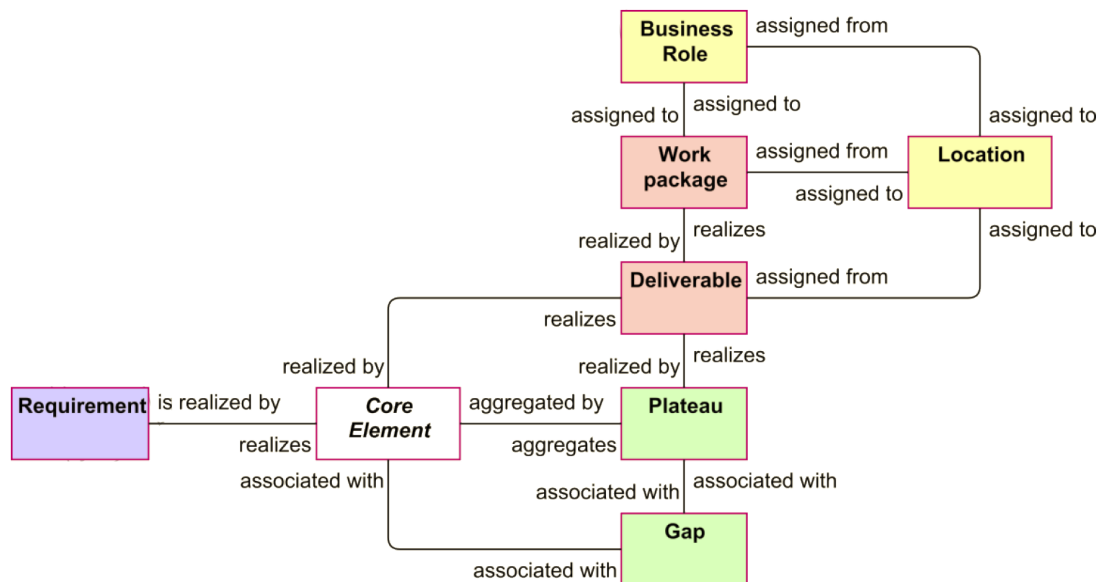


Fig. 5.7.: Relaciones entre la motivación, Core, Implementación y Migración

5.3.9 Archimate: Su relación con TOGAF

El lenguaje ArchiMate, tal como se describe en la Norma Técnica, complementa TOGAF, ya que proporciona un conjunto independiente de los proveedores de los conceptos, incluyendo una representación gráfica, que ayuda a crear un modelo integrado coherente "por debajo de la línea de flotación", que puede ser representado en forma de puntos de vista TOGAF.

Aunque algunos de los puntos de vista que se definen en TOGAF son difíciles de hacer corresponder con los puntos de vista ArchiMate, el idioma ArchiMate y sus técnicas de análisis sí apoyan los conceptos abordados en estos puntos de vista. Si bien no hay correlación de uno a uno entre ellos, todavía hay una buena cantidad de correspondencia entre los puntos de vista ArchiMate y los puntos de vista que se definen en TOGAF. Aunque los puntos de vista correspondientes de ArchiMate y TOGAF no necesariamente tienen la misma cobertura, podemos ver que muchos puntos de vista de ambos métodos abordan en gran medida los mismos problemas.

TOGAF y ArchiMate pueden ser fácilmente utilizados en conjunto y parecen cubrir gran parte de lo mismo, aunque con algunas diferencias en el alcance y enfoque.

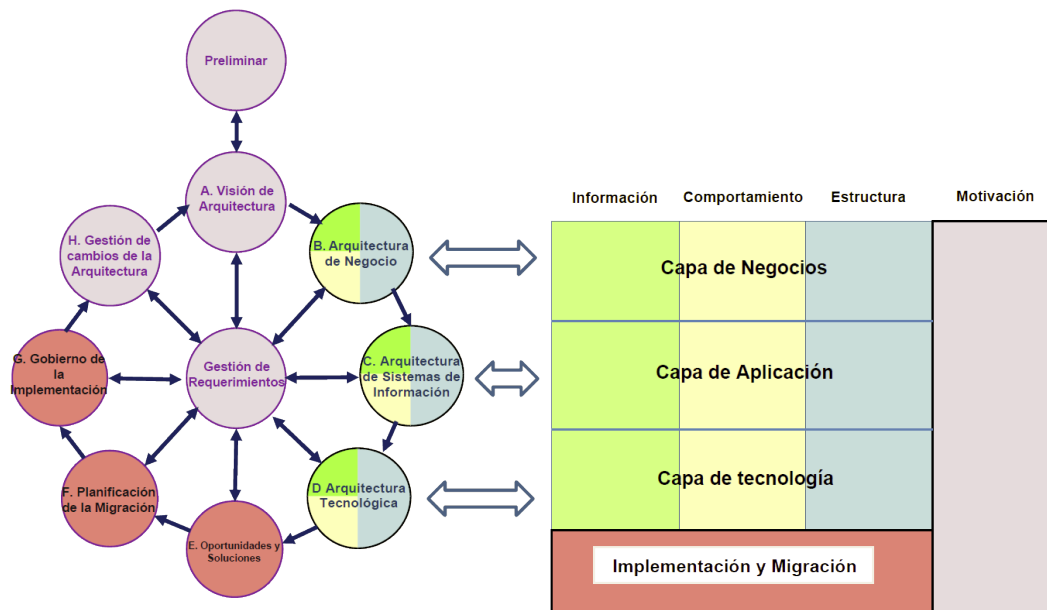


Fig. 5.8.: Correspondencia entre ArchiMate (incluyendo extensiones) y TOGAF

Patrones

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

6.1	Programación Orientada a objetos	38
6.2	Patrones de software	38
6.2.1	Definiciones	39
6.2.2	Clases de patrones software	39
6.3	Patrones de Diseño	40
6.3.1	Clasificación de los patrones de diseño	40

6.1 Programación Orientada a objetos

La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clase, como sucede en el mundo real.

Una ventaja de la POO frente al paradigma algorítmico es la facilidad que brinda a través de sus herramientas, de concebir, analizar, modelar, diseñar e implementar el mundo real de manera fiel a como se presenta en la realidad; el paso que hay desde la concepción y asimilación del problema hasta la implementación del mismo es un proceso que se hace de manera casi natural. Esto porque el mundo está lleno de objetos reales, los cuales se pueden representar como tales en una solución computarizada.

6.2 Patrones de software

Los patrones para el desarrollo de software son uno de los últimos avances de la Tecnología Orientada a Objetos. Los patrones son una forma literaria para resolver problemas de ingeniería del software, que tienen sus raíces en los patrones de la arquitectura.

Los diseñadores y analistas de software más experimentados aplican de forma intuitiva algunos criterios que solucionan los problemas de manera elegante y efectiva. La ingeniería del software se enfrenta a problemas variados que hay que identificar para poder utilizar la misma solución (aunque matizada) con problemas similares.

El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores para comunicar experiencia sobre los problemas y sus soluciones.

6.2.1 Definiciones

Los diferentes autores han dado diversas definiciones de lo que es un patrón software.

- *Dirk Riehle y Heinz Zullighoven:*
“Un patrón es la abstracción de una forma concreta que puede repetirse en contextos específicos.”.
- *Richard Gabriel:*
“Cada patrón es una regla de tres partes, la cual expresa una relación entre un cierto contexto, un conjunto de fuerzas que ocurren repetidamente en ese contexto y una cierta configuración software que permite a estas fuerzas resolverse por si mismas.”.

6.2.2 Clases de patrones software

Existen diferentes ámbitos dentro de la ingeniería del software donde se pueden aplicar los patrones:

1. **Patrones de Arquitectura:** Expresa una organización o esquema estructural fundamental para sistemas software. Proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades, e incluye una guía para organizar las relaciones entre ellos.
2. **Patrones de Diseño:** Proporciona un esquema para refinar los subsistemas o componentes de un sistema software, o las relaciones entre ellos. Describe estructuras repetitivas de comunicar componentes que resuelven un problema de diseño en un contexto particular.
3. **Patrones de Programación:** Un idioma es un patrón de bajo nivel de un lenguaje de programación específico. Describe como implementar aspectos de componentes o de las relaciones entre ellos utilizando las facilidades del lenguaje de programación dado.
4. **Patrones de Análisis:** Describen un conjunto de prácticas que aseguran la obtención de un buen modelo de un problema y su solución.
5. **Patrones Organizacionales:** Describen la estructura y prácticas de las organizaciones.

6.3 Patrones de Diseño

Los patrones de diseño tienen un cierto nivel de abstracción. Los patrones de diseño no son diseños tales como la realización de listas y tablas hash que pueden ser codificadas en clases y reutilizadas. Un algoritmo puede ser un ejemplo de implementación de un patrón, pero es demasiado incompleto, específico y rígido para ser un patrón. Una regla o heurística puede participar en los efectos de un patrón, pero un patrón es mucho más. Los patrones de diseño son descripciones de las comunicaciones de objetos y clases que son personalizadas para resolver un problema general de diseño en un contexto particular.

Un patrón de diseño nombra, abstrae e identifica los aspectos clave de un diseño estructurado, común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. Los patrones de diseño identifican las clases participantes y las instancias, sus papeles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se enfoca sobre un particular diseño orientado a objetos. Se describe cuando se aplica, las características de otros diseños y las consecuencias y ventajas de su uso.

Los patrones de diseño se pueden utilizar en cualquier lenguaje de programación orientado a objetos, adaptando los diseños generales a las características de la implementación particular.

6.3.1 Clasificación de los patrones de diseño

Dado que hay muchos patrones de diseño necesitamos un modo de organizarlos. En esta sección clasificamos los patrones de diseño de tal forma que podamos referirnos a familias de patrones relacionados. La clasificación nos ayuda a saber lo que hace un patrón. Según el libro “Patterns in Java (Volume 1)” existen seis categorías:

Fundamentales

Los patrones de esta categoría son los más fundamentales e importantes patrones de diseño conocidos. Estos patrones son utilizados extensivamente en otros patrones de diseño.

Creación

Los patrones de creación muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos un objeto delegará responsabilidades.

A menudo hay varios patrones de creación que puedes aplicar en una situación. Algunas veces se pueden combinar múltiples patrones ventajosamente. En otros casos se debe elegir entre los patrones que compiten.

Partición

Los patrones de esta categoría proveen la guía sobre como dividir actores complejos y casos de uso en múltiples clases.

Estructurales

Los patrones de esta categoría describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.

Comportamiento

Los patrones de este tipo son utilizados para organizar, manejar y combinar comportamientos.

Concurrencia

Los patrones de esta categoría permiten coordinar las operaciones concurrentes. Estos patrones se dirigen principalmente a dos tipos diferentes de problemas:

1. **Recursos compartidos:** Cuando las operaciones concurrentes acceden a los mismos datos o otros tipos de recursos compartidos, podría darse la posibilidad de que las operaciones interfirieran unas con otras si ellas acceden a los recursos al mismo tiempo. Para garantizar que cada operación se ejecuta

correctamente, la operación debe ser protegida para acceder a los recursos compartidos en solitario. Sin embargo, si las operaciones están completamente protegidas, entonces podrían bloquearse y no ser capaces de finalizar su ejecución.

2. **Secuencia de operaciones:** Si las operaciones son protegidas para acceder a un recurso compartido una cada vez, entonces podría ser necesario garantizar que ellas acceden a los recursos compartidos en un orden particular. Por ejemplo, un objeto nunca será borrado de una estructura de datos antes de que esté sea añadido a la estructura de datos.

Fundamentales	De Creación
<ul style="list-style-type: none"> ■ Delegation ■ Interface ■ Unmitable ■ Marker Interface ■ Proxy 	<ul style="list-style-type: none"> ■ Factory Method ■ Abstract Factory ■ Builder ■ Prototype ■ Singleton ■ Object Pool
De Partición	Estructurales
<ul style="list-style-type: none"> ■ Layered Initialization ■ Filter ■ Composite 	<ul style="list-style-type: none"> ■ Adapter ■ Iterator ■ Bridge ■ Facade ■ Flyweight ■ Dynamic Linkage ■ Virtual proxy ■ Decorator ■ Cache Management

De Comportamiento	De Concurrencia
<ul style="list-style-type: none"> ■ Chain of Responsibility ■ Command ■ Little Language ■ Mediator ■ Snapshot ■ Observer ■ State ■ Null Object ■ Strategy ■ Template Method ■ Visitor 	<ul style="list-style-type: none"> ■ Single Threaded Execution ■ Guarded Suspension ■ Balking ■ Scheduler ■ Read/Write Lock ■ Producer-Consumer ■ Two-Phase Termination

Tab. 6.1.: Catalogo de Patrones de Diseño

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

Contenido

7.1 Descripción	46
---------------------------	----

7.1 Descripción

Coloso es una plataforma de desarrollo dirigida a la gestión integral del proceso de Ingeniería de Software que nos permite desarrollar metodologías de arquitectura empresarial. Fue desarrollado por Colosoft, una comunidad dedicada a la investigación, desarrollo e innovación en ingeniería de software.

El software Coloso hace parte de un conjunto de herramientas que permiten abordar el desarrollo de software en todas sus etapas. Soporta diferentes metodologías de desarrollo generando los artefactos que aplica a cada metodología específica, Soporta las disciplinas como: Ingeniería de desarrollo, Arquitectura de software, Ingeniería de procesos de software, Desarrollo de alto nivel, Diseño de software, Ingeniería de sistemas, Análisis de sistemas, Ingeniería de software, Ingeniería de requerimientos, Patrones, Permite el modelamiento de arquitecturas empresariales y soporta los 25 puntos de vista del lenguaje Archimate.

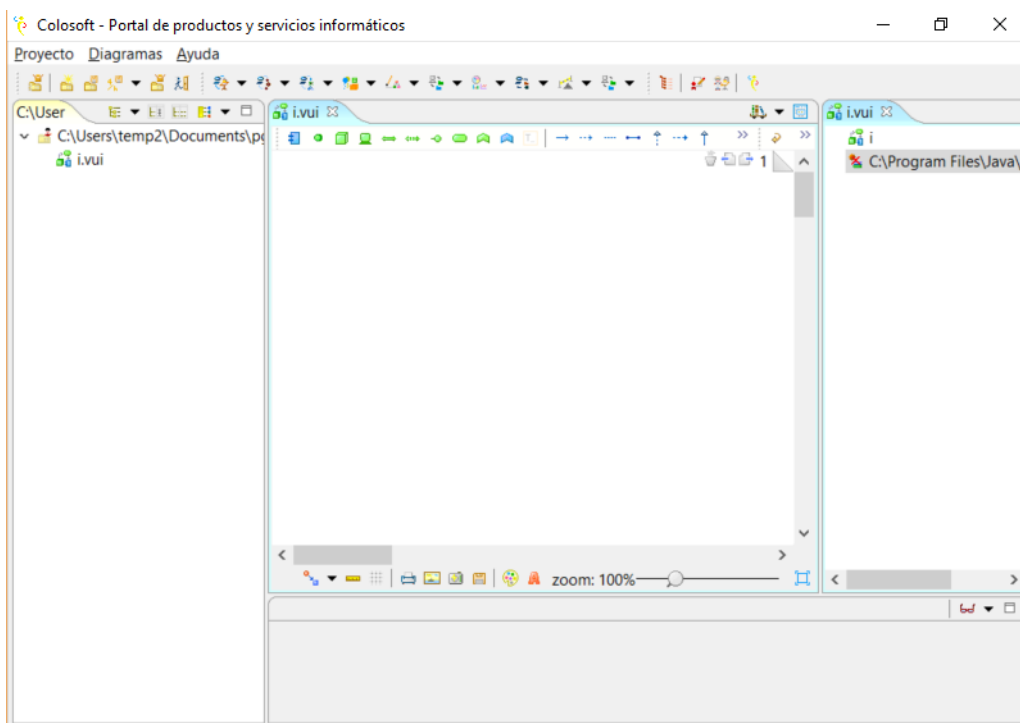


Fig. 7.1.: Interfaz Coloso

Parte II

Arquitectura Empresarial

Modelo de Negocio

This chapter introduces the theoretical, experimental and computational concepts used throughout the thesis

Contenido

8.1	Punto de Vista de la Organización	50
8.1.1	Metamodelo	50
8.1.2	Modelo mInstituto	51
8.2	Punto de Vista Cooperación de Actor	51
8.2.1	Metamodelo	52
8.2.2	Modelo mInstituto	53
8.3	Punto de Vista Función de Negocio	54
8.3.1	Metamodelo	54
8.3.2	Modelo mInstituto	55
8.4	Punto de Vista Proceso de Negocio	55
8.4.1	Metamodelo	56
8.4.2	Modelo mInstituto	57
8.5	Punto de Vista Cooperación de Proceso de Negocio	58
8.5.1	Metamodelo	58
8.5.2	Modelo mInstituto	59
8.6	Punto de Vista de Producto	60
8.6.1	Metamodelo	61
8.6.2	Modelo mInstituto	62

8.1 Punto de Vista de la Organización

El punto de vista de la organización se enfoca en el interior de la organización, un departamento, una red de empresas, es un punto de vista muy útil ya que permite identificar competencias, autoridad y responsabilidades en una organización.

Nombre	Organización
Stakeholders	Organización, arquitectos de dominio y proceso, gerentes, empleados, accionistas
Preocupaciones	Identificación de competencias, autoridad y responsabilidades
Propósito	Diseñar, decidir, informar
Nivel de Abstracción	Coherencia
Capa	Capa de negocio
Aspectos	Activo

Tab. 8.1.: Descripción Punto de Vista de la Organización

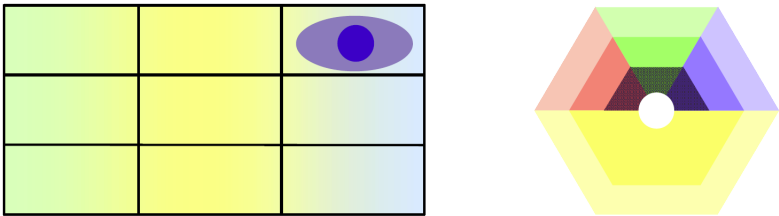


Fig. 8.1.: Posición del punto de vista de organización conceptualmente y marco del punto de vista

8.1.1 Metamodelo

En la Figura 2.2 se ilustra el metamodelo perteneciente al punto de vista de organización, el cual está compuesto de los conceptos de actor, rol, interface, colaboración y localización. En este punto de vista se destaca como concepto fundamental que el actor juega un rol el cual está ubicado en una localización haciendo parte de una colaboración de negocio y se comunica con su entorno a través de una interface.

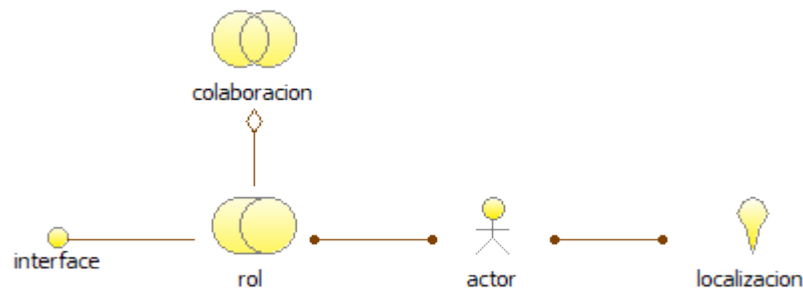


Fig. 8.2.: Metamodelo Punto de vista organización

8.1.2 Modelo mInstituto

Esta vista es la interpretación sobre el organigrama en donde se representan los roles desempeñados en la empresa para el proceso de Minstituto, cabe resaltar la importancia de identificar los roles de la estructura organizacional y su interacción que permitirá establecer las estrategias necesarias para controlar los procesos. Como actores se reconoce al desarrollador, el analista de pruebas y el ejecutivo comercial.

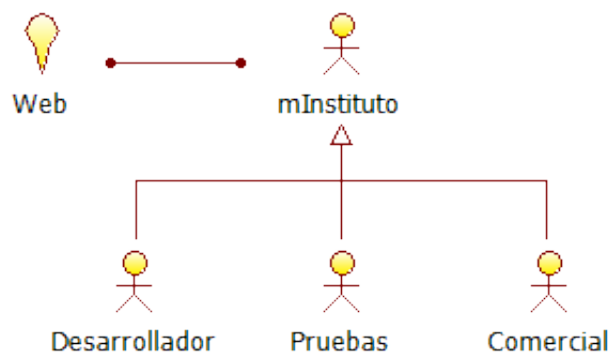


Fig. 8.3.: Modelo Punto de vista organización: minstituto

8.2 Punto de Vista Cooperación de Actor

Este punto de vista se enfoca en los actores y sus relaciones con el entorno que los cobija, es un punto de vista donde los Stakeholders o interesados son la organización, los procesos y los arquitectos de dominio. La finalidad u objetivo de este punto de vista es la de diseñar, decidir e informar.

Nombre	Cooperación de Actor
Stakeholders	Organización, arquitectos de dominio y proceso
Preocupaciones	Relación de actores con el entorno
Propósito	Diseñar, decidir, informar
Nivel de Abstracción	Detalle
Capa	Capa de negocio
Aspectos	Estructura Activa, Comportamiento

Tab. 8.2.: Descripción Punto de Vista Cooperación de Actor

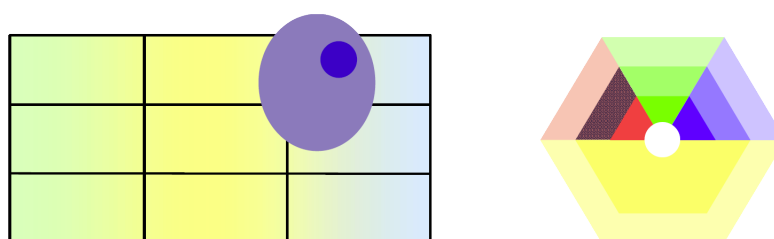


Fig. 8.4.: Posición del punto de vista cooperación de actor conceptualmente y marco del punto de vista

8.2.1 Metamodelo

En la Figura 2.5 se ilustra el metamodelo perteneciente al punto de vista de cooperación de actor el cual esta compuesto de los conceptos de actor, rol, interface, colaboración, servicio de negocio, servicio de aplicación, interface de comunicación de aplicación y componentes de aplicación.

En este punto de vista a el actor se le asigna un rol, el rol se compone de interfaces, a la interface se le asigna servicios y estos servicios van a una capa de aplicación a través de una interface que está conformada por componentes de aplicación y es usada por componentes de aplicación.

Otro uso importante del punto de vista de cooperación de actor es mostrar como un número de actores de negocio cooperantes y / o componentes de aplicación juntos realizan un proceso de negocio. Por lo tanto, en esta vista, tanto actores de negocio como roles y componentes de aplicación pueden aparecer.

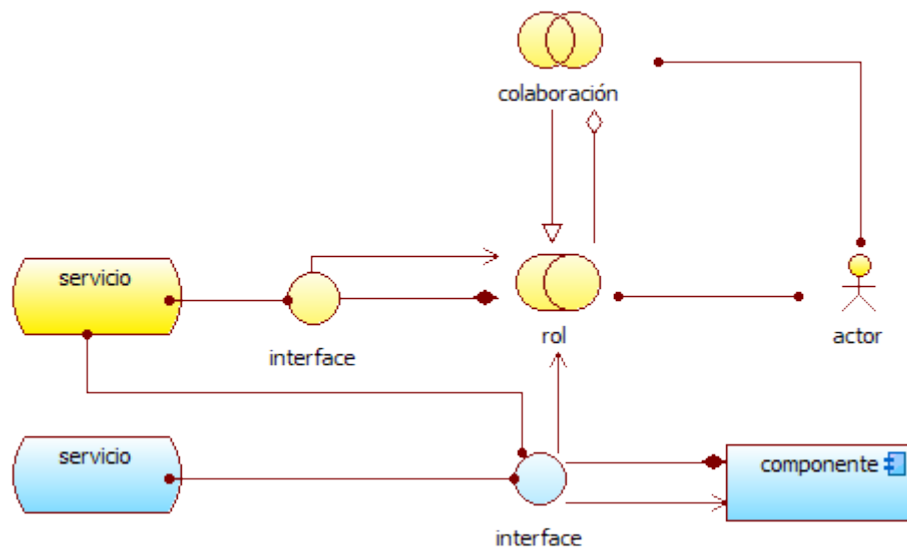


Fig. 8.5.: Metamodelo Punto de Vista Cooperación de Actor

8.2.2 Modelo mInstituto

Para la empresa se establece una clara interacción por parte del área comercial con el cliente y retroalimentación para el equipo de proyecto, el cual se compone del área de producción, es decir, el desarrollador y el analista de pruebas, que serán los encargados de dar trámite a las solicitudes y requerimientos del cliente.

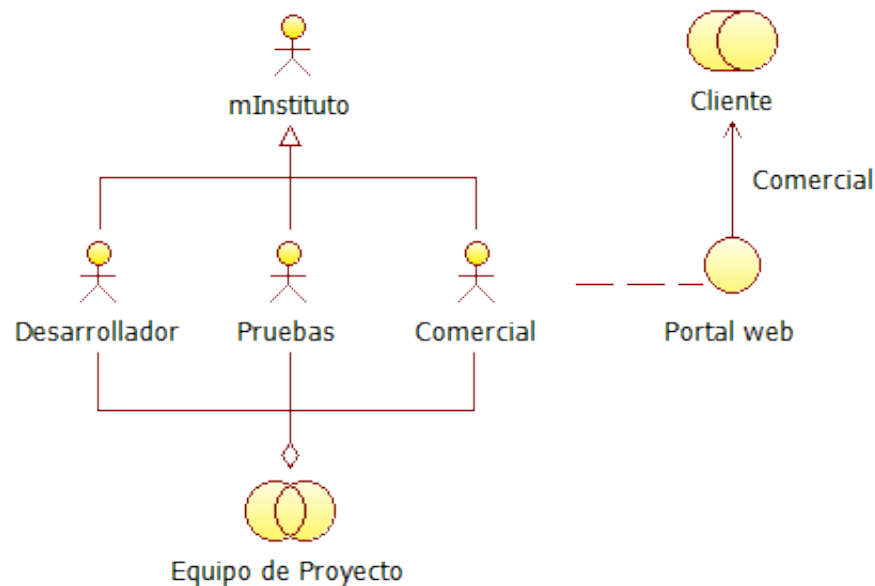


Fig. 8.6.: Modelo Punto de Vista Cooperación de Actor: minstituto

8.3 Punto de Vista Función de Negocio

En este punto de vista se muestran las principales funciones de negocio de la organización y sus relaciones en términos de los flujos de información, valor, o productos entre ellas, los Stakeholders o interesados son los procesos y arquitectos de dominio, administradores operacionales, tiene especial cuidado en la estructura de los procesos de negocio, su coherencia, integridad y las responsabilidades.

Nombre	Función de Negocio
Stakeholders	Organización, arquitectos de dominio y proceso
Preocupaciones	Identificación de competencias, identificación de actividades principales, reducción de la complejidad
Propósito	Diseñar
Nivel de Abstracción	Coherencia
Capa	Capa de negocio
Aspectos	Comportamiento (Activo)

Tab. 8.3.: Descripción Punto de Vista Función de Negocio

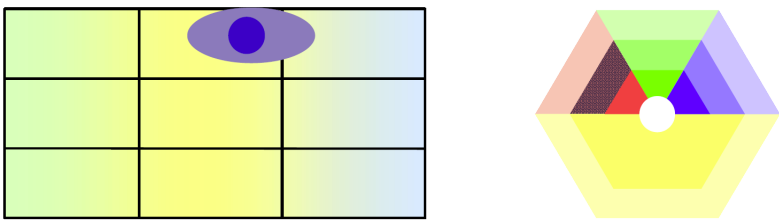


Fig. 8.7.: Posición del punto de vista función de negocio conceptualmente y marco del punto de vista

8.3.1 Metamodelo

La Figura 6.5 muestra los conceptos de actor, rol y función; aquí se asignan tareas o funciones a estos actores, en el modelo se extrae lo que interesa, lo que se quiere capturar o tener en la mente. En este metamodelo aparecen dos tipos de relaciones el flujo y los disparos.

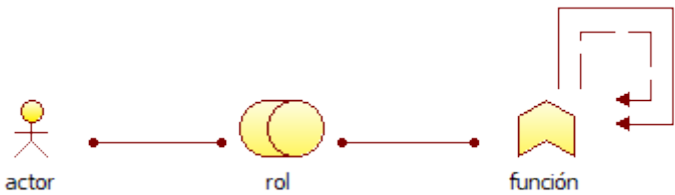


Fig. 8.8.: Metamodelo Punto de Vista Función de Negocio

8.3.2 Modelo mInstituto

En la vista se especifica la función principal de negocio la cual es la comercialización de licencias de uso del software a través de la captación de clientes por medio de material publicitario y la información que se presenta en el portal empresarial, la cual ha sido desarrollada por el diseñador.

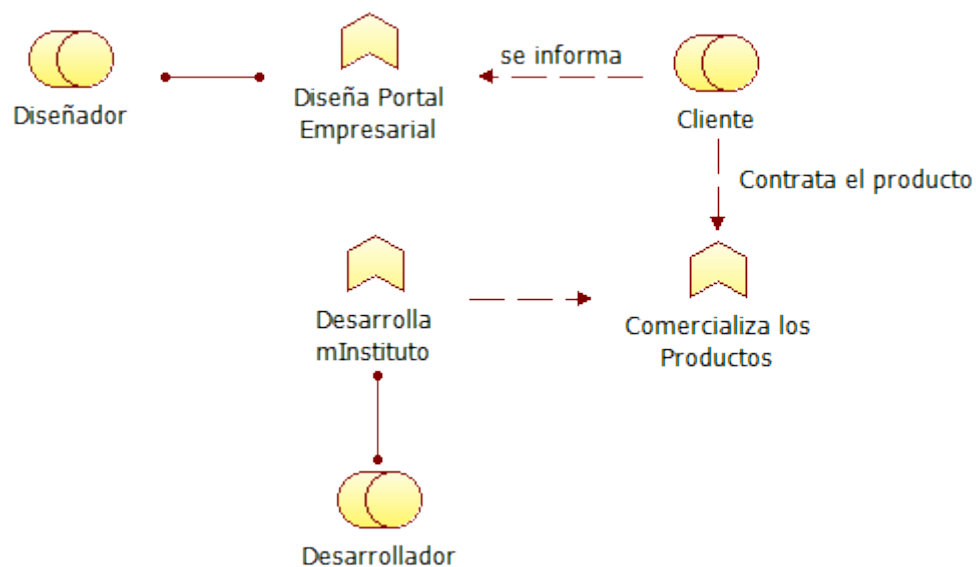


Fig. 8.9.: Modelo Punto de Vista Función de Negocio: minstituto

8.4 Punto de Vista Proceso de Negocio

El punto de vista de proceso de negocio es el encargado de mostrar una estructura de alto nivel y composición de uno o más procesos de negocio. Tiene una complejidad importante, se incorporan elementos de comportamiento se incluye el proceso y/o función de negocio como elemento central, el proceso y/o función de negocio se ve afectado por las mismas relaciones con los demás conceptos, este punto de vista llama la atención en que nos induce a las entrañas de las organizaciones porque se ve lo que ellas hacen.

Nombre	Proceso de Negocio
Stakeholders	Arquitectura de dominio y proceso, Gerentes de operación
Preocupaciones	Estructurar los procesos del negocio, consistencia, integridad y responsabilidades
Propósito	Diseñar
Nivel de Abstracción	Detalle
Capa	Capa de negocio
Aspectos	Comportamiento (Activo), (Pasivo)

Tab. 8.4.: Descripción punto de vista proceso de negocio



Fig. 8.10.: Posición del punto de vista proceso de negocio conceptualmente y marco del punto de vista

8.4.1 Metamodelo

La Figura 6.7 se aprecia que el proceso de negocio tiene que ver con un rol o conjunto de roles, el proceso de negocio es disparado por un evento y el proceso de negocio genera un evento o un proceso de eventos, los procesos no son máquinas infinitas todo proceso es iniciado por uno o un conjunto de eventos.

Los procesos generan objetos de negocio que es la representación del trabajo en la organización, el servicio de negocio es lo que el proceso de negocio lleva a cabo estableciéndose entre los dos una relación de realización, el servicio es el core de negocio lo que el cliente mira, el proceso de negocio es lo que implementa realiza, materializa el servicio y el proceso de negocio funciona porque existen unos roles que se encargan de realizar el proceso.

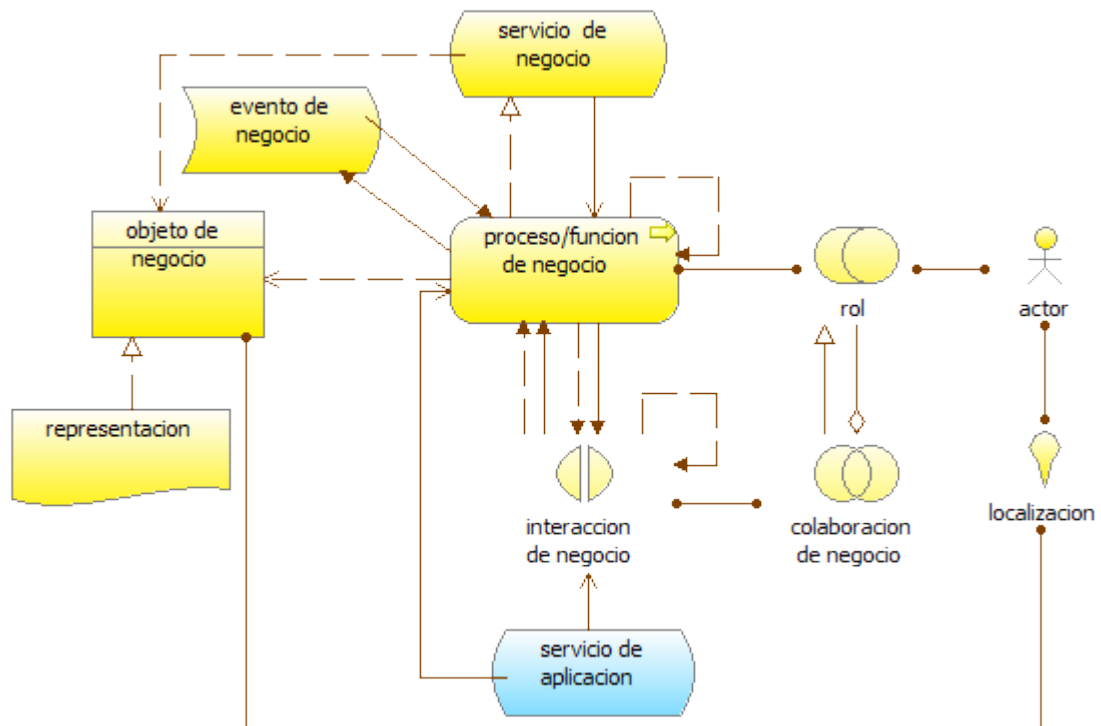


Fig. 8.11.: Metamodelo Punto de Vista Proceso de Negocio

8.4.2 Modelo mInstituto

En esta vista se evidencian los procesos que se deben alinear para obtener un producto con un mayor valor agregado, para Creatics, se identifican los procesos de recepción de solicitud, firma del contrato, recepción y/o confirmación del pago y por último la generación de la licencia de uso y acceso a la plataforma.

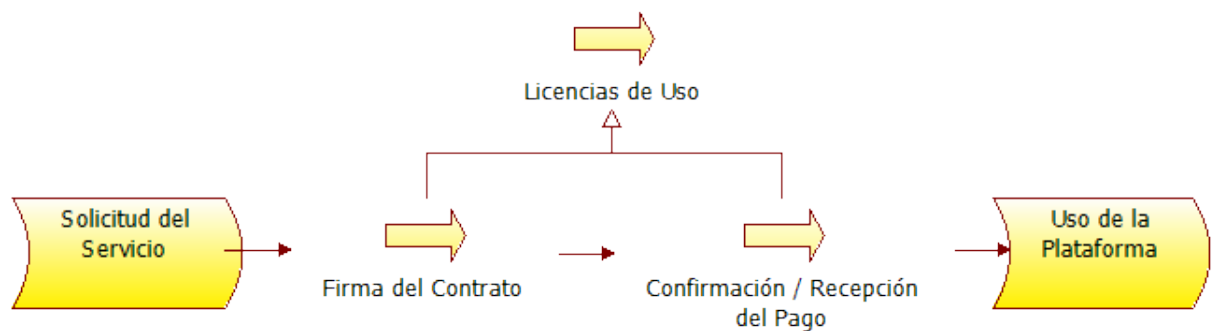


Fig. 8.12.: Modelo Punto de Vista Proceso de Negocio: minstituto

8.5 Punto de Vista Cooperación de Proceso de Negocio

El punto de vista de cooperación de proceso de negocio es usado para mostrar las relaciones de uno o mas procesos de negocio con los demás procesos de negocio y / o con su ambiente. Puede ser usado tanto para crear un diseño de alto nivel de procesos de negocio dentro de su contexto como para proveer un responsable administrador operacional para uno o mas de tales procesos con mando en sus dependencias.

Nombre	Cooperación de Proceso de Negocio
Stakeholders	Procesos, Arquitectos de domino, Gerentes de Operaciones
Preocupaciones	Dependencias de los procesos de negocio, Responsabilidades
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia
Capa	Capa de Negocio (Aplicación)
Aspectos	Comportamiento, (activo), (pasivo)

Tab. 8.5.: Descripción punto de vista de cooperación de proceso

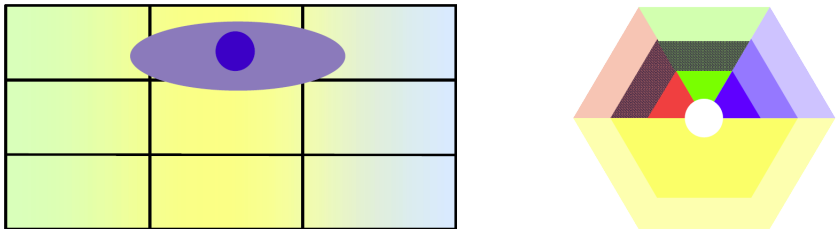


Fig. 8.13.: Posición del punto de vista de cooperación de proceso conceptualmente y marco del punto de vista

8.5.1 Metamodelo

En la Figura 2.5 se ilustra el metamodelo perteneciente al punto de vista de cooperación de proceso el cual esta compuesto de los conceptos de los procesos del negocio y sus responsabilidades. En este punto de vista a al proceso se le asigna un rol, el rol se compone de interfaces, a la interface se le asigna interacciones y estas interacciones van a una capa de aplicación a través de una interface.

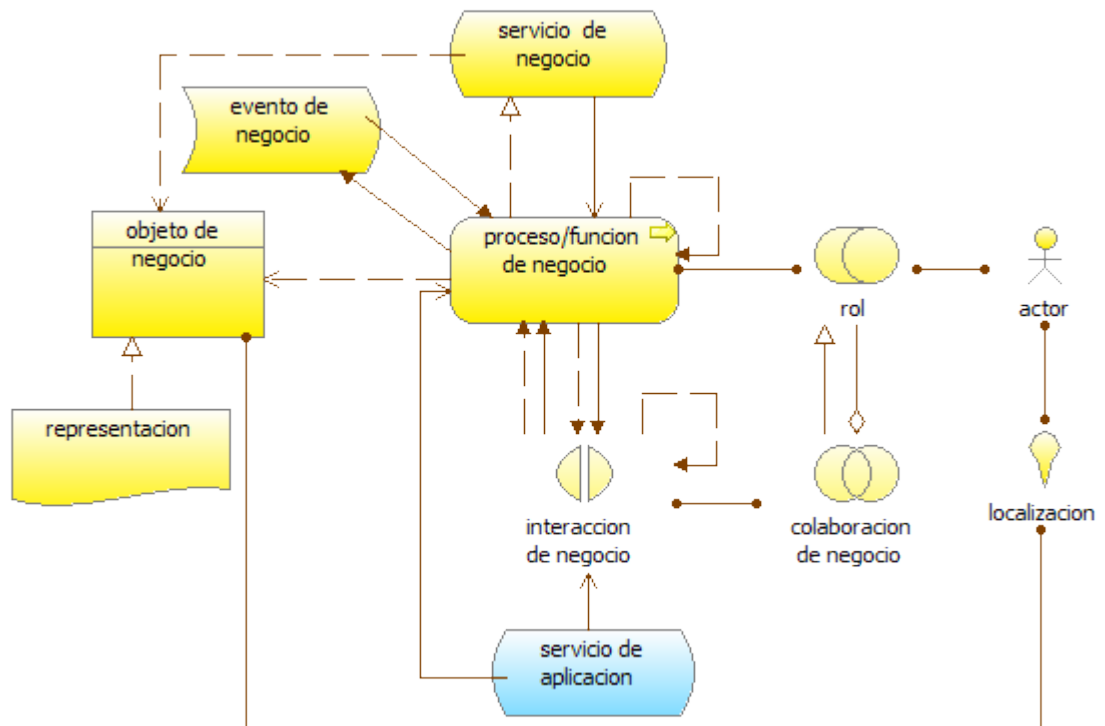


Fig. 8.14.: Metamodelo Punto de Vista de Producto

8.5.2 Modelo mInstituto

En la vista se describe el soporte y razón de ser de Minstituto.com que es generar soluciones tecnológicas aplicadas a la administración educativa, convirtiéndose en el objetivo estratégico de la organización.

La plataforma web que corresponde a la ubicación del producto, es el medio que proporciona al cliente la confidencialidad, integridad y disponibilidad que su información y funciones de negocio necesitan.

El acceso a la plataforma se establece a través de la generación de la licencia y en el contrato se determinan las diferentes características del producto, incluyendo actualizaciones y soporte especializado.

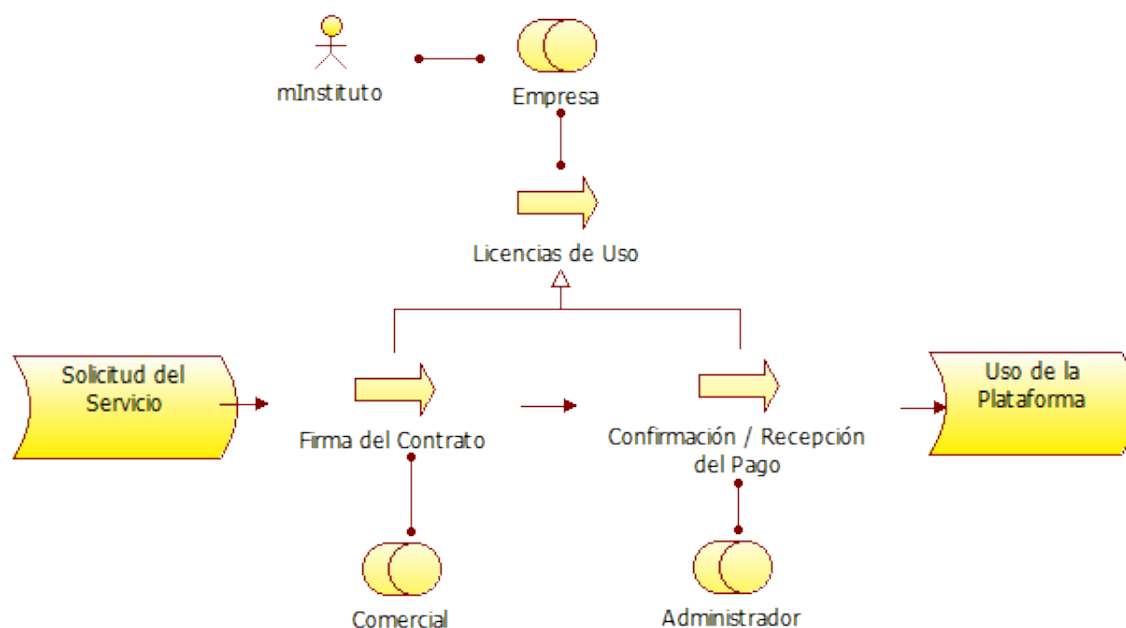


Fig. 8.15.: Modelo Punto de Vista de Producto: minstituto

8.6 Punto de Vista de Producto

Este punto de vista se describe como eje central el valor que uno o más productos ofrecen a la clientes u otras partes externas involucradas con la organización, muestra además la composición de uno o más productos en términos de cómo están compuestos, la asociación, el contrato y otros acuerdos. El punto de vista del producto se suele utilizar en el desarrollo de productos para diseñar un producto componiendo servicios existentes o mediante la identificación de nuevos servicios que se tienen que crear para este producto, dado el valor que un cliente espera de ella.

Nombre	Vista de Producto
Stakeholders	Diseñadores de producto, gerentes de producto, Arquitectos de proceso y de dominio
Preocupaciones	Desarrollo del producto y el valor que este ofrece a la organización
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia
Capa	Capa de Negocio (Aplicación)
Aspectos	Comportamiento, información, (activo)

Tab. 8.6.: Descripción Punto de Vista de Producto

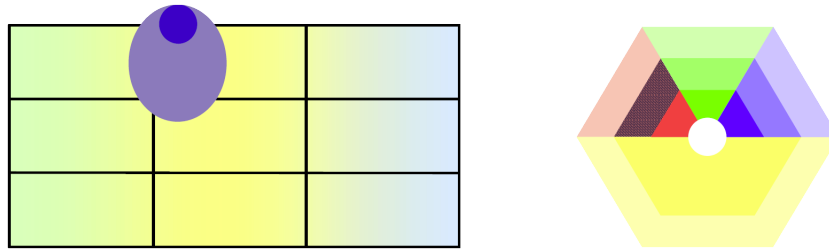


Fig. 8.16.: Posición del Punto de Vista de Producto conceptualmente y marco del punto de vista

8.6.1 Metamodelo

La Figura 6.9 ilustra el punto de vista de producto el cual es la convergencia de los puntos de vista anteriores, es el esfuerzo por conocer la estructura, el esfuerzo por saber qué hace cada persona todo converge en el punto de vista que apunta al producto, el cual es un conjunto de servicios al cual se le adhiere un contrato y como elemento clave se le destaca un valor; el producto reposa sobre los procesos que son hechos por unos roles de negocio los cuales corresponden a unos actores.

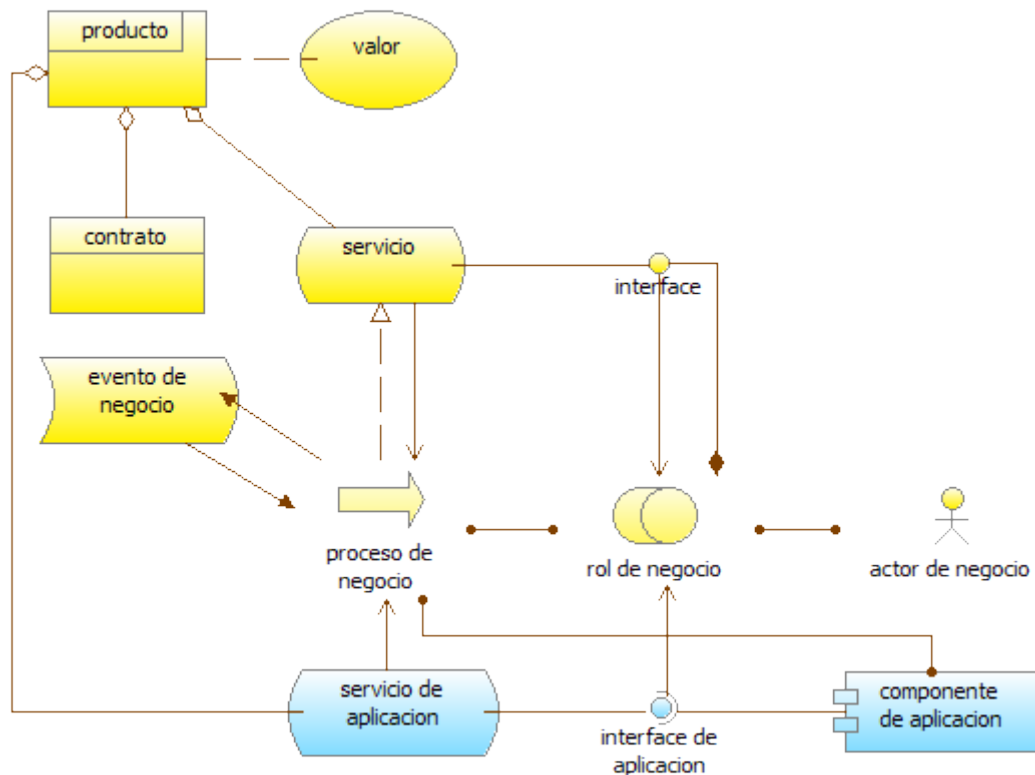


Fig. 8.17.: Metamodelo Punto de Vista de Producto

8.6.2 Modelo mInstituto

El soporte y razón de ser de Minstituto.com es generar soluciones tecnológicas aplicadas a la administración educativa, siendo la plataforma web el medio que proporciona al cliente la confidencialidad, integridad y disponibilidad que su información y funciones de negocio necesitan. El acceso a la plataforma se establece a través de la generación de la licencia y en el contrato se determinan las diferentes características del producto, incluyendo actualizaciones y soporte especializado.

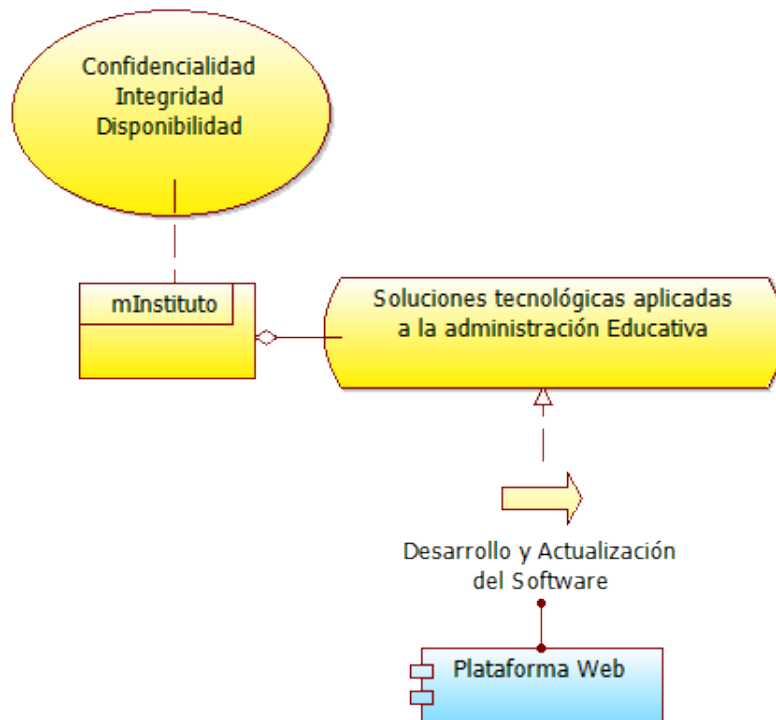


Fig. 8.18.: Modelo Punto de Vista de Producto: minstituto

Capa de Aplicación

This chapter introduces the theoretical, experimental and computational concepts used throughout the thesis

Contenido

9.1	Punto de Vista Comportamiento de Aplicación	64
9.1.1	Metamodelo	64
9.1.2	Modelo mInstituto	65
9.2	Punto de Vista Cooperación de Aplicación	66
9.2.1	Metamodelo	67
9.2.2	Modelo mInstituto	68
9.3	Punto de Vista Estructura de Aplicación	68
9.3.1	Metamodelo	69
9.3.2	Modelo mInstituto	70
9.4	Punto de Vista de Uso de Aplicación	70
9.4.1	Metamodelo	71
9.4.2	Modelo mInstituto	72

9.1 Punto de Vista Comportamiento de Aplicación

El punto de vista del comportamiento de aplicaciones describe el comportamiento interno de la aplicación, este punto de vista es útil en el diseño del comportamiento principal de aplicaciones, o en la identificación de solapamiento funcional entre diferentes aplicaciones.

Nombre	Comportamiento de Aplicación
Stakeholders	Arquitectos de la organización, proceso, aplicación y dominio
Preocupaciones	Estructurar las relaciones entre las aplicaciones, garantizar la consistencia e integridad, reducir la complejidad
Propósito	Diseñar
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de Aplicación (Aplicación)
Aspectos	Activo, comportamiento, (información)

Tab. 9.1.: Descripción punto de vista comportamiento de aplicación

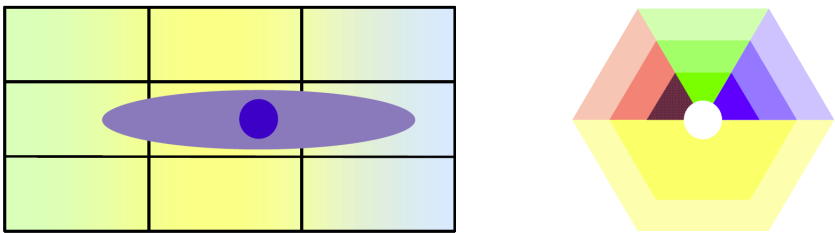


Fig. 9.1.: Posición del punto de vista comportamiento de aplicación conceptualmente y marco del punto de vista

9.1.1 Metamodelo

En la Figura 7.1 se ilustra el metamodelo perteneciente al punto de vista comportamiento de aplicación, el concepto clave para la estructura es el componente de aplicación, a este componentes se le asignan funciones de aplicación, las cuales realizan los servicios de aplicación donde estos servicios soportan los procesos de negocio, se generan unos objetos de datos; la interface es la encargada de interconectar los componentes, en el metamodelo aparece la colaboración de aplicación donde se reúnen componentes que aplican el concepto de colaboración.

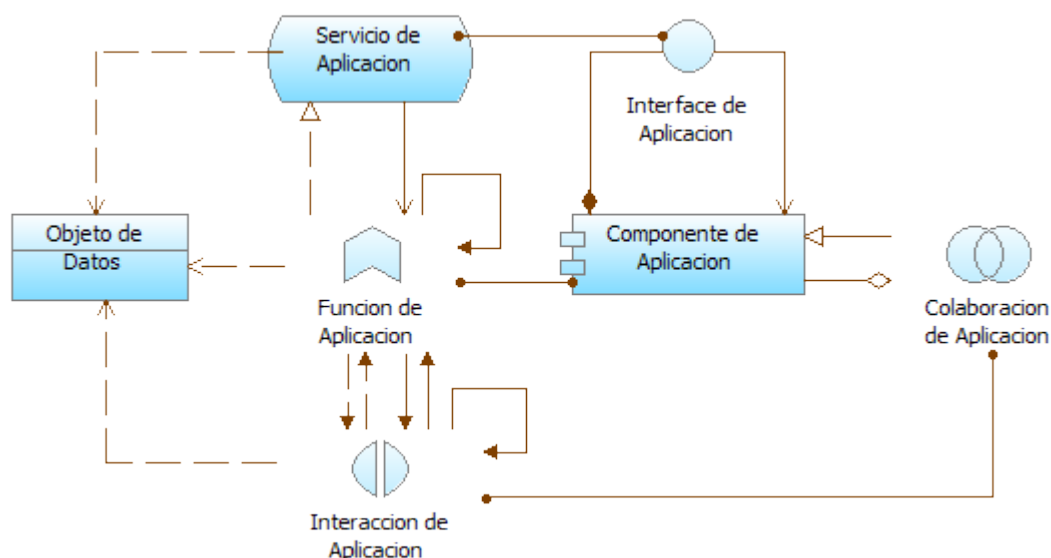


Fig. 9.2.: Metamodelo punto de vista comportamiento de aplicación

9.1.2 Modelo mInstituto

En el punto de vista se describe la funcionalidad de cada componente de aplicación, se encuentran los siguientes componentes y su funcionalidad; el licenciamiento tiene como funcionalidad permitir el acceso de los perfiles y usuarios configurados, garantizar transacciones seguras y generar los respectivos soportes contables de las transacciones.

El producto tiene como funcionalidad brindar la información necesaria en cuanto a las características del mismo y condiciones de adquisición, además incluye las respectivas actualizaciones del sistema. La comunidad tiene como funcionalidad proveer de herramientas de comunicación entre grupos de estudiantes así como enlazar la comunicación con las redes sociales.

Por último la el componente de publicidad tiene como funcionalidad ampliar la participación en el mercado a través de las estrategias de marketing que se utilicen.

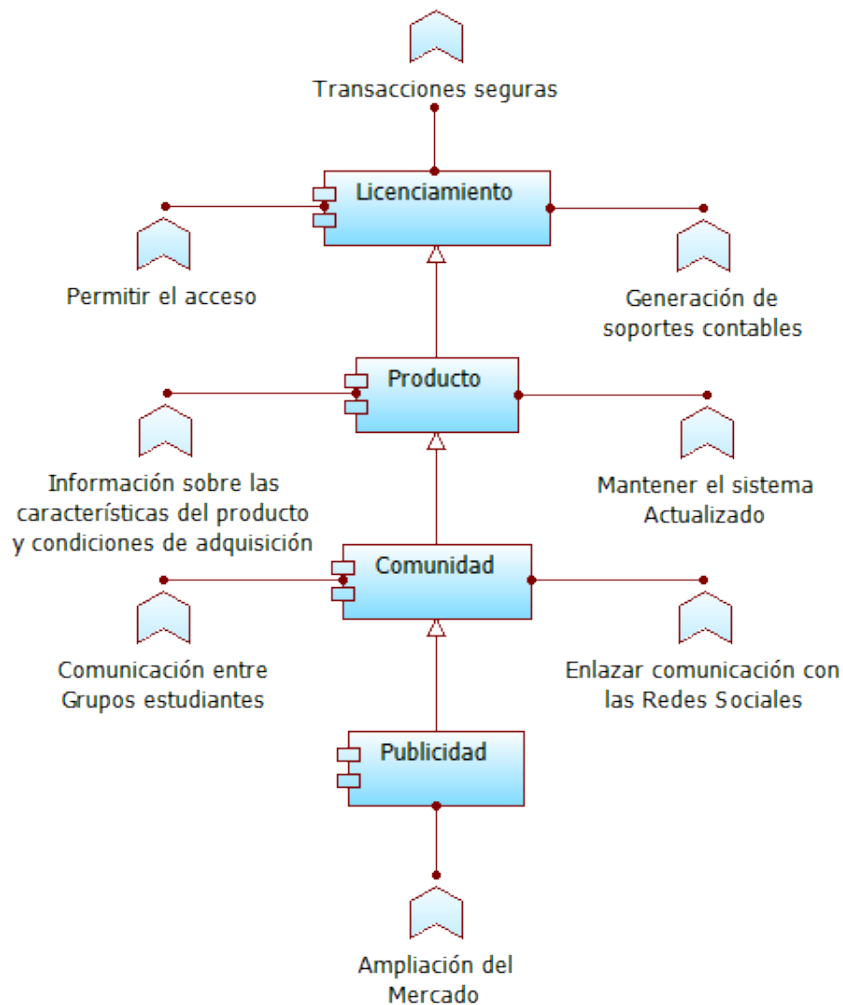


Fig. 9.3.: Modelo punto de vista comportamiento de aplicación: minstituto

9.2 Punto de Vista Cooperación de Aplicación

El punto de vista de Cooperación de la aplicación describe las relaciones entre los componentes de las aplicaciones en función de los flujos de información entre ellos, o en términos de los servicios que ofrecen y su uso.

Este punto de vista también se utiliza para expresar la cooperación (interna) o la orquestación de los servicios que en conjunto apoyan la ejecución de un proceso de negocio.

Nombre	Cooperación de Aplicación
Stakeholders	Arquitectos de la organización, proceso, aplicación y dominio
Preocupaciones	Estructurar las relaciones y dependencias entre las aplicaciones, orquestación de los servicios, garantizar la consistencia e integridad, reducir la complejidad
Propósito	Diseñar
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de Aplicación (Aplicación)
Aspectos	Activo, comportamiento

Tab. 9.2.: Descripción punto de vista cooperación de aplicación

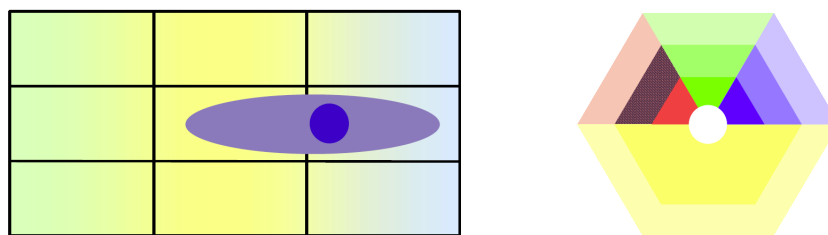


Fig. 9.4.: Posición del punto de vista cooperación de aplicación conceptualmente y marco del punto de vista

9.2.1 Metamodelo

En la figura Figura 7.3 se ilustra el metamodelo perteneciente al punto de vista cooperación de aplicación, el punto de vista se centra en la localización, ya que se agrupan los componentes en un front office y un back office.

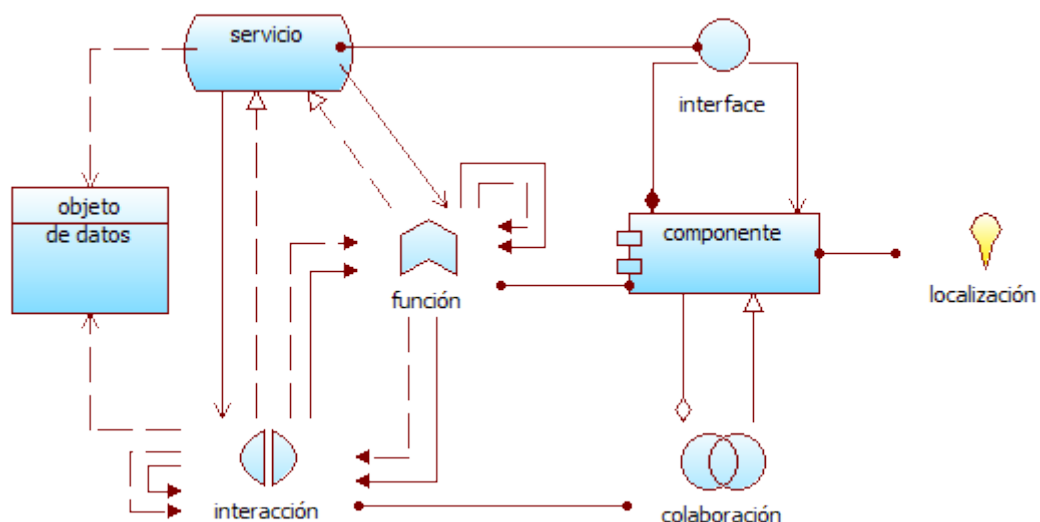


Fig. 9.5.: Metamodelo punto de vista cooperación de aplicación

9.2.2 Modelo mInstituto

En el punto de vista se evidencia que la localización del front office se encuentran los componentes de publicidad, minstituto y comunidad, por otro lado en la localización del back office se encuentran ubicados los componentes de diseño, análisis y desarrollo los cuales se relacionan directamente con la aplicación Minstituto.

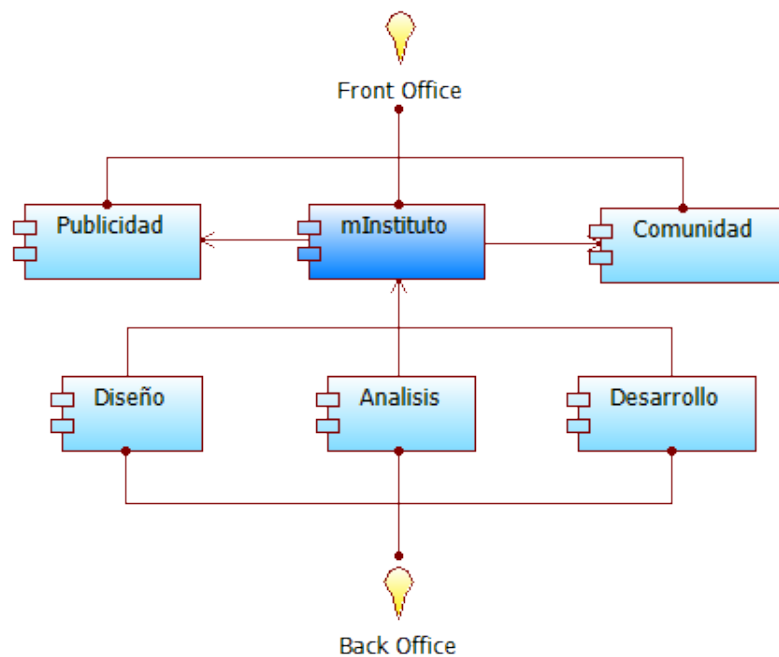


Fig. 9.6.: Modelo Punto de Vista Cooperación de Aplicación: minstituto

9.3 Punto de Vista Estructura de Aplicación

El punto de vista de estructura de la aplicación, muestra la estructura de componentes de una aplicación. Este punto de vista es útil en el diseño o la comprensión de la estructura principal de componentes de la aplicación y el uso de datos asociados.

Nombre	Estructura de Aplicación
Stakeholders	Arquitectos de la organización, aplicación y dominio
Preocupaciones	Estructura de la aplicación, garantizar la consistencia e integridad, reducir la complejidad
Propósito	Diseñar
Nivel de Abstracción	Detalle
Capa	Capa de Aplicación
Aspectos	Activo, (Pasivo)

Tab. 9.3.: Descripción punto de vista estructura de aplicación

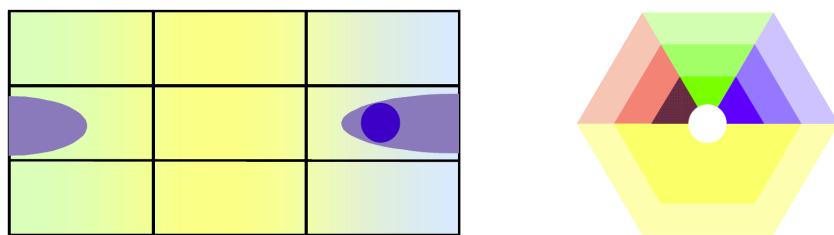


Fig. 9.7.: Posición del punto de vista estructura de aplicación conceptualmente y marco del punto de vista

9.3.1 Metamodelo

En la figura Figura 7.5 se ilustra el metamodelo perteneciente al punto de vista de estructura de la aplicación en este metamodelo se involucran el componente y la interfaz, se muestra la forma de interactuar el sistema con los componentes de software haciendo uso de las interfaces de comunicación.

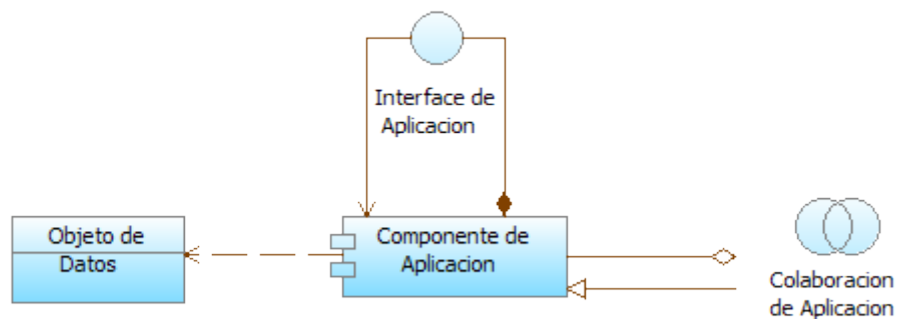


Fig. 9.8.: Metamodelo punto de vista estructura de aplicación

9.3.2 Modelo mInstituto

En la vista se describe el mecanismo de comunicación entre los componentes, se define como componente central minstituto siendo el componente más importante en el cumplimiento de la misión de la organización, la comunicación de éste con los demás componentes se genera a través de interfaces.

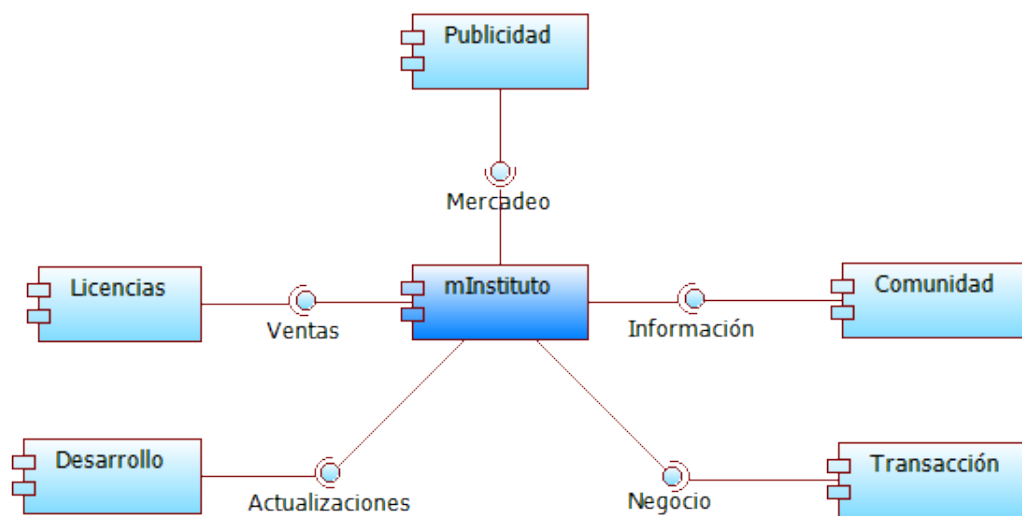


Fig. 9.9.: Modelo punto de vista estructura de aplicación: minstituto

9.4 Punto de Vista de Uso de Aplicación

El punto de vista de uso de aplicación describe como las aplicaciones son usadas para soportar uno o mas procesos de negocio, y como ellos son usados por otras aplicaciones. Puede ser usado en el diseño de una aplicación para identificar los servicios requeridos por los procesos de negocios y otras aplicaciones, o en el diseño de procesos de negocio para describir los servicios que están disponibles. Así, al identificar las dependencias de los procesos de negocio sobre las aplicaciones, puede ser útil para los administradores operativos responsables de estos procesos.

Nombre	Uso de Aplicación
Stakeholders	La empresa, el proceso, los arquitectos de aplicaciones, directores de operaciones
Preocupaciones	Consistencia e Integridad, Reducción de la complejidad
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia
Capa	Capa de Negocio y Aplicación
Aspectos	Comportamiento, estructura

Tab. 9.4.: Descripción punto de vista de Uso de Aplicación

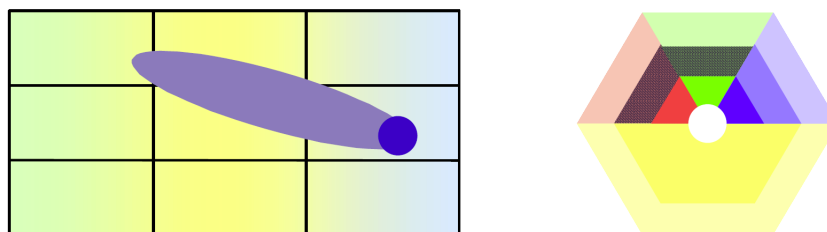


Fig. 9.10.: Posición del punto de vista de Uso de Aplicación conceptualmente y marco del punto de vista

9.4.1 Metamodelo

En la figura Figura 7.5 se ilustra el metamodelo perteneciente al punto de vista de Uso de Aplicación en este metamodelo se involucran el componente y la interfaz, se muestra la forma de usar el software haciendo uso de las interfaces y componentes.

Puede ser usado en el diseño de una aplicación para identificar los servicios requeridos por los procesos de negocios y otras aplicaciones, o en el diseño de procesos de negocio para describir los servicios que están disponibles. Así, al identificar las dependencias de los procesos de negocio sobre las aplicaciones, puede ser útil para los administradores operativos responsables de estos procesos.

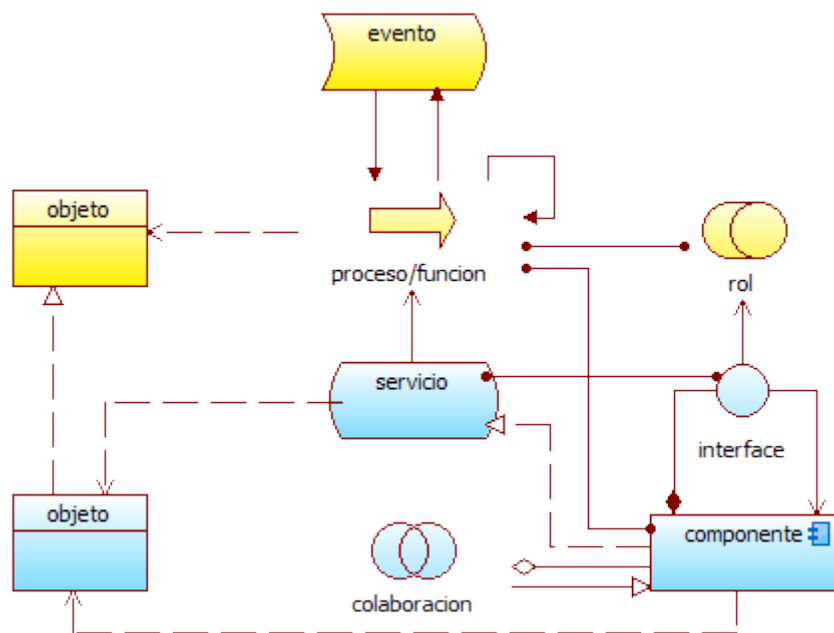


Fig. 9.11.: Metamodelo punto de vista de Uso de Aplicación

9.4.2 Modelo mInstituto

En la vista se destacan los servicios de aplicación fundamentales para el proceso de negocio, los cuales son el desarrollo de la aplicación y la generación de las licencias.

La generación de licencias incluye los servicios de acceso a la aplicación y la realización de los pagos utilizando el componente de transacciones. Por otro lado el desarrollo de aplicación incluye las mejoras y actualizaciones utilizando el componente de desarrollo.

Los dos procesos generados en la capa de aplicación están directamente relacionados con el cliente y pertenecen a los procesos misionales. Por una parte el cliente obtiene el acceso al software y por el otro a las actualizaciones del mismo.

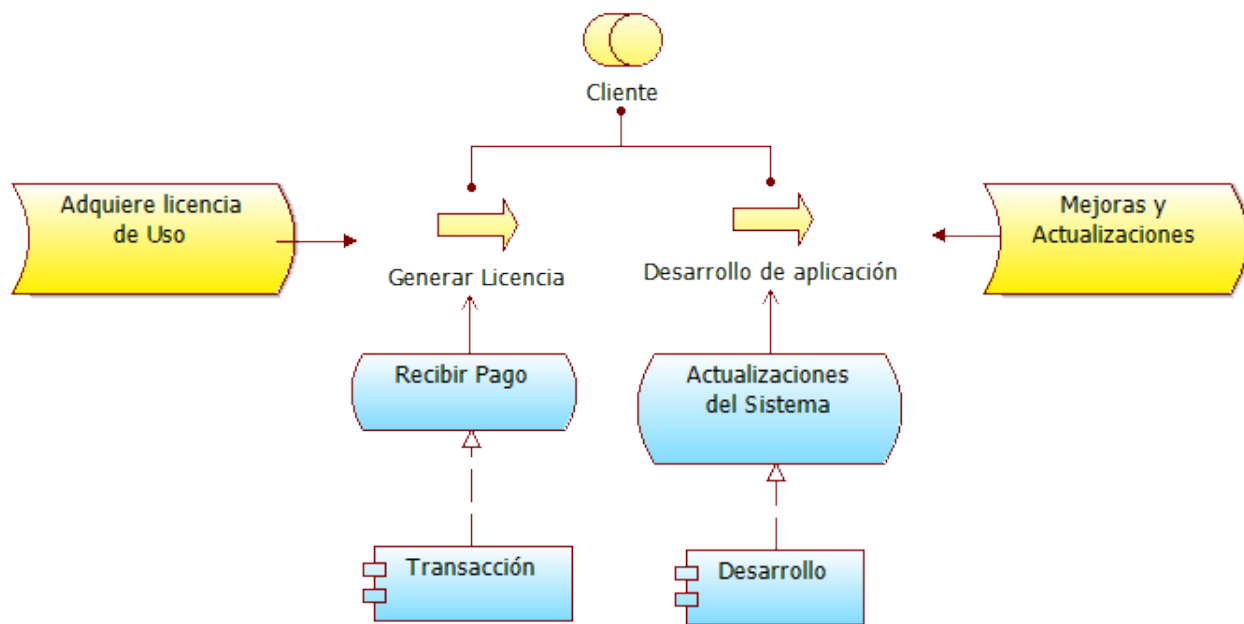


Fig. 9.12.: Modelo punto de vista de Uso de Aplicación: minstituto

Capa de Infraestructura

This chapter introduces the theoretical, experimental and computational concepts used throughout the thesis

Contenido

10.1 Punto de Vista de Infraestructura	76
10.1.1 Metamodelo	76
10.1.2 Modelo mInstituto	77
10.2 Punto de Vista Uso de Infraestructura	78
10.2.1 Metamodelo	79
10.2.2 Modelo mInstituto	79
10.3 Punto de Vista de Organización e Implementación	80
10.3.1 Metamodelo	81
10.3.2 Modelo mInstituto	82
10.4 Punto de Vista de Estructura de Información	83
10.4.1 Metamodelo	84
10.4.2 Modelo mInstituto	84
10.5 Punto de vista de Realización del Servicio	85
10.5.1 Metamodelo	86
10.5.2 Modelo mInstituto	86
10.6 Punto de Vista de Capas	87
10.6.1 Metamodelo	88

10.1 Punto de Vista de Infraestructura

El punto de vista de infraestructura contiene los elementos de la infraestructura de cómputo y hardware de comunicación de apoyo a la capa de aplicación, tales como dispositivos o redes físicas. La Tabla 8.1 describe el punto de vista.

Nombre	Vista de Infraestructura
Stakeholders	Arquitectos de infraestructura y aplicación gerentes de operación
Preocupaciones	Dependencias, escalabilidad y performance
Propósito	Diseñar
Nivel de Abstracción	Coherencia
Capa	Capa de Tecnología
Aspectos	Comportamiento, activa

Tab. 10.1.: Descripción punto de vista de Infraestructura

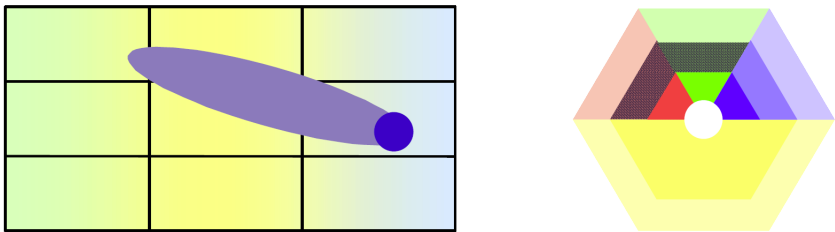


Fig. 10.1.: Posición del punto de vista de infraestructura conceptual y marco del punto de vista

10.1.1 Metamodelo

El metamodelo de la Figura 8.3 se centra en los nodos y los componentes, se muestra como las piezas de software están contenidas en los recursos físicos.

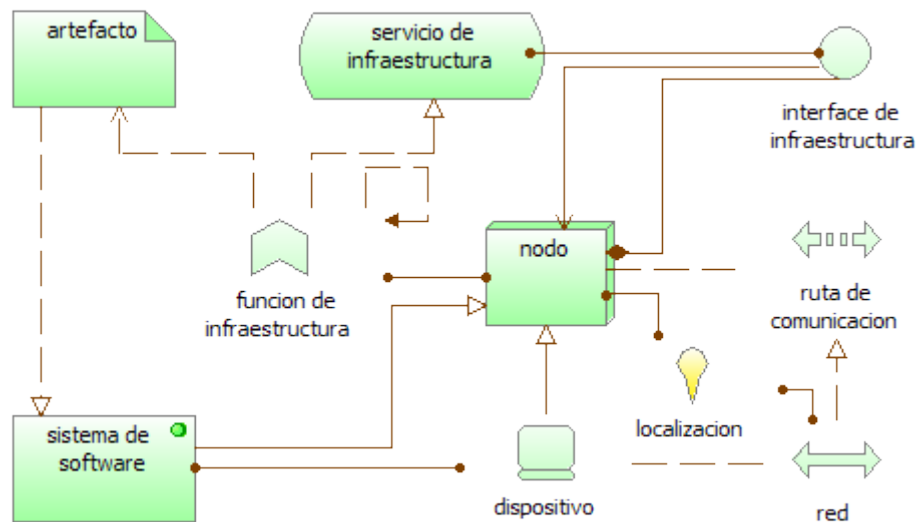


Fig. 10.2.: Metamodelo punto de vista de infraestructura

10.1.2 Modelo mInstituto

La infraestructura para el aplicativo mInstituto esta basada en un portal comercial el cual se ejecuta sobre un servidor de páginas web y necesitara de un hosting que cumpla las restricciones de costos y a su vez lleve la menor cantidad de soporte a nivel de seguridad ya que la funcionalidad principal es la de ofrecer información sobre los productos de la organización y realizar actualizaciones que no van mas allá de noticias relacionadas con la empresa, producto o el medio en el que nos desenvolvemos.

Por otro lado tenemos nuestro portal para el aplicativo, este estará hospedado en un proveedor de servicios que responda con las necesidades de base de datos así como la demanda de consulta de todos los módulos.

La comunicación se realizará a través del proveedor de servicios de Internet contratado por el cliente, utilizando la red y equipos de propiedad del cliente en su respectiva localización.

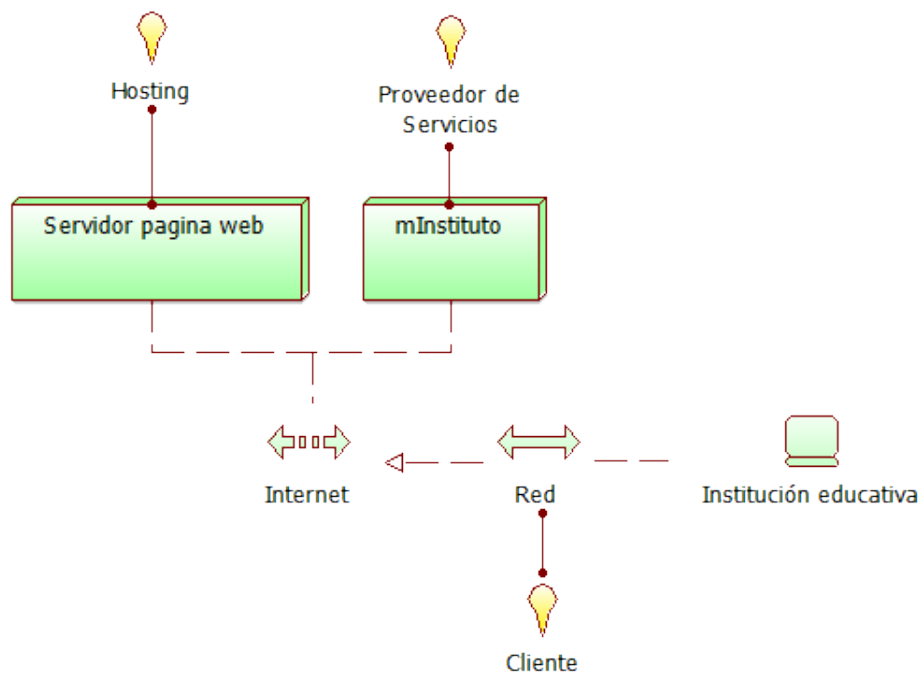


Fig. 10.3.: Modelo punto de vista de infraestructura: minstituto

10.2 Punto de Vista Uso de Infraestructura

El punto de vista de uso Infraestructura muestra cómo las aplicaciones son compatibles con el software y hardware de infraestructura: los servicios de infraestructura son entregados por los dispositivos; el software y redes de sistema se proporcionan a las aplicaciones. Este punto de vista desempeña un papel importante en el análisis de rendimiento y escalabilidad, puesto que se refiere la infraestructura física para el mundo lógico de aplicaciones. Es muy útil en la determinación de los requisitos de rendimiento y calidad en la infraestructura basada en las exigencias de las diferentes aplicaciones que lo utilizan.

Nombre	Uso de Infraestructura
Stakeholders	Arquitectos de infraestructura y aplicación, gerentes de operación
Preocupaciones	Dependencias, escalabilidad y rendimiento
Propósito	Diseñar
Nivel de Abstracción	Coherencia
Capa	Capa de Aplicación y Tecnología
Aspectos	Activo, (comportamiento)

Tab. 10.2.: Descripción punto de vista uso de infraestructura

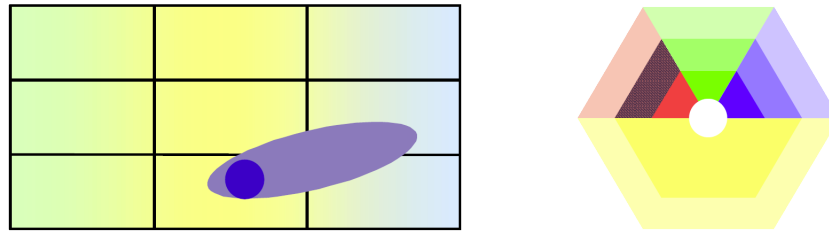


Fig. 10.4.: Posición del punto de vista uso de infraestructura conceptualmente y marco del punto de vista

10.2.1 Metamodelo

El metamodelo de la Figura 8.3 se centra en los nodos y los componentes, se muestra como las piezas de software están contenidas en los recursos físicos.

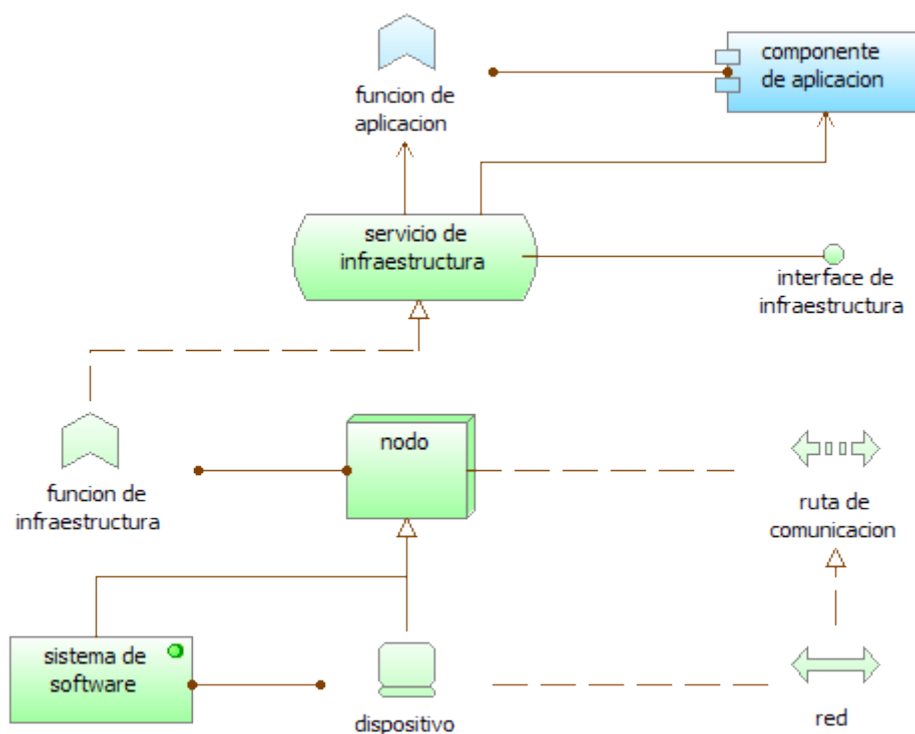


Fig. 10.5.: Metamodelo punto de vista uso de infraestructura

10.2.2 Modelo mInstituto

El servidor de la página web está destinado a la parte comercial de la empresa, tiene como principales componentes la administración de parte de las estrategias de mercadeo y publicidad y por otro lado se encuentra el acceso a una comunidad basada en redes sociales; este a su vez se comunica con el servidor de aplicaciones a

través del punto de acceso o pantalla de login donde nuestros clientes accederán a la aplicación de Minstituto.com, el aplicativo posee herramientas de accesibilidad para los usuarios y perfiles autorizados y tiene incluido un módulo de transacciones y pagos el cual cuenta con el nivel de seguridad informática acorde a las necesidades de la información

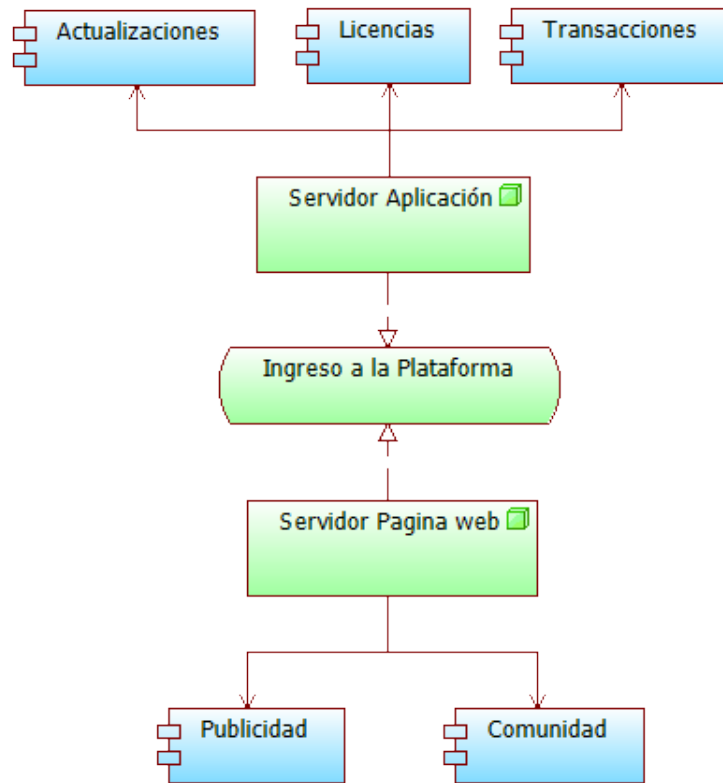


Fig. 10.6.: Modelo punto de vista uso de infraestructura: minstituto

10.3 Punto de Vista de Organización e Implementación

La aplicación y el punto de vista de implementación muestran cómo una o más aplicaciones se realizan o se colaboran dentro de la infraestructura, como las aplicaciones son soportadas por el software y el hardware de infraestructura, los servicios de infraestructura son entregados por los dispositivos; el software y las redes del sistema se proporcionan a las aplicaciones, este punto de vista desempeña un papel importante en el análisis de rendimiento y escalabilidad y que hacer referencia a la infraestructura física para el mundo lógico.

Nombre	Organización e Implementación
Stakeholders	Arquitectos de infraestructura y aplicación gerente de operación
Preocupaciones	Dependencias, seguridad, riesgos
Propósito	Diseñar
Nivel de Abstracción	Coherencia
Capa	Capa de tecnología, capa de aplicación
Aspectos	Activo (Comportamiento)

Tab. 10.3.: Descripción punto de vista de organización e implementación

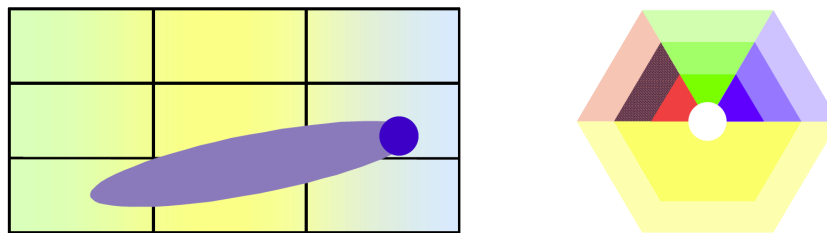


Fig. 10.7.: Posición de la organización e implementación conceptualmente y marco del punto de vista

10.3.1 Metamodelo

En el metamodelo de la Figura 8.5 se aprecia como los diferentes nodos de infraestructura que contienen los componentes de aplicación soportan las comunicación entre ellos, muestran como componen elementos de colaboración los cuales permiten tener sistemas de cruce entre sistemas y componentes de aplicación. En el metamodelo se aumenta un componente de infraestructura para reflejar el concepto de colaboración.

Visto de otro modo comprende la asignación de aplicaciones y componentes en artefactos y el mapeo de la información utilizada por estas aplicaciones y componentes en la infraestructura de almacenamiento, en pocas palabras establece las relaciones de la infraestructura física con el mundo lógico de las aplicaciones.

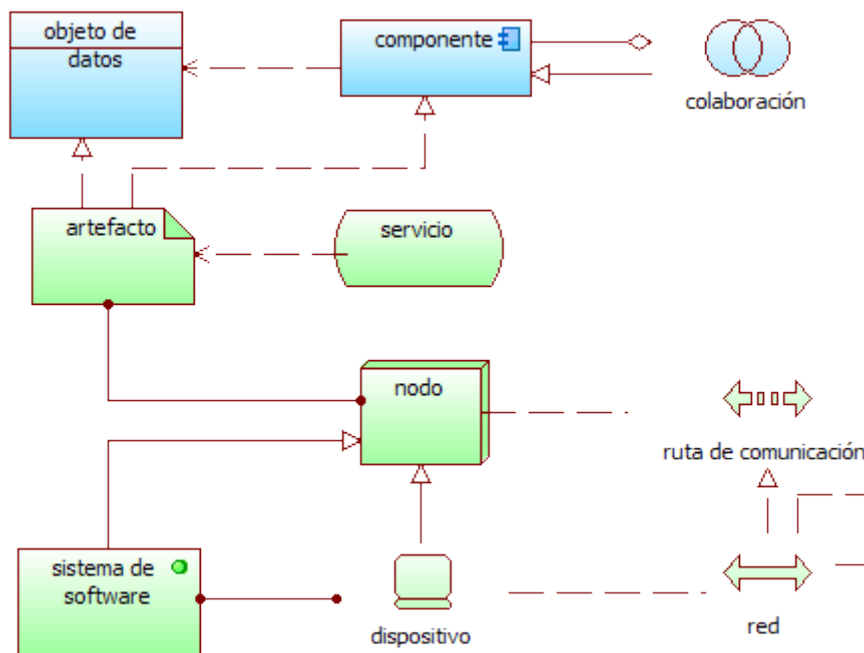


Fig. 10.8.: Metamodelo punto de vista de organización e implementación

10.3.2 Modelo mInstituto

En este punto de vista observamos que Minstituto.com utiliza un servidor Linux y sobre este se ejecuta el código Java utilizando su máquina virtual y conectado a una base de datos PostgreSQL.

Para el portal comercial tendremos en un hosting compartido, una página web elaborada en PHP con una base de datos MySQL. Las dos podrán ser consultadas por los clientes usando Internet de sus proveedor de servicios y un navegador que soporte lenguaje HTML en su última versión, así como tener habilitado la ejecución de código javascript para validaciones necesarias en ambos portales.

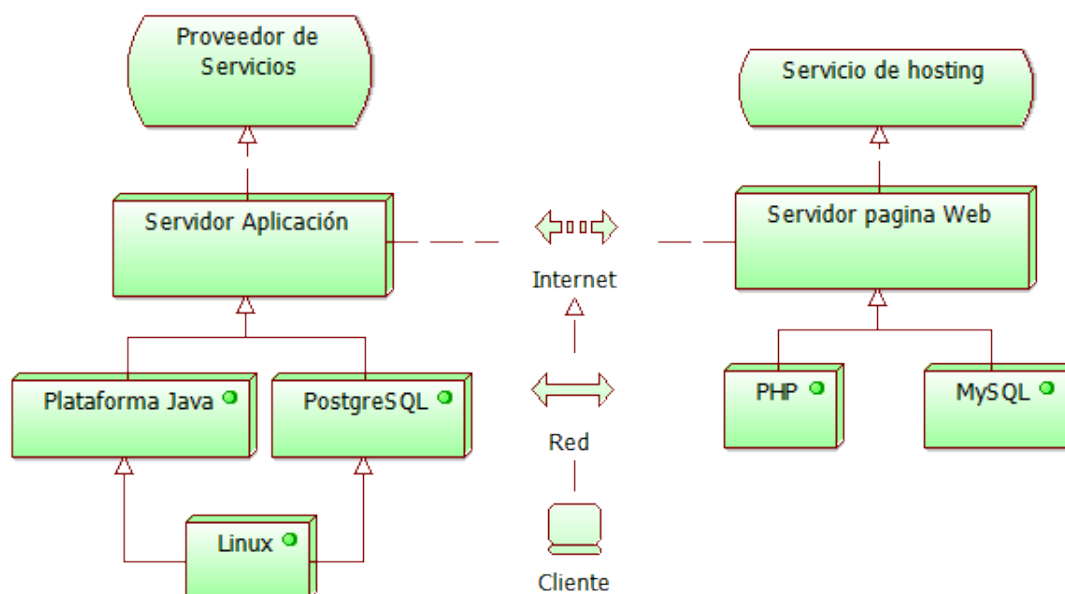


Fig. 10.9.: Modelo punto de vista de organización e implementación: minstituto

10.4 Punto de Vista de Estructura de Información

El punto de vista de estructura de información es comparable a los modelos tradicionales de información creados en el desarrollo de casi cualquier sistema de información. Se muestra la estructura de la información utilizada en la empresa o en un proceso de negocio específico o aplicación, en términos de tipos de datos o las estructuras de clase (orientada a objetos). Además, puede mostrar cómo se representa la información a nivel de empresa (objetos de negocio), a nivel de aplicación en forma de estructuras de datos utilizadas (objetos de datos), y cómo éstos se asignan a la infraestructura subyacente; por ejemplo por medio de un esquema de base de datos (artefacto).

Nombre	Estructura de Información
Stakeholders	Arquitectos de dominio e información
Preocupaciones	Estructura, dependencias e inconsistencia del uso de los datos y la información
Propósito	Diseñar
Nivel de Abstracción	Detalle
Capa	Capa de negocio, capa de tecnología, capa de aplicación
Aspectos	Pasivo

Tab. 10.4.: Descripción punto de vista de estructura de información

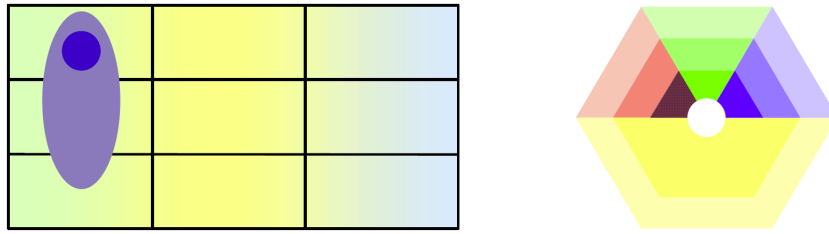


Fig. 10.10.: Posición del punto de vista de estructura de información conceptualmente y marco del punto de vista

10.4.1 Metamodelo

El metamodelo del punto de vista de estructura de información Figura 8.7 da una buena pista de cómo se modelan los elementos finales del negocio que empezamos a percibir, el objeto de negocios el cual baja a un objeto de datos el cual nosotros llamamos artefacto en la infraestructura, le podemos asociar representaciones. El metamodelo es un sistema de conceptos que constituye un sistema de información.

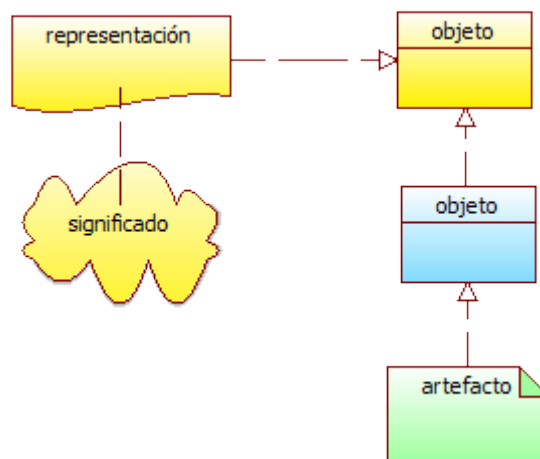


Fig. 10.11.: Metamodelo punto de vista de estructura de información

10.4.2 Modelo mInstituto

El manejo de información estará basado en el producto, este tendrá el manejo de comercialización y toda la parte de mercadeo entregando al cliente solo lo que impacte y genere en él una decisión de compra. Capacitación: brindar todas las herramientas necesarias para que se pueda hacer un uso correcto de la herramienta y que nos limite al máximo toda la parte de soporte post-venta. Documentación: mantener al día los manuales de usuario con nuevas funcionalidades así como nuevos desarrollos, tips y ayuda para cada una de la pantallas de la plataforma.

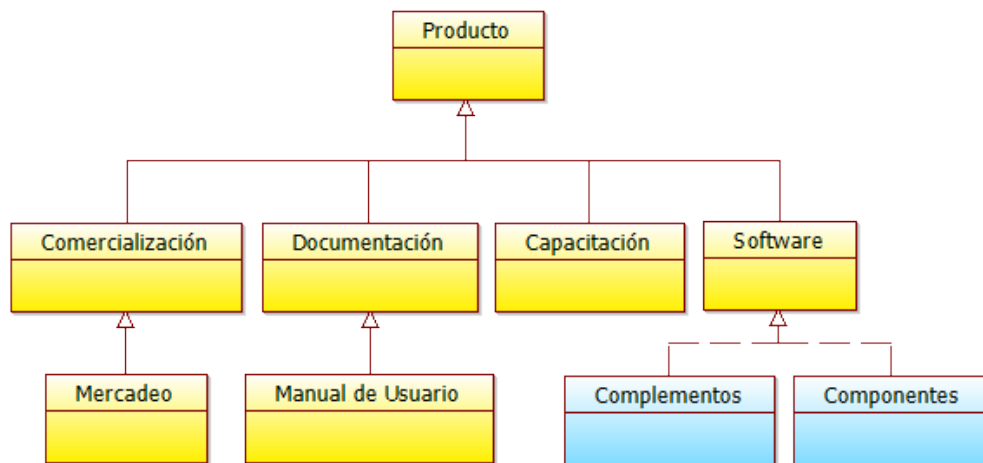


Fig. 10.12.: Modelo punto de vista de estructura de información: minstituto

10.5 Punto de vista de Realización del Servicio

El punto de vista de realización del servicio es usado para mostrar como uno o mas servicios de negocio son realizados por los procesos subyacentes(y algunas veces por componentes de aplicación). Así, se forma el puente entre el punto de vista de producto de negocio y la vista de proceso de negocio. Proporciona una "vista desde afuera"sobre uno o mas procesos de negocio.

Nombre	Realización del Servicio
Stakeholders	Proceso y de dominio, arquitectos y gerentes de producto operativos
Preocupaciones	Valor añadido de los procesos de negocio, Coherencia e integridad, responsabilidades
Propósito	Diseñar, Decidir
Nivel de Abstracción	Coherencia
Capa	Capa de Negocios (Aplicación)
Aspectos	Comportamiento, estructura, información

Tab. 10.5.: Descripción punto de vista de realización del servicio

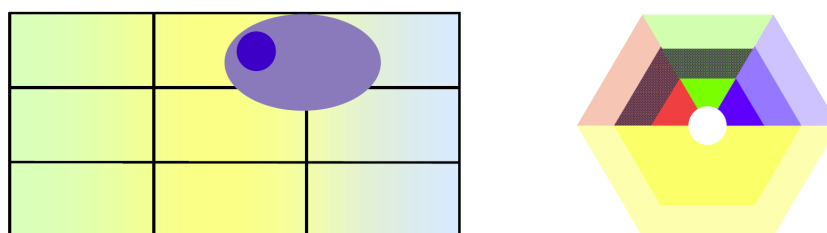


Fig. 10.13.: Posición del punto de vista de realización del servicio conceptualmente y marco del punto de vista

10.5.1 Metamodelo

El metamodelo del punto de vista de estructura de realización del servicio nos muestra cómo se modelan los elementos finales del negocio que empezamos a percibir, sus competencias al igual que las responsabilidades basado en la estructura y la información.

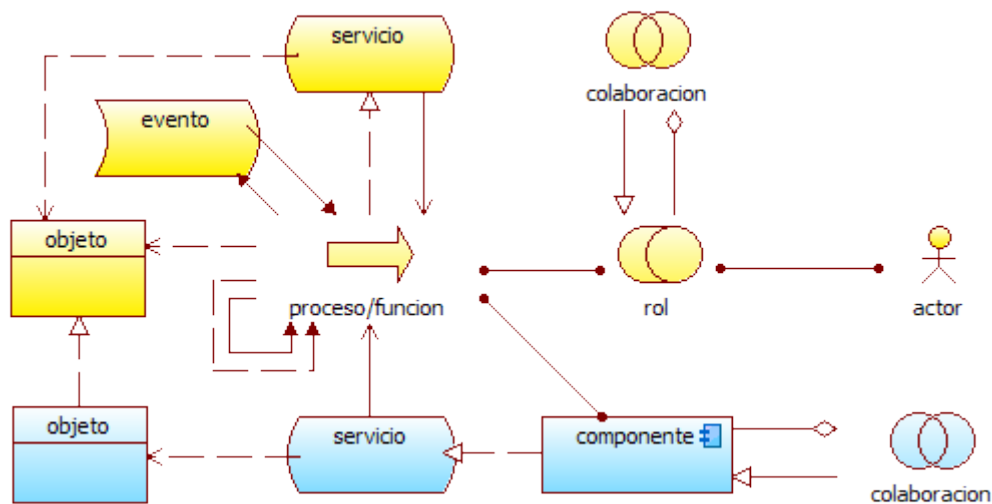


Fig. 10.14.: Metamodelo punto de vista de estructura de realización del servicio

10.5.2 Modelo mInstituto

En este modelo se presenta el producto que obtiene el cliente luego de un licenciamiento. Como punto de partida tenemos la comercialización que iniciará con una orden de compra y la aceptación de los términos y condiciones de uso de la plataforma, el cliente recibirá mes a mes su facturación a través del gestor contable. Software: MInstituto con todas las funcionalidades descritas en los términos y condiciones. Soporte: esa necesidad vital del ciclo del software que permite al aplicativo crecer por medio de la retroalimentación y generar comunicación y confianza de nuestros clientes a través de un excelente servicio post-venta, manejado por un gestor de clientes que recibirá todas las solicitudes en diferentes escalas y serán gestionadas por nuestros profesionales de las diferentes áreas.

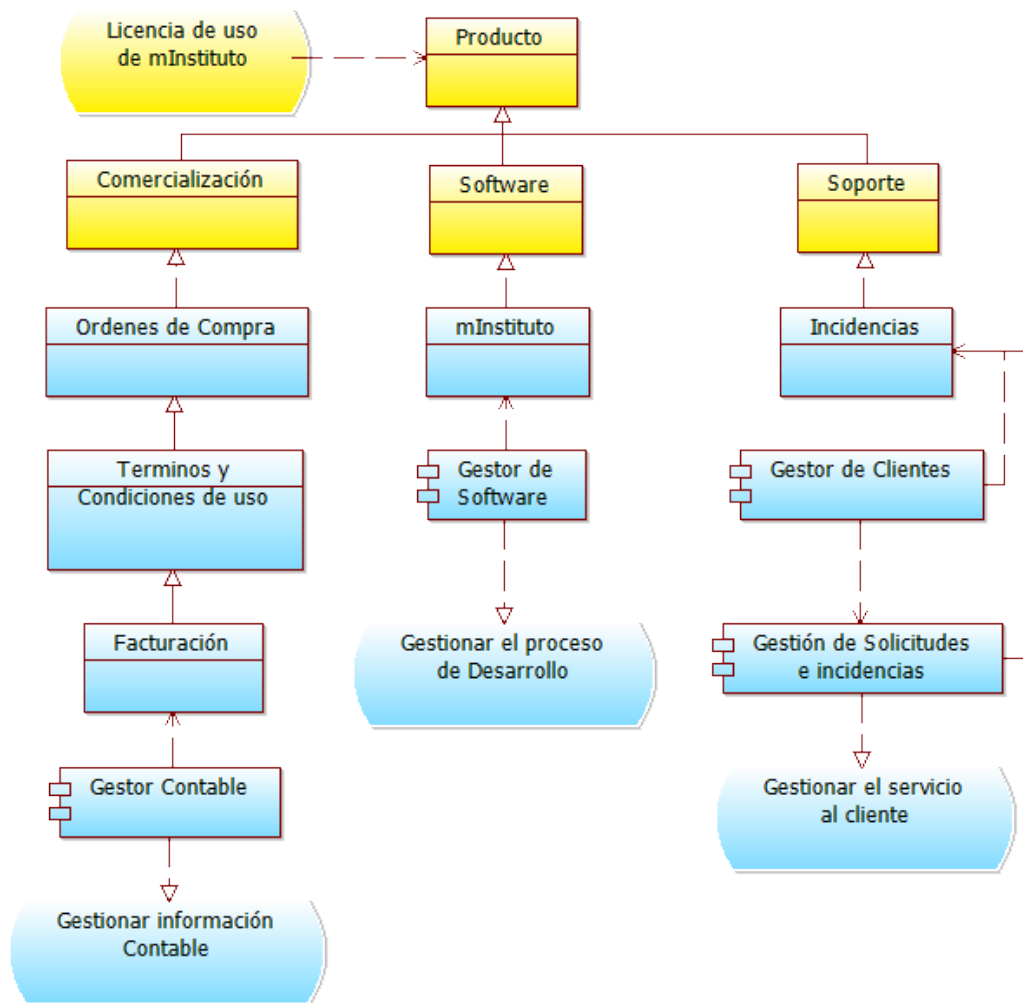


Fig. 10.15.: Modelo punto de vista de estructura de realización del servicio: minstituto

10.6 Punto de Vista de Capas

El punto de vista de capas ilustra en un diagrama de capas los aspectos de una arquitectura empresarial. Hay dos categorías de capas, capa dedicada y capa de servicios. Las capas son el resultado de la utilización de la relación “agrupación” de todo el conjunto de objetos y relaciones que pertenecen a un modelo. La infraestructura, la aplicación, el proceso y los actores / roles pertenecen a la categoría de capa dedicada. La Tabla 8.5 describe el punto de vista.

Nombre	Vista de Capas
Stakeholders	Arquitectos de Organización, proceso, aplicación, infraestructura y dominio
Preocupaciones	Consistencia, reducción de la complejidad, impacto del cambio y flexibilidad
Propósito	Diseñar, decidir, informar
Nivel de Abstracción	Información General
Capa	Capa de Negocio, Capa de Tecnología, Capa de Aplicación
Aspectos	Activo, comportamiento, pasivo

Tab. 10.6.: Descripción punto de vista de capas

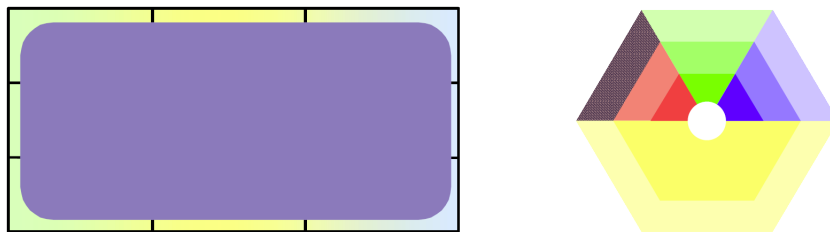


Fig. 10.16.: Posición del punto de vista de capas conceptualmente y marco del punto de vista

10.6.1 Metamodelo

El metamodelo Figura 8.9 expresa la conexión entre las capas por medio de los servicios o de los componentes que integran a cada una, la capa de infraestructura se relaciona por medio del servicio de aplicaciones con el componente de aplicación de la capa de aplicación, este componente a su vez se relaciona con el proceso de negocio de la capa de negocio y por ultimo este proceso de negocio se relaciona con los servicio de negocio de la organización.

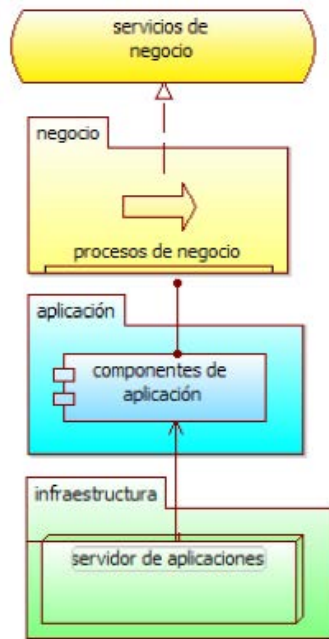


Fig. 10.17.: Metamodelo punto de vista de capas

Capa Motivacional

This chapter introduces the theoretical, experimental and computational concepts used throughout the thesis

Contenido

11.1 Punto de Vista de Stakeholder	92
11.1.1 Metamodelo	92
11.1.2 Modelo mInstituto	93
11.2 Punto de Vista de Realización de Objetivos	93
11.2.1 Metamodelo	94
11.2.2 Modelo mInstituto	95
11.3 Punto de Vista de Contribución	96
11.3.1 Metamodelo	97
11.3.2 Modelo mInstituto	97
11.4 Punto de Vista de Principios	98
11.4.1 Metamodelo	99
11.4.2 Modelo mInstituto	99
11.5 Punto de Vista de Realización de Requerimientos	100
11.5.1 Metamodelo	101
11.5.2 Modelo mInstituto	101
11.6 Punto de Vista de Motivación	102
11.6.1 Metamodelo	103
11.6.2 Modelo mInstituto	103

11.1 Punto de Vista de Stakeholder

El punto de vista de stakeholders nos permite al analizar para modelar las partes interesadas o implicados, los manejadores internos y externos para el cambio, y se efectúan evaluaciones (en términos de fortalezas, debilidades, oportunidades y amenazas) de estos manejadores. Los objetivos constituyen la base para el proceso de ingeniería de requisitos, incluyendo el refinamiento meta, contribución y análisis de conflictos.

Nombre	Vista de Stakeholder
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de la arquitectura y la estrategia, motivación
Propósito	Diseñar, decidir, informar
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de Negocios, Aplicación y tecnología
Aspectos	Motivación

Tab. 11.1.: Descripción punto de vista de stakeholders

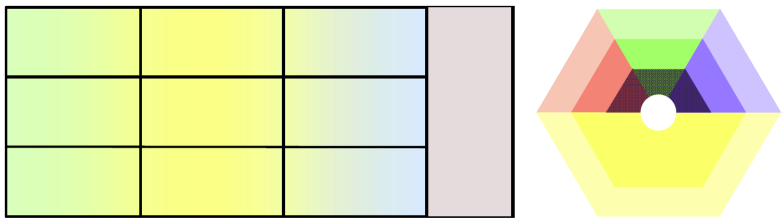


Fig. 11.1.: Posición del punto de vista de stakeholders conceptualmente y marco del punto de vista

11.1.1 Metamodelo

En la Figura 9.1 se ilustra el metamodelo del punto de vista de Stakeholder donde el pivote de todo es el objetivo, por medio del cual se establece la relación entre los stakeholders, los manejadores y la valoración o apreciaciones generadas desde los objetivos que le apuntan a situaciones propositivas para la organización; se hace énfasis en como los stakeholders juegan con los objetivos de la organización.

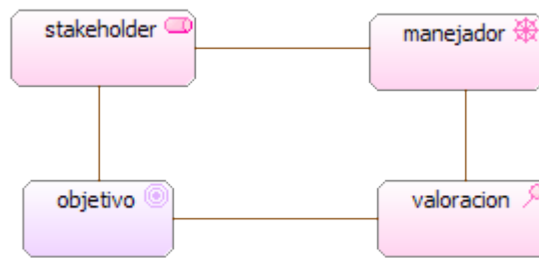


Fig. 11.2.: Metamodelo punto de vista de stakeholders

11.1.2 Modelo mInstituto

Dentro del modelo de stakeholder tenemos como principales componentes el desarrollador quien sera el encargado de proveer las soluciones tecnológicas de la organización, para nuestro caso el aplicativo para instituciones minstituto. También tenemos la parte comercial con un objetivo principal que es generar el reconocimiento de nuestra empresa y sus productos para cada día obtener una mayor participación del mercado.

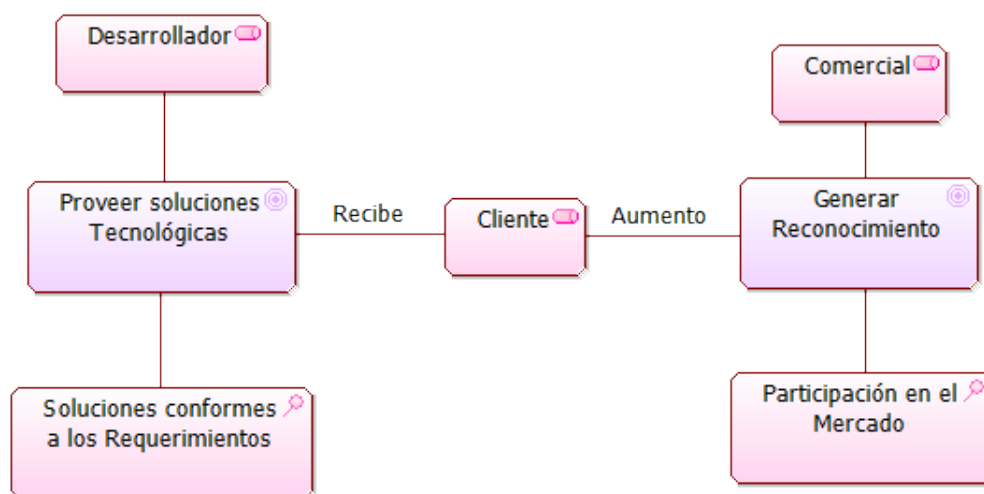


Fig. 11.3.: Modelo punto de vista de stakeholders: minstituto

11.2 Punto de Vista de Realización de Objetivos

El punto de vista de la realización de objetivos permite a un diseñador modelar y refinar los objetivos (de alto nivel) en objetivos más concretos, y el refinamiento de objetivos concretos en requisitos o limitaciones que describen las propiedades que se necesitan para alcanzar los objetivos. El refinamiento de objetivos en sub objetivos

se modela utilizando la relación de agregación. El refinamiento de objetivos en los requisitos se modela utilizando la relación realización.

Nombre	Realización de Objetivos
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de arquitectura, estrategia y táctica, motivación
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de negocio, tecnología
Aspectos	Motivación

Tab. 11.2.: Descripción punto de vista de la realización de objetivos

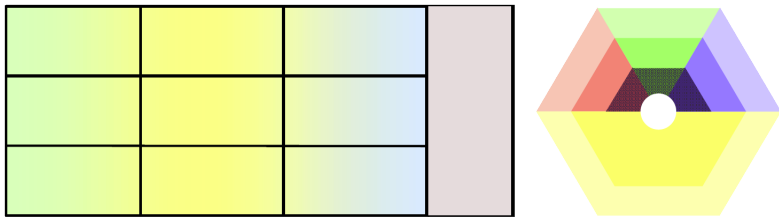


Fig. 11.4.: Posición del punto de vista de la realización de objetivos conceptualmente y marco del punto de vista

11.2.1 Metamodelo

En la Figura 9.3 se ilustra el metamodelo del punto de vista de Realización de Objetivos, donde un objetivo se convierte en un requerimiento, el objetivo se realiza con un requerimiento o una restricción.

Teniendo en cuenta que la restricción es un requerimiento pero lo diferencia porque tiene algún límite o alcance, el principio es un concepto para consolidar la arquitectura empresarial.

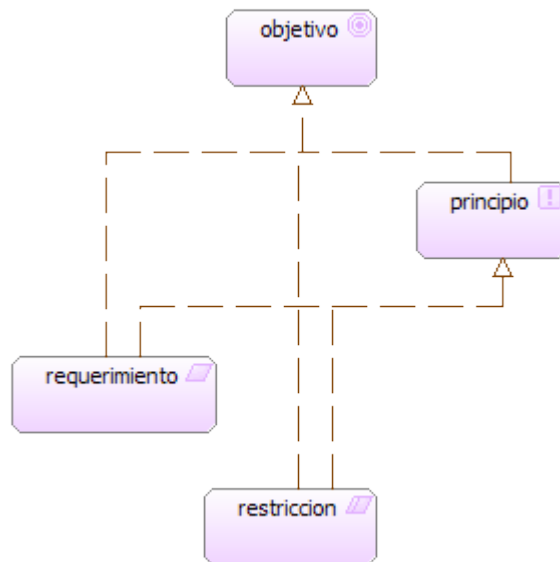


Fig. 11.5.: Metamodelo punto de vista de la realización de objetivos

11.2.2 Modelo mInstituto

Como principal objetivo de la organización tenemos proveer soluciones tecnológicas con un primer producto minstituto; cuya funcionalidad se basa en optimizar la gestión de la información en las instituciones educativas. Es importante resaltar que este no será un desarrollo a la medida de cada cliente sino que manejaremos un paquete de funcionalidades comunes englobadas en unas licencias de uso estándar.

Otro de nuestros objetivos es que los productos software nos den el reconocimiento necesario en cada uno de los mercados a los que nos enfoquemos. Esto solo será posible si tenemos clientes que puedan ser esos casos exitosos que podamos mostrar.

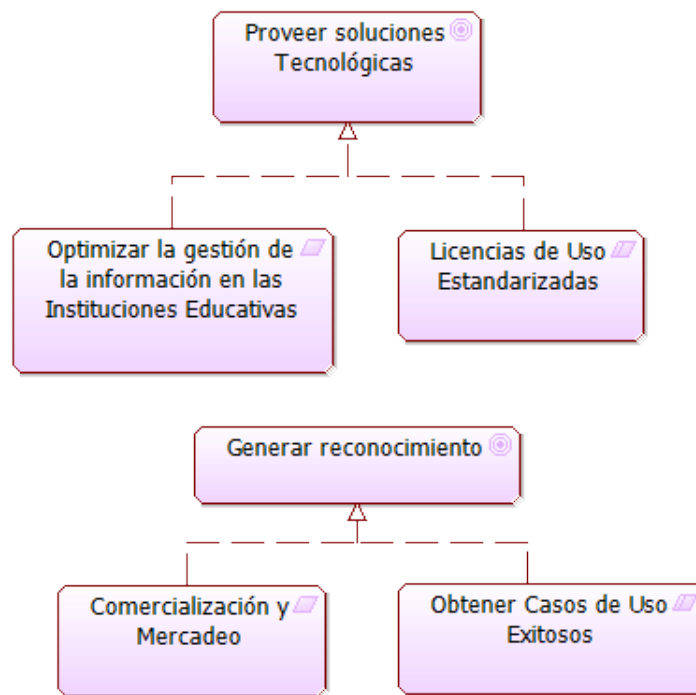


Fig. 11.6.: Modelo punto de vista de la realización de objetivos: minstituto

11.3 Punto de Vista de Contribución

El punto de vista de la contribución de objetivos permite a un diseñador o analista modelar las relaciones de influencia entre los objetivos y los requisitos. Los puntos de vista resultantes pueden utilizarse para analizar el impacto que las metas tienen para detectar conflictos entre los objetivos de las partes interesadas. La Tabla 9.3 describe el punto de vista.

Nombre	Vista de Contribución
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de arquitectura, estrategia y táctica, motivación
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de negocio, Capa de tecnología
Aspectos	Motivación

Tab. 11.3.: Descripción punto de vista de la contribución

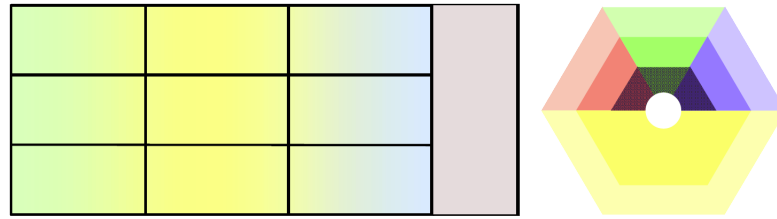


Fig. 11.7.: Posición del punto de vista de la contribución conceptualmente y marco del punto de vista

11.3.1 Metamodelo

En la figura Figura 9.5 se ilustra el metamodelo del punto de vista de Contribución, en el metamodelo se realiza la relación de contribución, esta relación tiene algo a destacar y son las contribuciones positivas y negativas, se apunta a objetivos que aporten de manera positiva y/o negativa hacia los requerimientos que se tratan de diferenciar.

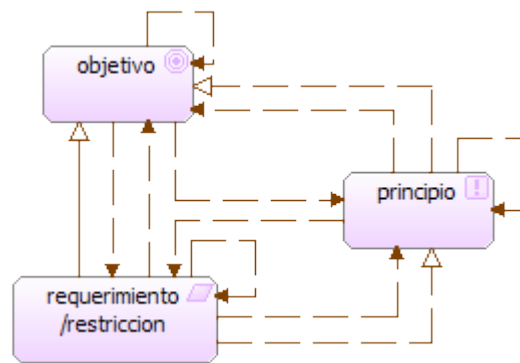


Fig. 11.8.: Metamodelo punto de vista de la contribución

11.3.2 Modelo mInstituto

La contribución que se espera generar con Minstituto.com en las instituciones educativas es la optimización de sus procesos administrativos generando agilidad y eficiencia en los mismos y logrando un mayor flujo de información influyendo en la agilidad del proceso de toma de decisiones con base en información actualizada y en reportes gerenciales que den una visión global dela organización en diferentes intervalos de tiempo.

Tenemos que tener en cuenta que para poder ser ese apoyo organizacional debemos hacer ver nuestro producto como una inversión antes que un costo sin valores agregados y que vamos a estar con ellos realizando el acompañamiento necesario

siendo un aliado estratégico en cada una de las etapas del proceso de implementación de un software dentro de una institución, así como en un proceso de capacitación que les permita obtener la mayor utilidad y explotar al máximo su funcionalidad.

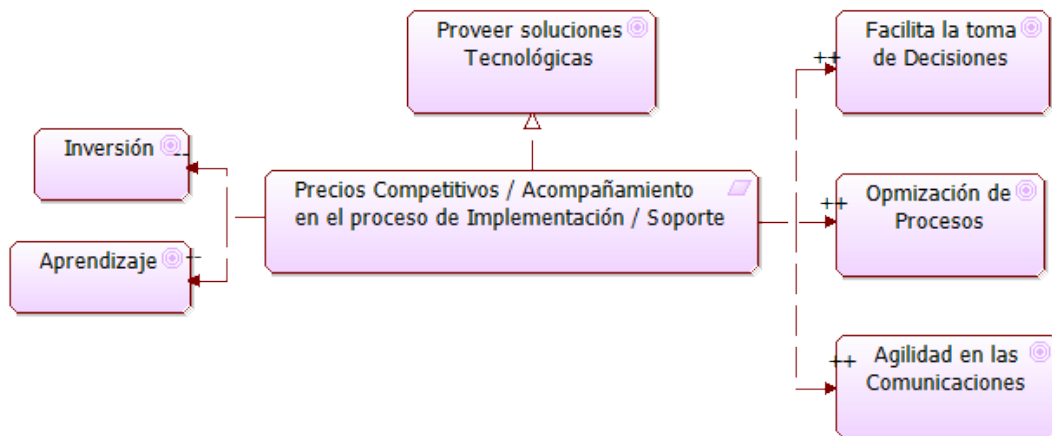


Fig. 11.9.: Modelo punto de vista de la contribución: minstituto

11.4 Punto de Vista de Principios

El punto de vista de los principios permite al analista o diseñador modelar los principios que son relevantes para el problema a diseñar, se incluyen los objetivos que motivan a estos principios. Además, las relaciones entre los principios y sus objetivos, pueden ser modelados. Por ejemplo, los principios pueden influir entre sí positiva o negativamente.

Nombre	Vista de Principios
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de arquitectura, estrategia y táctica, motivación
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de negocio, tecnología
Aspectos	Motivación

Tab. 11.4.: Descripción punto de vista de los principios

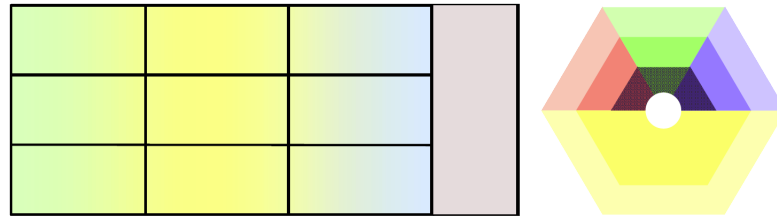


Fig. 11.10.: Posición del punto de vista de los principios conceptualmente y marco del punto de vista

11.4.1 Metamodelo

En la figura Figura 9.7 se ilustra el metamodelo del punto de vista de Principios, los principios son características que le dan caracterización, matizan o resaltan a la organización en los objetivos, le asocia adjetivos que apuntalan la naturaleza de la organización.

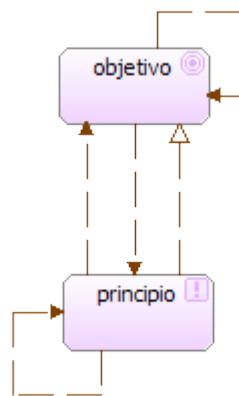


Fig. 11.11.: Metamodelo punto de vista de los principios

11.4.2 Modelo mInstituto

Cuando hablamos de principios de nuestra organización es importante resaltar que debemos tener un sistema que cumpla con los estándares de confidencialidad: Salvaguardar la información. Integridad: Que los datos sean consistentes y suplan las necesidades para la toma de decisiones correctas. Disponibilidad: disponer de un gestor que le brinde a cada usuario los datos autorizados y que de igual manera estén siempre visibles en el momento en que los clientes los quieran consultar. Innovación: Es importante resaltar que nuestra empresa y sus productos deben estar en un constante cambio para estar al día con las nuevas propuestas, tecnologías y exigencias del mercado.

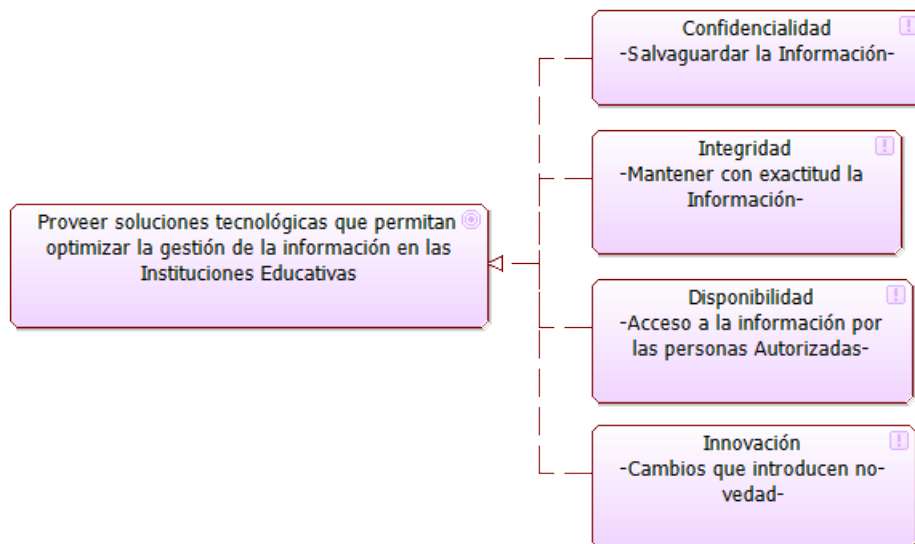


Fig. 11.12.: Modelo punto de vista de los principios: minstituto

11.5 Punto de Vista de Realización de Requerimientos

El punto de vista de realización de requerimientos permite al diseñador modelar la realización de los requerimientos para los elementos base o core, tales como los actores de negocios, servicios de negocios, procesos de negocios, servicios de aplicaciones, componentes de la aplicación, etc. Por lo general, los requisitos son el resultado del punto de vista de refinamiento del objetivo. Además, este punto de vista se puede usar para refinar los requisitos en requisitos más detallados. La relación de agregación se usa para este propósito.

Nombre	Realización de Requerimientos
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de arquitectura, estrategia y táctica, motivación
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de Aplicación, tecnología
Aspectos	Motivación

Tab. 11.5.: Descripción punto de vista de realización de requerimientos

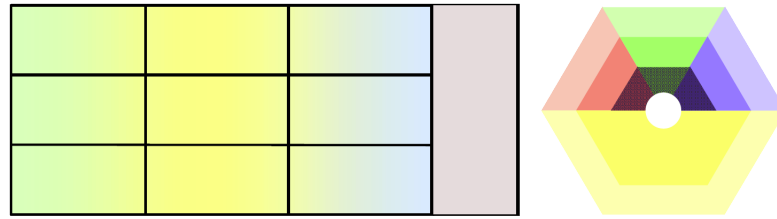


Fig. 11.13.: Posición del punto de vista de realización de requerimientos conceptualmente y marco del punto de vista

11.5.1 Metamodelo

En la Figura 9.9 se ilustra el metamodelo del punto de vista de Realización de Requerimientos, la realización de requerimientos tiene un vocabulario que se aleja del negocio y pone más en términos de requerimiento, se toma el objetivo y se desagrega; el lenguaje utilizado es un lenguaje apropiado para entenderlo

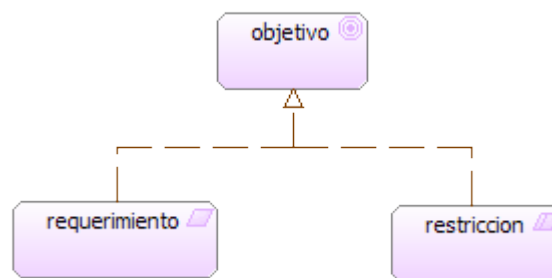


Fig. 11.14.: Metamodelo punto de vista de realización de requerimientos

11.5.2 Modelo mInstituto

Para proveer soluciones tecnológicas que permitan optimizar la gestión de la información en las instituciones educativas debemos conocer la dinámica de estas, identificar sus necesidades y dificultades a las que se enfrentan cada día al no tener una solución tecnológica que les permita disponer de información actualizada y precisa de la gestión de la institución, se debe concientizar al cliente de que la solución tecnológica representa una inversión y no un costo y debe ser un entorno amigable de fácil navegación.

Con respecto a las licencias debemos alinearnos con el modelo de requerimientos ofrecido a los clientes.

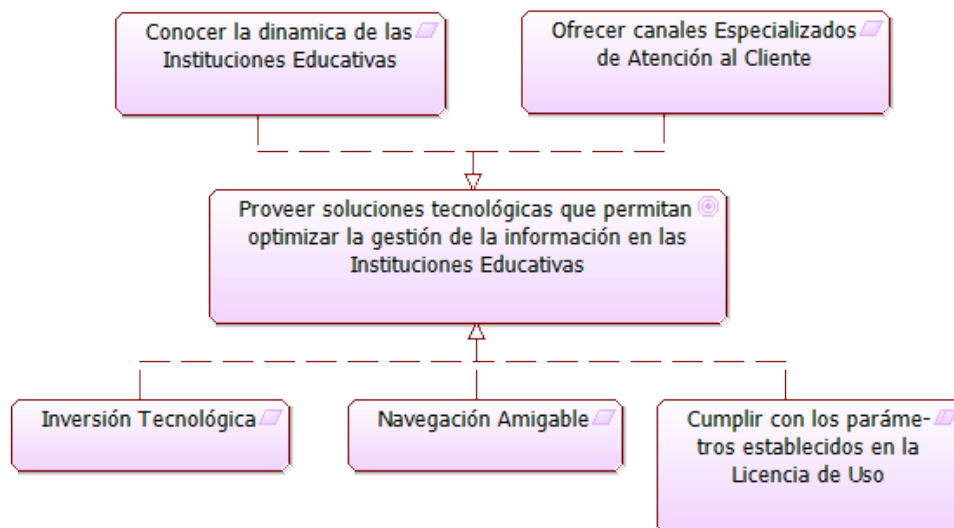


Fig. 11.15.: Modelo punto de vista de realización de requerimientos: minstituto

11.6 Punto de Vista de Motivación

El punto de vista de la motivación permite al diseñador o analista modelar el aspecto motivacional, sin centrarse en ciertos elementos dentro de este aspecto. Por ejemplo, este punto de vista se puede utilizar para presentar una visión completa o parcial del aspecto de motivación por parte de los stakeholders, los principales objetivos, los principios que se aplican, y los requerimientos principales de los servicios, procesos, aplicaciones y objetos.

Nombre	Motivación
Stakeholders	Stakeholders, administradores de empresas, arquitectos TIC, analistas de negocios, Analistas de requerimientos
Preocupaciones	Misión de arquitectura, estrategia y táctica, motivación
Propósito	Diseñar, decidir
Nivel de Abstracción	Coherencia, detalle
Capa	Capa de Aplicación, tecnología
Aspectos	Motivación

Tab. 11.6.: Descripción punto de vista de la motivación

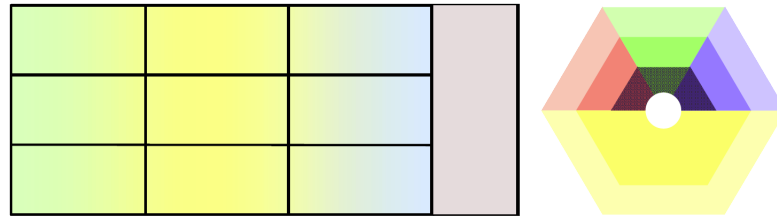


Fig. 11.16.: Posición del punto de vista de la motivación conceptualmente y marco del punto de vista

11.6.1 Metamodelo

En la Figura 9.11 se ilustra el metamodelo del punto de vista de Motivación, donde el punto angular es el participante que tiene una valoración y cuenta con un objetivo clave el cual se relaciona con los requerimientos, al participante se le asocia el manejador que es una asociación es parecida a un objetivo pero es propia del participante, se puede decir que es un objetivo del participante. Con este punto de vista se muestra la esencia de la parte motivacional. De esto se puede decir que el punto de vista se centra en el participante lo que él hace, los requerimientos que el genera, en la valoración que el produce, la asociación con el objetivo organizacional y el manejo que tiene con respecto al negocio.

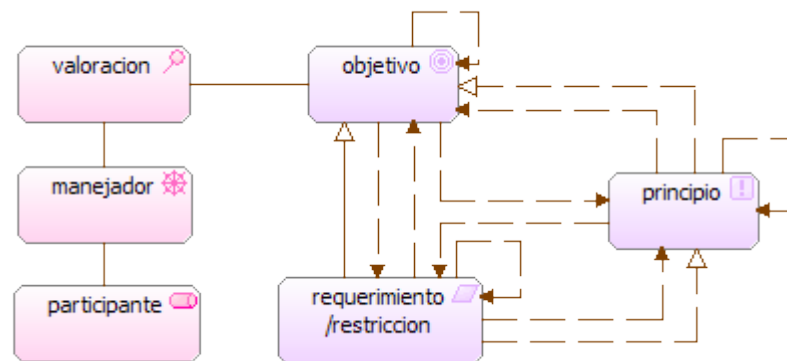


Fig. 11.17.: Metamodelo punto de vista de la motivación

11.6.2 Modelo mInstituto

Como principal motivación del proyecto a desarrollar por parte de nuestra organización es tener esa participación en el mercado de las instituciones educativas, que nos permita crecer como organización y como miembros de la misma. Sin embargo, para lograr ese objetivo es necesario que nuestro software esté en capacidad de soportar la dinámica de negocio de nuestros clientes, manteniendo la confidencialidad y disponibilidad de su información.

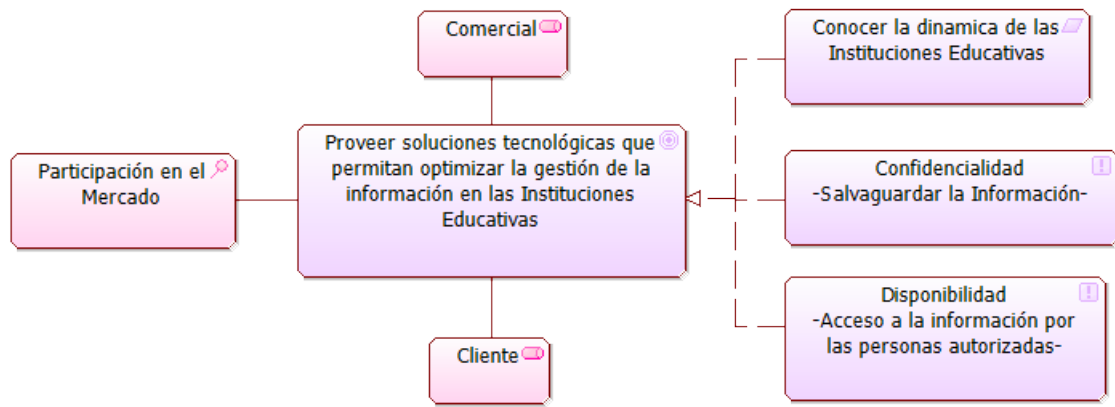


Fig. 11.18.: Modelo punto de vista de la motivación: minstituto

This chapter introduces the theoretical, experimental and computational concepts used throughout the thesis

Contenido

12.1 Punto de Vista de Proyecto	106
12.1.1 Metamodelo	107
12.1.2 Modelo mInstituto	107
12.2 Punto de Vista de Migración	108
12.2.1 Metamodelo	108
12.2.2 Modelo mInstituto	109
12.3 Punto de Vista de Migración e Implementación	109
12.3.1 Metamodelo	110
12.3.2 Modelo mInstituto	111

12.1 Punto de Vista de Proyecto

Un punto de vista del proyecto se utiliza principalmente para modelar el cambio de arquitectura. El proceso de migración de la “arquitectura” desde una situación o estado anterior a una nueva situación tiene consecuencias importantes en la estrategia de crecimiento a largo y mediano plazo, y el proceso de toma de decisiones. Algunas de las cuestiones que deben tenerse en cuenta por los modelos diseñados en este punto de vista son:

- El desarrollo de la arquitectura empresarial en toda la organización es una tarea que puede requerir varios años.
- Todos los sistemas y servicios deben seguir funcionando sin importar todas las modificaciones presumibles y cambios en la arquitectura de la empresa durante el proceso de cambio.
- El proceso de cambio puede tener que lidiar con estándares tecnológicos inmaduros (por ejemplo, mensajería, seguridad, datos, etc.).
- El cambio tiene consecuencias graves para el personal, la cultura, la forma de trabajo y la organización.

Nombre	Proyecto
Stakeholders	Gerentes (operativos), empresarios, arquitectos TIC, empleados, accionistas
Preocupaciones	Visión y políticas de la arquitectura, motivación
Propósito	Decidir, informar
Nivel de Abstracción	Conjunto
Capa	Capa de implementación y migración
Aspectos	Estructura pasiva, comportamiento, estructura activa

Tab. 12.1.: Descripción punto de vista del proyecto

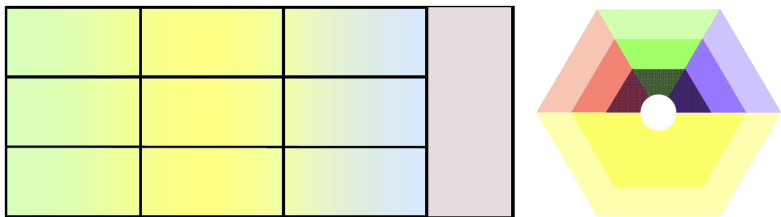


Fig. 12.1.: Posición del punto de vista del proyecto conceptualmente y marco del punto de vista

12.1.1 Metamodelo

En la Figura 10.1 se ilustra el metamodelo del punto de vista de Proyecto donde se resuelven los objetivos a partir de paquetes de trabajo, el paquete de trabajo es el conjunto de actividades con miras a generar entregables, esto está asociado a roles que a su vez está asignado a actores; los paquetes resuelven objetivos.

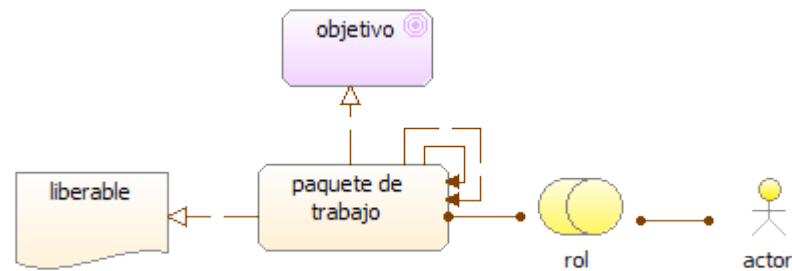


Fig. 12.2.: Metamodelo punto de vista del proyecto

12.1.2 Modelo mInstituto

En la vista se representa el vínculo existente entre elementos de la vista de motivación y los componentes presentes en la vista de negocio donde el objetivo de negocio principal es el de proveer soluciones tecnológicas que permitan optimizar la gestión de la información en las instituciones educativas, a este objetivo se encuentran asociados los roles principales de negocio, es decir, el desarrollador y el cliente. El paquete de trabajo unido al objetivo corresponde al desarrollo de la solución tecnológica.

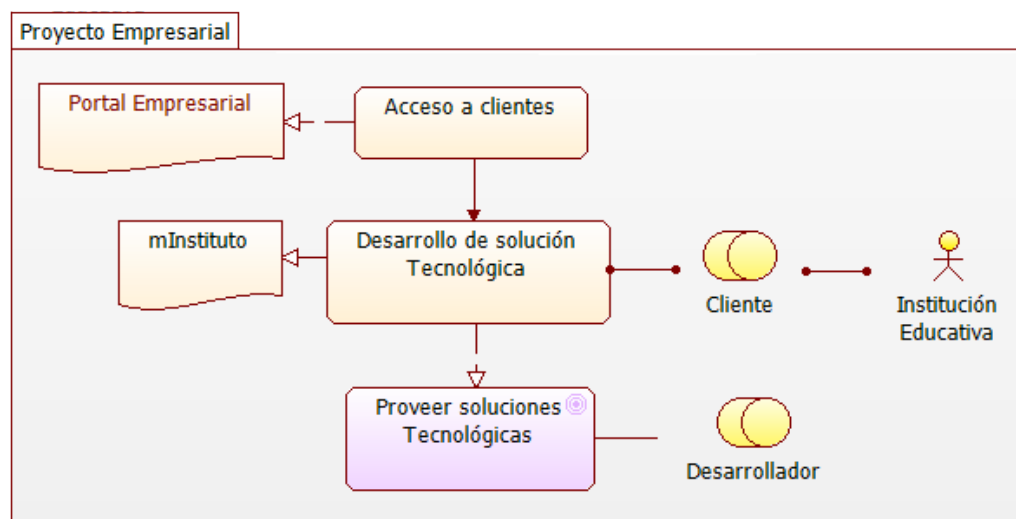


Fig. 12.3.: Modelo punto de vista del proyecto: minstituto

12.2 Punto de Vista de Migración

El punto de vista de migración implica modelos y conceptos que pueden ser utilizados para especificar la transición de una arquitectura existente a una arquitectura deseada.

El punto de vista de migración implica modelos y conceptos que pueden ser utilizados para especificar la transición de una arquitectura existente a una arquitectura deseada.

Nombre	Migración
Stakeholders	Arquitectos empresariales, arquitectos de proyecto, arquitectos de aplicación, arquitectos de dominio, arquitectos de infraestructura, empleados
Preocupaciones	Historia de los modelos
Propósito	Diseñar, decidir, informar
Nivel de Abstracción	Conjunto
Capa	Capa de implementación y migración
Aspectos	NA

Tab. 12.2.: Descripción punto de vista de migración



Fig. 12.4.: Posición del punto de vista de migración conceptualmente y marco del punto de vista

12.2.1 Metamodelo

En la Figura 10.3 se ilustra el metamodelo del punto de vista de Migración, En este punto de vista están los conceptos de brecha y platea, se deben tener en cuenta los hitos y como se puede pasar de un hito a otro. Una brecha es lo que se debe superar para alcanzar el otro lado, la platea es el hito en el que nos apoyamos para progresar en la arquitectura empresarial.



Fig. 12.5.: Metamodelo punto de vista de migración

12.2.2 Modelo mInstituto

En la vista se representan las plateas y brechas evidenciadas para el proyecto. Se identificaron como primera platea los requerimientos de la gestión de información de las instituciones educativas; la segunda platea es el análisis y diseño, la brecha entre las dos plateas corresponde al levantamiento de requerimientos, el cual se realiza con la información obtenida de los requerimientos. La tercera platea corresponde al licenciamiento y la brecha entre análisis y diseño y el licenciamiento es el proceso de desarrollo del software y las pruebas correspondientes.

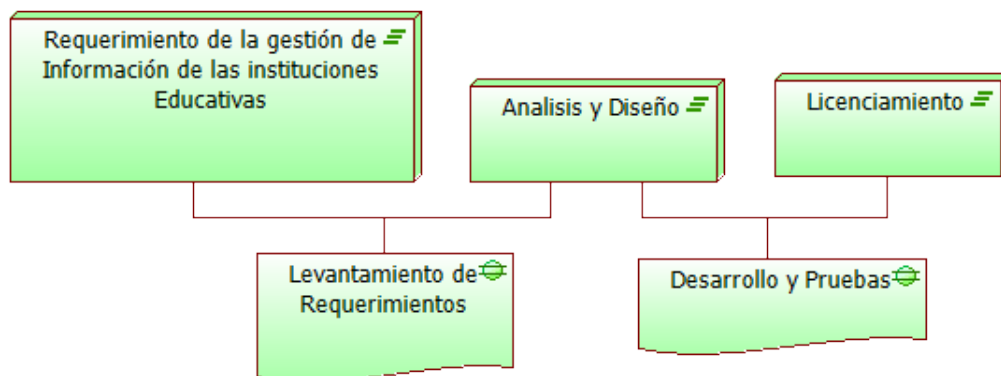


Fig. 12.6.: Modelo punto de vista de migración: minstituto

12.3 Punto de Vista de Migración e Implementación

El punto de vista de la implementación y migración se utiliza para relacionar los programas y proyectos con las partes de la arquitectura que se implementan. Esta visión permite el modelado del alcance de los programas, proyectos, actividades del proyecto en términos de los elementos individuales de la arquitectura que se ven afectados. Además, se puede indicar la forma en que los elementos se ven afectados por las relaciones entre componentes.

Nombre	Migración e Implementación
Stakeholders	Gerentes (operativos), organización, arquitectos TIC, empleados, accionistas
Preocupaciones	Visión y políticas de la arquitectura, motivación
Propósito	Diseñar, informar
Nivel de Abstracción	Conjunto
Capa	Capa de negocio, Aplicación, tecnología, implementación y extensión de la migración
Aspectos	Estructura pasiva, comportamiento, estructura activa

Tab. 12.3.: Descripción punto de vista de la implementación y migración



Fig. 12.7.: Posición del punto de vista de la implementación y migración conceptualmente y marco del punto de vista

12.3.1 Metamodelo

En la figura Figura 10.3 se ilustra el metamodelo del punto de vista de Migración e Implementación, es la conjunción integración de los puntos de vista de proyecto y de migración, es una síntesis del despliegue, nos resume el proyecto y la migración.

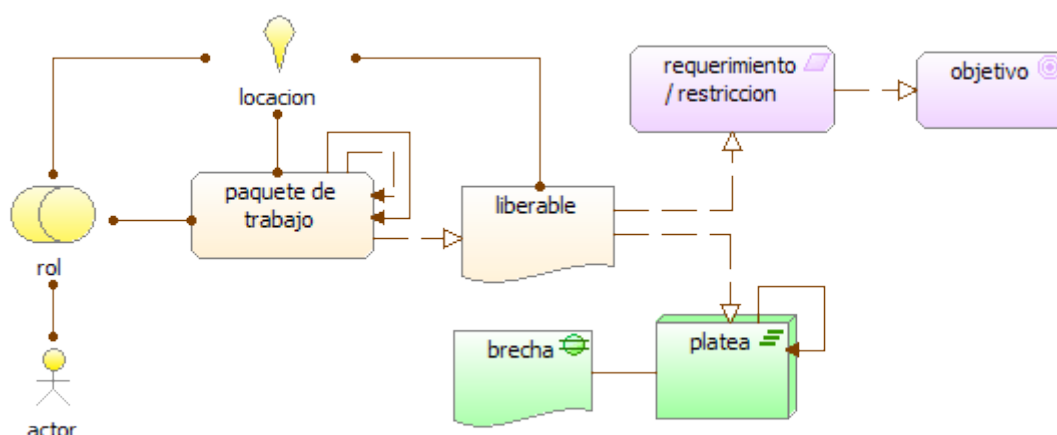


Fig. 12.8.: Metamodelo punto de vista de la implementación y migración

12.3.2 Modelo mInstituto

En la vista se evidencia el paquete de trabajo que es el desarrollo de la solución tecnológica, su objetivo es proveer soluciones tecnológicas de gestión de información de las instituciones educativas, para el cumplimiento del objetivo se identificaron instrumentos puntuales que permiten su ejecución los cuales son, requerimientos específicos del negocio, análisis y diseño y el licenciamiento con sus respectivas brechas. Se identifica como punto de unión entre los modelos de migración y proyecto el software con los requerimientos de la gestión de información de las instituciones educativas.

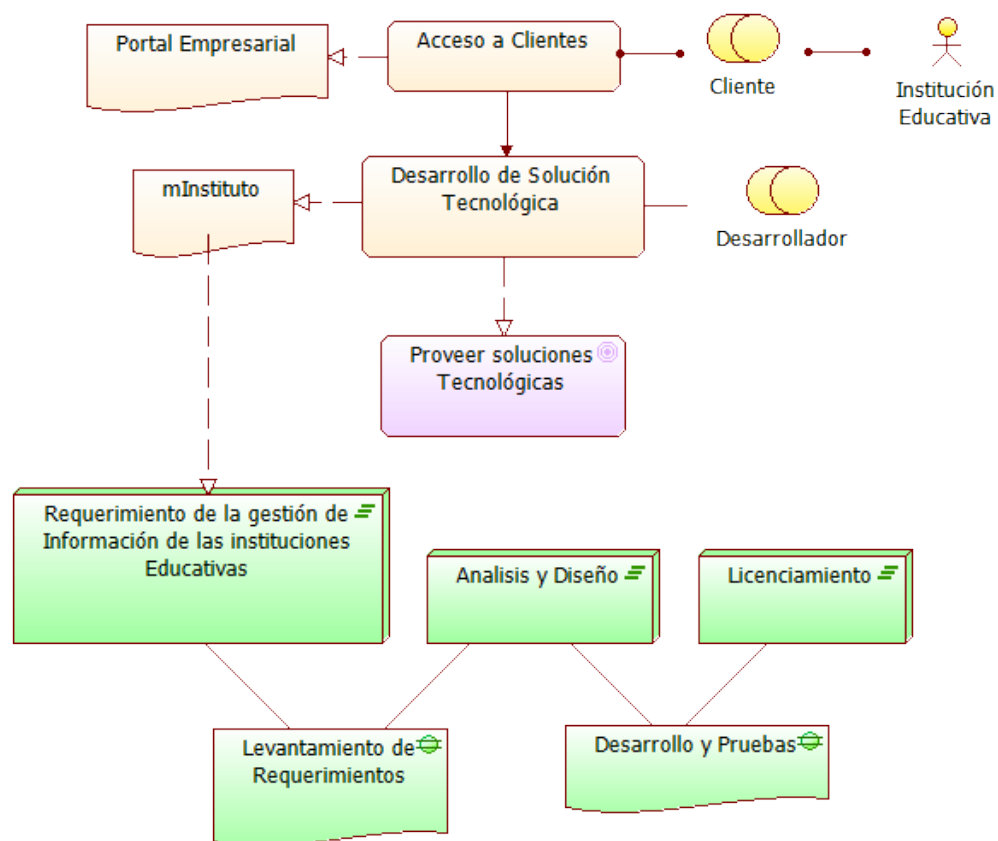


Fig. 12.9.: Modelo punto de vista de la implementación y migración: minstituto

Parte III

Conclusiones

CONCLUSIONES

Se evidenció que Archimate fue un lenguaje apropiado a las necesidades de identificación de factores y relaciones entre los mismos para el desarrollo de la arquitectura empresarial ya que a través de los diferentes puntos de vista que propone se logró abarcar todos los procesos y alinearlos con la filosofía organizacional.

El desarrollo de la Arquitectura Empresarial permitió identificar debilidades y fortalezas a nivel interno y oportunidades y amenazas a nivel externo teniendo en cuenta los diferentes stakeholders.

La alineación de todos los procesos con los objetivos estratégicos permite que las políticas y planes de acción generados en los procesos estratégicos y de apoyo tengan el mismo horizonte que los procesos misionales.

La arquitectura empresarial permite definir el estado ideal en el que se desea que se encuentre la organización, contribuyendo a la generación de una planeación estratégica con bases realmente sólidas que garantice que la organización cumple las expectativas y requerimientos de las instituciones educativas en cuando a necesidades de gestión y se encuentra en capacidad de responder acertadamente a los cambios que se presentan.

TRABAJOS FUTUROS

Con el desarrollo del proyecto, el siguiente paso para la empresa Creatics se centrara en la realización de un estudio de factibilidad del proyecto de diseño y desarrollo del software en donde se tendrán en cuenta variables económicas y otras variables influyentes como lo es el estudio de mercados.

Posterior al estudio de factibilidad se procederá a realizar el diseño y desarrollo de la solución tecnológica que permita optimizar la gestión de la información en las instituciones educativas, este desarrollo de software tendrá como base y fundamento la arquitectura empresarial propuesta para la empresa y la aplicación de los patrones necesarios.

[5, 6, 8, 7, 1, 2, 3]123

[@4, @9]456

Aple apricot avocado

Daniel

Bibliografía

- [1] Apple Inc. *Keynote '09 User Guide*. Apple Inc., 2010 (vid. pág. 117).
- [2] Apple Inc. *Numbers '09 User Guide*. Apple Inc., 2010 (vid. pág. 117).
- [3] Apple Inc. *Pages '09 User Guide*. Apple Inc., 2010 (vid. pág. 117).
- [5] Manuela Jürgens. *LaTeX: eine Einführung und ein bisschen mehr*. FernUniversität Gesamthochschule in Hagen, 2000 (vid. pág. 117).
- [6] Manuela Jürgens. *LaTeX: Fortgeschrittene Anwendungen*. FernUniversität Gesamthochschule in Hagen, 1995 (vid. pág. 117).
- [7] Markus Kohm y Jens-Uwe-Morawski. *KOMA-Script: Die Anleitung*. 2011 (vid. pág. 117).
- [8] André Miede. *A Classic Thesis Style: An Homage to The Elements of Typographic Style*. 2011 (vid. pág. 117).

Sitios Web

- [@4] Free Software Foundation, Inc. *GNU General Public License*. 2010. URL: <http://www.gnu.org/licenses/gpl.html> (visitado 27 de mayo de 2011) (vid. pág. 117).
- [@9] André Miede. *A Classic Thesis Style by André Miede*. 2011. URL: <http://www.miede.de/index.php?page=classicthesis> (visitado 27 de mayo de 2011) (vid. pág. 117).

Parte IV

Anexos

Patrones

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

A.1 Cadena de Responsabilidad

Permite a un objeto enviar un comando sin conocer que objeto o objetos lo recibirán, evitando el acoplamiento entre el que envía y el que recibe una petición. Esto permite el paso del comando a un objeto de la cadena que es parte de una gran estructura. Cada objeto de la cadena podría manejar el comando, pasar el comando al siguiente objeto de la cadena, o las dos cosas.

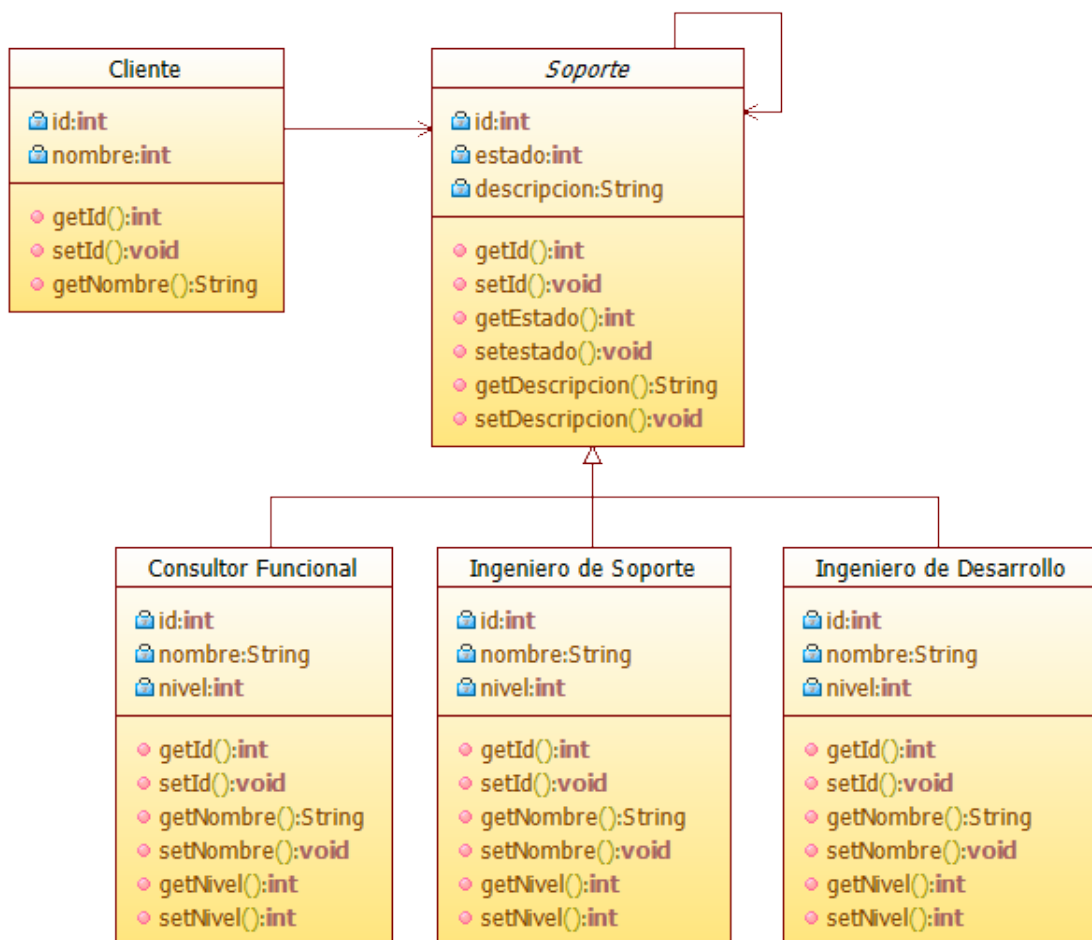


Fig. A.1.: Patrón cadena de responsabilidad mInstituto

A.1.1 Patrón cadena de responsabilidad mInstituto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A.1.2 Clases cadena de responsabilidad

```
package Cadena;
public class Cliente {
    private int id;
    private String nombre;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

Listing A.1: Cliente.java

```
package Cadena;
public abstract class Soporte {
    private int id;
    private int estado;
    private String descripcion;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getEstado() {
        return estado;
    }
    public void setEstado(int estado) {
        this.estado = estado;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```

```
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```

Listing A.2: Soporte.java

```
package Cadena;
public class Consultor extends Soporte {
    private int id;
    private String nombre;
    private int nivel;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public int getNivel() {
        return nivel;
    }
    public void setNivel(int nivel) {
        this.nivel = nivel;
    }
}
```

Listing A.3: Consultor.java

A.2 Fabrica Abstracta

Dado un conjunto de clases abstractas relacionadas, el patrón Abstract Factory permite el modo de crear instancias de estas clases abstractas desde el correspondiente conjunto de subclases concretas. Proporciona una interfaz para crear familias de objetos relacionados o dependientes sin especificar su clase concreta. El patrón Abstract Factory puede ser muy útil para permitir a un programa trabajar con una variedad compleja de entidades externas, tales como diferentes sistemas de ventanas con una funcionalidad similar.

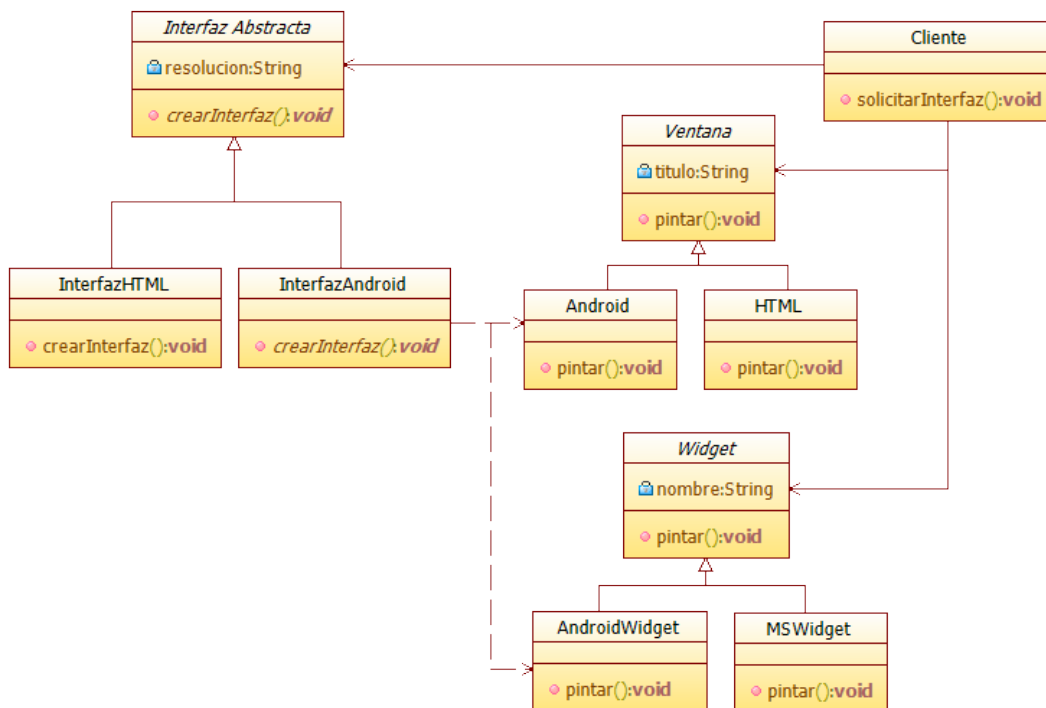


Fig. A.2.: Patrón fabrica abstracta mInstituto

A.2.1 Patrón Fabrica Abstracta mInstituto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a,

turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A.2.2 Clases fabrica abstracta

```
package Factory;
public class Cliente {
    public void ↵
        solicitarInterfaz() {
    }
}
```

Listing A.4: Cliente.java

```
package Factory;
abstract class Ventana {
    private String titulo;
    public String getTitulo() {
        return titulo;
    }
    public void setTitulo(String↵
        titulo) {
        this.titulo = titulo;
    }
    public abstract void pintar↵
        ();
}
```

Listing A.5: Ventana.java

```
package Factory;
public class Android extends ↵
    Ventana {
    @Override
    public void pintar() {
    }
}
```

Listing A.6: Android.java

```
package Factory;
public class html extends ↵
    Ventana {
    @Override
    public void pintar() {
    }
}
```

Listing A.7: html.java

```
package Factory;
abstract class Widget {
    private String nombre;
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String↵
        nombre) {
        this.nombre = nombre;
    }
    public abstract void pintar↵
        ();
}
```

Listing A.8: Widget.java

```
package Factory;
public class AndroidWidget ↵
    extends Widget {
    @Override
    public void pintar() {
    }
}
```

Listing A.9: AndroidWidget.java

```
package Factory;
public class MSWidget extends ↵
    Widget {
    @Override
    public void pintar() {
    }
}
```

Listing A.10: MSWidget.java

```
package Factory;
abstract class InterfazAbstracta↵
    {
    private String resolucio↵
        n;
    public abstract void ↵
        crearInterfaz();
}
```

Listing A.11: InterfazAbstracta.java

```

package Factory;
public class InterfazHTML <↵
    extends InterfazAbstracta{
        @Override
        public void crearInterfaz() <↵
            {
            }
    }
}

```

Listing A.12: InterfazHTML.java

```

package Factory;
public class InterfazAndroid <↵
    extends InterfazAbstracta{
        @Override
        public void crearInterfaz() <↵
            {
            }
    }
}

```

Listing A.13: InterfazAndroid.java

A.3 Decorador

Extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase.

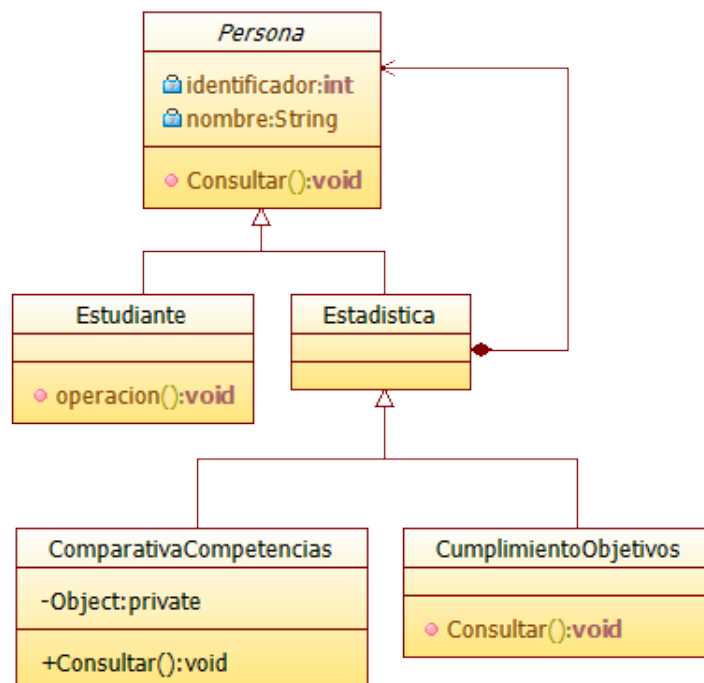


Fig. A.3.: Patrón decorador mInstituto

A.3.1 Patrón Decorador mInstituto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

A.3.2 Clases Decorador

```
package Decorator;
public abstract class Persona {
    private int identificador;
    private String nombre;
    public int getIdentificador() {
        return identificador;
    }
    public void setIdentificador(int identificador) {
        this.identificador = identificador;
    }
    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    public abstract void Consultar();
}
```

Listing A.14: Persona.java

```
package Decorator;
public class Estudiante extends Persona {
    @Override
    public void Consultar() {
    }
}
```

Listing A.15: Estudiante.java

```
package Decorator;
public abstract class Estadistica extends Persona {
    {
    }
}
```

Listing A.16: Estadistica.java

```
package Decorator;
public class ComparativaCompetencias extends Estadistica {
    @Override
    public void Consultar() {
    }
}
```

Listing A.17: ComparativaCompetencias.java

```
package Decorator;
public class CumplimientoObjetivos extends Estadistica {
    @Override
    public void Consultar() {
    }
}
```

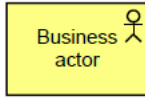

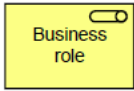

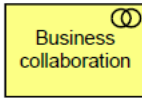
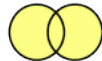
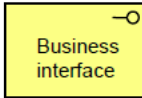




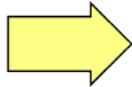




Listing A.18: CumplimientoObjetivos.java



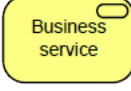

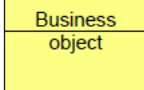
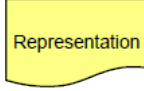
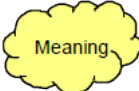
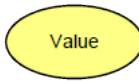
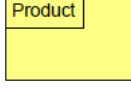
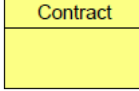
Resumen Simbología Archimate

2.1

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

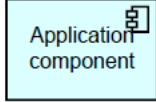
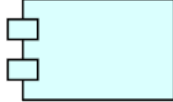
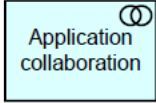




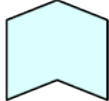


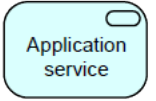
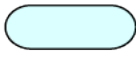
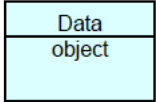
B.1 Capa de Negocios

Concepto	Definición	Notación	
Actor	Una entidad de organización que es capaz de realizar la conducta a través de un rol.		
Role	La responsabilidad de llevar a cabo un comportamiento específico, al que se le puede asignar un actor.		
Colaboración	Un agregado de dos o más funciones de negocios que trabajan en conjunto para llevar a cabo el comportamiento colectivo.		
Interface	Un punto de acceso donde se pone a disposición un servicio de negocio al entorno.		
Localización	Un punto conceptual o extensión en el espacio.		
Proceso	Un elemento de la conducta que el comportamiento de los grupos basa en un ordenamiento de las actividades. Se tiene la intención de producir un conjunto definido de productos o servicios.		
Función	Un elemento de la conducta que el comportamiento de los grupos basada en una selección en un conjunto seleccionado de criterios (recursos empresariales suelen ser necesarios y / o competencias).		
Interacción	Un elemento de comportamiento que describe el comportamiento de una colaboración de negocios.		

Evento	Algo que sucede (interna o externa) e influye en el comportamiento (de negocios proceso, la función empresarial, la interacción de negocios).		
Servicio	Un servicio que satisface una necesidad comercial de un cliente (interno o externo a la organización).		
Objeto	Un elemento pasivo que tiene relevancia desde un punto de vista comercial.		
Representación	Una forma perceptible de la información transportada por un objeto de negocio.		
Sentido	El conocimiento o experiencia presente en un objeto de negocio o de su representación, dado un contexto particular.		
Valor	El valor relativo, utilidad o importancia de un servicio de negocio o producto.		
Producto	Una colección coherente de los servicios, acompañada de un contrato / conjunto de acuerdos, que se ofrece en su conjunto a los clientes (internos o externos).		
Contrato	Una especificación formal o informal de acuerdo que especifica los derechos y obligaciones inherentes a un producto.		

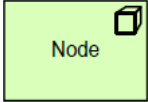
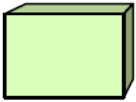
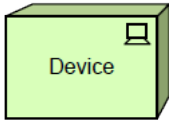

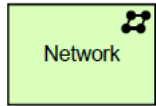

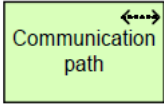

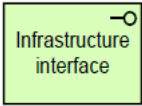

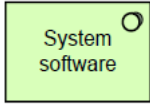
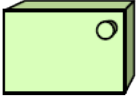
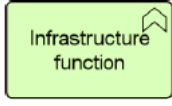

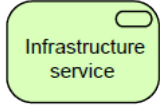
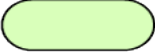
Tab. B.1.: Simbología capa de Negocios



B.2 Capa de Aplicación

Concepto	Definición	Notación	
Componente	Parte modular, desplegable, y reemplazable de un sistema de software que encapsula su comportamiento y los datos y expone a estos a través de un conjunto de interfaces.		
Colaboración	Un agregado de dos o más componentes de aplicaciones que funcionan en conjunto para llevar a cabo el comportamiento colectivo.		
Interfaces	Un punto de acceso donde se pone a disposición un servicio de aplicaciones para un usuario u otro componente de la aplicación.		
Función	Un elemento de comportamiento que los grupos de comportamiento que puede ser realizada por un componente de aplicación automatizado.		
Interacción	Un elemento de comportamiento que describe el comportamiento de una aplicación colaborativa.		
Servicio	Un servicio que expone comportamiento automatizado.		
Objeto	Un elemento pasivo que permite su tratamiento automatizado.		

Tab. B.2.: Simbología capa de Aplicación

B.3 Capa de Tecnología

Concepto	Definición	Notación	
Nodo	Un recurso computacional sobre los que los artefactos se pueden almacenar o desplegar para su ejecución.		
Dispositivo	Un recurso de hardware sobre el cual los artefactos se pueden almacenar o desplegar para su ejecución.		
Red	Un medio de comunicación entre dos o más dispositivos.		
Ruta de comunicación	Un enlace entre dos o más nodos, a través del cual estos pueden intercambiar datos.		
Infraestructura	Un punto de acceso en caso de los servicios de infraestructura ofrecidos por un nodo puede acceder a otros nodos y componentes de la aplicación.		
Software	Un entorno de software para los tipos específicos de componentes y objetos que se despliegan en ella en forma de artefactos.		
Función	Un elemento de comportamiento que agrupa infraestructura. Comportamiento que puede ser realizado por un nodo.		
Servicio	Una unidad externa visible de la funcionalidad proporcionada por uno o más nodos, expuesta a través de interfaces bien definidas, y significativo para el entorno.		



Artefacto	Una pieza física de datos que se utilizan o producen en un proceso de desarrollo de software, o por la implementación y operación de un sistema.	 
-----------	--	--

Tab. B.3.: Simbología capa de Tecnología

B.4 Relaciones



B.4.1 Estructurales

Concepto	Definición	Notación
Asociación	Los modelos de asociación de una relación entre los objetos que no está cubierta por otro, la relación más específica.	—————
Acceso	La relaciones de acceso en los modelos unen los conceptos de comportamiento a los objetos de negocio o de datos.>
Usado por	La relación de uso une los servicios por parte de los procesos, funciones o las interacciones y el acceso a las interfaces de funciones, componentes o colaboraciones.	————>
Realización	La relación realización vincula a una entidad lógica con una entidad más concreta que se da cuenta de ello.▷
Asignación	La relación de asignación vincula unidades de comportamiento con los elementos activos (por ejemplo, funciones, componentes) que las llevan a cabo, o roles con actores que las cumplen.	•————•

Agregación	La relación de agregación indica que un grupo de objetos es una serie de otros objetos.	
Composición	La relación composición indica que un objeto está compuesto por uno o más de otros objetos.	

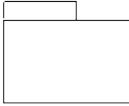

Tab. B.4.: Simbología de Relaciones Estructurales

B.4.2 Dinámicas

Concepto	Definición	Notación
Flujo	La relación de flujo describe el intercambio o de transferencia, por ejemplo, la información o el valor entre los procesos, funciones, interacciones y eventos.	
Disparo	La relación de disparo describe las relaciones temporales o causales entre procesos, funciones, interacciones y eventos.	

Tab. B.5.: Simbología de Relaciones Dinámicas



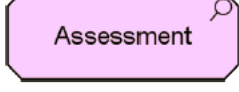

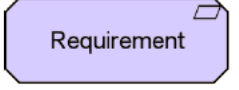
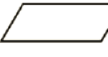
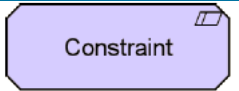

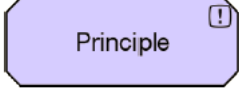
B.4.3 Otras

Concepto	Definición	Notación
Grupo	La relación indica que la agrupación de objetos del mismo tipo o de tipos diferentes, uno para el otro basado en alguna característica común.	
Unión	Una unión se utiliza para conectar las relaciones del mismo tipo.	

Especialización	La relación especialización indica que un objeto es una especialización de otro objeto.	→
-----------------	---	---

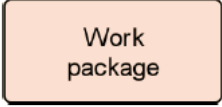


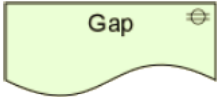
Tab. B.6.: Simbología de otras Relaciones

B.5 Motivación

Concepto	Definición	Notación
Stakeholder	El papel de un individuo, equipo u organización (o clases de los mismos) que representa sus intereses en, o preocupaciones en relación con, el resultado de la arquitectura.	
Manejador	Algo que crea, motiva y alimenta el cambio en una organización.	
Valoración	El resultado de un análisis de algún conductor.	
Objetivo	Un estado final de que una parte interesada tiene la intención de lograr.	
Requerimiento	Una declaración de necesidad que debe ser realizado por un sistema.	 
Restricción	Una restricción en el modo en que se realiza un sistema.	 
Principio	Una propiedad normativo de todos los sistemas en un contexto dado, o la forma en que se realizan.	

Tab. B.7.: Simbología capa de Motivación

B.6 Implementación y Migración

Concepto	Definición	Notación
Paquete de trabajo	Una serie de acciones destinadas a lograr un objetivo único dentro de un tiempo especificado.	
Entregable	Un resultado definido con precisión de un paquete de trabajo.	
Platea	Un estado relativamente estable de la arquitectura que existió durante un período de tiempo limitado.	
Brecha	Uno de los resultados de un análisis de la brecha entre dos plateas.	

Tab. B.8.: Simbología capa de Implementación y Migración

A Software Architecture Proposal Artistic Engineering Environment -AEE

The first chapter introduces fluorescence-based DNA technology and highlights the motivation of the research conducted in the thesis

A Software Architecture Proposal Artistic Engineering Environment -AEE-

Sandro Javier Bolaños Castro

Faculty of Engineering, District University Francisco José de Caldas, Bogotá, Colombia

Rubén González Crespo

College of Engineering and Architecture, Pontifical University of Salamanca, Madrid, Spain

Abstract—this article presents the “Artistic Engineering Environment” framework (AEE). It is an environment that supports service-centered components. These components facilitate software development based on a mixture of engineering and art. The aim is to promote the combination of these two disciplines to empower software engineering. Our fundamental purpose is to present the architecture that supports AEE.

Index Terms—Software engineering, art, framework, architecture.

I. INTRODUCTION

Process-centered Software Development Environments (PSEE) [1], tools like CASE (Computer Assisted Software Engineering) [2], and, in general, software intended to support development processes focus on facilitating programming, design, testing or management tasks, neglecting important aspects such as aesthetics, harmony, and software use itself [3]. Environments should not be swiss army knives or golden hammers [4], [5], they should have intersection points between complexity, aesthetics and functionality. The simplicity of a given tool is reflected in its easy use. Functionality must be complemented by style and an aesthetic awareness, which provide a harmonious set of engineering and art that leads to quality [6].

II. THE PHILOSOPHY OF ART-ENGINEERING ENVIRONMENT

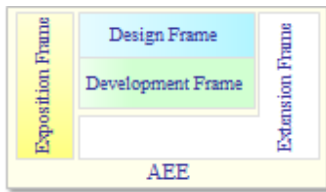


Fig. 1: AEE Scheme.

AEE draws attention to two fundamental pillars, namely engineering and art, Fig. 1.

A. Engineering

From an engineering point of view [7], AEE proposes a set of principles as follows:

1) *Simplicity*: Inspired by Occam’s razor principle [8], simplicity [9] represents the other side of the coin when addressing a problem, not from the standpoint where complexity is frightening right from the start, but from a peaceful counterpoint where the job to be done is suggested to be comfortable and manageable. AEE reflects its simplicity in a set of well-defined principles, even from an implementation perspective when we look at the small set of interfaces to be implemented.

2) *The Metaphor*: AEE’s framework was created using an art gallery as a metaphor [10]. The gallery offers frames for Exposition, Expression and Extension, and the purpose of the frames is to depict the nature of an application from viewpoints that range from general artifact creation to a view of the most specific artifacts. The main concept of this metaphor is the frame; this concept gathers internal semantics according to the environment. The Exposition frame, for example, holds the concept of author reference and also a description of the artifacts arranged in a hierarchical tree. The expression frame, on the other hand, is used to depict the work of art in the environment, highlighting the works that resemble paintings (i.e. modeling) and also those that resemble literature (i.e. development). The extension frame makes it possible to expand expression semantics using two approaches, namely an approach intended to integrate the different views about a problem, and a presentation approach that highlights visuals.

3) *Organization*: By creating a visual distribution, the aim is to have an impact upon creative development, which focuses on deduction, that is, we start from a framework and go down to the details making a thorough scanning of the concept from left to right and top to bottom. In an intentional fashion, the organization [11] decides that the Exposition frame, from which they reference the “work-of-art” project, be placed on the left, emphasizing that this project should be done first. The center is occupied by the expression frame, while the different views of the extension frame are placed at the bottom and the presentation of this latter frame is placed on the right.

B. Art

From the Artistic viewpoint, two principles stand out, namely aesthetics and style.

1) *Aesthetics*: This principle refers to aesthetics as the branch of philosophy that studies both the essence and the perception of beauty [12], [13] reflected in the color given to AEE. Yellow, blue and green are the colors chosen to paint frames. Yellow, which is the color chosen for the Exposition frame, reflects creativity and brightness [14]. Blue, which is the color chosen for the expression framework, represents harmony, science and intelligence [14]. Finally, green, which was a color reserved for the extension frame, represents everything that is fresh and functional. [14].

2) *Style*: This principle promotes customized environments, allowing the adjustment of iconography. At first universally recognized icons are proposed as a component; however, the environmental icon suite can be assembled according to the intentions. By default AEE's style [15] promotes the use of warm colors so that a sense of comfort, simplicity and easiness pervades the environment.

III. ARCHITECTURE

Software architecture is the structure of all structures within the system. This structure comprises software components and their properties together with the way they relate to each other [16]. Shaw and Garlan consider software architecture as the starting point, where modules are established and relations are defined [17]. Architecture allows configuring a system without having to get into implementation details; moreover, architecture leaves a distinguishing mark on every software solution, similar to conventional architecture and actual buildings. AEE's representation architecture uses graphic representations such as modules, interfaces and components that illustrate the way in which the environment's structure was conceived.

A. Three-tier Model

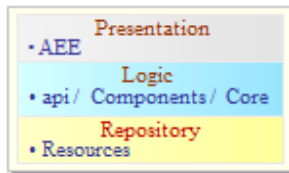


Fig. 2: AEE Multi-tier Architecture.

The most general view of the environment is defined using a layered architectural style [18], which consists of a presentation layer, a logical layer, and a repository layer. Fig 2.

In the presentation layer, the *ArtisticEngineeringFrame* module is defined as the fundamental abstraction, and it represents the ingress point to the platform. In the logical layer, the API (Application Programming Interface) is established. The following elements are condensed into the API: the application semantics, the components together with a set of services required by the environment, and the core, which is the assembly of all the frames that permit the construction of applications over the environment. Finally, the repository layer comprises the application resources, where icons and configuration files can be found.

B. AEE as a Framework

A framework is an architectural pattern that provides an extensible template for applications, facilitates development, and promotes components reuse. Within a framework, an architectural skeleton is specified together with slots, thumb indexes and fragments that are provided for users, who shall adapt the framework for their own context [19], [20].

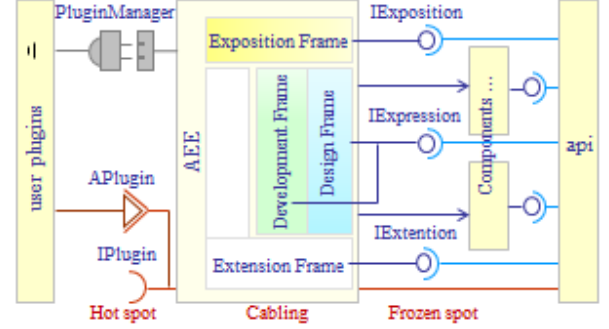


Fig. 3: AEE Framework.

The AEE framework has three abstraction layers, Fig. 3. The first layer comprises the api. This layer is completely abstract and defines the set of primary services from which Frozen Spots and Hot Spots are obtained. A Frozen Spot is the point in an application that has already been implemented and keeps the essential conditions for the framework to work; AEE has three main Frozen Spots that represent the interfaces implementation, namely *IExposition*, *IExpression* and *IExtension*. These spots appear together with other spots that are implemented with some components that will be used by the environment. A Hot Spot is a point that enables extending and customizing applications. These spots are left for users to implement them. The main Hot Spot is *IPlugin*, which allows creating connectors to the environment; these connectors will be managed using the *PluginManager* module.

The second layer of the framework is the cabling layer. This layer consists of the environment's frames, namely Exposition, Expression and Extension, included in and managed from the *ArtisticEngineeringFrame*, which is the starting point to assemble the user's customized applications. The last layer is the user layer, also known as user plugin. User's applications to be deployed in the environment are found in this last layer.

C. API

API is a mechanism to conceive applications from a higher level of abstraction. APIs facilitate problem understanding from software upper layers; they also improve modularity [21] and constitute a fundamental ingredient for the components assembly.

Application programmable interfaces model the main functionalities that will be provided by the environment. For the Exposition frame implemented by the *IExposition* interface, functionalities of some renown are developed, namely movement, erasure and closing. The Expression frame implemented by the *IExpression* interface develops the following functionalities: storage, erasure and printing. Furthermore, the

Exposition frame implements the special interface called *IPart*, which allows any extension of this concept to be considered as a multiple-piece component of the user's customized jigsaw. Finally, the Extension frame should implement the *IExtension* interface, which is the interface that makes it possible to add special controls to manipulate its content, e.g. copy and erasure controls. Fig. 4.



Fig. 4: api AEE.

D. Component

A component represents a modular piece from either a logical or physical system whose external behavior can be more consistently described than its implementation. Components are connected to one another through interfaces, allowing a free exchange and replacement of components as long as the new components are supported by the corresponding interface. Likewise, a component in the inside can be made up of other components so as to constitute more complex modules [20]. In order to promote code reuse, components must be standardize, independent, compose-able, deploy-able and document-able [22]. Fig. 5.

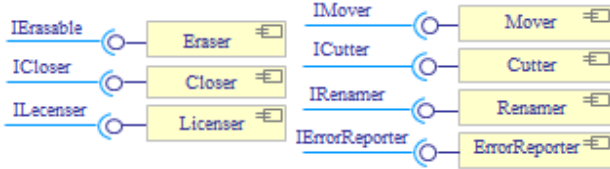


Fig. 5: AEE Components.

AEE Components, Figure 5, allow solving the following functionalities:

- *Eraser*: eliminates a particular node from the Exposition frame tree.
- *Closer*: closes the thumb indexes addressed by the selected node within the expression frame, which in turn also closes its corresponding extension frame.
- *Mover*: allows moving an Exposition frame leaf-node to any package within the same Exposition framework.
- *Renamer*: allows changing the name of a node.
- *Cutter*: is used in the extension frame to take a copy of the corresponding frame as an image and so write reports easily.
- *Licenser*: manages the information that will be displayed by the environment for copyright purposes.
- *ErrorReporter*: was designed to deal with the defects that may appear in the environment. It will automatically send reports to an Internet e-mail, or else to an error log.

E. Frames

A particular type of organization for the art-engineering environment is the one made by using the frames approach. Frames establish the limit and the modular content. Frames are represented as high-level classes and serve to model the Exposition, the Expression and the Extension in an environment.

1) *Exposition Frame* : The Exposition frame *ExpositionFrame* uses the functionalities implemented by the following components: *Eraser*, *Closer*, *Mover* and *Renamer*. This allows managing the application from this framework. Such functionalities represent the realization commitment as expressed through the interface *IExposition*, but the commitment is delegated to each of the components according to the delegation principle [23], which promotes reuse and flexibility within the application. Fig. 6.

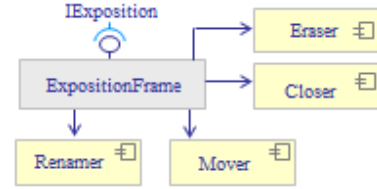


Fig. 6: Exposition Frame.

2) *Expression Frame*: The expression frame *AExpressionFrame* has two specialized possibilities, namely the development framework and the modeling framework. Fig.7.

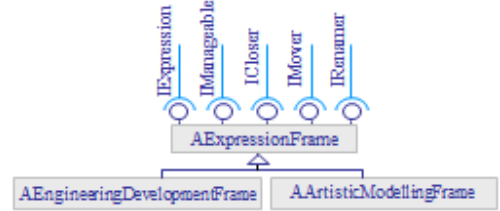


Fig. 7: Expression Frame.

The development framework can be used for the integration of plugins such as text editors, whereas the modeling framework is used for the integration of design plugins. An expression frame is managed through control services, namely closing, movement, and renaming; moreover, plugins can be printed, stored and deleted.

3) *Extension Frame* : The Extension frame *ExtensionFrame* allows the expression-frame information to be expanded through both perspective and presentation specializations.

Perspective permits having expression-frame projections, which allows the semantics of an application to be expanded. Some of the projections are, for example, a metrics view or a profile view. The perspective specialization uses the component *Cutter* to take a screenshot that can be cut and then pasted somewhere else, which facilitates making reports. The presentation specialization is primarily used to display the expression-frame information in a hierarchical fashion (tree). It can also be used to display a miniaturized view of the expression frame when the size of such a frame makes it

difficult to explore it. An extension frame is controllable, that is, it can have controls like the erasure control. Fig. 8.



Fig. 8: Extension Frame.

4) *Artistic-Engineering Frame*: The artistic-engineering frame *ArtisticEngineeringFrame* is the fundamental frame. This frame comprises the exposition, expression and extension frameworks. Plugins are managed from this principal frame by using the *PluginManager* module. The artistic-engineering frame also manages resources, licensing (copyright) and the error reporter by using the corresponding components. This fundamental frame is located in the presentation layer and constitutes the framework's ingress point, Fig. 9.

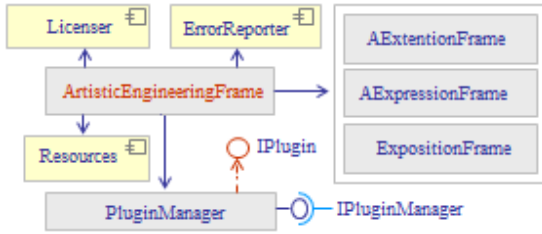


Fig. 9: Artistic-Engineering Frame.

F. Resources

Resources are a basic component of the environment, which is made up of menus, labels, dialogues, tool-tips, and icons. Internationalization is managed through resources [24], which are fundamental to the universalization of an application. Initially, the platform supports two languages, namely English and Spanish, and it has the possibility to be easily extended to more languages without altering the structure of the application.

G. AEE as Plugin Support

A plugin is an application that is linked to other application in order to provide it with more or better functionalities, normally very specific functionalities, Fig. 10.

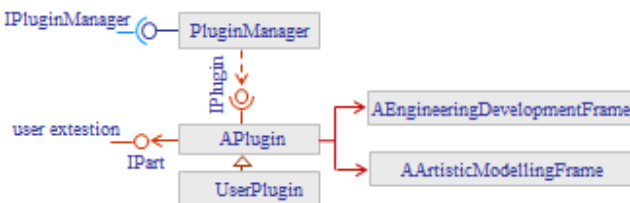


Fig. 10: AEE as Plugin Support.

In AEE the benefits of the plugin concept are exploited. The fundamental idea is to allow new components to be plugged to the existent system. The AEE-plugins architecture focuses its

development of plugins on the *APlugin* class, which should specialize in a particular plugin according to the developer (user).

The *APlugin* class has the user's engineering development frame specializations and the artistic modeling specializations, as well as the extensions, by means of the interface *IPart*. *APlugin* is an implementation of *IPlugin*, which is uploaded on the AEE platform using reflection mechanisms, exploiting the dynamic-language characteristics of java. This plugin task is performed by *PluginManager*, the specific role of this module is to read the plugin's route, i.e. the manifesto [25], which indicates the properties of a given plugin.

H. AEE design Patterns

Christopher Alexander put forward the following definition: "Each pattern describes a problem, which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it in the same way twice." [26].

1) *GoF Patterns*: The GoF design patterns [26], together with other architectural patterns [27], [28], compile a set of design templates whose solutions permit software to be flexible and robust. For the AEE platform, two patterns are used to establish the relationship between the different parts, namely mediator and observer.

a) *Mediator*: One of the main difficulties is the strong interaction between the Exposition and Expression frames due to the massive information flow between them. The expressions that need to be managed in the environment should pass through the expression frame, which is the place where both the design and development frames are created and managed; therefore these two frames must have a close but decoupled bond in order to avoid stiffness in the application. The mediator pattern has the precise features to model a solution for this scheme. This pattern defines an object that encapsulates the way a set of objects interact. The pattern promotes a coupling loss by concentrating the references and allowing independent interaction.

b) *Observer*: The extension frames that are created share a common characteristic, namely they display information about the expression frame where they were created; following these ideas, extension frame observe their own behavior continuously in order to update the perspective they exhibit. The observer pattern defines a one-to-many dependence between objects so that whenever an object changes its state, all its dependencies are notified and automatically updated.

2) *VRAPS Patterns*: VRAPS patterns are a set of patterns gathered according to five principles, namely Vision, Rhythm, Anticipation, Partnering and Simplification [29].

a) *Vision*: Vision is the mapping of a future value to architectural constraints as a realization measurement of architecture structure and objectives, which should be clear congruent and flexible. Out of the patterns proposed by the Vision principle, for the case of AEE architecture, the chosen pattern was *Generative Vision*, since the architecture was developed to fulfill the requirements of a simple and robust

architecture. This requirements phase was conducted with users of platforms such as *netbeans* [30] and *eclipse* [31], which are robust but yet complex.

b) *Rhythm*: Rhythm is the recurrent and predictable change of work products with respect to an architectural product, through clients and providers. Out of the patterns proposed by the Rhythm principle for the case of AEE architecture, the pattern chosen was *Drop Pass*. This pattern is evident in terms of releases. These releases have kept pace with development, where the most complex functionalities have been solved during the evolution of the environment, without the architecture being affected thanks to the separation of functions on a component-basis. The strategy developed to cope with the development of the architecture is a gradual increase in the implementation, considering a set of minimal functions per release.

c) *Anticipation*: Anticipation allows the people who build and implement the architecture to predict, validate and adapt the architecture according to the technological changes, competences, and necessities of users. Out of the patterns proposed by the anticipation principle, for the case of AEE architecture, the chosen pattern was *Architecture Review*. This pattern is evident in the permanent reviews of the AEE architecture. This review has followed the evolution of both the java implementation language and the SWT framework. Besides updating the different versions, such monitoring has been useful to correct the mistakes found in previous versions due to the evolution of SWT and the appearance of java version 5.0.

d) *Partnering*: Partnering clearly defines cooperative roles. Out of the patterns proposed by the partnering principle, for the case of AEE architecture, the pattern chosen was *Know the Stakeholders* because of the importance associated to users who have used Beta versions of the product and whose opinions have been highly valued in order to improve AEE.

e) *Simplification*: Simplification means intelligent minimization and clarification of both the architecture and the organizational environment where operation occurs. Out of the patterns proposed by the simplification principle, for the case of AEE architecture, the pattern chosen was *Slow down to Speed up*. This pattern allowed acceleration in the architectural development, especially when considering that this pattern required a significant period of time for reflection on how to build a simple, yet robust structure.

IV. RESULTS

TABLE I: Coloso Platform

Frameworks	plugins	components
Engineering:		
- AEE	- UML diagrams	- GoF patterns
- uml	- archimate viewpoint	- profiles
- archimate	- java compiler	- metrics
- java	- java debugger	- process template
- process	- process viewpoint	- appearance
Artistic:		
- appearance		- icons

AEE architecture represent the structure of the Coloso platform [32], where a set of frameworks, plugins and software

design components have been implemented; the artistic approach of the platform, supported by its visual layout (colors) and its icons is provided with a framework and a component that can be customized and/or replaced according to the user's needs Table I.

The Coloso platform has been used for academic and entrepreneurial purposes, Fig. 11.

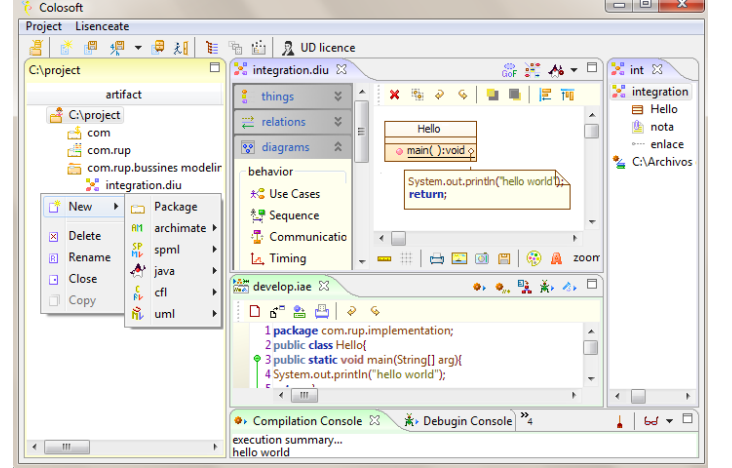


Fig. 11: Coloso Version.

some findings of its use include the following:

- The environment is easy-to-use once its philosophy of use has been explained; the first impression made on most useful scenarios tends to underestimate its functionalities due to the proposed simplicity-based design.
- The implemented languages offer robust support when modeling, and although non-supported mechanisms have been detected, these have been an important input to refine the modules.
- The difference of using the Coloso platform, compared to other environments, was more noticeable when identifying significant contributions of the components associated to metrics, traceability, patterns, profiles, and mostly the development process-and-methodology templates, which make of Coloso a PSEE platform that integrates components and plugins into a single conceptual unit.
- Still unexpected bugs are found, and the repair protocol, triggered from the tool, has allowed gradually debugging and stabilizing the platform.
- The main weakness, explicit from its use, was identified in the lack of modules that serve to integrate other implementation languages like c++, php and sql.

Fig. 12 highlights the most important features of the environment that was developed on the proposed architecture.

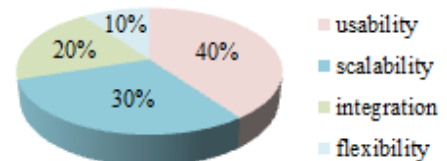


Fig. 12: AEE Main Features.

V. CONCLUSIONS

The motivation to develop AEE was to obtain a simple, yet robust and independent architecture to integrate components intended to support software development. This proposal is aimed at developers who wish to integrate and use components to support software development. The combination of engineering concepts and art concepts strengthens software development due to the usability conditions that can be obtained from such integration.

Open platforms with plugin support, such as eclipse and netbeans, have permitted extensive use of modules that enrich software development tasks. However, this type of platforms is instable when running plugins of different versions; this is caused by the change in their basic structures. AEE keeps a stable architectural concept, which is scalable throughout versions without having to sacrifice its fundamental structure.

VI. FUTURE WORK

As future work, we propose to open this platform with the purpose of implementing more components in an open-source format through the proposed facility-and-use interface. The environment will soon be available at www.colosoft.com.co

ACKNOWLEDGMENT

The authors would like to thank the MIS (Modelamiento en Ingeniera de Sistemas) research group at Universidad Distrital, and also to students Arturo Arismendi and Edwin Castaeda for their supportive job.

REFERENCES

- [1] J. C. Derniame, B. A. Kaba, and D. Wastell, *Software process: principles, methodology, and technology*. Berlin Heidelberg: Springer-Verlag, 1999.
- [2] I. Sommerville, *Ingeniera del Software*. Addison-Wesley, 2005.
- [3] N. Tractinsky, A. S. Katz, and D. Ikar, "What is beautiful is usable," *Interacting with Computers*, vol. 13, no. 2, pp. 127–145, 2000.
- [4] W. J. Brown, R. C. Malveau, W. H. "Skip" McCormick, S. W. Thomas, and T. H. (ed), *Anti-Patterns in Project Management*. John Wiley & Sons, 2000.
- [5] P. A. Laplante and J. N. Colin, *Antipatterns: Identification, Refactoring and Management*. Auerbach Publications, 2005.
- [6] J. McCall, P. Richards, and G. Walters, "Factors in software quality," in *NTIS AD-A049-014*, vol. 3, 1997.
- [7] J. O'Connor and I. McDermott, *Introduccion al pensamiento Sistemico*. Ediciones Urano, 1998.
- [8] J. T. Trevors and M. H. Saier, "Occam's razor and human society," *Water, Air, & Soil Pollution*, vol. 205, no. 1, pp. 69–70, 2010.
- [9] J. Maeda, *Las leyes de la simplicidad*. Gedisa, 2008.
- [10] G. Lakoff and M. Jonhson, *Metaforas de la Vida Cotidiana*. Catedra, 2004.
- [11] M. Overdijk and W. van Diggelen, "Appropriation of a shared workspace: Organizing principles and their application," *International Journal of Computer-Supported Collaborative Learning*, vol. 3, no. 2, pp. 165–192, 2008.
- [12] K. Karvonen, "The beauty of simplicity," in *Proceedings on the 2000 conference on Universal Usability*, Arlington, Virginia, United States, ACM New York, NY, USA, 2000, pp. 85–90.
- [13] U. K. Yusof, L. K. Khaw, H. Yang, and B. J. Neow, "Balancing between usability and aesthetics of web design," in *Information Technology (ITSim), 2010 International Symposium in*, 2010, pp. 1–6.
- [14] E. Heller, *Psicologa del color*. Gustavo Gili, 2004.
- [15] H. Cohen, "Style as emergence (from what?)," *The Structure of Style, part I*, pp. 3–20, 2010.
- [16] L. Bass, P. Clements, and R. Kasman, *Software Architecture in Practice*. Addison-Wesley, 1998.
- [17] M. Shaw and D. Garlan, *Software Architecture*. Prentice Hall, 1996.
- [18] R. Pressman, *Ingeniera del Software Un Enfoque Practico, Quinta Edicion*. Mc Graw Hill, 2002.
- [19] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language User Guide, second edition*. Addison-Wesley, 2005.
- [20] —, *The Unified Modeling Language Reference Manual, second edition*. Addison-Wesley, 2006.
- [21] B. Meyer, *Construccin de Software Orientado a Objetos, Segunda Edicion*. Prentice Hall, 1999.
- [22] J. Cheesman and J. Daniels, *UML Components*. Addison-Wesley, 2001.
- [23] T. Budd, *Introduccion a la Programacion Orientda a Objetos*. Addison-Wesley, 1994.
- [24] C. Horsmann and G. Cornell, *Core Java Volumen II- Advanced Features*. Sun Microsystem Press A Prentice Hall Title, 2000.
- [25] I. Horton, *Beginning Java 2*. Wrox, 2000.
- [26] E. Gamma, R. Helm, R. Jonson, and J. Vlissides, *Design Patterns Elements of Reusable Object-Orienyed Software*. Addison-Wesley, 1995.
- [27] F. Buschmann and et al., *Pattern-Oriented Software Architecture a System of Pattern*. Wiley, 1996.
- [28] P. Kuchana, *Software Architecture Design Patterns in Java*. Auerbach, 2004.
- [29] D. Dikel, D. Kane, and J. Wilson, *Software Architecture Organizational Principles and Patterns*. Prentice Hall, 2001.
- [30] (2012) Netbeans. [Online]. Available: <http://www.netbeans.org/>
- [31] (2012) Eclipse communit. [Online]. Available: <http://www.eclipse.org/>
- [32] (2012) Coloso community. [Online]. Available: <http://www.colosoft.com.co/>

Índice alfabético

Apple, 117	Kumquat, 117
Apricot, 117	Legume, 117
Avocado, 117	Lemon, 117
Banana, 117	Lime, 117
Bilberry, 117	Lychee, 117
Blackberry, 117	Mandarine, 117
Blackcurrant, 117	Mango, 117
Blueberry, 117	Melon, 117
Cherimoya, 117	Nectarine, 117
Cherry, 117	Orange, 117
Clementine, 117	Peach, 117
Currant, 117	Pear, 117
Damson, 117	Physalis, 117
Daniel, 117	Pineapple, 117
Date, 117	Pitaya, 117
Dragonfruit, 117	Plum, 117
Durian, 117	Pomegranate, 117
Eggplant, 117	Purple Mangosteen, 117
Elderberry, 117	Raisin, 117
Feijoa, 117	Rambutan, 117
Gooseberry, 117	Raspberry, 117
Grape, 117	Redcurrant, 117
Grapefruit, 117	Salal berry, 117
Guava, 117	Satsuma, 117
Huckleberry, 117	Star fruit, 117
Jackfruit, 117	Strawberry, 117
Jambul, 117	Tangerine, 117
Kiwi fruit, 117	Tomato, 117

Ugli fruit, 117

Watermelon, 117

Ziziphus mauritiana, 117