

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE INGENIERÍA  
ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO(A) DE SISTEMAS Y COMPUTACIÓN

“ANÁLISIS, DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA  
EL SEGUIMIENTO DEL DESARROLLO DE HABILIDADES DE PERSONAS CON  
DISCAPACIDAD MENTAL”

JUAN CARLOS FERNÁNDEZ ORTIZ  
ZOILA ESTEFANÍA NARVÁEZ BASTIDAS

DIRECTOR: ING. FABIÁN DE LA CRUZ MSC.

QUITO, 2014

## **1. Aspectos Preliminares**

### **1.1. Dedicatoria**

El presente trabajo está dedicado a nuestros padres que con amor, paciencia, comprensión y esfuerzo nos han apoyado siempre, y de manera especial en los momentos difíciles; además de motivarnos constantemente en la consecución de nuestras metas. Esta dedicatoria va también dirigida a nuestros hermanos y familiares, quienes siempre nos han acompañado y han compartido nuestros triunfos y derrotas.

### **1.2. Agradecimientos**

Queremos extender nuestro agradecimiento a la Fundación para la Integración del Niño Especial, particularmente a su directora la Ms. Yolanda Ortiz, por su oportuno y certero aporte en las fases de requerimientos durante el desarrollo del caso de estudio que se encuentra en el presente trabajo.

Adicionalmente queremos agradecer a nuestra alma mater, la Pontificia Universidad Católica del Ecuador, donde nos han impartido los conocimientos necesarios para desarrollarnos como profesionales, así como los valores Ignacianos que rigen a la universidad.

A nuestros estimados profesores que han guiado nuestra vida por la senda del saber, por su paciencia y apoyo que se ven reflejados en la culminación de este trabajo; en especial a nuestro director el Ing. Fabián de la Cruz, quien con sus conocimientos y experiencia ha sido un aporte crucial en el campo de la Ingeniería de Software.

### 1.3. Tabla de Contenidos

1. Aspectos Preliminares .....	i
1.1. Dedicatoria.....	i
1.2. Agradecimientos .....	i
1.3. Tabla de Contenidos .....	ii
1.4. Tabla de Ilustraciones .....	x
1.5. Tabla de Tablas.....	xiv
2. Introducción .....	1
2.1. Datos de la Organización.....	1
2.1.1. Misión.....	1
2.1.2. Visión.....	1
2.1.3. Valores.....	2
2.1.4. Situación Actual .....	2
2.1.5. Fortalezas.....	3
2.1.6. Oportunidades.....	3
2.1.7. Debilidades .....	4
2.1.8. Amenazas.....	4
2.1.9. Cadena de Valor .....	4
2.2. Justificación .....	7
2.3. Alcance .....	7
3. Fundamento Teórico .....	8
3.1. El Aprendizaje en las Personas con Discapacidad Mental .....	8
3.1.1. Características del Aprendizaje de Personas con Discapacidad Mental.....	9
3.1.2. Capacitación Ocupacional .....	10

3.1.2.1.	Talleres Ocupacionales .....	11
3.2.	Programa Acción Educativa Síndrome de Down y Guías de Habilidades Sociales... 11	11
3.3.	Inserción Laboral .....	12
3.3.1.	Campo Laboral .....	13
3.3.2.	La Inserción Laboral en el Ecuador.....	14
4.	Metodología de Desarrollo.....	16
4.1.	TSPi.....	16
4.1.1.	Principios de TSPi .....	16
4.1.2.	La Lógica de TSPi .....	17
4.1.3.	Roles de TSPi .....	19
4.1.3.1.	Líder .....	20
4.1.3.1.1.	Objetivos del Líder .....	21
4.1.3.1.2.	Actividades Principales del Líder.....	21
4.1.3.2.	Administrador de Desarrollo.....	21
4.1.3.2.1.	Objetivos del Administrador de Desarrollo .....	22
4.1.3.2.2.	Actividades Principales del Administrador de Desarrollo .....	22
4.1.3.3.	Administrador de Planificación.....	23
4.1.3.3.1.	Objetivos del Administrador de Planificación .....	23
4.1.3.3.2.	Actividades Principales del Administrador de Planificación .....	23
4.1.3.4.	Administrador de Calidad .....	24
4.1.3.4.1.	Objetivos del Administrador de Calidad .....	24
4.1.3.4.2.	Actividades Principales del Administrador de Calidad .....	25
4.1.3.5.	Administrador de Configuración.....	26
4.1.3.5.1.	Objetivos del Administrador de Configuración.....	26

4.1.3.5.2.	Actividades Principales del Administrador de Configuración .....	26
4.1.4.	Proceso de TSPi.....	27
4.1.4.1.	Lanzamiento.....	29
4.1.4.1.1.	Objetivos del Equipo .....	29
4.1.4.1.2.	Objetivos de los Miembros y Roles del Equipo .....	30
4.1.4.2.	Estrategia de Desarrollo .....	30
4.1.4.3.	Planificación.....	31
4.1.4.3.1.	Niveles de Estimación .....	31
4.1.4.3.2.	Proceso de Planificación.....	33
4.1.4.4.	Definición de Requerimientos.....	35
4.1.4.4.1.	Cambios en los Requerimientos .....	35
4.1.4.5.	Diseño.....	39
4.1.4.6.	Implementación.....	40
4.1.4.7.	Integración y Pruebas .....	41
4.1.4.7.1.	Documentación.....	42
4.1.4.8.	Postmortem.....	43
5.	Tecnologías de Desarrollo.....	44
5.1.	Java .....	44
5.1.1.	Breve Historia.....	44
5.1.2.	Principales características.....	44
5.1.3.	Enterprise Java Beans.....	46
5.2.	MySQL .....	48
6.	Desarrollo del Sistema .....	49
6.1.	Primer Ciclo.....	49

6.1.1.	Lanzamiento .....	49
6.1.1.1.	Asignación de Roles.....	49
6.1.1.2.	Objetivos del Equipo.....	50
6.1.1.3.	Objetivos de Miembro.....	50
6.1.1.4.	Objetivos del Rol.....	51
6.1.1.5.	Objetivos del Producto .....	53
	Antecedentes .....	53
	Funcionalidades .....	53
6.1.1.6.	Objetivos de Calidad.....	54
6.1.2.	Estrategia de Desarrollo.....	54
6.1.2.1.	Criterios de Estrategia .....	54
6.1.2.2.	Estrategia de Reuso .....	55
6.1.3.	Planificación.....	56
6.1.3.1.	Distribución de Módulos.....	56
6.1.4.	Definición de Requerimientos .....	57
	Especificación de Requerimientos del Sistema (SRS) .....	57
	I Introducción.....	57
	II Requerimientos Funcionales .....	60
	III Requerimientos no Funcionales .....	76
	IV Otros Requerimientos .....	77
	V Plan de Pruebas del Sistema.....	79
6.1.5.	Diseño.....	83
	Especificación de Diseño del Sistema (SDS).....	83
	1. Diseño Arquitectónico.....	83

2.	Diagramas de Clases.....	85
3.	Diagramas de Secuencia.....	88
4.	Diagramas de E/R.....	96
5.	Plan de Pruebas de Integración (PPI) .....	97
6.1.6.	Implementación .....	102
6.1.7.	Integración y Pruebas .....	116
6.1.7.1.	Pruebas de Integración .....	116
6.1.7.2.	Pruebas del Sistema.....	117
6.1.8.	Postmortem.....	118
6.1.8.1.	Análisis de Aciertos .....	118
6.1.8.2.	Análisis de Fallos .....	119
6.1.8.3.	Formulario PEER Consolidado.....	120
	Análisis .....	121
6.1.8.4.	Análisis del ciclo.....	121
6.2.	Segundo Ciclo .....	123
6.2.1.	Lanzamiento .....	123
6.2.1.1.	Asignación de Roles.....	123
6.2.1.2.	Objetivos del Equipo.....	123
6.2.1.3.	Objetivos de Miembro.....	124
6.2.1.4.	Objetivos del Rol.....	124
6.2.1.5.	Objetivos de Calidad.....	126
6.2.2.	Estrategia de Desarrollo.....	126
6.2.2.1.	Criterios de Estrategia .....	126
6.2.2.2.	Estrategia de Reuso .....	127

6.2.3.	Planificación.....	128
6.2.3.1.	Distribución de Módulos.....	128
6.2.4.	Definición de Requerimientos.....	129
	Especificación de Requerimientos del Sistema (SRS) .....	129
	I Introducción.....	129
	II Requerimientos Funcionales .....	131
	III Requerimientos no Funcionales.....	148
	IV Otros Requerimientos .....	149
	V Plan de Pruebas del Sistema.....	152
6.2.5.	Diseño.....	157
	Especificación de Diseño del Sistema (SDS) .....	157
	1. Diseño Arquitectónico.....	157
	2. Diagramas de Clases.....	159
	3. Diagramas de Secuencia.....	162
	4. Diagramas de E/R.....	171
	5. Diagramas de Actividades: Proceso de Evaluación .....	172
	6. Plan de Pruebas de Integración (PPI) .....	173
6.2.6.	Implementación .....	176
6.2.7.	Integración y Pruebas .....	192
6.2.7.1.	Pruebas de Integración .....	192
6.2.7.2.	Pruebas del Sistema.....	193
6.2.8.	Postmortem.....	193
6.2.8.1.	Análisis de Aciertos .....	194
6.2.8.2.	Análisis de Fallos .....	194

6.2.8.3.	Formulario PEER Consolidado.....	194
Análisis .....		195
6.2.8.4.	Análisis del ciclo.....	196
7.	Conclusiones .....	198
8.	Recomendaciones.....	200
9.	Bibliografía.....	201
10.	Anexos.....	204
10.1.	Primer Ciclo .....	204
10.1.1.	Formas Fases de Lanzamiento, Estrategia y Planificación.....	204
10.1.2.	Casos de Uso a Detalle .....	204
10.1.3.	Plan de Pruebas del Sistema .....	204
10.1.4.	Formas Fase de Requerimientos.....	204
10.1.5.	Diagramas de Secuencia.....	205
10.1.6.	Formas Fase de Diseño.....	205
10.1.7.	Código Fuente.....	205
10.1.8.	Formas Fase de Implementación .....	205
10.1.9.	Formas Fase de Pruebas y Postmortem .....	205
10.2.	Segundo Ciclo .....	205
10.2.1.	Formas Fase de Lanzamiento, Estrategia y Planificación .....	205
10.2.2.	Casos de Uso a Detalle .....	206
10.2.3.	Plan de Pruebas del Sistema .....	206
10.2.4.	Diagramas de Secuencia.....	206
10.2.5.	Formas Fase de Requerimientos y Diseño .....	206
10.2.6.	Código Fuente.....	206

10.2.7. Formas Fase de Implementación .....207

10.2.8. Fase de Pruebas y Postmortem .....207

10.2.9. Manual de Usuario.....207

10.2.10. Manual de Instalación.....207

## 1.4. Tabla de Ilustraciones

Ilustración 1: Macroprocesos y Procesos de FINE.....	5
Ilustración 2: Subprocesos “Capacitar a personas con discapacidad” .....	6
Ilustración 3: Estructura y Flujo de TSPi .....	28
Ilustración 4: Partes y Colecciones.....	32
Ilustración 5: Proceso Planificación .....	34
Ilustración 6: Requerimientos Especiales.....	37
Ilustración 7: Arquitectura Neutral de Java.....	46
Ilustración 8: Arquitectura EJB .....	47
Ilustración 9: Diagrama de Casos de Uso General.....	60
Figura 10: Diagrama de Casos de Uso Siguiete Nivel de F0 .....	61
Ilustración 11: Diagrama de Casos de Uso Siguiete Nivel de F1 .....	61
Ilustración 12: Diagrama de Casos de Uso Siguiete Nivel de F2.....	62
Ilustración 13: Diagrama de Casos de Uso Siguiete Nivel de F3 .....	62
Ilustración 14: Diagrama de Casos de Uso Siguiete Nivel de F4.....	63
Ilustración 15: Diagrama de Casos de Uso Siguiete Nivel de F5.....	63
Ilustración 16: Diagrama de Casos de Uso Siguiete Nivel de F6.....	64
Ilustración 17: Diagrama de Casos de Uso Siguiete Nivel de F7 .....	64
Ilustración 18: Diagrama de Casos de Uso Siguiete Nivel de F8.....	65
Ilustración 19: Diagrama de Casos de Uso a Detalle de F3.1 .....	65
Ilustración 20: Diagrama de Casos de Uso a Detalle de F3.2 .....	67
Ilustración 21: Diagrama de Casos de Uso a Detalle de F3.3 .....	68
Ilustración 22: Diagrama de Casos de Uso a Detalle de F3.4 .....	69
Ilustración 23: Diagrama de Casos de Uso a Detalle de F8.1 .....	70

Ilustración 24: Diagrama de Casos de Uso a Detalle de F8.2 .....	72
Ilustración 25: Diagrama de Casos de Uso a Detalle de F8.3 .....	73
Ilustración 26: Diagrama de Casos de Uso a Detalle de F8.4 .....	74
Ilustración 27: Diagrama de clases conceptual Managed Beans .....	75
Ilustración 28: Diagrama de clases conceptual Session Beans.....	76
Ilustración 29: Diagrama de clases conceptual Entities .....	76
Ilustración 30: Página Principal.....	77
Ilustración 31: Administración de Perfiles .....	77
Ilustración 32: Administración de Usuarios .....	78
Ilustración 33: Diagrama de paquetes .....	84
Ilustración 34: Diagrama de Clases de Managed Beans a Detalle .....	85
Ilustración 35: Diagrama de Clases de Session Beans a Detalle .....	86
Ilustración 36: Diagrama de Clases de Entities a Detalle.....	87
Ilustración 37: Diagrama de Secuencia F3.1 .....	88
Ilustración 38: Diagrama de Secuencia F3.2 .....	89
Ilustración 39: Diagrama de Secuencia F3.3 .....	90
Ilustración 40: Diagrama de Secuencia F3.4 .....	91
Ilustración 41: Diagrama de Secuencia F8.1 .....	92
Ilustración 42: Diagrama de Secuencia F8.2 .....	93
Ilustración 43: Diagrama de Secuencia F8.3 .....	94
Ilustración 44: Diagrama de Secuencia F8.4.....	95
Ilustración 45: Diagrama Entidad Relación.....	96
Ilustración 46: Diagrama de dependencia de clases .....	98
Ilustración 47: Diagrama de dependencia de clases Grupo 1 .....	99

Ilustración 48: Diagrama de dependencia de clases Grupo 2 .....	100
Ilustración 49: Diagrama de dependencia de clases Grupo 3 .....	101
Ilustración 50: Diagrama de Casos de Uso General .....	131
Ilustración 51: Diagrama de Casos de Uso Siguiete Nivel de F9.....	132
Ilustración 52: Diagrama de Casos de Uso Siguiete Nivel de F10.....	132
Ilustración 53: Diagrama de Casos de Uso Siguiete Nivel de F11 .....	133
Ilustración 54: Diagrama de Casos de Uso Siguiete Nivel de F12.....	133
Ilustración 55: Diagrama de Caso de Uso a Detalle de F10.1 .....	134
Ilustración 56: Diagrama de Casos de Uso a Detalle de F10.2 .....	135
Ilustración 57: Diagrama de Casos de Uso a Detalle de F10.3 .....	137
Ilustración 58: Diagrama de Casos de Uso a Detalle de F10.4 .....	138
Ilustración 59: Diagrama de Caso de Uso a Detalle de F11.1 .....	139
Ilustración 60: Diagrama de Casos de Uso a Detalle de F11.2 .....	141
Ilustración 61: Diagrama de Casos de Uso a Detalle de F11.3 .....	143
Ilustración 62: Diagrama de Casos de Uso a Detalle de F11.4 .....	144
Ilustración 63: Diagrama de Casos de Uso a Detalle de F11.5 .....	146
Ilustración 64: Diagrama de clases conceptual Managed Beans .....	147
Ilustración 65: Diagrama de clases conceptual Session Beans.....	148
Ilustración 66: Diagrama de clases conceptual Entities .....	148
Ilustración 67: Administracion de Asignación de Profesoras (res) a Talleres .....	149
Ilustración 68: Administración de Evaluaciones Iniciales .....	150
Ilustración 69: Administración de Evaluaciones Mensuales .....	151
Ilustración 70: Reporte Quimestral.....	151
Ilustración 71: Diagrama de paquetes .....	158

Ilustración 72: Diagrama de Clases de Managed Beans a Detalle .....	159
Ilustración 73: Diagrama de Clases de Session Beans a Detalle .....	160
Ilustración 74: Diagrama de Clases de Entities a Detalle.....	161
Ilustración 75: Diagrama de Secuencia F10.1 .....	162
Ilustración 76: Diagrama de Secuencia F10.2 .....	163
Ilustración 77: Diagrama de Secuencia F10.3 .....	164
Ilustración 78: Diagrama de Secuencia F10.4 .....	165
Ilustración 79: Diagrama de Secuencia F11.1 .....	166
Ilustración 80: Diagrama de Secuencia F11.2 .....	167
Ilustración 81: Diagrama de Secuencia F11.3 .....	168
Ilustración 82: Diagrama de Secuencia F11.4 .....	169
Ilustración 83: Diagrama de Secuencia F11.5 .....	170
Ilustración 84: Diagrama Entidad Relación.....	171
Ilustración 85: Diagrama de Actividades .....	172
Ilustración 86: Diagrama de dependencia de clases .....	174
Ilustración 87: Diagrama de dependencia de clases Grupo 1 .....	175
Ilustración 88: Diagrama de dependencia de clases Grupo 2 .....	175

## 1.5. Tabla de Tablas

Tabla 1: Asignación de roles. Ciclo I .....	49
Tabla 2: Objetivos de Calidad del Producto .....	54
Tabla 3: Distribución de Funcionalidades por Ciclo .....	55
Tabla 4: Distribución de Módulos. Ciclo I .....	57
Tabla 5: Información del Equipo. Ciclo I .....	59
Tabla 6: Flujo Principal de F3.1 .....	66
Tabla 7: Flujo Alternativo de F3.1 .....	66
Tabla 8: Excepciones de F3.1 .....	66
Tabla 9: Flujo Principal de F3.2 .....	67
Tabla 10: Flujo Alternativo de F3.2 .....	67
Tabla 11: Excepciones de F3.2 .....	68
Tabla 12: Flujo Principal de F3.3 .....	68
Tabla 13: Flujo Alternativo de F3.3 .....	69
Tabla 14: Excepciones de F3.3 .....	69
Tabla 15: Flujo Principal de F3.4 .....	70
Tabla 16: Excepciones de F3.4 .....	70
Tabla 17: Flujo Principal de F8.1 .....	71
Tabla 18: Flujo Alternativo de F8.1 .....	71
Tabla 19: Excepciones de F8.1 .....	71
Tabla 20: Flujo Principal de F8.2 .....	72
Tabla 21: Flujo Alternativo de F8.2 .....	72
Tabla 22: Excepciones de F8.2 .....	73
Tabla 23: Flujo Principal de F8.3 .....	73

Tabla 24: Flujo Alterno de F8.3 .....	74
Tabla 25: Excepciones de F8.3 .....	74
Tabla 26: Flujo Principal de F8.4 .....	75
Tabla 27: Excepciones de F8.4 .....	75
Tabla 28: Pruebas de Integración. Ciclo I .....	116
Tabla 29: Pruebas de Sistema. Ciclo I.....	118
Tabla 30: Análisis del Ciclo I.....	121
Tabla 31: Asignación de Roles. Ciclo II.....	123
Tabla 32: Objetivos de Calidad del Producto.....	126
Tabla 33: Funciones desarrolladas durante el Ciclo I.....	127
Tabla 34: Funciones Planificadas para el Ciclo II .....	127
Tabla 35: Distribución de Módulos para el Ciclo II.....	129
Tabla 36: Información del Equipo. Ciclo II. ....	130
Tabla 37: Flujo Principal de F10.1 .....	135
Tabla 38: Flujo Alterno de F10.1 .....	135
Tabla 39: Excepciones de F10.1 .....	135
Tabla 40: Flujo Principal de F10.2 .....	136
Tabla 41: Flujo Alterno de F10.2 .....	136
Tabla 42: Excepciones de F10.2.....	136
Tabla 43: Flujo Principal de F10.3 .....	137
Tabla 44: Flujo Alterno de F10.3 .....	137
Tabla 45: Excepciones de F10.3.....	138
Tabla 46: Flujo Principal de F10.4.....	138
Tabla 47: Excepciones de F10.4.....	139

Tabla 48: Flujo Principal de F11.1 .....	140
Tabla 49: Flujo Alterno de F11.1 .....	140
Tabla 50: Excepciones de F11.1 .....	140
Tabla 51: Flujo Principal de F11.2 .....	142
Tabla 52: Flujo Alterno de F11.2 .....	142
Tabla 53: Excepciones de F11.2 .....	142
Tabla 54: Flujo Principal de F11.3 .....	143
Tabla 55: Flujo Alterno de F11.3 .....	144
Tabla 56: Excepciones de F11.3 .....	144
Tabla 57L Flujo Principal de F11.4 .....	145
Tabla 58: Flujo Alterno de F11.4 .....	145
Tabla 59: Excepciones de F11.4 .....	146
Tabla 60: Flujo Principal F11.5 .....	147
Tabla 61: Flujo Alterno de F11.5 .....	147
Tabla 62: Excepciones de F11.5 .....	147
Tabla 63: Pruebas de Integración. Ciclo II .....	192
Tabla 64: Pruebas de Sistema. Ciclo II .....	193
Tabla 65: Análisis del Ciclo. Ciclo II .....	196

## **2. Introducción**

El presente capítulo contiene información sobre la organización incluyendo el proceso específico, que forma parte de la cadena de valor de la organización, alrededor del cual se ha planteado este trabajo de disertación. Además contiene la justificación y el alcance que limitará este trabajo.

### **2.1. Datos de la Organización**

#### **2.1.1. Misión**

*“FINE es una organización sin fines de lucro, constituida por padres y profesionales, que brinda capacitación integral a personas con discapacidad intelectual y múltiple, mejorando su calidad de vida y permitiéndoles incluirse en su familia y la sociedad. La capacitación se enfoca en: destrezas sociales, motrices básicas, autonomía personal, capacitación y acompañamiento laboral.”<sup>1</sup>*

#### **2.1.2. Visión**

*“En cinco años, FINE liderará los procesos de inserción laboral protegida de personas con discapacidad múltiple y se convertirá en un referente nacional. Mejoraremos y estandarizaremos los procesos relacionados con la rehabilitación y el modelo formativo de capacitación mediante el desarrollo de metodología y mallas curriculares para posicionar la implementación de talleres protegidos en otras organizaciones a nivel nacional.*

*Sensibilizaremos y capacitaremos a la población en general, diseñando charlas y planificando estrategias que nos permitan llegar al mayor número de personas. Consolidaremos la socialización de las personas con discapacidad con sus pares,*

---

<sup>1</sup> Fundación para la Integración del Niño Especial. F.I.N.E (2013, mayo 25). Planificación Estratégica. Quito, Pichincha, Ecuador.

*atendiendo sus necesidades afectivas y emocionales, reforzando su propia autoestima y proyectándolos a una vida familiar y social equilibrada.”<sup>2</sup>*

### **2.1.3. Valores**

Los valores en los que se fundamenta la organización son:

- ✓ Mística
- ✓ Altruismo
- ✓ Compromiso
- ✓ Honestidad
- ✓ Profesionalismo

### **2.1.4. Situación Actual**

La Fundación para la Integración del Niño Especial, realiza una evaluación de cada una de las personas con discapacidad que llegan a su centro. Posteriormente elabora un programa individual de trabajo que recoge un inventario de habilidades sociales, motrices, cognitivas y de autonomía personal que son la base para su plan de trabajo, así también evidencia sus limitaciones.

Para elaborar los programas individuales de trabajo que se desarrollan, ejecutan y evalúan de manera quimestral, las educadoras cuentan con una matriz de habilidades tanto sociales, de autonomía personal, cognitivas y destrezas motrices; a partir de las cuales priorizan las habilidades que se pueden trabajar en una persona.

A continuación se determina si esta persona, con un direccionamiento, podría incluirse en el medio laboral.

Una vez finalizado el quimestre, se entrega un informe a los padres de familia o representantes de las personas con discapacidad, en el cual se resume el trabajo realizado y los logros en cuanto a las habilidades que se planteó desarrollar.

---

<sup>2</sup> Fundación para la Integración del Niño Especial. F.I.N.E (2013, mayo 25). Planificación Estratégica. Quito, Pichincha, Ecuador.

Cuando la persona es apta para ser insertada, se busca una organización donde es incluida laboralmente. Posteriormente se realizan visitas a la empresa con el fin de brindar asesoría a la persona que hace las funciones de jefe inmediato, así como para evaluar las destrezas en el puesto de trabajo de la persona con discapacidad y garantizar su bienestar.

Todo este proceso se lleva a cabo manualmente y posteriormente es pasado a un editor de texto, lo cual genera pérdidas de datos y dificulta el análisis de indicadores en el tiempo. Es imposible mantener una estandarización si la rotación de personal se hace constante, esto también sería superado con este sistema.

#### **2.1.5. Fortalezas**

Las fortalezas identificadas en la organización se detallan a continuación:

- ✓ Instalaciones propias sin barreras arquitectónicas.
- ✓ Metodología.
- ✓ Formalización de procesos contables.
- ✓ Personería jurídica.
- ✓ Experticia (Know How).
- ✓ Personal comprometido y capacitado.

#### **2.1.6. Oportunidades**

Las oportunidades con las que cuenta la organización son las siguientes:

- ✓ Políticas públicas en el campo de la discapacidad.
- ✓ Requerimiento del mercado en capacitación para el tema de discapacidades.
- ✓ Interés de organizaciones internacionales para actuar como consultores de FINE en temas estratégicos.
- ✓ Alianza estratégica con la PUCE.

### **2.1.7. Debilidades**

A continuación se presentan las debilidades que se han detectado en la organización:

- ✓ Existen pocos líderes dentro de la organización, especialmente en el campo técnico.
- ✓ Necesidad de un sistema de evaluación de desempeño por cumplimiento de objetivos.
- ✓ No se ha definido una estrategia financiera para ajustes salariales.
- ✓ No se cuenta con un equipo completo de profesionales.
- ✓ Sueldos bajos.
- ✓ Se debe definir procesos organizacionales.

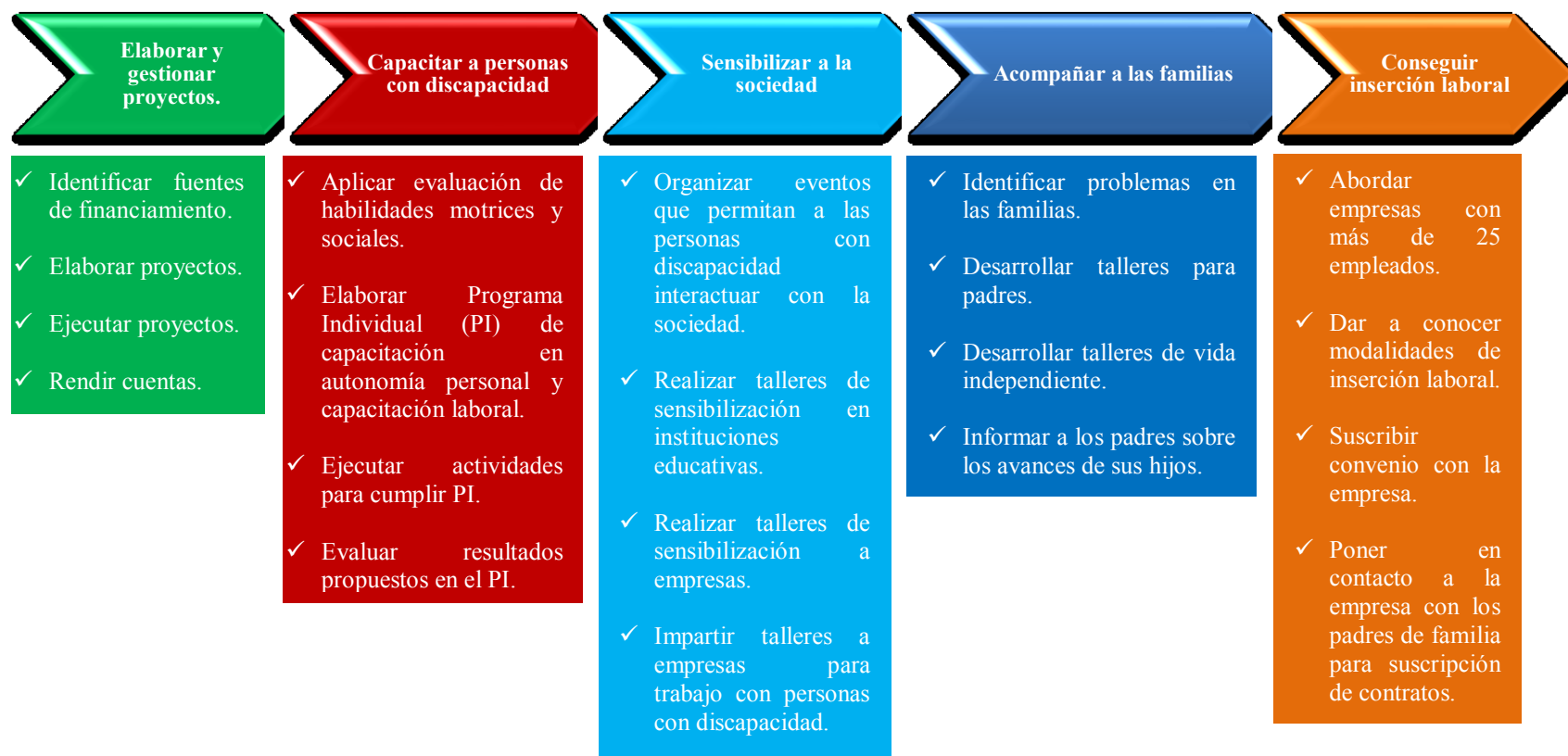
### **2.1.8. Amenazas**

Las amenazas que se han detectado son las siguientes:

- ✓ Políticas públicas no definidas en tema de fundaciones y organizaciones de la sociedad civil en general.
- ✓ Falta de reglamento de talleres protegidos.

### **2.1.9. Cadena de Valor**

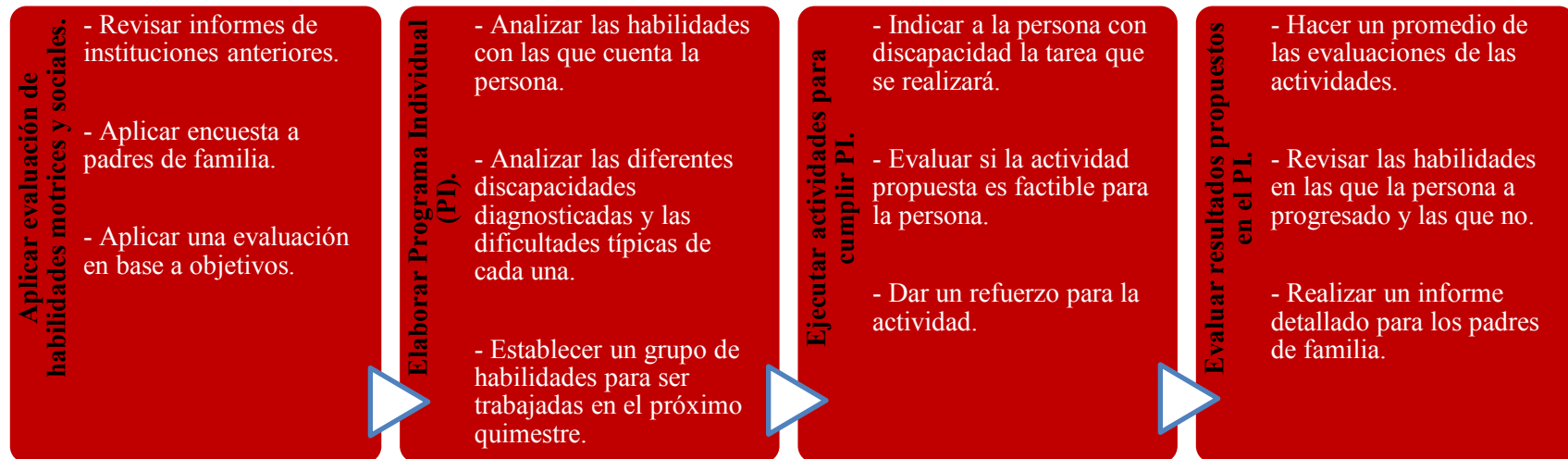
A continuación se muestra la cadena de valor de la Fundación para la Integración del Niño Especial.



**Ilustración 1: Macroprocesos y Procesos de FINE<sup>3</sup>**

<sup>3</sup> Ortiz, Y. (4 de marzo de 2014). Procesos de FINE. (E. Narváez Bastidas, & J. C. Fernández Ortiz, Entrevistadores)

El sistema que se plantea, pretende optimizar el macroproceso “Capacitar a personas con discapacidad”. A continuación se detallan los subprocesos.



**Ilustración 2: Subprocesos “Capacitar a personas con discapacidad”<sup>4</sup>**

<sup>4</sup> Ortiz, Y. (4 de marzo de 2014). Procesos de FINE. (E. Narváez Bastidas, & J. C. Fernández Ortiz, Entrevistadores)

## **2.2. Justificación**

El sistematizar la información, permite guardar la información de manera ordenada y facilita el acceso a la misma, reduciendo el riesgo de pérdida e inconsistencia de los datos.

El contar con un sistema de gestión de programas educativos sería de gran ayuda para la Fundación para la Integración del Niño Especial, ya que les permitiría tener la información organizada, facilitaría el control de resultados en función de los planes individuales propuestos y la revisión de indicadores en el tiempo.

Las personas que logran desarrollar habilidades sociales y motrices básicas, pueden ingresar en un proceso de capacitación para adquirir habilidades y destrezas que garanticen el buen desempeño de las tareas que les son designadas en una empresa o empleo. Las habilidades son desarrolladas mediante talleres de capacitación ocupacional enfocados a un grupo de destrezas como la tolerancia a la jornada de trabajo, que es el hábito a laborar, la convivencia con otros individuos, entre muchas otras que les sirven como herramientas para poder ingresar al mundo laboral.

La Fundación para la Integración del Niño Especial - FINE se especializa en llevar a cabo el proceso de capacitación pre laboral y laboral. Mediante talleres, realiza una prospección de habilidades, inclinaciones y aptitudes concluyendo con el seguimiento, cuando ya se ha efectuado la transición total al puesto de trabajo, en los casos en que es posible una inserción laboral.

## **2.3. Alcance**

El presente trabajo finalizará con la entrega de un prototipo funcional de un sistema de información para el seguimiento del desarrollo de habilidades de personas con discapacidad mental.

### **3. Fundamento Teórico**

Este capítulo contiene información general sobre el proceso de aprendizaje de las personas con discapacidad mental, la capacitación que las mismas reciben y la situación laboral de las personas con discapacidad en el Ecuador.

#### **3.1. El Aprendizaje en las Personas con Discapacidad Mental**

El aprendizaje produce un cambio de conducta en los individuos, que no tiene relación con la maduración, donde el cambio es permanente y el estudiante debe ser una parte activa de dicho proceso.

La deficiencia mental engloba a un grupo muy heterogéneo de situaciones, que tienen como factor común un deterioro general del funcionamiento intelectual y del comportamiento adaptativo, manifestado durante la edad del desarrollo. Por lo que las necesidades educativas especiales serán diferentes para cada persona, e incluso diferentes en cada etapa de la vida de una persona.<sup>5</sup>

La deficiencia mental resulta de las dificultades de adaptación al entorno que experimenta una persona debido a sus limitaciones intelectuales. Éstas pueden tener diversos orígenes, desde anomalías genéticas, traumatismos obstétricos y enfermedades que dañan el sistema nervioso, hasta graves limitaciones de estimulación recibida en períodos críticos del desarrollo.

La potenciación de las áreas de desarrollo menos afectadas y la adecuada estimulación de las que lo están en mayor medida se convierten en tareas educativas básicas, que hacen posible la disminución de las limitaciones que supone el déficit intelectual de la persona afectada, lo que permite atenuar su discapacidad.

---

<sup>5</sup> González Manjón, D. (2006). NEE más frecuentes en la Educación Infantil. Cádiz: Departamento de Psicología - Universidad de Cádiz.

### **3.1.1. Características del Aprendizaje de Personas con Discapacidad Mental**

Es difícil llegar a describir las características de aprendizaje de las personas con discapacidad mental, ya que cuanto más se investiga, se pueden encontrar más diferencias, que permiten clasificarlas en más y más subgrupos; sin embargo existen ciertas características que se pueden apreciar con mayor frecuencia. Estudios de diversos autores, como Verdugo, Canal, Del Barrio, Garrido, Molina, Arraiz y Varn Hagen, convergen en que las personas con discapacidad intelectual presentan dificultades selectivas en la memoria de trabajo a corto plazo y en la memoria semántica a largo plazo. Verdugo las resume de la siguiente manera:<sup>6</sup>

#### ✓ **Problemas en la memoria de trabajo a corto plazo**

Las personas con discapacidad intelectual tienen dificultades en tareas que requieren emplear estrategias activas de recuerdo, así como en el uso espontáneo de mnemotecnias, que les permitan mejorar el almacenamiento de información. También se evidencia una dificultad para generalizar tareas o técnicas de repetición a tareas nuevas.

#### ✓ **Problemas en la memoria a largo plazo**

Los problemas de memoria a largo plazo están condicionados por problemas atencionales, de memoria sensorial y de memoria a corto plazo, que crean dificultades en la asimilación de nueva información. Se evidencian más las dificultades en las personas con discapacidad mental, especialmente cuando se trata a un nivel semántico o significativo.

Las dificultades en los procesos de memoria se derivan en problemas atencionales, así como en los procesos denominados metacognitivos o de autorregulación.

Los problemas atencionales se evidencian en tareas que exigen un alto nivel de procesamiento central, es decir, tener una conciencia de qué se está haciendo, cómo se está haciendo y cómo mejorarlo. En consecuencia, se dan problemas para desatender a estímulos irrelevantes o poco apropiados, así como mantener la atención durante el tiempo necesario para hacer una discriminación correcta de los diferentes estímulos.

---

<sup>6</sup> Verdugo, M.A. (2003). Programa de Habilidades Sociales: Programas Conductuales Alternos. España: Amarú.

Los problemas metacognitivos o de procesos ejecutivos, se relacionan con la capacidad de, a partir del conocimiento de los recursos mentales disponibles, gestionarlos; analizar las tareas, plantear objetivos que deben alcanzarse, seleccionar las estrategias más apropiadas para conseguirlo, supervisar el proceso para ver si es preciso modificar las estrategias elegidas, etc. El control ejecutivo permite que la persona controle su aprendizaje y su pensamiento.

### **3.1.2. Capacitación Ocupacional**

Según el Diccionario Enciclopédico de la Educación Especial, la capacitación ocupacional de la persona con discapacidad, debe integrar no solo los conocimientos del individuo, sino también su entorno familiar, la realidad económica del país, sus deseos, expectativas, intereses y personalidad.

La capacitación laboral se desarrolla en base a distintas fases: el diagnóstico, la orientación y el seguimiento y la evaluación.

El diagnóstico es una parte indispensable del proceso, ya que permiten al orientador conocer más a fondo las características del individuo y seleccionar los programas y acciones que se llevaran a cabo durante la capacitación. Según Mary B. Smith, se pueden utilizar herramientas como pruebas de criterio referencial, observación del individuo en situaciones reales de empleo, historias clínicas y entrevistas.

En base a los resultados obtenidos en el diagnóstico, se realiza una capacitación a la persona recalando las habilidades y destrezas con las que cuenta y motivándole a establecer objetivos y compromisos, que cumplirá progresivamente. Esta actividad se realiza en la mayoría de casos mediante los talleres ocupacionales.

La capacitación ocupacional tiene como objetivo la enseñanza práctica de una ocupación mediante la adquisición de hábitos, conductas y disciplina personal. La edad en la que se inicie la capacitación es un punto crucial en su rehabilitación, mientras más temprana sea la intervención, mejores serán los resultados.

Una vez que se ha culminado con la capacitación laboral, se debe contar con un sistema de seguimiento y evaluación continua que nos permita medir el desarrollo del individuo y sus necesidades, ya sea que haya sido insertado laboralmente o no.

### ***3.1.2.1. Talleres Ocupacionales***

Los talleres ocupacionales brindan entrenamiento a la persona con discapacidad acorde a sus habilidades, destrezas y deseos; dándole la capacidad de desempeñarse en el entorno laboral. Durante el desarrollo del taller se debe hacer una permanente evaluación de los progresos y dificultades que enfrente el individuo.

Los talleres ocupacionales buscan principalmente capacitar a la persona con discapacidad, incrementando sus destrezas y habilidades e incentivando su interés y disposición para el trabajo. Además se enfocan en aumentar su autonomía en la ejecución de tareas, potenciar su autoestima y satisfacción por el trabajo realizado; y favorecer las relaciones interpersonales. Durante cada uno de los talleres se pretende ejercitar al individuo en la realización de tareas correspondientes a un cargo o profesión específica.

Algunos ejemplos de talleres ocupacionales que se pueden llevar a cabo pueden ser: la crianza de animales, cerámica, pintura, servicios de archivo, manualidades, carpintería, panadería, secretariado, mensajería, entre muchos otros; desarrollados de acuerdo a las habilidades e intereses de cada persona con discapacidad.

Los talleres deben ser adaptados para cada uno de los individuos, tomando en cuenta sus fortalezas, necesidades y limitaciones. Además, es indispensable que se tengan claros los objetivos que se desean alcanzar con cada actividad que se realice durante el taller. Estos objetivos guiarán al orientador durante la ejecución de las actividades y permitirán medir el desempeño de la persona con discapacidad, por lo que es importante que los mismos sean simples y precisos.

## **3.2. Programa Acción Educativa Síndrome de Down y Guías de Habilidades Sociales.**

El Programa de Acción Educativa Síndrome de Down, es un estudio longitudinal, realizado por un grupo de investigadores en el Centro Príncipe de Asturias, en las Islas Baleares, España; bajo la dirección del Dr. Juan Perera. El estudio, nace a partir del deseo de un grupo de padres de familia de niños con Síndrome de Down, que querían que sus hijos fueran “tratados”. En el estudio se trabajó con un grupo de 58 niños con Síndrome de Down durante

dos años, posteriormente se aplicó el programa diseñado durante cuatro años, quedando así validado.

Este programa sistemático de acción educativa, nace tras descartar la posibilidad de modificar el cerebro de los pacientes farmacológicamente, quedando hasta la actualidad como modelo de un método terapéutico que aporta un valor importante, tras llevar a cabo una acción educativa tempranamente iniciada y mantenida continuamente.<sup>7</sup>

El análisis y adaptación que FINE ha realizado tanto del Programa de Acción Educativa Síndrome de Down, como de las Guías de Habilidades sociales a la realidad cultural del Ecuador, le ha facilitado el manejo de los planes de desarrollo, diseñados para los diferentes niveles de funcionamiento de los usuarios. Estos sirven como guías para los facilitadores, tutores o capacitadores, pues es importante emprender los nuevos retos de la enseñanza especial, acorde a las características del individuo.

El Programa de Acción Educativa Síndrome de Down al ser una confluencia del análisis de varias guías evolutivas, proporciona los parámetros que sirven para un posterior afianzamiento de las destrezas que se necesitan para instaurar la concentración, tolerancia y hábito de trabajo.

Las habilidades sociales que se desarrollan a lo largo de la vida, de hecho requieren esta base de desarrollo evolutivo. Ellas marcan las pautas para un buen desempeño en la enseñanza de habilidades que pueden ayudar a cualquier ser humano a desarrollar un mejor nivel de convivencia, y una mejor canalización y resolución de problemas.

### **3.3. Inserción Laboral**

La inserción laboral se enfoca en brindar acompañamiento y capacitación a personas con discapacidad que han sido excluidas laboralmente y/o socialmente con el fin de mejorar su calidad de vida y permitirles ingresar al mundo laboral, dándoles continuo seguimiento y capacitando constantemente a los miembros de la organización donde se desarrollarán.

---

<sup>7</sup> Perera Mezquida, J.(1995). Síndrome de Down. Programa de Acción Educativa. Madrid: CEPE S.A.

Los cursos de capacitación a personas con discapacidad deben ser realizados en base a los niveles de competencia de cada individuo, tratando de trabajar en tareas que impliquen procesos cognitivos; así como, manipulativo. El trabajo en equipo es muy importante, debido a que aporta en el desarrollo habilidades comunicativas. Además cada curso debe tener la propiedad de adaptarse en el tiempo según las necesidades de cada persona.

### **3.3.1. Campo Laboral**

Las personas con discapacidad que van a ser incluidas laboralmente serán parte del mismo mercado laboral que cualquier otra persona, por lo que estarán expuestos a la escasez de plazas de empleo, que es generada por la automatización de procesos en las organizaciones, la presencia de la mujer en el mundo laboral, el aumento en la población y el incremento en la competitividad.

Adicionalmente hay que tomar en cuenta que existen algunas variables que afectan en la inserción de personas con discapacidad como: las adecuaciones físicas que debe tener el entorno de trabajo, la capacitación que es necesaria para los empleados que trabajarán de manera directa con la persona con discapacidad o sus jefes inmediatos, la capacitación de la persona con discapacidad, entre otras condicionantes que deben ser solventadas durante el proceso de inclusión.

Frente a todos los factores antes mencionados y como una salida para las personas con discapacidad existen algunas modalidades de empleo conocidas como empleo protegido y abarcan a los centros especiales de empleo y a los centros ocupacionales.

Los centros especiales de empleo son organizaciones que realizan una actividad laboral, donde la mayor parte de trabajadores son personas con discapacidad. Mientras que los centros ocupacionales son destinados a brindar asistencia a personas que por su severa discapacidad no pueden ocupar un puesto de trabajo, donde se realizan ciertas actividades como: elaboración de tarjetas, cerámicas, pinturas, manualidades u otros objetos para una organización, a cambio de una remuneración económica.

El equipo encargado de la orientación laboral de la persona con discapacidad decidirá si es necesario insertar a la persona en la modalidad de empleo protegido en base a las evaluaciones y talleres que ha desarrollado con anterioridad.

Además existe una modalidad de empleo con apoyo, que se orienta a personas que necesitan supervisión y asistencia mientras realizan las actividades que se les han designado dentro de una organización. La persona encargada de la supervisión debe estar capacitada para poder llevar a cabo el proceso correctamente.

### **3.3.2. La Inserción Laboral en el Ecuador**

En agosto de 2013, el Secretario de Empleo y Salarios del Ministerio de Relaciones Laborales en funciones, Xavier Estupiñán, manifestó que desde el año 2010 hasta julio del 2013 se han contabilizado un total de 56 000 ingresos de personas con discapacidad a diferentes empresas del país. Registrándose el valor más alto en el año 2012 con un total de 28 000. En los primeros siete meses del año 2013 se registran 12000 ingresos de personas con discapacidad al mundo laboral.<sup>8</sup>

En el Ecuador, desde el año 2009, se está intentando garantizar el derecho al trabajo y a la equidad de las personas con discapacidad, mediante la disposición de algunos artículos que ayuden a llevarlo a cabo; teniendo como uno de los entes controladores de su correcta ejecución al Ministerio de Relaciones Laborales.

Uno de los artículos del Código del Trabajo impone que los empleadores, que cuenten con al menos veinticinco empleados, deben contratar, al menos, una persona con discapacidad en el primer año de vigencia de la ley, aumentando en un 1% por año, hasta llegar al quinto año en

---

<sup>8</sup> El Telégrafo. (2013, julio de 30) El Telégrafo. Recuperado el 27 de agosto, 2013 de En 3 años se quintuplicó la cifra de personas con discapacidad que trabajan: <http://www.telegrafo.com.ec/noticias/sociedad/item/en-3-años-se-quintuplico-cifra-de-discapacitados-que-trabajan.html>

donde la contratación será del 4% del total de empleados. Este valor se mantendrá constante por los años consiguientes.<sup>9</sup>

Este numeral pretende garantizar la inserción laboral de las personas con discapacidad, imponiendo sanciones mensuales a quien incumpla con este numeral de diez remuneraciones básicas unificadas y en entidades del Estado también se sancionará a la autoridad responsable con una remuneración básica adicional.

En el numeral 35 del artículo 42, Obligaciones del empleador, del Código del Trabajo se responsabiliza a la organización de realizar adaptaciones necesarias de conformidad con la disposición de la Ley de Discapacidad para facilitar la inserción laboral.<sup>10</sup>

Adicionalmente el Ministerio de Relaciones Labores cuenta con una bolsa de empleo en la cual la persona con discapacidad que esté en busca de una plaza de empleo puede registrarse para ser destinado a las distintas vacantes en las entidades públicas con el fin de facilitar el proceso de búsqueda de trabajo.

---

<sup>9</sup> Dirección Nacional de Asesoría Jurídica de la PGE. (2013, Mayo) Código del Trabajo Ecuador.

<sup>10</sup> Dirección Nacional de Asesoría Jurídica de la PGE. (2013, Mayo) Código del Trabajo Ecuador.

## **4. Metodología de Desarrollo**

El capítulo actual, presenta la metodología Introduction to the Team Software Process, que se aplicará para el desarrollo de la parte práctica del presente trabajo. Contiene una descripción de cada una de sus fases, así como los roles que se ejecutarán.

### **4.1. TSPi**

Introduction to the Team Software Process o TSPi es un marco de trabajo que enseña a las personas, cómo aplicar los conocimientos de ingeniería de software y de procesos en un ambiente de trabajo en equipo. TSPi se basa en la metodología TSP, mantiene los mismos conceptos y métodos pero reduce la complejidad, debido a que TSP está diseñado para ejecutar proyectos a mayor escala que pueden tomar años en desarrollarse y donde los equipos de trabajo están formados por alrededor de veinte personas.

TSPi se enfoca en mejorar la productividad del equipo debido a que la forma de planificar que propone nos permite gestionar de mejor manera el tiempo. Además brinda pautas para construir y mantener un equipo de trabajo efectivo, donde todos los miembros persiguen los mismos objetivos, se apoyan y están dispuestos a pedir ayuda cuando la necesiten.

Adicionalmente permite definir roles, delegar responsabilidades y nos brinda métodos que permitan hacer un seguimiento de actividades. TSPi cuenta con formularios que permiten evaluar los procesos. El evaluar de manera continua optimiza el trabajo y brinda información a los otros miembros del equipo sobre el rendimiento de cada uno.

#### **4.1.1. Principios de TSPi**

TSPi se basa en cuatro principios básicos que se describen a continuación:

1. El aprendizaje se facilita cuando el equipo se guía en un proceso bien definido y se obtiene una pronta retroalimentación. TSPi cuenta con un proceso que ayuda a que el aprendizaje sea sistematizado. Además el desarrollo en ciclos permite la retroalimentación en plazos cortos, basada en evaluaciones realizadas mediante los formularios establecidos.

2. Un equipo de trabajo debe contar con objetivos establecidos con los que todos estén comprometidos, un ambiente de soporte y apoyo, y un líder innato que guíe a los miembros para poder convertirse en un equipo productivo.
3. Cuando los miembros del equipo tengan problemas durante el desarrollo deben contar con una guía de soluciones efectivas que han sido probadas con anterioridad como los consejos expuestos por TSPi.
4. La capacitación es más eficiente cuando está basada en conocimientos previos obtenidos a partir de experiencias en otros proyectos de desarrollo de software.

#### **4.1.2. La Lógica de TSPi**

El TSPi propone un marco que favorece el trabajo en equipo, de manera colaborativa, planificada y con calidad. Según DeMarco, las causas más frecuentes de fracaso en un proyecto de software no son problemas técnicos; son generalmente problemas personales entre los miembros del equipo, además del mal manejo de la presión.<sup>11</sup>

El manejo de la presión es un tema que se debe tener siempre en cuenta y para lo cual se tiene que estar siempre preparado. Un equipo bien constituido, con sus roles y funciones bien definidos y con objetivos comunes, es la base para resistir la presión. La presión es generalmente algo natural, algo que cada persona genera al intentar cumplir las metas propuestas por sus jefes o compañeros de trabajo.

Otra razón frecuente de fracaso en proyectos de software es una planificación poco realista, ya que simplemente aumenta la preocupación y la presión sobre los miembros del equipo por cumplir metas inalcanzables, lo cual puede desencadenar incluso en problemas de autoestima en los miembros del equipo.

---

<sup>11</sup> DeMarco, T. (1988). Looking for Lost Keys. Software Magazine.

Los problemas más comunes que se presentan entre los miembros de un equipo de trabajo, son según Pournaghshbanb, problemas de liderazgo, participación, cooperación, aplazamiento, calidad, atribución de funciones y la evaluación.<sup>12</sup>

Un liderazgo no efectivo puede desencadenar el incumplimiento de la planificación o la dificultad en mantener la disciplina de los miembros del equipo en cuanto al desarrollo de su trabajo. A pesar de que existen líderes innatos, muchos tienen que aprender a desarrollar las habilidades necesarias para liderar equipos.

La falta de cooperación es un problema que se presenta a menudo, cuando un miembro del equipo no se siente comprometido con el proyecto, entonces no se involucra en el trabajo y es necesario separar a este miembro para evitar más problemas en el equipo.

Cuando un equipo trabaja de manera conjunta por lograr un objetivo, y cada miembro hace su mayor esfuerzo, es posible lograr cosas muy complejas, sin embargo, si un miembro del equipo no se esfuerza, los demás miembros del equipo lo notarán, y se empezará a generar un ambiente de malestar que puede llegar a dañar todo el ambiente de trabajo.

El aplazamiento de responsabilidades y el retraso en el trabajo, es algo que se ve comúnmente cuando no se han fijado límites de tiempo claros, o cuando se toma demasiado tiempo para empezar el proyecto; esto suele ser la consecuencia de no tener objetivos y procesos claros, así como una planificación definida.

La baja calidad de un producto se debe principalmente a un levantamiento de requerimientos pobre, a un diseño con una descripción insuficiente o a una implementación que no sigue buenas prácticas de codificación.

La atribución de funciones es un problema que se suele presentar durante el diseño, ya que naturalmente, un ingeniero busca mejorar los productos que diseña. Muchas veces no está claro el límite entre el diseño de los requerimientos y las mejoras que se implementan.

---

<sup>12</sup> Pournaghshbanb, H. (Febrero de 1990). Twenty-First SIGCSE Technical Symposium on Computer Science Education. SIGCSE Bulletin 22, No. 1. The Students' Problems in Courses with Team Projects.

La evaluación en algunos casos crea competencia entre los miembros del equipo, especialmente, cuando los miembros que tienen una motivación alta, reciben una evaluación baja.

Un equipo eficiente que trabaje con calidad requiere una buena relación entre sus miembros, el conocimiento del producto que se realizará por parte de todos los miembros del equipo y el avance del mismo. Es importante que el trabajo represente un reto y sea interesante, de manera que el equipo se comprometa con los objetivos.

Se puede lograr un equipo sólido a través del tiempo, o por medio de un proceso consciente que permita construirlo; para lo cual TSPi cuenta con diferentes fases, que aportan al proceso de trabajo en equipo.

Una de las fases iniciales para la construcción de un equipo sólido, es lanzamiento; donde se definen los roles que tendrá cada miembro del equipo. TSPi, cuenta con una distribución de actividades para cada uno de los roles, lo cual facilita la organización del trabajo.

Posteriormente, el equipo debe acordar una estrategia para el desarrollo del producto; para lo cual, decide la forma más adecuada para dividir el producto final en diferentes ciclos de desarrollo, así como los plazos en los que se entregará cada producto.

En todo momento debe existir comunicación entre los miembros del equipo. TSPi sugiere mantener una reunión semanal, en donde los miembros del equipo puedan conocer y dar a conocer el avance de su trabajo, de modo que no se creen obstáculos para el trabajo que cada uno realiza.

#### **4.1.3. Roles de TSPi**

La definición de roles es una actividad fundamental que ayuda a establecer las responsabilidades de cada miembro del equipo, de esta forma cada persona sabrá que actividades debe ejecutar y en qué momento, cuales son los objetivos del rol que debe desempeñar y como conseguirlos.

Según Humphrey, todos los miembros deben tener un objetivo en común: ser un miembro colaborador y efectivo. Es esencial que todos hagan su mejor esfuerzo para producir productos

de alta calidad dentro del tiempo establecido. Además deben ayudar a los otros ingenieros cuando lo necesiten y colaborar en la resolución de problemas, ya que de este esfuerzo personal depende el éxito del equipo.<sup>13</sup>

Los roles propuesto por TSPi, se describen a continuación:

#### ***4.1.3.1. Líder***

Un líder debe disfrutar el guiar a su equipo, motivarlo y apoyarlo. Es muy importante que el líder sepa identificar a tiempo síntomas de posibles problemas, conversar con los miembros del equipo para buscar una solución y tomar decisiones cuando no sea posible llegar a consensos de manera rápida.

El líder debe también estar preparado para cubrir a cualquier miembro del equipo o apoyar a los miembros que mayor carga de trabajo tengan asignada y no vayan a lograr completarla en los tiempos acordados.

Un líder debe ser respetuoso con los miembros de su equipo y tener paciencia para conocer la forma de trabajo de cada uno, así como sus habilidades, destrezas e ideas creativas; de modo que pueda sacar lo mejor de cada uno de ellos.

También, el líder debe vigilar que los miembros del equipo realicen su trabajo de acuerdo a la planificación, ya que en muchos casos, cuando los miembros de un equipo cuentan con una planificación muy exigente, tienden a tomar atajos y omitir partes que consideran menos importantes.

El líder tiene que detectar cuando un miembro del equipo no pone igual empeño en el trabajo que realiza que los demás miembros o no quiere dedicar el tiempo suficiente para realizar su trabajo. Es importante solucionar pronto este tipo de situaciones para evitar que se generen malestares internos en el equipo.

---

<sup>13</sup> Humphrey, W. S. (2000). Introduction to the Team Software Process. Addison Wesley Longman Inc.

#### **4.1.3.1.1. Objetivos del Líder**

Un líder debe mantener un ambiente adecuado que permita el desenvolvimiento del equipo de manera eficiente y productiva. Además, motivar a los miembros del equipo para que den su mejor esfuerzo durante la ejecución del proyecto, lo que permitirá cumplir con los plazos planificados y los parámetros de calidad acordados.

También es el llamado a identificar a tiempo los problemas que se presenten entre los miembros del equipo y dar soluciones objetivas, con la finalidad de evitar obstáculos para el desarrollo del proyecto.

Adicionalmente, el líder debe mantener informado a su gerente sobre el estado del proyecto, así como de sus avances y problemas que se puedan presentar.

#### **4.1.3.1.2. Actividades Principales del Líder**

El líder, además de motivar a los miembros del equipo para que realicen su trabajo de la mejor manera posible, debe ser quien dirija la reunión semanal, creando la agenda a ser tratada y actuando como facilitador con el fin de evitar que la discusión sobre cualquier tema pendiente salga de control.

El líder debe presentar semanalmente el cuaderno del proyecto con todos los productos elaborados hasta el momento, así como informar sobre el avance y los problemas que haya tenido el equipo. Además, debe mantener el cuaderno del proyecto y guiar a los demás miembros del equipo en la elaboración del reporte de ciclo.

Adicionalmente, el líder debe asignar las tareas, de manera que cada miembro del equipo desee hacer la tarea que se le asigne o escoja. El líder deberá hacer una lista de las tareas que nadie escogió así como de los miembros con tiempo disponible, para asignarles dichas tareas.

#### **4.1.3.2. Administrador de Desarrollo**

El administrador de desarrollo disfruta el construir y desarrollar cosas, ya que el desarrollo de software es la creación de sistemas informáticos que den soluciones a problemas que se puedan presentar o que optimicen procesos.

Además debe contar con aptitudes para el diseño y desarrollo de productos de software con el fin de guiar al equipo de trabajo para obtener un producto de calidad que cumpla con los requerimientos básicos que han sido establecidos.

Una característica importante, y a veces difícil de desarrollar, es escuchar a los miembros del equipo. Debe tener la capacidad de captar y utilizar el conocimiento y el talento de todo el equipo de la forma más óptima posible. Es sumamente importante escuchar de manera objetiva y entender completamente todas las propuestas a pesar de que algunas veces estén en contra de los ideales del administrador.

#### **4.1.3.2.1. Objetivos del Administrador de Desarrollo**

El administrador de desarrollo debe guiar al equipo durante el desarrollo del producto de software, potenciando y utilizando todos los conocimientos y aptitudes de cada miembro, con el fin de que el producto cumpla altos estándares de calidad y plasme los requerimientos básicos, que han sido correctamente transmitidos desde el SRS.

#### **4.1.3.2.2. Actividades Principales del Administrador de Desarrollo**

El administrador de desarrollo es una parte esencial en la definición de la estrategia y de los requerimientos debido a que cuenta con conocimientos guiará al equipo durante estas actividades para generar productos de alta calidad.

Además debe liderar al equipo durante el diseño. Es recomendable invertir un poco más de tiempo en el desarrollo de un diseño simple, preciso y completo, que invertirlo posteriormente. El administrador de desarrollo también se encargará de designar las actividades de durante la implementación a cada miembro del equipo de trabajo.

Es indispensable que verifique que el plan de pruebas sea completo y que no requiera muchos recursos para ejecutarlo. Usualmente durante la documentación y la revisión de la planificación de pruebas, es donde se puede detectar gran cantidad de defectos antes de que las pruebas comiencen.

#### ***4.1.3.3. Administrador de Planificación***

El administrador de planificación se distingue por estar siempre interesado en planificar cualquier actividad que realice, por su deseo de apoyar al equipo y por estar enfocado en la calidad del producto final. Es el responsable de guiar, ayudar y presionar a los miembros del equipo a planificar y hacer el seguimiento de su trabajo, promoviendo la importancia de planificar y los beneficios que produciría para el desarrollo del producto.

Adicionalmente el administrador de planificación debe disfrutar del procesamiento de datos que le permitirán posteriormente evaluar el avance de equipo en el desarrollo del proyecto, que facilitará la toma de decisiones.

##### **4.1.3.3.1. Objetivos del Administrador de Planificación**

El administrador de planificación se encargará de realizar un plan que sea completo y preciso, es decir, que contemple todas las actividades necesarias para el desarrollo del producto; donde el tiempo promedio de cada actividad debe ser de cinco horas y no mayor a diez horas.

Al igual que todos los roles, éste depende de todos los miembros, que en este caso deben llenar y enviar las formas con su información personal actualizada. Es importante que los miembros lo realicen hasta un tiempo límite acordado durante la fase de lanzamiento. Si no fuera así, el administrador de planificación, debe presionar a los miembros a hacerlo y si continua teniendo dificultades, escalar el problema.

##### **4.1.3.3.2. Actividades Principales del Administrador de Planificación**

Resulta complicado realizar la planificación antes de la definición de requerimientos; sin embargo, es sumamente importante hacerlo debido a que de esa forma se evitará invertir grandes cantidades de tiempo en la recolección de requerimientos. El administrador de planificación es el encargado de liderar al equipo durante el desarrollo de la planificación.

Es primordial que el administrador de planificación se encargue de balancear el trabajo que realizará cada integrante. Cuando la planificación no está balanceada es muy probable que algunos ingenieros tengan más trabajo que otros, lo cual puede producir retrasos.

Durante la producción de un plan balanceado se tiene como objetivo utilizar de la manera más óptima al equipo por lo que es recomendable que esta actividad se realice con todos los miembros presentes, debido a que de esta forma se agilizará el proceso. También es importante considerar que en ocasiones se tienen que asignar tareas a personas que parecen no ser lo suficientemente capaces de hacerlo; sin embargo, se debe tener en cuenta que muchas personas son más capaces de lo que parecen o creen.

El realizar un continuo seguimiento de las tareas asignadas provee al equipo información sobre el progreso del proyecto lo que les permite tomar decisiones que garanticen que el producto se finalice dentro del tiempo estimado. El Administrador de Planificación debe liderar al equipo durante la ejecución de esta actividad.

#### ***4.1.3.4. Administrador de Calidad***

Un administrador de calidad debe ser una persona interesada en la calidad del software y el producto final que su equipo elabora. Un administrador de calidad tiene que estar dispuesto a apoyar a su equipo y hacer críticas constructivas que permitan mejorar.

Además, debe ser una persona que muestre interés por los procesos, así como las métricas que permiten evaluar las diferentes fases de los mismos. El administrador de calidad será de preferencia alguien con experiencia en inspecciones de calidad, es decir revisiones en busca de defectos.

##### **4.1.3.4.1. Objetivos del Administrador de Calidad**

El administrador de calidad debe asegurar que todos los miembros del equipo reporten de manera adecuada su trabajo durante el proceso. Es el encargado de guiar y aclarar al equipo cualquier duda referente al proceso.

También debe liderar la identificación de problemas de calidad, así como la documentación de soluciones y sugerencias de mejora para el proceso.

Es importante que el equipo cuente con estándares que indiquen las directivas de trabajo, para esto el administrador de calidad, liderará la elaboración de estándares de codificación, nombramiento y reutilización.

Además, el administrador de calidad debe velar por que se siga el proceso de manera ordenada y estructurada, sin saltar etapas. A menudo, algunos miembros del equipo se verán tentados a tomar atajos para acortar el proceso. El administrador de calidad deberá informar al líder en caso de que un miembro del equipo no muestre una predisposición para seguir el proceso.

Adicionalmente, el administrador de calidad debe velar porque todas las inspecciones de código sean documentadas y reportadas adecuadamente, al igual que todas las reuniones y acuerdos alcanzados.

#### **4.1.3.4.2. Actividades Principales del Administrador de Calidad**

El administrador de calidad es el encargado de liderar la elaboración y el seguimiento de un plan de calidad. Para esto, desde la fase de planificación, se acordará con los miembros del equipo los estándares de calidad a cumplir. También estará a cargo de monitorear que se cumplan los indicadores acordados, por lo que en cada reunión, deberá reportar los errores encontrados, de manera que todos los miembros del equipo conozcan sobre éstos y los eviten.

Adicionalmente, el administrador de calidad, deberá mantener una documentación de las propuestas de mejora para el proceso que elabore cualquier miembro del equipo.

El administrador de calidad liderará la elaboración y mantendrá los estándares que servirán de guía para el trabajo del equipo, como estándares de codificación, de diseño, de documentación, entre otros. También velará por el cumplimiento de los estándares, ya que esto permitirá mantener el trabajo de cada miembro del equipo entendible para todos.

Además, deberá revisar que todos productos hayan cumplido con el proceso y su respectiva recolección de datos en cada fase, con las metas de calidad acordadas antes de ser enviado a la mesa de control de configuración para ser adjuntado a la línea base.

El administrador de calidad deberá actuar como moderador en las inspecciones realizadas por los diferentes miembros del equipo, revisando que se haya tomado el tiempo necesario en cada una de estas, así como verificar los datos que correspondan a la documentación de los defectos encontrados.

Adicionalmente, deberá registrar todas las decisiones y acuerdos a los que el equipo haya llegado durante las reuniones, esto incluye actividades, responsables y tiempos acordados.

También deberá participar en la producción del reporte del ciclo, discutiendo con los miembros del equipo las metas de calidad conseguidas, las razones por las que no se consiguieron algunas metas; así como los defectos que quedan por corregir.

#### ***4.1.3.5. Administrador de Configuración***

El administrador de configuración deberá conocer las herramientas y métodos que se utilizarán durante el desarrollo del proyecto, de modo que pueda guiar a los demás miembros del equipo durante el proceso. Además deberá tener facilidad por el uso y aprendizaje de nuevas herramientas y sistemas que puedan ser de ayuda al equipo durante el ciclo de desarrollo.

##### **4.1.3.5.1. Objetivos del Administrador de Configuración**

El administrador de configuración tendrá entre sus objetivos, el revisar las herramientas y métodos que el equipo utilizará durante el desarrollo del trabajo, brindando apoyo a cualquier miembro del equipo que lo necesitare.

Además se asegurará que ningún cambio se realice a la línea base sin que éste sea autorizado por la mesa de control.

Verificará que los riesgos y problemas del equipo se registren adecuadamente y sean expuestos durante las reuniones.

También será el encargado de documentar y decidir qué código puede reusarse, de manera que el equipo cumpla con sus metas de reutilización de código.

##### **4.1.3.5.2. Actividades Principales del Administrador de Configuración**

El administrador de configuración es el encargado de guiar al equipo en la configuración de un entorno de trabajo igual para todos los miembros del equipo; para ello, tomará en cuenta las experiencias que los miembros del equipo hayan tenido con diferentes herramientas.

El administrador de configuración deberá liderar la mesa de control de configuración, estableciendo un procedimiento para el manejo de los cambios que se planteen a la línea base. Además debe establecer el procedimiento que los miembros de equipo deberán seguir para obtener o agregar un producto a la línea base, así como el manejo de versiones de la misma.

También deberá mantener el glosario del sistema, que llevará el nombre del sistema, nombre de módulos, componentes y demás términos que vayan apareciendo durante el desarrollo del sistema y necesiten ser documentados para un mejor entendimiento de todo el equipo.

Adicionalmente, el administrador de configuración deberá llevar registro de los riesgos y problemas encontrados, de manera que no quede ningún problema sin ser resuelto y se asignen responsables en caso de ser necesario.

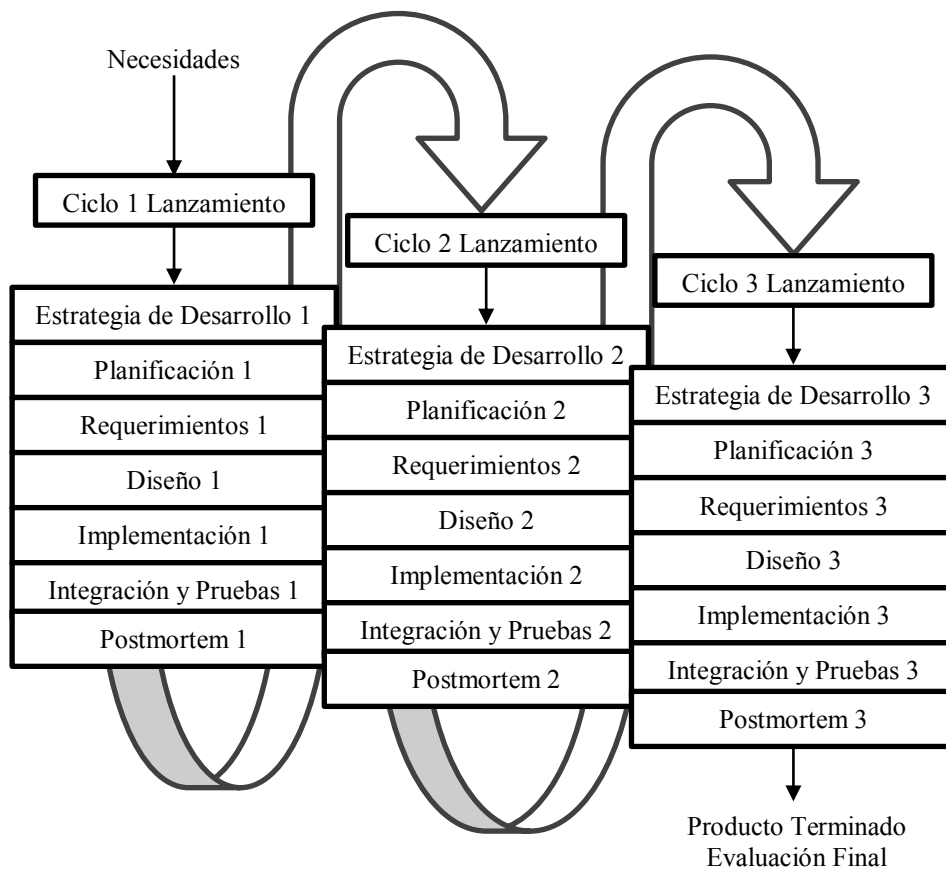
El administrador de configuración procurará que todos los miembros del equipo piensen en la reutilización durante las fases de diseño y desarrollo para contar con componentes reutilizables, los mismos que deberán ser mantenidos y documentados para que todos los miembros del equipo puedan hacer uso de ellos.

El administrador de configuración también deberá formar parte en la elaboración de reporte del ciclo de desarrollo, especialmente en la sección de soporte.

#### **4.1.4. Proceso de TSPi**

TSPi recomienda desarrollar los productos en ciclos, debido a que permite identificar problemas de funcionalidad, diseño y usabilidad cuando es más fácil solucionarlos. Además nos brinda una retroalimentación para mejorar en todos los aspectos posibles la calidad del producto.

En los ciclos iniciales se deben realizar las versiones del producto que son más viables; tomando en cuenta que en cada ciclo se debe producir una versión del producto que pueda ser probada. La versión desarrollada debe tener el tamaño adecuado para que pueda ser culminada en el tiempo establecido.



**Ilustración 3: Estructura y Flujo de TSPi<sup>14</sup>**

En la figura anterior se muestra la estructura y el flujo de TSPi dividida en tres ciclos de desarrollo, en cada ciclo se debe seguir el proceso establecido hasta obtener el producto final. El proceso de TSPi está formado por ocho fases que son: Lanzamiento, Estrategia de Desarrollo, Planificación, Definición de Requerimientos, Diseño, Implementación, Integración y Pruebas y Postmortem.

<sup>14</sup> Humphrey, W. S. (2000). Introduction to the Team Software Process. Addison Wesley Longman Inc.

A continuación se describen las fases de TSPi.

#### ***4.1.4.1. Lanzamiento***

Un equipo debe tener objetivos en común con el fin de que todos se sientan comprometidos con el trabajo que se desea realizar. Además es indispensable que cada miembro tenga su o sus roles definidos, de esta forma sabrán cuáles son sus responsabilidades en el desarrollo del producto. Estas actividades se realizan durante la fase de lanzamiento con el fin de mantener una mejor relación entre los miembros del equipo.

Adicionalmente se debe definir el día y la hora en la que se llevará a cabo la reunión semanal, que es indispensable para desarrollar la comunicación, tomar decisiones y analizar los productos desarrollados, los datos proporcionados por cada rol y los posibles problemas que se puedan presentar. También se debe analizar el estado del proyecto en relación a lo planificado, esto se realiza con ayuda de los formularios del equipo, que son los consolidados de los datos de cada miembro.

En la fase de lanzamiento es importante establecer estándares de documentación y de almacenamiento de la información en el cuaderno del proyecto. El cuaderno del proyecto debe ser desarrollado por el líder del proyecto en colaboración de todos los miembros, y contiene toda la documentación desarrollada en cada fase, incluyendo estándares, formularios y productos.

##### **4.1.4.1.1. Objetivos del Equipo**

La definición de objetivos es una parte fundamental en la creación de un equipo ya que establecen las metas que desean cumplir y comprometen a los miembros a mejorar su rendimiento durante el desarrollo del producto. Adicionalmente nos brinda bases para poder resolver problemas, desarrollar estrategias y planificar el trabajo.

Los objetivos definidos deben motivar al equipo a luchar por ellos, deben representar un reto; es decir, no ser muy fáciles pero tampoco inalcanzables.

Los objetivos deben ser revisados después de cada ciclo para que sean modificados de acuerdo a las nuevas necesidades que se puedan presentar. Los objetivos que no se han conseguido

durante un ciclo deben ser examinados de forma que se pueda encontrar alternativas para poder alcanzarlos.

#### **4.1.4.1.2. Objetivos de los Miembros y Roles del Equipo**

El esfuerzo que cada miembro realice para desarrollar el producto de software, será crucial para garantizar la calidad del producto final, por lo que cada individuo debe establecer los objetivos que desea cumplir como miembro del equipo y en relación a cada uno de los roles que desempeñará durante cada ciclo, con el fin de mejorar su rendimiento y la calidad de los productos de los cuales es responsable.

#### **4.1.4.2. Estrategia de Desarrollo**

Actualmente los sistemas son desarrollados por módulos y es importante contar con una estrategia que nos guíe durante el desarrollo de cada uno y la integración de los mismos. Estos módulos deben ser desarrollados en ciclos, donde en cada ciclo se generará una versión del producto probada, que se integrará a las versiones subsecuentes.

Durante esta fase se debe desarrollar un Diseño Conceptual, donde se detallen las características del producto que se desea entregar, como se llevará a cabo la construcción del producto de software, que componentes y funciones serán necesarios, que tamaño tendrán y que tiempo se necesitará para desarrollar todo el producto.

Un problema que ocurre con frecuencia durante esta fase, es que los ingenieros pueden comenzar a diseñar antes con el fin de tener medidas más exactas, esto se debe evitar por completo, la estimación realizada en esta fase es hipotética.

Además se debe identificar y evaluar los principales riesgos del proyecto. El pensar en los impactos y ocurrencia de los riesgos en una etapa temprana, otorgará la oportunidad de evitar que se produzcan o controlarlos en caso de que sucedan.

Una estrategia para evitar algunos riesgos, es la reutilización. El administrador de configuraciones debe establecer que productos pueden ser reutilizados y colocarlos en una librería que se encuentre al alcance de todos los miembros del equipo. El ejemplo más común, es la reutilización de código, se debe tratar de desarrollar código que pueda solventar múltiples

necesidades, al que solo baste con añadir algunas líneas de código o no realizar ninguna modificación para ser utilizados en otros proyectos. La reutilización de componentes reduce el esfuerzo y el tiempo, aumentando la productividad.

#### ***4.1.4.3. Planificación***

El planificar un proyecto ayuda a realizar el trabajo de una manera más eficiente, optimizando recursos como tiempo y dinero, y dando la prioridad adecuada a cada actividad.

En esta fase se definen las actividades que deben realizarse y en qué orden, lo que además de mejorar el trabajo de cada miembro, impedirá que se pase por alto el desarrollo de algún componente.

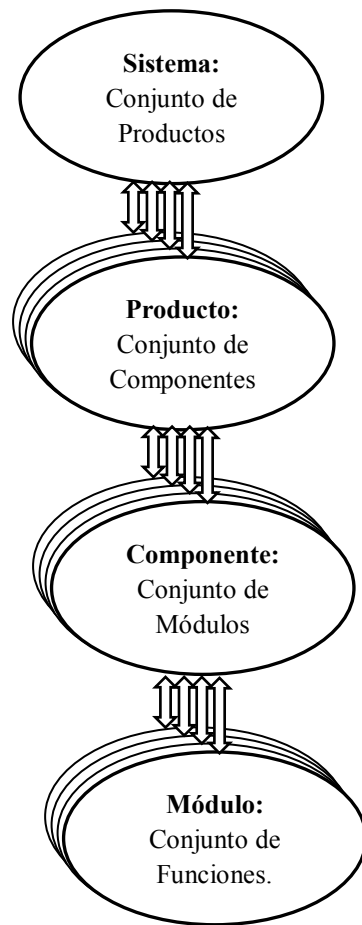
Durante el desarrollo del proyecto es probable que se presenten actividades que no fueron contempladas en la planificación y muchas veces el volver a desarrollar un plan que incluya esas actividades puede tomar más tiempo que el requerido para llevar a cabo dichas tareas, por esta razón TSPi recomienda que se designe el 10% del tiempo total del proyecto a estas actividades varias que se las conoce como Management & Miscellaneous (M&M).

##### **4.1.4.3.1. Niveles de Estimación**

Cada miembro del equipo debe realizar su planificación individual para que sea consolidada en la planificación del equipo. Es importante respetar el tiempo que cada ingeniero ha destinado a cada actividad, ya que estos serán tiempos reales.

Los ingenieros deben realizar estimaciones de tamaño, tiempo y defectos inyectados desde los niveles más bajos de detalle hacia los niveles más altos, finalizando con las estimaciones de todo el sistema.

TSPi propone una estructura de partes y colecciones que pueden representar a un sistema. El nivel más bajo está constituido por las funciones que serán parte de módulos, los mismos que formaran componentes. El conjunto de componentes será un producto y finalmente el sistema estará constituido por uno o varios productos.



**Ilustración 4: Partes y Colecciones<sup>15</sup>**

Es recomendable realizar la estimación a partir de los módulos, debido a que la planificación se realiza antes del diseño detallado por lo que es posible que no se consideren todas las funciones necesarias para cada módulo.

---

<sup>15</sup> Humphrey, W. S. (2000). Introduction to the Team Software Process. Addison Wesley Longman Inc.

#### **4.1.4.3.2. Proceso de Planificación**

El proceso de planificación toma como parámetros de entrada los productos de la fase de estrategia. Todos los datos relacionados a estimaciones del tamaño de los componentes deben ser ingresados en la forma SUMS.

Posteriormente se debe producir un plan del equipo, donde se realiza una lista de las actividades que se llevarán a cabo con los recursos necesarios para cada una. Toda la información desarrollada debe ser documentada en la formas del equipo.

A continuación se estima el número de defectos que serán inyectados, detectados y corregidos en cada fase. Estos valores serán parte del plan de calidad y de la forma SUMQ.

Después se generan los formularios para cada ingeniero, con las tareas que deben realizar y los tiempos límite de entrega para que sean revisados. Los formularios finales deben ser entregados al administrador de planificación para que se realicen los ajustes necesarios.

Una vez que se han realizado las modificaciones finales, se debe balancear el trabajo. Si se detecta que un miembro está planeando demorarse más que otros. Se debe dividir sus actividades en subtareas y repartirlas entre los otros miembros.

Si se realizar modificaciones se debe volver a ejecutar el proceso desde el paso tres. Cuando el plan esté balanceado se generará el plan final. Es esencial distribuir copias de todos los productos de esta fase a los miembros del equipo.

A continuación se presenta una figura que resume el proceso de planificación, representando los pasos que se debe realizar mediante elipses y los productos que se obtendrán mediante rectángulos.

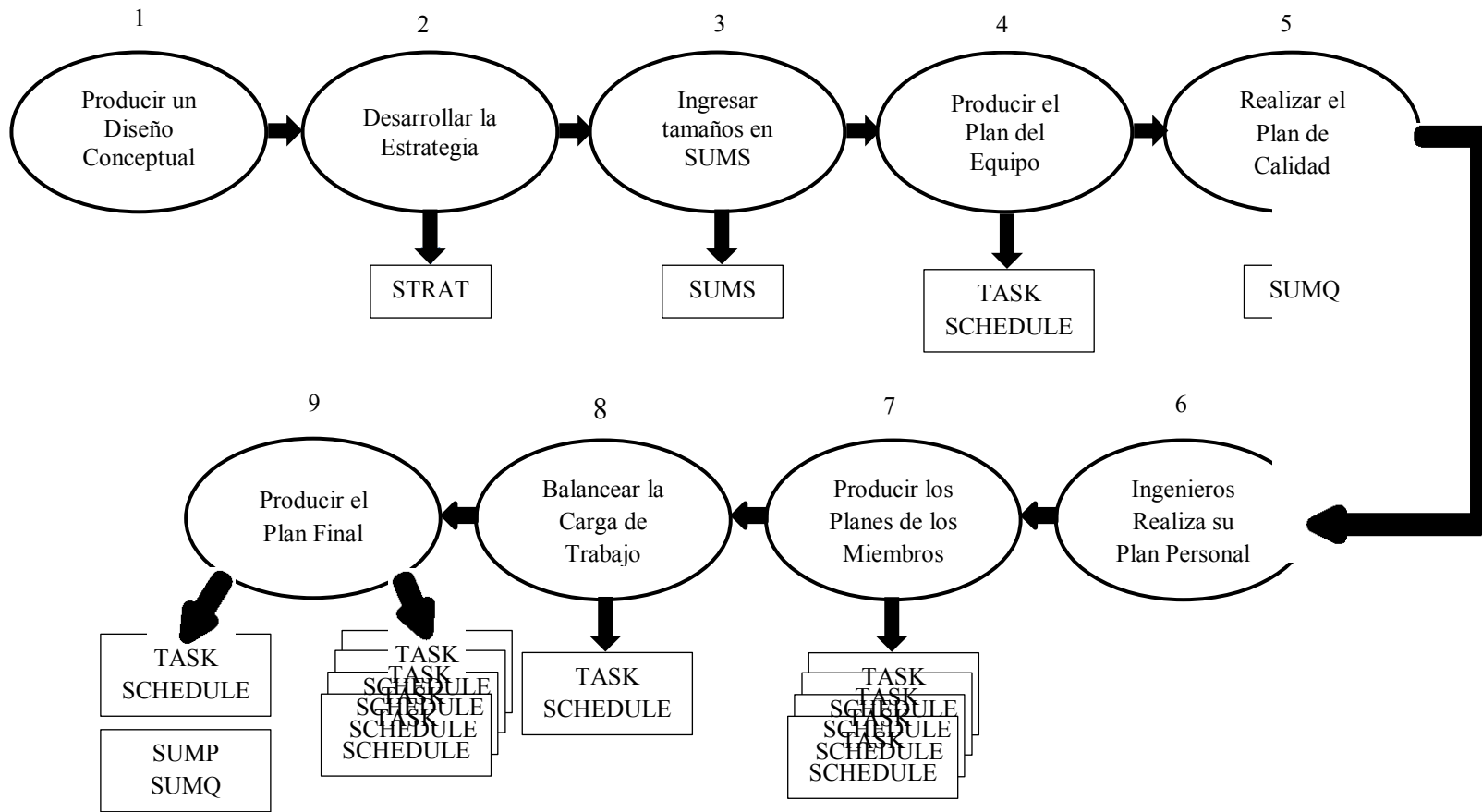


Ilustración 5: Proceso Planificación<sup>16</sup>

<sup>16</sup> Humphrey, W. S. (2000). Introduction to the Team Software Process. Addison Wesley Longman Inc.

#### ***4.1.4.4. Definición de Requerimientos***

Durante esta fase se establecen las características con las que contará el producto final. Es sumamente importante no suponer ningún requerimiento, ya que esto dará como resultado un producto de mala calidad; en su lugar se deben realizar todas las preguntas que se consideren necesarias para aclarar los requerimientos que no estén completamente claros.

Posteriormente se reescriben los requerimientos y se pide al usuario final que revise y acepte los mismos.

El producto resultante de la fase de requerimientos se conoce como Especificación de Requerimientos de Software SRS (Software Requirements Specification) y contiene una descripción clara del producto final y los criterios de evaluación que asegurarán que el producto cumpla con los requerimientos.

##### **4.1.4.4.1. Cambios en los Requerimientos**

Los cambios en los requerimientos son muy probables y los miembros del equipo deben tratar de gestionarlos de la mejor manera para poder llegar a un conceso y lograr entregar un producto de calidad.

Estos cambios se dan debido a que los usuarios no saben lo que realmente desean hasta tener la oportunidad de interactuar con el producto final, por lo que los ingenieros deben ayudar a los usuarios a definir sus verdaderas necesidades.

El SRS es el documento que permite contar con una especificación de los requerimientos que fueron acordados al inicio del desarrollo y aprobados por el cliente; a partir de éste se podrán definir costos extras para los cambios que se realicen sobre el mismo.

El SRS debe registrar todos los requerimientos funcionales, que describen que funciones realizará el sistema, así como los no funcionales, que especifican las propiedades que tendrá el producto final.

Los requerimientos no funcionales se pueden especificar en dos grupos: las restricciones de diseño e implementación y los requerimientos especiales.

En el primer grupo se especifican la metodología que se debe utilizar para llevar a cabo el diseño, que tipo de arquitectura se usará, el lenguaje de programación, el sistema gestor de bases de datos, el servidor de aplicación, los estándares de codificación, entre otras características del desarrollo del sistema.

Según Sommerville, los requerimientos especiales se clasifican en tres grupos principales que se resumen en el mapa conceptual presentado a continuación:

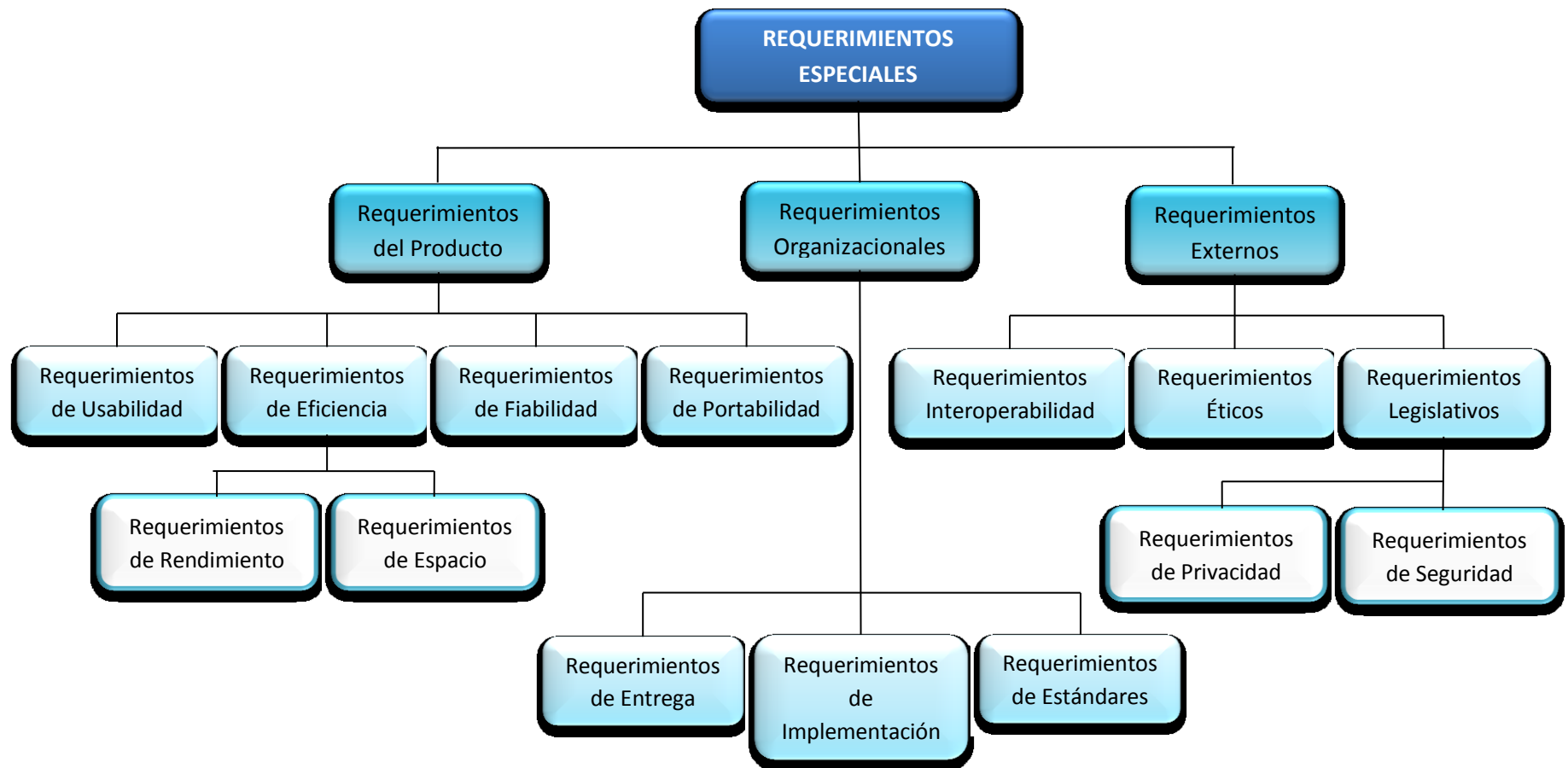


Ilustración 6: Requerimientos Especiales<sup>17</sup>

<sup>17</sup> Sommerville, I. (2005). Ingeniería del Software (7ma ed.). Madrid, Pearson Addison Wesley. pp. 111 y 112.

El primer grupo se enfoca en las propiedades y comportamiento del producto. Está compuesto por varios requerimientos como:

- ✓ **Requerimientos de Usabilidad:** se refiere a la capacidad del sistema, de ser comprendido, aprendido y usado; así como la facilidad de recordar lo aprendido si se dejase de usar por algún tiempo.
- ✓ **Requerimientos de Eficiencia:** Se refiere a las limitaciones de espacio físico disponible para el sistema y a los tiempos máximos en los que puede reaccionar éste.
- ✓ **Requerimientos de Fiabilidad:** se refiere a la tasa de fallos del sistema.
- ✓ **Requerimientos de Portabilidad:** se refiere a la capacidad que debe tener el sistema de correr en computadores con plataformas distintas.

El segundo grupo se relaciona con las políticas y procesos de la organización del cliente y de la que desarrolla el producto. Está compuesto por:

- ✓ **Requerimientos de Entrega:** Se definen fechas y condiciones de entrega
- ✓ **Requerimientos de Implementación:** Se define el lenguaje de programación, base de datos, métodos que deben ser utilizados para el diseño, entre otras.
- ✓ **Requerimientos de Estándares:** Se define el método que se utilizará para el desarrollo.

El tercer grupo se refiere a los factores externos al sistema. Está compuesto por:

- ✓ **Requerimientos de Interoperabilidad:** se define la compatibilidad que debe tener el producto con otros sistemas.
- ✓ **Requerimientos Éticos:** aseguran que el sistema será aceptado por los usuarios y el público en general, es decir que mantenga confidencialidad de los datos de la empresa y que no sea usado para perjudicar a alguien.
- ✓ **Requerimientos Legislativos:** aseguran que el sistema funcione dentro de la ley en relación a privacidad y seguridad.

#### ***4.1.4.5. Diseño***

El diseño es un proceso mediante el cual se especifica cómo un producto debe construirse, describiendo la interacción e integración que existirá entre las partes.

La producción del diseño detallado puede dividirse en componentes que serán desarrollados por cada ingeniero basándose en el diseño de alto nivel que define como las partes encajarán para formar el producto final.

Es importante que todos los miembros del equipo sean parte de la definición del diseño de alto nivel, aunque no lleven a cabo la documentación del mismo, de esta forma se podrán aprovechar las ideas y percepciones que cada ingeniero tenga sobre el producto que se desee desarrollar. Además esta actividad les permitirá comprender de mejor manera la especificación del diseño.

Durante esta fase se desarrollarán bases precisas, completas y de alta calidad que se tomarán de referencia durante la implementación del producto, permitiendo cumplir con todos los requerimientos funcionales y no funcionales e implementar componentes que puedan ser reusados posteriormente.

El reusar componentes optimiza el proceso de desarrollo de un producto, reduciendo el tiempo necesario para producirlo. Además disminuye considerablemente el número de defectos inyectados, debido a que se ha realizado un control de calidad previamente que garantice el correcto funcionamiento de los componentes reutilizados.

Una actividad esencial en el reuso de componentes, es la definición de estándares. Es importante definir el nombrado de variables, estandarizar los mensajes que se presentarán y las respuestas a errores que debe tener el sistema. Adicionalmente, es substancial declarar que tipo de parámetros recibirán y retornarán las funciones. El documentar correctamente los componentes reusables con una descripción precisa del componente e instrucciones de cómo usarlo es una práctica que garantizará el reuso.

#### ***4.1.4.6. Implementación***

Es importante realizar un diseño de alto nivel del sistema y subdividirlo en tantos componentes como sea necesario, hasta adquirir un entendimiento del problema o la lógica del negocio que se automatizará.

Luego, es substancial definir estándares de codificación, de tamaño de errores y también un glosario del sistema.

Es esencial acordar con el equipo los estándares de codificación que se utilizarán, así como tomar en cuenta estándares con los que los miembros del equipo estén familiarizados. No se debe intentar crear estándares correctos desde cero, es recomendable utilizar estándares creados y adaptando de acuerdo a la experiencia. Además, un buen código debe estar comentado, de manera que las inspecciones se faciliten y los componentes reusables sean más entendibles para todo el equipo. Se debe mantener un glosario actualizado y verificar que los miembros del equipo utilizan la misma nomenclatura.

El llevar un registro del tamaño de los productos permite hacer una planificación más certera y verificar el avance en el desarrollo de los mismos. Si un producto es demasiado pequeño, probablemente el dimensionarlo no será un aporte mayor para el proyecto, por lo que no se debería invertir mucho tiempo en ello. Con el tiempo, se puede incluso establecer categorías con tiempos de trabajo para cada una de ellas, de acuerdo al tamaño de los productos.

El definir estándares de clasificación de defectos ayuda a mejorar el proceso de desarrollo, pero es importante recordar que no se debe incrementar la lista de tipos de defectos si no se tiene suficiente documentación que justifique la necesidad de crearlos, ya que el tener una lista muy grande con tipos de defectos, podría causar una demora innecesaria al registrar defectos durante las inspecciones.

El llevar un registro de los defectos, permite redefinir el proceso de desarrollo o incluir estándares de desarrollo que ayuden a prevenir los defectos más frecuentes, que serán los que más tiempo tome detectar durante las inspecciones. Además, para prevenir defectos, es importante la educación; es decir continuar aprendiendo sobre el lenguaje o las herramientas.

Cuando se realizan revisiones del código, se debe comenzar desde los componentes de más bajo nivel, revisando incluso su funcionalidad cuando no esté clara, y continuar progresivamente con los componentes que tienen dependientes.

Para poder reutilizar el código, es importante seguir estándares de codificado, es decir al documentar un código, se debe especificar lo que hace al principio del mismo. Es de gran ayuda en muchos casos el mantener reuniones periódicas con los miembros del equipo para conocer qué funciones se han desarrollado y exponer los nuevos componentes que se considere puedan ser útiles a otros miembros del equipo.

#### ***4.1.4.7. Integración y Pruebas***

En esta fase los componentes se someterán a pruebas que verifique la calidad de cada uno y a la vez del producto final. Los componentes de baja calidad detectados en esta fase, deben pasar por un proceso que solucione los defectos y después ser evaluados nuevamente.

Existen tres tipos de pruebas que se ejecutarán en esta fase las pruebas de construcción, las pruebas de integración y las pruebas del sistema.

Las pruebas de construcción no pretenden probar la funcionalidad de cada componente, sino que deben verificar que todos los componentes que deben ser integrados, se hayan terminado completamente y entregado.

Las pruebas de integración se encargan de revisar que la interacción entre los componentes sea la adecuada.

Existen varias estrategias para realizar las pruebas de integración, algunas se enumeran a continuación:

- ✓ **Big Bang:** Esta estrategia une todos los componentes para probarlos después. Este método requiere menos tiempo de pruebas pero es recomendado para sistemas de cumplan con altos estándares de calidad, ya que los defectos serán más difíciles de detectar.
- ✓ **Uno a la vez:** Esta estrategia trata de integrar uno a uno los componentes para que se realicen pruebas en cada integración. Esta estrategia requiere más tiempo de

pruebas, pero los componentes son probados más a fondo por lo que los defectos serán detectados con mayor facilidad.

- ✓ **Grupal:** En esta estrategia la integración se realiza en grupos que serán probados, lo que reduce el tiempo necesario para realizar las pruebas.
- ✓ **Plano:** El propósito de esta fase es integrar los componentes de alto nivel y después sucesivamente agregar componentes adicionales, realizando pruebas de cada componente adicional.

Durante las pruebas del sistema se debe verificar si el sistema cumple con los requerimientos funcionales y no funcionales que fueron establecidos durante la fase de requerimientos bajo condiciones normales y extremas o adversas.

El tiempo que se puede invertir en la ejecución de pruebas puede ser muy extenso, por lo que es importante contar con un plan de pruebas que cubra las más importantes y que guíe a los ingenieros durante esta fase.

Además, es esencial contar con un documento del seguimiento de cada prueba, donde se especifique la fecha en la que ejecutó, el nombre del ingeniero responsable de llevar a cabo la prueba, el nombre del producto que se probó, recursos utilizados, los resultados esperados y obtenidos en cada una y los defectos detectados.

#### **4.1.4.7.1. Documentación**

Algunos ingenieros deben dedicarse al desarrollo de la documentación mientras los demás ingenieros ejecutan las pruebas de construcción, de integración y del sistema.

Uno de los documentos más importantes de este proceso, es el manual de usuarios. TSPi recomienda que comience con una introducción al sistema, seguido de una explicación de las funcionalidades con las que cuenta y finalmente todas las instrucciones de lo que se necesite saber tomando en cuenta que debe tener un acceso fácil.

Es importante incluir un glosario de ser necesario, una sección de mensajes de error, procesos de recuperación y de solución de problemas. Además debe contar con un índice que guíe al lector a puntos clave del manual.

#### ***4.1.4.8. Postmortem***

Una característica que es esencial en los procesos, es que éstos deben estar sujetos a una mejora continua. Es importante que los procesos sean sometidos a una evaluación que nos permita detectar sus fortalezas y debilidades con el fin de generar productos de alta calidad.

La fase final del proceso de TSPi está relacionada con el mejoramiento de los procesos, en la cual se realiza una evaluación del trabajo que se ha desarrollado durante el ciclo, verificando el cumplimiento de objetivos. Además se rescatan las mejores prácticas que se detectaron durante el desarrollo del producto y que pueden ayudar en la optimización del proceso.

La fase de postmortem comienza con una revisión de todos los productos, liderada por el administrador de calidad, en la que se identifican los puntos donde los procesos fallaron y en qué punto los procesos generaron los resultados deseados.

Además, durante esta fase se evalúa el desempeño que tuvo el equipo durante el ciclo en relación con el cumplimiento de objetivos, el avance contra lo planificado, y el número de defectos inyectados.

Posteriormente, el líder guiará al equipo en la ejecución de la evaluación de cada miembro, los parámetros que se tomarán en cuenta pueden ser: la forma en la que cada miembro realizó su trabajo, el cumplimiento de las metas planteadas y problemas que ha experimentado y las medidas que tomó para solucionarlo.

Finalmente, se realiza un reporte que contenga los resultados obtenidos en cada evaluación, especificando soluciones para los problemas que se hayan presentado y describiendo mejoras que deben implementarse en los procesos.

## **5. Tecnologías de Desarrollo**

El presente capítulo describirá la tecnología Java, la arquitectura EJB y la el sistema gestor de bases de datos MySQL; los mismos que se utilizarán para el desarrollo de este trabajo.

### **5.1. Java**

#### **5.1.1. Breve Historia**

Java, que en sus inicios se denominó “Oak”, es un lenguaje de programación, cuya primera versión funcional fue lanzada 1992. La implementación estuvo a cargo de James Gosling, Patrick Naughton, Chris Warth, Ed Frank y Mike Sheridan, integrantes de un equipo de trabajo de la compañía Sun Microsystems Inc. Luego de trabajar en algunas mejoras, con el aporte de otros colaboradores, en 1995 se lanzó el lenguaje oficialmente con el nombre “Java”<sup>18</sup>.

Java tomó muchas de las características de orientación a objetos de C++, así como la sintaxis de C; lenguajes que dominaban el mercado en aquel momento, debido a su robustez y facilidad de uso por parte de los programadores, ya que se trataba de lenguajes de alto nivel.

Adicionalmente, Java surgió para solventar el problema de portabilidad que se presentaba ante la necesidad de contar con un compilador diferente para cada tipo de CPU, lo cual era muy costoso debido al tiempo que tomaba su desarrollo. Además, Java fue diseñado y desarrollado por programadores que se encontraban en un ambiente real de trabajo. Al mismo tiempo, Java constituyó la solución al surgimiento de la computación distribuida.

#### **5.1.2. Principales características**

Java es un lenguaje de programación de alto nivel ya que su sintaxis es más cercana al lenguaje humano que al lenguaje máquina que es utilizado para enviar comandos al

---

<sup>18</sup> Schildt, H. (2009). Java Manual de Referencia (7ma ed.). México D.F.: McGraw-Hill Interamericana Editores.

procesador. Se distingue por contar con las siguientes características: seguro, robusto, multihilos, portable, de arquitectura neutral, interpretado, dinámico, distribuido y orientado a objetos.

Java cuenta con componentes que implementan algoritmos y estándares de seguridad que abarcan varias áreas como criptografía, infraestructura de clave pública, comunicación segura autenticación y control de acceso. Adicionalmente, cuenta con un ambiente controlado que permite la ejecución segura de las aplicaciones sin comprometer los recursos del sistema.<sup>19</sup>

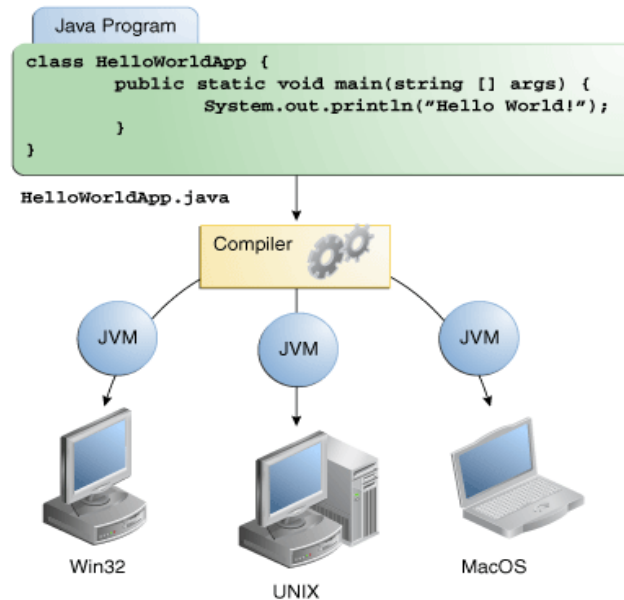
Además de los componentes de seguridad, Java es robusto debido a la administración de memoria que realiza gracias a la abstracción de los punteros de datos; así como, la aritmética de punteros para el programador, la liberación automática de memoria por medio del Garbage Collector y el manejo de excepciones que permite el tratamiento de los posibles errores de ejecución.

Java proporciona la clase Thread, que permite la creación y ejecución de múltiples hilos de manera simultánea e independiente. También cuenta con funciones que permiten la sincronización de hilos.

Java es un lenguaje de programación que permite la portabilidad, gracias a su arquitectura neutral; esto es, una vez compilado el código fuente, se genera un bytecode que puede ser ejecutado en múltiples plataformas de hardware y software que cuenten con la Java Virtual Machine (JVM), la cual especifica tamaños y comportamientos para los diferentes tipos de datos, así como para los operadores aritméticos.

---

<sup>19</sup> Oracle Corporation (20 de marzo de 2013). Oracle. Recuperado el 1 de marzo de 2014, de Java SE Security: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html>



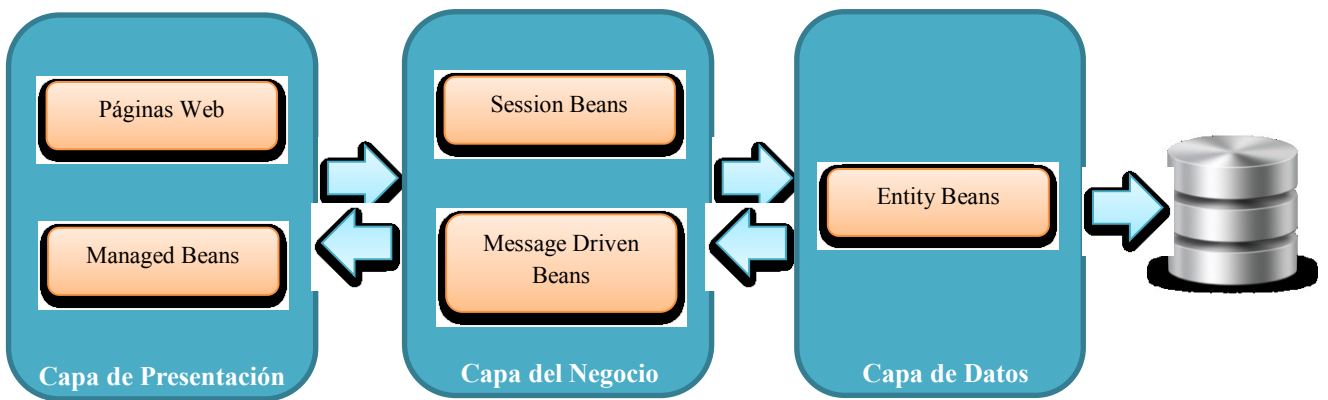
**Ilustración 7: Arquitectura Neutral de Java<sup>20</sup>**

A Java se le atribuye también la característica de distribuido, ya que gestiona los protocolos TCP/IP, permitiendo el acceso a recursos que se encuentran en otras ubicaciones físicas. Además soporta la invocación remota de métodos; es decir, que puede invocar métodos de objetos que se encuentran en un computador diferente.

### 5.1.3. Enterprise Java Beans

Es una arquitectura diseñada para facilitar la creación de aplicaciones multicapa, basadas en componentes, seguras, escalables, distribuidas, transaccionales y multiusuario.

<sup>20</sup> Oracle Corporation (20 de diciembre de 2013). Oracle. Recuperado el 1 de marzo de 2014, de The Java TM Tutorials: <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>



**Ilustración 8: Arquitectura EJB**

Como muchas tecnologías, Enterprise Java Beans separa 3 conceptos fundamentales que son: la lógica del negocio, la capa de presentación y la capa de acceso a los datos.

En la capa de presentación tenemos las páginas web y los Managed Beans. Un Managed Bean es un objeto que permite la interacción entre una página y la capa de la lógica del negocio.

En la segunda capa, conocida como la capa de la lógica del negocio, se encuentran los Session Beans y los Message Driven Beans.

Un Session Bean es un componente que puede ser considerado como el proxy del lado del servidor para el cliente. Este componente contiene los métodos del negocio. Existen tres tipos de Session Beans que son:

- ✓ **Stateless:** Los session beans de este tipo se crean y se eliminan de manera aleatoria con el fin de hacer más eficiente la ejecución de la aplicación. Las peticiones son atendidas por cualquier session bean existente de manera aleatoria.
- ✓ **Stateful:** Los session beans de este tipo se crean cuando un cliente realiza una petición al servidor.
- ✓ **Singleton:** Los session beans se crean una sola vez por cada sesión. Un sólo session bean atiende todas las peticiones de la sesión a la que fue asignado.

Los Message Driven Beans permiten una comunicación asíncrona entre el cliente y el servidor, es decir que cada petición será colocada en un pool a la espera de una respuesta. El cliente puede realizar cualquier actividad diferente mientras espera la respuesta.

En la capa de acceso a los datos, tenemos los Entity Beans, que son una representación en objetos Java de las entidades que se encuentran definidas en la base de datos. Este proceso se conoce también como mapeo de objetos relacionales (ORM).

## 5.2. MySQL

MySQL es un motor gestor de bases de datos multiplataforma, multiusuario y multihilos desarrollado por Monty Widenius de MySQL AB.

MySQL se popularizó debido a las ventajas que puede ofrecer en relación a velocidad, escalabilidad, utilización de recursos, costos y la capacidad de interactuar con otros sistemas.

MySQL no requiere muchos recursos para funcionar correctamente, pruebas realizadas han comprobado que necesita como mínimo un sistema con 32MB de RAM y un procesador de 166 MHz y no necesita estar completamente dedicado a MySQL.<sup>21</sup>

Este software puede ser utilizado bajo la licencia Open Source o bajo una licencia comercial. La primera alternativa se distribuye sin ningún costo bajo los términos de GPL, mientras que la segunda alternativa incluye soporte comercial y se distribuye a costos bajos en relación con los otros gestores de base de datos disponibles en el mercado.

Varios lenguajes de programación como PHP, Perl, Java, Ruby, entre otros; cuentan con bibliotecas que permiten la conexión e interacción sencilla con MySQL. Además, soporta ODBC que brinda otra alternativa de conexión con los diferentes lenguajes de programación.

---

<sup>21</sup> Sasha Pachey, A. (2003). MySQL Enterprise Solutions. Wiley Publishing Inc.

## 6. Desarrollo del Sistema

En el capítulo actual se presentan los productos y la documentación generados en cada fase dividida en dos ciclos, durante el proceso de desarrollo del producto de software.

### 6.1. Primer Ciclo

Los resultados obtenidos durante el primer ciclo se presentan a continuación:

#### 6.1.1. Lanzamiento

La fase de Lanzamiento es una de las más importantes, debido a que durante ésta se definen los objetivos que perseguirán el equipo y cada uno de sus miembros para lograr un producto de calidad. Es esencial que todos los integrantes formen parte de la definición de estos objetivos, ya que de ésta forma, los ingenieros estarán comprometidos de manera personal a cumplirlos.

Además, durante esta fase se establecen los roles que desempeñará cada integrante y cada uno define los objetivos que está dispuesto a conseguir para que el aporte del rol sea constructivo para el equipo.

##### 6.1.1.1. Asignación de Roles

Los roles que desempeñará cada miembro se resumen en la tabla presentada a continuación:

Roles	Ingeniero
Líder Administrador de Calidad Administrador de Configuraciones	Juan Carlos Fernández Ortiz
Administrador de Planificación Administrador de Desarrollo	Estefanía Narváez

Tabla 1: Asignación de roles. Ciclo I.

### ***6.1.1.2. Objetivos del Equipo***

Los objetivos que el equipo se ha comprometido a cumplir durante el desarrollo son los siguientes:

- ✓ Desarrollar un producto de calidad, inyectando el menor número de defectos posible y siguiendo una planificación, que se ajuste a la realidad, para no exceder del tiempo límite de entrega.
- ✓ Seguir de forma estricta un proceso bien definido, que nos guíe en el desarrollo.
- ✓ Estudiar los riesgos que tiene el proyecto y colaborar para eliminar o atenuar sus efectos.
- ✓ Analizar problemas que se presenten durante el proceso, para poder realizar una mejora continua.
- ✓ Desarrollar un ambiente adecuado de trabajo en equipo basado en la confianza y la comunicación entre los miembros del grupo, lo que ayudará a compartir conocimientos y responsabilidades de manera equitativa.
- ✓ Colaborar mutuamente para mejorar la productividad de cada uno de los miembros y desarrollar sus habilidades, sin importar si estos cuentan con dichas habilidades al comenzar a desempeñar su rol.
- ✓ Mejora la efectividad de los miembros, optimizando el tiempo de desarrollo del producto, sin afectar a la calidad del mismo.

### ***6.1.1.3. Objetivos de Miembro***

Los objetivos que cada miembro se ha comprometido a cumplir durante este ciclo se describen a continuación:

#### **Juan Carlos Fernández Ortiz**

- ✓ Apoyar a todos los miembros del equipo sin que esto signifique absorber sus responsabilidades.
- ✓ Promover una comunicación adecuada entre todos los miembros del equipo.
- ✓ Aprender más sobre las herramientas y el lenguaje que usaremos para el desarrollo, a fin de aprovechar las ventajas que estos ofrecen.
- ✓ Realizar el trabajo de forma efectiva, con el fin de obtener productos de calidad.

- ✓ Hacer lo posible por solucionar los problemas de grupo o al menos disminuir considerablemente sus efectos.

#### **Estefanía Narváez Bastidas**

- ✓ Inyectar el menor número de defectos en el desarrollo de los productos y tratar de reducir constantemente, los que se inyecten, estudiando las posibles causas y soluciones.
- ✓ Cumplir de la forma más estricta posible con el plan que se ha definido y las tareas asignadas.
- ✓ Estar dispuesta a establecer una comunicación abierta con todos los integrantes del grupo, logrando un ambiente de confianza.
- ✓ Ayudar a todos los integrantes con las dudas que se les puedan presentar, para mejorar el trabajo del equipo.

#### **6.1.1.4. Objetivos del Rol**

Los objetivos que cada miembro se ha comprometido en cumplir para desempeñar de mejor manera su rol se describen a continuación:

#### **Líder (Juan Carlos Fernández Ortiz)**

- ✓ Asegurar que todos los miembros del equipo cumplan con la planificación.
- ✓ Escuchar a todos los miembros del equipo y tomar en cuenta sus opiniones para la toma de decisiones.
- ✓ Verificar que todos los miembros del equipo estén realizando una carga de trabajo similar, de manera que se sientan comprometidos con el proyecto.

#### **Administrador de Desarrollo (Estefanía Narváez Bastidas)**

- ✓ Capacitarme en el uso de la metodología y tecnologías que se utilizarán en el desarrollo del producto con el fin de guiar al equipo durante el proceso de desarrollo.
- ✓ Transmitir de forma precisa los requerimientos del sistema a todos los miembros del equipo.

- ✓ Identificar, desarrollar y aprovechar las habilidades y destrezas con las que cuenta cada miembro del equipo.

#### **Administrador de Planificación (Estefanía Narváez Bastidas)**

- ✓ Desarrollar un plan de trabajo, que abarque todas las tareas que se deben cumplir y que se ajuste, en lo posible, a la realidad.
- ✓ Presentar reportes semanales, que nos ayuden a saber cómo está avanzando el equipo y tomar decisiones al respecto, para lograr cumplir con los tiempos límite de entrega
- ✓ Analizar y comprender las formas que corresponden al rol, para guiar al equipo en su utilización, cuando se presenten dudas.
- ✓ Motivar al equipo a seguir la planificación y a hacer un seguimiento de su trabajo correctamente.

#### **Administrador de Calidad (Juan Carlos Fernández Ortiz)**

- ✓ Diseñar estándares que permitan al equipo trabajar de forma ordenada y revisar que se cumpla con estos a fin de obtener un producto de calidad
- ✓ Ayudar al equipo para conseguir una mejora constante en el trabajo de todos sus integrantes.
- ✓ Mantener una comunicación adecuada y oportuna con el líder, que permita informar a tiempo en caso de presentarse problemas.
- ✓ Realizar un control de calidad de todos los productos o artefactos antes de que pasen a línea base.
- ✓ Controlar que todos los reportes de las formas de TSPi se estén llevando adecuadamente.
- ✓ Mantener registro con todos los asuntos tratados en cada reunión.

#### **Administrador de Configuraciones (Juan Carlos Fernández Ortiz)**

- ✓ Ayudar a todos los miembros del equipo a que cuenten con las herramientas adecuadas para realizar su trabajo, así como investigar sobre su uso, para ayudar al equipo en todo lo que sea posible.
- ✓ Vigilar que no se realice ningún cambio en documentos de línea base sin que se haya seguido el proceso adecuado, que se deberá también diseñar.

- ✓ Llevar control de todos los riesgos y asuntos que se presenten para poder darles un mejor seguimiento de manera semanal.
- ✓ Administrar todos los artefactos reusables que puedan ayudar al equipo en fases posteriores y en el siguiente ciclo.

#### ***6.1.1.5. Objetivos del Producto***

##### **Antecedentes**

La Fundación para la Integración del Niño Especial, realiza una evaluación de cada una de las personas con discapacidad que llegan a su centro. Posteriormente elabora un programa individual de trabajo para lo cual las educadoras cuentan con una matriz de habilidades tanto sociales, de autonomía personal, cognitivas y destrezas motrices; a partir de las cuales priorizan las habilidades que se pueden trabajar.

A continuación se determina si esta persona, con un direccionamiento, podría incluirse en el medio laboral.

Una vez finalizado el quimestre, se entrega un informe a los padres de familia o representantes de las personas con discapacidad, en el cual se resume el trabajo realizado y los logros en cuanto a las habilidades que se planteó desarrollar.

Todo este proceso se lleva a cabo manualmente y posteriormente es pasado a un editor de texto, lo cual genera pérdidas de datos y dificulta el análisis de indicadores en el tiempo.

##### **Funcionalidades**

- ✓ **F0:** El sistema permitirá ingresar mediante usuario y contraseña.
- ✓ **F1:** El sistema permitirá administrar usuarios.
- ✓ **F2:** El sistema permitirá administrar perfiles.
- ✓ **F3:** El sistema permitirá administrar estudiantes.
- ✓ **F4:** El sistema permitirá administrar quimestres.
- ✓ **F5:** El sistema permitirá administrar habilidades principales.

- ✓ **F6:** El sistema permitirá administrar habilidades específicas.
- ✓ **F7:** El sistema permitirá administrar tareas.
- ✓ **F8:** El sistema permitirá administrar talleres.
- ✓ **F9:** El sistema permitirá administrar evaluaciones iniciales.
- ✓ **F10:** El sistema permitirá administrar evaluaciones mensuales.
- ✓ **F11:** El sistema permitirá obtener reportes de las evaluaciones a los estudiantes.

#### ***6.1.1.6. Objetivos de Calidad***

Desarrollar un producto de calidad, con el menor número de defectos posibles evitando sobrepasar los defectos estimados que se presentan a continuación:

Ciclo	Defectos Estimados
Primero	94
Segundo	80

**Tabla 2: Objetivos de Calidad del Producto**

#### **6.1.2. Estrategia de Desarrollo**

TSPi propone utilizar una estrategia basada en ciclos, que debe estar formada por al menos dos, basados en un diseño modular. En cada uno de estos ciclos se debe presentar una versión del producto, que será integrada a las versiones subsecuentes.

##### ***6.1.2.1. Criterios de Estrategia***

El presente proyecto se desarrollará en dos ciclos; debido a que se tiene un contacto cercano con el cliente, existe cierta documentación sobre el proceso a automatizar y el dominio del problema no presenta una complejidad considerable.

El producto que se desarrollará será orientado a la web, se utilizarán las tecnologías Java y MySQL debido a las ventajas ya mencionadas en el fundamento teórico de este trabajo.

Las funcionalidades del sistema serán repartidas en 2 ciclos de la siguiente manera:

Ciclos	Funcionalidades
Primero	F0: Ingresar mediante usuario y contraseña. F1: Administrar usuarios. F2: Administrar perfiles. F3: Administrar estudiantes. F4: Administrar quimestres. F5: Administrar habilidades principales. F6: Administrar habilidades específicas. F7: Administrar tareas. F8: Administrar talleres.
Segundo	F9: Administrar evaluaciones iniciales. F10: Administrar evaluaciones mensuales. F11: Obtener reportes de las evaluaciones a los estudiantes.

**Tabla 3: Distribución de Funcionalidades por Ciclo**

### ***6.1.2.2. Estrategia de Reuso***

En el primer ciclo se reusarán algunos componentes debido a que esta práctica incrementa la productividad, reduce la complejidad, mejora la interoperabilidad, reduce costos y mejora la calidad de los sistemas ya que estos componentes han sido probados y cumplen con estándares altos de calidad.

Los componentes que se han identificado como aptos para la reutilización son los siguientes:

- ✓ Documentos de Lanzamiento
  - Gestión de Configuración del Software.

- Estándares de Documentación y Comunicación.
- ✓ Documento de Implementación
  - Estándares de codificación.

### 6.1.3. Planificación

Una adecuada planificación ayuda a dimensionar el producto y todas las actividades necesarias para terminarlo, permitiendo conocer el avance del producto con el fin de detectar problemas de manera temprana.

Un plan de trabajo debe ser balanceado, es decir, todos los miembros del equipo deben tener una carga de trabajo similar para que éste no se retrase. Es importante que todos los miembros del equipo se comprometan con el plan y estén conscientes de que deben seguirlo para llegar al objetivo anhelado.

#### 6.1.3.1. Distribución de Módulos

Los módulos que debe desarrollarse durante el presente ciclo se han distribuido entre los miembros del equipo de la siguiente forma:

Módulo	Responsable
F0: Ingresar mediante usuario y contraseña.	Juan Carlos Fernández Ortiz
F1: Administrar usuarios.	
F2: Administrar perfiles.	
F3: Administrar estudiantes.	
F4: Administrar quimestres.	Estefanía Narváez
F5: Administrar habilidades principales.	
F6: Administrar habilidades específicas.	
F7: Administrar tareas.	

Módulo	Responsable
F8: Administrar talleres.	

**Tabla 4: Distribución de Módulos. Ciclo I.**

#### **6.1.4. Definición de Requerimientos**

En esta fase se especifican los requerimientos del usuario con respecto al producto final. A continuación se presenta el documento de especificación de requerimientos del sistema.

#### ***Especificación de Requerimientos del Sistema (SRS)***

### **I Introducción**

#### **1.1.Objetivo**

El presente documento contiene todos los requerimientos de forma clara, precisa y no ambigua, con la finalidad de ser revisado y aceptado por el cliente, para verificar que el producto cumplirá con sus necesidades. Este documento constituye la base sobre la cual se llevarán a cabo las siguientes fases para el desarrollo del producto.

#### **1.2.Definición General del Sistema**

La Fundación para la Integración del Niño Especial, en su proyecto CEDIN - La Dolorosa (Centro de Desarrollo Integral - La Dolorosa), cuenta con profesionales que planifican y ejecutan actividades, a través de las cuales se fomenta el desarrollo de las personas con discapacidad en diferentes ámbitos. El presente sistema, pretende automatizar el seguimiento del desarrollo de las habilidades en las personas con discapacidad mental, optimizando el proceso pedagógico.

Para el ciclo actual se implementarán las siguientes funcionalidades:

**a) Ingresar mediante usuario y contraseña**

Esta funcionalidad incorpora seguridad en el acceso al sistema. Restringe el acceso sólo a usuarios que cuenten con un nombre de usuario y una contraseña.

**b) Administrar usuarios**

Esta funcionalidad permitirá la creación, modificación y eliminación de los usuarios, así como la asignación de perfiles.

**c) Administrar perfiles**

Esta funcionalidad permitirá la creación, modificación y eliminación de perfiles y privilegios.

**d) Administrar estudiantes**

Esta funcionalidad permitirá la creación, modificación y eliminación de estudiantes, sobre los cuales se registrará la información sobre desarrollo de habilidades.

**e) Administrar quimestres**

Esta funcionalidad permitirá la creación, modificación y eliminación de quimestres, que son la unidad de tiempo en la que se desempeña un ciclo de trabajo en el CEDEIN - La Dolorosa de FINE.

**f) Administrar habilidades principales**

Esta funcionalidad permitirá la creación, modificación y eliminación de habilidades principales, las cuales engloban de manera general a las habilidades específicas.

**g) Administrar habilidades específicas**

Esta funcionalidad permitirá la creación, modificación y eliminación de habilidades específicas, que corresponden a destrezas sociales, motrices, entre otras.

**h) Administrar tareas**

Esta funcionalidad permitirá la creación, modificación y eliminación de tareas, las cuales ayudan a desarrollar habilidades específicas y a evaluar el progreso de las personas con discapacidad en destrezas específicas.

**i) Administrar talleres**

Esta funcionalidad permitirá la creación, modificación y eliminación de talleres que constituyen áreas de trabajo más avanzadas, que buscan desarrollar habilidades y

capacitar a las personas con discapacidad con el objetivo de que puedan ser insertadas en un medio laboral, además constituyen talleres ocupacionales.

### 1.3. Información de Equipo

A continuación se presenta información básica sobre el equipo de trabajo.

Integrante	Rol	Distribución de Módulos
Juan Carlos Fernández	Líder. Administrador de Calidad. Administrador de Configuraciones.	<b>F0:</b> Ingresar mediante usuario y contraseña. <b>F1:</b> Administrar usuarios. <b>F2:</b> Administrar perfiles. <b>F3:</b> Administrar estudiantes.
Estefanía Narváez	Administrador de Planificación. Administrador de Desarrollo.	<b>F4:</b> Administrar quimestres. <b>F5:</b> Administrar habilidades principales. <b>F6:</b> Administrar habilidades específicas. <b>F7:</b> Administrar tareas. <b>F8:</b> Administrar talleres.

Tabla 5: Información del Equipo. Ciclo I.

## II Requerimientos Funcionales

### 2.1. Diagrama de Casos de Uso

#### 2.1.1. Diagrama de Casos de Uso General

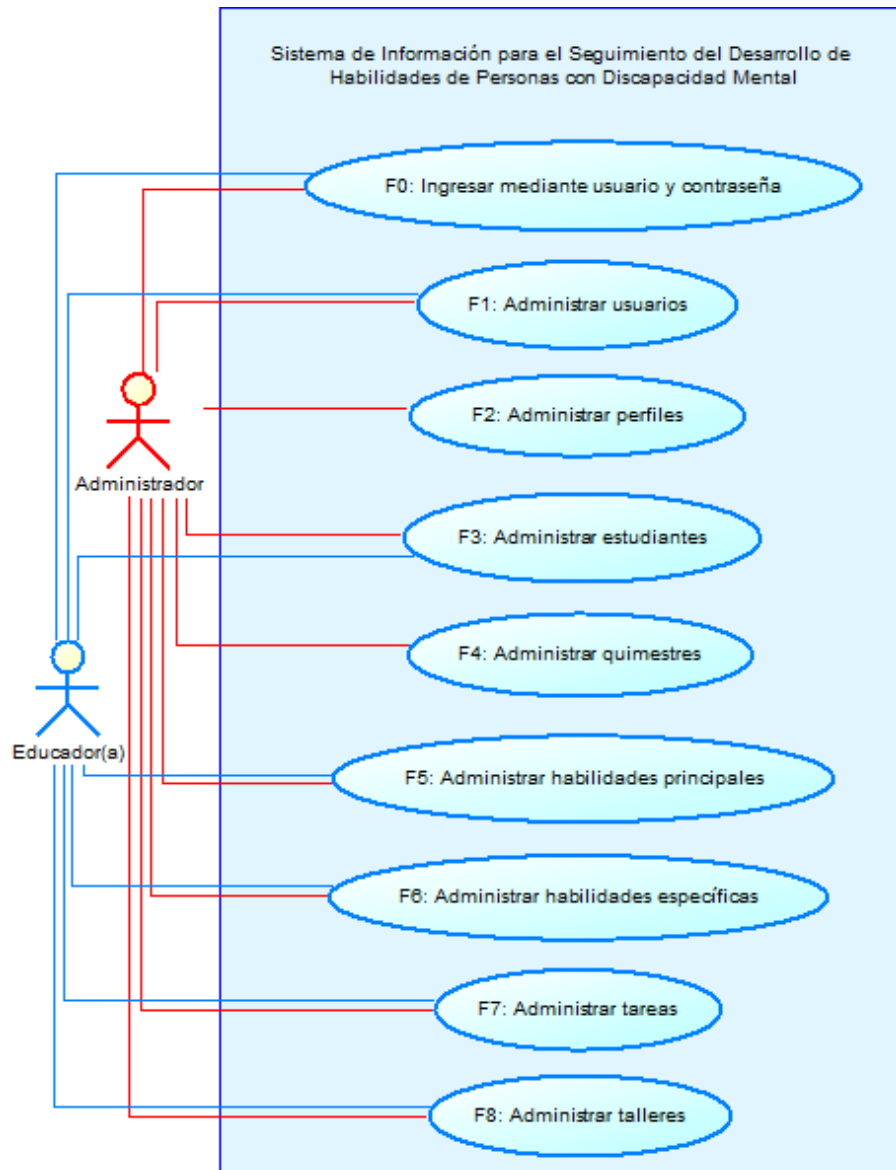


Ilustración 9: Diagrama de Casos de Uso General

## 2.1.2. Diagrama de Casos de Uso Siguiete Nivel

### 2.1.2.1. F0: Ingresar mediante usuario y contraseña.



Figura 10: Diagrama de Casos de Uso Siguiete Nivel de F0

### 2.1.2.2. F1: Administrar usuarios.

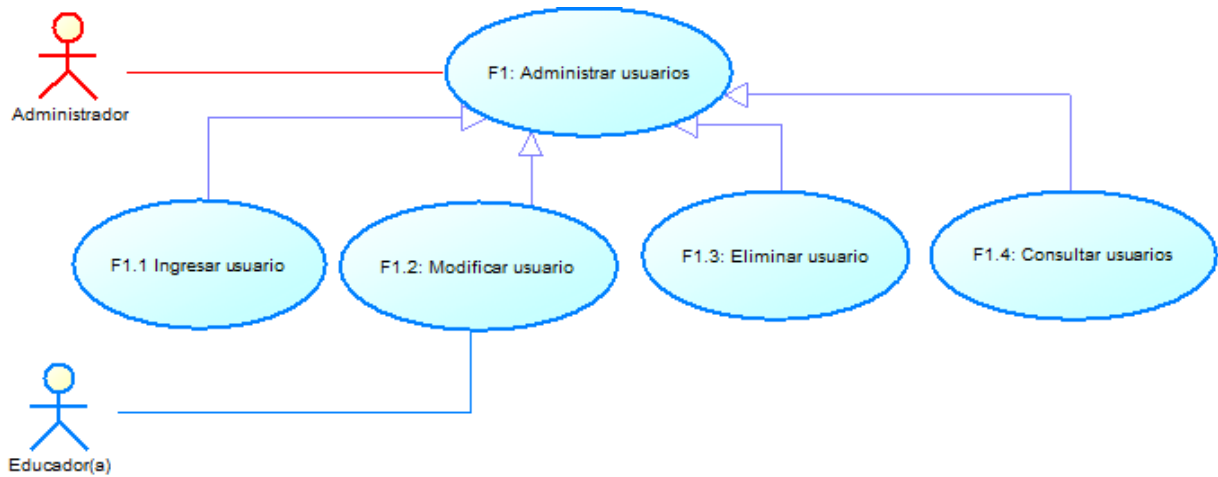
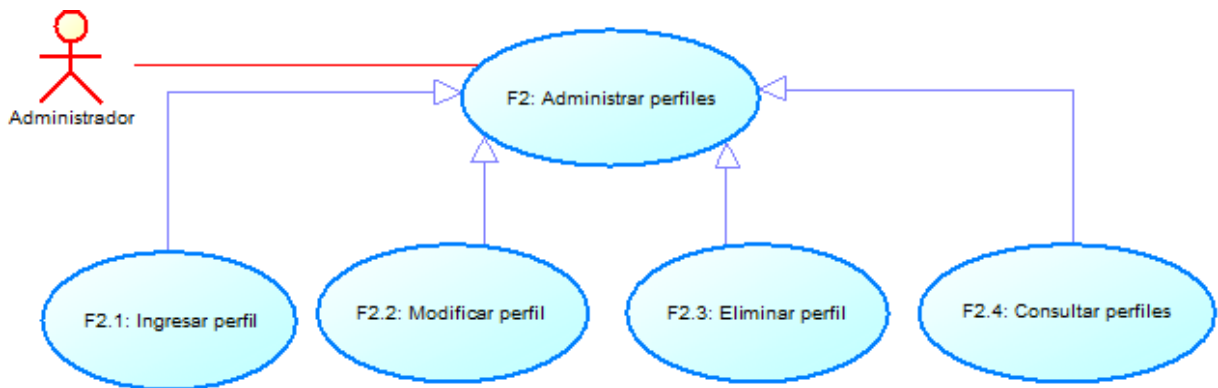


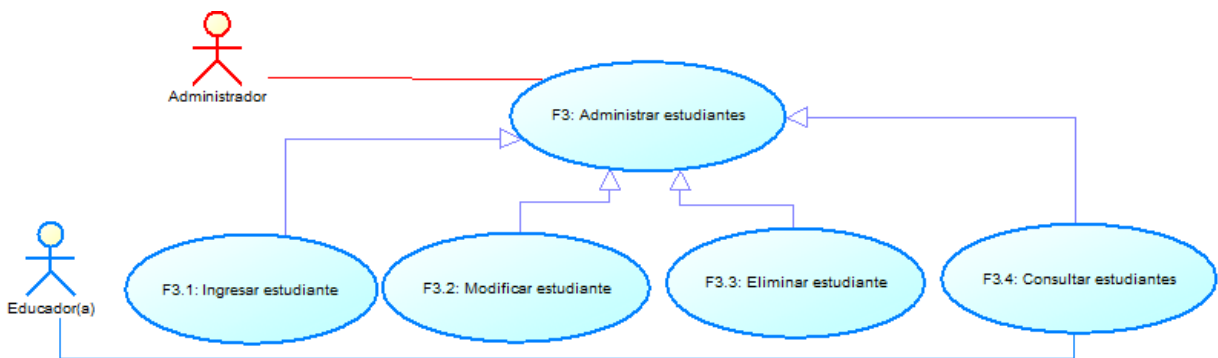
Ilustración 11: Diagrama de Casos de Uso Siguiete Nivel de F1

**2.1.2.3. F2: Administrar perfiles.**



**Ilustración 12: Diagrama de Casos de Uso Siguiete Nivel de F2**

**2.1.2.4. F3: Administrar estudiantes.**



**Ilustración 13: Diagrama de Casos de Uso Siguiete Nivel de F3**

2.1.2.5. *F4: Administrar quimestres.*

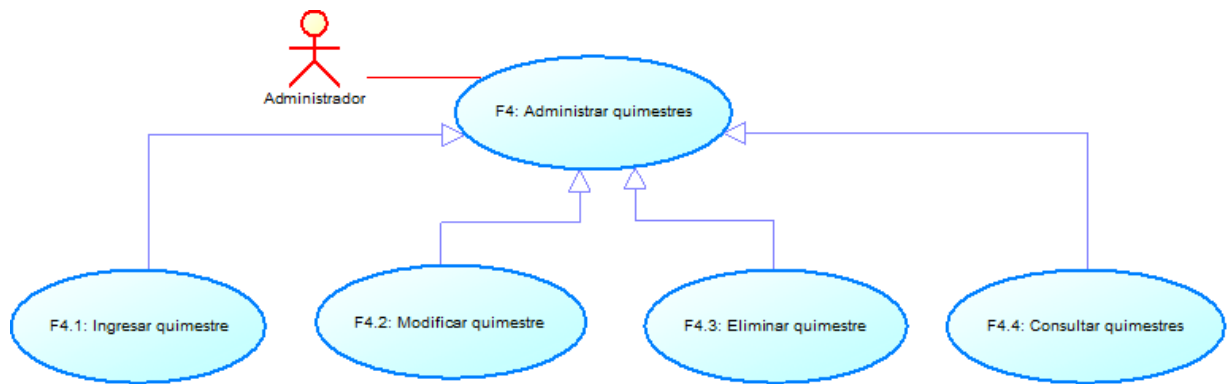


Ilustración 14: Diagrama de Casos de Uso Siguiete Nivel de F4

2.1.2.6. *F5: Administrar habilidades principales.*

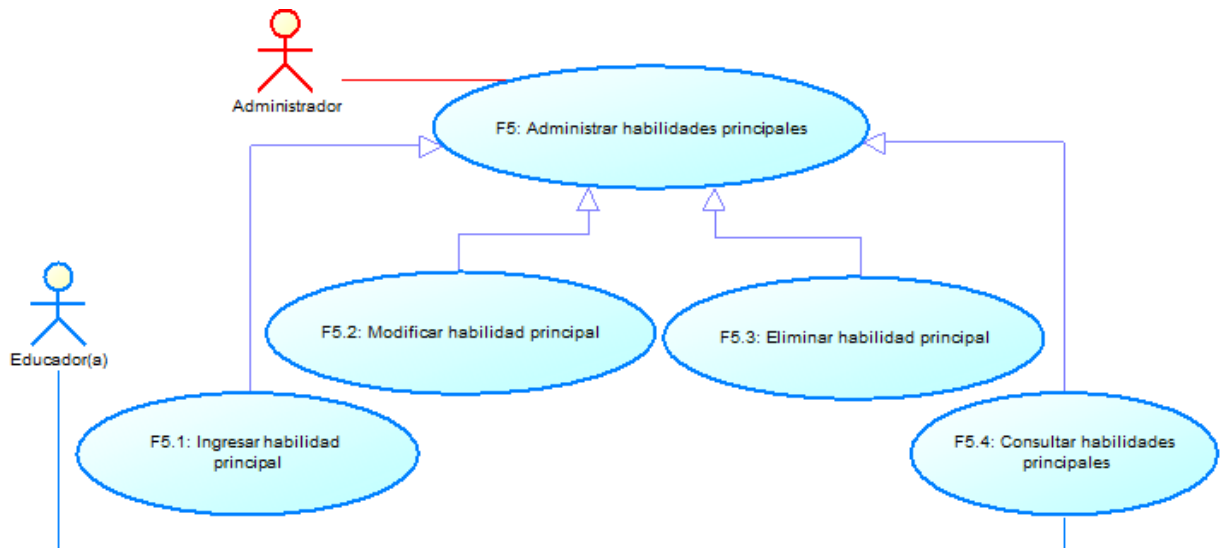
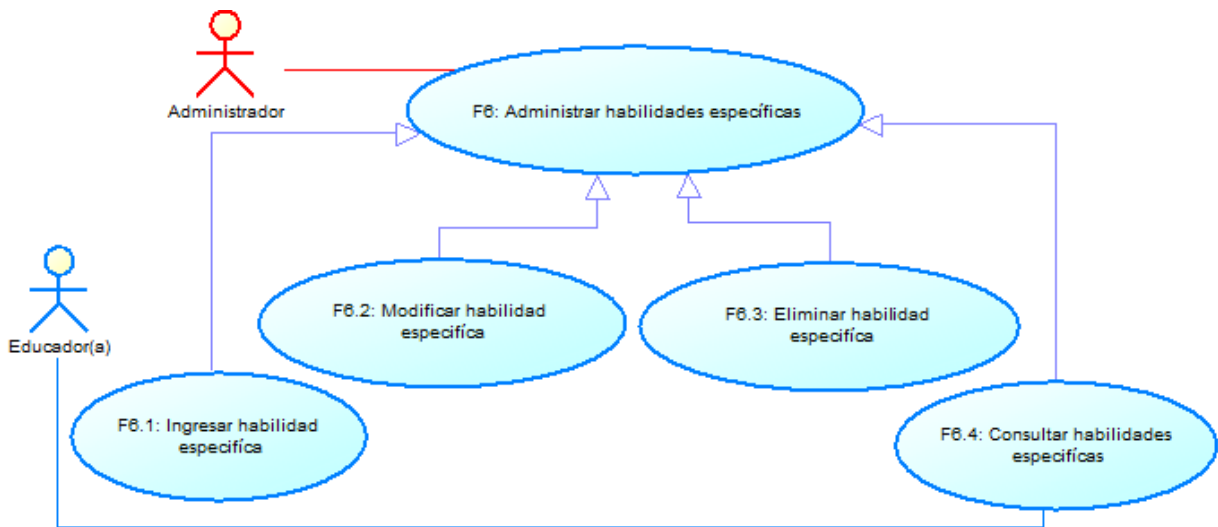


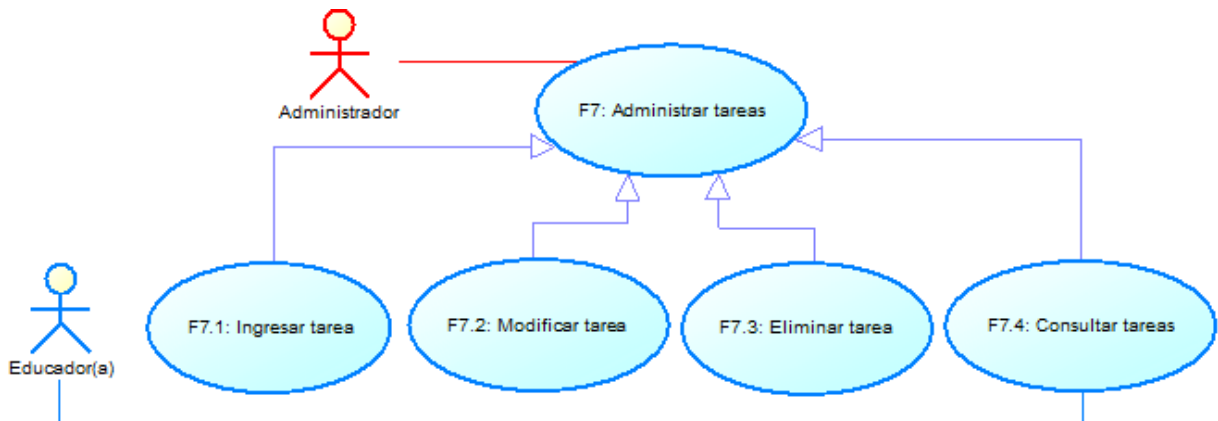
Ilustración 15: Diagrama de Casos de Uso Siguiete Nivel de F5

**2.1.2.7. F6: Administrar habilidades específicas.**



**Ilustración 16: Diagrama de Casos de Uso Siguiendo Nivel de F6**

**2.1.2.8. F7: Administrar tareas.**



**Ilustración 17: Diagrama de Casos de Uso Siguiendo Nivel de F7**

### 2.1.2.9. F8: Administrar talleres.

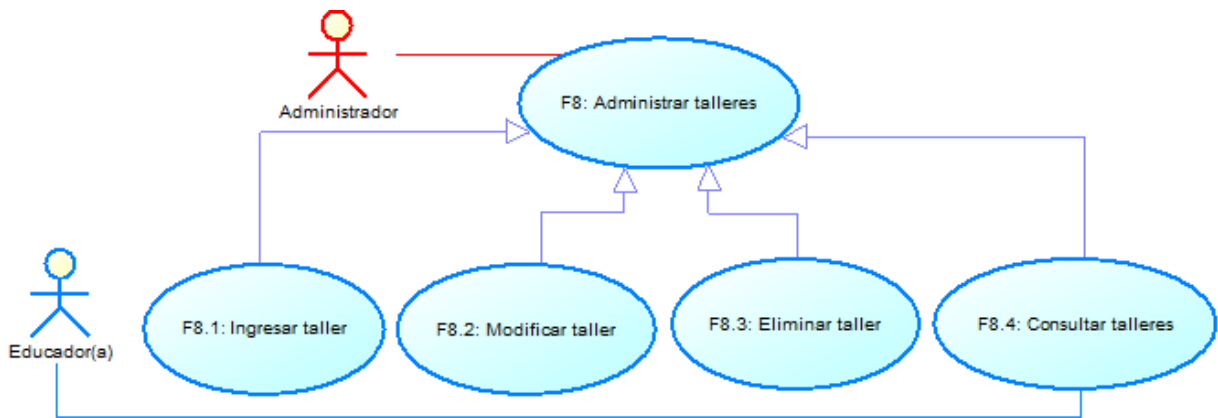


Ilustración 18: Diagrama de Casos de Uso Siguiendo Nivel de F8

### 2.1.3. Diagramas de Casos de Uso A Detalle

A continuación se presentan los casos de uso a detalle de las funcionalidades F3: Administración de Estudiantes y F8: Administración de Talleres. Los casos de uso a detalle de las funcionalidades restantes se encuentran en el anexo “Casos de Uso a Detalle” del primer ciclo.

#### 2.1.3.1. F3.1: Ingresar estudiante.

**Descripción:** Función que permite registrar la información de un estudiante.

**Actores:** Administrador.

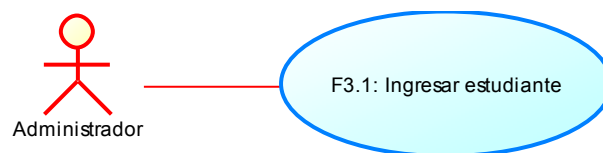


Ilustración 19: Diagrama de Casos de Uso a Detalle de F3.1

### Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Estudiantes del menú principal.	2	Presenta una página que permite la administración de estudiantes.	E1
3	Ingresa la información del estudiante.			
4	Presiona el botón Guardar.	7	Valida la información.	
		8	Almacena la información.	E1

Tabla 6: Flujo Principal de F3.1

### Flujo Alternativo:

Paso	Caso de Uso
7	F3.2 ó F3.3

Tabla 7: Flujo Alternativo de F3.1

### Excepciones:

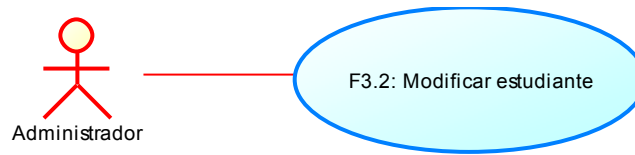
Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 8: Excepciones de F3.1

#### 2.1.3.2. F3.2: Modificar estudiante.

**Descripción:** Función que permite modificar la información de un estudiante.

**Actores:** Administrador.



**Ilustración 20: Diagrama de Casos de Uso a Detalle de F3.2**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Estudiantes del menú principal.	2	Presenta una página que permite la administración de estudiantes.	E1
3	Selecciona el estudiante a modificar.	4	Carga la información del estudiante seleccionado.	E1
5	Modifica la información			
6	Presiona el botón Guardar.	7	Valida la información.	
		8	Almacena la información.	E1

**Tabla 9: Flujo Principal de F3.2**

**Flujo Alternativo:**

Paso	Caso de Uso
3	F3.1

**Tabla 10: Flujo Alternativo de F3.2**

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con	Refrescar la página. Si el problema

Código	Mensaje	Alternativa
	la BDD.	persiste, contacte al departamento técnico.

Tabla 11: Excepciones de F3.2

**2.1.3.3. F3.3: Eliminar estudiante.**

**Descripción:** Función que permite desactivar un estudiante.

**Actores:** Administrador.

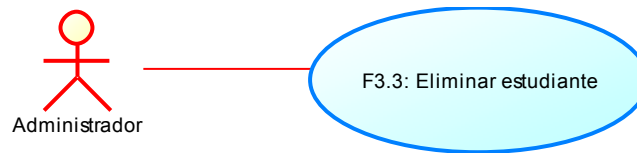


Ilustración 21: Diagrama de Casos de Uso a Detalle de F3.3

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Estudiantes del menú principal.	2	Presenta una página que permite la administración de estudiantes.	E1
3	Selecciona el estudiante a eliminar.	4	Elimina el estudiante.	E1

Tabla 12: Flujo Principal de F3.3

**Flujo Alternativo:**

Paso	Caso de Uso
3	F3.1

**Tabla 13: Flujo Alternativo de F3.3**

**Excepciones:**

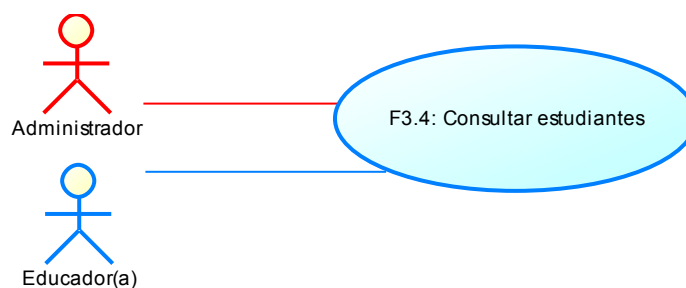
Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

**Tabla 14: Excepciones de F3.3**

**2.1.3.4. F3.4: Consultar estudiantes.**

**Descripción:** Función que permite consultar estudiantes.

**Actores:** Administrador, Educador(a).



**Ilustración 22: Diagrama de Casos de Uso a Detalle de F3.4**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Estudiantes del menú principal.	2	Presenta una página con los estudiantes.	E1

Tabla 15: Flujo Principal de F3.4

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 16: Excepciones de F3.4

**2.1.3.5. F8.1: Ingresar taller.**

**Descripción:** Función que permite registrar la información de un taller.

**Actores:** Administrador.

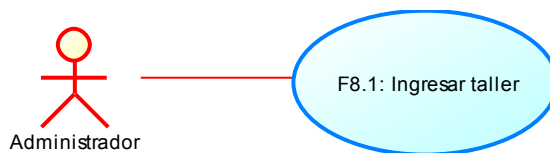


Ilustración 23: Diagrama de Casos de Uso a Detalle de F8.1

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción	2	Presenta una página que	E1

Paso	Actor	Paso	Sistema	Excepción
	Administrar Talleres del menú principal.		permite la administración de talleres.	
<b>3</b>	Ingresa nombre del taller.			
<b>4</b>	Seleccionar las habilidades correspondientes al taller.			
<b>5</b>	Presiona el botón Guardar.	<b>6</b>	Valida la información.	
		<b>7</b>	Almacena la información.	E1

**Tabla 17: Flujo Principal de F8.1**

**Flujo Alternativo:**

Paso	Caso de Uso
<b>6</b>	F8.2 ó F8.3

**Tabla 18: Flujo Alternativo de F8.1**

**Excepciones:**

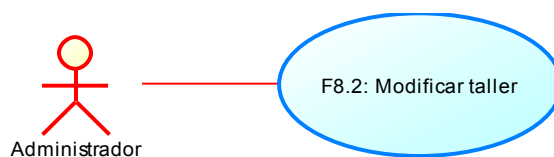
Código	Mensaje	Alternativa
<b>E1</b>	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

**Tabla 19: Excepciones de F8.1**

**2.1.3.6. F8.2: Modificar taller.**

**Descripción:** Función que permite modificar la información de un taller.

**Actores:** Administrador.



**Ilustración 24: Diagrama de Casos de Uso a Detalle de F8.2**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Talleres del menú principal.	2	Presenta una página para la administración de talleres.	E1
3	Selecciona el taller que desea modificar.	4	Carga la información del taller.	E1
5	Modifica la información.			
6	Presiona el botón Guardar.	7	Valida la información.	
		8	Almacena la información.	E1

**Tabla 20: Flujo Principal de F8.2**

**Flujo Alternativo:**

Paso	Caso de Uso
3	F8.1

**Tabla 21: Flujo Alternativo de F8.2**

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 22: Excepciones de F8.2

**2.1.3.7. F8.3: Eliminar taller.**

**Descripción:** Función que permite desactivar un taller.

**Actores:** Administrador.

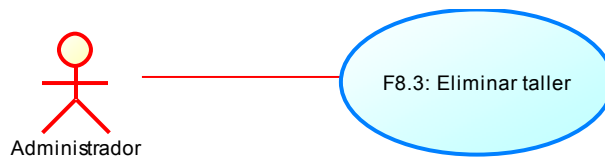


Ilustración 25: Diagrama de Casos de Uso a Detalle de F8.3

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Talleres del menú principal.	2	Presenta una página para la administración de talleres.	E1
3	Selecciona la opción eliminar del taller que desee.	4	Elimina el taller.	E1

Tabla 23: Flujo Principal de F8.3

**Flujo Alternativo:**

Paso	Caso de Uso
3	F8.1

Tabla 24: Flujo Alternativo de F8.3

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 25: Excepciones de F8.3

**2.1.3.8. F8.4: Consultar talleres.**

**Descripción:** Función que permite consultar los talleres.

**Actores:** Administrador, Educador(a).

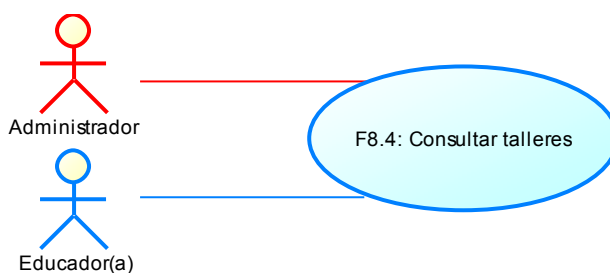


Ilustración 26: Diagrama de Casos de Uso a Detalle de F8.4

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Talleres del menú principal.	2	Presenta una página con los talleres ingresados.	E1

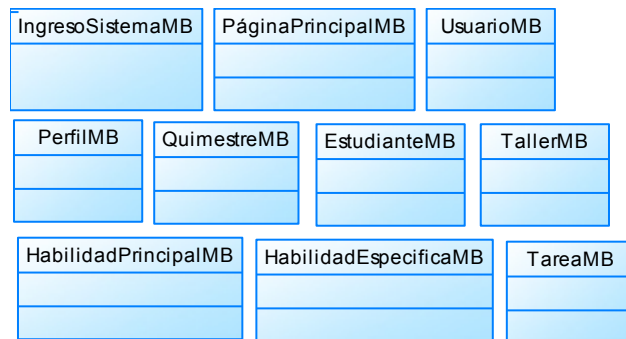
**Tabla 26: Flujo Principal de F8.4**

**Excepciones:**

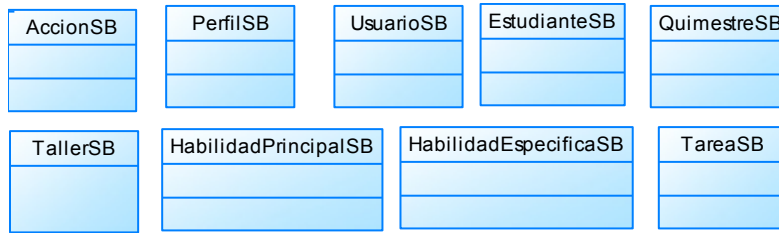
Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Actualizar la aplicación. Si el problema persiste, contacte al departamento técnico.

**Tabla 27: Excepciones de F8.4**

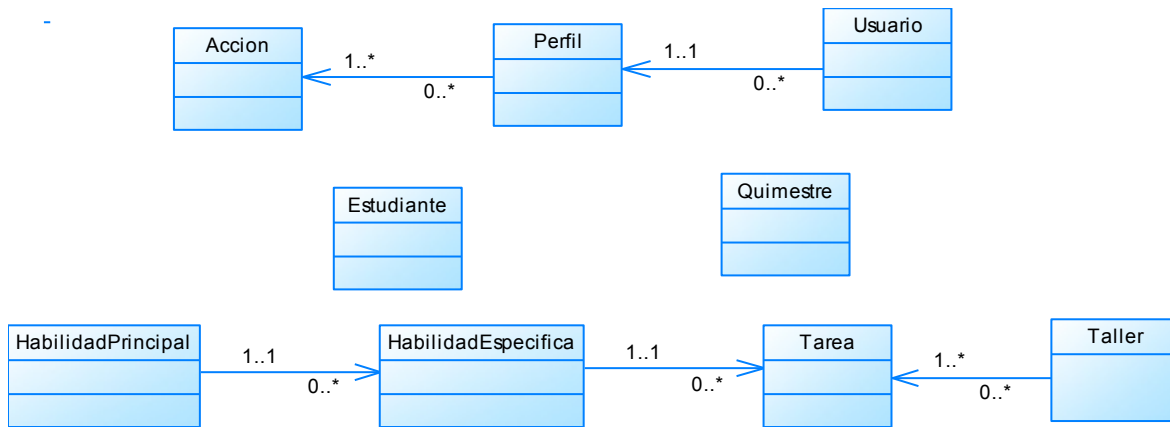
**2.2.Diagramas de Clases a Nivel Conceptual**



**Ilustración 27: Diagrama de clases conceptual Managed Beans**



**Ilustración 28: Diagrama de clases conceptual Session Beans**



**Ilustración 29: Diagrama de clases conceptual Entities**

### III Requerimientos no Funcionales

#### 3.1. Restricciones de Diseño e Implementación

**Metodología de Diseño:** Orientada a Objetos

**Arquitectura:** EJB

**Lenguaje de Programación:** Java (Orientado a web)

**Sistema Gestor de Bases de Datos:** MySQL

**Servidor de Aplicaciones:** JBoss

**Estándares de Codificación:** Basados en convenciones de código para el lenguaje Java

## IV Otros Requerimientos

### 4.1.De Interface Externa

La interface externa se desarrollará de acuerdo al siguiente diseño:



Ilustración 30: Página Principal



Ilustración 31: Administración de Perfiles

**Fundación para la Integración del Niño Especial**

SISDHAB

Inicio / Administración / Ayuda

### Administración de Usuarios

Cédula:

Apellidos:

Nombres:

Cargo:

Perfil:

E-mail institucional:

E-mail alternativo:

Teléfono Convencional:

Teléfono Móvil:

Contraseña:

Confirmar contraseña:

Cedula	Apellidos	Nombres	Acciones
1234567890	Fernández Ortiz	Juan Carlos	

©2014 Fundación para la Integración del Niño Especial

**Ilustración 32: Administración de Usuarios**

## 4.2.De Hardware

**RAM (mínima): 2 GB**

**RAM (recomendada): 4 GB**

**Procesador: Intel Pentium IV de 2,6 GHz o equivalente.**

**Espacio en Libre en Disco (mínimo): 500 MB**

**Espacio en Libre en Disco (recomendado): 1 GB**

### 4.3.De Software

Navegador de Internet:

- ✓ Mozilla Firefox: versión 27 o superior,
- ✓ Google Chrome: versión 33 o superior, o
- ✓ Internet Explorer: versión 11 o superior.

JDK 1.6

JBoss 7.1

MySQL 5.6

### V Plan de Pruebas del Sistema

Las pruebas del sistema, son similares a las pruebas de caja negra, es decir, que se enfocan en los requerimientos establecidos y en las funcionalidades, pero éstas lo hacen a nivel de resultados, como un todo, abarcando las partes combinadas del mismo.

Su principal objetivo, es verificar que el sistema cumpla con los requerimientos, por lo que es sumamente importante que éstos sean documentados y medibles.

A continuación se presenta el plan de pruebas del sistema para las funcionalidades F3: Administración de Estudiantes y F8: Administración de Talleres. El plan de pruebas del sistema que abarca todas las funcionalidades se encuentran en el anexo “Plan de Pruebas del Sistema” del primer ciclo.

#### Prueba 10:

- ✓ **Funcionalidad:** F3.1
- ✓ **Evaluador:** Estefanía Narváez
- ✓ **Descripción:** Evaluar el correcto funcionamiento del ingreso de estudiantes.
- ✓ **Diseño:**
  1. Ingresar un estudiante, llenando todos los campos obligatorios.
  2. Ingresar un estudiante, dejando en blanco los campos obligatorios.

3. Ingresar un estudiante existente.

✓ **Resultados Esperados:**

1. El sistema almacena correctamente la información ingresada.
2. El sistema emite un mensaje por cada campo obligatorio que se deje vacío después de presionar guardar. Presenta el mensaje de alerta, hasta que la información esté completa.
3. El sistema debe indicar que el estudiante que se desea ingresar ya existe.

**Prueba 11:**

✓ **Funcionalidad:** F3.2

✓ **Evaluador:** Estefanía Narváez

✓ **Descripción:** Evaluar el correcto funcionamiento de la modificación de estudiantes.

✓ **Diseño:**

1. Modificar un estudiante existente.
2. Modificar un estudiante borrando campos obligatorios.

✓ **Resultados Esperados:**

1. El sistema almacena correctamente los campos, sin alterar el código único de la entidad.
2. El sistema almacenará con datos predefinidos en los campos obligatorios.

**Prueba 12:**

✓ **Funcionalidad:** F3.3

✓ **Evaluador:** Estefanía Narváez

✓ **Descripción:** Evaluar el correcto funcionamiento de la eliminación de estudiantes.

✓ **Diseño:**

1. Eliminar un estudiante existente.

✓ **Resultados Esperados:**

1. El estudiante se elimina correctamente.

**Prueba 13:**

✓ **Funcionalidad:** F3.4

✓ **Evaluador:** Estefanía Narváez.

✓ **Descripción:** Evaluar el correcto funcionamiento de la consulta de estudiantes.

✓ **Diseño:**

1. Consultar de manera general los registros ingresados

✓ **Resultados Esperados:**

1. El sistema presenta los registros que han sido ingresados al sistema.

**Prueba 30:**

✓ **Funcionalidad:** F8.1

✓ **Evaluador:** Juan Carlos Fernández

✓ **Descripción:** Evaluar el correcto funcionamiento del ingreso de talleres.

✓ **Diseño:**

1. Ingresar un taller, llenando todos los campos obligatorios.
2. Ingresar un taller, dejando en blanco los campos obligatorios.

✓ **Resultados Esperados:**

1. El sistema almacena correctamente la información ingresada.
2. El sistema emite un mensaje por cada campo obligatorio que se deje vacío después de presionar guardar. Presenta el mensaje de alerta, hasta que la información esté completa.

### **Prueba 31:**

- ✓ **Funcionalidad:** F8.2
- ✓ **Evaluador:** Juan Carlos Fernández
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la modificación de talleres.
- ✓ **Diseño:**
  1. Modificar un taller existente.
  2. Modificar un taller borrando campos obligatorios.
- ✓ **Resultados Esperados:**
  1. El sistema almacena correctamente los campos, sin alterar el código único de la entidad.
  2. El sistema almacenará con datos predefinidos en los campos obligatorios.

### **Prueba 32:**

- ✓ **Funcionalidad:** F8.3
- ✓ **Evaluador:** Juan Carlos Fernández
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la eliminación de taller.
- ✓ **Diseño:**
  1. Eliminar un taller existente.
- ✓ **Resultados Esperados:**
  1. El registro se elimina correctamente.

### **Prueba 33:**

- ✓ **Funcionalidad:** F8.4
- ✓ **Evaluador:** Juan Carlos Fernández.
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la consulta de talleres.

✓ **Diseño:**

1. Consultar de manera general los registros ingresados

✓ **Resultados Esperados:**

1. El sistema presenta los registros que han sido ingresados al sistema.

### 6.1.5. Diseño

A continuación se presenta el producto generado en la fase de diseño del primer ciclo:

#### *Especificación de Diseño del Sistema (SDS)*

##### 1. Diseño Arquitectónico

La arquitectura de software es una especificación de los subsistemas y de la forma en la que se relacionan entre ellos. Es una estructura jerárquica de los componentes que especifica la manera en que interactúan dichos componentes y la estructura de datos que utilizan.

##### 1.1.Arquitectura EJB

En la definición de una arquitectura se deben tener en cuenta los conceptos de cohesión y acoplamiento. Cohesión nos indica la facilidad con que un componente puede unirse con otros, mientras que el acoplamiento, es responsable de evaluar las relaciones que se establecen entre las partes; de tal forma que un nivel bajo de acoplamiento permitiría la reutilización de código.

Como muchas tecnologías, Enterprise Java Beans separa 3 conceptos fundamentales que son:

- ✓ **Capa de Presentación o Interfaz de Usuario:** Está formada por las páginas con las que interactúa directamente el usuario y los controles que se encuentran en dichas páginas.

- ✓ **Capa de Negocio:** Está formada por los session beans que representan objetos que van a ser gestionados por la aplicación. Contiene el código al que recurre la capa de presentación para recuperar los datos deseados.
- ✓ **Capa de Acceso a Datos:** Contiene clases que interactúan con la base de datos y permiten realizar todas las operaciones de forma transparente para la segunda capa.

## 1.2. Diagrama de Paquetes

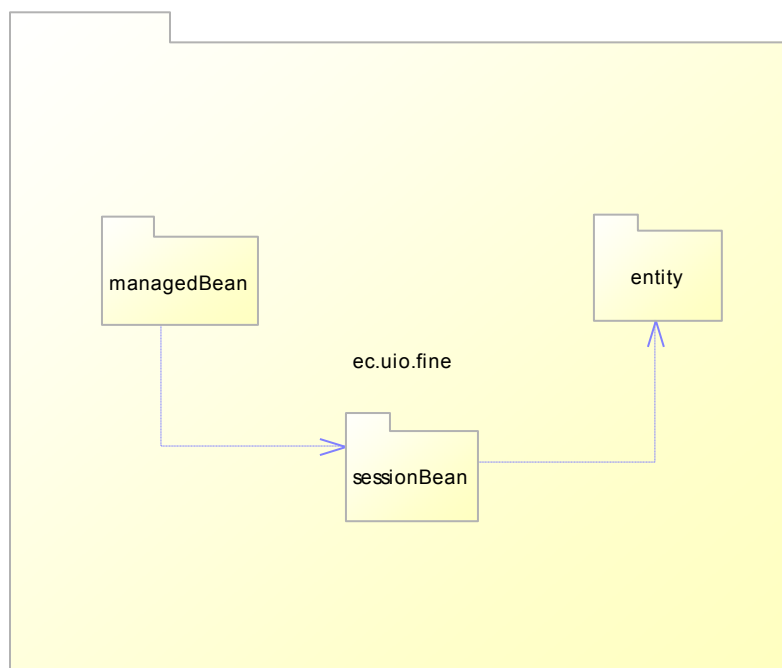


Ilustración 33: Diagrama de paquetes

## 2. Diagramas de Clases

### 2.1. Clases de Managed Beans

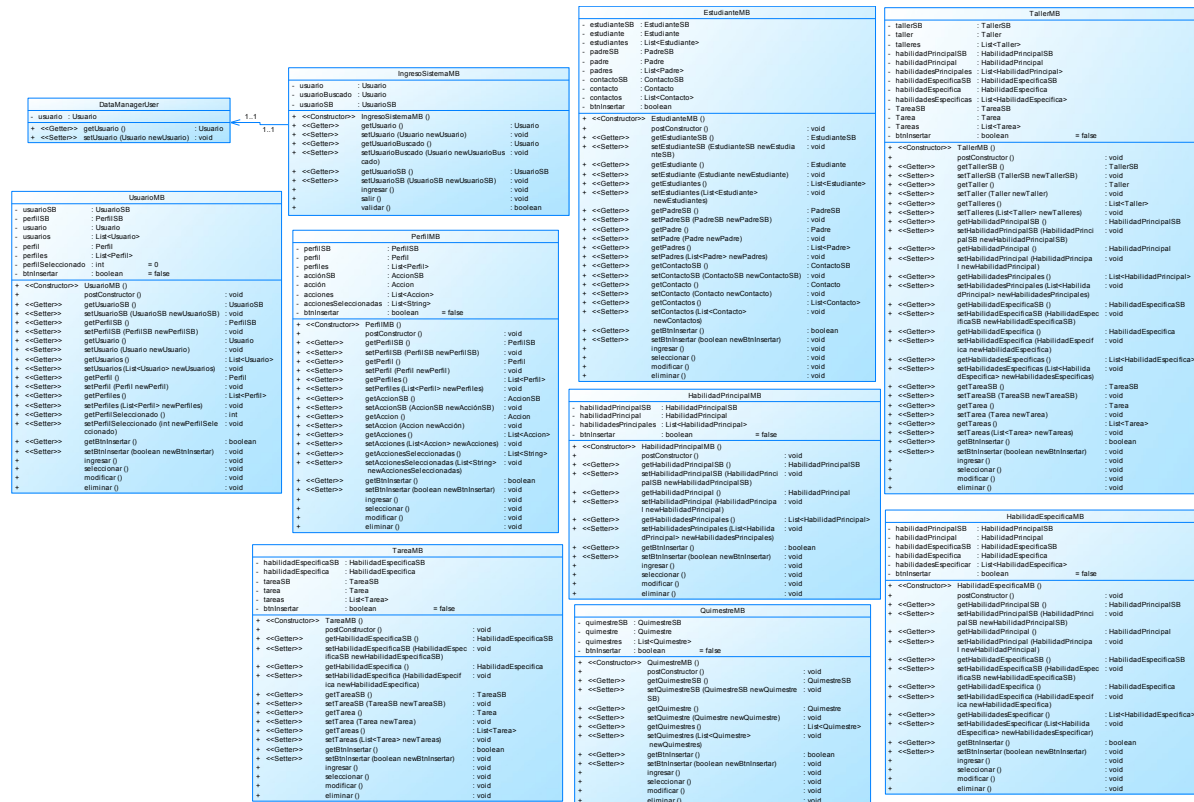


Ilustración 34: Diagrama de Clases de Managed Beans a Detalle

## 2.2. Clases de Session Beans

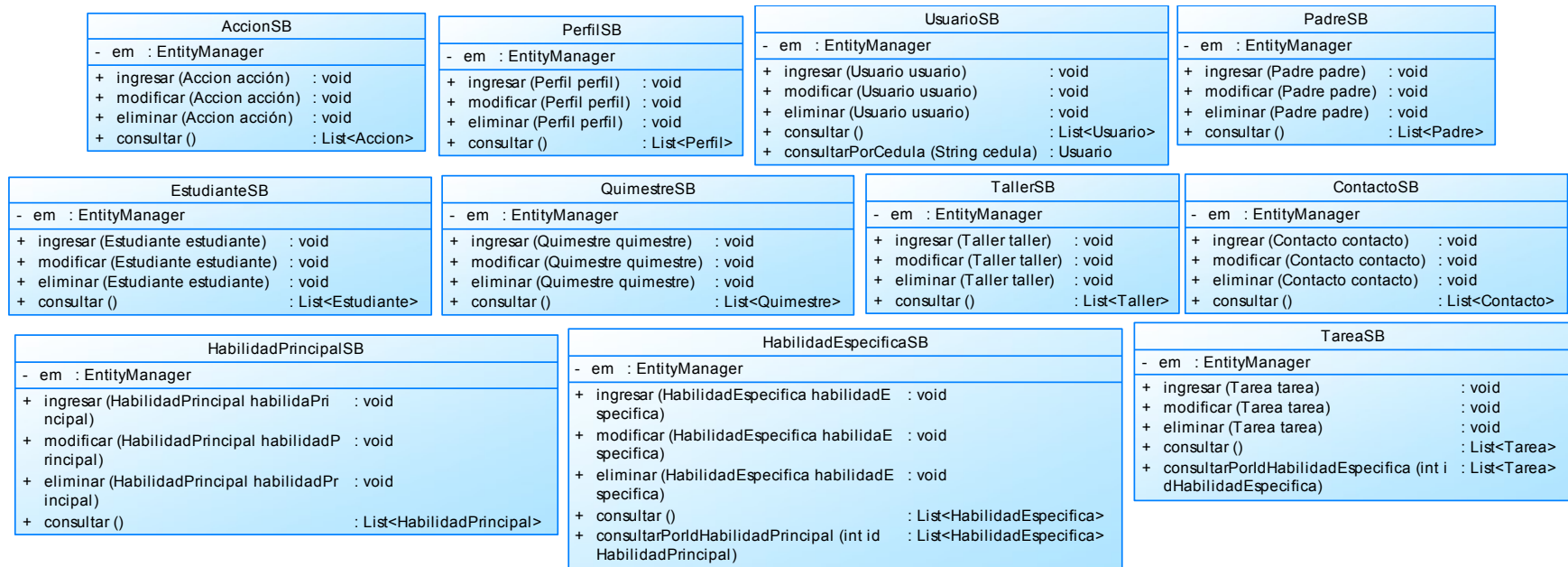


Ilustración 35: Diagrama de Clases de Session Beans a Detalle

## 2.3. Clases de Entities



Ilustración 36: Diagrama de Clases de Entidades a Detalle

### 3. Diagramas de Secuencia

A continuación se presentan los diagramas de secuencia de las funcionalidades F3: Administración de Estudiantes y F8: Administración de Talleres. Los diagramas de secuencia de las funcionalidades restantes se encuentran en el anexo “Diagramas de Secuencia” del primer ciclo.

#### 3.1.F3.1: Ingresar estudiante.

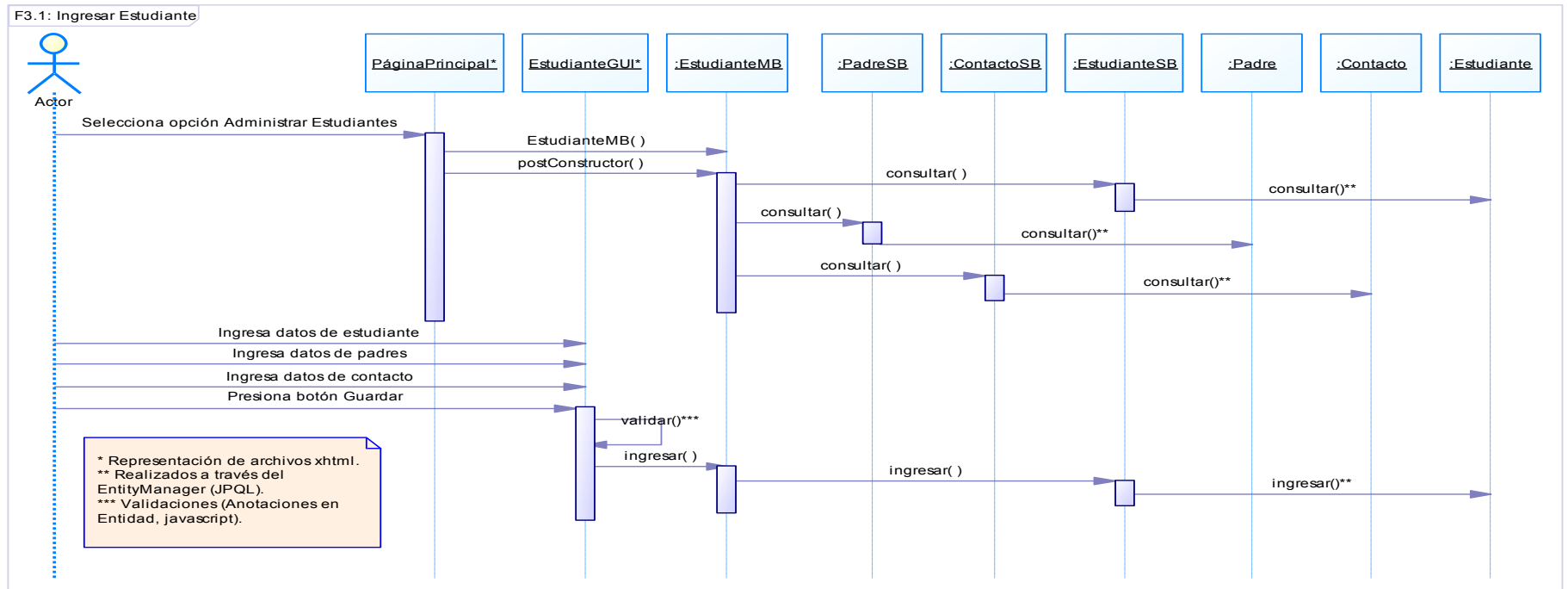


Ilustración 37: Diagrama de Secuencia F3.1

### 3.2.F3.2: Modificar estudiante.

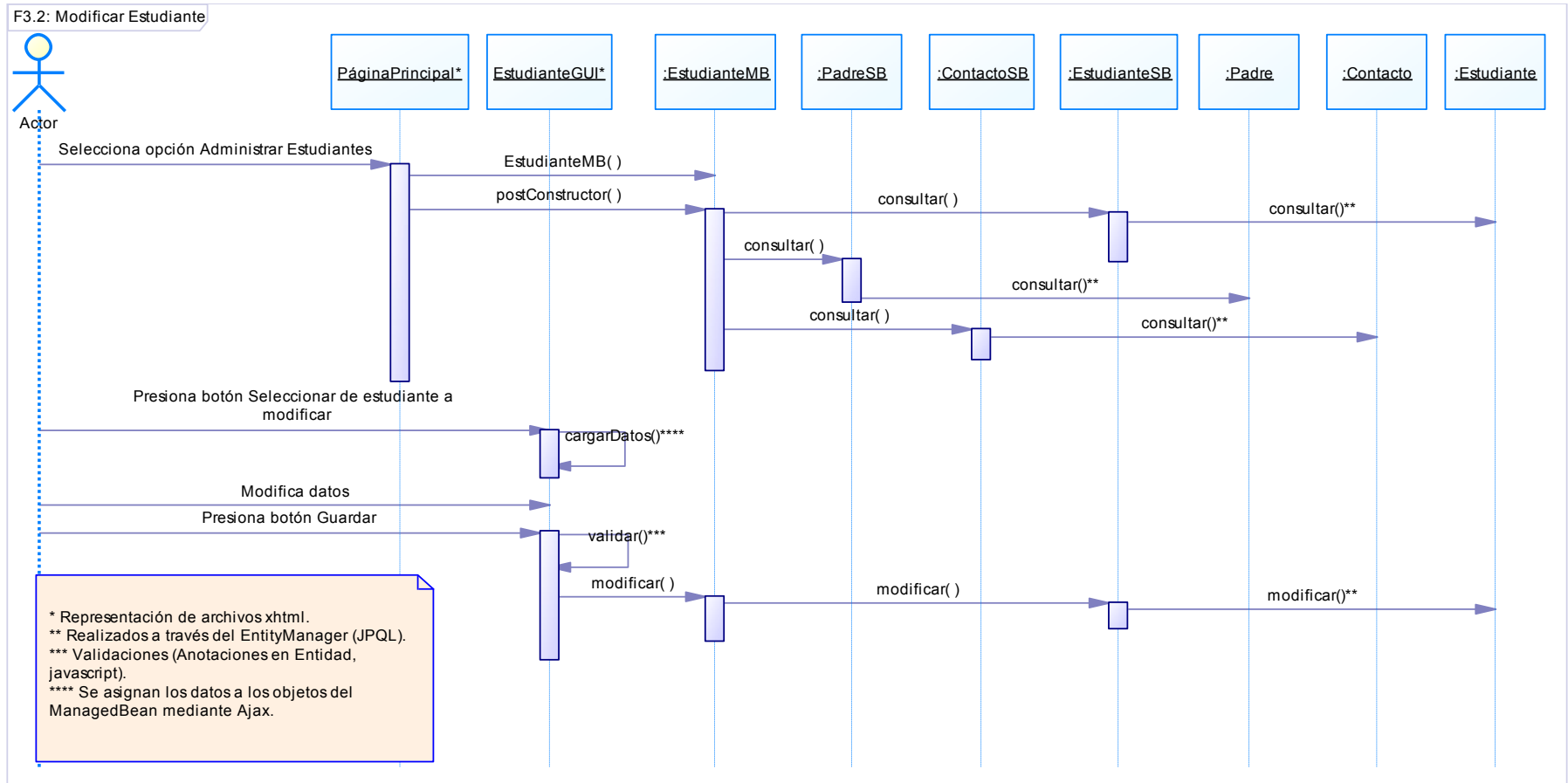


Ilustración 38: Diagrama de Secuencia F3.2

### 3.3.F3.3: Eliminar estudiante.

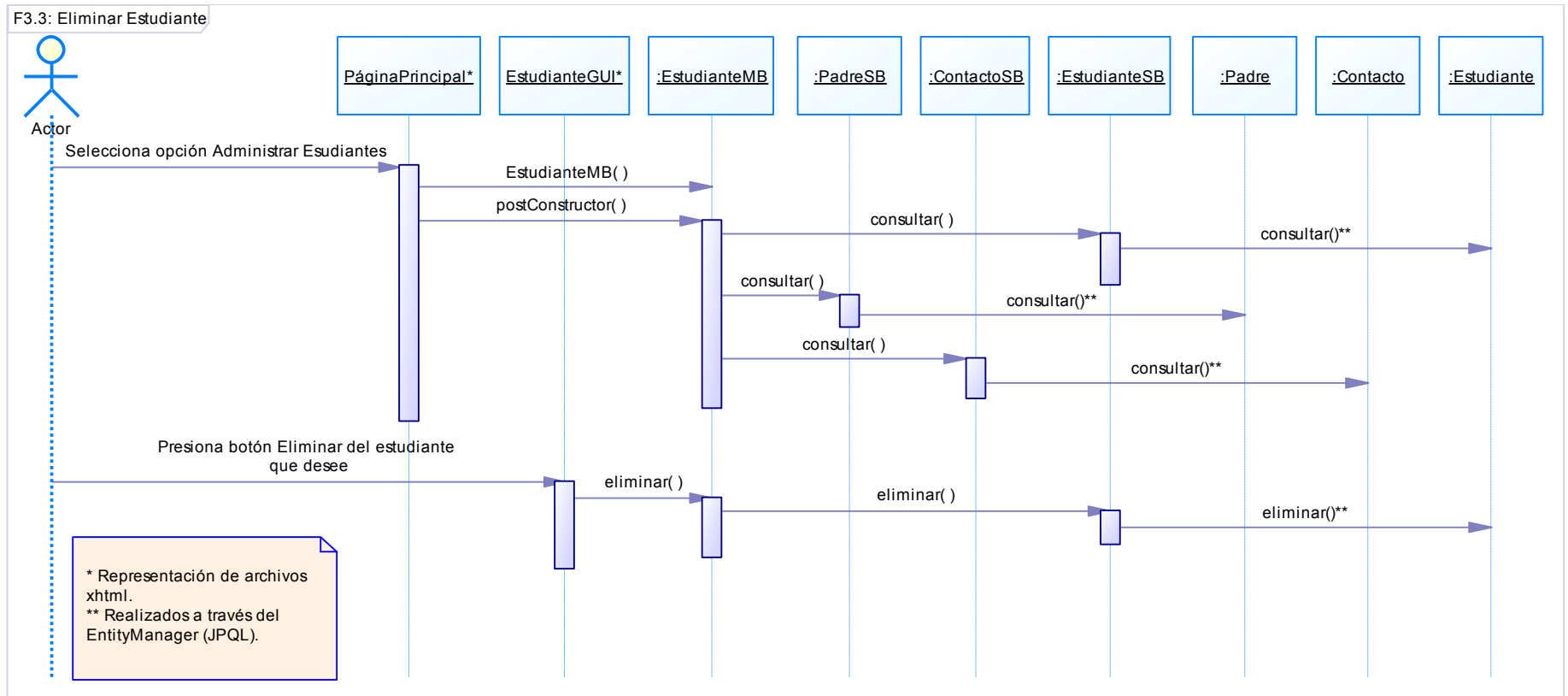


Ilustración 39: Diagrama de Secuencia F3.3

### 3.4.F3.4: Consultar estudiantes.

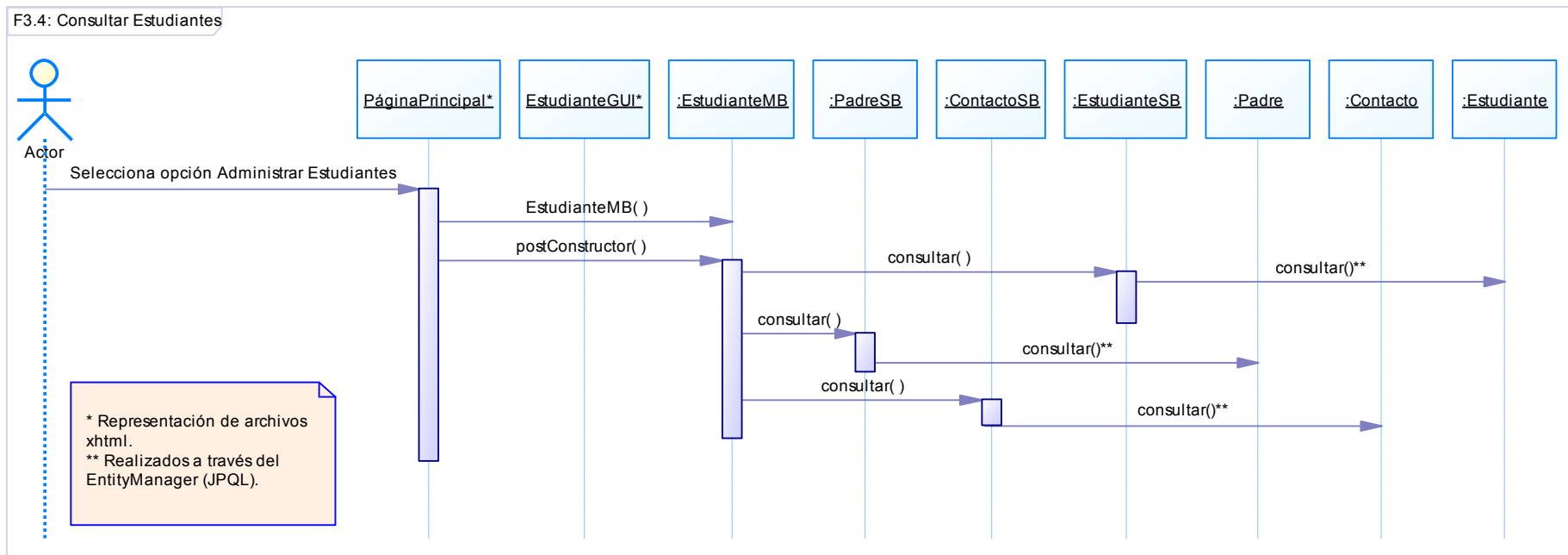


Ilustración 40: Diagrama de Secuencia F3.4

### 3.5.F8.1: Ingresar taller.

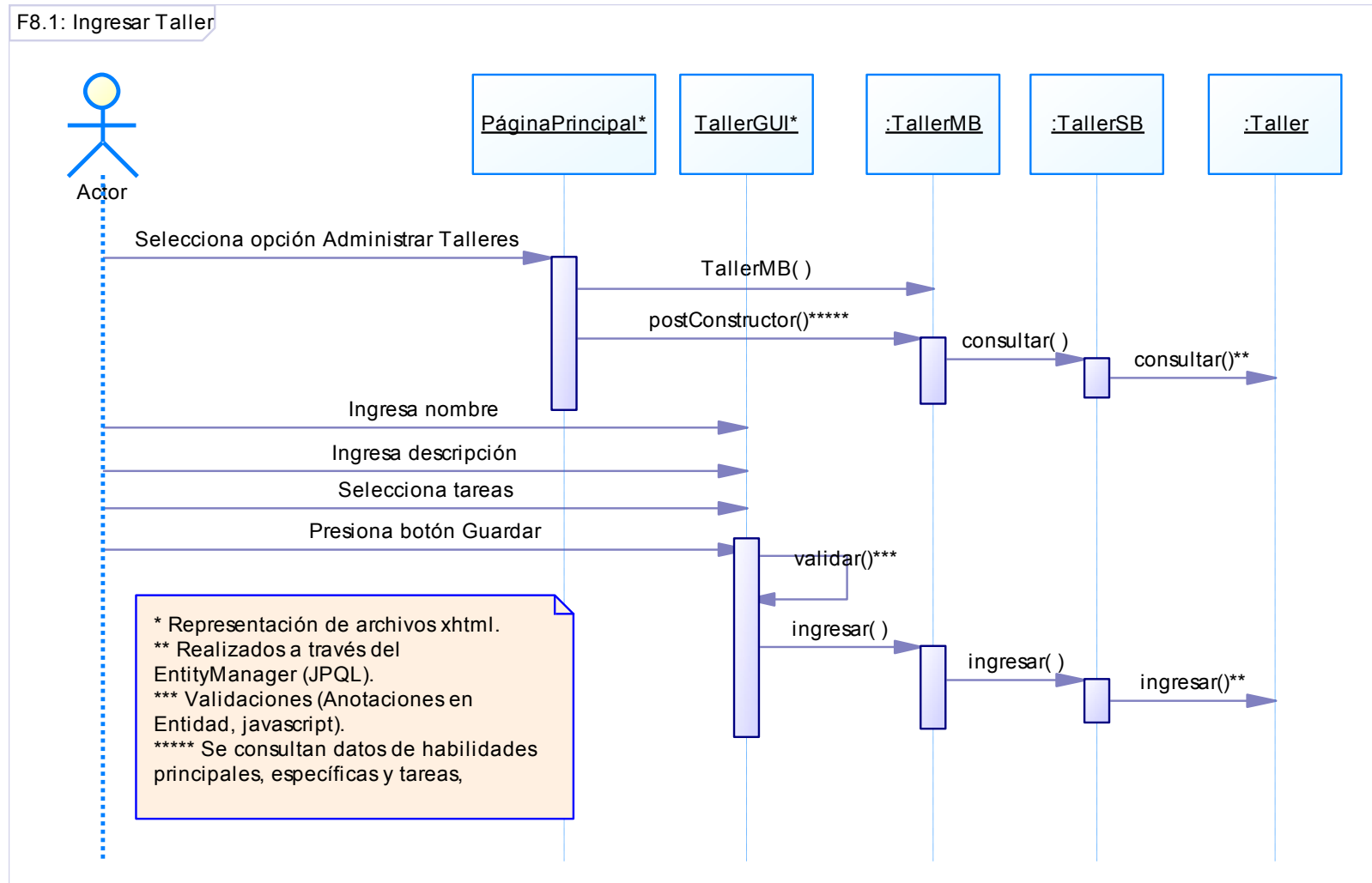


Ilustración 41: Diagrama de Secuencia F8.1

### 3.6.F8.2: Modificar taller.

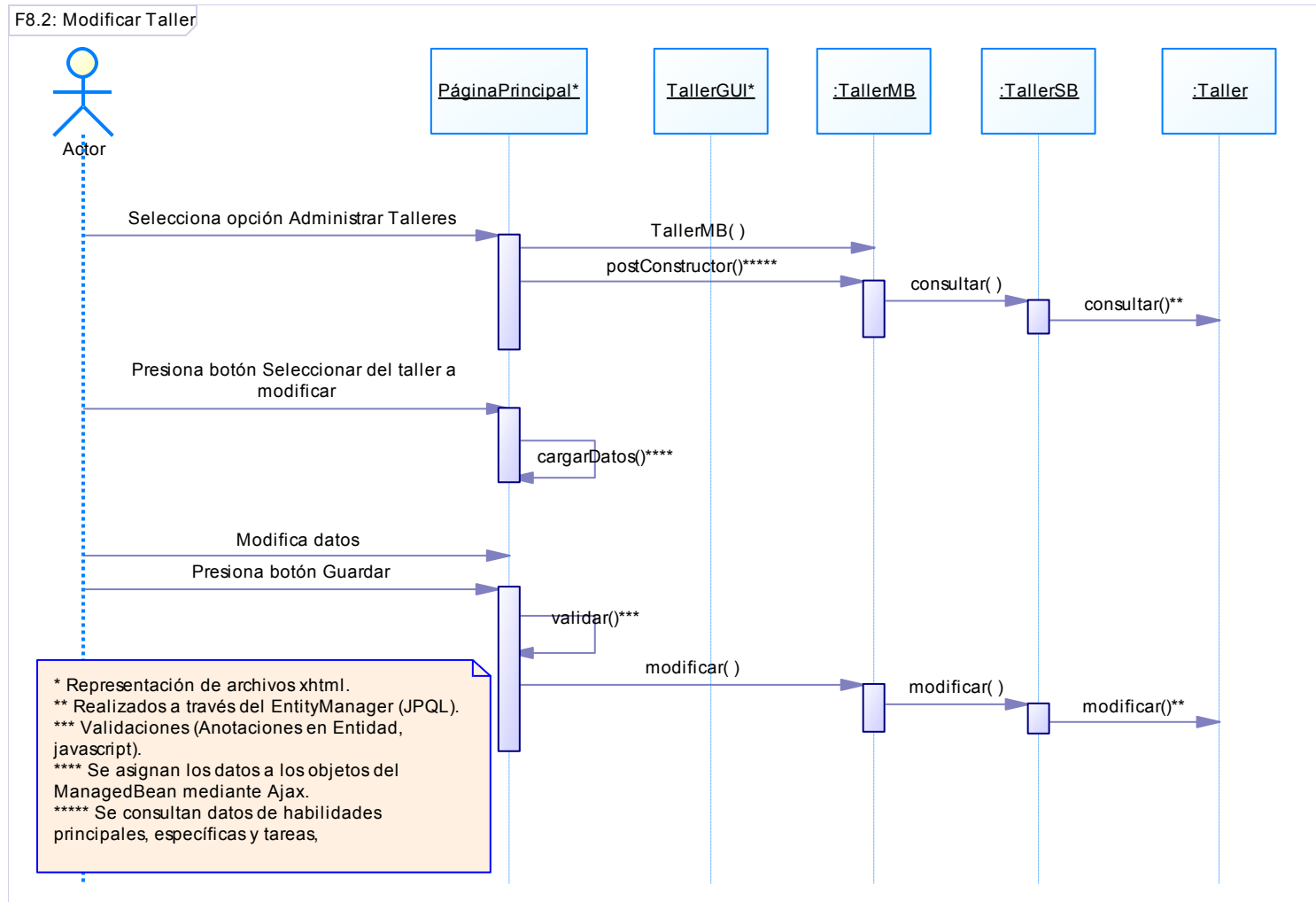


Ilustración 42: Diagrama de Secuencia F8.2

### 3.7.F8.3: Eliminar taller.

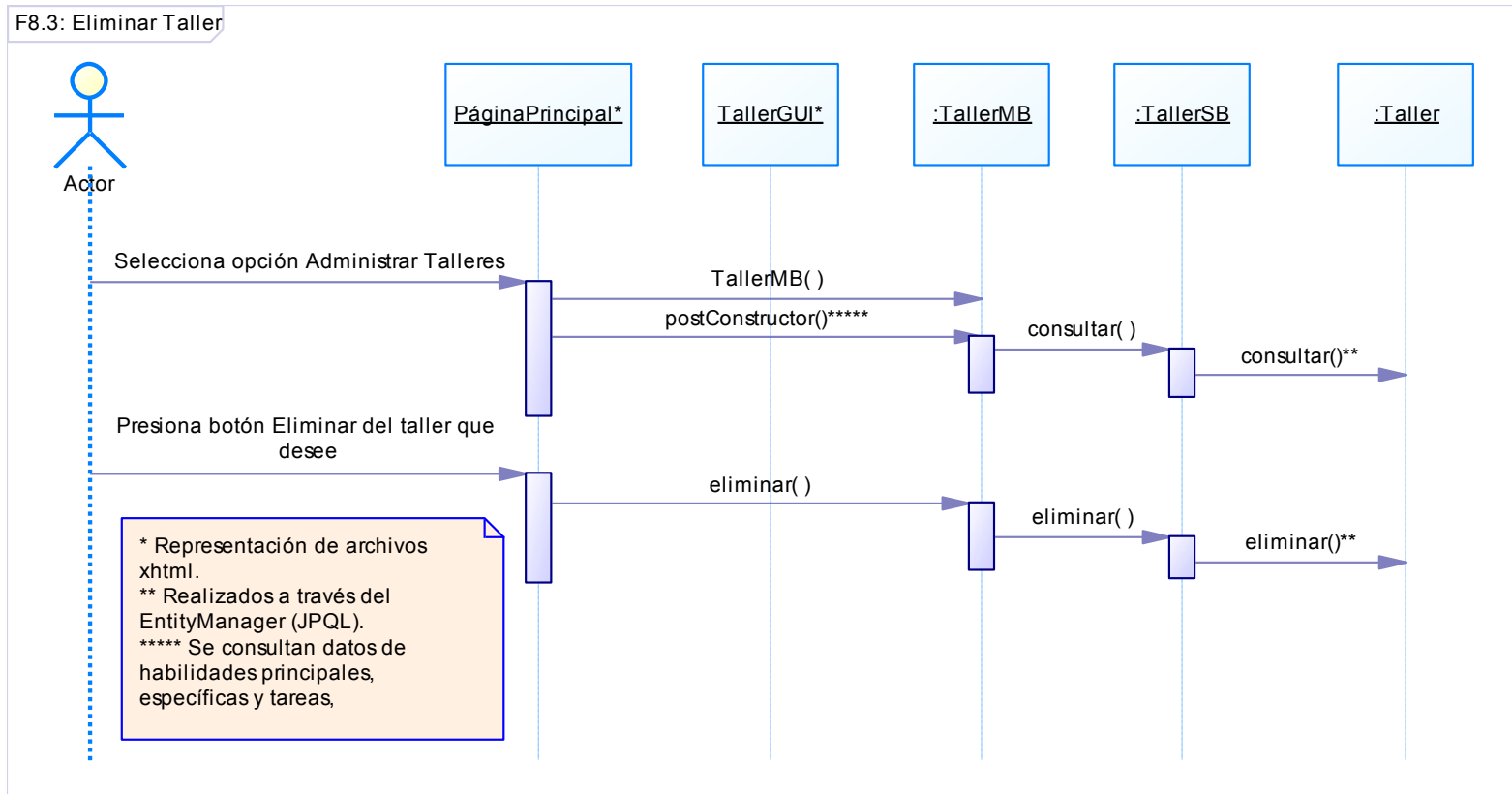


Ilustración 43: Diagrama de Secuencia F8.3

### 3.8.F8.4: Consultar talleres.

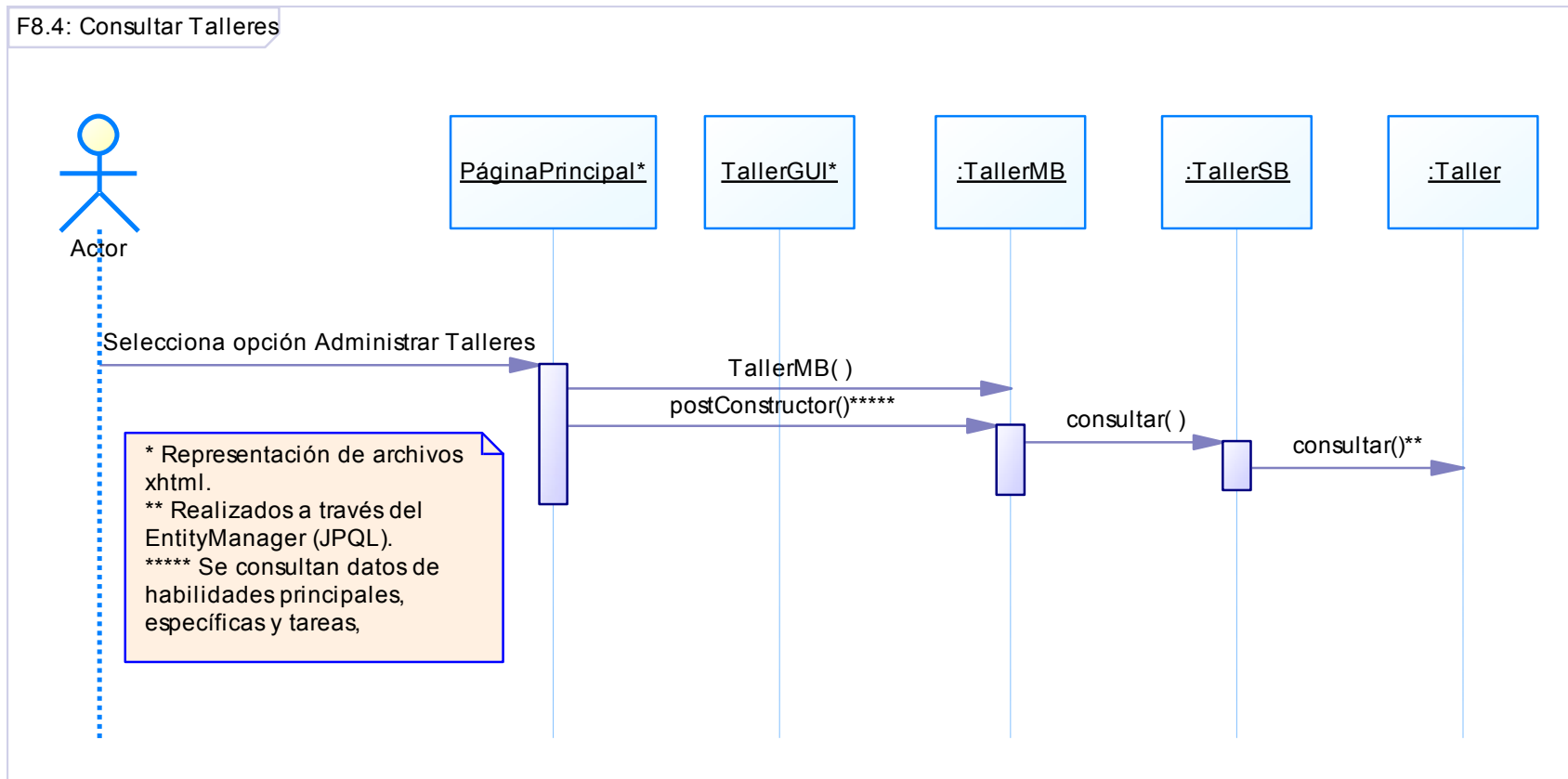


Ilustración 44: Diagrama de Secuencia F8.4

#### 4. Diagramas de E/R

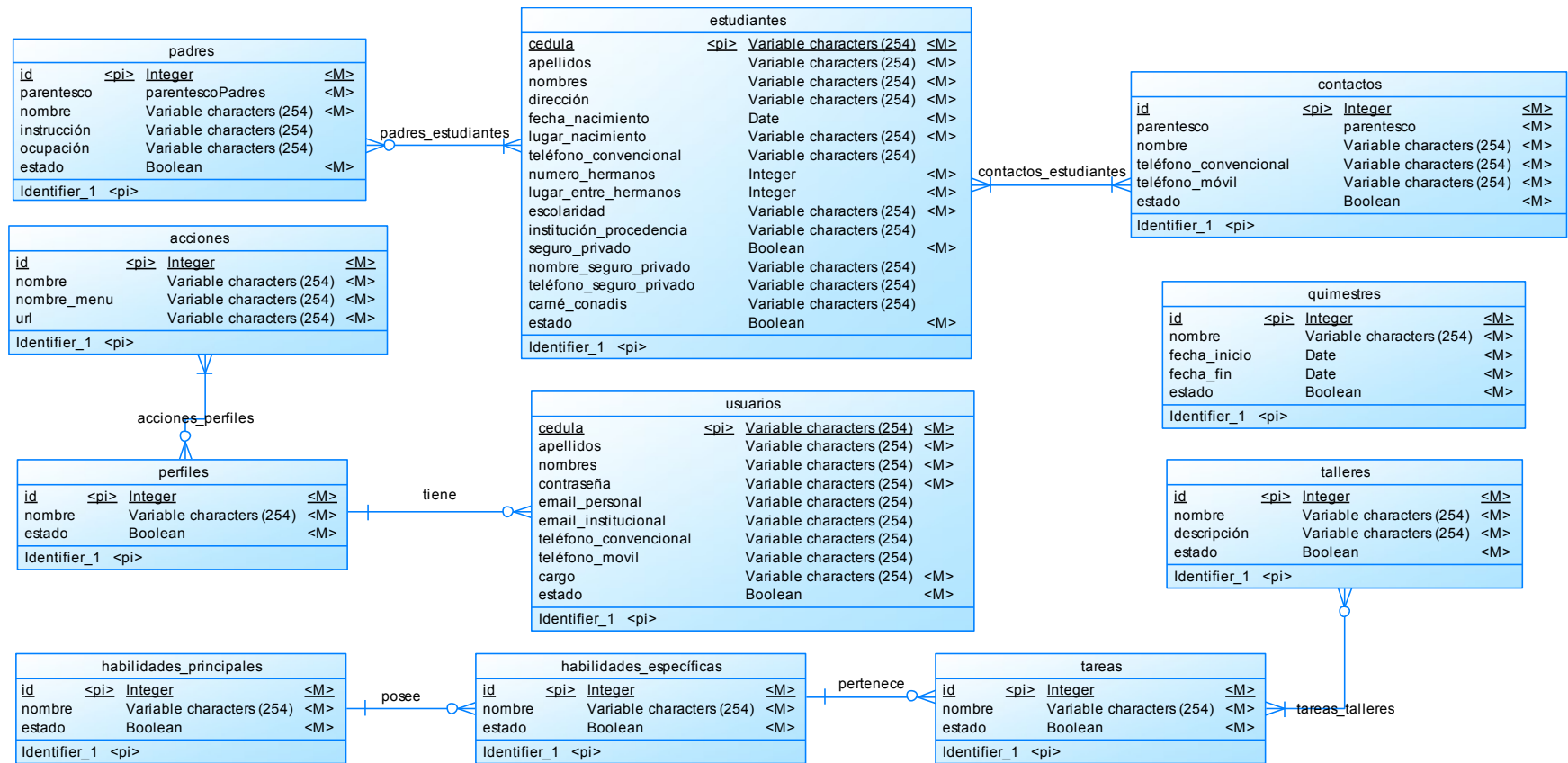


Ilustración 45: Diagrama Entidad Relación

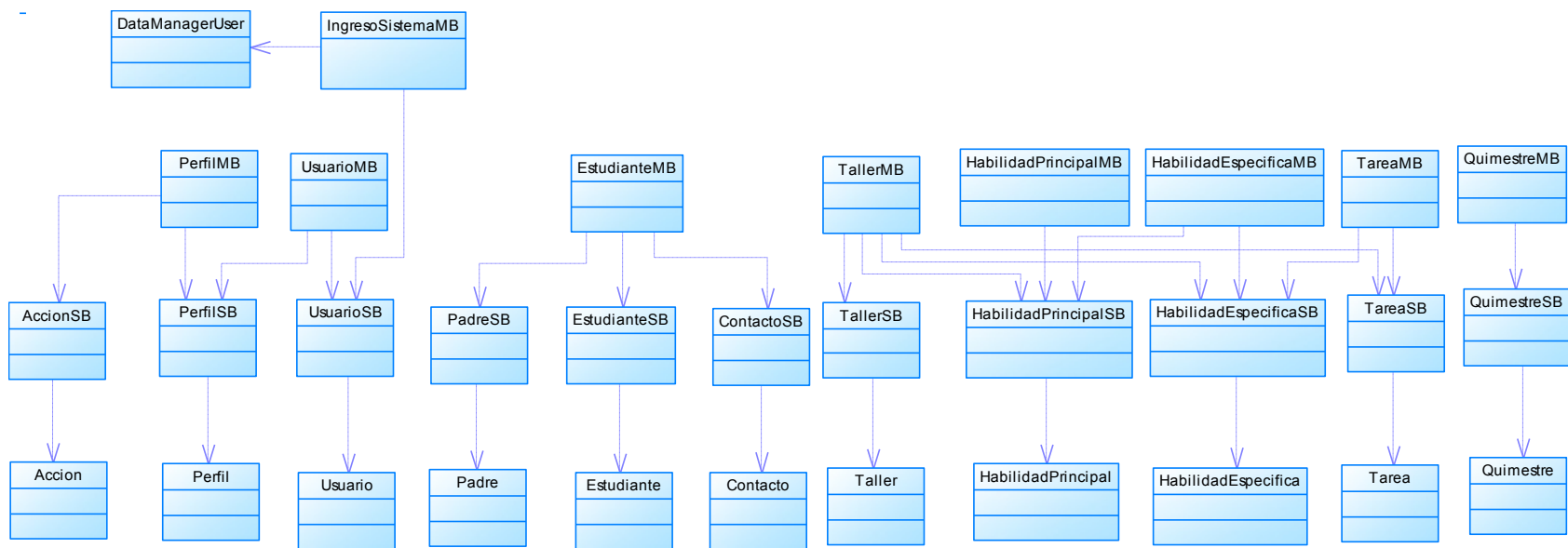
## **5. Plan de Pruebas de Integración (PPI)**

Las pruebas de integración, tienen como objetivo fundamental, comprobar el correcto funcionamiento de los componentes cuando estos trabajan de forma conjunta con otros, verificando que son llamados de forma apropiada y que transfieren los datos adecuados en el tiempo preciso.

La integración del sistema implica identificar grupos de componentes, basándose en las dependencias existentes entre ellos, e integrarlos añadiendo código que haga que funcionen conjuntamente.

La principal dificultad que surge durante las pruebas de integración, es la localización de defectos, por lo que es recomendable utilizar una aproximación incremental para la integración y pruebas, es decir, iniciar con los componentes básicos y el menor número posible e ir añadiendo componentes poco a poco, siguiendo un orden jerárquico.

A continuación se presenta el diagrama de las dependencias existentes entre las clases:



**Ilustración 46: Diagrama de dependencia de clases**

## Grupo 1:

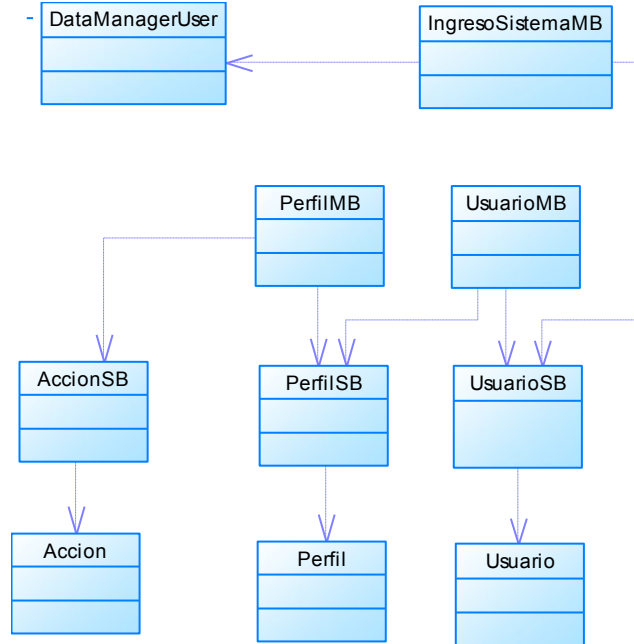


Ilustración 47: Diagrama de dependencia de clases Grupo 1

### Orden: 1

**Evaluador:** Estefanía Narváez

### Diseño:

1. Cargar la página de ingreso al sistema.
2. Cargar la página de administración de perfiles.
3. Cargar la página de administración de usuarios.

### Resultados Esperados:

1. El sistema debe presentar la página de ingreso al sistema.
2. El sistema debe presentar la página de administración de perfiles con los datos de acciones.
3. El sistema debe presentar la página de administración de usuarios con los datos de perfiles.

## Grupo 2:

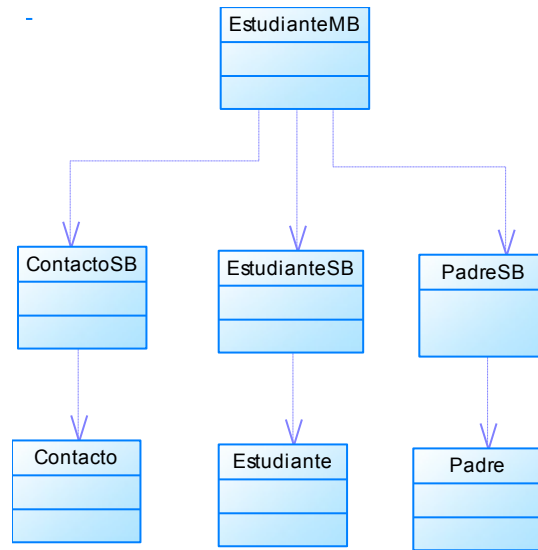


Ilustración 48: Diagrama de dependencia de clases Grupo 2

## Orden: 2

**Evaluador:** Estefanía Narváez

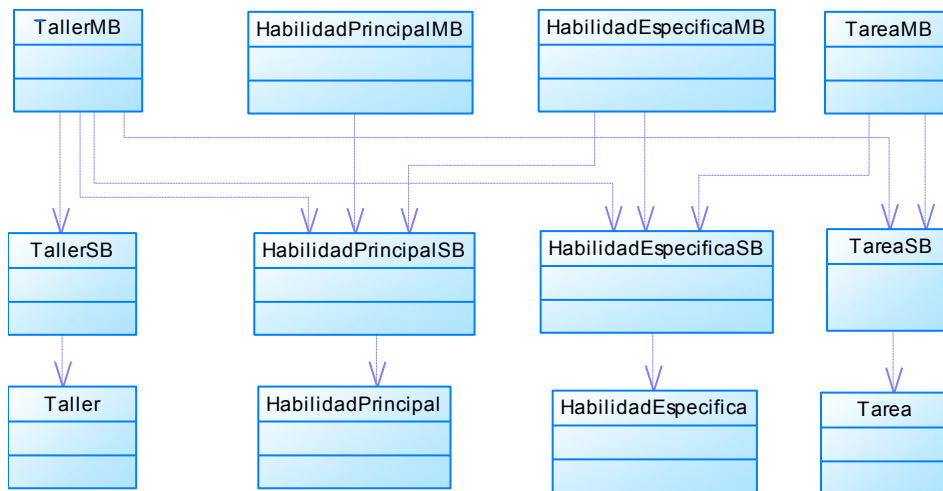
## Diseño:

1. Cargar la página de administración de estudiantes.

## Resultados Esperados:

1. El sistema debe cargar la página de administración de estudiantes correctamente con los datos de contactos y padres.

### Grupo 3:



**Ilustración 49: Diagrama de dependencia de clases Grupo 3**

### Orden: 1

**Evaluador:** Juan Carlos Fernández

### Diseño:

1. Cargar la página de administración de talleres.
2. Cargar la página de administración de habilidades principales.
3. Cargar la página de administración de habilidades específicas.
4. Cargar la página de administración de tareas.

### Resultados Esperados:

1. Se carga correctamente la página de administración de talleres con los datos de habilidades principales, habilidades específicas y tareas.
2. Se carga correctamente la página de administración de habilidades principales.
3. Se carga correctamente la página de administración de habilidades específicas.
4. Se carga correctamente la página de administración de tareas.

## 6.1.6. Implementación

En esta fase se han desarrollado todas las funcionalidades planificadas. A continuación se presenta el código fuente de la funcionalidad que permite la administración de usuarios. El código fuente correspondiente a las funcionalidades restantes se encuentra en el anexo “Código Fuente” del primer ciclo.

El código fuente presentado forma parte del archivo Usuario.xhtml que representa la interfaz gráfica de usuario.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

<ui:composition template="/templates/TemplateFine.xhtml">
  <ui:define name="formTitle">
    <h:outputText value="Administraci&oacute;n de Usuarios" />
  </ui:define>
  <ui:define name="content">
    <h:form id="frmUsuario">
      <p:messages autoUpdate="true" />
      <h:panelGrid columns="2">
        <h:outputText value="Cédula:" />
        <p:inputText value="#{usuarioMB.usuario.cedula}"
                    onkeypress="return validarEnteros(event)"
                    disabled="#{usuarioMB.btnInsertar}" />
        <h:outputText value="Apellidos:" />
        <p:inputText value="#{usuarioMB.usuario.apellidos}"
                    onkeypress="return validarNombres(event)" />
        <h:outputText value="Nombres:" />
        <p:inputText value="#{usuarioMB.usuario.nombres}"
                    onkeypress="return validarNombres(event)" />
        <h:outputText value="Cargo:" />
        <p:inputText value="#{usuarioMB.usuario.cargo}"
                    onkeypress="return validarNombres(event)"/>
        <h:outputText value="Perfil:" />
        <p:selectOneMenu value="#{usuarioMB.perfilSeleccionado}">
          <f:selectItem itemLabel="Seleccione" itemValue=""
                    />
          <f:selectItems value="#{usuarioMB.perfiles}"
                       var="perfil" itemLabel="#{perfil.nombre}"
                       itemValue="#{perfil.id}" />
        </p:selectOneMenu>
        <h:outputText value="E-mail institucional:" />
        <p:inputText
          value="#{usuarioMB.usuario.emailInstitucional}"
          onkeypress="return validarNombres(event)"/>
        <h:outputText value="E-mail alternativo:" />
        <p:inputText value="#{usuarioMB.usuario.emailPersonal}"
                    onkeypress="return validarNombres(event)" />
        <h:outputText value="Teléfono Convencional:" />
        <p:inputText
          value="#{usuarioMB.usuario.telefonoConvencional}"
```

```

        onkeypress="return validarNombres(event)" />
<h:outputText value="Teléfono Móvil:" />
<p:inputText value="#{usuarioMB.usuario.telefonoMovil}"
    onkeypress="return validarNombres(event)" />
<h:outputText value="Contraseña:" />
<p:password value="#{usuarioMB.usuario.contrasena}"
    placeholder=""
        feedback="true" promptLabel="Ingresar contraseña"
        weakLabel="Seguridad Débil" goodLabel="Seguridad
        Media"
        strongLabel="Seguridad Fuerte" />
<h:outputText value="Confirmar contraseña:" />
<p:password value="#{usuarioMB.confirmarContrasena}"
    placeholder="" />
</h:panelGrid>
<br />
<br />
<h:panelGrid columns="3">
    <p:commandButton id="btnGuardar" value="Guardar"
        action="#{usuarioMB.ingresar()}" update="@form"
        disabled="#{usuarioMB.btnInsertar}" />
    <p:commandButton id="btnModificar" value="Modificar"
        action="#{usuarioMB.modificar()}" update="@form"
        disabled="#{!usuarioMB.btnInsertar}" />
    <p:commandButton value="Cancelar"
        action="#{usuarioMB.limpiar()}"
        update="@form" process="@this" />
</h:panelGrid>
<br />
<br />
<p:dataTable id="tblUsuarios" value="#{usuarioMB.usuarios}"
    var="usuario" border="1" paginator="true" rows="10"
    paginatorPosition="bottom" emptyMessage="No existen
    coincidencias">
    <p:column sortBy="#{usuario.cedula}">
        <f:facet name="header">
            <h:outputText value="Cedula" />
        </f:facet>
        <h:outputText value="#{usuario.cedula}" />
    </p:column>
    <p:column sortBy="#{usuario.apellidos}">
        <f:facet name="header">
            <h:outputText value="Apellidos" />
        </f:facet>
        <h:outputText value="#{usuario.apellidos}" />
    </p:column>
    <p:column sortBy="#{usuario.nombres}">
        <f:facet name="header">
            <h:outputText value="Nombres" />
        </f:facet>
        <h:outputText value="#{usuario.nombres}" />
    </p:column>
    <p:column>
        <f:facet name="header">
            <h:outputText value="Acciones" />
        </f:facet>
        <p:commandLink
            action="#{usuarioMB.seleccionar()}"
            process="@this"
            update="@form" title="Seleccionar">
            <h:graphicImage height="23px"
                value="/images/eye.png" />
        </p:commandLink>
    </p:column>
</p:dataTable>

```

```

        <f:setPropertyActionListener
        target="#{usuarioMB.usuario}"
        value="#{usuario}" />
        <f:setPropertyActionListener
        target="#{usuarioMB.perfilSeleccionado}"
        value="#{usuario.perfile.id}" />
    </p:commandLink>
    <p:commandLink title="Eliminar"
    onclick="dialog.show()"
    process="@this" update="@form">
        <h:graphicImage height="23px"
        value="/images/trash.png" />
        <f:setPropertyActionListener
        target="#{usuarioMB.usuario}"
        value="#{usuario}" />
    </p:commandLink>
</p:column>
</p:dataTable>
<br />
<br />
    <p:blockUI block="frmUsuario" trigger="btnGuardar,
    btnModificar">
        <p:graphicImage value="/images/ajax-loader.gif" />
    </p:blockUI>
</h:form>
</ui:define>
<ui:define name="dialog">
    <p:dialog id="dialog1" widgetVar="dialog" header="ELIMINAR"
    modal="true" resizable="false">
        <h:form>
            <h:outputText value="Desea eliminar el usuario?" />
            <p:commandButton value="Si"
            action="#{usuarioMB.eliminar()}"
            update=":frmUsuario" process="@this"
            onclick="dialog.hide()" />
            <p:commandButton value="No" onclick="dialog.hide()" />
        </h:form>
    </p:dialog>
</ui:define>
</ui:composition>
</html>

```

El código fuente que implementa a la interface entre la interfaz gráfica de usuario y el session bean es el siguiente:

```

/**
 * Nombre de la Clase: UsuarioMB
 * Author: JFernández
 * Fecha: 2014/05/02
 */
package ec.uio.fine.managedbeans;

import java.security.MessageDigest;
import java.util.ArrayList;
import java.util.List;

import javax.annotation.PostConstruct;
import javax.ejb.EJB;

```

```

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import javax.faces.context.FacesContext;

import ec.uio.fine.entities.Perfil;
import ec.uio.fine.entities.Usuario;
import ec.uio.fine.sessionbeans.PerfilSB;
import ec.uio.fine.sessionbeans.UsuarioSB;

@ManagedBean
@ViewScoped
public class UsuarioMB {
    @EJB
    private UsuarioSB usuarioSB;
    @EJB
    private PerfilSB perfilSB;

    private Usuario usuario;
    private List<Usuario> usuarios;

    private Perfil perfil;
    private List<Perfil> perfiles;

    private String perfilSeleccionado = "";

    private boolean btnInsertar = false; // Indicará si el botón Guardar está
        // activo y el botón Modificar inactivo
        // o viceversa.
        // También indicará si el campo cédula
        // está activo para su edición.

    private String confirmarContrasena; // Campo que ayudará a confirmar la
        // contraseña ingresada por el usuario.
    private String contrasenaActual; // Campo que almacenará la contraseña
        // cuando al editar un usuario no se la
        // quiera cambiar.

    /**
     * Constructor
     */
    public UsuarioMB() {
        usuario = new Usuario();
    }

    /**
     * Método que se ejecutará inmediatamente después del constructor.
     */
    @PostConstruct
    public void postConstructor() {
        btnInsertar = false;
        usuarios = usuarioSB.consultar();
        perfilSeleccionado = "";
        perfiles = perfilSB.consultar();
        perfil = new Perfil();
        confirmarContrasena = "";
        contrasenaActual = "";
    }

    /**
     * Getters y Setters
     */

```

```

public UsuarioSB getUsuarioSB() {
    return usuarioSB;
}

public void setUsuarioSB(UsuarioSB newUsuarioSB) {
    usuarioSB = newUsuarioSB;
}

public PerfilSB getPerfilSB() {
    return perfilSB;
}

public void setPerfilSB(PerfilSB newPerfilSB) {
    perfilSB = newPerfilSB;
}

public Usuario getUsuario() {
    return usuario;
}

public void setUsuario(Usuario newUsuario) {
    usuario = newUsuario;
}

public List<Usuario> getUsuarios() {
    return usuarios;
}

public void setUsuarios(List<Usuario> newUsuarios) {
    usuarios = newUsuarios;
}

public Perfil getPerfil() {
    return perfil;
}

public void setPerfil(Perfil newPerfil) {
    perfil = newPerfil;
}

public List<Perfil> getPerfiles() {
    return perfiles;
}

public void setPerfiles(List<Perfil> newPerfiles) {
    perfiles = newPerfiles;
}

public String getPerfilSeleccionado() {
    return perfilSeleccionado;
}

public void setPerfilSeleccionado(String newPerfilSeleccionado) {
    perfilSeleccionado = newPerfilSeleccionado;
}

public boolean getBtnInsertar() {
    return btnInsertar;
}

public void setBtnInsertar(boolean newBtnInsertar) {
    btnInsertar = newBtnInsertar;
}

```

```

public String getConfirmarContrasena() {
    return confirmarContrasena;
}

public void setConfirmarContrasena(String confirmarContrasena) {
    this.confirmarContrasena = confirmarContrasena;
}

public String getContrasenaActual() {
    return contrasenaActual;
}

public void setContrasenaActual(String contrasenaActual) {
    this.contrasenaActual = contrasenaActual;
}

/**
 * Método que permite registrar un usuario en la base de datos, realizando
 * las validaciones necesarias.
 */
public void ingresar() {
    /**
     * Se verifica que el usuario que se intenta ingresar no esté
     * registrado en la base de datos.
     */
    Usuario aux = usuarioSB.consultarPorCedula(usuario.getCedula());

    if (aux != null && aux.getCedula() == null) {

        /**
         * Se verifica que haya al ingresado al menos un e-mail.
         */
        if ((usuario.getEmailInstitucional() != null
            && (!usuario.getEmailInstitucional().equals(""))
            || (usuario.getEmailPersonal() != null
            && (!usuario.getEmailPersonal().equals("")))) {
            /**
             * Se verifica que se haya ingresado al menos un
teléfono.

            */
            if ((usuario.getTelefonoConvencional() != null
            && (!usuario.getTelefonoConvencional().equals(""))
            || (usuario.getTelefonoMovil() != null
            && (!usuario.getTelefonoMovil().equals("")))) {
                /**
                 * Se verifica que el campo contraseña coincida
con

                * el campo confirmar contraseña
                */
                if (confirmarContrasena()) {
                    try {
                        // Convierte la contraseña en una
                        // cadena de caracteres
                        byte[] auxPassEncrypt = usuario
                            .getContrasena().getBytes();
                        MessageDigest md = MessageDigest
                            .getInstance("MD5"); // Crea un
digest // para codificación
MD5
                        md.update(auxPassEncrypt);
                    }
                }
            }
        }
    }
}

```

```

        // Obtiene Hash MD5 de la
contraseña
        byte[] auxEncryptedPass =
md.digest();

        // Convierte byte a string
        StringBuffer sb = new
StringBuffer();
        for (int i = 0;
i < auxEncryptedPass.length; i++)
{

            sb.append(Integer.toString((a
uxEncryptedPass[i] & 0xff) +
0x100, 16).substring(1));
        }
        usuario.setContrasena(sb.toString()
);
    } catch (Exception e) {
        FacesMessage mensaje = new
FacesMessage(FacesMessage
.SEVERITY_ERROR, "Contraseña no pudo
ser almacenada. " + e.getMessage(),
"null");

        FacesContext.getCurrentInstance()
.addMessage(null, mensaje);
    }

perfil = perfilSB
.consultar(perfilSeleccionado);
usuario.setPerfile(perfil);
usuario.setEstado((byte) 1);

try {
    usuarioSB.ingresar(this.usuario);
    FacesMessage mensaje = new
FacesMessage(FacesMessage
.SEVERITY_INFO, "Usuario ingresado
con éxito.\n", "null");

    FacesContext.getCurrentInstance()
.addMessage(null, mensaje);
} catch (Exception e) {
    FacesMessage mensaje = new
FacesMessage(FacesMessage
.SEVERITY_ERROR, "El usuario no
pudo ser ingresado. " +
e.getMessage(), e.getMessage());

    FacesContext.getCurrentInstance()
.addMessage(null, mensaje);
}

/**
 * Encera variables de la vista
 */
usuarios = usuarioSB.consultar();
usuario = new Usuario();
perfilSeleccionado = "";
perfiles = perfilSB.consultar();

```

```

        perfil = new Perfil();
        confirmarContrasena = "";
    } else {
        FacesMessage mensaje = new FacesMessage(
            FacesMessage.SEVERITY_ERROR, "Contrasena
            no coincide con campo de
            confirmación","null");

        FacesContext.getCurrentInstance()
            .addMessage(null,mensaje);
    }
} else {
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_ERROR, "Debe ingresar al
        menos un número de teléfono.", "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}
} else {
    FacesMessage mensaje = new FacesMessage(FacesMessage
        .SEVERITY_ERROR, "Debe ingresar al menos un E-mail.",
        "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}
} else {
    /**
     * Si el usuario que se intenta ingresar se había eliminado
     * anteriormente.
     */
    if (aux.getEstado() == 0) {
        this.modificar();
    } else {
        FacesMessage mensaje = new FacesMessage(FacesMessage
            .SEVERITY_ERROR, "El usuario que intenta ingresar ya
            está registrado.", "null");
        FacesContext.getCurrentInstance()
            .addMessage(null, mensaje);
    }
}
}

/**
 * Método que permite cargar los datos de un usuario seleccionado para
 * visualizar el detalle de los mismos o editarlos.
 */
public void seleccionar() {
    btnInsertar = true;
    contrasenaActual = usuario.getContrasena();
}

/**
 * Método que permite modificar un usuario existente en la base de datos,
 * realizando las validaciones necesarias.
 */
public void modificar() {
    /**
     * Se verifica que haya al ingresado al menos un e-mail.
     */
    if ((usuario.getEmailInstitucional() != null
        && (!usuario.getEmailInstitucional().equals(""))))
        || (usuario.getEmailPersonal() != null
        && (!usuario.getEmailPersonal().equals("")))) {

```

```

/**
 * Se verifica que se haya ingresado al menos un teléfono.
 */
if ((usuario.getTelefonoConvencional() != null
&& (!usuario.getTelefonoConvencional().equals(""))
|| (usuario.getTelefonoMovil() != null
&& (!usuario.getTelefonoMovil().equals("")))) {
    if (usuario.getContrasena().equals("")) {
        usuario.setContrasena(contrasenaActual);
    } else {
        /**
         * Se verifica que el campo contraseña coincida
         * con el campo confirmar contraseña,
         */
        if (confirmarContrasena()) {
            try {
                // Convierte la contraseña en una
                // cadena de caracteres
                byte[] auxPassEncrypt = usuario
                    .getContrasena().getBytes();

                //Crea un digest para codificación
                MD5
                MessageDigest md = MessageDigest
                    .getInstance("MD5");

                md.update(auxPassEncrypt);

                // Obtiene Hash MD5 de la
                // contraseña
                byte[] auxEncryptedPass =

                // Convierte byte a string
                StringBuffer sb = new
                StringBuffer();
                for (int i = 0;
                    i < auxEncryptedPass.length; i++)
                {

                    sb.append(Integer.toString((a
                    uxEncryptedPass[i] & 0xff) +
                    0x100, 16).substring(1));
                }
                usuario.setContrasena(sb.toString()
                );
            } catch (Exception e) {
                FacesMessage mensaje = new
                FacesMessage(FacesMessage
                .SEVERITY_ERROR, "Contraseña no
                pudo ser almacenada. " +
                e.getMessage(), "null");

                FacesContext.getCurrentInstance()
                .addMessage(null, mensaje);
            }
        } else {
            FacesMessage mensaje = new FacesMessage(
            FacesMessage.SEVERITY_ERROR, "Contraseña
            no coincide con campo de
            confirmación","null");

```

md.digest());

```

        FacesContext.getCurrentInstance()
            .addMessage(null, mensaje);
    }
}
perfil = perfilSB.consultar(perfilSeleccionado);
usuario.setPerfile(perfil);
usuario.setEstado((byte) 1);

try {
    usuarioSB.modificar(usuario);
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_INFO, "Usuario modificado
        con éxito.\n", "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
} catch (Exception e) {
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_ERROR, "El usuario no pudo
        ser modificado. " + e.getMessage(),
        e.getMessage());
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}

/**
 * Encera variables de la vista
 */
usuarios = usuarioSB.consultar();
usuario = new Usuario();
perfilSeleccionado = "";
perfiles = perfilSB.consultar();
perfil = new Perfil();
confirmarContrasena = "";
btnInsertar = false;
} else {
    FacesMessage mensaje = new FacesMessage(FacesMessage
        .SEVERITY_ERROR, "Debe ingresar al menos un número de
        teléfono.", "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}
} else {
    FacesMessage mensaje = new FacesMessage(FacesMessage
        .SEVERITY_ERROR, "Debe ingresar al menos un E-mail.", "null");
    FacesContext.getCurrentInstance().addMessage(null, mensaje);
}
}

/**
 * Método que permite desactivar un usuario existente en la base de datos
 * (borrado lógico)
 */
public void eliminar() {
    try {
        usuarioSB.eliminar(this.usuario);
        FacesMessage mensaje = new FacesMessage(FacesMessage
            .SEVERITY_INFO, "Usuario eliminado con éxito. ", "null");
        FacesContext.getCurrentInstance().addMessage(null, mensaje);
    } catch (Exception e) {
        FacesMessage mensaje = new FacesMessage(FacesMessage.
            SEVERITY_ERROR, "El usuario no pudo ser eliminado. " +
            e.getMessage(), e.getMessage());
        FacesContext.getCurrentInstance().addMessage(null, mensaje);
    }
}

```

```

    }
    btnInsertar = false;
    usuarios = usuarioSB.consultar();
    perfilSeleccionado = "";
    perfiles = perfilSB.consultar();
    perfil = new Perfil();
    confirmarContrasena = "";
    usuario = new Usuario();
}

/**
 * Método que permite encerar las variables de la vista
 */
public void limpiar() {
    btnInsertar = false;
    usuarios = usuarioSB.consultar();
    perfilSeleccionado = "";
    perfiles = perfilSB.consultar();
    perfil = new Perfil();
    confirmarContrasena = "";
    contrasenaActual = "";
    usuario = new Usuario();
}

/**
 * Método que permite verificar si la confirmación de la contraseña es
 * correcta.
 */
public boolean confirmarContrasena() {
    if (usuario.getContrasena().equals(confirmarContrasena)) {
        return true;
    }
    return false;
}
}
}

```

A continuación se presenta el código fuente de la lógica del negocio, contenido en el archivo UsuarioSB.java:

```

/**
 * Nombre de la Clase: UsuarioSB
 * Autor: JFernández
 * Fecha: 2014/05/08
 */
package ec.uio.fine.sessionbeans;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import ec.uio.fine.entities.Estudiante;
import ec.uio.fine.entities.Usuario;

@Stateless
public class UsuarioSB {
    @PersistenceContext
    private EntityManager em;
}

```

```

/**
 * Permite registrar un usuario en la base de datos.
 * @param usuario
 */
public void ingresar(Usuario usuario) {
    em.persist(usuario);
}

/**
 * Permite modificar un usuario existente en la base de datos.
 * @param usuario
 */
public void modificar(Usuario usuario) {
    em.merge(usuario);
}

/**
 * Permite cambiar el estado de un usuario existente en la base de datos
 * (borrado lógico).
 * @param usuario
 */
public void eliminar(Usuario usuario) {
    usuario.setEstado((byte) 0);
    em.merge(usuario);
}

/**
 * Permite consultar todos los usuarios activos en la base de datos.
 */
public List<Usuario> consultar() {
    Query query = em.createQuery("SELECT u FROM Usuario u WHERE
    u.estado=:estado");
    query.setParameter("estado", ((byte)1));
    return query.getResultList();
}

/**
 * Permite consultar un usuario en la base de datos por medio de su cédula.
 * @param cedula
 */
public Usuario consultarPorCedula(String cedula) {
    Usuario resultado;
    Query query = em.createQuery("SELECT u FROM Usuario u WHERE
    u.cedula=:cedula");
    query.setParameter("cedula", cedula);
    try{
        resultado = (Usuario) query.getSingleResult();
    }catch(Exception e){
        resultado = new Usuario();
    }
    return resultado;
}
}

```

Finalmente se presenta la entidad Usuario.java.

```

package ec.uio.fine.entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;

```

```

import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;

/**
 * The persistent class for the usuarios database table.
 */
@Entity
@Table(name="usuarios")
public class Usuario {

    @Id
    @NotNull(message="Cédula es un campo obligatorio")
    private String cedula;

    @NotNull(message="Apellidos es un campo obligatorio")
    private String apellidos;

    @NotNull(message="Cargo es un campo obligatorio")
    private String cargo;

    @Size(min=6, max=100, message="Contraseña debe tener al menos {min}
caracteres.")
    private String contrasena;

    @Column(name="email_institucional")
    private String emailInstitucional;

    @Column(name="email_personal")
    private String emailPersonal;

    private byte estado;

    @NotNull(message="Nombres es un campo obligatorio")
    private String nombres;

    @Column(name="telefono_convencional")
    private String telefonoConvencional;

    @Column(name="telefono_movil")
    private String telefonoMovil;

    //bi-directional many-to-one association to Perfil
    @ManyToOne
    @JoinColumn(name="id_perfil")
    private Perfil perfile;

    public Usuario() {
    }

    public String getCedula() {
        return this.cedula;
    }

    public void setCedula(String cedula) {
        this.cedula = cedula;
    }

    public String getApellidos() {
        return this.apellidos;
    }
}

```

```

public void setApellidos(String apellidos) {
    this.apellidos = apellidos;
}

public String getCargo() {
    return this.cargo;
}

public void setCargo(String cargo) {
    this.cargo = cargo;
}

public String getContrasena() {
    return this.contrasena;
}

public void setContrasena(String contrasena) {
    this.contrasena = contrasena;
}

public String getEmailInstitucional() {
    return this.emailInstitucional;
}

public void setEmailInstitucional(String emailInstitucional) {
    this.emailInstitucional = emailInstitucional;
}

public String getEmailPersonal() {
    return this.emailPersonal;
}

public void setEmailPersonal(String emailPersonal) {
    this.emailPersonal = emailPersonal;
}

public byte getEstado() {
    return this.estado;
}

public void setEstado(byte estado) {
    this.estado = estado;
}

public String getNombres() {
    return this.nombres;
}

public void setNombres(String nombres) {
    this.nombres = nombres;
}

public String getTelefonoConvencional() {
    return this.telefonoConvencional;
}

public void setTelefonoConvencional(String telefonoConvencional) {
    this.telefonoConvencional = telefonoConvencional;
}

public String getTelefonoMovil() {
    return this.telefonoMovil;
}

```

```

}

public void setTelefonoMovil(String telefonoMovil) {
    this.telefonoMovil = telefonoMovil;
}

public Perfil getPerfile() {
    return this.perfile;
}

public void setPerfile(Perfil perfile) {
    this.perfile = perfile;
}

```

### 6.1.7. Integración y Pruebas

En esta fase se presentan las pruebas realizadas en el sistema y los resultados obtenidos en cada una, excluyendo el resumen de los resultados obtenidos al ejecutar las pruebas de unidad.

#### 6.1.7.1. Pruebas de Integración

La ejecución del plan de pruebas de integración produjo resultados satisfactorios que se resumen a continuación:

Grupo	Prueba	Resultados Esperados	Observaciones
1	1	Sí	Resultado satisfactorio.
	2	Sí	Resultado satisfactorio.
	3	Sí	Resultado satisfactorio.
2	1	Sí	Resultado satisfactorio.
3	1	Sí	Resultado satisfactorio.
	2	Sí	Resultado satisfactorio.
	3	Sí	Resultado satisfactorio.
	3	Sí	Resultado satisfactorio.

Tabla 28: Pruebas de Integración. Ciclo I

### 6.1.7.2. Pruebas del Sistema

Los resultados obtenidos durante la ejecución del plan de pruebas del sistema desarrollado durante la fase de requerimientos del presente ciclo se resumen a continuación:

Prueba	Resultados Esperados	Observaciones
1	Sí	Resultado satisfactorio.
2	Sí	Resultado satisfactorio.
3	Sí	Resultado satisfactorio.
4	Sí	Resultado satisfactorio.
5	Sí	Resultado satisfactorio.
6	Sí	Resultado satisfactorio.
7	Sí	Resultado satisfactorio.
8	Sí	Resultado satisfactorio.
9	Sí	Resultado satisfactorio.
10	Sí	Resultado satisfactorio.
11	Sí	Resultado satisfactorio.
12	Sí	Resultado satisfactorio.
13	Sí	Resultado satisfactorio.
14	Sí	Resultado satisfactorio.
15	Sí	Resultado satisfactorio.
16	Sí	Resultado satisfactorio.
17	Sí	Resultado satisfactorio.
18	Sí	Resultado satisfactorio.
19	Sí	Resultado satisfactorio.
20	Sí	Resultado satisfactorio.
21	Sí	Resultado satisfactorio.
22	Sí	Resultado satisfactorio.

Prueba	Resultados Esperados	Observaciones
23	Sí	Resultado satisfactorio.
24	Sí	Resultado satisfactorio.
25	Sí	Resultado satisfactorio.
26	Sí	Resultado satisfactorio.
27	Sí	Resultado satisfactorio.
28	Sí	Resultado satisfactorio.
29	Sí	Resultado satisfactorio.
30	Sí	Las habilidades principales y específicas se repiten para cada tarea perteneciente a éstas, sería recomendable hacer alguna modificación para evitar esto.
31	Sí	Resultado satisfactorio.
32	Sí	Resultado satisfactorio.
33	Sí	Resultado satisfactorio.

**Tabla 29: Pruebas de Sistema. Ciclo I**

### **6.1.8. Postmortem**

La fase de postmortem es esencial en la mejora continua debido a que en esta fase se evalúa el trabajo realizado hasta el momento, los aciertos y fallos del proceso, así como el estado del ciclo. Este análisis permitirá establecer correctivos en ciclos posteriores, e incluso en proyectos futuros.

#### **6.1.8.1. Análisis de Aciertos**

Los aciertos detectados durante este ciclo son:

- ✓ Buena comunicación entre los miembros del equipo.

- ✓ A pesar de la baja disponibilidad de tiempo, los miembros del equipo siempre estuvieron dispuestos a cumplir con las actividades asignadas.
- ✓ Los miembros del equipo aprendieron y aplicaron de una manera óptima los conceptos de diseño utilizando esta nueva arquitectura (nueva para los miembros del equipo).
- ✓ Los miembros del equipo siempre estuvieron dispuestos a aprender la arquitectura propuesta.
- ✓ Se lograron muchos productos de alta calidad.

#### ***6.1.8.2. Análisis de Fallos***

Los fallos que se presentaron durante este ciclo se describen a continuación:

- ✓ El desconocimiento de la arquitectura que se utilizó se vio reflejado en las horas necesarias para desarrollar cada funcionalidad.
- ✓ Los miembros del equipo tuvieron que trabajar en algunas ocasiones por más de ocho horas al día para lograr cumplir con la planificación debido a que contaban con solo dos días a la semana para dedicarse al proyecto actual.

### 6.1.8.3. Formulario PEER Consolidado

A continuación se presenta la forma PEER consolidada de la evaluación de cada miembro:

#### TSPi Team and Peer Evaluation: Form PEER

For each role, evaluate the work required and the relative difficulty in % during this cycle.		
Role	Work Required	Role Difficulty
Team Leader	12.5	10
Development Manager	22.5	17.5
Planning Manager	25	27.5
Quality/Process Manager	25	30
Support Manager	15	15
Total Contribution (100%)	<b>100</b>	<b>100</b>

Rate the overall team against each criterion. Circle one number from 1 (inadequate) to 5 (superior).					
Team spirit	1	2	3	4	5
Overall effectiveness	1	2	3	4	5
Rewarding experience	1	2	3	4	5
Team productivity	1	2	3	4	5
Process quality	1	2	3	4	5
Product quality	1	2	3	4	5

Rate role for overall contribution. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	5
Development Manager	1	2	3	4	5
Planning Manager	1	2	3	4	5
Quality/Process Manager	1	2	3	4	5
Support Manager	1	2	3	4	5

Rate each role for helpfulness and support. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	5
Development Manager	1	2	3	4	5
Planning Manager	1	2	3	4	5
Quality/Process Manager	1	2	3	4	5
Support Manager	1	2	3	4	5

Rate each role for how well it was performed. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	5
Development Manager	1	2	3	4	5

Planning Manager	1	2	3	4	5
Quality/Process Manager	1	2	3	4	5
Support Manager	1	2	3	4	5

### Análisis

El desempeño del grupo ha sido bueno, se han logrado desarrollar todos los productos planificados, aunque no se lograron al 100% los objetivos propuestos. A pesar de pequeños inconvenientes que se pueden haber presentado por la disponibilidad de tiempo, en general el grupo se acopla muy bien.

#### 6.1.8.4. Análisis del ciclo

A continuación se presenta un análisis detallado del ciclo:

	Tiempo	Tamaño			Defectos
		LOC	Páginas	Diagramas	
<b>Actual</b>	179.8	3162	40	51	34
<b>Planificado</b>	167.3	4240	28	51	94

Tabla 30: Análisis del Ciclo I

**% de Avance:** 100

**Desviación del porcentaje:**

$$\text{Desviación del porcentaje} = \frac{\text{Real}}{\text{Planificado}} (100)$$

**% Tiempo:** 107.5%

**% Tamaño:**

- **LOC:** 74.6%
- **Páginas:** 142.9%
- **Diagramas:** 100%

**% Defectos: 36.2%**

### **Error Porcentual**

$$E_{\%} = \frac{\text{Planificado} - \text{Real}}{\text{Planificado}} (100)$$

**% Tiempo: -7.5%**

**% Tamaño:**

- **LOC: 25.4%**
- **Páginas: -42.9%**
- **Diagramas: 0%**

**% Defectos: 63.8%**

- ✓ El tiempo utilizado para el desarrollo del ciclo excedió en un 7.5% al tiempo planificado, por lo que no se logró cumplir con el margen de error deseado de  $\pm 5\%$ .
- ✓ En cuanto al tamaño: El error en el número de LOCs planificadas es de 25.4% más que lo real. El número de páginas planificado es 42.9% menor al real, mientras que el número de diagramas el error porcentual es del 0%. En los dos primeros casos se supera el margen de error deseado de  $\pm 20\%$ .
- ✓ En cuanto a defectos, se cometió un error porcentual de 63.8%. Esto se debió principalmente a que se utilizó un IDE para la implementación, lo cual fue de gran ayuda.

## 6.2. Segundo Ciclo

Los resultados obtenidos durante el segundo ciclo se presentan a continuación:

### 6.2.1. Lanzamiento

En el segundo ciclo se han definido los objetivos del equipo, así como de cada uno de sus integrantes, en base a los resultados obtenidos durante el primer ciclo.

#### 6.2.1.1. Asignación de Roles

Para el segundo ciclo se ha decidido mantener la asignación de roles establecida en el ciclo anterior, como se presenta a continuación:

Roles	Ingeniero
Líder Administrador de Calidad Administrador de Configuraciones	Juan Carlos Fernández Ortiz
Administrador de Planificación Administrador de Desarrollo	Estefanía Narváez

Tabla 31: Asignación de Roles. Ciclo II.

#### 6.2.1.2. Objetivos del Equipo

Los objetivos que el equipo se ha comprometido a cumplir durante el segundo ciclo, son los siguientes:

- ✓ Desarrollar un producto de calidad, inyectando el menor número de defectos posible y siguiendo una planificación, que se ajuste a la realidad, para no exceder del tiempo límite de entrega.
- ✓ Seguir de forma estricta un proceso bien definido, que nos guíe en el desarrollo.
- ✓ Conseguir la reducción de los errores porcentuales de planificación, de tiempo a  $\pm 6\%$ ; estimación de tamaño a  $\pm 30\%$  y de estimación de defectos a  $\pm 50\%$ .
- ✓ Analizar problemas que se presenten durante el proceso, para poder realizar una mejora continua.

- ✓ Colaborar mutuamente para mejorar la productividad de cada uno de los miembros y desarrollar sus habilidades, sin importar si estos cuentan con dichas habilidades al comenzar a desempeñar su rol.
- ✓ Mejora la efectividad de los miembros, optimizando el tiempo de desarrollo del producto, sin afectar a la calidad del mismo.

### ***6.2.1.3. Objetivos de Miembro***

Los objetivos que cada miembro se ha comprometido a cumplir durante este ciclo se describen a continuación:

#### **Juan Carlos Fernández Ortiz**

- ✓ Apoyar a todos los miembros del equipo sin que esto signifique absorber sus responsabilidades.
- ✓ Promover una comunicación adecuada entre todos los miembros del equipo.
- ✓ Realizar el trabajo de forma efectiva, con el fin de obtener productos de calidad.
- ✓ Hacer lo posible por solucionar los problemas de grupo o al menos disminuir considerablemente sus efectos.

#### **Estefanía Narváez Bastidas**

- ✓ Inyectar el menor número de defectos en el desarrollo de los productos y tratar de reducir constantemente, los que se inyecten, estudiando las posibles causas y soluciones.
- ✓ Cumplir de la forma más estricta posible con el plan que se ha definido y las tareas asignadas.
- ✓ Ayudar a todos los integrantes con las dudas que se les puedan presentar, para mejorar el trabajo del equipo.

### ***6.2.1.4. Objetivos del Rol***

Los objetivos que cada miembro se ha comprometido en cumplir para desempeñar de mejor manera su rol durante este ciclo, se describen a continuación:

### **Líder (Juan Carlos Fernández Ortiz)**

- ✓ Asegurar que todos los miembros del equipo cumplan con la planificación.
- ✓ Escuchar a todos los miembros del equipo y tomar en cuenta sus opiniones para la toma de decisiones.
- ✓ Verificar que todos los miembros del equipo estén realizando una carga de trabajo similar, de manera que se sientan comprometidos con el proyecto.

### **Administrador de Desarrollo (Estefanía Narváez Bastidas)**

- ✓ Transmitir de forma precisa los requerimientos del sistema a todos los miembros del equipo.
- ✓ Identificar, desarrollar y aprovechar las habilidades y destrezas con las que cuenta cada miembro del equipo.

### **Administrador de Planificación (Estefanía Narváez Bastidas)**

- ✓ Desarrollar un plan de trabajo, que abarque todas las tareas que se deben cumplir y que se ajuste, en lo posible, a la realidad.
- ✓ Motivar al equipo a seguir la planificación y a hacer un seguimiento de su trabajo correctamente.

### **Administrador de Calidad (Juan Carlos Fernández Ortiz)**

- ✓ Revisar, y de ser necesario implementar mejoras en los estándares, que permitan al equipo trabajar de forma ordenada y revisar que se cumpla con éstos a fin de obtener un producto de calidad.
- ✓ Realizar un control de calidad de todos los productos o artefactos antes de que pasen a línea base.
- ✓ Mantener registro con todos los asuntos tratados en cada reunión.

### **Administrador de Configuraciones (Juan Carlos Fernández Ortiz)**

- ✓ Vigilar que no se realice ningún cambio en documentos de línea base sin que se haya seguido el proceso adecuado, que se deberá también diseñar.
- ✓ Llevar control de todos los riesgos y asuntos que se presenten para poder darles un mejor seguimiento de manera semanal.

- ✓ Administrar todos los artefactos reusables que puedan ayudar al equipo en fases posteriores y en el siguiente ciclo.

#### 6.2.1.5. *Objetivos de Calidad*

Desarrollar un producto de calidad, con el menor número de defectos posibles evitando sobrepasar los defectos estimados que se presentan a continuación:

Ciclo	Defectos
Primero	34 (reales)
Segundo	20 (estimados)

Tabla 32: **Objetivos de Calidad del Producto**

#### 6.2.2. **Estrategia de Desarrollo**

De acuerdo con la estrategia planificada inicialmente, una vez terminado el primer ciclo, se da paso al segundo.

##### 6.2.2.1. *Criterios de Estrategia*

Durante el primer ciclo, se desarrollaron las siguientes funcionalidades:

Ciclo	Funcionalidades
Primero	F0: Ingresar mediante usuario y contraseña. F1: Administrar usuarios. F2: Administrar perfiles. F3: Administrar estudiantes. F4: Administrar quimestres. F5: Administrar habilidades principales.

Ciclo	Funcionalidades
	F6: Administrar habilidades específicas. F7: Administrar tareas. F8: Administrar talleres.

**Tabla 33: Funciones desarrolladas durante el Ciclo I**

Durante el segundo ciclo, se desarrollarán las siguientes funcionalidades:

Ciclo	Funcionalidades
Segundo	F9: Administrar asignación de profesores a talleres F10: Administrar evaluaciones iniciales. F11: Administrar evaluaciones mensuales. F12: Obtener reportes de las evaluaciones a los estudiantes.

**Tabla 34: Funciones Planificadas para el Ciclo II**

#### **6.2.2.2. Estrategia de Reuso**

Los componentes que se han identificado como aptos para la reutilización en el segundo ciclo, son los siguientes:

- ✓ Documentos de Lanzamiento
  - Objetivos del equipo.
  - Objetivos de miembros.
  - Objetivos de roles.
  - Objetivos de calidad.
  - Gestión de configuración del software.
  - Estándares de documentación y comunicación.
  
- ✓ Documentos de Estrategia

- Estrategia de desarrollo.
- Estrategias de reuso.
- STRAT.
  
- ✓ Documentos de Planificación
  - Formas.
  
- ✓ Documentos de Requerimientos
  - Requerimientos no funcionales.
  - Plan de pruebas del sistema.
  
- ✓ Documentos de Diseño
  - Diagrama de clases.
  - Diagrama entidad relación.
  - Plan de pruebas de integración.
  
- ✓ Documento de Implementación
  - Estándares de codificación.
  - Código.

### 6.2.3. Planificación

A continuación, se presenta la distribución de módulos para el presente ciclo.

#### 6.2.3.1. Distribución de Módulos

Los módulos que debe desarrollarse durante el presente ciclo se han distribuido entre los miembros del equipo de la siguiente forma:

Módulo	Responsable
F9: Administrar asignación de profesores a talleres	Juan Carlos Fernández Ortiz
F10: Administrar evaluaciones iniciales.	Estefanía Narváez

Módulo	Responsable
F11: Administrar evaluaciones mensuales.	Juan Carlos Fernández Ortiz
F12: Obtener reportes de las evaluaciones a los estudiantes.	Estefanía Narváez

Tabla 35: Distribución de Módulos para el Ciclo II

#### 6.2.4. Definición de Requerimientos

En esta fase se especifican los requerimientos del usuario con respecto al producto final. A continuación se presenta el documento de especificación de requerimientos del sistema.

#### *Especificación de Requerimientos del Sistema (SRS)*

### I Introducción

#### 1.4. Objetivo

El presente documento contiene todos los requerimientos de forma clara, precisa y no ambigua, con la finalidad de ser revisado y aceptado por el cliente, para verificar que el producto cumplirá con sus necesidades. Este documento constituye la base sobre la cual se llevarán a cabo las siguientes fases para el desarrollo del producto.

#### 1.5. Definición General del Sistema

El presente sistema, pretende automatizar el seguimiento del desarrollo de las habilidades en las personas con discapacidad mental, optimizando el proceso pedagógico.

Para el ciclo actual se implementarán las siguientes funcionalidades:

##### a. Administrar asignación de profesores a talleres

Esta funcionalidad permitirá asignar los profesores que serán responsables de cada taller.

**b. Administrar evaluaciones iniciales**

Esta funcionalidad permitirá consultar, registrar, modificar y eliminar la información de la evaluación inicial que se aplica a cada persona con discapacidad al ingresar al centro.

**c. Administrar evaluaciones mensuales**

Esta funcionalidad permitirá consultar, crear, evaluar, modificar y eliminar la información de las evaluaciones mensuales que se aplica a cada persona con discapacidad.

**d. Obtener reportes de las evaluaciones a los estudiantes**

Esta funcionalidad permitirá generar reportes de las evaluaciones que se realicen a las personas con discapacidad.

### 1.6. Información de Equipo

A continuación se presenta información básica sobre el equipo de trabajo.

Integrante	Rol	Distribución de Módulos
Juan Carlos Fernández	Líder. Administrador de Calidad. Administrador de Configuraciones.	<b>F9:</b> Administrar asignación de profesores a talleres. <b>F11:</b> Administrar evaluaciones mensuales.
Estefanía Narváez	Administrador de Planificación. Administrador de Desarrollo.	<b>F10:</b> Administrar evaluaciones iniciales. <b>F12:</b> Obtener reportes de las evaluaciones a los estudiantes.

Tabla 36: Información del Equipo. Ciclo II.

## II Requerimientos Funcionales

### 2.3. Diagrama de Casos de Uso

#### 2.3.1. Diagrama de Casos de Uso General

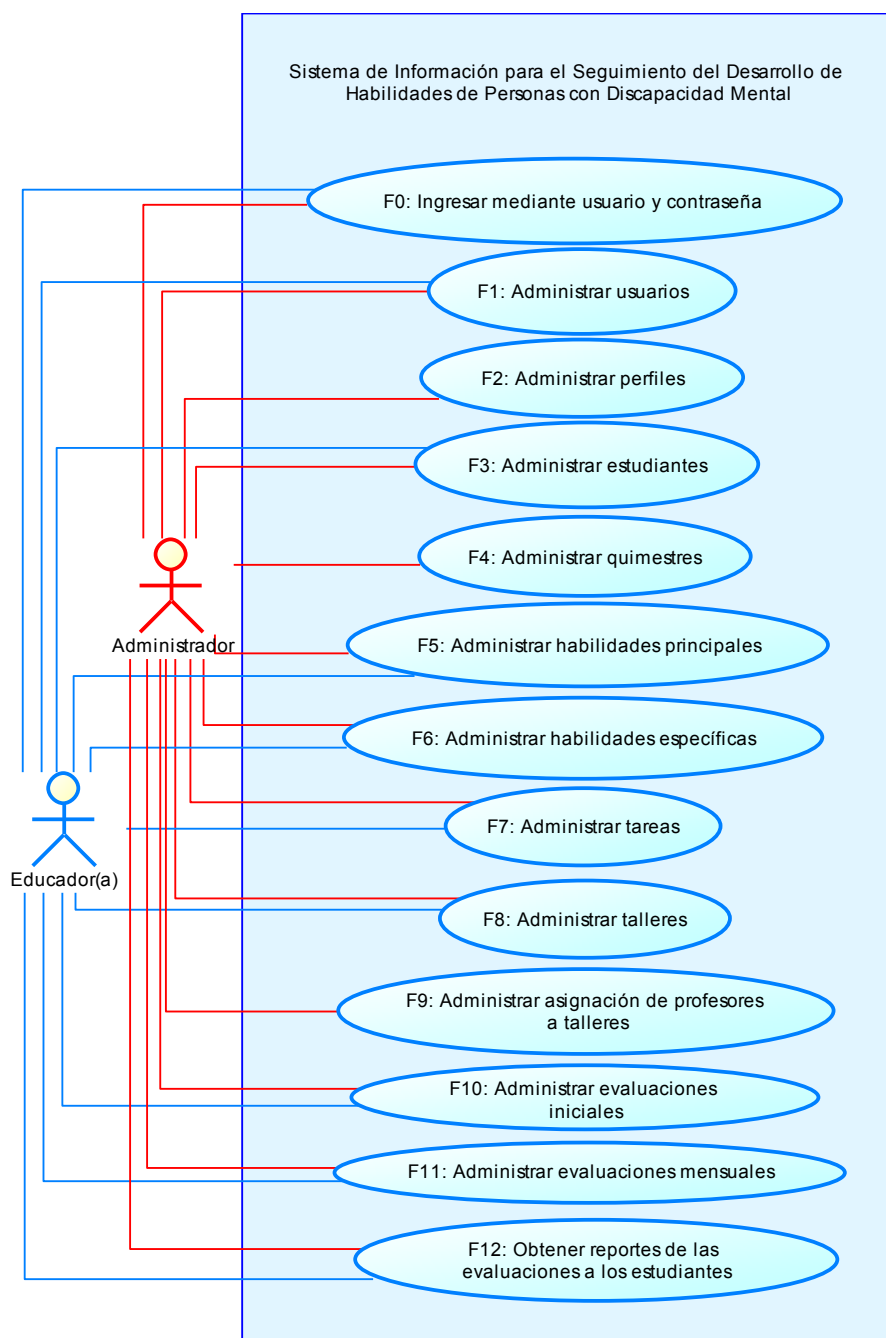


Ilustración 50: Diagrama de Casos de Uso General

## 2.3.2. Diagrama de Casos de Uso Siguiete Nivel

### 2.3.2.1. F9: Administrar asignación de profesores a talleres

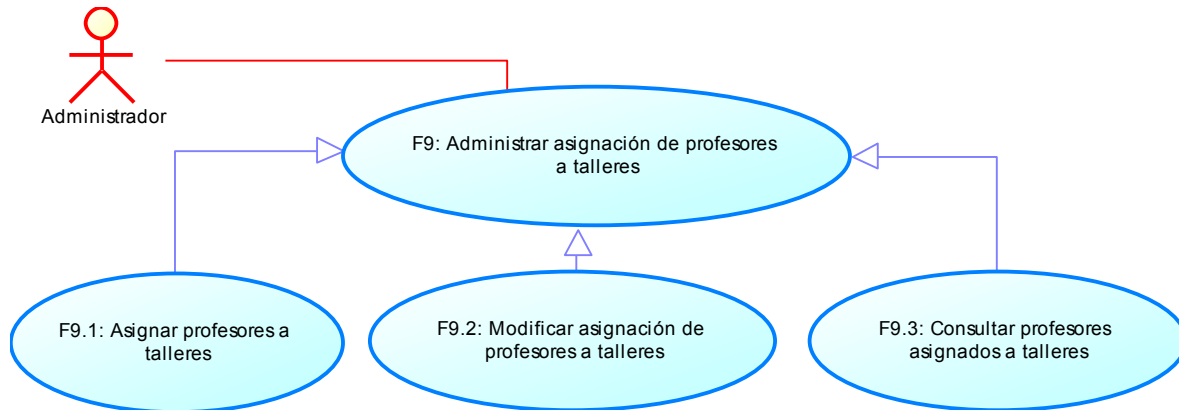


Ilustración 51: Diagrama de Casos de Uso Siguiete Nivel de F9

### 2.3.2.2. F10: Administrar evaluaciones iniciales

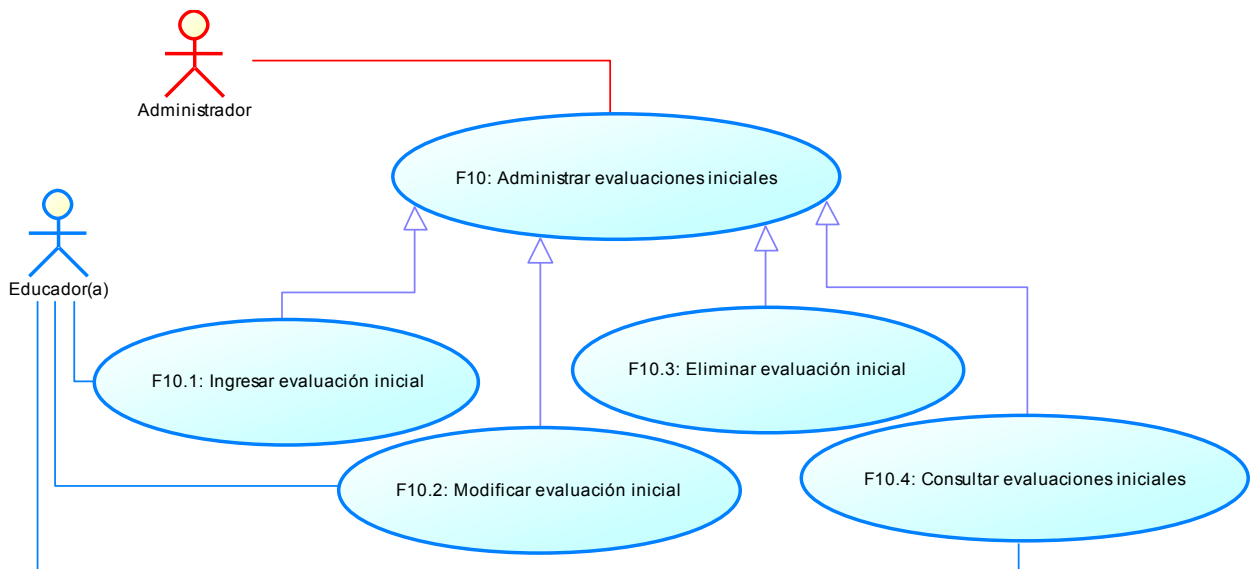
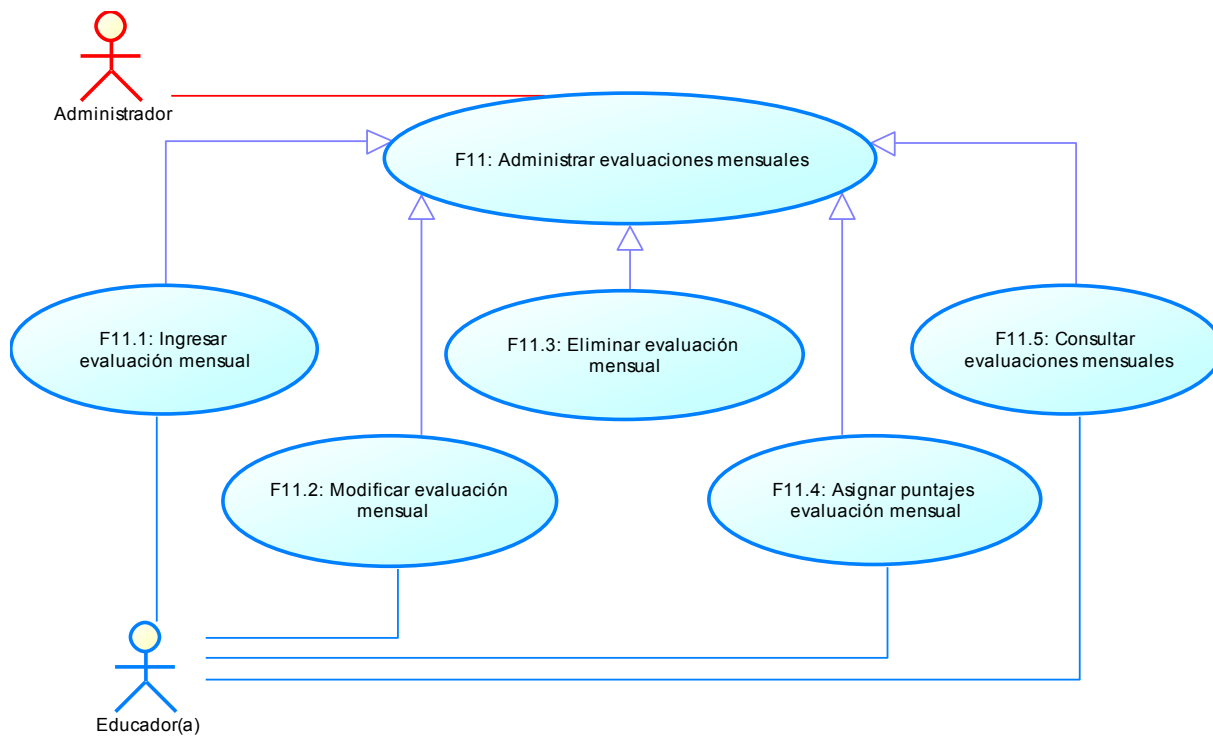


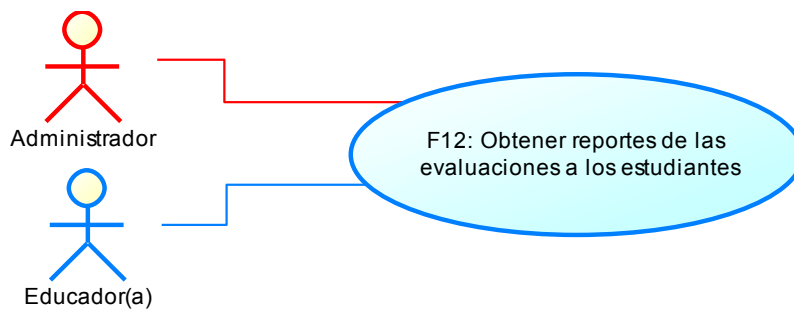
Ilustración 52: Diagrama de Casos de Uso Siguiete Nivel de F10

**2.3.2.3. F11: Administrar evaluaciones mensuales.**



**Ilustración 53: Diagrama de Casos de Uso Siguiendo Nivel de F11**

**2.3.2.4. F12: Obtener reportes de las evaluaciones a los estudiantes**



**Ilustración 54: Diagrama de Casos de Uso Siguiendo Nivel de F12**

### 2.3.3. Diagramas de Casos de Uso A Detalle

A continuación se presentan los casos de uso a detalle de las funcionalidades F10: Administración de Evaluaciones Iniciales y F11: Administración de Evaluaciones Mensuales. Los casos de uso a detalle de las funcionalidades restantes se encuentran en el anexo “Casos de Uso a Detalle” del segundo ciclo.

#### 2.3.3.1. F10.1: Ingresar evaluación inicial

**Descripción:** Función que permite registrar los resultados obtenidos por un estudiante en la evaluación de habilidades básicas.

**Actores:** Administrador, Educador(a).

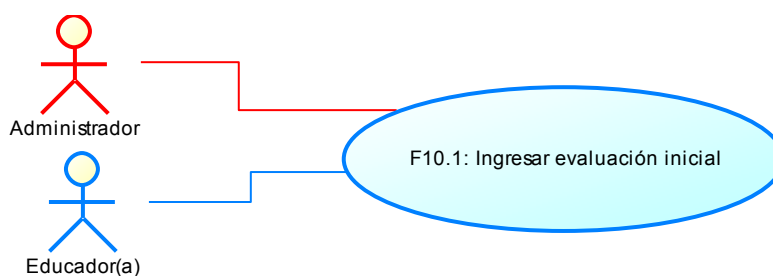


Ilustración 55: Diagrama de Caso de Uso a Detalle de F10.1

#### Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluaciones Iniciales del menú principal.	2	Presenta una página que permite la administración de evaluaciones iniciales.	E1
3	Selecciona el estudiante a evaluar.			
4	Ingresa los resultados de la evaluación.			
5	Ingresa observaciones.			

Paso	Actor	Paso	Sistema	Excepción
6	Presiona el botón Guardar.	7	Valida la información.	
		8	Almacena la información.	E1

Tabla 37: Flujo Principal de F10.1

**Flujo Alternativo:**

Paso	Caso de Uso
3	3.1
4	8.1
7	F10.2 ó 10.3

Tabla 38: Flujo Alternativo de F10.1

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 39: Excepciones de F10.1

**2.3.3.2. F10.2: Modificar evaluación inicial.**

**Descripción:** Función que permite modificar la información de una evaluación inicial.

**Actores:** Administrador.

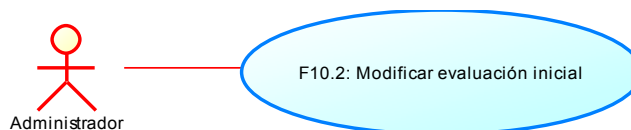


Ilustración 56: Diagrama de Casos de Uso a Detalle de F10.2

### Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluaciones Iniciales del menú principal.	2	Presenta una página que permite la administración de evaluaciones iniciales.	E1
3	Selecciona la evaluación inicial que desea modificar.	4	Carga la información de la evaluación seleccionada.	E1
5	Modifica la información.			
6	Presiona el botón Guardar.	7	Valida la información.	
		8	Almacena la información.	E1

Tabla 40: Flujo Principal de F10.2

### Flujo Alternativo:

Paso	Caso de Uso
3	3.1
4	8.1
3	F10.1

Tabla 41: Flujo Alternativo de F10.2

### Excepciones:

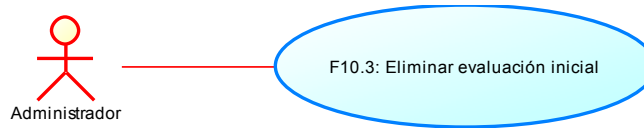
Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 42: Excepciones de F10.2

**2.3.3.3. F10.3: Eliminar evaluación inicial.**

**Descripción:** Función que permite desactivar una evaluación inicial.

**Actores:** Administrador.



**Ilustración 57: Diagrama de Casos de Uso a Detalle de F10.3**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluaciones Iniciales del menú principal.	2	Presenta una página que permite la administración de evaluaciones iniciales.	E1
3	Selecciona la evaluación inicial a eliminar.	4	Elimina la evaluación seleccionada.	E1

**Tabla 43: Flujo Principal de F10.3**

**Flujo Alternativo:**

Paso	Caso de Uso
3	F10.1

**Tabla 44: Flujo Alternativo de F10.3**

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 45: Excepciones de F10.3

**2.3.3.4. F10.4: Consultar evaluaciones iniciales.**

**Descripción:** Función que permite consultar la información de las evaluaciones iniciales.

**Actores:** Administrador, Educador(a).

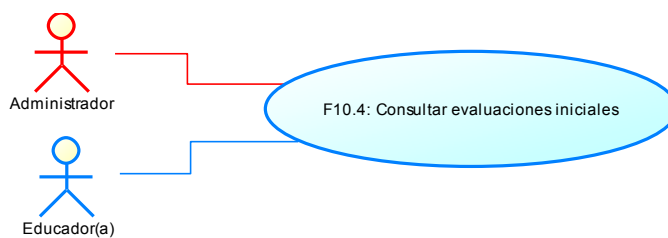


Ilustración 58: Diagrama de Casos de Uso a Detalle de F10.4

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluaciones Iniciales del menú principal.	2	Presenta una página con las evaluaciones iniciales.	E1

Tabla 46: Flujo Principal de F10.4

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 47: Excepciones de F10.4

**2.3.3.5. F11.1: Ingresar evaluación mensual**

**Descripción:** Función que permite crear una evaluación de las tareas que se evaluarán en un taller durante un quimestre.

**Actores:** Administrador, Educador(a).

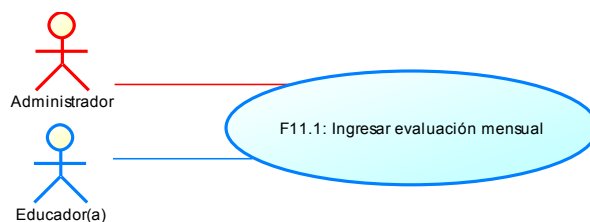


Ilustración 59: Diagrama de Caso de Uso a Detalle de F11.1

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Evaluaciones Mensuales del menú principal.	2	Presenta una página que permite la administración de evaluaciones mensuales.	E1
3	Selecciona el quimestre para el cual se va a crear la evaluación			
4	Selecciona el estudiante a evaluar.			
5	Selecciona el taller a evaluar			

Paso	Actor	Paso	Sistema	Excepción
6	Presiona el botón Continuar.	7	Presenta las tareas del taller seleccionado	E1
8	Selecciona las tareas a evaluar.			
9	Presiona el botón Guardar.	10	Almacena la información.	E1

Tabla 48: Flujo Principal de F11.1

**Flujo Alternativo:**

Paso	Caso de Uso
3	F4.1
4	F3.1
5	F8.1, F9.1 ó F9.2

Tabla 49: Flujo Alternativo de F11.1

**Excepciones:**

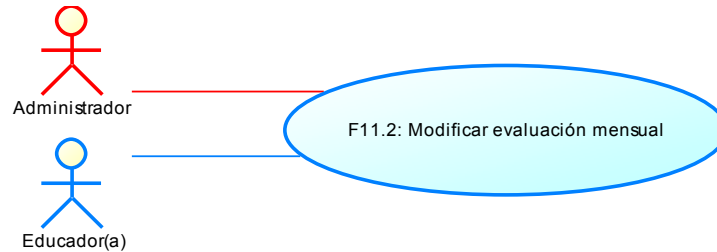
Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 50: Excepciones de F11.1

**2.3.3.6. F11.2: Modificar evaluación mensual.**

**Descripción:** Función que permite modificar la evaluación que se realizará durante un quimestre para un estudiante.

**Actores:** Administrador, Educador(a).



**Ilustración 60: Diagrama de Casos de Uso a Detalle de F11.2**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Evaluaciones Mensuales del menú principal.	2	Presenta una página que permite la administración de evaluaciones mensuales.	E1
3	Selecciona el quimestre para el cual se va a modificar la evaluación			
4	Selecciona el estudiante a cuya evaluación se va a modificar.			
5	Se selecciona el taller cuya evaluación se va a modificar			
6	Presiona el botón Continuar	7	Se presentan las tareas del taller seleccionado y se cargan las tareas seleccionadas	E1

Paso	Actor	Paso	Sistema	Excepción
			anteriormente	
<b>8</b>	Se modifica la información.			
<b>9</b>	Presiona el botón Guardar.	<b>10</b>	Almacena la información.	E1

**Tabla 51: Flujo Principal de F11.2**

**Flujo Alternativo:**

Paso	Caso de Uso
<b>3</b>	F4.1 ó F4.2
<b>4</b>	F3.1
<b>5</b>	F8.1, F8.2, F9.1 ó F9.2

**Tabla 52: Flujo Alternativo de F11.2**

**Excepciones:**

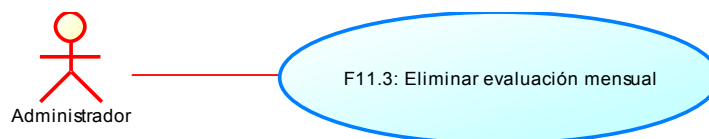
Código	Mensaje	Alternativa
<b>E1</b>	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

**Tabla 53: Excepciones de F11.2**

**2.3.3.7. F11.3: Eliminar evaluación mensual.**

**Descripción:** Función que permite desactivar una evaluación mensual.

**Actores:** Administrador.



**Ilustración 61: Diagrama de Casos de Uso a Detalle de F11.3**

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Administrar Evaluaciones Mensuales del menú principal.	2	Presenta una página que permite la administración de evaluaciones mensuales.	E1
3	Selecciona el quimestre para el cual se va a eliminar la evaluación			
4	Selecciona el estudiante a cuya evaluación se va a eliminar.			
5	Se selecciona el taller cuya evaluación se va a eliminar.			
6	Presiona el botón Continuar	7	Se presentan las tareas del taller seleccionado	E1
8	Presiona el botón Eliminar.	9	Elimina la información.	E1

**Tabla 54: Flujo Principal de F11.3**

### Flujo Alternativo:

Paso	Caso de Uso
3	F4.1 ó F4.2
4	F3.1
5	F8.1, F8.2, F9.1 ó F9.2

Tabla 55: Flujo Alternativo de F11.3

### Excepciones:

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 56: Excepciones de F11.3

#### 2.3.3.8. F11.4: Asignar puntajes evaluación mensual.

**Descripción:** Función que permite asignar puntajes a una evaluación mensual de un estudiante.

**Actores:** Administrador, Educador(a).

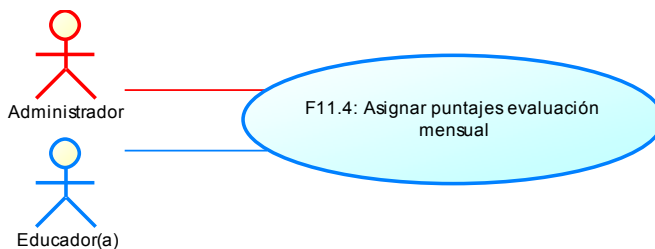


Ilustración 62: Diagrama de Casos de Uso a Detalle de F11.4

**Flujo Principal:**

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluar del menú principal.	2	Presenta una página que permite asignar puntajes.	E1
3	Selecciona el quimestre para el cual se va a asignar puntaje			
4	Selecciona el estudiante a cuya evaluación se va asignar puntaje.			
5	Selecciona el taller a evaluar.			
6	Presiona el botón Continuar	7	Presenta las tareas del taller seleccionado.	E1
8	Ingresa los puntajes			
9	Presiona el botón Guardar	10	Almacena la información.	E1

Tabla 57L Flujo Principal de F11.4

**Flujo Alternativo:**

Paso	Caso de Uso
3	F4.1 ó F4.2
4	F3.1
5	F8.1, F8.2, F9.1 ó F9.2

Tabla 58: Flujo Alternativo de F11.4

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con	Refrescar la página. Si el problema

Código	Mensaje	Alternativa
	la BDD.	persiste, contacte al departamento técnico.

Tabla 59: Excepciones de F11.4

### 2.3.3.9. F11.5: Consultar evaluaciones mensuales.

**Descripción:** Función que permite consultar la información de las evaluaciones mensuales.

**Actores:** Administrador, Educador(a).

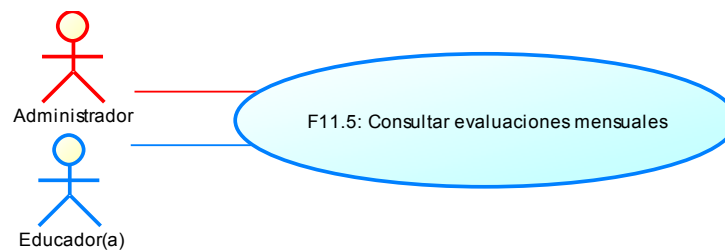


Ilustración 63: Diagrama de Casos de Uso a Detalle de F11.5

#### Flujo Principal:

Paso	Actor	Paso	Sistema	Excepción
1	Selecciona la opción Evaluar del menú principal.	2	Presenta una página que permite asignar puntajes.	E1
3	Selecciona el quimestre para el cual se va a asignar puntaje			
4	Selecciona el estudiante a cuya evaluación se va asignar puntaje.			
5	Selecciona el taller a evaluar.			
6	Presiona el botón Continuar	7	Presenta las tareas del taller seleccionado con sus	E1

Paso	Actor	Paso	Sistema	Excepción
			respectivos puntajes.	

Tabla 60: Flujo Principal F11.5

**Flujo Alternativo:**

Paso	Caso de Uso
3	F4.1 ó F4.2
4	F3.1
5	F8.1, F8.2, F9.1 ó F9.2

Tabla 61: Flujo Alternativo de F11.5

**Excepciones:**

Código	Mensaje	Alternativa
E1	Problema de conexión con la BDD.	Refrescar la página. Si el problema persiste, contacte al departamento técnico.

Tabla 62: Excepciones de F11.5

**2.4. Diagramas de Clases a Nivel Conceptual**

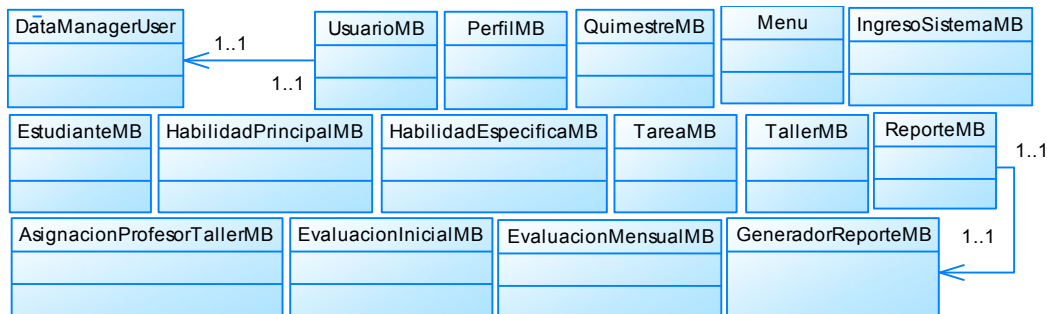


Ilustración 64: Diagrama de clases conceptual Managed Beans



**Estándares de Codificación:** Basados en convenciones de código para el lenguaje Java

## IV Otros Requerimientos

### 4.1. De Interface Externa

La interface externa se desarrollará de acuerdo al siguiente diseño:

The screenshot shows a web application interface for 'Fundación para la Integración del Niño Especial' (FINE). The header includes the FINE logo and the text 'Fundación para la Integración del Niño Especial' and 'SISDHAB'. A navigation bar shows 'Inicio' and 'Administración'. The main content area is titled 'Asignación de Educadoras(es) a Talleres'. It features a dropdown menu for 'Educador(a):' with 'Suasnavas Mery' selected. Below this is a 'Talleres' section with a table of available workshops. The first workshop, 'TALLER No. 01 : ACTIVIDADES DE LA VIDA DIARIA', is selected. Below the table are buttons for 'Guardar', 'Modificar', and 'Cancelar'. At the bottom, there is a table of assigned educators with columns for 'Educador(a)' and 'Acciones'.

Educador(a) ↕	Acciones
Suasnavas Mery	👁
Fernández Ortiz Juan Carlos	👁

**Ilustración 67: Administración de Asignación de Profesoras (res) a Talleres**



### Administración de Evaluaciones Iniciales

Estudiante:

+ Aciertos

- Observaciones

254 caracteres restantes.

Estudiante	Acciones
No existen coincidencias	

Ilustración 68: Administración de Evaluaciones Iniciales

Fundación para la Integración del Niño Especial

SISDHAB

Inicio > Administración >

Administración de Evaluaciones Mensuales

Quimestre: Primer Quimestre 2014-2015

Estudiante: Narváez Bastidas Estefanía

Taller: TALLER No. 01 : ACTIVIDADES DE LA VIDA DIARIA

Continuar

Tareas

Tarea
<input type="checkbox"/> Regular la temperatura del agua
<input type="checkbox"/> Lavarse las manos y la cara
<input type="checkbox"/> Secarse las manos y la cara después de lavarse
<input type="checkbox"/> Lavarse las axilas y utilizar desodorante
<input type="checkbox"/> Ducharse
<input checked="" type="checkbox"/> Lavarse y secarse el pelo
<input type="checkbox"/> Peinarse y/o arreglarse el pelo
<input type="checkbox"/> Lavarse los dientes y enjuagarse la boca

1 2 3 4 5 6 7 8 9 10

Guardar Modificar Eliminar Cancelar

©2014 Fundación para la Integración del Niño Especial

**Ilustración 69: Administración de Evaluaciones Mensuales**

Fundación para la Integración del Niño Especial

SISDHAB

Inicio > Administración > Evaluar > Reportes >

Reporte Quimestral

Quimestre: Seleccione

Estudiante: Seleccione

Taller: Seleccione

Generar Reporte

©2014 Fundación para la Integración del Niño Especial

**Ilustración 70: Reporte Quimestral**

## **4.2. De Hardware**

**RAM (mínima):** 2 GB

**RAM (recomendada):** 4 GB

**Procesador:** Intel Pentium IV de 2,6 GHz o equivalente.

**Espacio en Libre en Disco (mínimo):** 500 MB

**Espacio en Libre en Disco (recomendado):** 1 GB

## **4.3. De Software**

Navegador de Internet:

- ✓ Mozilla Firefox: versión 27 o superior,
- ✓ Google Chrome: versión 33 o superior, o
- ✓ Internet Explorer: versión 11 o superior.

JDK 1.6

JBoss 7.1

MySQL 5.6

## **V Plan de Pruebas del Sistema**

Las pruebas del sistema, son similares a las pruebas de caja negra, es decir, que se enfocan en los requerimientos establecidos y en las funcionalidades, pero éstas lo hacen a nivel de resultados, como un todo, abarcando las partes combinadas del mismo.

Su principal objetivo, es verificar que el sistema cumpla con los requerimientos, por lo que es sumamente importante que éstos sean documentados y medibles.

A continuación se presenta el plan de pruebas del sistema para las funcionalidades F10: Administración de Evaluaciones Iniciales y F11: Administración de Evaluaciones Mensuales. El plan de pruebas del sistema que abarca todas las funcionalidades se encuentran en el anexo “Plan de Pruebas del Sistema” del segundo ciclo.

#### **Prueba 4:**

- ✓ **Funcionalidad:** F10.1
- ✓ **Evaluador:** Juan Carlos Fernández Ortiz
- ✓ **Descripción:** Evaluar el correcto funcionamiento del ingreso de evaluaciones iniciales.
- ✓ **Diseño:**
  1. Ingresar una evaluación inicial, llenando todos los campos obligatorios.
  2. Ingresar una evaluación inicial, dejando en blanco los campos obligatorios.
- ✓ **Resultados Esperados:**
  1. El sistema almacena correctamente la información ingresada.
  2. El sistema emite un mensaje por cada campo obligatorio que se deje vacío después de presionar guardar. Presenta el mensaje de alerta, hasta que la información esté completa.

#### **Prueba 5:**

- ✓ **Funcionalidad:** F10.2
- ✓ **Evaluador:** Juan Carlos Fernández Ortiz
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la modificación de evaluaciones iniciales.
- ✓ **Diseño:**
  1. Modificar una evaluación inicial existente.
- ✓ **Resultados Esperados:**
  1. El sistema almacena correctamente los campos, sin alterar el código único de la entidad.

#### **Prueba 6:**

- ✓ **Funcionalidad:** F10.3
- ✓ **Evaluador:** Juan Carlos Fernández Ortiz
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la eliminación de evaluaciones iniciales.
- ✓ **Diseño:**
  1. Eliminar una evaluación inicial existente.
- ✓ **Resultados Esperados:**
  1. La evaluación inicial se elimina correctamente.

#### **Prueba 7:**

- ✓ **Funcionalidad:** F10.4
- ✓ **Evaluador:** Juan Carlos Fernández Ortiz
- ✓ **Descripción:** Evaluar el correcto funcionamiento de la consulta de evaluaciones iniciales.
- ✓ **Diseño:**
  1. Consultar de manera general las evaluaciones iniciales ingresadas
- ✓ **Resultados Esperados:**
  1. El sistema presenta los registros que han sido ingresados al sistema.

#### **Prueba 8:**

- ✓ **Funcionalidad:** F11.1
- ✓ **Evaluador:** Estefanía Narváez
- ✓ **Descripción:** Evaluar el correcto funcionamiento del ingreso de evaluación mensual.

✓ **Diseño:**

1. Ingresar una evaluación mensual, llenando todos los campos obligatorios.
2. Ingresar una evaluación mensual, dejando en blanco los campos obligatorios.

✓ **Resultados Esperados:**

1. El sistema almacena correctamente la información ingresada.
2. El sistema emite un mensaje por cada campo obligatorio que se deje vacío después de presionar guardar. Presenta el mensaje de alerta, hasta que la información esté completa.

**Prueba 9:**

✓ **Funcionalidad:** F11.2

✓ **Evaluador:** Estefanía Narváez

✓ **Descripción:** Evaluar el correcto funcionamiento de la modificación de evaluaciones mensuales.

✓ **Diseño:**

1. Modificar un estudiante existente.

✓ **Resultados Esperados:**

1. El sistema almacena correctamente los campos, sin alterar el código único de la entidad.

**Prueba 10:**

✓ **Funcionalidad:** F11.3

✓ **Evaluador:** Estefanía Narváez

✓ **Descripción:** Evaluar el correcto funcionamiento de la eliminación de evaluaciones mensuales.

✓ **Diseño:**

1. Eliminar una evaluación mensual existente.

✓ **Resultados Esperados:**

1. La evaluación se elimina correctamente.

**Prueba 11:**

✓ **Funcionalidad:** F11.4

✓ **Evaluador:** Estefanía Narváez.

✓ **Descripción:** Evaluar el correcto funcionamiento de la asignación del puntaje de la evaluación mensual.

✓ **Diseño:**

1. Asignar puntaje a una evaluación mensual existente.

✓ **Resultados Esperados:**

1. El sistema almacena la información correctamente.

**Prueba 12:**

✓ **Funcionalidad:** F11.5

✓ **Evaluador:** Estefanía Narváez.

✓ **Descripción:** Evaluar el correcto funcionamiento de la consulta de evaluaciones mensuales.

✓ **Diseño:**

1. Consultar de manera general los registros ingresados

✓ **Resultados Esperados:**

1. El sistema presenta los registros que han sido ingresados al sistema.

## 6.2.5. Diseño

A continuación se presenta el producto generado en la fase de diseño del segundo ciclo:

### *Especificación de Diseño del Sistema (SDS)*

#### 1. Diseño Arquitectónico

La arquitectura de software es una especificación de los subsistemas y de la forma en la que se relacionan entre ellos. Es una estructura jerárquica de los componentes que especifica la manera en que interactúan dichos componentes y la estructura de datos que utilizan.

##### 1.1. Arquitectura EJB

En la definición de una arquitectura se deben tener en cuenta los conceptos de cohesión y acoplamiento. Cohesión nos indica la facilidad con que un componente puede unirse con otros, mientras que el acoplamiento, es responsable de evaluar las relaciones que se establecen entre las partes; de tal forma que un nivel bajo de acoplamiento permitiría la reutilización de código.

Como muchas tecnologías, Enterprise Java Beans separa 3 conceptos fundamentales que son:

- ✓ **Capa de Presentación o Interfaz de Usuario:** Está formada por las páginas con las que interactúa directamente el usuario y los controles que se encuentran en dichas páginas.
- ✓ **Capa de Negocio:** Está formada por los session beans que representan objetos que van a ser gestionados por la aplicación. Contiene el código al que recurre la capa de presentación para recuperar los datos deseados.
- ✓ **Capa de Acceso a Datos:** Contiene clases que interactúan con la base de datos y permiten realizar todas las operaciones de forma transparente para la segunda capa.

## 1.2. Diagrama de Paquetes

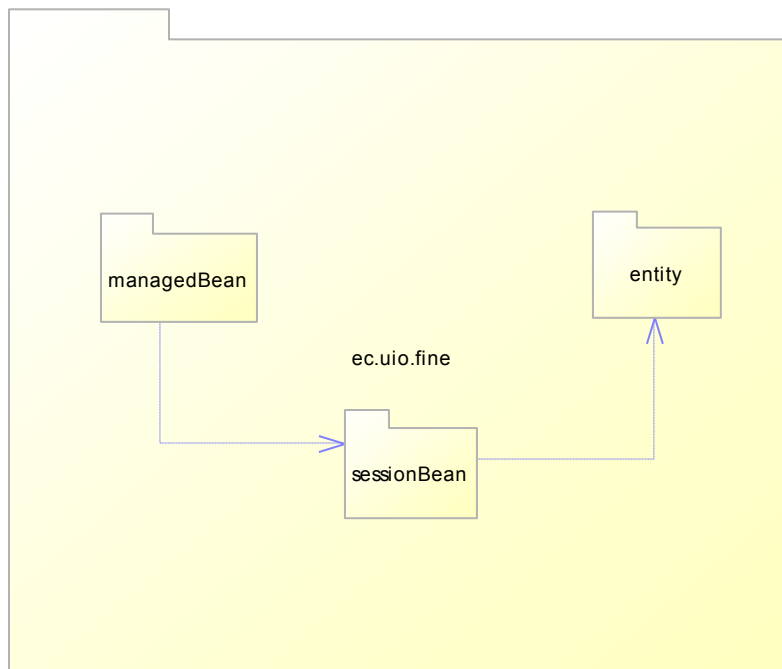


Ilustración 71: Diagrama de paquetes



## 2.2. Clases de Session Beans

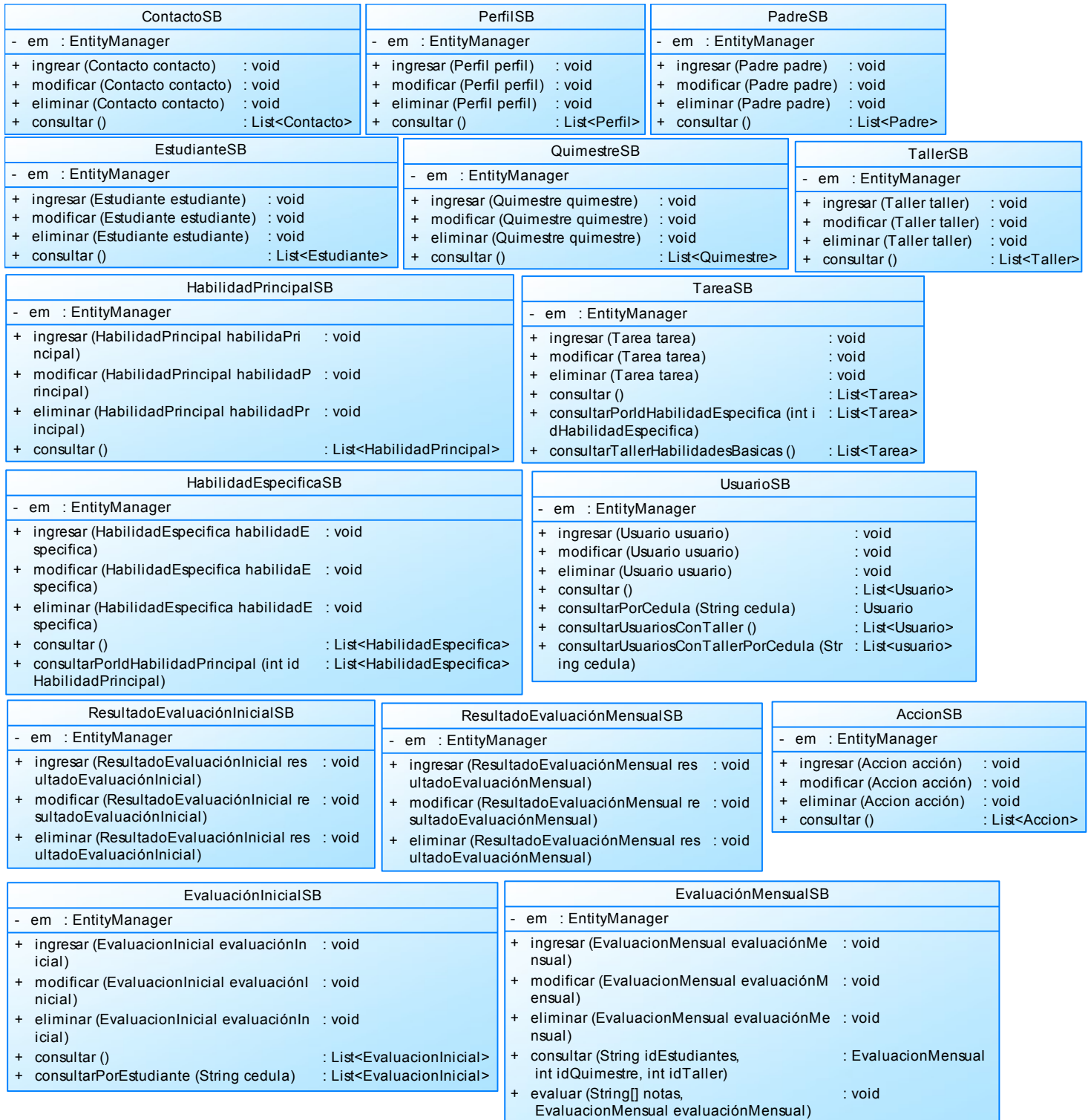


Ilustración 73: Diagrama de Clases de Session Beans a Detalle

### 2.3. Clases de Entities

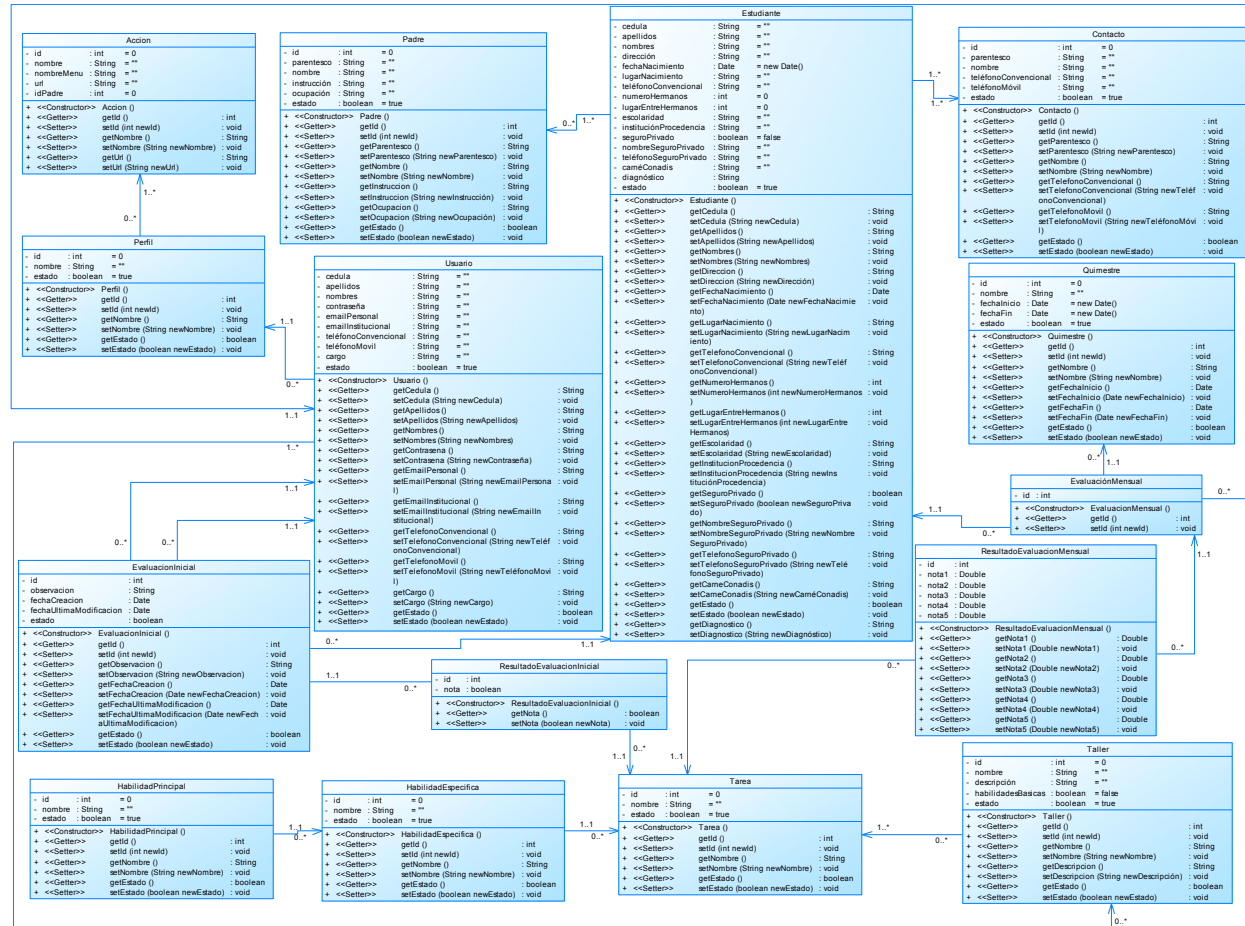


Ilustración 74: Diagrama de Clases de Entities a Detalle

### 3. Diagramas de Secuencia

A continuación se presentan los diagramas de secuencia de las funcionalidades F10: Administración de Evaluaciones Iniciales y F11: Administración de Evaluaciones Mensuales. Los diagramas de secuencia de las funcionalidades restantes se encuentran en el anexo “Diagramas de Secuencia” del segundo ciclo.

#### 3.1. F10.1: Ingresar evaluación inicial.

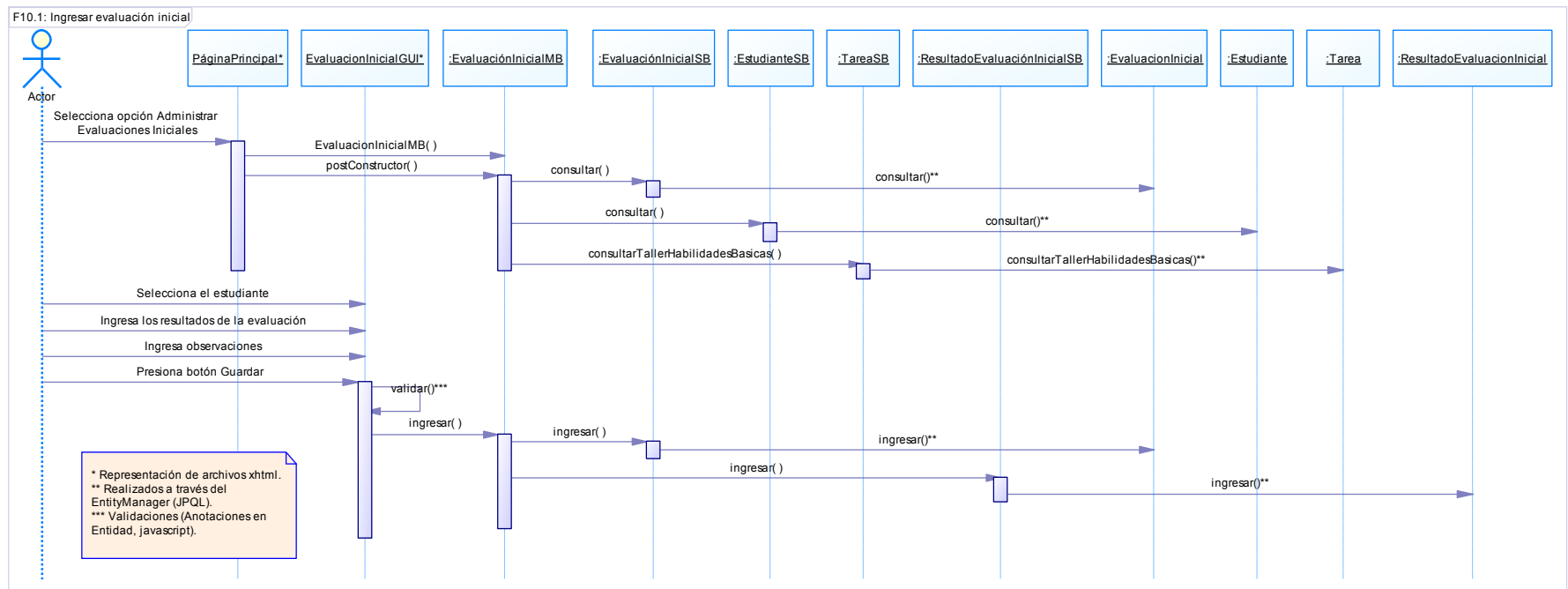


Ilustración 75: Diagrama de Secuencia F10.1

### 3.2. F10.2: Modificar evaluación inicial.

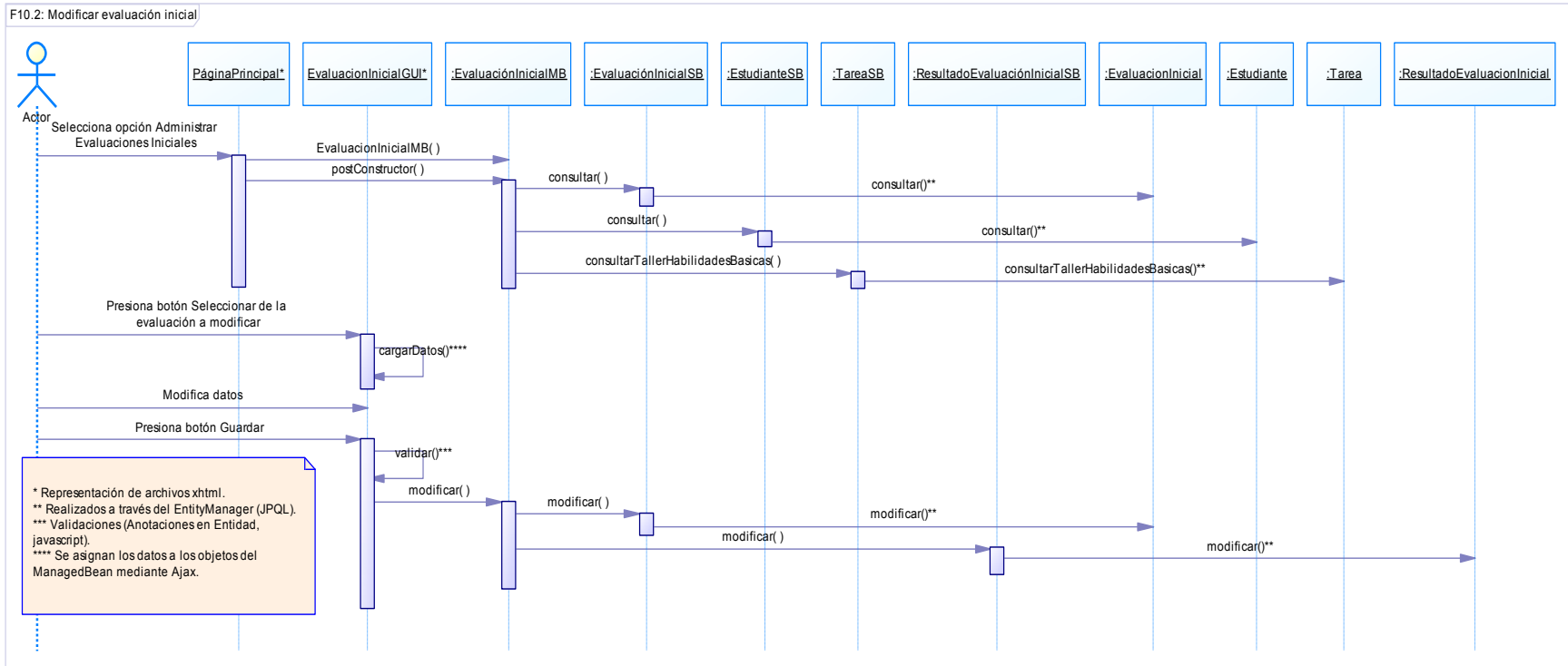


Ilustración 76: Diagrama de Secuencia F10.2

### 3.3. F10.3: Eliminar evaluación inicial.

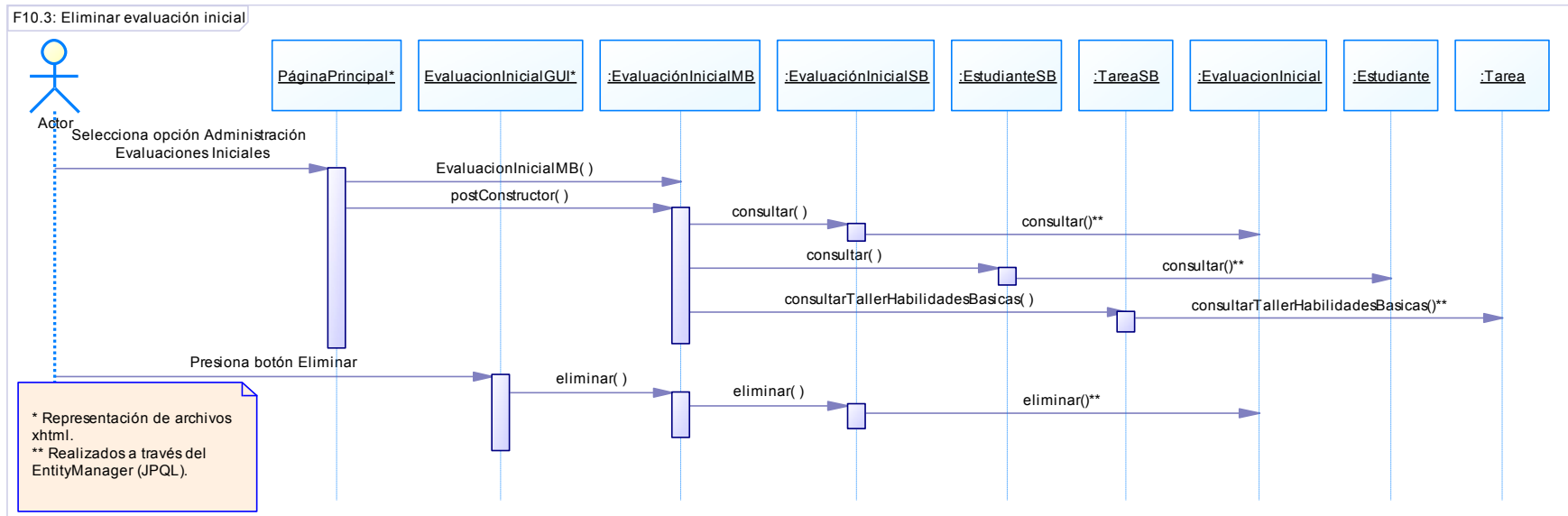


Ilustración 77: Diagrama de Secuencia F10.3

### 3.4. F10.4: Consultar evaluaciones iniciales.

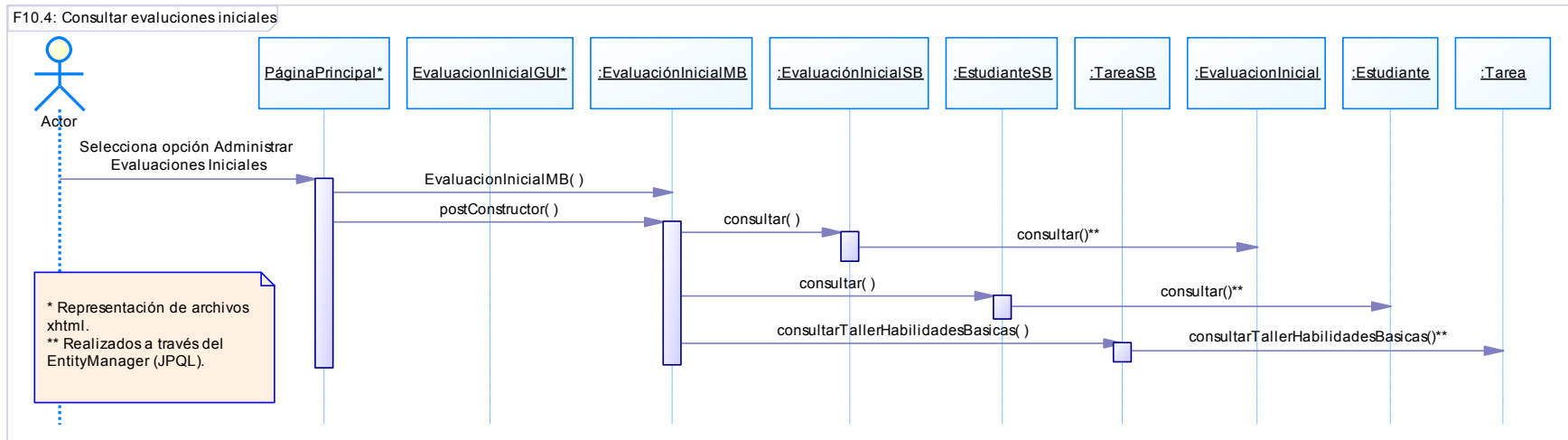


Ilustración 78: Diagrama de Secuencia F10.4

### 3.5. F11.1: Ingresar evaluación mensual.

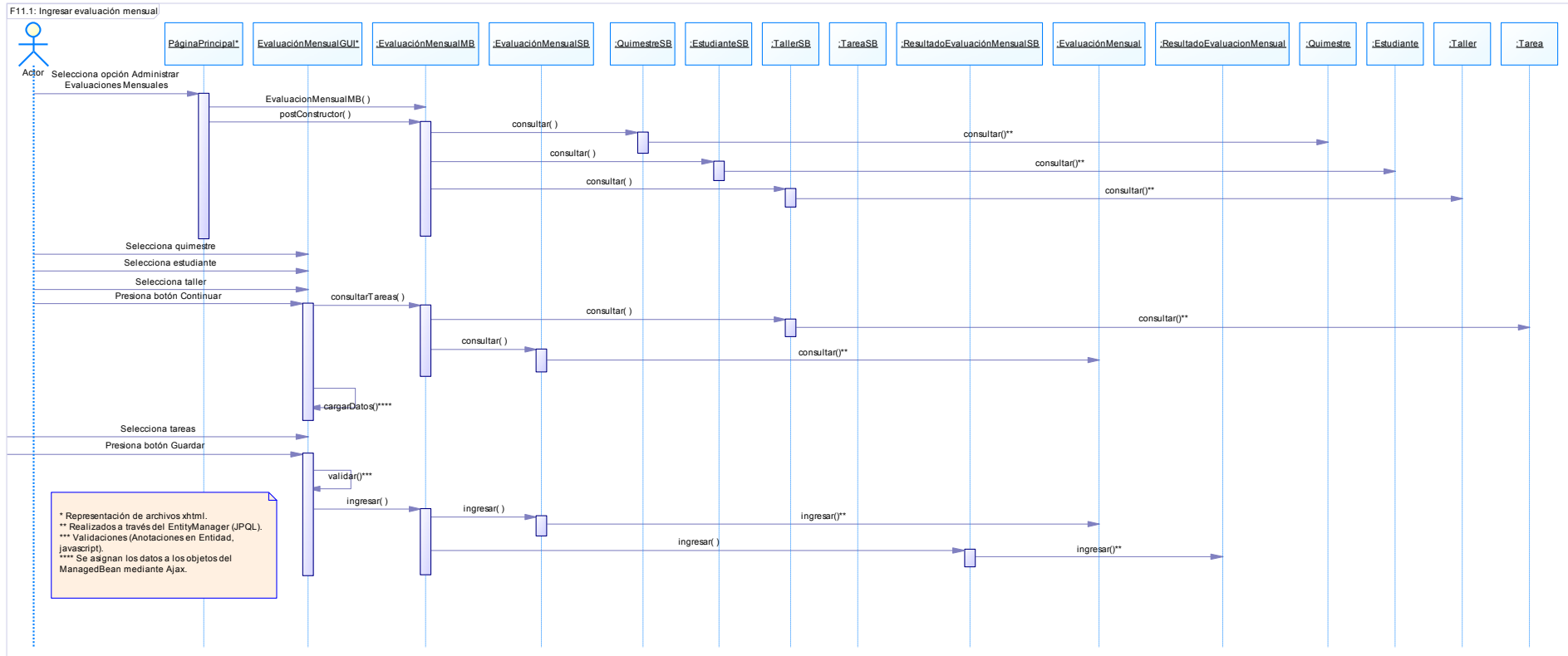


Ilustración 79: Diagrama de Secuencia F11.1

### 3.6. F11.2: Modificar evaluación mensual.

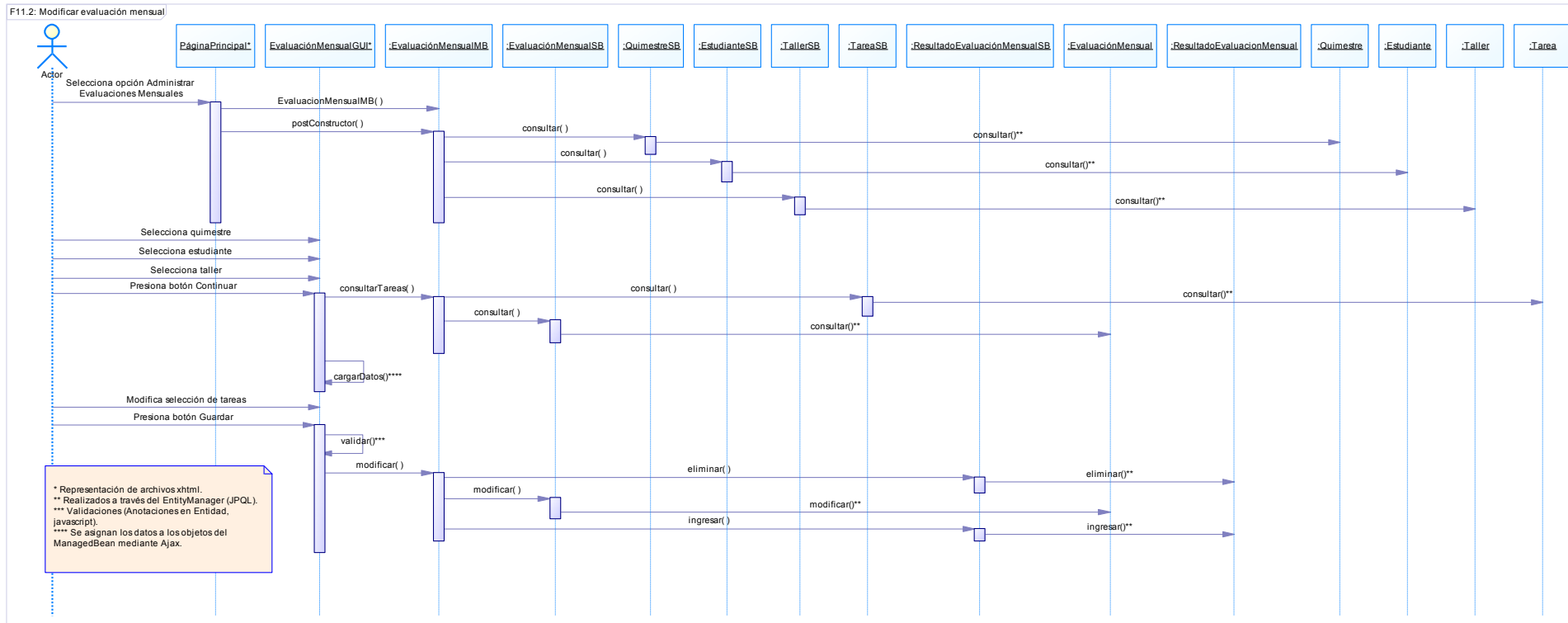


Ilustración 80: Diagrama de Secuencia F11.2

### 3.7. F11.3: Eliminar evaluación mensual.

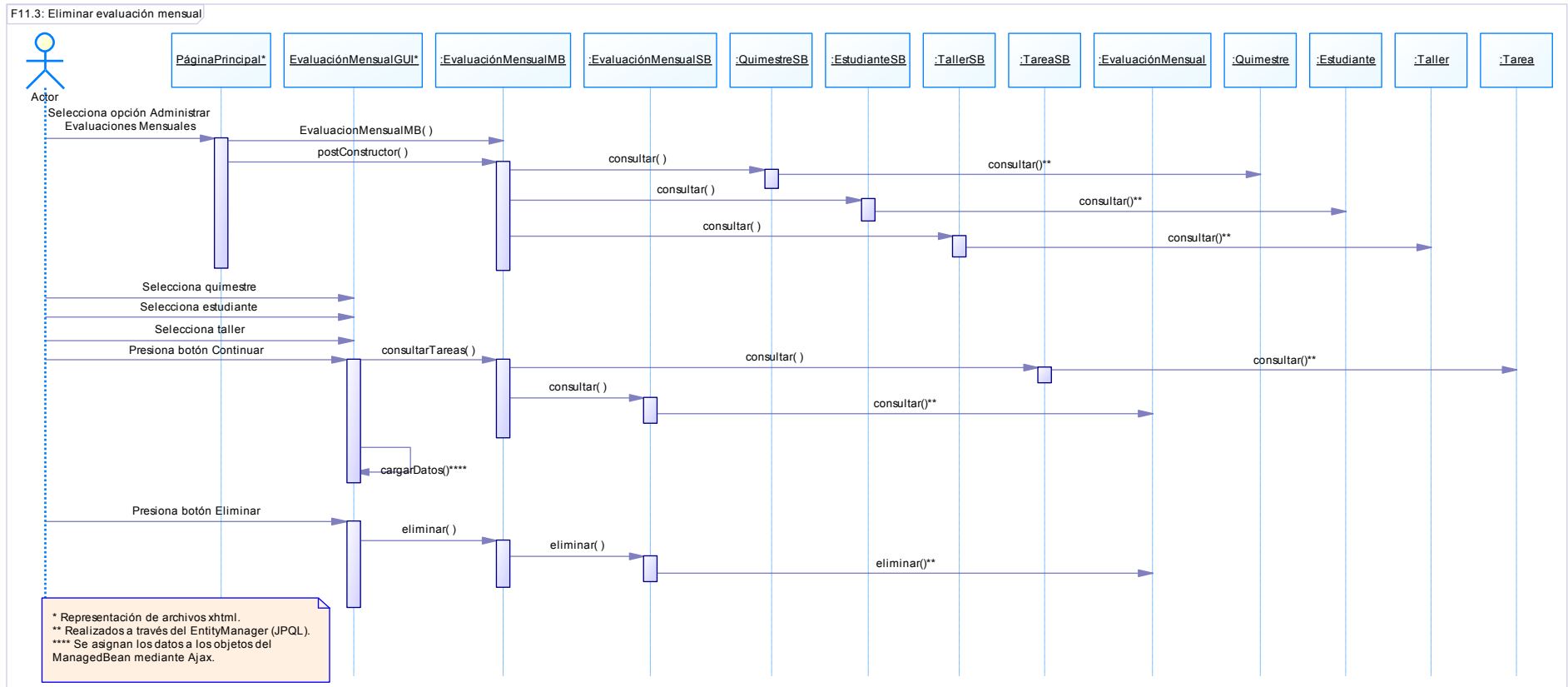


Ilustración 81: Diagrama de Secuencia F11.3



### 3.9. F11.5: Consultar evaluaciones mensuales.

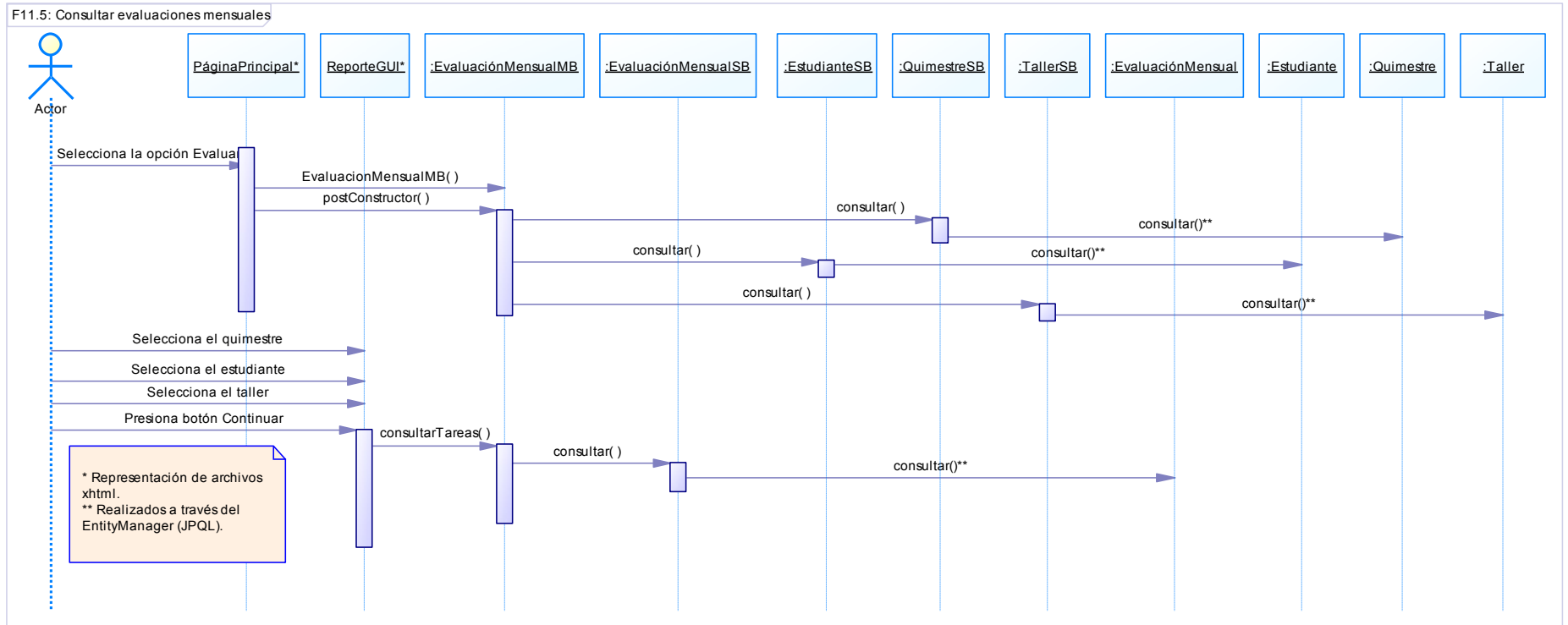


Ilustración 83: Diagrama de Secuencia F11.5



## 5. Diagramas de Actividades: Proceso de Evaluación

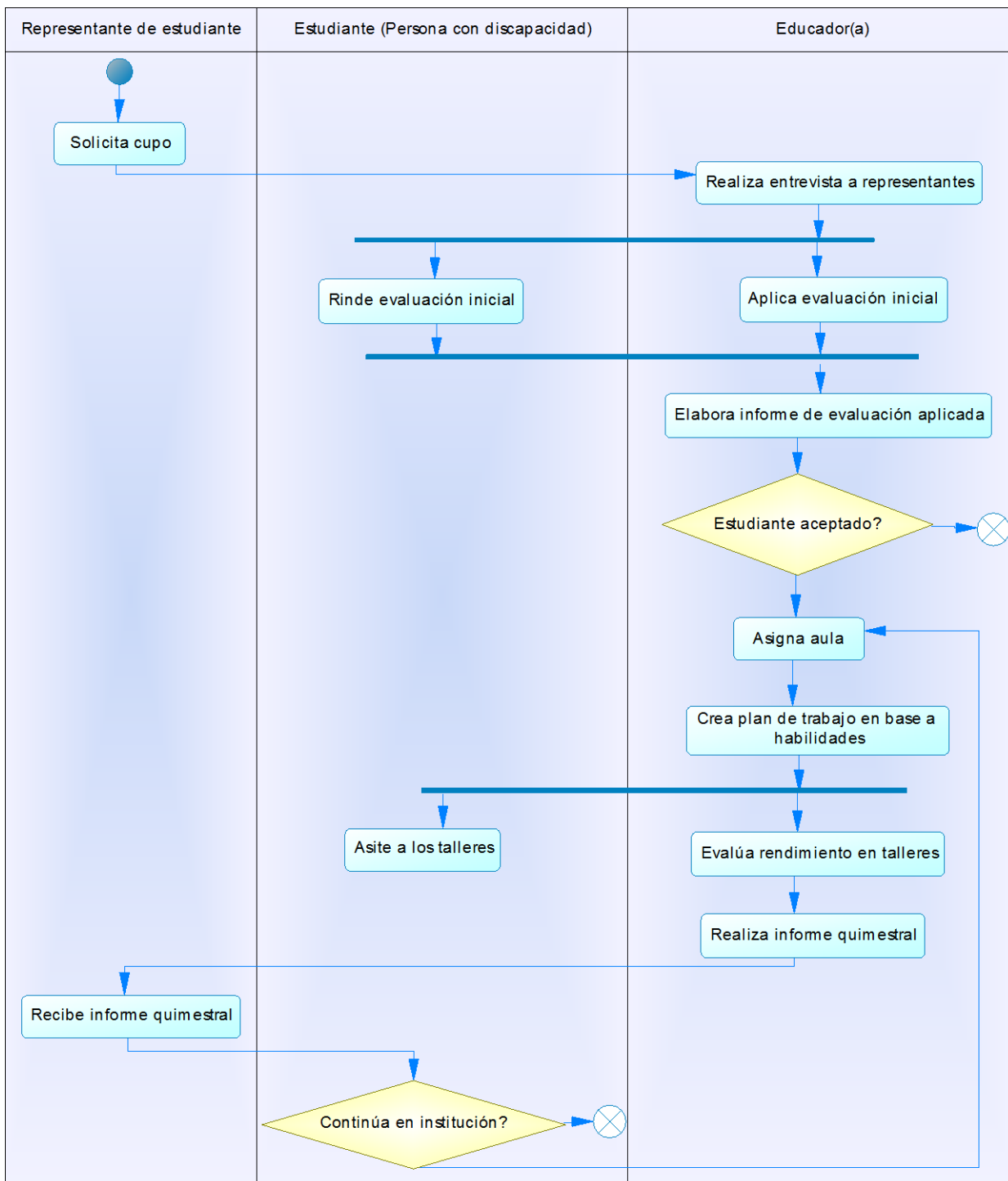


Ilustración 85: Diagrama de Actividades

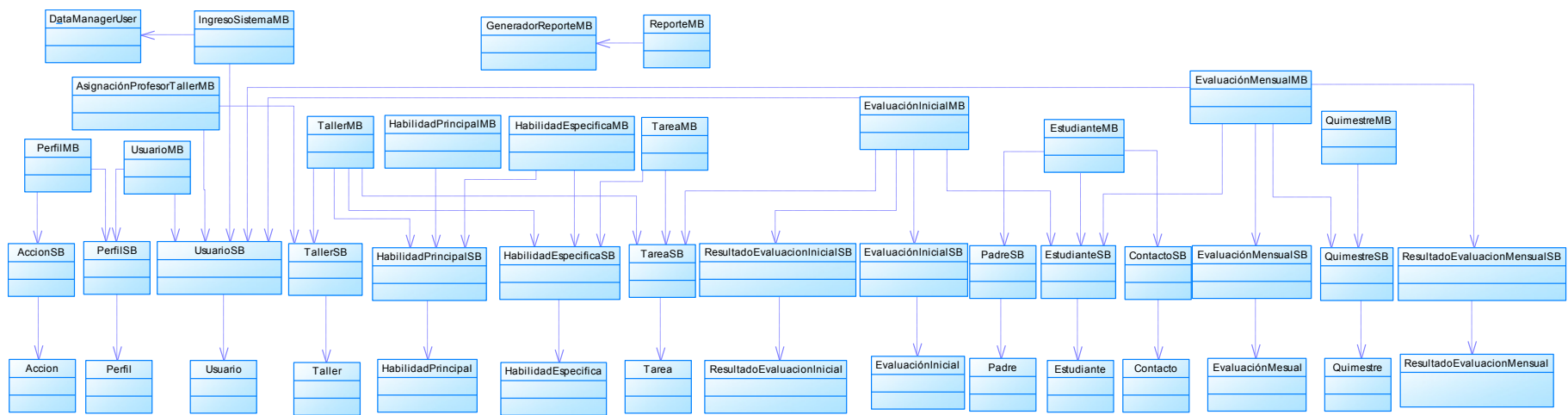
## **6. Plan de Pruebas de Integración (PPI)**

Las pruebas de integración, tienen como objetivo fundamental, comprobar el correcto funcionamiento de los componentes cuando estos trabajan de forma conjunta con otros, verificando que son llamados de forma apropiada y que transfieren los datos adecuados en el tiempo preciso.

La integración del sistema implica identificar grupos de componentes, basándose en las dependencias existentes entre ellos, e integrarlos añadiendo código que haga que funcionen conjuntamente.

La principal dificultad que surge durante las pruebas de integración, es la localización de defectos, por lo que es recomendable utilizar una aproximación incremental para la integración y pruebas, es decir, iniciar con los componentes básicos y el menor número posible e ir añadiendo componentes poco a poco, siguiendo un orden jerárquico.

A continuación se presenta el diagrama de las dependencias existentes entre las clases:



**Ilustración 86: Diagrama de dependencia de clases**

## Grupo 1:

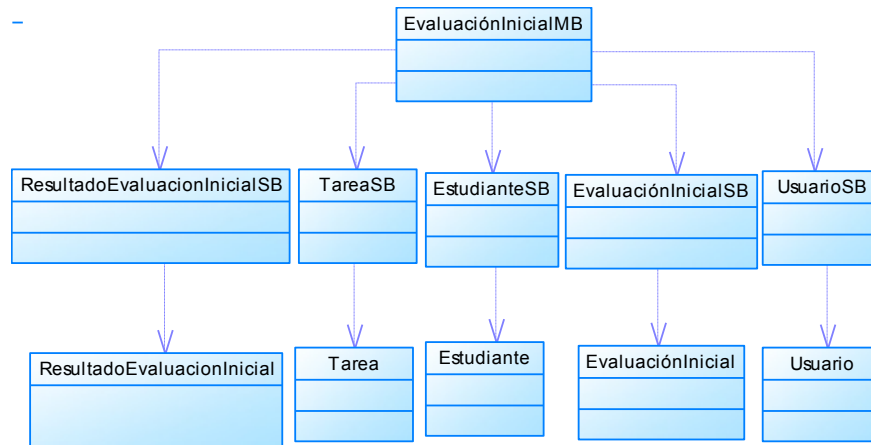


Ilustración 87: Diagrama de dependencia de clases Grupo 1

**Orden:** 1

**Evaluador:** Juan Carlos Fernández Ortiz

**Diseño:**

1. Cargar la página de administración de evaluaciones iniciales.

**Resultados Esperados:**

1. El sistema debe presentar la página de administración de evaluaciones iniciales con los datos de estudiantes, talleres, quimestres.

## Grupo 2:

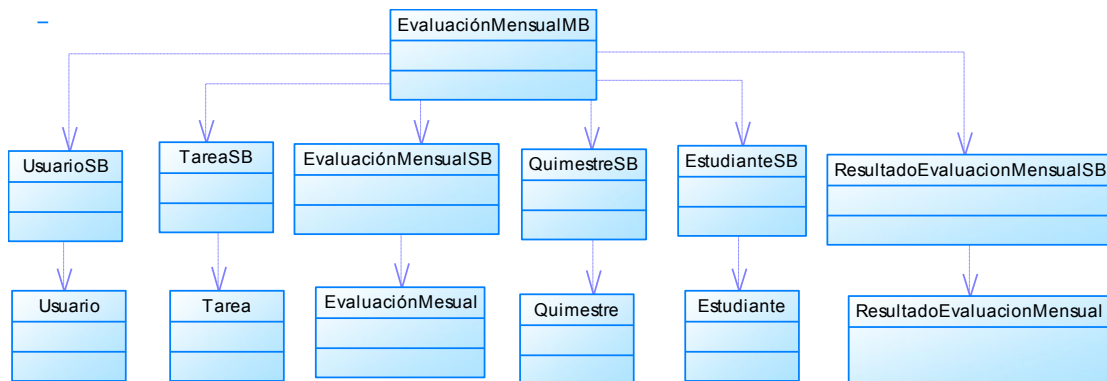


Ilustración 88: Diagrama de dependencia de clases Grupo 2

**Orden:** 1

**Evaluador:** Estefanía Narváez

**Diseño:**

1. Cargar la página de administración de evaluaciones mensuales.

**Resultados Esperados:**

1. El sistema debe presentar la página de administración de evaluaciones mensuales con los datos de estudiantes, talleres, quimestres. Después de seleccionar los tres campos anteriores debe cargar la información de tareas.

### 6.2.6. Implementación

En esta fase se han desarrollado todas las funcionalidades planificadas. A continuación se presenta el código fuente de la funcionalidad que permite la administración de evaluaciones iniciales. El código fuente correspondiente a las funcionalidades restantes se encuentra en el anexo “Código Fuente” del segundo ciclo.

El código fuente presentado forma parte del archivo EvaluacionInicial.xhtml que representa la interfaz gráfica de evaluación inicial.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

<ui:composition template="/templates/TemplateFine.xhtml">
  <ui:define name="formTitle">
    <h:outputText value="Administraci&oacute;n de Evaluaciones
      Iniciales" />
  </ui:define>

  <ui:define name="content">
    <h:form id="frmEvaluacionInicial">
      <p:messages autoUpdate="true"></p:messages>

      <h:panelGrid columns="2">
        <h:outputText value="Estudiante:" />
      </h:panelGrid>
    </h:form>
  </ui:define>
</ui:composition>
<p:selectOneMenu disabled="#{evaluacionInicialMB.btnInsertar}"
  value="#{evaluacionInicialMB.estudianteSeleccionado}" filter="true"
```

```

        filterMatchMode="contains"
        panelStyle="width:350px"
        style="width:350px">
        <f:selectItem itemValue=""
        itemLabel="Seleccione" />
        <f:selectItems
        value="#{evaluacionInicialMB.estudiantes}"
        var="estudiante"
        itemValue="#{estudiante.cedula}"
        itemLabel="#{estudiante.cedula} -
        #{estudiante.nombres} #{estudiante.apellidos}"
        />
    </p:selectOneMenu>
</h:panelGrid>
<br />

<p:fieldset legend="Aciertos" toggleable="true"
toggleSpeed="500"
collapsed="#{evaluacionInicialMB.contraerPaneles}"
widgetVar="resultadosFieldSet">
    <h:panelGrid columns="1" style="text-align:left!important">
        <h:outputText
            value="Seleccione las tareas que cumple
            el estudiante: " />
    </h:panelGrid>
    <p:dataTable value="#{evaluacionInicialMB.tareas}"
    var="tarea"
    paginator="true" paginatorPosition="bottom"
    rows="10"
    emptyMessage="No existen coincidencias. "
    selection="#{evaluacionInicialMB.tareasSeleccionadas}"
    rowKey="#{tarea.id}">
        <p:column>
            <f:facet name="header">
                <h:outputText value="Tarea" />
            </f:facet>
            <h:outputText value="#{tarea.nombre}"
            title="#{tarea.habilidadesEspecificas.nombre}
            (#{tarea.habilidadesEspecificas.habilidadesPrincipales.nombre})"
            />
        </p:column>
        <p:column selectionMode="multiple"
        style="width:2%;" />
    </p:dataTable>
</p:fieldset>
<br />

<p:fieldset legend="Observaciones" toggleable="true"
toggleSpeed="500"
collapsed="#{evaluacionInicialMB.contraerPaneles}"
widgetVar="observacionesFieldSet">
    <h:panelGrid columns="1">
        <p:inputTextarea
value="#{evaluacionInicialMB.evaluacionInicial.observacion}"

```

```

        onkeypress="return
        validarNombres(event)" maxlength="254"
        rows="5"
        cols="50" counter="counter"
        counterTemplate="{0} caracteres
        restantes. " />
    <h:outputText id="counter" />
</h:panelGrid>
</p:fieldset>

<br />
<br />

<h:panelGrid columns="3">
    <p:commandButton value="Guardar" update="@form"
    id="btnGuardar"
    action="#{evaluacionInicialMB.ingresar()}"
    disabled="#{evaluacionInicialMB.btnInsertar}"
    oncomplete="resultadosFieldSet.toggle(true),
    observacionesFieldSet.toggle(true)" />
    <p:commandButton value="Modificar" update="@form"
    id="btnModificar"
    action="#{evaluacionInicialMB.modificar()}"
    disabled="#{!evaluacionInicialMB.btnInsertar}"
    oncomplete="resultadosFieldSet.toggle(true),
    observacionesFieldSet.toggle(true)" />
    <p:commandButton value="Cancelar" process="@this"
    update="@form"
    id="btnCancelar"
    action="#{evaluacionInicialMB.limpiar()}" />
</h:panelGrid>

<br />
<br />

<p:dataTable id="tblEvaluacionInicial"
    value="#{evaluacionInicialMB.evaluacionesIniciales}"
    var="evaluacionInicial" rows="10"
    emptyMessage="No existen coincidencias"
    paginator="true"
    paginatorPosition="bottom">
    <p:column>
        <f:facet name="header">
            <h:outputText value="Estudiante" />
        </f:facet>
        <h:outputText
            value="#{evaluacionInicial.estudiante.nombres}
            #{evaluacionInicial.estudiante.apellidos}" />
    </p:column>
    <p:column style="width:10%">
        <f:facet name="header">
            <h:outputText value="Acciones" />
        </f:facet>
        <p:commandLink title="Seleccionar"
            process="@this" update="@form"
            action="#{evaluacionInicialMB.seleccionar()}">

```

```

                <h:graphicImage height="23px"
                    value="/images/eye.png" />
                <f:setPropertyActionListener
                    target="#{evaluacionInicialMB.evaluacionInicial}"
                    value="#{evaluacionInicial}" />
                <f:setPropertyActionListener
                    target="#{evaluacionInicialMB.estudianteSeleccionado}"
                    value="#{evaluacionInicial.estudiante.cedula}" />
            </p:commandLink>
        </p:column>
    </p:dataTable>
    <br />
    <br />

    <p:blockUI block="frmEvaluacionInicial"
        trigger="btnGuardar, btnModificar, btnCancelar">
        <p:graphicImage value="/images/ajax-loader.gif" />
    </p:blockUI>
</h:form>
</ui:define>
</ui:composition>
</html>

```

El código fuente que implementa a la interface entre la interfaz gráfica de evaluación inicial y el session bean es el siguiente:

```

/**
 * Nombre de la Clase: EvaluacionInicialMB
 * Author: ENarvaez
 * Fecha: 2014/07/17
 */
package ec.uio.fine.managedbeans;

import java.util.Calendar;
import java.util.List;

import javax.annotation.PostConstruct;
import javax.ejb.EJB;
import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import javax.faces.context.FacesContext;

import ec.uio.fine.entities.Estudiante;
import ec.uio.fine.entities.EvaluacionInicial;
import ec.uio.fine.entities.ResultadoEvaluacionInicial;
import ec.uio.fine.entities.Taller;
import ec.uio.fine.entities.Tarea;
import ec.uio.fine.entities.Usuario;
import ec.uio.fine.sessionbeans.EstudianteSB;
import ec.uio.fine.sessionbeans.EvaluacionInicialSB;
import ec.uio.fine.sessionbeans.ResultadoEvaluacionInicialSB;
import ec.uio.fine.sessionbeans.TallerSB;
import ec.uio.fine.sessionbeans.TareaSB;
import ec.uio.fine.sessionbeans.UsuarioSB;

```

```

@ManagedBean
@ViewScoped
public class EvaluacionInicialMB {
    @EJB
    private EstudianteSB estudianteSB;
    @EJB
    private TareaSB tareaSB;
    @EJB
    private UsuarioSB usuarioSB;
    @EJB
    private EvaluacionInicialSB evaluacionInicialSB;
    @EJB
    private ResultadoEvaluacionInicialSB resultadoEvaluacionInicialSB;
    @EJB
    private TallerSB tallerSB;

    private Estudiante estudiante;
    private String estudianteSeleccionado;
    private List<Estudiante> estudiantes;
    private List<Tarea> tareas;
    private Usuario usuario;
    private EvaluacionInicial evaluacionInicial;
    private List<EvaluacionInicial> evaluacionesIniciales;
    private ResultadoEvaluacionInicial resultadoEvaluacionInicial;
    private Tarea[] tareasSeleccionadas;
    private boolean btnInsertar; // Indicará que botón está activo.
    private boolean contraerPaneles; // Indicará el estado de los FieldSet.
    private Taller tallerHabilidadesBasicas;

    /**
     * Constructor
     */

    public EvaluacionInicialMB() {
        evaluacionInicial = new EvaluacionInicial();
    }

    /**
     * Método que se ejecutará inmediatamente después del constructor.
     */
    @PostConstruct
    public void postConstructor() {
        this.btnInsertar = false;
        this.contraerPaneles = true;

        this.estudianteSeleccionado = "";
        this.tareasSeleccionadas = null;

        this.evaluacionesIniciales = evaluacionInicialSB.consultar();
        this.estudiantes = estudianteSB.consultar();
        this.tareas = tareaSB.consultarTallerHabilidadesBasicas();
    }

    /**
     * Getters y Setters
     */
    public EstudianteSB getEstudianteSB() {
        return estudianteSB;
    }

    public void setEstudianteSB(EstudianteSB newEstudianteSB) {

```

```

        estudianteSB = newEstudianteSB;
    }

    public Estudiante getEstudiante() {
        return estudiante;
    }

    public void setEstudiante(Estudiante newEstudiante) {
        estudiante = newEstudiante;
    }

    public String getEstudianteSeleccionado() {
        return estudianteSeleccionado;
    }

    public void setEstudianteSeleccionado(String newEstudianteSeleccionado) {
        estudianteSeleccionado = newEstudianteSeleccionado;
    }

    public List<Estudiante> getEstudiantes() {
        return estudiantes;
    }

    public void setEstudiantes(List<Estudiante> newEstudiantes) {
        estudiantes = newEstudiantes;
    }

    public TareaSB getTareaSB() {
        return tareaSB;
    }

    public void setTareaSB(TareaSB newTareaSB) {
        tareaSB = newTareaSB;
    }

    public List<Tarea> getTareas() {
        return tareas;
    }

    public void setTareas(List<Tarea> newTareas) {
        tareas = newTareas;
    }

    public UsuarioSB getUsuarioSB() {
        return usuarioSB;
    }

    public void setUsuarioSB(UsuarioSB newUsuarioSB) {
        usuarioSB = newUsuarioSB;
    }

    public Usuario getUsuario() {
        return usuario;
    }

    public void setUsuario(Usuario newUsuario) {
        usuario = newUsuario;
    }

    public EvaluacionInicialSB getEvaluacionInicialSB() {
        return evaluacionInicialSB;
    }
}

```

```

public void setEvaluacionInicialSB(
    EvaluacionInicialSB newEvaluacionInicialSB) {
    evaluacionInicialSB = newEvaluacionInicialSB;
}

public EvaluacionInicial getEvaluacionInicial() {
    return evaluacionInicial;
}

public void setEvaluacionInicial(EvaluacionInicial newEvaluacionInicial) {
    evaluacionInicial = newEvaluacionInicial;
}

public List<EvaluacionInicial> getEvaluacionesIniciales() {
    return evaluacionesIniciales;
}

public void setEvaluacionesIniciales(
    List<EvaluacionInicial> newEvaluacionesIniciales) {
    evaluacionesIniciales = newEvaluacionesIniciales;
}

public ResultadoEvaluacionInicial getResultadoEvaluacionInicial() {
    return resultadoEvaluacionInicial;
}

public void setResultadoEvaluacionInicial(
    ResultadoEvaluacionInicial newResultadoEvaluacionInicial) {
    resultadoEvaluacionInicial = newResultadoEvaluacionInicial;
}

public ResultadoEvaluacionInicialSB getResultadoEvaluacionInicialSB() {
    return resultadoEvaluacionInicialSB;
}

public void setResultadoEvaluacionInicialSB(
    ResultadoEvaluacionInicialSB newResultadoEvaluacionInicialSB) {
    resultadoEvaluacionInicialSB = newResultadoEvaluacionInicialSB;
}

public Tarea[] getTareasSeleccionadas() {
    return tareasSeleccionadas;
}

public void setTareasSeleccionadas(Tarea[] tareasSeleccionadas) {
    this.tareasSeleccionadas = tareasSeleccionadas;
}

public boolean isBtnInsertar() {
    return btnInsertar;
}

public void setBtnInsertar(boolean btnInsertar) {
    this.btnInsertar = btnInsertar;
}

public boolean isContraerPaneles() {
    return contraerPaneles;
}

public void setContraerPaneles(boolean contraerPaneles) {

```

```

        this.contraerPaneles = contraerPaneles;
    }

    public TallerSB getTallerSB() {
        return tallerSB;
    }

    public void setTallerSB(TallerSB tallerSB) {
        this.tallerSB = tallerSB;
    }

    public Taller getTallerHabilidadesBasicas() {
        return tallerHabilidadesBasicas;
    }

    public void setTallerHabilidadesBasicas(Taller tallerHabilidadesBasicas) {
        this.tallerHabilidadesBasicas = tallerHabilidadesBasicas;
    }

    /**
     * Método que permite registrar una evaluación en la base de datos,
     * realizando las validaciones necesarias.
     */

    public void ingresar() {

        try {

            byte auxNota = (byte) 0;
            // Se obtiene una instancia de la fecha
            Calendar cal = Calendar.getInstance();

            /**
             * Se verifica si se ha seleccionado un estudiante.
             */

            if (estudianteSeleccionado != "") {
                estudiante =
                    estudianteSB.consultar(estudianteSeleccionado);
                List<EvaluacionInicial> estudianteEvaluacionInicial =
                    evaluacionInicialSB
                        .consultarPorEstudiante(estudiante.getCedula());

                usuario = usuarioSB
                    .consultarPorCedula(IngresoSistemaMB.dataManagerUser
                        .getUsuario().getCedula());

                /**
                 * Verifica si se ha realizado una evaluación inicial del
                 * estudiante
                 */
                if (estudianteEvaluacionInicial.size() > 0) {
                    FacesMessage mensaje = new FacesMessage(
                        FacesMessage.SEVERITY_ERROR,
                        "El estudiante ya tiene asignada una evaluación
                        inicial. Seleccione al estudiante para modificar.",
                        "null");
                    FacesContext.getCurrentInstance().addMessage(null,
                        mensaje);
                } else {
                    this.tallerHabilidadesBasicas = tallerSB
                        .consultarTallerHabilidadesBasicas();
                }
            }
        }
    }

```

```

        this.evaluacionInicial.setEstado((byte) 1);
        this.evaluacionInicial.setEstudiante(estudiante);
        this.evaluacionInicial.setEvaluador(usuario);
        this.evaluacionInicial.setFechaCreacion(
            cal.getTime());

        this.evaluacionInicialSB.ingresar(
            this.evaluacionInicial);
        tallerHabilidadesBasicas.setBloquearEdicion(
            (byte)1);
        tallerSB.modificar(tallerHabilidadesBasicas);

        /*
         * Se almacena los resultados de la evaluación
         */
        for (Tarea tareaAux : tareas) {
            this.resultadoEvaluacionInicial =
                new ResultadoEvaluacionInicial();
            this.resultadoEvaluacionInicial
                .setEvaluacionInicial(evaluacionInicial);

            this.resultadoEvaluacionInicial
                .setTarea(tareaAux);

            auxNota = (byte) 0;
            for (int i = 0;
                i < tareasSeleccionadas.length; i++) {
                if (tareaAux.
                    equals(tareasSeleccionadas[i])) {
                    auxNota = (byte) 1;
                }
            }

            this.resultadoEvaluacionInicial
                .setNota(auxNota);
            this.resultadoEvaluacionInicialSB
                .ingresar(resultadoEvaluacionInicial);
        }

        FacesMessage mensaje = new FacesMessage(
            FacesMessage.SEVERITY_INFO,
            "Evaluación inicial ingresada con éxito. ",
            "null");
        FacesContext.getCurrentInstance()
            .addMessage(null, mensaje);
    }
} else {
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_ERROR,
        "Debe seleccionar un estudiante. ", "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}
} catch (Exception ex) {
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_ERROR,
        "La evaluación inicial no pudo ser ingresada. "
        + ex.getMessage(), ex.getMessage());
    FacesContext.getCurrentInstance().addMessage(null, mensaje);
}
}

```

```

/**
 * Encera variables de la vista
 */

this.contraerPaneles = true;
this.btnInsertar = false;

this.estudianteSeleccionado = "";
this.tareasSeleccionadas = null;
this.evaluacionInicial = new EvaluacionInicial();

this.evaluacionesIniciales = evaluacionInicialSB.consultar();
this.estudiantes = estudianteSB.consultar();
this.tareas = tareaSB.consultarTallerHabilidadesBasicas();
}

/**
 * Método que permite cargar los datos de una evaluación inicial
 * seleccionada para visualizar el detalle de la misma o editarla.
 */
public void seleccionar() {

    List<ResultadoEvaluacionInicial> resultadosEvaluacionesCumplidas =
        resultadoEvaluacionInicialSB
            .consultarPorNota(this.evaluacionInicial.getId(), (byte) 1);

    // Se setean las tareas cumplidas
    if (resultadosEvaluacionesCumplidas.size() > 0) {
        Tarea[] tareasSeleccionadasAux =
            new Tarea[resultadosEvaluacionesCumplidas.size()];

        int i = 0;

        for (ResultadoEvaluacionInicial evaluacionInicialAux
            : resultadosEvaluacionesCumplidas) {
            tareasSeleccionadasAux[i]
                = evaluacionInicialAux.getTarea();
            i++;
        }

        this.tareasSeleccionadas = tareasSeleccionadasAux;
    }

    this.contraerPaneles = false;
    this.btnInsertar = true;
}

/**
 * Método que permite modificar un usuario existente en la base de datos,
 * realizando las validaciones necesarias.
 */
public void modificar() {

    try {
        byte auxNota = (byte) 0;
        Calendar cal = Calendar.getInstance();
        Calendar call = Calendar.getInstance();
        call.setTime(this.evaluacionInicial.getFechaCreacion());
        call.add(Calendar.DAY_OF_YEAR, 3);

        estudiante = estudianteSB.consultar(estudianteSeleccionado);
    }
}

```

```

usuario = usuarios
.consultarPorCedula(IngresoSistemaMB.dataManagerUser.
.getUsuario().getCedula());

/**
 * Se verifica si el usuario tiene permisos para modificar.
 */

if (usuario.getPerfile().getId() != 1
&& (cal.getTime().after(call.getTime()))) {
    FacesMessage mensaje = new FacesMessage(
    FacesMessage.SEVERITY_ERROR,
    "Ha excedido el tiempo permitido para realizar cambios.
    Contacte al administrador del sistema.",
    "null");
    FacesContext.getCurrentInstance()
    .addMessage(null, mensaje);
} else {
    this.evaluacionInicial.setEstudiante(this.estudiante);
    this.evaluacionInicial
    .setUsuarioUltimaModificacion(this.usuario);
    this.evaluacionInicial
    .setFechaUltimaModificacion(cal.getTime());

    this.evaluacionInicialSB
    .modificar(this.evaluacionInicial);

    /*
     * Se modifican los resultados de la evaluación inicial
     */
    this.tareas = tareaSB.consultarTallerHabilidadesBasicas();

    ResultadoEvaluacionInicial resultadoEvaluacionInicialAux;

    for (Tarea tareaAux : tareas) {
        resultadoEvaluacionInicialAux =
        this.resultadoEvaluacionInicialSB
        .consultarPorIdEvaluacionIdTarea(
        this.evaluacionInicial.getId(),
        tareaAux.getId());

        this.resultadoEvaluacionInicial =
        this.resultadoEvaluacionInicialSB
        .consultarPorId(resultadoEvaluacionInicialAux
        .getId());

        auxNota = (byte) 0;

        for (int i = 0; i < tareasSeleccionadas.length;
        i++) {
            if(tareaAux.getId() == tareasSeleccionadas[i]
            .getId()) {
                auxNota = (byte) 1;
            }
        }

        this.resultadoEvaluacionInicial
        .setEvaluacionInicial(this.evaluacionInicial);
        this.resultadoEvaluacionInicial.setTarea(tareaAux);
        this.resultadoEvaluacionInicial.setNota(auxNota);
        this.resultadoEvaluacionInicialSB
        .modificar(this.resultadoEvaluacionInicial);
    }
}

```

```

    }

    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_INFO,
        "Evaluación inicial modificada con éxito.", "null");
    FacesContext.getCurrentInstance()
        .addMessage(null, mensaje);
}

} catch (Exception ex) {
    FacesMessage mensaje = new FacesMessage(
        FacesMessage.SEVERITY_ERROR,
        "La evaluación inicial no pudo ser modificada. "
        + ex.getMessage(), "null");
    FacesContext.getCurrentInstance().addMessage(null, mensaje);
}

/**
 * Encera variables de la vista
 */
this.contraerPaneles = true;
this.btnInsertar = false;

this.estudianteSeleccionado = "";
this.tareasSeleccionadas = null;
this.evaluacionInicial = new EvaluacionInicial();

this.evaluacionesIniciales = evaluacionInicialSB.consultar();
this.estudiantes = estudianteSB.consultar();
this.tareas = tareaSB.consultarTallerHabilidadesBasicas();
}

/**
 * Método que permite desactivar un usuario existente en la base de datos
 * (borrado lógico)
 */
public void eliminar() {
    try {
        estudiante = estudianteSB.consultar(estudianteSeleccionado);
        // Se obtiene una instancia de la fecha
        Calendar cal = Calendar.getInstance();
        usuario = usuarioSB
            .consultarPorCedula(IngresoSistemaMB.dataManagerUser
                .getUsuario().getCedula());

        this.evaluacionInicial.setUsuarioUltimaModificacion(usuario)
            this.evaluacionInicial.setFechaUltimaModificacion(
                cal.getTime());
        this.evaluacionInicial.setEstudiante(estudiante);
        this.evaluacionInicialSB.eliminar(this.evaluacionInicial);

        FacesMessage mensaje =
            new FacesMessage(FacesMessage.SEVERITY_INFO,
                "Evaluación inicial eliminada con éxito. ", "null");
        FacesContext.getCurrentInstance().addMessage(null, mensaje);
    } catch (Exception ex) {
        FacesMessage mensaje = new FacesMessage(
            FacesMessage.SEVERITY_ERROR,
            "La evaluación inicial no pudo ser eliminada. "
            + ex.getMessage(), "null");
    }
}

```

```

        FacesContext.getCurrentInstance().addMessage(null, mensaje);
    }

    /**
     * Encera variables de la vista
     */
    this.contraerPaneles = true;
    this.btnInsertar = false;

    this.estudianteSeleccionado = "";
    this.tareasSeleccionadas = null;
    this.evaluacionInicial = new EvaluacionInicial();

    this.evaluacionesIniciales = evaluacionInicialSB.consultar();
    this.estudiantes = estudianteSB.consultar();
    this.tareas = tareaSB.consultarTallerHabilidadesBasicas();
}

/**
 * Método que permite encerrar las variables de la vista
 */
public void limpiar() {
    this.btnInsertar = false;
    this.contraerPaneles = true;

    this.estudianteSeleccionado = "";
    this.tareasSeleccionadas = null;
    this.evaluacionInicial = new EvaluacionInicial();

    this.evaluacionesIniciales = evaluacionInicialSB.consultar();
    this.estudiantes = this.estudianteSB.consultar();
    this.tareas = tareaSB.consultarTallerHabilidadesBasicas();
}
}

```

A continuación se presenta el código fuente de la lógica del negocio, contenido en el archivo **EvaluacionInicialSB.java**:

```

/**
 * Nombre de la Clase: EvaluacionInicialMB
 * Author: ENarvaez
 * Fecha: 2014/07/17
 */
package ec.uio.fine.sessionbeans;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

import ec.uio.fine.entities.EvaluacionInicial;

@Stateless
public class EvaluacionInicialSB {
    @PersistenceContext

```

```

private EntityManager em;

/**
 * Permite registrar una evaluación inicial en la base de datos.
 * @param evaluacionInicial
 */
public void ingresar(EvaluacionInicial evaluacionInicial) {
    em.persist(evaluacionInicial);
}

/**
 * Permite modificar una evaluación inicial existente en la base de datos.
 * @param evaluacionInicial
 */
public void modificar(EvaluacionInicial evaluacionInicial) {
    em.merge(evaluacionInicial);
}

/**
 * Permite eliminar logicamente una evaluación inicial existente
 * en la base de datos.
 * @param evaluacionInicial
 */
public void eliminar(EvaluacionInicial evaluacionInicial) {
    evaluacionInicial.setEstado((byte) 0);
    em.merge(evaluacionInicial);
}

/*
 * Devuelve todas las evaluaciones iniciales activas
 */
public List<EvaluacionInicial> consultar() {
    Query query = em
        .createQuery("SELECT DISTINCT ei FROM EvaluacionInicial ei JOIN
            ei.resultadosEvaluacionesIniciales rei WHERE ei.estado=:estado");
    query.setParameter("estado", ((byte) 1));
    return query.getResultList();
}

/*
 * Verifica si el estudiante tiene asignada una evaluación inicial
 */
public List<EvaluacionInicial> consultarPorEstudiante(String cedula) {
    Query query = em
        .createQuery("SELECT ei FROM EvaluacionInicial ei JOIN ei.estudiante e
            WHERE ei.estado=:estado AND e.cedula=:cedula");
    query.setParameter("estado", ((byte) 1));
    query.setParameter("cedula", cedula);
    return query.getResultList();
}
}

```

Finalmente se presenta la entidad EvaluacionInicial.java.

```

package ec.uio.fine.entities;

import java.util.Date;
import java.util.List;

import javax.persistence.Column;

```

```

import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * The persistent class for the evaluaciones_iniciales database table.
 */
@Entity
@Table(name="evaluaciones_iniciales")
public class EvaluacionInicial {

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    private String observacion;

    //bi-directional many-to-one association to Estudiante
    @ManyToOne
    @JoinColumn(name="cedula_estudiante")
    private Estudiante estudiante;

    //bi-directional many-to-one association to Usuario
    @ManyToOne (fetch=FetchType.LAZY)
    @JoinColumn(name="cedula_usuario")
    private Usuario evaluador;

    //bi-directional many-to-one association to Usuario
    @ManyToOne (fetch=FetchType.LAZY)
    @JoinColumn(name="cedula_usuario_ultima_modificacion")
    private Usuario usuarioUltimaModificación;

    //bi-directional many-to-one association to ResultadosEvaluacionesIniciales
    @OneToMany(mappedBy="evaluacionInicial")
    private List<ResultadoEvaluacionInicial> resultadosEvaluacionesIniciales;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name="fecha_creacion")
    private Date fechaCreacion;

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name="fecha_ultima_modificacion")
    private Date fechaUltimaModificación;

    private byte estado;

    public EvaluacionInicial() {
    }

    public int getId() {
        return this.id;
    }

```

```

}

public void setId(int id) {
    this.id = id;
}

public String getObservacion() {
    return this.observacion;
}

public void setObservacion(String observacion) {
    this.observacion = observacion;
}

public Estudiante getEstudiante() {
    return this.estudiante;
}

public void setEstudiante(Estudiante estudiante) {
    this.estudiante = estudiante;
}

public Usuario getEvaluador() {
    return this.evaluador;
}

public void setEvaluador(Usuario evaluador) {
    this.evaluador = evaluador;
}

public List<ResultadoEvaluacionInicial> getResultadosEvaluacionesIniciales()
{
    return this.resultadosEvaluacionesIniciales;
}

public void setResultadosEvaluacionesIniciales
(List<ResultadoEvaluacionInicial> resultadosEvaluacionesIniciales) {
    this.resultadosEvaluacionesIniciales = resultadosEvaluacionesIniciales;
}

public ResultadoEvaluacionInicial addResultadoEvaluacionInicial
(ResultadoEvaluacionInicial resultadoEvaluacionInicial) {
    getResultadosEvaluacionesIniciales().add(resultadoEvaluacionInicial);
    resultadoEvaluacionInicial.setEvaluacionInicial(this);

    return resultadoEvaluacionInicial;
}

public ResultadoEvaluacionInicial removeResultadoEvaluacionInicial
(ResultadoEvaluacionInicial resultadoEvaluacionInicial) {
    getResultadosEvaluacionesIniciales()
    .remove(resultadoEvaluacionInicial);
    resultadoEvaluacionInicial.setEvaluacionInicial(null);

    return resultadoEvaluacionInicial;
}

public byte getEstado() {
    return estado;
}

public void setEstado(byte estado) {

```

```

        this.estado = estado;
    }

    public Usuario getUsuarioUltimaModificacion() {
        return usuarioUltimaModificacion;
    }

    public void setUsuarioUltimaModificacion(Usuario usuarioUltimaModificacion) {
        this.usuarioUltimaModificacion = usuarioUltimaModificacion;
    }

    public Date getFechaCreacion() {
        return fechaCreacion;
    }

    public void setFechaCreacion(Date fechaCreacion) {
        this.fechaCreacion = fechaCreacion;
    }

    public Date getFechaUltimaModificacion() {
        return fechaUltimaModificacion;
    }

    public void setFechaUltimaModificacion(Date fechaUltimaModificacion) {
        this.fechaUltimaModificacion = fechaUltimaModificacion;
    }
}

```

### 6.2.7. Integración y Pruebas

En esta fase se presentan las pruebas realizadas en el sistema y los resultados obtenidos en cada una, excluyendo el resumen de los resultados obtenidos al ejecutar las pruebas de unidad.

#### 6.2.7.1. Pruebas de Integración

La ejecución del plan de pruebas de integración produjo resultados satisfactorios que se resumen a continuación:

Grupo	Prueba	Resultados Esperados	Observaciones
1	1	Sí	Resultado satisfactorio.
2	1	Sí	Resultado satisfactorio.

**Tabla 63: Pruebas de Integración. Ciclo II**

### 6.2.7.2. Pruebas del Sistema

Los resultados obtenidos durante la ejecución del plan de pruebas del sistema desarrollado durante la fase de requerimientos del presente ciclo se resumen a continuación:

Prueba	Resultados Esperados	Observaciones
1	Sí	Resultado satisfactorio.
2	Sí	Resultado satisfactorio.
3	Sí	Resultado satisfactorio.
4	Sí	Resultado satisfactorio.
5	Sí	Resultado satisfactorio.
6	Sí	Resultado satisfactorio.
7	Sí	Resultado satisfactorio.
8	Sí	Resultado satisfactorio.
9	Sí	Resultado satisfactorio.
10	Sí	Resultado satisfactorio.
11	Sí	Resultado satisfactorio.
12	Sí	Resultado satisfactorio.
13	Sí	Resultado satisfactorio.

Tabla 64: Pruebas de Sistema. Ciclo II

### 6.2.8. Postmortem

La fase de postmortem es esencial en la mejora continua debido a que en esta fase se evalúa el trabajo realizado hasta el momento, los aciertos y fallos del proceso, así como el estado del ciclo. Este análisis permitirá establecer correctivos en ciclos posteriores, e incluso en proyectos futuros.

### 6.2.8.1. Análisis de Aciertos

Los aciertos detectados durante este ciclo son:

- ✓ Buena comunicación entre los miembros del equipo.
- ✓ A pesar de la baja disponibilidad de tiempo, los miembros del equipo siempre estuvieron dispuestos a cumplir con las actividades asignadas.
- ✓ Se lograron muchos productos de alta calidad.

### 6.2.8.2. Análisis de Fallos

Los fallos que se presentaron durante este ciclo se describen a continuación:

- ✓ Los miembros del equipo continuaron teniendo que trabajar en algunas ocasiones por más de ocho horas al día para lograr cumplir con la planificación debido a que contaban con solo dos días a la semana para dedicarse al proyecto actual.

### 6.2.8.3. Formulario PEER Consolidado

A continuación se presenta la forma PEER consolidada de la evaluación de cada miembro:

#### TSPi Team and Peer Evaluation: Form PEER

For each role, evaluate the work required and the relative difficulty in % during this cycle.		
Role	Work Required	Role Difficulty
Team Leader	17.5	20
Development Manager	25	15
Planning Manager	25	27.5
Quality/Process Manager	22.5	30
Support Manager	10	10
Total Contribution (100%)	<b>100</b>	<b>100</b>

Rate the overall team against each criterion. Circle one number from 1 (inadequate) to 5 (superior).					
Team spirit	1	2	3	4	5
Overall effectiveness	1	2	3	4	5
Rewarding experience	1	2	3	4	5
Team productivity	1	2	3	4	5

Process quality	1	2	3	④	5
Product quality	1	2	3	④	5

Rate role for overall contribution. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	⑤
Development Manager	1	2	3	4	⑤
Planning Manager	1	2	3	4	⑤
Quality/Process Manager	1	2	3	4	⑤
Support Manager	1	2	3	4	⑤

Rate each role for helpfulness and support. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	⑤
Development Manager	1	2	3	4	⑤
Planning Manager	1	2	3	4	⑤
Quality/Process Manager	1	2	3	4	⑤
Support Manager	1	2	3	4	⑤

Rate each role for how well it was performed. Circle one number from 1 (inadequate) to 5 (superior).					
Team Leader	1	2	3	4	⑤
Development Manager	1	2	3	4	⑤
Planning Manager	1	2	3	4	⑤
Quality/Process Manager	1	2	3	4	⑤
Support Manager	1	2	3	4	⑤

### Análisis

El desempeño del grupo ha sido bueno, se han logrado desarrollar todos los productos planificados, aunque no se lograron al 100% los objetivos propuestos. A pesar de pequeños inconvenientes que se pueden haber presentado por la disponibilidad de tiempo, en general el grupo se acopla muy bien.

#### 6.2.8.4. Análisis del ciclo

A continuación se presenta un análisis detallado del ciclo:

	Tiempo	Tamaño			Defectos
		LOC	Páginas	Diagramas	
<b>Actual</b>	158.8	2296	26	24	7
<b>Planificado</b>	159.1	3896	25	23	20

Tabla 65: Análisis del Ciclo. Ciclo II

**% de Avance:** 100

**Desviación del porcentaje:**

$$\text{Desviación del porcentaje} = \frac{\text{Real}}{\text{Planificado}} (100)$$

**% Tiempo:** 99.8%

**% Tamaño:**

- **LOC:** 58.9%
- **Páginas:** 104%
- **Diagramas:** 104.3%

**% Defectos:** 35%

**Error Porcentual**

$$E_{\%} = \frac{\text{Planificado} - \text{Real}}{\text{Planificado}} (100)$$

**% Tiempo:** 0.2%

**% Tamaño:**

- **LOC:** 44.1%
- **Páginas:** -4%
- **Diagramas:** -4.3%

**% Defectos: 65%**

- ✓ El tiempo planificado para el desarrollo del ciclo excedió en un 0.2% al tiempo utilizado, por lo que se logró cumplir con el margen de error deseado de  $\pm 6\%$ .
- ✓ En cuanto al tamaño: El error en el número de LOCs planificadas es de 44.1% más que lo real. El número de páginas planificado es 4% menor al real, mientras que el número de diagramas planificados es 4.3% menor al real. En los dos últimos casos se cumple con el margen de error deseado de  $\pm 30\%$ .
- ✓ En cuanto a defectos, se cometió un error porcentual de 65%.

## 7. Conclusiones

- La Fundación para la Integración del Niño Especial es una organización que se especializa en la capacitación de personas con múltiples discapacidades, para lo cual ha desarrollado una matriz de habilidades que se utiliza como base para una evaluación inicial de las destrezas de la persona, para posteriormente, diseñar un programa de trabajo individual.
- La discapacidad mental implica un deterioro general del funcionamiento intelectual y del comportamiento adaptativo de una persona, que genera necesidades educativas especiales para cada individuo, e incluso para cada etapa de su vida.
- Team Software Process es un marco de trabajo que aplica los conceptos de Ingeniería de Software y procesos, enfocándose en el trabajo en equipo de manera eficiente, con el fin de obtener productos de alta calidad.
- Se puede definir a TSPi como un marco de trabajo que cuenta con un modelo espiral de ciclo de vida de software; ya que sus fases, cubren las etapas de dicho modelo. Dentro de la etapa del qué, se encuentran las fases lanzamiento y definición de requerimientos; donde se definen los objetivos y alcances del producto final. La etapa del cómo está compuesta por las fases de estrategia, planificación y diseño; donde se determina la mejor manera para cumplir con los objetivos y requerimientos. En la etapa de acción, se encuentra la fase de implementación; donde se desarrolla el software en base a las especificaciones de diseño. Por último, la etapa de mejoramiento, que incluye a las fases de pruebas y postmortem; donde se evalúan los resultados obtenidos durante el ciclo y se plantean acciones que permitan una mejora continua.
- Java es un lenguaje de programación de alto nivel caracterizado por ser seguro, robusto, multihilos, portable, de arquitectura neutral, interpretado, dinámico, distribuido y orientado a objetos.
- Enterprise Java Beans es una arquitectura que separa tres conceptos: la lógica del negocio, la capa de presentación y la capa de acceso a los datos. En la capa de presentación tenemos las páginas web y los Managed Beans. En la segunda capa, conocida como la capa de la lógica del negocio, se encuentran los Session Beans y los

Message Driven Beans. Por último, la capa de acceso a los datos está conformada por los Entity Beans.

- El producto desarrollado en el presente trabajo pretende convertirse en una herramienta de apoyo para la sistematización del proceso de capacitación de personas con discapacidad, que constituye una parte fundamental de la cadena de valor de la Fundación para la Integración del Niño Especial, contribuyendo en la mejora de su planificación y evaluación.
- Dadas las características del producto, así como el proceso seguido para su desarrollo, el sistema podría ser implementado en varias organizaciones que persigan objetivos similares a la Fundación para la Integración del Niño Especial, realizando modificaciones mínimas.

## 8. Recomendaciones

- Durante la definición de requerimientos para el producto desarrollado en el presente trabajo, se identificaron necesidades adicionales de la organización, que sería importante tomar en cuenta para trabajos posteriores, como la implementación de un módulo que permita el seguimiento de la inserción laboral de las personas con discapacidad, basado en evaluaciones realizadas por el empleador o supervisor directo sobre el desempeño del trabajador.
- Evaluar la posibilidad de adicionar al reporte implementado una sección que permita conocer de manera global el estado de una persona con discapacidad en relación al desarrollo de las habilidades principales o específicas establecidas por la organización, así como su evolución en el tiempo.
- Implementar un módulo que permita el registro de fichas médicas y biopsicosociales de las personas con discapacidad que son parte de la organización, así como la evolución de indicadores en el tiempo, que permitan establecer mejoras en la calidad de vida de las personas con discapacidad.
- Durante el desarrollo de un software, es indispensable utilizar estándares de codificación, diseño y documentación con el fin de obtener un producto mantenible y de alta calidad.
- Es importante definir las limitaciones del sistema, así como de sus funcionalidades durante la fase de requerimientos de manera clara y precisa, para evitar contratiempos durante las siguientes fases.
- Es esencial detectar los riesgos a los que está expuesto el equipo de trabajo oportunamente, a fin de tomar acciones para evitar o mitigar sus efectos.
- Adaptar e implantar el sistema informático en organizaciones que persigan objetivos similares a la Fundación para la Integración del Niño Especial.

## 9. Bibliografía

- Beirne-Smith, M., Patton, J., & Ittenbach, R. (1994). *Mental Retardation*. Ontario: Maxwell Macmillan.
- Cade, M., & Sheil, H. (2010). *Sun Certified Enterprise Architect for Java EE Study Guide* (2da ed.). Crawfordsville: Pearson Education.
- De la Cruz, F. (2007, Marzo 09). Ingeniería de Software 1. *Modelos de Ciclo de Vida de Software*. Quito, Ecuador.
- DeMarco, T. (1988). Looking for Lost Keys. *Software Magazine*.
- Diagonal Santillana. (1985). *Diccionario Enciclopédico de Educación Especial*. Madrid: Santillana.
- Dirección Nacional de Asesoría Jurídica de la PGE. (2013, mayo). Código del Trabajo. Ecuador.
- El Telégrafo. (2013, julio 30). *El Telégrafo*. Retrieved agosto 27, 2013, from En 3 años se quintuplicó la cifra de personas con discapacidad que trabajan: <http://www.telegrafo.com.ec/noticias/sociedad/item/en-3-anos-se-quintuplico-cifra-de-discapacitados-que-trabajan.html>
- Fernández Ortiz, J. C., & Narváez, E. (2012, Mayo 14). Sistema de Asignación de Horarios y Materias para la Dirección de Pastoral Universitaria de la PUCE. Quito, Ecuador.
- Fundación para la Integración del Niño Especial F.I.N.E. (2013, mayo 25). Planificación Estratégica. Quito, Pichincha, Ecuador.
- González Manjón, D. (2006). *NEE más frecuentes en la Educación Infantil*. Cádiz: Departamento de Psicología - Universidad de Cádiz.
- Gosling, J., & McHilton, H. (1996, mayo). *Oracle*. Retrieved marzo 1, 2014, from The Java Language Environment. A White Paper: <http://www.oracle.com/technetwork/java/intro-141325.html#367>

- Herrera Ramírez, M. d., & Martínez Bravo, D. C. (1999). *Programa de Orientación y Capacitación Ocupacional en Talleres para Adultos con Retraso Mental de la Fundación para la Integración del Niño Especial*. Quito: Pontificia Universidad Católica del Ecuador - Facultad de Psicología.
- Hugh, E. W., & Tahaghoghi, S. (2007). *Learning MySQL*. Sebastopol: O'Reilly Media.
- Humphrey, W. S. (2000). *Introduction to the Team Software Process*. Addison Wesley Longman Inc.
- Iturbide, R., & Serrano, P. (2004). *Manual de Orientación Laboral para Personas con Discapacidad*. Bizkaia: Lantegi Batuak.
- Ministerio de Relaciones Laborales. (2013). *Ministerio de Relaciones Laborales*. Retrieved agosto 25, 2013, from Inserción de personas con discapacidad: <http://www.relacioneslaborales.gob.ec/insercion-de-personas-con-discapacidades/>
- Myers, P. I., & Hammill, D. D. (1990). *Métodos para Educar Niños con Dificultades en el Aprendizaje. Métodos para su Educación*. México: Noriega Limusa.
- Oracle Corporation. (n.d.). *Oracle*. Retrieved Febrero 16, 2014, from Java Language Specification: <http://docs.oracle.com/javase/specs/jls/se7/html/jls-1.html>
- Oracle Company. (2014). *MySQL*. Retrieved Marzo 2, 2014, from MySQL 5.0 Reference Manual: <http://dev.mysql.com/doc/refman/5.0/es/introduction.html>
- Oracle Corporation. (2010). *Java Programming Language, Java SE6 (1.1 ed.)*. Redwood Shores, California: Oracle University.
- Oracle Corporation. (2013, marzo 20). *Oracle*. Retrieved marzo 1, 2014, from Java SE Security: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136007.html>
- Oracle Corporation. (2013, diciembre 20). *Oracle*. Retrieved marzo 1, 2014, from The Java TM Tutorials: <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>

- Oracle Corporation. (n.d.). *Oracle*. Retrieved Febrero 16, 2014, from Java Virtual Machine Specification: <http://docs.oracle.com/javase/specs/jvms/se7/html/jvms-1.html#jvms-1.1>
- Ortiz, Y. (2014, marzo 4). Procesos de FINE. (E. Narváez Bastidas, & J. C. Fernández Ortiz, Interviewers)
- Perera Mezquida, J. (1995). *Síndrome de Down. Programa de Acción Educativa*. Madrid: CEPE S.A.
- Pournaghshbanb, H. (1990, Febrero). Twenty-First SIGCSE Technical Symposium on Computer Science Education. SIGCSE Bulletin 22, No. 1. *The Students' Problems in Courses with Team Projects*.
- Rubinger, A. L., & Burke, B. (2010). *Enterprise Java Beans 3.1* (6ta ed.). Sebastopol, California: O'Reilly Media Inc.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2007). *El Lenguaje Unificado del Modelado. Manual de Referencia* (Segunda ed.). Madrid, España: Pearson Educación S.A.
- Sasha Pachev, A. (2003). *MySQL Enterprise Solutions*. Wiley Publishing Inc.
- Schildt, H. (2009). *Java Manual de Referencia* (7ma ed.). México D.F.: McGraw-Hill Interamericana Editores.
- Sierra, K., & Bates, B. (2003). *Head First EJB*. Sebastopol, California: O'Reilly Media Inc.
- Sommerville, I. (2005). *Ingeniería del Software* (Séptima ed.). Madrid: Pearson Addison Wesley.
- Verdugo, M. A. (2003). *Programa de Habilidades Sociales: Programas Conductuales Alternos*. España: Amarú.
- Wetherbee, J., Rathod, C., Kodali, R., & Zadrozny, P. (2013). *Beginning EJB 3 Java EE 7 Edition*. New York: Apress Media.

## **10. Anexos**

### **10.1. Primer Ciclo**

#### **10.1.1. Formas Fases de Lanzamiento, Estrategia y Planificación**

Las formas de TSPi correspondientes a estas fases, así como los estándares de documentos y comunicación y gestión de configuración se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana1 Lanzamiento, Estrategia y Planificación”, en el documento “Entregable\_LSP.pdf”.

#### **10.1.2. Casos de Uso a Detalle**

Los casos de uso a detalle correspondientes a la fase de Definición de Requerimientos del primer ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana2 Requerimientos”, en el documento “Casos de Uso a Detalle.pdf”.

#### **10.1.3. Plan de Pruebas del Sistema**

El plan de pruebas del sistema correspondientes a la fase de Definición de Requerimientos del primer ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana2 Requerimientos”, en el documento “Plan de Pruebas del Sistema.pdf”.

#### **10.1.4. Formas Fase de Requerimientos**

Las formas de TSPi correspondientes a esta fase, así como los estándares para diseño se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana2 Requerimientos”, en el documento “Entregable\_REQ.pdf”.

### **10.1.5. Diagramas de Secuencia**

Los diagramas de secuencia correspondientes a la fase de Diseño del primer ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana3 Diseño”, en el documento “Diagramas de Secuencia.pdf”.

### **10.1.6. Formas Fase de Diseño**

Las formas de TSPi correspondientes a esta fase se encuentran adjuntas en el directorio “Anexos/Ciclo I/Semana3 Diseño”, en el documento “Entregable\_DIS.pdf”.

### **10.1.7. Código Fuente**

El código fuente de las funcionalidades correspondientes a este ciclo se encuentra adjunto en el directorio “Anexos/Ciclo I/ Semana4 Implementación/SistemaHabilidades”.

### **10.1.8. Formas Fase de Implementación**

Las formas de TSPi correspondientes a esta fase, así como los estándares de desarrollo se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana4 Implementación”, en el documento “Entregable\_IMP.pdf”.

### **10.1.9. Formas Fase de Pruebas y Postmortem**

Las formas de TSPi correspondientes a esta fase se encuentran adjuntas en el directorio “Anexos/Ciclo I/Semana5 Pruebas y Postmortem”, en el documento “Entregable\_PyP.pdf”.

## **10.2. Segundo Ciclo**

### **10.2.1. Formas Fase de Lanzamiento, Estrategia y Planificación**

Las formas de TSPi correspondientes a estas fases se encuentran adjuntas en el directorio “Anexos/Ciclo II/Semana1 Lanzamiento, Estrategia y Planificación”, en el documento “Entregable\_LSP.pdf”.

### **10.2.2. Casos de Uso a Detalle**

Los casos de uso a detalle correspondientes a la fase de Definición de Requerimientos del primer ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo I/Semana2 Requerimientos y Diseño”, en el documento “Casos de Uso a Detalle.pdf”.

### **10.2.3. Plan de Pruebas del Sistema**

El plan de pruebas del sistema correspondientes a la fase de Definición de Requerimientos del segundo ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo II/Semana2 Requerimientos y Diseño”, en el documento “Plan de Pruebas del Sistema.pdf”.

### **10.2.4. Diagramas de Secuencia**

Los diagramas de secuencia correspondientes a la fase de Diseño del segundo ciclo se encuentran adjuntos en el directorio “Anexos/Ciclo II/Semana2 Requerimientos y Diseño”, en el documento “Diagramas de Secuencia.pdf”.

### **10.2.5. Formas Fase de Requerimientos y Diseño**

Las formas de TSPi correspondientes a estas fases se encuentran adjuntas en el directorio “Anexos/Ciclo II/ Semana2 Requerimientos y Diseño”, en el documento “Entregable\_RyD.pdf”.

### **10.2.6. Código Fuente**

El código fuente de las funcionalidades correspondientes a este ciclo se encuentra adjunto en el directorio “Anexos/Ciclo II/ Semana3 Implementación/SistemaHabilidades”.

### **10.2.7. Formas Fase de Implementación**

Las formas de TSPi correspondientes a esta fase se encuentran adjuntas en el directorio “Anexos/Ciclo II/Semana3 Implementación”, en el documento “Entregable\_IMP.pdf”.

### **10.2.8. Fase de Pruebas y Postmortem**

Las formas de TSPi correspondientes a esta fase se encuentran adjuntas en el directorio “Anexos/Ciclo II/Semana4 Pruebas y Postmortem”, en el documento “Entregable\_PyP.pdf”.

### **10.2.9. Manual de Usuario**

El manual de usuario se encuentra adjunto en el directorio “Manuales”, en el documento “Manual de Usuario.pdf”.

### **10.2.10. Manual de Instalación**

El manual de instalación se encuentra adjunto en el directorio “Manuales”, en el documento “Manual de Instalación.pdf”.