



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN SOFTWARE

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN SOFTWARE**

**AUTORES: ACOSTA RODRÍGUEZ YMIR CAMILO
PAREDES ANDINO JUAN FRANCISCO**

**TEMA: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
REGULADOR DE DISTRIBUCIÓN Y CONTROL DE LOS
BIENES SERVIBLES, REPARABLES Y CONDENADOS
ASIGNADOS AL DEPARTAMENTO DE LOGÍSTICA DE LA
ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS
(EPAE), ACANTONADA EN LA CIUDAD DE LATACUNGA,
APLICANDO LA METODOLOGÍA ESTÁNDAR USADA EN LA
FUERZA AÉREA ECUATORIANA”.**

**DIRECTOR: ING. GUERRA LUIS
CODIRECTOR: ING. NAVAS PATRICIO**

LATACUNGA, OCTUBRE 2014

UNIVERSIDAD DE LA FUERZAS ARMADAS - ESPE**CARRERA DE INGENIERÍA EN SOFTWARE****CERTIFICADO**

ING. LUIS GUERRA (DIRECTOR)
ING. PATRICIO NAVAS (CODIRECTOR)

CERTIFICAN:

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA REGULADOR DE DISTRIBUCIÓN Y CONTROL DE LOS BIENES SERVIBLES, REPARABLES Y CONDENADOS ASIGNADOS AL DEPARTAMENTO DE LOGÍSTICA DE LA ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS (EPAE), ACANTONADA EN LA CIUDAD DE LATACUNGA, APLICANDO LA METODOLOGÍA ESTÁNDAR USADA EN LA FUERZA AÉREA ECUATORIANA” realizado por los señores: Ymir Acosta Rodríguez y Juan Paredes Andino, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas-ESPE.

Debido a que constituye un trabajo de excelente contenido científico que coadyuvará a la aplicación de conocimientos y al desarrollo profesional, se recomiendan su publicación.

El mencionado trabajo consta de un empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a los señores: Ymir Acosta Rodríguez y Juan Paredes Andino que lo entregue al Ing. Lucas Garcés, en su calidad de Director de Carrera.

Latacunga, 27 de Octubre 2014.

ING. LUIS GUERRA
DIRECTOR

ING. PATRICIO NAVAS
CODIRECTOR

UNIVERSIDAD DE LA FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA EN SOFTWARE

DECLARACIÓN DE RESPONSABILIDAD

NOSOTROS, ACOSTA RODRÍGUEZ YMIR CAMILO
PAREDES ANDINO JUAN FRANCISCO

DECLARAMOS QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA REGULADOR DE DISTRIBUCIÓN Y CONTROL DE LOS BIENES SERVIBLES, REPARABLES Y CONDENADOS ASIGNADOS AL DEPARTAMENTO DE LOGÍSTICA DE LA ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS (EPAE), ACANTONADA EN LA CIUDAD DE LATACUNGA, APLICANDO LA METODOLOGÍA ESTÁNDAR USADA EN LA FUERZA AÉREA ECUATORIANA” ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, cuyas citas se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, 27 de Octubre del 2014.

Ymir Acosta Rodríguez
CI: 172349775-4

Juan Paredes Andino
CC: 171354330-2

UNIVERSIDAD DE LA FUERZAS ARMADAS - ESPE

CARRERA DE INGENIERÍA EN SOFTWARE

AUTORIZACIÓN

**NOSOTROS, ACOSTA RODRÍGUEZ YMIR CAMILO
PAREDES ANDINO JUAN FRANCISCO**

Autorizamos a la Universidad de las Fuerzas Armadas - ESPE la publicación, en la biblioteca virtual de la Institución del trabajo “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA REGULADOR DE DISTRIBUCIÓN Y CONTROL DE LOS BIENES SERVIBLES, REPARABLES Y CONDENADOS ASIGNADOS AL DEPARTAMENTO DE LOGÍSTICA DE LA ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS (EPAE), ACANTONADA EN LA CIUDAD DE LATACUNGA, APLICANDO LA METODOLOGÍA ESTÁNDAR USADA EN LA FUERZA AÉREA ECUATORIANA”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, 27 de Octubre del 2014.

Ymir Acosta Rodríguez
CI: 172349775-4

Juan Paredes Andino
CC: 171354330-2

DEDICATORIA

A mi papito Dios que siempre es fiel conmigo y con todas las personas que confían en él, a mi querida esposa Jeanneth, a mis preciosas hijas Valentina e Isabella por todo el apoyo que me brindaron en todo momento, a mis queridos padres Hugo y Carmen a quienes les debo todo lo que hasta hoy en día soy; a ellos que me enseñaron a ser un buen hombre, hijo, esposo y un padre para mis hijas, con los principios y valores que me inculcaron desde que nací. A todos ellos les dedico mis logros alcanzados tanto en mi vida personal como profesional, gracias por ser el pilar fundamental en mi vida.

Francisco Paredes A.

A mi Madre, que con su comprensión, confianza y que en el trayecto de mi vida me ha demostrado su amor corrigiendo mis faltas y celebrando mis triunfos, ella ha sido una pieza fundamental para alcanzar este logro en mi vida, a Víctor que desde el inicio creyó en mí, brindándome todo su apoyo, siendo un consejero cuando más lo necesitaba, dándome animo, su cariño incondicional, a ellos que me han dado todo lo que soy como persona, principios, valores, carácter y coraje para conseguir mis objetivos, a mi primo Henry por ser un ejemplo de constancia y entereza en la vida.

Ymir Acosta

AGRADECIMIENTO

A mi Dios que siempre me acompaña, a toda mi querida y amada familia por ser el soporte en mi vida, a la Escuela Superior Politécnica del Ejército (ESPE) en la cual me inicié, actualmente llamada Universidad de las Fuerzas Armadas – ESPE extensión Latacunga, a cada uno de los docentes de esta prestigiosa institución que impartieron sus conocimientos, a nuestra querida Ingeniera Ximena López quien brindo su apoyo incondicional y abnegado en todo momento, a mi compañero y amigo Ymir Camilo por su responsabilidad y dedicación en el desarrollo de la Tesis.

Francisco Paredes A.

Al terminar un trabajo laborioso como lo es una tesis es inevitable recordar los obstáculos que se presentaron a lo largo del proceso y con ellos a las personas que me ayudaron y apoyaron para poder sustentarlos satisfactoriamente, por esta razón agradezco primero a Dios por haberme permitido llegar a este momento de mi vida , a mi Madre por ayudarme a no desfallecer en momentos de dificultad, a Víctor por su guía; extendiendo mi agradecimiento a la Ingeniera Ximena López que con su conocimiento, colaboración, apoyo y dedicación hizo posible la realización de este proyecto, a mis profesores por su conocimiento y guía, a mi compañero Francisco Paredes por la feliz culminación de nuestra tesis.

Ymir Acosta

ÍNDICE DE CONTENIDOS

CARÁTULA.....	i
CERTIFICADO	ii
DECLARACIÓN DE RESPONSABILIDAD	iii
AUTORIZACIÓN	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
ÍNDICE DE CONTENIDOS.....	vii
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE FORMATOS	xi
RESUMEN.....	xii
ABSTRACT.....	xiii
CAPÍTULO 1	1
FAMILIARIZACIÓN CON LA METODOLOGÍA QUE SE APLICA EN LA FAE.....	1
CAPÍTULO 2	10
MARCO TEÓRICO.....	10
2.1 Arquitectura	10
2.1.1 Arquitectura Cliente/Servidor.....	10
2.1.2 Diferencias	12
2.1.3 Ventajas.....	13
2.2 StarUml.....	13
2.2.1 Lenguaje Unificado de Modelado	14
2.2.2 ¿Por qué utilizar UML?.....	15
2.3 Lenguaje Python.....	16
2.3.1 Historia.....	17
2.3.2 Características	18
2.4 Wireframes	20
2.4.1 ¿Qué es en Wireframe?	20
2.4.2 ¿Para qué sirve?.....	21
2.4.3 ¿En qué proyectos se aplica?	21
2.5 Django Framework 1.6.....	22

2.5.1	Introducción	22
2.6	Especificación de requisitos (ERS) IEEE 830.	23
2.6.1	Introducción.	23
2.6.2	Objetivos ERS.....	24
2.6.3	Características.	24
2.7	Db Designer.	¡Error! Marcador no definido.
CAPÍTULO 3		28
DESARROLLO DE PRODUCTO SOFTWARE		28
3.1.	Metodología	28
3.1.1	Modelo Conceptual de Necesidad.....	28
3.1.2	Estándares Técnicos.....	32
3.1.3	Diseño.....	33
3.1.4	Construcción y pruebas.....	44
3.2.	Aplicación de la metodología	53
3.2.1	Modelo de Conceptual de Necesidad.....	53
3.2.2	Estándares Técnicos.....	60
3.2.3	Diseño.....	65
CAPÍTULO 4		78
LIBERACIÓN DEL PRODUCTO SOFTWARE		78
4.1	Descripción del procedimiento.....	78
4.2	Acta de satisfacción del producto software	79
4.3	Aceptación del producto software	80
CAPÍTULO 5		81
CONCLUSIONES Y RECOMENDACIONES		81
5.1	Conclusiones	81
5.2	Recomendaciones	81
Referencias Bibliográficas		82
ANEXOS		¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 1.	Nómina de personal.....	6
Tabla 2.	Distribución de Productos	7
Tabla 3.	Ingreso de Productos.....	7
Tabla 4.	Prestación Material de Campaña.....	8
Tabla 5.	Ingreso Material de Campaña	8
Tabla 6.	Versiones Python.....	17
Tabla 7.	Origen Django.....	23
Tabla 8.	Análisis de Bloques AP-MER.....	35
Tabla 9.	Análisis de Bloques AP-MREL.....	39
Tabla 10.	Análisis de Bloques AP-MOCK.....	42
Tabla 11.	Análisis de Bloques AP-DETER.....	46
Tabla 12.	Análisis de Bloques AP-AUCH.....	49
Tabla 13.	Análisis de Bloques AP-SOL.....	51
Tabla 14.	Clasificación de repositorios	62
Tabla 15.	Procedimiento Deploy Synaps	78

ÍNDICE DE FIGURAS

Figura 1. Kardex Excel.....	2
Figura 2. Hoja PE 2012.....	3
Figura 3. Recibo de Custodia Material.....	3
Figura 4. Oficio de requerimiento.....	4
Figura 5. Acta de Baja.....	5
Figura 6. Procedimiento Vinculado para la Sistematización.	6
Figura 7. Arquitectura Cliente-Servidor.....	10
Figura 8. Análisis sobre la utilización de UML.....	15
Figura 9. Wireframe Tipo.	22
Figura 10. Modelo Entidad-Relación.....	36
Figura 11. Estructura de un proyecto Django.....	63
Figura 12. Acta satisfacción del producto software.....	79
Figura 13. Acta de aceptación del producto software.	80

ÍNDICE DE FORMATOS

Formato 1. Modelo Conceptual	29
Formato 2. Modelo Entidad-Relación.....	36
Formato 3. Modelo Relacional	39
Formato 4. Mockup	43
Formato 5. Detección de Errores	46
Formato 6. Autorización de Cambio.....	49
Formato 7. Solución.....	51

RESUMEN

Una vez realizado el análisis de los procesos y la recepción de información histórica acerca de sistemas informáticos en la EPAE, se obtuvo como resultado que mencionada Escuela no cuenta con sistemas desarrollados por ella, como también metodología de desarrollo alguna; luego de identificar la carencia de estos dos factores y basándonos en la experiencia de desarrollos anteriores procedimos a proponer una metodología ágil bautizada SEPAE que apoye el rápido y bien documentado desarrollo de un producto software denominado Synaps que contribuya con el correcto desempeño de la dependencia, por motivación propia de los autores de este proyecto y por la tendencia actual de desarrollo de sistemas hacia nuevos lenguajes de programación se determinó que el lenguaje a desarrollar en el presente trabajo sería Python, este lenguaje soporta múltiples paradigmas de programación y además es multipropósito, de la mano con una herramienta llamada Django que es un Framework de desarrollo web que, en su versión 1.6 centra su atención en el principio DRY para generar aplicaciones fáciles de mantener y que rindan bajo el manejo de un nivel elevado de información; basándonos en la utilización de estas dos herramientas se obtuvo el producto software requerido.

Palabras claves: Metodología de desarrollo, producto software, lenguaje de programación, Framework Django, Python.

ABSTRACT

Once the process analysis was made and the reception of the historical information about the computer systems in EPAE, it showed that this dependency does not have developed systems made by itself and any development methodology, either. Furthermore, after checking the absence of these two factors and based on the experience of previous developments, we proceeded to propose an agile methodology called SEPAAE which can help the fast and well documented developments of the informatics solution named Synaps and contribute to the correct performance of the dependency. According to self-motivation of the authors of the following project and because of the current tendency of the systems development towards new languages programming, we determined that the language to develop the present job will be Python, this program supports multiple programming paradigms and it is multipurpose too, it will work together with a tool called Django that is a development framework web in its version 1.6 focuses its attention in DRY principle to generate easy applications to maintain and last under the management of a high level of information. Based on the use of these two tools we got the software required product.

Key Words: Software development methods, software program, programming languages, Django Framework, Python.

CAPÍTULO 1

FAMILIARIZACIÓN CON LA METODOLOGÍA QUE SE APLICA EN LA FAE.

El desarrollo de la aviación militar es acelerado y por tanto, es indispensable que el talento humano que está a cargo del mantenimiento y funcionamiento de equipos de la aviación de combate, aviación de rescate y aviación de transporte que son tan delicados y costosos se perfeccionen adecuadamente y sean capaces de aportar responsablemente al desarrollo y seguridad del país.

La Fuerza Aérea Ecuatoriana durante los últimos años a través del Instituto Tecnológico Superior Aeronáutico ITSA, ha venido dictando los cursos de perfeccionamiento que el personal de aerotécnicos en los diferentes grados debe realizar para cumplir los requisitos de ascenso establecidos en la Ley de Personal de las Fuerzas Armadas en su Art. 117 lit. b).

Por necesidades institucionales y considerando que el ITSA es un Instituto de Educación Superior que se rige por la normativa establecida por el antiguo CONESUP llamado hoy SENECHYT, se determina que los cursos de perfeccionamiento para los aerotécnicos de la FAE deben ser dictados en la Escuela creada exclusivamente para este fin, conforme lo establece el Reglamento General de Educación de la Fuerza Aérea.

El 31 de enero del 2006, el COSEFA analiza la solicitud presentada por el señor Rector del ITSA, relacionada a los trámites de creación y legalización de la EPAE. En este Consejo de Educación de la Fuerza Aérea aprueba la creación de ésta última de conformidad a las resoluciones tomadas en sesión del 17 de mayo del 2006. El viernes 09 de Junio 2006 por disposición del Comandante General de la Fuerza Aérea se realiza la ceremonia militar donde queda oficialmente creada la Escuela de Perfeccionamiento de Aerotécnicos EPAE [1].

Cabe indicar que la Escuela de Perfeccionamiento de Aerotécnicos utiliza como sistema operativo base Windows 7 Professional, el cual es específicamente utilizado por defecto en dependencias de Fuerza Aérea para manejo de documentación administrativa. Para la ejecución de este primer apartado se da inicio a una investigación encaminada a la recopilación de datos históricos o documentación en la cual se evidencie la utilización de una metodología de desarrollo de software, obteniendo como resultado que la dependencia EPAE como sistemas de aplicación es dueña de un sistema de control de notas, alumnos y cursos de los cuales ella es responsable, este sistema en la actualidad se encuentra en desuso y el proceso para el que fue desarrollado se realiza de forma manual, luego de conocer el estado actual del sistema se procedió a buscar información de algún proceso o metodología de desarrollo, se obtuvo como resultado que no se cuenta con ningún tipo de documentación donde conste el procedimiento o metodología que fue utilizada para el desarrollo del mencionado sistema y mucho menos, para el control y distribución de productos en el departamento de logística, este departamento es el encargado de suplir las necesidades a todos los departamentos de la escuela, utilizado para ello hojas de cálculo en Excel para llevar el inventario de productos existente en la bodega como muestra la Figura.1 Kardex Excel.

PARTIDA 53-08-05 MATERIA DE ASEO
KARDEX

DESCRIPCION			PARTIDA : 53-08-05		TARJETA No.			EXISTENCIAS							PRECIOS				
FRANELA ROJA			MATERIAL DE ASEO					INGRESOS			EGRESOS			SALDO BODEGA					
CORTADA DE 1 METRO			DE QUIEN O		DOCUMENTO			SERV.	REP.	MISCE	COND.	ENTREG.	REP.	MISCL.	SERV.	REP.	MISCEL	VALOR UNIT	VALOR TOTAL
MES	DIA	COMPROBANTE	PARA QUIEN	NUMERO DE	NUMERO DE	6	7	8	9	10	11	12	13	14	15	16	17		
1	12	2		37694				8									8	1.30	10.40
2	2011																		
3	4	4		67733				25									25	1.61	40.48
4	4	8		67625	AET			18									33	1.61	29.14
5																	51		
6	7	29			AET												18		33
6	3	13															15		18
7	5	4															1		17
8	5	16															2		15
9	5	18															1		14
10	8	22															4		10
11	7	11															2		8
12	10	3															2		6
13	10	26															2		4
14	2012																		
15	4	17		7120														12	16
16	4	22															1		15
17	5	16															1		14
18	8	21		7627														10	24
19	11	12		8396														24	1.58
20	11	30		1355														1	48
21	5	31																1	49
22	6	22																1	49
22	9	9																2	46
23	9	9																1	45
24	9	24																	1
25	11	6																	1
	11	16																	2
	12	4																	1
	2013																		1
	2	8																	1
	2	8																	1
	3	18																	1
	3	18																	1
	3	18																	1
	3	18																	3
	3	18																	3
	3	18																	3

Figura 1. Kardex Excel
Fuente. (Dpto. Logística EPAE). 2010

De igual manera la entrega de productos se lo realiza con una hoja denominada PE 2012, en la cual consta tanto cantidad, descripción, denominación del producto a entregar como se observa en Figura. 2 Hoja PE 2012.

ENTREGA DE MATERIAL		Fecha: 31-may-14	Documento N.º: 525140074			
De: ABASTECIMIENTOS EPAE		Fecha embarque: 31-may-14				
Para: SGOP. CHASILOA HERNAN DPTO. SEGUIMIENTO Y EVALUACION ACADEMICA		Forma embarque: Personal				
Aeronaves:		Observacion:				
Material	Ref.	Mon.	Cant.	U/E	Accion Abas.	
PAPEL BOND 75 GR.	I	USA	10	RS	TS-10	
PORTAMINAS	I	USA	3	EA	TS-03	
ESFEROS COLOR AZUL	I	USA	2	EA	TS-02	
CORRECTOR	I	USA	3	EA	TS-03	
CINTA DE EMBALAJE	I	USA	3	EA	TS-03	
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX						
Códigos de referencia: I: Inicial; Z: Reemplazo; S: Servible; R: Reparable; C: Condenado; B: Baja						
PEDIDO INTERNO			PEDIDO EXTERNO			
Visto Bueno (Fecha) TCRN VICTOR CLAVIJO SUBDIRECTOR DE LA EPAE		Cant. Recib. por:	Fecha:	Firma:		
Entrega (Fecha) SGOP. ENCALADA WILMER ENCARGADO BODEGA EPAE		Cant. Legall. por:	Fecha:	Firma:		
Recibe (Fecha) SGOP. CHASILOA HERNAN DPTO. SEGUIMIENTO Y EVALUACION ACADEMICA		Cant. Verifl. por:	Fecha:	Firma:		

FORM FAE PE 2012

Figura 2. Hoja PE 2012
Fuente. (Dpto. Logística EPAE). 2014

Para el préstamo de productos se trabaja con la entrega de un recibo de custodia, constando los respectivos ítems de información indicados en Figura. 3 Recibo de Custodia Material.

FUERZA AEREA ECUATORIANA					
ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS			SECCION ABASTECIMIENTOS EPAE		
R E C I B O D E C U S T O D I A					
OBJETIVO:		GRADO NOMBRE:			
FECHA:		C.I.			
ORD.	ARTICULO	CANT.	U/E	ACCION	CONDICION
					Nuevo
					Buen Estado
					Mal Estado
					Condenado
AUTORIZADO POR:		ENTREGADO PO	RECIBI	CONFORME:	

Figura 3. Recibo de Custodia Material
Fuente. (Dpto. Logística EPAE). 2014

El proceso en general se lo activa mediante un oficio dirigido al supervisor de departamento de logística EPAE como muestra la Figura. 4 Oficio de requerimiento. Se verifica la existencia de suministros en la hoja correspondiente al producto en Excel (anteriormente indicada), luego se realiza la entrega y se ejecuta el registro de la transacción actualizando las existencias del mismo.


FUERZA AÉREA


ESCUELA DE PERFECCIONAMIENTO DE AEROTÉCNICOS

Memorando No. FA-EX-H-C-2014-0023-O
Latacunga, 27 de mayo del 2014

PARA : Sgop. Téc. Avc. Wilmer Encalada
SUPERVISOR DPTO. LOGÍSTICA EPAE.

ASUNTO : Requerimiento Material Reproducción

Por el presente sírvase usted señor Sargento Primero, disponer a quien corresponda se entregue material de reproducción y material de oficina al Dpto. Seguimiento y Evaluación Académica, cabe indicar que dicho material será utilizado en la reproducción de los verificadores finales del XXXIV de Gerencia Administración Militar y V Curso de Promoción; conforme siguiente detalle:

ORD.	ARTICULO	CANT.	
01	Resmas de papel bond tamaño A-4	10 EA	
02	Flash memory	02 EA	NO HAY
03	Porta minas 0.5 mm (lápiz)	03 EA	
04	Esferográficos (Azul)	02 EA	
05	Liquid paper	03 EA	
06	Cinta scotch (Delgada)	02 EA	NO HAY
07	Cinta de embalaje transparente	03 EA	

Atentamente,
DIOS, PATRIA Y LIBERTAD


 Víctor Clavijo Rodríguez
 Teniente Coronel EMT. Avc.
SUBDIRECTOR EPAE.

Copia. :

- Archivo

 FSV/H. Chasiloa

Figura 4. Oficio de requerimiento
 Fuente. (Dpto. Logística EPAE). 2014

El departamento logístico de la EPAE es administrado por personal de aerotécnicos, los cuales cumplen la función de: un supervisor, un ayudante y una persona de bodega el último es encargado de controlar y distribuir los suministros a los demás departamentos de la escuela, el aerotécnico debe permanecer en el puesto alrededor de 1 año existiendo rotación del personal, por el mismo motivo que se maneja adquisición del productos y materiales según la regulación de pases dispuesta por la Dirección General

de Talento Humano FAE, o de acuerdo a las necesidades institucionales si el caso lo amerita.

El procedimiento para mantener el abastecimiento total en bodega, se realiza por medio de la planificación del PAPP (Plan Anual de Política Pública) del año siguiente.

El manejo del stock de los productos de bodega se maneja de similar manera considerando las fechas de caducidad de cada producto, realizando la constatación física con el estado de ítem como muestra la Figura. 5 Acta de Baja.

El Ecuador ha sido, es y será País Amazónico

FUERZA AÉREA ECUATORIANA
ESCUELA TÉCNICA DE LA FUERZA AEREA

ACTA DE BAJA

En la ciudad de Latacunga, a los 20 días del mes del Agosto del 2009, según recomendación por parte de Auditoría la misma que se realizó en la Escuela Técnica de la Fuerza Aérea por el periodo comprendido del 2007-05-01 al 2009-06-30, se procede a dar de baja Material de Campaña y Prendas Militares (fungibles), que por su utilización cumplieron con su vida útil por un valor de USD. 21401,10, y que a continuación se detalla:

ITEM	DESCRIPCION	CANT	UE	CONDICION	P.UNIT	P.TOTAL
1	BANDOLERAS	16	EA	MAL ESTADO	5	90
2	CHARRETERAS	92	PR	MAL ESTADO	2	104
3	EVILLA	48	EA	MAL ESTADO	6	288
4	CORDONES BICOLOR	49	EA	MAL ESTADO	7	280
5	GORRAS AZUL USADAS	38	EA	MAL ESTADO	60	2280
6	PAÑOLETAS AZULES	180	EA	MAL ESTADO	1,5	270
7	TERNO GRAN PARADA	125	EA	MAL ESTADO	73,8	9408
8	LINTERNAS	3	EA	MAL ESTADO	5	15
9	PORTA FUSIL BLANCO	3	EA	MAL ESTADO	7	21
10	CANTIMPLORA CON ESTUCHE	75	EA	MAL ESTADO	6	456
11	MOCHILAS VERDE	83	EA	MAL ESTADO	30	2490
12	MEDIAS CARPAS DE LONA	4	EA	MAL ESTADO	30	120
13	SLEEPING	1	EA	MAL ESTADO	23	23
14	CANTIMPLORA SIN ESTUCHE	70	EA	MAL ESTADO	4	280
15	OCHO DESCENSO	42	EA	MAL ESTADO	12	504
16	MOSQUETON	24	EA	MAL ESTADO	14	336
17	VAJILLA DE CAMPANA	91	EA	MAL ESTADO	20	1820
18	JARROS DE CAMPANA	66	EA	MAL ESTADO	4	264
19	MACHETE CON VAINA	67	EA	MAL ESTADO	9	603
20	GUANTES DE CUERO	7	PR	MAL ESTADO	5	35
21	MACHETES	21	EA	MAL ESTADO	6	126
22	LINTERNA	37	EA	MAL ESTADO	6,8	251,6
23	BRUJULAS	132	EA	MAL ESTADO	8	1056
24	CHALECCOS DE SEGURIDAD	4	EA	MAL ESTADO	4,5	18
25	PAÑOLETAS AZUL	31	EA	MAL ESTADO	1,5	46,5

① PAPP
② AH (65.245)
con IVA incluido

Figura 5. Acta de Baja
Fuente. (Dpto. Logística EPAAE). 2009

La Escuela de Perfeccionamiento de Aerotécnicos en el procedimiento de entrega de productos, habilita a la persona en la función hoy en día asignada a tal función, las personas mencionadas se ilustran en la Tabla 1. Nómina de personal.

Nómina de personal autorizado a legalizar productos

Tabla 1. Nómina de personal.

DEPARTAMENTO	NOMBRE	FUNCIÓN
Subdirección	Tcrn. Clavijo V.	Subdirector
Administración Académica	Mayo. Flores M.	Jefe
Planificación Académica	Sgop. Mendoza W.	Supervisor
Seguimiento y Evaluación Académica	Subs. Sevilla F.	Supervisor
Dpto. Talento Humano	Sgop. Claudio H.	Supervisor
Dpto. Finanzas	Tnte. Sisa M.	Jefe
Dpto. Sicom	Sgop. Quilumba F.	Supervisor
Dpto. Sis.	Sgos. Chiluisa	Supervisor
Dpto. CC.SS.	Sgop. Zuñiga C.	Supervisor

Elaborado por (Acosta Y. / Paredes J.)

Fuente. (EPAE 2014)

El procedimiento de administración de préstamo de productos, puede activarlo cualquier persona de la escuela con la legalización del supervisor del departamento de Logística EPAE, que en el momento de recibir dicho material tiene la responsabilidad total en la custodia del bien como ilustra la Figura 6.

Procedimiento Vinculado para la Sistematización

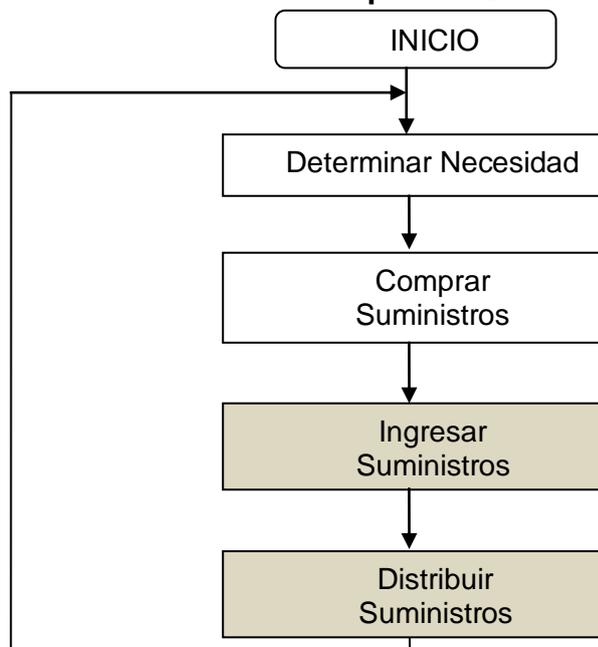


Figura 6. Procedimiento Vinculado para la Sistematización.

Fuente. (Acosta Ymir / Paredes Juan)

Tabla 2. Distribución de Productos

ORD.	DETALLE	OPERACIÓN					DISTANCIA		OBSERV. TECNICA
		○	□	⇒	D	▽	mts	cm	
1	Realizar oficio	●							
2	Legalizar oficio	●							
3	Autorizar oficio	●	●						
4	Enviar a sellar oficio			●			2		
5	Entrega pedido a bodega		●						
6	Revisar inventario		●						
7	Registrar egreso	●				●			Hoja Excel kardex
8	Generar PE 2012	●							
9	Entregar suministro	●							
10	Legalizar entrega	●							
11	Archivar analógicamente					●			PE 2012

Elaborado por: (Acosta Y. / Paredes J.)

Fuente. (EPAE 2014)

Tabla 3. Ingreso de Productos

ORD.	DETALLE	OPERACIÓN					DISTANCIA		OBSERV. TECNICA
		○	□	⇒	D	▽	mts	cm	
1	Revisar factura		●						
2	Constatar y revisar producto	●	●		●				
3	Acusar conformidad	●							
4	Enviar factura (Recibo conforme)			●			1		
5	Actualizar Kardex	●				●			Hoja Excel kardex
6	Archivar factura correspondiente					●			factura no cancelada

Elaborado por: (Acosta Y. / Paredes J.)

Fuente. (EPAE 2014)

Tabla 4. Prestación Material de Campaña.

ORD.	DETALLE	OPERACIÓN					DISTANCIA		OBSERV. TECNICA
		○	□	⇒	D	▽	mts	cm	
1	Verificar material en bodega		●						
2	Registrar préstamo	●				●			
3	Generar recibo de custodia	●							
4	Entregar material de campaña	●							
5	Legalizar recibo de custodia	●							
6	Archivar analógicamente					●			

Elaborado por: (Acosta Y. / Paredes J.)

Fuente. (EPAE 2014)

Tabla 5. Ingreso Material de Campaña

ORD.	DETALLE	OPERACIÓN					DISTANCIA		OBSERV. TECNICA
		○	□	⇒	D	▽	mts	cm	
1	Revisar factura		●						
2	Constatar y revisar el material	●	●		●				
3	Acusar conformidad	●							
4	Enviar factura (Recibo conforme)			●			1		
5	Actualizar inventario	●				●			Hoja Excel kardex
6	Archivar factura correspondiente					●			

Elaborado por: (Acosta Y. / Paredes J.)

Fuente. (EPAE 2014)

Todos los productos se encuentran en custodia del departamento de Logística EPAE, para la administración y dotación a los diferentes departamentos y personal militar respectivamente.

El equipo de aviación tiene procedimientos específicos que exige la casa productora de aeronaves, quedando el almacén como un espacio de transición en el instante que se requiera generar un overhaul de tercer y cuarto escalón; el mismo que es realizado fuera del país, en su traslado se adjunta kardex de la vida del mismo, cambiándose el custodio del bien y activándose el seguro de traslado.

Por lo anotado, la sistematización en el entorno de este bien no amerita involucrar esfuerzo, recurso y procedimiento de sistematización alguno.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Arquitectura

2.1.1 Arquitectura Cliente/Servidor

Una arquitectura de manera general, es una mezcla de componentes funcionales que contiene diferentes estándares, convenciones, reglas, procesos, lo cual permite integrar una amplia gama de productos y servicios tanto tecnológicos como informáticos, para que sean utilizados eficazmente en el entorno de la organización.

La arquitectura cliente-servidor es un modelo que nos permite construir sistemas de información, que se fundamenta en la repartición de la información y el tratamiento de los datos que se transmiten por todo el sistema informático, que a su vez permite mejorar el rendimiento del sistema general de información.

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información de forma transparente aún en entornos multiplataforma. Se trata pues, de la arquitectura más extendida en la realización de Sistemas Distribuidos [2], según se ilustra en la Figura 7.

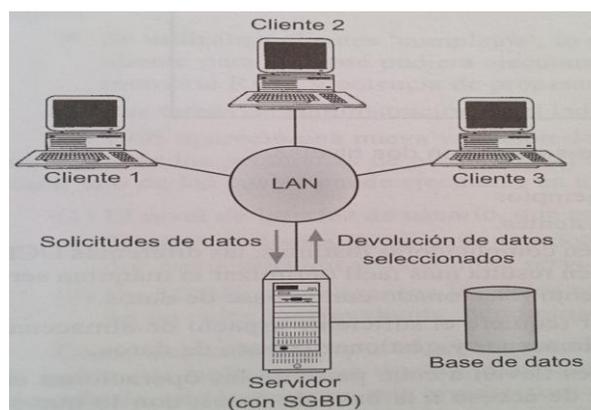


Figura 7. Arquitectura Cliente-Servidor
Fuente (Connolly, T.M; Begg, C.E., 2005)

Una base de datos es un almacén que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos [3].

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro [4].

El **equipo cliente** se suele encargar de la parte lógica y de mostrar la información al usuario que realiza una petición. El cliente MySQL en modo interactivo nos permite tanto la introducción de sentencias SQL para trabajar con la base de datos (crear tablas, hacer consultas y ver resultados.) como ejecución de comandos propios del SGBD (Sistema de Gestión de Base de Datos) para obtener información sobre las tablas, índices o ejecutar operaciones de administración, es todo proceso que reclama servicios de otro, por lo tanto cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor también se lo conoce como front-end.

El cliente específicamente manipula todas las funciones que se relacionan administración y despliegue de datos, ya que estos se desarrollan sobre plataformas que nos permiten construir interfaces gráficas de usuario (GUI), las funciones que lleva a cabo el proceso cliente son [5]:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

El **equipo servidor** MySQL Server, se encarga de administrar la base de datos, de gestionar los recursos del servidor. (CPU, memoria.), y por

supuesto de resolver y devolver en forma de solución la petición realizada por el cliente.

Un servidor es todo proceso que proporciona un servicio a otros. Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se lo conoce con el término back-end. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las principales funciones que lleva a cabo el proceso servidor se enumeran a continuación [6]:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

2.1.2 Diferencias

- Un servidor es una aplicación que ofrece un servicio a usuarios de Internet; mientras tanto el cliente es el que pide ese servicio.
- Tanto el cliente como el servidor pueden ejecutar aplicaciones en un mismo o diferente sistema.
- El cliente es el que realiza múltiples peticiones, mientras que el servidor es un programa que recepta esas peticiones o solicitudes, brinda el servicio que el cliente requiere y devuelve los resultados a un mismo tiempo de respuesta.
- El equipo que puede entrar y salir de cualquier router, ftp y es el encargado del administrar la red es el servidor, mientras que el cliente tiene acceso restringido, es así que puede entrar solo a las partes que el servidor le permita.

- Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte.

2.1.3 Ventajas

Los aspectos que más se han originado en el uso de sistemas Cliente/Servidor, son las plataformas de hardware cada vez más baratas, esta es la ventaja más relevante en este esquema, dando como resultado final la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

El esquema Cliente/Servidor nos ayuda a la integración entre diferentes sistemas permitiendo compartir información, a la vez que maquinas puedan utilizar interfaces mas amigables al usuario, integrando medianos y grandes sistemas con Pc's, sin necesidad de usar un mismo sistema operacional.

Los sistemas construidos bajo este esquema tienen mayor interacción con el usuario teniendo así mayor ventaja con uno que es centralizado ya que para éste no es necesario transmitir gráficamente la información por la red alojándose en el mismo cliente, permitiéndonos tener un mejor ancho de banda en la red.

Es fundamental que se pueda integrar nuevas tecnologías y de infraestructura computacional a la vez, esto se facilita porque su estructura es netamente modular por departamentos, beneficiando la escalabilidad como tal, proporcionando a la organización soluciones locales.

2.2 StarUml

StarUml es un proyecto de código abierto para un diseño rápido, flexible, extensible de funcionalidades, y de libre disposición plataforma UML / MDA que se ejecuta en la plataforma Win32. El objetivo del proyecto StarUML es construir una herramienta de modelado de software y también la plataforma que es un reemplazo convincente de herramientas UML comerciales como Rational Rose, Together.

UML 2.0: UML está en continua expansión estándar gestionado por OMG (Object Management Group) . Recientemente, UML 2.0 se libera y el apoyo StarUML UML 2.0 y apoyará estándar UML latest [7].

2.2.1 Lenguaje Unificado de Modelado

Este lenguaje es una herramienta que sirve para modelar un sistema tanto en hardware como software generando diagramas organizados que capten las ideas de todo un proceso, obteniendo como resultado final que todas las personas o miembros de un equipo de desarrolladores de software puedan tener una fácil comprensión de lo que pretenden hacer, deduciendo en una sola meta que el proceso unificado es orientar al equipo de desarrolladores en la implementación y distribución eficiente de sistemas que se ajusten a las necesidades de los usuarios.

UML es una abreviación para Lenguaje Unificado de Modelado, cada una de estas palabras habla sobre un aspecto importante del UML:

- **Lenguaje:** Permite comunicar el sujeto en el desarrollo de un sistema, este sujeto incluye los requerimientos y el sistema, es así que sin lenguaje es difícil comunicarse para los miembros de un equipo y colaborar en el desarrollo del sistema de manera satisfactoria.
- **Modelo:** Se refiere a la representación de un sujeto, el modelo captura un sin número de ideas llamadas abstracciones, es difícil para los miembros de un equipo tener un alcance de conocimiento generalizado de los requerimientos y del sistema sin tener un modelo que represente lo que debemos realizar. Es fundamental describir que un modelo UML puntualiza lo que un sistema hará en un supuesto, pero no dice como implementar dicho sistema.
- **Unificado:** Refiere al hecho de relacionar el Object Management Group (OMG), que es una organización de estandarización reconocida por la industria y el Rational Software Corporation, creando el UML para traer juntos los sistemas de información y las mejores prácticas de ingeniería

en la industria y tecnología. Estas prácticas conjugan técnicas que permiten desarrollar sistemas con la finalidad de mantener un fácil uso, simple, extensible e independiente del proceso y la implementación.

2.2.2 ¿Por qué utilizar UML?

Antes del advenimiento del UML, los programadores desarrollaban códigos de sistemas de acuerdo a los requerimientos planteados por un cliente, sin importar si el cliente y/o los usuarios lo entendían ó no, y en muchas ocasiones el producto final no era exactamente lo que el cliente necesitaba. El paradigma tradicional ignora los datos a favor de las operaciones. Por el contrario el paradigma orientado a objetos presta igual atención a las operaciones que a los datos. Del análisis del estudio realizado por la empresa de investigación Standish Group [8], sobre 280.000 proyectos desarrollados durante el año 2000 se obtuvieron los siguientes resultados como muestra la Figura 8.

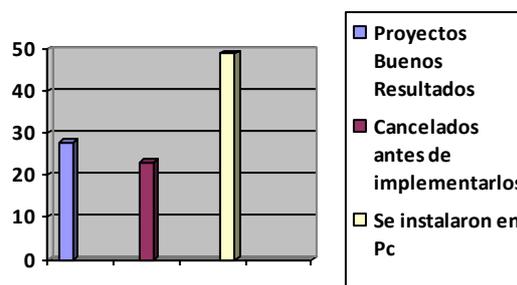


Figura 8. Análisis sobre la utilización de UML

Sin embargo, aquellos proyectos estaban por debajo del presupuesto, estaban retrasados o tenían menos características y funcionalidad que la especificada inicialmente. Estos resultados indican que para el año 2000 sólo uno de cada cuatro proyectos de desarrollo era exitoso.

Debido al avance tecnológico que hoy en día existe, es necesario contar con un plan de desarrollo bien estructurado y organizado, donde todas aquellas personas que se encuentran directamente relacionadas como: desarrolladores (realizan el código de un sistema), usuarios (utilizan

el producto final) y clientes (pagan por el desarrollo) entiendan lo que hará el sistema que se va a desarrollar. En tal virtud, un sistema que es elaborado bajo conceptos UML debe ser puntual, fácil de entender, cambiar y comunicar a otros sistemas.

Es así que; por lo dicho anteriormente, la ventaja de utilizar UML radica en que aún cuando los usuarios y clientes no conozcan de programación, serán capaces de entender los diagramas de especificaciones elaborados bajo este concepto que serán entregados a los desarrolladores, permitiendo que tanto clientes como usuarios podrán hacer modificaciones a las especificaciones presentadas hasta lograr que éstas cumplan con todas sus necesidades y que al momento de que el modelo describe lo que debe realizar el sistema, no le dice como implementar dicho sistema, es por eso que es independiente del lenguaje de programación que se utilice.

2.3 Lenguaje Python

Python es un lenguaje de programación multipropósito de alto nivel, su filosofía de diseño enfatiza la productividad del programador y la legibilidad del código. Tiene un núcleo sintáctico minimalista con unos pocos comandos básicos y simple semántica, pero además tiene una enorme y variada librería estándar, que incluye una Interfaz de Programación de Aplicaciones (API) para muchas de las funciones en el nivel del sistema operativo (OS). El código Python, aunque minimalista, define objetos incorporados como listas enlazadas (list), tuplas (tuple), tablas hash (dict), y enteros de longitud arbitraria (long).

Python soporta múltiples paradigmas de programación, incluyendo programación orientada a objetos (class), programación imperativa (def) y funcional (lambda). Python tiene un sistema de tipado dinámico y manejo automatizado de memoria utilizando conteo de referencias (similar a Perl, Ruby y Scheme) [9].

Básicamente, Python es un lenguaje de programación de alto nivel, interpretado y multipropósito como lo nombrado anteriormente. En los últimos años su utilización ha ido constantemente creciendo y en la

actualidad es uno de los lenguajes de programación más empleados para el desarrollo de software.

Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar los más populares, como Windows, Mac OS X y Linux, además Python puede funcionar en Smartphones, Nokia desarrollo un intérprete de este lenguaje para su sistema operativo *Symbian*. ¿Tiene Python un ámbito específico? Algunos lenguajes de programación sí que lo tienen. Por ejemplo, PHP fue ideado para desarrollar aplicaciones web. Sin embargo este no es el caso de Python. Con este lenguaje podemos desarrollar software para aplicaciones científicas, para comunicaciones de red, para aplicaciones de escritorio con interfaz gráfica de usuario (GUI), para crear juegos, para smartphones y por supuesto para aplicaciones web.

2.3.1 Historia

El origen de Python se remonta a principios de los noventa. Por este tiempo un investigador holandés llamado Guido van Rossum, que trabajaba en el centro de investigación CWI (Centrum Wiskunde & Informatica) de Ámsterdam, es asignado a un proyecto que consistía en el desarrollo de un sistema operativo distribuido llamado Amoeba. Por aquel tiempo, el CWI utilizaba un lenguaje de programación llamado ABC. En lugar de emplear este lenguaje para el proyecto Amoeba, Guido decide crear un nuevo que pueda superar las limitaciones y problemas que se había encontrado al trabajar en otros proyectos con ABC se ilustran las versiones en la Tabla 6.

Tabla 6. Versiones Python

Versiones	Año	Descripción	Observación
1.0	1994	Licencia OS	
1.6	2000	GNU	Mantenidas por separado
2.7	2010		
3.2	2011		

Fuente (Fernandez A, 2012)

Entre las características de las primeras versiones de Python cabe destacar el soporte de la orientación a objetos, el manejo de excepciones y el soporte de estructura de datos de alto nivel, como, por ejemplo, las listas y los diccionarios. Además, desde su desarrollo inicial, se tuvo en cuenta que el código escrito en este lenguaje fuera fácil de leer y de entender, sin que esto suponga renunciar a características y funcionalidades avanzadas.

Muchos se preguntan el origen del nombre de este lenguaje de programación, Guido van Rossum decidió darle este nombre en honor a la serie de televisión Monty Python's Flying Circus, de la cual era fan. Esta es una serie protagonizada por el grupo de humoristas Monty Python, famoso por películas como La Vida de Brian o El Sentido de la Vida. Desde el principio de su diseño, se pretendía que Python fuera un lenguaje que resulta divertido de utilizar, de ahí que en el nombre influyera la mencionada serie cómica. También resulta curioso que, tanto en tutoriales, como en ejemplos de código, se suelen utilizar referencias a los Monty Python. Por ejemplo, en lugar de emplear los tradicionales nombres de variables foo y bar, se suelen utilizar spam y eggs, en referencia a sketches de este grupo de cómicos.

El desarrollo y promoción de Python se lleva a cabo a través de una organización sin ánimo de lucro, llamado Python Software Foundation, que fue creada en marzo de 2001. Entre las actividades que realiza esta organización destacan el desarrollo y distribución oficial de Python, la gestión de la propiedad intelectual del código y documentos realizados, así como la organización de conferencias y eventos dedicados a poner en contacto a todas aquellas personas interesadas en este lenguaje de programación.

Python tiene un claro carácter open source y la Python Software Foundation invita a cualquiera que quiera hacerlo, a contribuir al desarrollo y promoción de este lenguaje de programación.

2.3.2 Características

Dos de las principales características del lenguaje Python son, por un lado que es *interpretado* y por otro lado, que es *multiplataforma*. Por lo

primero significa que no es necesario compilar su código para su ejecución, ya que existe un intérprete que se encarga de leer el fichero fuente y ejecutarlo. Gracias a este funcionamiento es posible ejecutar el mismo código en distintas plataformas y sistemas operativos sin necesidad de cambiar el código fuente, bastará con tener instalado el intérprete. Eso sí, la versión de este intérprete es nativa para cada plataforma. En este sentido, Python es similar Perl o a Ruby y difiere de otros lenguajes como C++ y Objective-C.

Habitualmente, a los programas en Python se les denomina scripts. En realidad scripts es el término que se suele emplear para los ficheros de código fuente escritos en Python, pudiendo un programa contar con uno o más de estos scripts.

La interacción con el intérprete del lenguaje se puede hacer directamente a través de la consola, durante la instalación de Python se instala un componente llamado *shell* o consola que permite ejecutar directamente código Python a través de una terminal o interfaz de comandos.

Python carece de tipos propiamente dichos, es decir, es un lenguaje de tipado dinámico. Los programadores de C++ y Java están acostumbrados a declarar cada variable de un tipo específico. Este proceso no es necesario en Python, ya que el tipo de cada variable se fija en el momento de su asignación. Como consecuencia de este hecho, una variable puede cambiar su tipo durante su ciclo de vida sin necesidad explícita de ser declarado, dado que puede ser interesante el tipo de una variable en un momento dado. Python nos ofrece una serie de funciones que nos dan este tipo de información.

Además de soportar la orientación a objetos, Python también nos permite otros paradigmas de programación, como por ejemplo, la programación funcional y la imperativa. En la actualidad, Python es considerado uno de los lenguajes que más facilidades ofrecen para enseñar programación orientada a objetos. A esto contribuye su sintaxis.

Con respecto a su sintaxis, una de las diferencias más destacables es el uso de la indentación. Diferentes niveles de indentación son utilizados

para marcar las sentencias que corresponden al mismo bloque. Por ejemplo, todas las sentencias que deban ser ejecutadas dentro de un bloque *if* llevaran en mismo nivel de indentación, mientras que el resto utilizaran un nivel diferente, incluida la sentencia que contiene la condición o condiciones del mencionado *if*. Además, cada sentencia no necesita un punto y coma (;), como si ocurre en lenguajes como C/C++, PHP y Java, en Python basta con que cada sentencia vaya en una línea diferente. Por otro lado, tampoco se hace uso de las llaves ({}), para indicar el principio y fin de bloque. Tampoco se emplean palabras clave como *begin* y *end*. Simplemente se usan los dos puntos (:) para marcar en comienzo de bloque y el cambio de indentación se encarga de indicar el final.

Para facilitar la programación, Python incluye una serie de estructura de datos de alto nivel, como son, por ejemplo, las listas, los diccionarios, cadenas de texto (strings), tuplas y conjuntos. Por otro lado, su librería estándar incorpora multitud de funciones que pueden ser utilizadas en diversos ámbitos, entre ellas podemos mencionar, desde aquellas básicas para manejar strings, hasta las que pueden ser usadas en programación criptográfica, pasando por otros de nivel intermedio como son las que permiten manejar ficheros ZIP, trabajar con ficheros CSV o realizar comunicaciones de red a través de protocolos estándar, todo ello sin necesidad de instalar librerías adicionales. Comúnmente se emplea la frase *batteries included* para resaltar este hecho [10].

2.4 Wireframes

2.4.1 ¿Qué es en Wireframe?

Un wireframe es un bosquejo de la ubicación de elementos en una pantalla. Dependiendo de la etapa de desarrollo del proyecto, generamos el wireframe básico, que expresa los elementos de una pantalla genérica, aplicable a todo un sistema; wireframe tipo, que expresa la estructura de pantallas de una misma categoría, como canales o páginas terminales; y wireframe detallado, que especifica la composición de páginas específicas (se aplica especialmente a pantallas que implican la realización de acciones).

A ellos debe sumarse, para ciertos proyectos en que tan relevante como ubicar los elementos en pantalla es simular el funcionamiento, los wireframes funcionales, que no se limitan a una estructura estática, sino que también permiten la navegación.

Un wireframe grafica básicamente:

- **Elementos de navegación:** menús, accesos directos e hipervínculos.
- **Elementos de información:** contenidos de texto e imágenes.
- **Elementos de interacción:** herramientas o funcionalidades que el usuario puede realizar.
- **Elementos de apoyo:** ítems de ayuda y orientación, como mapas de navegación o faqs.
- **Elementos promocionales:** espacio dedicado a banners publicitarios o a destacados internos del propio producto.

2.4.2 ¿Para qué sirve?

Los wireframes sirven para anticipar cuál será la estructura de cada página de un sistema, antes de aplicar el diseño gráfico definitivo. Dado que el wireframe posee una composición bastante sencilla, generalmente en blanco y negro sin mayores detalles, resulta fácil su modificación para analizar diferentes posibilidades de ubicación de elementos.

Es en este esquema que el equipo de diseño gráfico debe basarse para la elaboración del producto definitivo. Las propuestas de interfaz gráfica no pueden o no deben partir desde cero, dada la complejidad de su elaboración (y su posterior modificación en caso de requerirse). Tienen un mínimo común denominador que es la estructura. Cuando un wireframe es aprobado por el cliente, los comentarios posteriores sobre la presentación de una página pueden estar asociados a la percepción de diseño, uso de colores, tipografía. Pero ya no debiera cuestionarse la estructura y disposición de elementos de una página.

2.4.3 ¿En qué proyectos se aplica?

Cualquier proyecto web, desde el más sencillo hasta el más complejo, debe contemplar la realización al menos de un wireframe tipo. Por

sofisticada que hoy en día resulte esta herramienta, el wireframe sigue cumpliendo la función de ser un bosquejo. Y siempre será más sencillo hacer ajustes a un bosquejo que a un producto final, para anticipar el coherente desarrollo de los procesos antes de proceder a su programación y diseño definitivos [11].

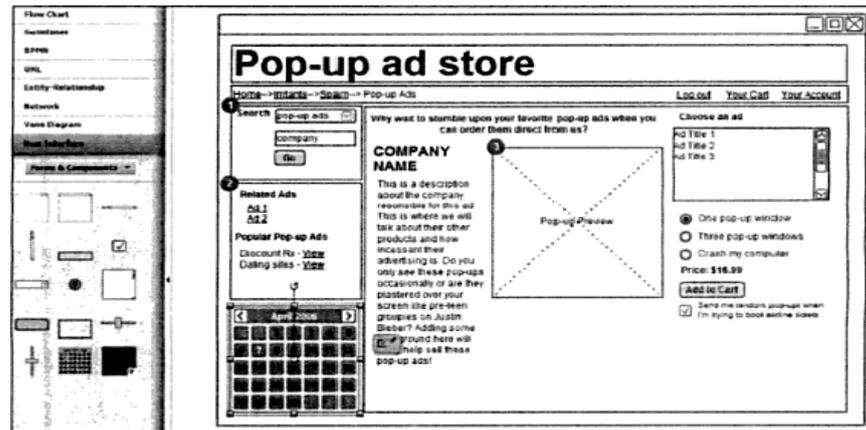


Figura 9. Wireframe Tipo.
Fuente (Didier Mazier, 2011)

2.5 Django Framework 1.6

2.5.1 Introducción

Django nació naturalmente de aplicaciones de la vida real escritas por un equipo de desarrolladores web en Lawrence, Kansas. Nació en 2003 cuando los desarrolladores web del diario Lawrence Journal-World, Adrian Holovaty y Simon Willison, comenzaron a usar Python para crear sus aplicaciones. El equipo de The World Online, responsable de la producción y mantenimiento de varios sitios locales de noticias, prosperaban en un entorno de desarrollo dictado por las fechas límite del periodismo [12] como se muestra en la Tabla 7.

Continúa →

Tabla 7. Origen Django

Origen	Año	Desarrollador web	Lugar	Logo
Aplicaciones de la vida real escritas	2003	Adrian Holovaty/ Simon Willison	Lawrence	 (J.Forcier, P.Bissex, W. Chun 2008)
Framework	2005	Equipo World online/ Jacob Kaplan-Moss		

Django nació en un entorno de noticias, ofrece varias características (en particular la interfaz admin) que son particularmente apropiadas para sitios de *contenido*, sitios como eBay, craigslist.org y washingtonpost.com que ofrece información de base de datos.

Como resultado de eso, Django es activamente mejorado casi diariamente. Los desarrolladores del framework tienen un alto grado de interés en asegurarse de que Django les ahorre a los desarrolladores, produzca aplicaciones que son fáciles de mantener y rindan bajo mucha carga [13].

2.6 Especificación de requisitos (ERS) IEEE 830.

2.6.1 Introducción.

El análisis de requisitos es una de las tareas más importantes en el ciclo de vida del desarrollo de software, puesto que en ella se determinan los planos de la nueva aplicación.

El análisis de requisitos se puede definir como el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos, definición proporcionada por el IEEE. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado [14].

Así pues, el documento de especificación de requisitos debe ser legible por el cliente, con lo que se evita el malentendido de determinadas situaciones, ya que el cliente participa activamente en la extracción de dichos requisitos.

2.6.2 Objetivos ERS.

Los principales objetivos que se identifican en la especificación de requisitos software son:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software: El cliente debe participar activamente en la especificación de requisitos, ya que tiene una visión mucho más detallada de los procesos que se llevan a cabo. Así mismo, el cliente se siente partícipe del propio desarrollo.
- La ERS permite al cliente definir todos los requisitos que desea y al mismo tiempo los desarrolladores tienen una base fija en la que trabajar. Si no se realiza una buena especificación de requisitos, los costes de desarrollo pueden incrementarse considerablemente, ya que se deben hacer cambios durante la creación de la aplicación [15].

2.6.3 Características.

Las características deseables para una buena ERS en el IEEE son las siguientes:

- ***Corrección***

La ERS es correcta si y sólo si todo requisito que figura en ella refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.

- ***Ambigüedad***

Cada característica del producto final debe ser descrita utilizando un término único y, en caso de que se utilicen términos similares en distintos contextos, se deben indicar claramente las diferencias entre ellos. Incluso se puede incluir un glosario en el que indicar cada significado específicamente.

Los analistas deben poner un cuidado especial a la hora de especificar los requisitos. El hecho de utilizar el lenguaje natural para hacer la ERS comprensible a los usuarios supone un riesgo muy elevado, porque el lenguaje natural puede llegar a ser muy ambiguo.

En términos generales, el lenguaje natural es de los más ambiguos. Por el contrario existen los lenguajes formales que no son ambiguos, pero son más difíciles de aprender y menos comprensibles para el que no los conoce.

- **Compleitud**

Una ERS es completa si:

Incluye todos los requisitos significativos del software (relacionados con la funcionalidad, ejecución, diseño, atributos de calidad o interfaces externas).

Existe una definición de respuestas a todas las posibles entradas, tanto válidas como inválidas, en todas las posibles situaciones.

Aparecen etiquetadas todas las figuras, tablas, diagramas, así como definidos todos los términos y unidades de medida empleados.

- **Verificabilidad**

Un requisito es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina puedan verificar que el software satisface dicho requerimiento.

- **Consistencia**

Una ERS es consistente si y sólo si ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto. Se pueden dar tres casos:

- Requisitos que describen el mismo objeto real utilizando distintos términos.

- Las características especificadas de objetos reales. Un requisito establece que todas las luces son verdes y otro que son azules.
- Conflicto lógico o temporal entre dos acciones determinadas. Se llega a un punto en el que dos acciones serían perfectamente válidas.

- **Clasificación**

No todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por diversos criterios:

- Importancia: Pueden ser esenciales, condicionales u opcionales.
- Estabilidad: Cambios que pueden afectar al requisito.

Lo ideal es el establecimiento de prioridades, de modo que la implementación de un requisito de menor prioridad no emplee excesivos recursos.

- **Modificabilidad**

Una ERS es modificable si cualquier cambio puede realizarse de manera fácil, completa y consistente. Para ello, es deseable tener una organización coherente y fácil de usar en la que aparezca el índice o una tabla de contenidos fácilmente accesible.

- **Explorabilidad (Traceability)**

Una ERS es explorable si el origen de cada requerimiento es claro tanto hacia atrás (origen que puede ser un documento, una persona.) como hacia delante (componentes del sistema que realizan dicho requisito).

- **Utilizable Durante Las Tareas De Mantenimiento Y Uso**

En la ERS también se deben tener en cuenta las necesidades de mantenimiento. El personal que no ha intervenido directamente en el desarrollo debe ser capaz de encargarse de su mantenimiento. Así, dicha

ERS actúa a modo de plano de la aplicación, permitiendo incluso modificaciones que no requieran un cambio en el diseño [16].

DBDesigner 4 es un sistema de diseño de base de datos visual que integra el diseño de bases de datos, modelado, creación y mantenimiento en un único entorno sin fisuras. Combina características profesionales y una interfaz de usuario clara y sencilla para ofrecer la forma más eficiente de manejar sus bases de datos, es desarrollado y optimizado para el código abierto MySQL-Base de datos para apoyar a los usuarios de MySQL con una herramienta de diseño de gran alcance y gratuita.

Todas las características de MySQL han sido reforzadas para ofrecer la forma más conveniente para diseñar y mantener el control de la base de datos MySQL, es un proyecto Open Source que son bienvenidos a convertirse en un miembro de la comunidad para compartir sus experiencias, informar de errores y enviar solicitudes de características, ofrece características de gran alcance para crear un modelo visual de cualquier base de datos.

A partir de un motor de ingeniería inversa para recuperar automáticamente un modelo de bases de datos existentes, amplias herramientas de modelado y editores a una función para sincronización que aplica los cambios del modelo de forma automática a la base de datos subyacente que todo es parte de DBDesigner 4 [17].

CAPÍTULO 3

DESARROLLO DE PRODUCTO SOFTWARE

3.1. Metodología

La metodología que se utilizará para construir el producto resultado de este trabajo teórico-práctico y sirviendo la poca información proporcionada en la FAE y el conocimiento aplicativo de metodologías ágiles nos permite construir la metodología de desarrollo para EPAE denominada SEPAE, está constituida de cuatro fases documentales hasta lograr la liberación del producto generado que son:

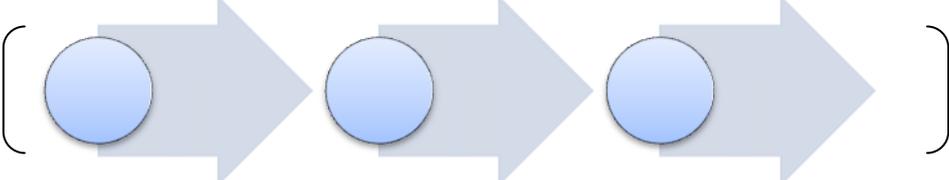
- Modelo conceptual de necesidad
- Estándares Técnicos.
- Diseño
- Construcción y pruebas

3.1.1 Modelo Conceptual de Necesidad

Esta fase nos permitirá conocer, documentar, validar y aprobar las necesidades requeridas para constituir el producto software desde el punto de vista operativo de la línea de negocio y dentro del marco técnico del desarrollo de la ingeniería de software, como parte de esta fase tenemos algunos artefactos que la constituyen siempre apoyándonos en estándares internacionales que nos garantizan la madurez de la toma de información, con este fin se utilizan los siguientes formatos.

Continúa →

Formato 1. Modelo Conceptual

Sistema	Fecha	Versión
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato [#.#.#]
Descripción General:		
En este apartado se deberán identificar y caracterizar el aspecto, situación, evento o manejo que genere características negativas en la línea de negocio; de ser posible plasmar ventajas y desventajas del proceso físico. Formato[x..... 200 o másx]		
Funcionalidad		
Aquí se deberá detallar en si el proceso o la gestión de agente central de la razón de ser del producto software. Formato[x..... 200 o másx]		
Restricciones		
En este apartado se deberán registrar todas las restricciones que representen una interrupción del flujo esperado del proceso, pueden ser de dos tipos: <ul style="list-style-type: none"> • Limitaciones de ley, enunciando la misma. • Limitaciones de naturaleza agente. 		
Grafico		
Aquí se deberá plasmar gráficamente el proceso descrito en la funcionalidad. Formato		
		
Realizado Por	Autorizado Por	
Firma Nombre	Firma Nombre	
Fecha	Fecha	
Formato [##/##/####,DD/MM/YYYY]	Formato [##/##/####,DD/MM/YYYY]	
Formato: AP-MCN		

Fuente (Acosta Ymir / Paredes Juan)

El Formato 1, ofrece una visión general del proceso físico que es llevado a cabo en la dependencia, permite determinar funcionalidades vitales del área de negocio y las posibles restricciones que estas manejan; mediante la utilización de este formato se entabla una comunicación clara y directa con el usuario acerca de las áreas que se pretenden sistematizar, por este

motivo en este apartado se utilizara únicamente lenguaje natural evitando posibles ambigüedades y palabras técnicas que puedan confundir al usuario o apartarlo del objetivo principal que es la descripción del proceso.

Luego de tener una descripción del proceso físico se procede a la fase de levantamiento de requerimientos del sistema, en lo referente a la descripción como tal de un requerimiento se utilizarán los consejos proveídos por el estándar ANSI/IEEE 830, 1998, con el **Anexo B** se indican un número de reglas que se deben seguir para que los requerimientos se encuentren correctamente estructurados y enunciados.

- **Correcto:** Un requerimiento es correcto si y solo si el sistema a desarrollar lo satisface de manera total, no existe herramienta o procedimiento que asegure si es correcto o no, alternamente el cliente o usuario está en la capacidad de determinar si el requerimiento enunciado refleja las necesidades reales de la línea de negocio.
- **Sin ambigüedades:** Un requerimiento no es ambiguo cuando en su momento de declaración solo presenta una única interpretación, en caso que sea un término utilizado en un contexto particular se deberá incluir un glosario de términos donde se hará más específica su descripción.
- **Completo:** Un requerimiento solo está completo si cumple con los siguientes elementos:
 - Debe tener una relación con funcionalidad, rendimiento o diseño.
 - Se debe definir las respuestas de sistema a todos los tipos de entradas posibles, es necesario especificar que se deben definir las respuestas para datos de entrada validos como no validos.

- **Consistente:** Se refiere a la coherencia interna, si un requerimiento no está de acuerdo con algún documento de nivel superior como otro requerimiento, este no está correcto.

- **Rango por importancia o estabilidad:** Un requerimiento es clasificado por importancia o estabilidad si consta de un identificador único que lo diferencia de los demás requerimientos, por lo general todos los requerimientos se refieren a un producto de software y son igualmente importantes, algunos son estrictamente esenciales y otros solamente deseables.

- **Verificable:** Un requerimiento es verificable si y solo si existe algún proceso finito costo-efectividad que una máquina o una persona pueda validar que el producto software satisface esta necesidad, un requerimiento no es verificable si incluye:
 - Buen funcionamiento.
 - Buena interfaz humana.
 - Aprobado.

- **Modificable:** Un requerimiento es modificable si, y solo si, su estructura y estilo son tales que cualquier cambio que se le aplique se puede hacer fácilmente y de manera completa y coherente manteniendo su estructura. Un requerimiento es modificable si cumple con:
 - Tener una organización que sea coherente y fácil de usar.
 - No ser redundante.
 - Expresar cada requerimiento por separado en lugar de mezclarlos con otros requerimientos.

La redundancia en si no es un error, pero puede conducir fácilmente a errores pero a su vez permite en ocasiones que el documento sea más legible, el problema surge cuando se desean

hacer modificaciones en dos o más sitios donde aparezca un requerimiento, si se presenta el caso en que la redundancia sea necesaria se debe incluir las referencias cruzadas explícitas.

- **Trazable:** Un requerimiento es trazable si y solo si su origen es claro y se facilita la referencia de cada uno de ellos a través de las diferentes versiones del documento, la importancia de esta característica se hace visible cuando el desarrollo entra en la fase de pruebas y mantenimiento dado que cuando se altere un requerimiento es necesario saber el grupo de requerimientos asociados que se van a ver afectados por el cambio.

3.1.2 Estándares Técnicos

En esta área se procede a la definición de ciertos parámetros necesarios para dar comienzo a la fase de codificación, entre estos tenemos el lenguaje de programación y el conjunto de herramientas que se van a utilizar, junto con recomendaciones y convenciones que se deben seguir a la hora de codificar y estructurar el proyecto.

Lenguaje de programación

Para la correcta selección del lenguaje o lenguajes a utilizar para el desarrollo del producto software es necesario tomar en cuenta algunos aspectos, que se proceden a describir a continuación:

- **Dominio del lenguaje**

Identificar el nivel de conocimiento y desenvolvimiento en el lenguaje por parte del equipo de desarrollo, de presentarse un alto nivel en esta característica es posible que el tiempo de desarrollo tenga una duración menor que la prevista, por otro lado un nivel bajo podría provocar que el tiempo de desarrollo se prolongue más de lo esperado; se calificará el nivel con un valor numérico especificando el rango del mismo.
- **Estado de la comunidad**

Reconocer el nivel de actividad que presenta la comunidad, en gran parte de los casos durante el desarrollo se presentan dudas o

problemas que pueden ser solventados por la comunidad de programadores miembros; de esta manera es posible evitar un estancamiento del proyecto.

Codificación y estructura

Para la codificación y estructura se pretende establecer una convención para que el equipo desarrollador de software se comuniquen a través del código que realiza, facilitando así la comunicación técnica del equipo y facilitará el mantenimiento del código.

- **Base de Datos**

Se procede a enumerar las reglas de codificación que se aplicarán al esquema de base de datos, se especifica el motor de base de datos que se va a utilizar.

- **Código**

Se procede a enumerar las reglas de codificación provistas por el lenguaje junto con las recomendaciones brindadas por la comunidad o por elementos del equipo con un nivel superior de experiencia.

Herramientas adicionales

Se enumeran las herramientas que se van a utilizar en el desarrollo actual, se enunciarán características utilizadas de cada una, debe constar el nombre y versión como básicos junto a la información que sea relevante.

- **Estructura del sistema**

Se procede a brindar la estructura que tendrá la aplicación software a desarrollar, es posible que esta estructura se vea modificada por la utilización de herramientas o por integrantes del equipo con una mayor experiencia.

3.1.3 Diseño

Esta fase nos permitirá modular el sistema informático que brinde solución a las necesidades extraídas en la fase anterior, para construir la solución es necesario apoyarse en técnicas que permitan abstraer entidades

y relaciones que intervienen en los diferentes procesos físicos, este apartado se subdivide en dos diferentes áreas, estas son:

- Abstracción de datos.
- Prototipos de interfaz.

Abstracción de datos

El primer artefacto AP-MER nos apoyará en esta área permitiendo representar gráficamente y esquemáticamente la estructura de la base de datos que representa la línea de negocio que debemos sistematizar, que forma parte del diseño del sistema; el artefacto consta de 3 bloques fundamentales, estos son:

1. Bloque 1

Sistema: Registrará el nombre del sistema actual, mismo que opera bajo la estructura lógica empleada.

Fecha: Registra la fecha de autorización de uso del formato a aplicar.

Versión: Registrará numéricamente y en forma secuencial las modificaciones realizadas en el contenido.

2. Bloque 2

Diagrama Entidad – Relación: Se graficará la abstracción de la estructura a utilizarse en el área analizada considerando lo definido en el modelo conceptual de necesidad.

Descripción: Contiene aclaratorios definidos y puntos de vista que permitirán describir el diagrama de Entidad – Relación entregando propiedad en la información del grupo de diseñadores.

3. Bloque 3

Realizado Por: Aquí se registra el nombre del técnico en tecnología, su firma y fecha de presentación del documento como tal.

Autorizado Por: Se registrará nombre, función (experto o CEO del negocio), de la persona autorizada a validar y aprobar su interpretación y documentación con fecha correspondiente.

Formato: Nombre identificativo del formato.

El análisis de los bloques anteriores se ilustra en la Tabla 8.

Tabla 8. Análisis de Bloques AP-MER

Bloque 1		
Sistema	Fecha	Versión
Nombre del sistema	Uso formato	Modificación contenido
Bloque 2		
Diagrama E-R	Descripción	
Gráfico estructura	Describe diagrama E-R	
Bloque3		
Autor	Experto	Formato
Registra nombre, firma de técnico.	Registra nombre, función CEO de negocio	AP-MER2014

Fuente (Acosta Y. / Paredes J.)

En esta sección se utilizará técnicas de modelado de datos, una de ellas es el modelo entidad-relación [18] esta ayuda al analista con tres conceptos semánticos principales: entidades, relaciones y atributos. Quienes utilizamos el modelo E-R en el análisis de datos debemos describir la empresa en términos de:

- Entidades, que son objetos distintos en una empresa.
- Relaciones, que son interacciones significativas entre objetos.
- Atributos, que describen las entidades y relaciones.

Para usar el modelo E-R en el análisis de sistemas es probable que identifique primero los conjuntos de entidad, luego los conjuntos de

relación, y por último los atributos de cada conjunto, como se muestra en la figura 10.

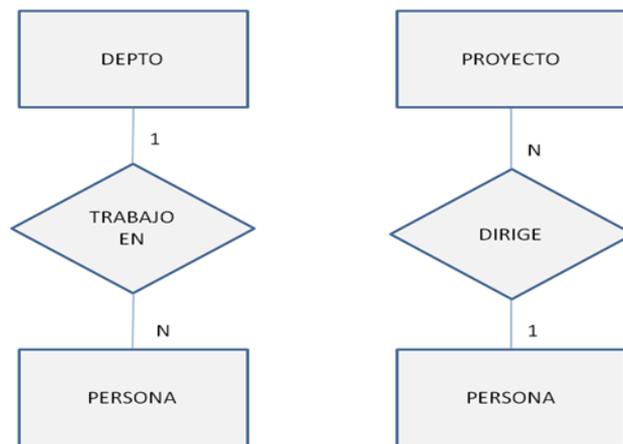


Figura 10. Modelo Entidad-Relación
Fuente (I. T. HAWRYSZKIEWYCZ/ 1994)

A continuación se plantea el formato para el seguimiento de versiones y modificaciones de documento:

Formato 2. Modelo Entidad-Relación

Sistema	Fecha	Versión
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato [##.##.##]
Diagrama Entidad-Relación		
Descripción		

Continúa →

Entidad_1: <ul style="list-style-type: none"> • Atributo_1: Valor representativo. Formato[x..... 300x] • Atributo_2: Valor representativo. Formato[x..... 300x] • Atributo_3: Valor representativo. Formato[x..... 300x] • Atributo_4: Valor representativo. Formato[x..... 300x] Relación: Descripción de la relación. Formato[x..... 100x] <ul style="list-style-type: none"> • Atributo_Relación: Valor representativo Formato[x..... 300x]. Entidad_2: <ul style="list-style-type: none"> • Atributo_5: Valor representativo. Formato[x..... 300x] • Atributo_6: Valor representativo. Formato[x..... 300x] • Atributo_7: Valor representativo. Formato[x..... 300x] 	
Realizado Por	Autorizado Por
Firma Nombre	Firma Nombre
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Fuente (Acosta Y. / Paredes J.)

Una vez obtenido el diagrama de entidad-relación se procede a normalizarlo para dar lugar a obtención del modelo relacional [19], el modelo relacional de datos es un modelo simple, potente y formal para representar la realidad. También ofrece una base firme para enfocar y analizar formalmente muchos problemas relacionados con la gestión de bases de datos, como el diseño incorrectamente aplicado, redundancia y distribución.

El elemento básico del modelo es la relación, y un esquema de base de datos relacional es una colección de definición de relaciones.

Un caso de relación es una tabla con filas y columnas. Las columnas de las relaciones corresponden a los atributos; las filas, denominadas tuplas, son colecciones de valores tomados de cada atributo, y desempeñan la misma función que los casos individuales de entidades en el modelo entidad relación. El grado de una relación es el número de columnas, cardinalidad de una relación es el número de tuplas.

El segundo artefacto es considerado como uno de los pilares fundamentales del desarrollo y por tanto un error presentado aquí se considera crítico, para poder sobrepasar esta sección de manera satisfactoria se propone el formato AP-MREL que consta de tres bloques y

representa un apoyo fundamental para contrarrestar posibles problemas, se procede a describir el contenido:

1. Bloque 1

Sistema: Registrará el nombre del sistema actual, mismo que opera bajo la estructura lógica empleada.

Fecha: Registra la fecha de autorización de uso del formato a utilizar.

Versión: Registrará numéricamente en forma secuencial las modificaciones realizadas en el contenido.

2. Bloque 2

Modelo Relacional: Se graficará el modelo relacional resultado de la normalización aplicada al formato AP-MER.

Descripción: Se divide en dos secciones relacionadas entre sí:

- **SQL:** Enumera las tablas y los campos junto a los atributos con sus respectivas propiedades, todo descrito en lenguaje SQL.
- **ORM:** Describe clases mapeadas junto con los atributos y sus propiedades en el lenguaje a desarrollar la solución.

3. Bloque 3

Realizado Por: Aquí se registra el nombre del técnico en tecnología, su firma y fecha de presentación del documento como tal.

Autorizado Por: Se registrará nombre del director de proyecto o división responsable junto con la fecha correspondiente.

Formato: Nombre identificativo del formato.

El análisis de los bloques anteriores se ilustra en la Tabla 9

Tabla 9. Análisis de Bloques AP-MREL

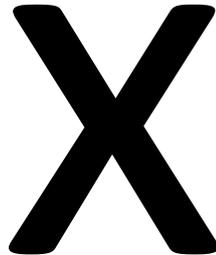
Bloque 1		
Sistema	Fecha	Versión
Nombre del sistema	Uso formato	Modificación contenido
Bloque 2		
Modelo relacional	Descripción	
Gráfico estructura aplicado formato AP-MER	SQL: enumera tablas, campos	ORM: describe clases y atributos
Bloque3		
Autor	Experto	Formato
Registra nombre, firma de técnico.	Registra nombre de director proyecto	AP-MERL2014

Fuente (Acosta Y. / Paredes J.)

Formato 3. Modelo Relacional

Sistema	Fecha	Versión
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato [##.##.##]
Modelo Relacional		

Continúa →



Descripción	
SQL	ORM
Nombre_de_Tabla: <ul style="list-style-type: none"> • Campo_1: Características • Campo_2: Características • Campo_3: Características • Campo_4: Características 	Nombre_de_Clase: <ul style="list-style-type: none"> • Campo_1: Propiedades • Campo_2: Propiedades • Campo_3: Propiedades • Campo_4: Propiedades
Realizado Por	Autorizado Por
Firma Nombre	Firma Nombre
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-MREL2014	

Fuente (Acosta Y. / Paredes J.)

Prototipos de Interfaz

Una vez superadas la abstracción de datos, es necesario que los grupos involucrados cliente y desarrolladores emitan propuestas específicas acerca

de los formatos de interfaz; centrándose en la aprobación por parte del usuario y en búsqueda de obtener la aprobación técnica.

Los wireframes representan un mapa aproximado de cada uno de los formatos de interfaz, que eventualmente será utilizado por los diseñadores gráficos para crear diseños preliminares y finales del sitio junto con los arquitectos de información que proveen los datos a mostrar. Los wireframes son aproximaciones bidimensionales donde los principales elementos de navegación y contenido del sistema tienen la posibilidad de aparecer en el, es recomendable mantener los diseños simples, sin adornos y evitando totalmente la utilización de tipografías distintivas, promueva la utilización de tipografías clásicas y genéricas; el uso de tonalidades grises es necesario para distinguir y definir aéreas funcionales, respecto a los gráficos e imágenes manejados es recomendable indicar únicamente su ubicación.

El tercer artefacto AP-MOCK nos permitirá documentar los prototipos de interfaz que trabajan como un aliado ágil de los formatos AP-MER y AP-MREL otorgando la unificación grafica de estos dos formatos proporcionando así expectativas claras para las partes involucradas, a continuación se procede a describir sus bloques de información.

1. Bloque 1

Sistema: Registrará el nombre del sistema actual, mismo que opera bajo la estructura lógica empleada.

Fecha: Registra la fecha de autorización de uso del formato a utilizar.

Versión: Registrará numéricamente en forma secuencial las modificaciones realizadas en el contenido

Id: Registra un código único y secuencial que permitirá identificar el wireframe actual.

2. Bloque 2

Id_anterior: Registra el código identificador del wireframe que invocó el prototipo de pantalla actual.

Wireframe: Se graficará el prototipo de interfaz o wireframe.

Acción: Se describe las posibles acciones que maneja el wireframe.

3. Bloque 3

Realizado por: Se registra el nombre del diseñador junto con su firma y fecha de presentación del documento.

Validado por: Registrará el nombre y firma de usuario responsable, y consta como aprobación por parte del mismo.

Autorizado por: Registrará el nombre y firma del actor encargado de verificar la aplicación de estándares en el prototipo.

Formato: Nombre identificativo del formato.

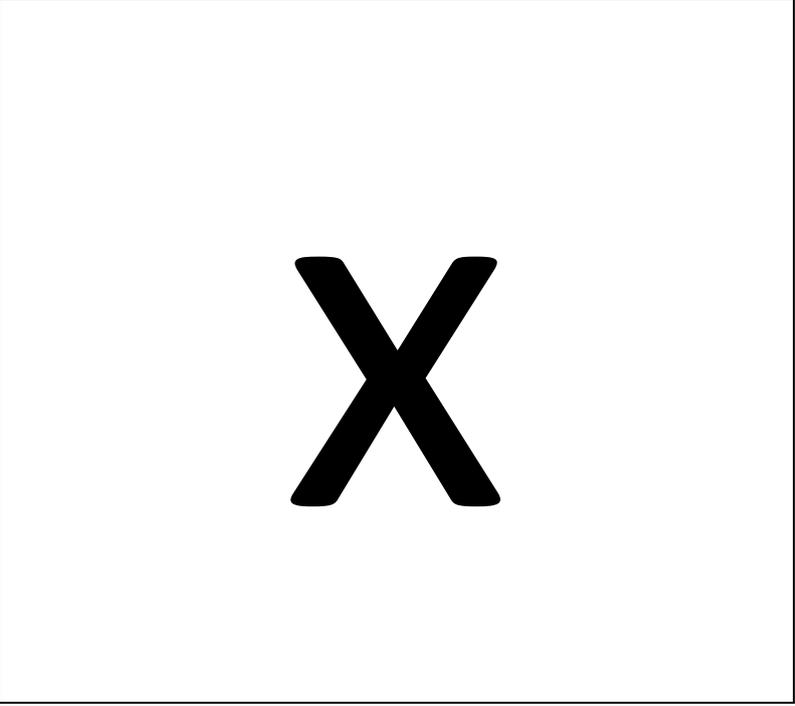
El análisis de los bloques anteriores se ilustra en la Tabla 10.

Tabla 10. Análisis de Bloques AP-MOCK

Bloque 1			
Sistema	Fecha	Versión	Id
Nombre del sistema	Uso formato	Modificación contenido	Código único
Bloque 2			
Id anterior	wireframe		
Registra código wireframe	Gráfica prototipo		
Bloque3			
Autor	Autentica	Experto	Formato
Registra nombre, diseñador, firma, fecha.	Registra nombre, firma usuario responsable	Registra nombre, firma actor.	AP-MOCK2014

Fuente (Acosta Y. / Paredes J.)

Formato 4. Mockup

Sistema	Fecha	Versión	Id
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato [#.#.#]	Identificador de Wireframe Formato [mck###]
Id_anterior	Wireframe		
Wireframe anterior Formato [mck###]			
	Acción		
	Formato [...80...]		
Realizado Por	Validado Por	Autorizado Por	
Firma Nombre	Firma Nombre	Firma Nombre	
Fecha	Fecha	Fecha	
Formato [##/##/####,DD/MM/YYYY]	Formato [##/##/####,DD/MM/YYYY]	Formato [##/##/####,DD/MM/YYYY]	
Formato: AP-MOCK2014			

Fuente (Acosta Y. / Paredes J.)

3.1.4 Construcción y pruebas

En esta fase se procede a la construcción del producto software definiendo ciertos parámetros como lenguaje y herramientas entre otros y las respectivas pruebas aplicadas.

- Construcción
- Pruebas

Construcción

En esta fase se procederá a explicar el método ya sea manual o utilizando un sistema de control de versiones que se va a utilizar para mantener el código del proyecto actualizado, versionado y estructurado.

Pruebas

Pruebas unitarias

Una prueba unitaria es probar una porción de código en sus diferentes escenarios para saber si funciona, detectar las fallas de esa porción de código [20].

Gestión de Configuración

Esta área nos permitirá registrar un cambio encontrado en cualquier etapa del desarrollo de la solución informática, lo que nos lleva a enunciar que los formatos utilizados aquí se constituyen como un apoyo a la documentación generada durante todas y cada una de las fases con sus respectivos artefactos, a continuación se enuncian las etapas de la fase:

- Identificar
- Aprobar
- Ejecutar

El primer artefacto AP-DETER nos va a permitir registrar e ingresar a la documentación la detección y descripción de un error; el formato consta de tres bloques, estos son:

1. Bloque 1

Sistema: Registrará el nombre del sistema en el que se está trabajando, mismo que opera bajo la estructura lógica empleada.

Fecha: Es la fecha de autorización de uso del formato a emplearse.

Versión: Registrará en forma secuencial las modificaciones realizadas el contenido del mismo sin descuidar la integridad de la información.

2. Bloque2

Información del Error: Se divide en dos secciones que permiten situar el error.

- **Nombre Formato:** Aquí se registrará el nombre del formato donde se hizo visible el error.
- **Versión:** Registrará la versión del formato donde se identifico el error.

Descripción del Error: Registrará la descripción detallada del error encontrado, de ser posible se enuncia una posible solución al mismo.

Grafico: En caso de referirse a un formato que contenga descripciones basadas en gráficos o diagramas se procede a graficar únicamente la zona donde se encuentra el error.

3. Bloque 3

Realizado por: Se registra el nombre del actor que reconoce el error junto con su firma y fecha de presentación del documento.

Revisado por: Miembro encargado del equipo de diseño/desarrollo que verifica la existencia del error.

Formato: Nombre identificativo del formato.

El análisis de los bloques anteriores se ilustra en la Tabla 11.

Tabla 11. Análisis de Bloques AP-DETER

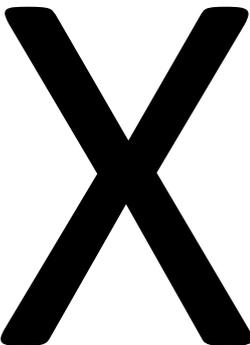
Bloque 1			
Sistema	Fecha	Versión	
Nombre del sistema	Uso formato	Modificación contenido	
Bloque 2			
Información del error		Descripción del error	Gráfico
Nombre formato: registra donde se visualizo el error	Versión formato de error	Destalle de error encontrado	Graficar zona de error
Bloque3			
Autor	Experto	Formato	
Registra nombre, firma de técnico.	Verifica existencia de error	AP-DETER	

Fuente (Acosta Y. / Paredes J.)

Formato 5. Detección de Errores

Sistema:	Fecha:	Versión:
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato[#.#.#]
Información del Error		
Formato: AP - [x..... 8x] Versión: Formato[#.#]		
Descripción del Error		
Descripción detallada del error Formato [x..... 800x]		
Gráfico		

Continúa →

	
Realizado Por	Revisado Por
Nombre Firma	Nombre Firma
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-DETER	

Fuente (Acosta Y. / Paredes J.)

El segundo artefacto AP-AUCH se centra en la autorización por parte del manager de proyecto de si se debe o no efectuar el cambio identificado en el área anterior, haciendo referencia al formato anterior AP-DETER, se procede a describir el formato.

1. Bloque 1

Sistema: Registrará el nombre del sistema en el que se está trabajando, mismo que opera bajo la estructura lógica empleada.

Fecha: Es la fecha de autorización de uso del formato a emplearse.

Versión: Registrará en forma secuencial las modificaciones realizadas el contenido del mismo sin descuidar la integridad de la información.

2. Bloque 2

Referencia: Aquí se hace referencia al documento que solicita aprobación y se divide en:

- **Formato:** Registrará el nombre del formato que presenta el posible cambio.
- **Versión:** Registrará la versión del formato que presenta el posible cambio.

Calificación: Aquí se registra el estado del presente formato, junto con una descripción de las razones de dicho estado.

- **Estado:** Registrará dos valores, aprobado o denegado.
- **Justificación:** Aquí se registrará un breve argumento de las razones que se tomaron en consideración para determinar el estado del presente documento.

3. Bloque 3

Autorizado por: Se registrará nombre del director de proyecto o división responsable.

Formato: Nombre identificativo del formato.

El análisis de los bloques anteriores se ilustra en la Tabla 12.

Continúa →

Tabla 12. Análisis de Bloques AP-AUCH

Bloque 1			
Sistema	Fecha	Versión	
Nombre del sistema	Uso formato	Modificación contenido	
Bloque 2			
Referencia		Calificación	
Formato: registra donde habrá posible cambio	Versión: formato de posible cambio	Estado: aprobado o denegado	Justificación: razones de estado de documento
Bloque3			
Experto		Formato	
Registra nombre director de proyecto o responsable		AP-AUCH	

Fuente (Acosta Y. / Paredes J.)

Formato 6. Autorización de Cambio

Sistema:	Fecha:	Versión:
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato[#.#.#]
Referencia		
Formato: AP - DETER Versión: Formato[#.#.#]		
Calificación		
Estado: Formato[Aprobado / Denegado] Exposición: Formato[x..... 800x]		
Autorizado Por		
Nombre Firma		
Fecha		
Formato[##/##/####,DD/MM/YYYY]		
Formato: AP-AUCH		

Fuente (Acosta Y. / Paredes J.)

El tercer formato AP-SOL representa el formato que brindará una solución al error encontrado y aprobado anteriormente, consta de:

1. Bloque 1

Sistema: Registrará el nombre del sistema en el que se está trabajando, mismo que opera bajo la estructura lógica empleada.

Fecha: Es la fecha de autorización de uso del formato a emplearse.

Versión: Registrará en forma secuencial las modificaciones realizadas el contenido del mismo sin descuidar la integridad de la información.

2. Bloque 2

Referencia: Aquí se hace referencia al documento que autoriza el cambio a efectuar.

- **Formato:** Nombre del formato que concede la autorización de cambio.
- **Versión:** La versión del documento que concede la autorización de cambio.

Gráfico: De ser el caso que el formato donde se reporto el error contenga información de manera gráfica se registrara aquí la solución gráfica de la misma manera, de no ser el caso se dejara en blanco.

Solución: Registrará la descripción de la solución de una manera detallada y concisa.

3. Bloque 3

Realizado Por: Aquí se registra el nombre del técnico en tecnología, su firma y fecha de presentación del documento como tal.

Autorizado Por: Se registrará nombre del director de proyecto o división responsable.

Formato: Nombre identificativo del formato.

El análisis de los bloques anteriores se ilustra en la Tabla 13

Continúa →

Tabla 13. Análisis de Bloques AP-SOL

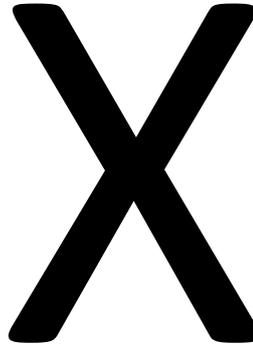
Bloque 1			
Sistema	Fecha	Versión	
Nombre del sistema	Uso formato	Modificación contenido	
Bloque 2			
Referencia		Gráfico	Solución
Formato: concede autorización de cambio	Versión: concede autorización de cambio	Registra la solución	Registra solución detallada y concisa
Bloque3			
Autor	Experto	Formato	
Registra nombre, firma de técnico.	Registra nombre director proyecto	AP-SOL	

Fuente (Acosta Y. / Paredes J.)

Formato 7. Solución

Sistema	Fecha	Versión
Nombre del sistema Formato [x..... 20x]	Fecha de emisión de documento. Formato [18/09/2014]	Versión del documento. Formato[#.#.#]
Referencia		
Formato: AP - DETER Versión: Formato[#.#.#]		
Gráfico		

Continúa →


Solución

 Descripción de la solución propuesta
 Formato[x..... 800x]

Realizado Por	Aprobado Por
Nombre Firma	Nombre Firma
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-SOL	

Fuente (Acosta Y. / Paredes J.)

3.2. Aplicación de la metodología

3.2.1 Modelo de Conceptual de Necesidad

Gestión de Productos

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.1
Descripción General:		
<p>En la presente sección se tiene el control de flujo de los productos que almacena y distribuye la dependencia; en su gran mayoría los productos mencionados presentan similitudes en su utilización, estas similitudes facilitan su referencia por <i>Cuentas</i>, a pesar de contar con un producto aislado este debe pertenecer a una única cuenta en particular para su correcto registro.</p> <p>Son evidentes los problemas de registro en el flujo de productos, el incorrecto manejo de información ofrece una gran facilidad de fallo al momento de informar al sistema algún cambio físico en la bodega, ya sea por ingreso o por salida de algún producto en particular, el encargado debe registrar el nuevo cambio de dos a tres veces en diferentes formatos, lo que propensa el error aún mas, como se mencionó anteriormente los productos son clasificados por cuentas, y el usuario también debe informar a este apartado el cambio realizado; existe otro apartado de reportes en donde también se debe informar del cambio, es decir éste registro genera diversas salidas.</p>		
Funcionalidad		
<p>Producto: El control de productos se encuentra relacionado con cuentas y con el nivel de disponibilidad, las cuentas son quienes clasifican dicho producto y permiten su correcta administración, en lo referente, la disponibilidad es necesario mantener un pleno conocimiento del estado de la bodega en cualquier momento que se requiera.</p> <p>El sistema ofrecerá un control estricto en los ámbitos de disponibilidad y administración de productos, es decir el usuario tendrá un registro y reporte casi inmediato al momento de requerir la información de un producto en particular y de igual manera el Dpto. de logística contará con esta información de todos los productos en stock lo que permitirá la toma de decisión a su nivel.</p> <p>El producto software en diseño presenta una interface amigable y fácil de utilizar en la administración de los productos.</p>		
Restricciones		Continúa →

<ul style="list-style-type: none"> • Personas no autorizadas • No se permitirá productos con fechas próximas a caducar. 	
Gráfico	
<p>Encargado de Bodega</p> <pre> graph LR Actor1(()) -- Hoja Requisición --> V[Verifica producto] V --> G[Genera Cuenta] G -- Reporte --> R[Recibe Producto] R -- Entrega Documento --> K[Genera Kardex] K -- Legaliza --> Actor2((USUARIO SOLICITANTE)) </pre>	
Realizado Por	Autorizado Por
Juan Francisco Paredes Andino	Ymir Camilo Acosta Rodríguez
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MCN	

Cuentas

Req(01) Para administrar los productos se generarán cuentas, las cuales deberán tener un código, un nombre y una descripción.

Req(02) La búsqueda de cuentas se la realizará según el código de partida de la cuenta registrada en el clasificador **Anexo C**, o por identificador de base de datos.

Productos

Req(03) Los productos contarán con un código, un nombre y una descripción para la administración y el manejo adecuado.

Req(04) Se mantendrá actualizada la lista de productos cada vez que se entreguen a los usuarios sus pedidos.

Req(05) La búsqueda se la realizará según el nombre del producto o código.

Gestión de Ingresos (Productos)

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.2
Descripción General:		
<p>Se maneja el control de ingresos de los productos que se mantienen almacenados en la bodega del Departamento de Logística de la EPAE; en general los ingresos se los realiza de manera manual (física), verificando las características del producto (N° de serie, especificaciones técnicas) según requerimiento de los usuarios, lo cual genera inconvenientes ocasionando que ocurra pérdidas de documentos. Posteriormente se ubica los productos en sus estantes respectivos dentro de la bodega y registrando en el kardex su ingreso.</p> <p>Si no se considera lo anteriormente mencionado, se presentan dificultades en el momento de la constatación física y manejo de inventario al momento de obtener un reporte final, el incorrecto manejo de esta información genera un fallo al momento del ingreso físico de cualquier producto a la bodega.</p>		
Funcionalidad		
<p>Ingresos (productos): El control de ingreso de productos se encuentra ligado con ingresos y proveedores, los ingresos son los que incorporan una serie de campos relacionados con el producto correspondiente y que permiten llevar a cabo un proceso referido.</p>		
Restricciones		
<ul style="list-style-type: none"> No se permitirán ingresar productos que no cumplan con las especificaciones técnicas requeridas por el usuario. No se ingresarán productos que no pertenezcan a partida alguna. No se registrará ingresos de productos sin el visto bueno de ingreso. 		
Gráfico: (Productos)		
<p>Encargado de Bodega</p> <pre> graph LR User1((Encargado de Bodega)) --> Step1[Ingresar Productos] Step1 -- Manejo Interfaz --> Step2[Verificar estado productos] Step2 -- Especific. Técnicas --> Step3[Ubicar productos] Step3 -- Ubicación en estantes --> Step4[Registrar productos] Step4 -- Hoja Kardex --> User2((USUARIO SOLICITANTE)) Step2 -- No cumple Espec. Técnicas --> Step5[No legalización] </pre>		
Realizado Por	Autorizado Por	
Juan Francisco Paredes Andino	Ymir Camilo Acosta Rodríguez	
Fecha	Fecha	
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	
Formato: AP-MCN		

Productos

Req(06) Se ingresarán los productos con un código, fecha, tipo, proveedor y el listado de productos.

Req(07) Las transacciones serán registradas al día.

Req(08) Se ingresará la lista de productos con un código, fecha, un tipo, donación al igual que un proveedor denominado donación tomando en cuenta que existen un tipo de ingreso estandar (adquisición) y donación.

Gestión de Ingresos (Proveedores)

Sistema	Fecha	Versión
Synaps	[###/###/####,DD/MM/YYYY]	0.0.3
Descripción General:		
<p>El control de ingresos de proveedores que se encuentran registrados en el Departamento de Logística de la EPAE; se ingresan manualmente llevando el seguimiento en las hojas Excel, verificando características como (Ruc, Rup, contacto) siendo información utilizada en cada compra, manteniendo fidelidad y confianza del banco de proveedores, hoy en día las compras son públicas y el banco de proveedores son calificados por el ministerio correspondiente, manteniendo nuestro banco por situaciones de proceso e inconformidades.</p>		
Funcionalidad		
<p>Ingresos (proveedores): Los proveedores son personas naturales y jurídicas autorizadas a la comercialización de productos con la empresa pública constituyéndose un campo referencial en la administración de ingreso de productos.</p> <p>Este sistema brindará un control completo en este ámbito es decir, el usuario tendrá un reporte de proveedores al momento de requerirlo y de igual forma tendrá una presentación amigable.</p>		
Restricciones		
<ul style="list-style-type: none"> • Si no poseen Rup no pueden ser proveedores. • No se puede almacenar un proveedor perpetuamente. 		
Gráfico: (Proveedores)		
<p>Encargado de Bodega</p> <pre> graph LR Encargado[Encargado de Bodega] -- Verifica (Revisión de datos) --> Ingresa[Ingresa datos (Manejo Interfaz)] Ingresa -- Autentica (Aprobación de personal) --> Usuario[USUARIO SOLICITANTE] Usuario -- Continúa --> Continua[Continúa] </pre>		

Realizado Por	Autorizado Por
Juan Francisco Paredes Andino	Ymir Camilo Acosta Rodríguez
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-MCN	

Proveedores

Req(09) El ingreso de proveedores tendrá un Ruc, Rup, razón social, ciudad, dirección, y teléfono.

Req(10) La búsqueda de proveedores se realizará mediante la razón social, ruc o rup.

Continúa →

Gestión de Egresos (Productos)

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.4
Descripción General:		
<p>Una vez autorizada la entrega del producto se procede a realizar la constatación de existencias, registrando junto al ítem la no existencia y procede a entregar lo que se posee.</p> <p>Esta actividad aporta con registros innecesarios, posee riesgo de traspapelar, doble entrega, o de ser el caso entrega sin justificación por pérdida de documento.</p> <p>Evidenciando de esta manera problemas al momento del egreso de productos, el incorrecto manejo de información de datos ofrece tener un porcentaje alto de fallo al momento de informar al sistema la salida de productos de bodega, el encargado debe retomar el conteo y verificar de nuevo el producto, lo que genera pérdida de tiempo, esfuerzo e información.</p>		
Funcionalidad		
<p>Egreso (Productos): El entorno de egresos de productos se lo procesa con un módulo que ofertamos, que el usuario del mismo poseerá datos reales al instante de realizado el egreso.</p> <p>La información registrada permitirá obtener datos consolidados de todos los productos almacenados, se mantendrá actualizado el listado de productos según las salidas presentadas.</p>		
Restricciones		
<ul style="list-style-type: none"> No se podrá descargar productos sin la autorización respectiva. Solo se entregarán productos que existan en bodega. No se autoriza la entrega a personal no involucrado en el proceso, perteneciente a la EPAE. 		
Gráfico: Egresos(Productos)		
<pre> graph LR U1[Usuario autorizado] -- Realiza Oficio de Pedido --> A1[Oficios, Memorandos] A1 -- Control egreso --> A2[Hoja Kardex] A2 -- Verifica Existencias --> A3[Ubicación en bodega, estantes.] A3 -- Firma Recibido --> A4[Entrega] U2[Encargado de Bodega] -- Firma Recibido --> A4 U3[USUARIO SOLICITANTE] -- Entrega --> A4 </pre>		
Realizado Por	Autorizado Por	
Juan Francisco Paredes Andino	Ymir Camilo Acosta Rodríguez	
Fecha	Fecha	
Formato[##/##/####,DD/MM/YYY]	Formato[##/##/####,DD/MM/YYYY]	
Formato: AP-MCN		

Egresos (Productos)

Req(11) Se registrará fecha, persona que recibe el producto, número de oficio/memorando y la lista de productos, para que puedan ser entregados al usuario cuando se requiera.

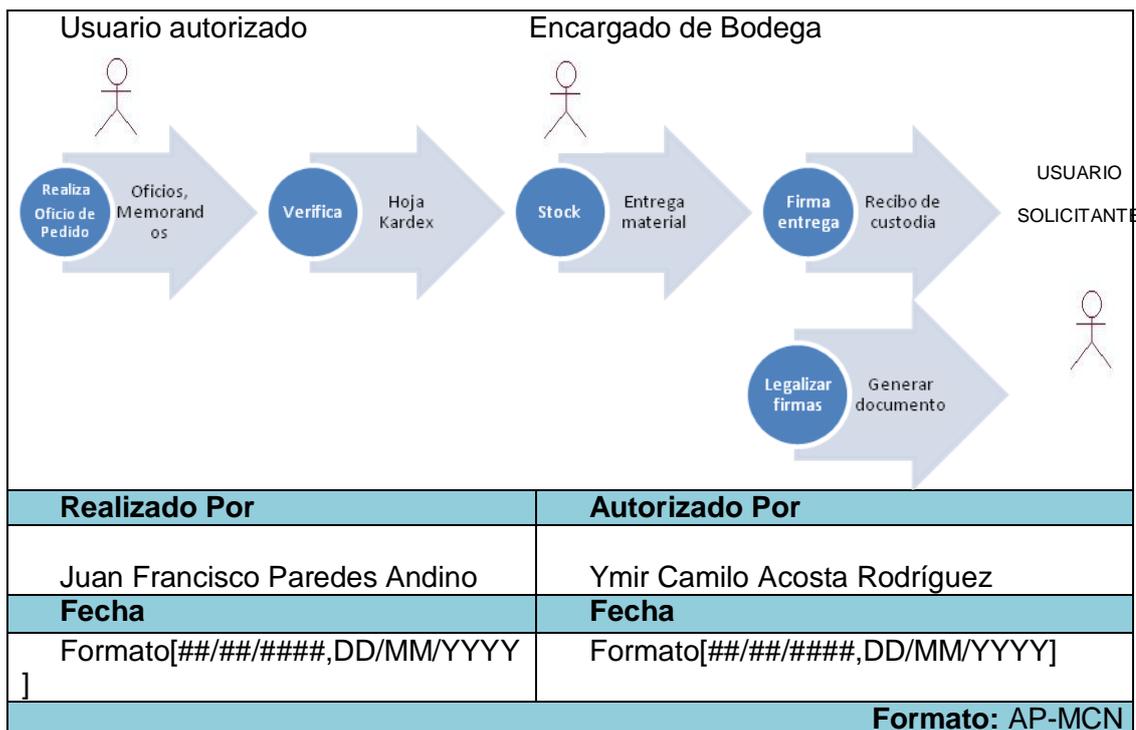
Req(12) Se descontará del numérico de productos mientras consten registrados y una vez entregado al usuario.

Req(13) Mantendremos consultas de egreso mediante código, fecha o persona encargada.

Gestión de Egresos (Préstamos)

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.5
Descripción General:		
<p>El control y distribución de producto designados a la EPAE, se la realiza mediante autorización del departamento que necesita de acuerdo a un oficio o memorando según corresponda, se maneja un documento (Recibo de custodia) para descargo manteniendo firmas de responsables (autorizado, entrega, recibido) registrando el préstamo en la hoja kardex (archivo) verificando las existencias en bodega, al tener un número elevado de préstamos por la cantidad de personal de los cursos de perfeccionamiento, el registro se lo hace general o en grupo, de tal manera que al no mantener un control individualizado, ocasione redundancia en la entrega de material, generando problemas al instante del préstamo; el incorrecto manejo de información de datos genera un alto porcentaje de fallo al momento de informar al sistema la salida de dicho producto; el encargado debe retomar el conteo y verificar que coincida el numero de existencias, lo que crea pérdida de tiempo en el trabajo y pérdidas económicas.</p>		
Funcionalidad		
<p>Egreso (Préstamos): La administración de egresos la relacionamos con un módulo de de préstamos, quienes especifican los materiales que a la vez permiten una adecuada gestión, manteniendo el control estricto y exacto de lo que se presta según sea el caso.</p> <p>El usuario tendrá un registro exacto de lo que da en calidad de préstamo teniendo en cuenta a la persona responsable de la devolución; a su vez, el sistema será de fácil navegabilidad para facilitar la administración evitando repetir varias veces el proceso.</p>		
Restricciones		
<ul style="list-style-type: none"> • No se prestará material sin autorización respectiva (recibo custodia). • Solo se prestará material que exista en bodega. • No se autoriza el préstamo a personal no involucrado en el proceso, perteneciente a la EPAE. 		
Gráfico: Egresos(Préstamos)		

Continúa →



Egresos (Préstamos)

Req(14) Se registrarán préstamos teniendo una fecha de emisión, fecha de entrega-recepción, número de oficio/memorando y persona responsable de la devolución y la lista de material.

Req(15) Se verificará el material que ha sido prestado por la EPAE al usuario.

Req(16) Verificaremos si el material que se dió en préstamo fué devuelto o no.

Req(17) Las transacciones en calidad de préstamo y devolución se mantendrán registradas al día.

3.2.2 Estándares Técnicos.

Lenguaje de programación

Para la selección del lenguaje de programación se toman en cuenta los aspectos de dominio del lenguaje y estado de la comunidad, resultado de estos análisis se determina que el lenguaje a utilizar para el presente desarrollo es Python y se procede a justificar esta elección.

- **Dominio del lenguaje**

En referencia al dominio del programa evaluamos los actores de este trabajo teórico-práctico considerando que son ellos los encargados de planificar, diseñar, construir y liberar el producto, en la institución que será la encargada de explotarlo coadyuvando al desarrollo de los logros planteados para alcanzar la misión de la EPAE.

El equipo presenta un nivel de conocimiento junior en el lenguaje es decir de 5/10, se llega a esta calificación dado que los integrantes del presente proyecto poseen conocimientos medios en manejo de instrucciones simples, condicionales e iterativas; conocimiento medio de clases y objetos junto con conocimientos básicos en el sistema de paquetes y referencia de librerías.

La lógica de programación presentada por el equipo técnico es acorde con el nivel que se requiere para familiarizarse con el lenguaje elegido, python es un lenguaje de programación que opera bajo la licencia [21] declarada como software libre.

- **Estado de la comunidad**

En la página oficial de Python encontramos un apartado destinado para la comunidad [22] que cuenta con diferentes canales de ayuda como FAQs, conferencias, tutoriales por semana y un chat IRC con los canales #python para usuarios, #python-dev para desarrolladores y #distutils para discusiones acerca de los paquetes; en Latinoamérica existen también movimientos similares como en Argentina con PyAr que se puede contactar en su página oficial [23], o la que presenta más actividad Python Perú que se puede contactar en su página oficial [24], otra referencia importante acerca de la elección del lenguaje a utilizar es el número de repositorios de desarrolladores en la red social GitHub, como se ilustra en la Tabla 14. Clasificación de repositorios [25].

Tabla 14 Clasificación de repositorios

LENGUAJES	NÚMERO DE REPOSITORIOS
JavaScript	160.644
Python	89,528
Ruby	88,276
Java	73,921
Php	70,755
C	42,005
C++	33,160
Objective-C	31,308
Shell	22,370
C#	20,944

Codificación y escritura

- **Base de datos**

En el área de persistencia de información se procede a utilizar el motor de base de datos MySQL, concretamente porque el equipo de desarrollo tiene un alto nivel de familiarización con el manejo de la herramienta y sus diferentes interfaces, ya sean gráficas o de consola.

- Utilizar nombres de máximo 64 caracteres, restricción de MySQL.
- Utilizar underscore (_) para separar palabras en nombres.
- MySQL es case sensitive en los nombre de las bases de datos y las tablas, no siendo así en su estructura, por lo que se recomienda utilizar todo en minúsculas.
- El nombre de la variable deberá guardar relación con el contenido que va a manejar.

- Luego de cada identificador de tabla, se deberán poner las claves foráneas también llamadas de referencia.

- **Código**

El lenguaje seleccionado para el desarrollo posee una guía de estilo denominada PEP-8 en donde constan las referencias y guías ejemplificadas de los parámetros a seguir a la hora de estandarizar el código generado.

- Documento oficial donde se registran las recomendaciones [26].

Herramientas adicionales

Para permitir un desarrollo ágil del producto software que se pretende generar, se va a utilizar una herramienta denominada "Django" que es un framework de desarrollo escrito en python, entre sus principales características tenemos:

- Aplica la filosofía DRY que fomenta la máxima reutilización de código que sea posible.
- Django Admin, este framework desde su versión 1.6 viene con un sistema de administración de información activo y listo para utilizar.
- Mapeo objeto-relacional es una librería de django que simplifica las tareas básicas de acceso a datos, mediante el mapeo de la base de datos.

- **Estructura del sistema**

Al plantear la utilización del framework django nos regimos por la estructura que nos establece, tanto de carpetas como de ficheros; al crear un proyecto el framework nos provee de una estructura como se ilustra en la Figura 12. Estructura de un proyecto Django [27].

```
mysite/  
  manage.py  
  mysite/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```

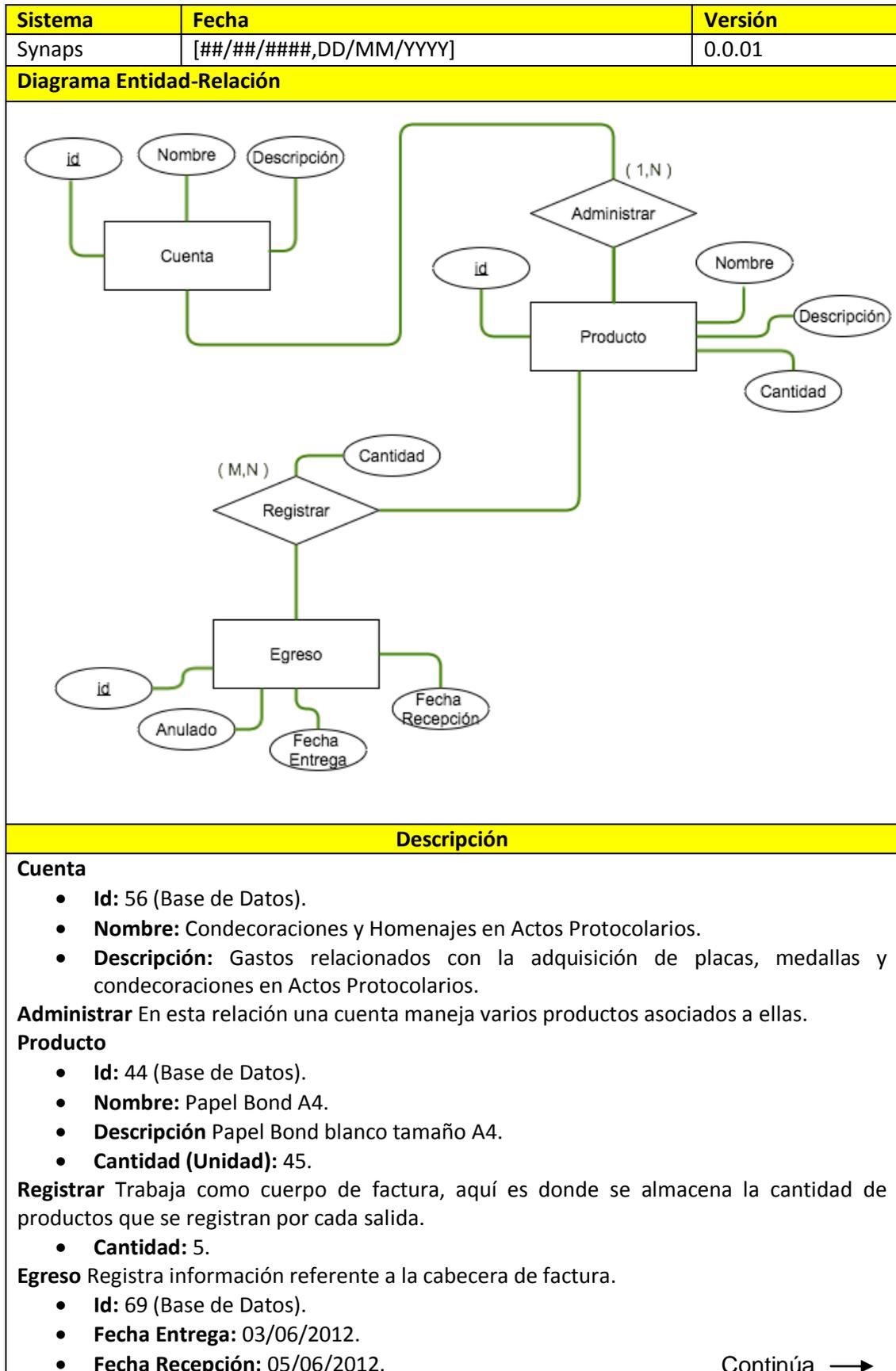
Figura 11. Estructura de un proyecto Django.

En donde:

- La primera línea `mysite/` es la carpeta raíz del proyecto.
- `Manage.py` es una utilidad de línea de comandos que nos permite interactuar con las herramientas administrativas disponibles.
- `Mysite/` es el directorio que trabaja como paquete en python.
- `__init__.py`: Es un fichero python vacío que indica al lenguaje que este directorio trabaja como un paquete.
- `Setting.py`: Fichero python que contiene todas las configuraciones que indican cómo está estructurado el proyecto junto con las librerías y aplicaciones internas utilizadas.
- `Urls.py`: Fichero python que contiene todas las direcciones internas que pueden ser llamadas desde el cliente.
- `Wsgi.py`: fichero python que contiene la configuración para que la aplicación funcione desde un servidor WSGI.

Continúa →

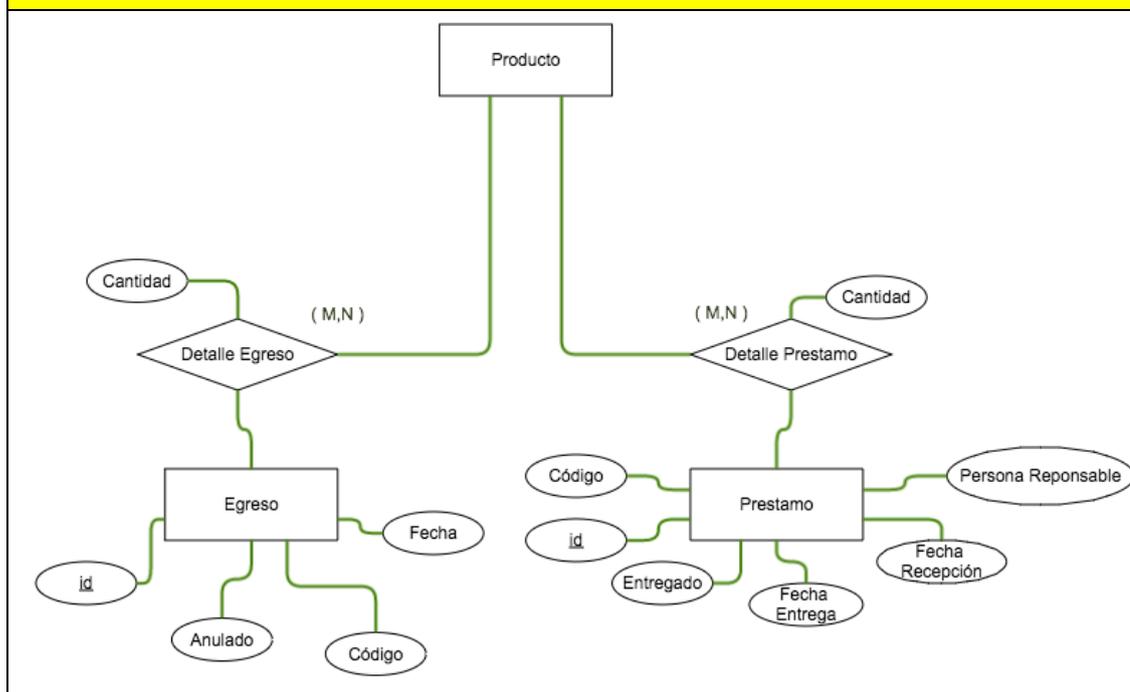
3.2.3 Diseño



<ul style="list-style-type: none"> • Anulado: Si/No. 	
Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
Formato[##/##/####,DD/MM/YYYY]	Formato[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.02

Diagrama Entidad-Relación



Descripción

Cuenta: Versión 0.0.01.

Administra: Versión 0.0.01.

Producto: Versión 0.0.01.

Registrar: 0.0.01 funcionalidad reemplazada por **Detalle_Egreso** y **Detalle_Prestamo**.

Detalle_Egreso: Funciona como cuerpo de factura y registra los productos junto con la cantidad de los mismos que egresan por cada salida.

- **Cantidad:** 5.

Egreso: Trabaja como cabecera y registra:

- **Id:** 87. (Base de Datos).
- **Código:** 2014006.
- **Anulado:** No/Si.
- **Fecha:** 03/06/2012.

Detalle_Prestamo: Funciona como cuerpo de factura y relaciona los productos con la cantidad de los mismos que son prestados por la dependencia.

- **Cantidad:** 4.

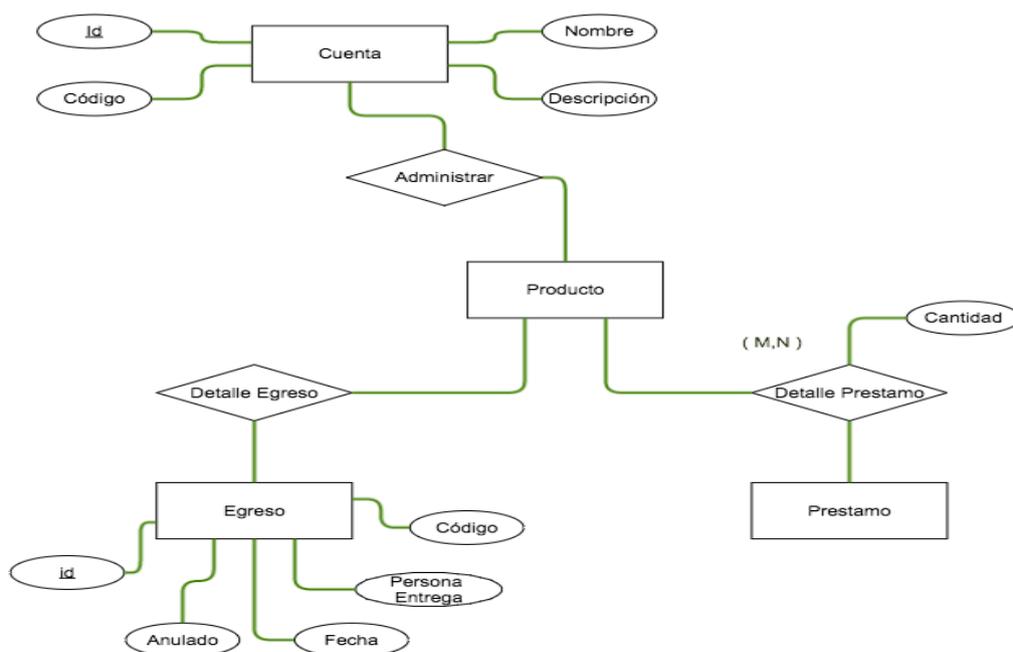
Préstamo: Trabaja como cabecera y registra:

Continúa →

<ul style="list-style-type: none"> • Id: 3 (Base de Datos). • Código: 2014006. • Entregado: Si/No. • Fecha Entrega: 06/08/2012. • Fecha Recepción: 06/09/2012. • Persona Responsable: Henry Beltrán. 	
Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.03

Diagrama Entidad-Relación



Descripción

Cuenta:

- **Id:** 56.
- **Código:** 530822(Clasificador Presupuestario de Ingresos y Gastos del Sector Público).
- **Nombre:** Condecoraciones y Homenajes en Actos Protocolarios.
- **Descripción:** Gastos relacionados con la adquisición de placas, medallas y condecoraciones en Actos Protocolarios.

Administrar: Versión 0.0.01.

Producto: Versión 0.0.01.

Detalle_Egreso: Versión 0.0.02.

Egreso:

Continúa →

- **Id:** 87. (Base de Datos).
- **Código:** 2014006.
- **Anulado:** No/Si.
- **Fecha:** 03/06/2012.
- **Persona Encargada:** Henry Beltrán.

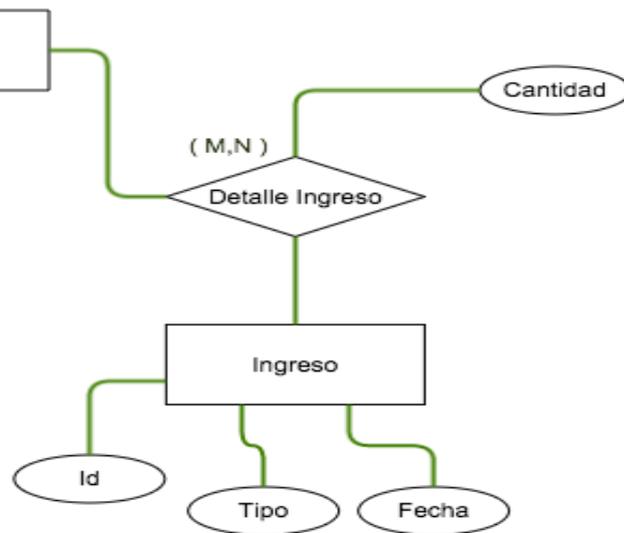
Detalle_Prestamo: Versión 0.0.02.

Préstamo: Versión 0.0.02.

Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.04

Diagrama Entidad-Relación



Continúa →

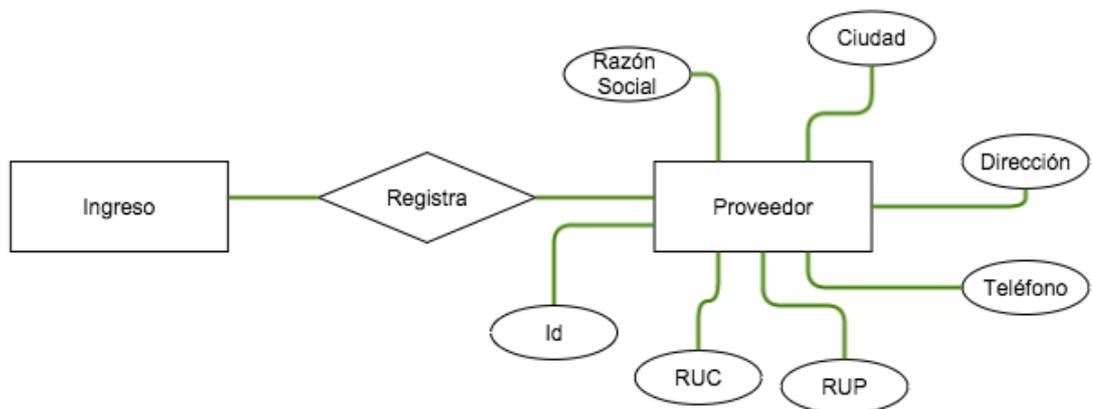
Descripción	
<p>Cuenta: Versión 0.0.03. Administrar: Versión 0.0.01. Producto: Versión 0.0.01. Detalle Ingreso: Funciona como cuerpo de factura y relaciona el ingreso con los respectivos productos que obtiene la dependencia.</p> <ul style="list-style-type: none"> • Cantidad: 4. <p>Ingreso: Trabaja como cabecera de cada ingreso y registra:</p> <ul style="list-style-type: none"> • Id: 50 (Base de Datos). • Fecha: 06/08/2012. • Tipo: Regular / Donación. <p>Detalle_Egreso: Versión 0.0.02. Egreso: Versión 0.0.03. Detalle_Prestamo: Versión 0.0.02. Préstamo: Versión 0.0.02.</p>	
Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.05
Diagrama Entidad-Relación		
<pre> erDiagram Producto --} Ingreso : "Detalle Ingreso (M,N)" Ingreso --} Proveedor : "Registra" Ingreso { string Id PK string Tipo string Fecha string Código } Proveedor { string Id PK string Razón Social string Ciudad string Dirección string Teléfono } </pre>		
Descripción		
<p>Cuenta: Versión 0.0.03. Administrar: Versión 0.0.01. Producto: Versión 0.0.01. Detalle Ingreso: Versión 0.0.04. Ingreso: Trabaja como cabecera de cada ingreso y registra:</p> <ul style="list-style-type: none"> • Id: 50 (Base de Datos). • Fecha: 06/08/2012. 		
		Continúa →

<ul style="list-style-type: none"> • Tipo: Regular/Donación • Código: 0000023466874328982734423. <p>Registra: Relación entre ingreso y producto.</p> <p>Proveedor:</p> <ul style="list-style-type: none"> • Id: 12 (Base de Datos). • Razón Social: Imprenta Carrillo. • Ciudad: Latacunga. • Dirección: Quito S/N entre Guayaquil y Padre Salcedo. • Teléfono: 2800481. <p>Detalle Ingreso: Versión 0.0.04.</p> <p>Detalle_Egreso: Versión 0.0.02.</p> <p>Egreso: Versión 0.0.03.</p> <p>Detalle_Prestamo: Versión 0.0.02.</p> <p>Préstamo: Versión 0.0.02.</p>	
Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión
Synaps	[##/##/####,DD/MM/YYYY]	0.0.06

Diagrama Entidad-Relación



Descripción

Cuenta: Versión 0.0.03.

Administrar: Versión 0.0.01.

Producto: Versión 0.0.01.

Detalle Ingreso: Versión 0.0.04.

Ingreso: Versión 0.0.05.

Registra: Versión 0.0.05.

Proveedor:

- **Id:** 12 (Base de Datos).
- **Razón Social:** Imprenta Carrillo.

Continúa →

- **Ciudad:** Latacunga.
- **Dirección:** Quito S/N entre Guayaquil y Padre Salcedo.
- **Teléfono:** 2800481.
- **RUP:** 43884756293884637192883372938371.
- **RUC:** 1723497754001.

Detalle_Egreso: Versión 0.0.02.

Egreso: Versión 0.0.03.

Detalle_Prestamo: Versión 0.0.02.

Préstamo: Versión 0.0.02.

Realizado Por	Autorizado Por
Ymir Camilo Acosta Rodríguez	Juan Francisco Paredes Andino
Fecha	Fecha
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]
Formato: AP-MER2014	

Sistema	Fecha	Versión	Id
Synaps	[##/##/####]	0.0.01	Mck001
Id_anterior	Wireframe		
N/A	<div style="border: 1px solid black; padding: 10px; text-align: center;"> <h2>Mensaje Bienvenida</h2> <p>Usuario:</p> <input style="width: 100%; height: 20px;" type="text"/> <p>Contraseña:</p> <input style="width: 100%; height: 20px;" type="password"/> <p style="margin-top: 20px;"><input type="button" value="Iniciar Sesión"/></p> </div>		
Acción:			
Login de la aplicación.			
Realizado Por	Validado Por	Autorizado Por	
Firma Nombre	Firma Nombre	Firma Nombre	
Fecha	Fecha	Fecha	
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	
Formato: AP-MOCK2014			

Sistema	Fecha	Versión	Id																		
Synaps	[##/##/####]	0.0.01	Mck002																		
Id_anterior	Wireframe																				
Mck001	<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> Sitio de Administración Usuario / Terminar Sesión </div> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #ADD8E6;"> <th colspan="2">Egreso</th> </tr> </thead> <tbody> <tr> <td>Egresos</td> <td style="text-align: right;">Añadir Modificar</td> </tr> <tr> <td>Prestamos</td> <td style="text-align: right;">Añadir Modificar</td> </tr> </tbody> </table> </div> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #ADD8E6;"> <th colspan="2">Ingreso</th> </tr> </thead> <tbody> <tr> <td>Ingresos</td> <td style="text-align: right;">Añadir Modificar</td> </tr> <tr> <td>Proveedores</td> <td style="text-align: right;">Añadir Modificar</td> </tr> </tbody> </table> </div> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #ADD8E6;"> <th colspan="2">Producto</th> </tr> </thead> <tbody> <tr> <td>Cuentas</td> <td style="text-align: right;">Añadir Modificar</td> </tr> <tr> <td>Productos</td> <td style="text-align: right;">Añadir Modificar</td> </tr> </tbody> </table> </div> </div>			Egreso		Egresos	Añadir Modificar	Prestamos	Añadir Modificar	Ingreso		Ingresos	Añadir Modificar	Proveedores	Añadir Modificar	Producto		Cuentas	Añadir Modificar	Productos	Añadir Modificar
	Egreso																				
	Egresos	Añadir Modificar																			
	Prestamos	Añadir Modificar																			
Ingreso																					
Ingresos	Añadir Modificar																				
Proveedores	Añadir Modificar																				
Producto																					
Cuentas	Añadir Modificar																				
Productos	Añadir Modificar																				
Acción:																					
Sitio de administración, página principal.																					
Realizado Por	Validado Por	Autorizado Por																			
Nombre	Nombre	Nombre																			
Fecha	Fecha	Fecha																			
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]																			
Formato: AP-MOCK2014																					

Sistema	Fecha	Versión	Id																
Synaps	[##/##/####]	0.0.01	Mck003																
Id_anterior	Wireframe																		
Mck002	<p>Sitio de Administración Usuario / Terminar Sesión</p> <p>Inicio > Ingreso > Ingresos</p> <p style="text-align: right;">Añadir Ingreso +</p> <p>Escoja Ingreso a Modificar</p> <p>Acción: <input type="text"/> <input type="button" value="Ir"/></p> <table border="1"> <thead> <tr> <th>Codigo</th> <th>Proveedor</th> <th>Fecha</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>XXXXXXXXXX</td> </tr> <tr> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>XXXXXXXXXX</td> </tr> <tr> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXXXX</td> <td>XXXXXXXXXX</td> <td>XXXXXXXXXX</td> </tr> </tbody> </table> <p># Ingresos</p>			Codigo	Proveedor	Fecha	Valor	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
	Codigo	Proveedor	Fecha	Valor															
XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX																
XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX																
XXXXXXXXXXXX	XXXXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX																
Acción:																			
Prototipo de interfaz de búsqueda de registros.																			
Realizado Por	Validado Por	Autorizado Por																	
Nombre	Nombre	Nombre																	
Fecha	Fecha	Fecha																	
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]																	
Formato: AP-MOCK2014																			

Sistema	Fecha	Versión	Id												
Synaps	[##/##/####]	0.0.01	Mck004												
Id_anterior	Wireframe														
Mck003	<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> Sitio de Administracion Usuario / Terminar Sesión </div> <p>Inicio > Ingreso > Ingresos > Añadir Ingreso</p> <h3 style="text-align: center;">Añadir Ingreso</h3> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Código: <input type="text" value="xxxxxxxxxx"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Fecha: <input type="text" value="xxxxxx"/> <input type="button" value="Hoy"/> <input checked="" type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Tipo: <input type="text" value="Estándar"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Proveedor: <input type="text" value="xxxxxxxxxxxxxxxxxx"/> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-bottom: 5px;"> <thead> <tr> <th colspan="4">Detalle</th> </tr> <tr> <th>Producto</th> <th>Cantidad</th> <th>Valor Unitario</th> <th>¿ Eliminar ?</th> </tr> </thead> <tbody> <tr> <td><input type="text" value="xxxxxxxxxx"/> +</td> <td><input type="text" value="xxxxxx"/></td> <td><input type="text" value="xxxxxxxxxx"/></td> <td style="text-align: center;"><input type="button" value="X"/></td> </tr> </tbody> </table> <div style="margin-bottom: 5px;"> <input type="button" value="Agregar detalle Adicional"/> </div> <div style="text-align: right; margin-top: 5px;"> <input type="button" value="Grabar y añadir otro"/> <input type="button" value="Grabar y continuar editando"/> <input type="button" value="Grabar"/> </div> </div>			Detalle				Producto	Cantidad	Valor Unitario	¿ Eliminar ?	<input type="text" value="xxxxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	<input type="text" value="xxxxxxxxxx"/>	<input type="button" value="X"/>
	Detalle														
Producto	Cantidad	Valor Unitario	¿ Eliminar ?												
<input type="text" value="xxxxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	<input type="text" value="xxxxxxxxxx"/>	<input type="button" value="X"/>												
Acción:	Ingreso de un nuevo registro.														
Realizado Por	Validado Por	Autorizado Por													
Nombre	Nombre	Nombre													
Fecha	Fecha	Fecha													
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]													
Formato: AP-MOCK2014															

Sistema	Fecha	Versión	Id						
Synaps	[##/##/####]	0.0.01	Mck005						
Id_anterior	Wireframe								
Mck004	<div style="border: 1px solid black; padding: 5px;"> <div style="display: flex; justify-content: space-between;"> Sitio de Administracion Usuario / Terminar Sesión </div> <p>Inicio > Ingreso > Ingresos > Nombre Ingreso</p> <h3>Modificar Ingreso</h3> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">Error general del formulario</div> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">Error del Campo</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Codigo: <input type="text" value="xxxxxxxxxxxxxxxxxxxx"/></div> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">Error del Campo</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Fecha: <input type="text" value="xxxxxx"/> Hoy <input type="checkbox"/></div> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">Error del Campo</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Tipo: <input type="text" value="xxxxxxxx"/></div> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">Error del Campo</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Persona: <input type="text" value="xxxxxxxx"/></div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Detalle</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Producto</th> <th>Cantidad</th> <th>¿ Eliminar ?</th> </tr> </thead> <tbody> <tr> <td><input type="text" value="xxxxxxxx"/> +</td> <td><input type="text" value="xxxxxx"/></td> <td style="text-align: center;">(X)</td> </tr> </tbody> </table> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Agregar detalle Adicional</div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <input type="button" value="X Eliminar"/> <input type="button" value="Grabar y añadir otro"/> <input type="button" value="Grabar y continuar editando"/> <input type="button" value="Grabar"/> </div> </div>			Producto	Cantidad	¿ Eliminar ?	<input type="text" value="xxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	(X)
Producto	Cantidad	¿ Eliminar ?							
<input type="text" value="xxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	(X)							
Acción:									
Prototipo de error en formulario de ingreso.									
Realizado Por	Validado Por	Autorizado Por							
Nombre	Nombre	Nombre							
Fecha	Fecha	Fecha							
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]							
Formato: AP-MOCK2014									

Sistema	Fecha	Versión	Id									
Synaps	[##/##/####]	0.0.01	Mck006									
Id_anterior	Wireframe											
Mck003	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: right;">Sito de Administracion Usuario / Terminar Sesión</p> <p>Inicio > Egreso > Egresos > Nombre Egreso</p> <h3 style="text-align: center;">Modificar Egreso</h3> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Fecha <input type="text" value="xxxxxx"/> Hoy <input type="checkbox"/> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <input checked="" type="checkbox"/> Anulado </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> Persona: <input type="text" value="xxxxxxxxxx"/> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">Detalle</th> </tr> <tr> <th>Producto</th> <th>Cantidad</th> <th>¿ Eliminar ?</th> </tr> </thead> <tbody> <tr> <td><input type="text" value="xxxxxxxxxx"/> +</td> <td><input type="text" value="xxxxxx"/></td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> </tbody> </table> <p style="font-size: small;">Agregar detalle Adicional</p> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <input type="button" value="X Eliminar"/> <input type="button" value="Grabar y añadir otro"/> <input type="button" value="Grabar y continuar editando"/> <input type="button" value="Grabar"/> </div> </div>			Detalle			Producto	Cantidad	¿ Eliminar ?	<input type="text" value="xxxxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	<input type="checkbox"/>
	Detalle											
Producto	Cantidad	¿ Eliminar ?										
<input type="text" value="xxxxxxxxxx"/> +	<input type="text" value="xxxxxx"/>	<input type="checkbox"/>										
Acción:												
Prototipo de modificación de registros.												
Realizado Por	Validado Por	Autorizado Por										
Nombre	Nombre	Nombre										
Fecha	Fecha	Fecha										
[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]	[##/##/####,DD/MM/YYYY]										
Formato: AP-MOCK2014												

CAPÍTULO 4

LIBERACIÓN DEL PRODUCTO SOFTWARE

4.1 Descripción del procedimiento

ORD.	DETALLE	OPERACIÓN					DISTANCIA		OBSERV. TECNICA
		○	□	⇒	D	▽	mts	cm	
1	Instalación de Python 2.7.3 plataforma windows	●							
2	Configuración de variables de entorno	●							
3	Instalación de Django	●							
4	Prueba de enlace Python-Django	●	●						Verificación de dependencias por parte de Django.
5	Instalación de framework .NET 4.0	●			●				
6	Instalación de MySql 5.6	●			●				
7	Instalación librería MySql-Python 1.2.4	●							Librería encargada de la comunicación de la BD y Python
8	Crear árbol Synaps	●	●						

Tabla 15 Procedimiento Deploy Synaps
Fuente (Acosta Y. / Paredes J.)

4.2 Acta de satisfacción del producto software



Latacunga, 29 de octubre del 2014

Constancia:

Que se recibe a entera satisfacción el sistema informático realizado para la Escuela de Perfeccionamiento de Aerotécnicos, con el objetivo de controlar la recepción y la distribución de productos de logística, el mismo que ha sido realizado por los Señores Ymir Camilo Acosta Rodríguez identificado con C.I. 172349775-4 y Juan Francisco Paredes Andino identificado con C.C. No. 171354330-2, en cumplimiento y certificación de la recepción de dicho programa que realmente es un aporte valioso para la Institución, la cual queda agradecida por el esfuerzo y la entereza prestada.

La presente se expide en base a la verdad y al funcionamiento exitoso del sistema informático entregado a esta dependencia para un mejor desempeño.

En constancia de lo anterior, firman los señores: Crnl. EMT. Avc. Rodrigo Cevallos Director de la EPAE y Sgop. Téc. Avc. Wilmer Encalada supervisor del Dpto. Logística EPAE, en la Ciudad de Latacunga, Provincia de Cotopaxi a los 29 días del mes de Octubre de 2014.

Atentamente,

DIOS, PATRIA Y LIBERTAD,


 Rodrigo Cevallos Ureña
 Coronel EMT. Avc.
DIRECTOR DE LA EPAE


 Wilmer Encalada
 Sgop. Téc. Avc.
SUPERVISOR DPTO. LOGÍSTICO EPAE.

Calle: Javier Espinosa entre Av. Amazonas y Av. Velasco Ibarra
 Teléfono 032811609
 Latacunga - Ecuador

Figura 12. Acta satisfacción del producto software.

Fuente (EPAE, 2014)

4.3 Aceptación del producto software



Latacunga, 27 de octubre del 2014

Supervisor Dpto. Logística EPAE

CERTIFICA

Que los señores Acosta Rodríguez Ymir Camilo con CI. 1723497754 y Paredes Andino Juan Francisco con CC. 1713543302 Egresados de la Universidad de las Fuerzas Armadas-ESPE de la carrera Ingeniería en Software, realizaron el proyecto de tesis titulado "Diseño e Implementación de un sistema regulador de bienes servibles, reparables y condenados asignados al Departamento de Logística de la EPAE aplicando la metodología estándar usada en la FAE"; el cual ha sido desarrollado de acuerdo a las necesidades de EPAE y aprobado a satisfacción.

Los interesados pueden hacer uso del presente como estimen conveniente, sin que ello signifique responsabilidad alguna para la EPAE.

Atentamente,

DIOS, PATRIA Y LIBERTAD,



 Wilmer Encalada,
 Sgop. Téc. Avc.
SUPERVISOR DPTO. LOGÍSTICA EPAE.

Calle: Javier Espinosa entre Av. Amazonas y Av. Velasco Ibarra
 Teléfono 032811609
 Latacunga - Ecuador

Figura 13. Acta de aceptación del producto software.

Fuente (EPAE, 2014)

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se diseñó e implementó un producto software que administre y distribuya los bienes servibles, reparables y condenados asignados al Departamento de Logística de la Escuela de Perfeccionamiento de Aerotécnicos EPAE, basándonos en una metodología ágil propuesta por los integrantes del presente proyecto por motivo que la Fuerza Aérea no cuenta con una metodología de desarrollo de software en la escuela.
- El departamento de logística de la EPAE no es el ente encargado de proporcionar un presupuesto que registre las necesidades de cada uno de los departamentos, motivo por el cual se determina que se debe encargarse de recibir, almacenar y distribuir los productos que le son adjudicados.
- No fue posible determinar el marco teórico del análisis, diseño e implantación por motivo que la Fuerza Aérea no cuenta con metodología de desarrollo de software alguna.
- En el Modelo Conceptual de la Necesidad de la metodología de desarrollo SEPAE se aplican las prácticas recomendadas por la IEEE 830 para la obtención de requerimientos de software.
- Se desplegó la aplicación Synaps en el equipo donde la persona encargada del manejo, control y distribución de productos lo requiere.

5.2 Recomendaciones

- Que se aplique la metodología SEPAE en futuros proyectos de desarrollo ágil de software que tengan sitio en la Escuela de Perfeccionamiento de Aerotécnicos de la Fuerza Aérea.
- El departamento de logística EPAE genere mediante documento el pedido de los requerimientos a cada uno de los departamentos y recopile las necesidades, se revise en el inventario la existencia actual de bodega y emita las necesidades reales al respectivo organismo encargado de la ejecución del PAPP para su adquisición.

Referencias Bibliográficas

Bibliografía

	#Pág.
[3] Jacinto díaz marcilla, j. E. (2012). <i>Organizacion Control</i>	11
[10] Arturo, F. (2012). <i>Python 3 al descubierto</i> . C.V.,México: Alfa omega grupo editor.	20
[14] Piattini Mario G. Análisis y diseño detallado de aplicaciones informáticas de gestión. 1ª ed. RA-MA Editorial, Madrid, 1996. (s.f.).	24
[16] Piattini Mario G. Análisis y diseño detallado de aplicaciones informáticas de gestión. 1ª ed. RA-MA Editorial, Madrid, 1996. (s.f.).	26
[15] Chalmeta R. “ADSI II. 2º Boletín de transparencias”. UJI, 1999. (s.f.).	24
[18] Hawryskiewycz, i. T. (1994). Análisis y diseño de base de datos. México: 1ra edición . P.P chen (1976)	35
Connolly, T. B. (2005). <i>Sistemas de Base de Datos</i> . Madrid: Pearson Educacion S.A.	10
J. Forcier, P. B. (2008). <i>Python Web Development with Django</i> . Boston: Pearson Education.	23
Mazier, D. (2011). <i>Gestión de Proyectos</i> . Barcelona: ENI Ediciones Software	22

Netgrafía

[1] http://epae.mil.ec/index.php?option=com_content&view=article&id=12&itemid=55	1
[2] http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html	10
[4] http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/	11
[5] http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html	11
[6] http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html	12
[7] http://staruml.sourceforge.net/en/	14
[8] https://es.scribd.com/doc/59619428/Monografia-Ejemplo-Uml	15

- [9] <http://web2py.com/books/default/chapter/36/02/el-lenguaje-python#Acerca-dePython> 16
- [11] <http://www.arquitecturadeinformacion.cl/como/wireframe2.html> 22
- [12] <http://www.djangobook.com/en/2.0/chapter01.html> 22
- [13] <http://bibing.us.es/proyectos/abreproy/12051/fichero/libros%252Flibro-django.pdf> 23
- [17] <http://www.fabforce.net/dbdesigner4/> 27
- [19] www.uazuay.edu.ec/analisis/EI%20modelo%20relacional.pdf 37
- [20] <http://xmeele.wordpress.com/2010/09/19/pruebas-unitarias-que-son/> 44
- [21] http://es.wikipedia.org/wiki/Python_Software_Foundation_Licence 61
- [22] <https://www.python.org/community> 61
- [23] <http://python.org.ar/> 61
- [24] <http://python.pe/> 61
- [25] <https://github.com/search?q=stars%3A%3E1&s=forked&type=Repositories> 61
- [26] <http://legacy.python.org/dev/peps/pep-0008/> 63
- [27] <https://docs.djangoproject.com/en/1.6/intro/tutorial01/> 63