

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA

EN SISTEMAS COMPUTACIONALES

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN
SISTEMAS COMPUTACIONALES**

TEMA:

DESARROLLO E IMPLEMENTACIÓN DE UNA PLATAFORMA GRÁFICA DE DISEÑO DE TERNOS AL ESTILO SASTRE PARA HOMBRE DE LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA” UTILIZANDO PRIMEFACES.

APLICATIVO:

PLATAFORMA GRÁFICA DE DISEÑO DE TERNOS AL ESTILO SASTRE PARA HOMBRE DE LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA” UTILIZANDO PRIMEFACES.

AUTOR:

Daniela Evelyn Cabrera Pozo

DIRECTOR:

Ing. Mauricio Rea

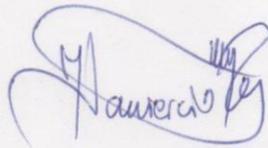
Ibarra – Ecuador

2017

CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniero en Sistemas Computacionales con el tema: **“Desarrollo e implementación de una plataforma gráfica de diseño de ternos al estilo sastre para hombre de la Microempresa Artesanal “SASTRERÍA ARTEAGA” utilizando Primefaces”**, ha sido desarrollada y terminada en su totalidad por la Srta. Daniela Evelyn Cabrera Pozo con C.C. 040154091-9 bajo mi supervisión para lo cual firmo en constancia.

Atentamente,



Ing. Mauricio Rea

DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Daniela Evelyn Cabrera Pozo, con cédula de identidad Nro. 04001540919, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado “Desarrollo e implementación de una plataforma gráfica de diseño de ternos al estilo sastre para hombre de la Microempresa Artesanal “SASTRERÍA ARTEAGA” utilizando Primefaces.” que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

(Firma):

Nombre: Daniela Evelyn Cabrera Pozo

Cédula: 040154091-9

Ibarra, junio del 2017



UNIVERSIDAD TÉCNICA DEL NORTE BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO	
CÉDULA DE IDENTIDAD:	040154091-9
NOMBRES Y APELLIDOS:	CABRERA POZO DANIELA EVELYN
DIRECCIÓN:	AV. RICARDO SÁNCHEZ Y BARTOLOMÉ GARCÍA 6-54
EMAIL:	evelyncabrera1991@gmail.com
TELÉFONO MÓVIL:	0992445993
DATOS DE LA OBRA	
TÍTULO:	“DESARROLLO E IMPLEMENTACIÓN DE UNA PLATAFORMA GRÁFICA DE DISEÑO DE TERNOS AL ESTILO SASTRE PARA HOMBRE DE LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA” UTILIZANDO PRIMEFACES”.
AUTOR:	CABRERA POZO DANIELA EVELYN
FECHA:	Junio del 2017
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
DIRECTOR:	ING. MAURICIO REA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Daniela Evelyn Cabrera Pozo, con cédula de identidad Nro. 0401540919, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.



(Firma):

Nombre: Daniela Evelyn Cabrera Pozo

Cédula: 040154091-9

AUTORÍA

Yo, **DANIELA EVELYN CABRERA POZO**, portadora de la cédula de ciudadanía número 0401540919-9, declaro bajo juramento que el trabajo **“DESARROLLO E IMPLEMENTACIÓN DE UNA PLATAFORMA GRÁFICA DE DISEÑO DE TERNOS AL ESTILO SASTRE PARA HOMBRE DE LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA” UTILIZANDO PRIMEFACES”** es de mi autoría, y que no ha sido previamente presentado para ningún otro fin de orden académico o profesional y que los resultados de la investigación que se incluyen en este documento son de mi responsabilidad.

A handwritten signature in blue ink, appearing to read 'Daniela Cabrera', is written over a circular stamp or seal that is partially obscured by the ink.

DANIELA EVELYN CABRERA POZO

CI.040154091-9



Microempresa Artesanal "SASTRERÍA ARTEAGA"

Confección de Tornos de Alta Costura al estilo SASTRE

San Gabriel, 28 de junio de 2017

Sr. René Germán Arteaga, Maestro en la Rama Artesanal de Sastrería.

CERTIFICA:

Que siendo auspiciante del proyecto de tesis de la Egresada **DANIELA EVELYN CABRERA POZO** con CI: 0401540919 quien desarrolló su trabajo con el tema: **DESARROLLO E IMPLEMENTACIÓN DE UNA PLATAFORMA GRÁFICA DE DISEÑO DE TERNOS AL ESTILO SASTRE PARA HOMBRE DE LA MICROEMPRESA ARTESANAL "SASTRERÍA ARTEAGA", UTILIZANDO PRIMEFACES**, me es grato informar que ha superado satisfactoriamente las pruebas técnicas y la revisión del cumplimiento de los requerimientos funcionales, por lo que recibo el proyecto como culminado por parte de la egresada: **DANIELA EVELYN CABRERA POZO**.

La egresada **DANIELA EVELYN CABRERA POZO** puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,

Sr. Germán Arteaga
Maestro en la Rama Artesanal de Sastrería

DEDICATORIA

Esta tesis se la dedico primeramente **a Dios**, por darme la fortaleza para salir adelante, vencer los obstáculos que se presentan día a día y demostrarme que con paciencia, responsabilidad y perseverancia todo es posible realizar.

A mis padres **Sra. Magdalena Pozo** y **Sr. Germán Arteaga**, por ser el pilar fundamental en mi vida, brindándome su apoyo constante, consejos oportunos para conseguir mis metas u objetivos; además de su comprensión, amor incondicional y por haber inculcado en mí principios y valores de superación y humildad para enfrentar los retos de la vida.

A **mis hermanas**, por las palabras de apoyo y aliento en las etapas en que mis fuerzas flaqueaban y ellas con su cariño incondicional estuvieron ahí acompañándome para poderme realizar profesionalmente.

A mis amigos y amigas entre ellos quien en vida fue **Carlos Andino**, conocido como **Justin**, quien, con su apoyo y amistad incondicional, supo guiarme en los buenos y malos momentos, los recuerdos de todo lo compartido son y serán inolvidables e inigualables y siempre los llevaré en mi corazón en cualquier lugar que me encuentre.

A todas las personas que con su granito de arena aportaron en mi formación académica y compartieron momentos de experiencia y aprendizaje para mí, dentro y fuera de mi querida Universidad Técnica del Norte.

AGRADECIMIENTO

A mi querida **Universidad Técnica del Norte** en especial a la carrera de Ingeniería en Sistemas Computacionales, por los conocimientos y la experiencia que adquirí en cada una de las aulas, agradezco infinitamente a los docentes quienes marcaron con sus enseñanzas el futuro de todos tanto en la parte académica como personal, contribuyendo al crecimiento de principios y valores durante estos cinco años de estudio.

A la **Microempresa Artesanal “SASTRERÍA ARTEAGA”**, en especial al Sr. Germán Arteaga por el apoyo e información facilitada para el desarrollo de la plataforma gráfica de diseño de ternos al estilo sastre para hombre, adquiriendo de esta manera experiencia profesional.

A mi director de tesis **Ingeniero Mauricio Rea**, por su paciencia, sus consejos, ánimo y la confianza en mí que me sirvió de inspiración para continuar en esta ardua tarea de culminar el trabajo de grado.

Mi eterna gratitud a mis padres y hermanas, porque gracias al apoyo, comprensión y consejos incondicionales durante todo el proceso del ciclo académico, pude llegar a culminar esta fase de mi vida, la mejor herencia que pude recibir, y este logro simboliza la responsabilidad, perseverancia, dedicación inculcada y sostenida por ellos, por esta y muchas más razones me queda por decir infinitas gracias y siempre los llevaré en mi corazón.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	ii
CESIÓN DE DERECHOS DE AUTOR.....	iii
1. IDENTIFICACIÓN DE LA OBRA	iv
2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD	v
DEDICATORIA	viii
AGRADECIMIENTO.....	ix
ÍNDICE DE CONTENIDOS	x
ÍNDICE DE ILUSTRACIONES.....	xii
ÍNDICE DE TABLAS	xiv
ABSTRACT.....	xvi
CAPITULO I.....	1
1. INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.2. SITUACIÓN ACTUAL.....	2
1.3. OBJETIVOS	2
1.3.1. OBJETIVO GENERAL	2
1.3.2. OBJETIVOS ESPECÍFICOS	2
1.4. PROBLEMA.....	3
1.4.1. DESCRIPCIÓN DEL PROBLEMA	3
1.5. ALCANCE	3
1.5.1. MÓDULOS DEL SISTEMA PARA LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA”	4
1.5.2. ARQUITECTURA DEL SOFTWARE A DESARROLLAR PARA LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA”.	5
1.5.3. ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA GRÁFICA.....	5
1.6. JUSTIFICACIÓN	6
1.7. PROSPECTIVA.....	6
CAPÍTULO II	7
2. MARCO TEÓRICO	7
2.1. DEFINICIONES	7
2.1.1. PLATAFORMA GRÁFICA	7
2.1.2. EVOLUCIÓN DE LA INTERFAZ GRÁFICA DE USUARIO	7
2.1.3. HERRAMIENTAS DE DISEÑO PARA INTERFAZ GRÁFICA	14
2.2. METODOLOGÍA DE DESARROLLO XP	29

2.2.1. PROGRAMACIÓN EXTREMA	30
2.2.2. CICLO DE VIDA DE LA METODOLOGÍA XP	31
CAPÍTULO III	33
3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA UTILIZANDO LA METODOLOGÍA XP.	33
1.1. DEFINICIÓN DE APLICACIONES WEB.....	33
1.2. PLANIFICACIÓN Y ESPECIFICACIONES	35
3.2.1 HISTORIAS DE USUARIO	35
3.2.2. VALORACIÓN DEL SOFTWARE	50
1.3. PROCESO DE DESARROLLO DE LA APLICACIÓN.....	51
3.3.1 CASOS DE USO.....	51
3.3.2. DIAGRAMA DE BLOQUES	52
3.3.3 ITERACIONES.....	54
CAPÍTULO IV	67
4. IMPLEMENTACIÓN DEL SISTEMA	67
4.1. FASE I: DISEÑO BASE DE DATOS	67
4.2. FASE II: CREACIÓN DEL PROYECTO	80
4.3. FASE III: INTERFAZ DE USUARIO.....	81
CAPÍTULO V	90
5. CONCLUSIONES Y RECOMENDACIONES	90
5.1. CONCLUSIONES	90
5.2. RECOMENDACIONES	91
5.3. IMPACTOS DEL SISTEMA INFORMÁTICO.....	92
5.3.1 CARACTERÍSTICAS Y BENEFICIOS DE ESTE SISTEMA.....	92
5.3.2. IMPACTOS.....	93
REFERENCIAS BIBLIOGRÁFICAS	94

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Módulos del Sistema.....	4
Ilustración 2: Arquitectura del aplicativo por capas.....	5
Ilustración 3: Arquitectura Tecnológica de la Aplicación.....	5
Ilustración 4: Primera interfaz gráfica de Usuario (1973-1981)- Xerox Corporation.....	8
Ilustración 5: Interfaz gráfica de escritorio - Xerox 8010.....	9
Ilustración 6: Evolución de Interfaces de Usuario	10
Ilustración 7: Interfaz gráfica GUI - Apple LISA	11
Ilustración 8: Software VisiCalc	12
Ilustración 9: Interfaz Gráfica- Windows 8.....	13
Ilustración 10: Interfaz Gráfica: PC, Tablets, Dispositivos móviles	13
Ilustración 11: Interfaz Natural de Usuario.....	14
Ilustración 12: Arquitectura de Java EE.....	15
Ilustración 13: Arquitectura General JavaScript.....	16
Ilustración 14: Arquitectura PostgreSQL.....	17
Ilustración 15: Arquitectura Eclipse IDE.....	18
Ilustración 16: Arquitectura Jerárquica y Modular del Servidor Apache Tomcat.	19
Ilustración 17: Arquitectura Modelo-Vista-Controlador.....	20
Ilustración 18: Respuesta a una solicitud del cliente para una página Java Server Faces.....	22
Ilustración 19: Arquitectura de JSF.....	26
Ilustración 20: Logo de Primefaces.....	27
Ilustración 21: Temas y presentaciones de Primefaces	27
Ilustración 22: Arquitectura Canvas.....	29
Ilustración 23: Valores de la Metodología XP.....	29
Ilustración 24: Fases de la metodología XP	31
Ilustración 25: Ciclo de vida de la Metodología XP	32
Ilustración 26: Arquitectura de Aplicaciones Web	33
Ilustración 27: Definición de la Web	34
Ilustración 28: Evolución de la Web.....	34
Ilustración 29: Estructura de una dirección web	35
Ilustración 30: Diagrama de Procesos – Registro Clientes.....	37
Ilustración 31: Diagrama de procesos- Registro empleados.....	38
Ilustración 32: Diagrama de Procesos- Registro inventarios.....	43
Ilustración 33: Diagrama de Procesos- Creación Contrato.....	45
Ilustración 34: Diagrama Procesos- Pago a clientes.....	47

Ilustración 35: Diagrama de Procesos- Pago empleados.....	49
Ilustración 36: Diagrama de casos de uso	52
Ilustración 37: Diagrama de bloques.....	53
Ilustración 38: Diagrama de bloques.....	53
Ilustración 39: Login del Sistema- 1era Iteración	59
Ilustración 40: Registro de usuario- 1era Iteración.	60
Ilustración 41: Inicio del Sistema- Módulos- 2da Iteración.	60
Ilustración 42: Administración de Clientes- 1era Iteración.....	61
Ilustración 43: Administración de clientes- 2da Iteración.....	61
Ilustración 44: Administración Clientes - 3era Iteración	62
Ilustración 45: Edición Cliente - 4ta iteración.....	62
Ilustración 46: Administración empleados - 1era iteración.....	63
Ilustración 47: Administración empleados - 2da iteración.....	63
Ilustración 48: Diseño prendas - 1era Iteración.....	64
Ilustración 49: Diseño prendas - 2da iteración.....	64
Ilustración 50: Elección de prendas - 1era iteración.	65
Ilustración 51: Registro de medidas de clientes - 1era iteración.	65
Ilustración 52: Módulo diseño - 2da iteración.....	65
Ilustración 53: Registro inventarios - 3era iteración	66
Ilustración 54: Esquema de base de datos del sistema.	79
Ilustración 55: Estructura Framework JSF.....	80
Ilustración 56: Login del Sistema.....	81
Ilustración 57: Pantalla de inicio del sistema.	82
Ilustración 58: Creación del Contrato.	82
Ilustración 59: Creación contrato versión final.	83
Ilustración 60: Registro de las medidas de la prenda.	83
Ilustración 61: Costo de la prenda.....	84
Ilustración 62: Lista de contratos realizados.	84
Ilustración 63: Ventana para la toma de medidas de los clientes.	85
Ilustración 64: Ventana para el ingreso de medidas de los clientes.	85
Ilustración 65: Lista de contratos a cancelar.	86
Ilustración 66: Pago de Clientes por Contrato.....	86
Ilustración 67: Reporte del contrato creado.	87
Ilustración 68: Hoja de Impresión- Contrato firmado por el cliente.	87
Ilustración 69: Pago empleado por contrato.....	88
Ilustración 70: Ventana para el Registro de Inventario.	88
Ilustración 71: Reporte de Inventario.....	89

Ilustración 72: Lista de Valor Estimado en metros de tela invertidos por contrato firmado en la fecha correspondiente.....	89
--	----

ÍNDICE DE TABLAS

Tabla 1: Ámbitos en el que se coloca el bean administrado.....	24
Tabla 2: Diferencias entre metodologías ágiles y no ágiles.	32
Tabla 3: Historia de Usuario 1	36
Tabla 4: Historia de Usuario 2.	37
Tabla 5: Historia de Usuario 3.	39
Tabla 6: Historia de Usuario 4.	40
Tabla 7: Historia de Usuario 5.	41
Tabla 8: Historia de Usuario 6.	42
Tabla 9: Historia de Usuario 7.	44
Tabla 10: Historia de Usuario 8.	46
Tabla 11: Historia de Usuario 9.	48
Tabla 12: Historia de Usuario 10.	50
Tabla 13: Costo del proyecto.	51
Tabla 14: Iteraciones por Actividades.....	54
Tabla 15: Iteraciones por Tareas.	56

RESUMEN

El presente proyecto es el desarrollo e implementación de una plataforma gráfica de diseño de ternos al estilo sastre para hombre utilizando Primefaces, la cual será implantada en la Microempresa Artesanal “Sastrería Arteaga”, ubicada en San Gabriel provincia del Carchi,

El sistema permite almacenar toda la información acerca de los clientes, empleados, diseños e inventario de materia prima para llevar a cabo el proceso de confección de ternos desde de la microempresa de una manera sistemática y ordenada, de tal manera que la información se facilite de forma rápida y con datos veraces, para la toma de decisiones.

El Capítulo I, trata de una pequeña descripción de antecedentes, justificación, objetivos, alcance y lo referente a la problemática que generó el desarrollo del presente sistema.

El Capítulo II, se analiza lo referente a las herramientas que se utilizarán durante el desarrollo, analiza el costo beneficio, además de los detalles de la metodología que se aplicará en el sistema informático titulado: "Desarrollo e implementación de una plataforma gráfica de diseño de ternos al estilo sastre para hombre utilizando Primefaces, la cual será implantada en la Microempresa Artesanal “Sastrería Arteaga” utilizando Primefaces".

El Capítulo III, analiza los procesos que intervienen en el diseño y confección de las prendas (ternos al estilo sastre para hombre), con toda la documentación de las actividades y tareas que se realizan en la Microempresa Artesanal "Sastrería Arteaga".

Además, se realiza el diseño y análisis de la automatización de estos procesos mediante la herramienta tecnológica propuesta, así como la documentación de requisitos, de acuerdo al diagrama de la metodología que se aplica en su desarrollo.

El Capítulo IV, detalla todo el desarrollo (arquitectura del software), implementación y pruebas del sistema, siguiendo la metodología XP (eXtreme Programming).

El Capítulo V, aquí se encuentra lo referente a las conclusiones y recomendaciones que se adquirieron durante el proceso de investigación, análisis y desarrollo del sistema informático.

ABSTRACT

This project refers to the development and implementation of a graphic design platform that creates tailor-style suits for men using Primefaces Database Software, which will be implemented at the ‘Sastreria Arteaga’ a microenterprise located in the Town of San Gabriel, Province of Carchi.

This system enables the storage information pertaining to clients, employees, designs as well as prime material to carry on the suit tailoring process for the business in a systematic and orderly way so that data access is fast and reliable when applying the decision making process.

Chapter I provides a brief background description, grounds, objectives and extent to the principles that motivated the creation of this database software.

Chapter II evaluates tools that will be used during the process of cost benefit apart from the methodology details that will be implemented to an IT System called *‘Developing and Implementing of a graphic design platform that creates tailor-style made suits for men using Primefaces Database Software’* which will be implemented at the ‘Sastreria Arteaga’.

Chapter III evaluates the processes involved in the design and tailoring of the clothing (men tailor-made suits) including the proper records of business procedures performed at the ‘Sastreria Arteaga’. Moreover, the analysis and design automation processes are executed through a technological tool together with the key requirements that comply with the methodology diagram required by its own development.

Chapter IV details software architecture, system tests and implementation according to the XP (*eXtreme Programming*).

Chapter V refers to conclusions and recommendations acquired during the research process, evaluation in addition to the IT system development.

CAPÍTULO I

1. INTRODUCCIÓN

En este capítulo se expone la descripción de antecedentes de la Microempresa Artesanal “Sastrería Arteaga” y su área para la cual se implantará el presente sistema informático, donde se describe el problema, objetivos generales, objetivos específicos, justificación y alcance del presente proyecto.

1.1. ANTECEDENTES

Dentro del desarrollo tecnológico, inicialmente se utilizaba aplicaciones de escritorio con procesos importantes para aplicar a un negocio o empresa en particular, por ejemplo, los sistemas de facturación, los cuales son utilizados por cooperativas, bancos y demás organizaciones. Con el pasar del tiempo se sintió la necesidad de migrar la información hacia otros puntos de almacenamiento que conjuntamente con los avances tecnológicos se empezó a trabajar con conexiones de redes, de tal manera que la información sea accesible para el usuario desde cualquier parte del mundo.

Mediante esta innovación surge la necesidad del uso de sistemas web, los cuales facilitan este trabajo; debido a que ahorran costos de hardware y software, resultan fáciles de usar tanto de manera colaborativa como a distancia, son escalables, de fácil actualización, provocan menos errores o problemas y además los datos que se manejan son más seguros.

La Microempresa Artesanal “SASTRERÍA ARTEAGA” tiene 40 años de existencia y dedicación en las labores de arte y confección de prendas al estilo sastre para hombre, brindando un servicio de calidad a sus clientes.

Por lo tanto, se debe considerar la utilización de tecnologías de información y comunicación para una rápida, ordenada y sistematizada gestión de procesos, así como el beneficio de disponer de los datos en cualquier momento.

1.2. SITUACIÓN ACTUAL

Actualmente la Microempresa Artesanal “SASTRERÍA ARTEAGA” no posee una plataforma gráfica para presentar sus diseños en confección de ternos estilo sastre para hombre a diferentes partes del mundo, además el registro de información de sus clientes, el inventario de los materiales de bodega y el pago a sus empleados se los realiza manualmente y se almacena de forma física en libros.

También cabe mencionar que el uso de recursos es muy alto y requiere de mucho tiempo, esto en varias ocasiones demanda de tiempo en acceder a la información de los clientes y a la presentación de la gama de estilos de ternos que ofrece la microempresa lo que genera una desventaja frente a la competencia.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar e implementar una plataforma gráfica de diseño de ternos al estilo sastre para hombre de la Microempresa Artesanal “SASTRERÍA ARTEAGA” utilizando Primefaces.

1.3.2. OBJETIVOS ESPECÍFICOS

- ✓ Analizar los procesos de registro y control de clientes de la microempresa artesanal “SASTRERÍA ARTEAGA”.
- ✓ Examinar los procesos de pago a los empleados.
- ✓ Conocer el proceso de inventario de los materiales de corte y confección.
- ✓ Realizar una investigación acerca de una de las aplicaciones ricas como son JSF acompañado de una de sus bibliotecas llamada Primefaces.
- ✓ Desarrollar la plataforma gráfica utilizando la metodología XP.
- ✓ Implementar la plataforma gráfica en la microempresa artesanal “SASTRERÍA ARTEAGA”.

1.4. PROBLEMA

Presentación de diseños estilo sastre para hombre, no innovadora ni accesible desde diferentes partes del mundo, impidiendo el crecimiento y desarrollo de la Microempresa Artesanal “SASTRERÍA ARTEAGA”.

1.4.1. DESCRIPCIÓN DEL PROBLEMA

En el desarrollo de plataformas gráficas, se han establecido nuevas arquitecturas de desarrollo y nuevos enfoques en algunas áreas del conocimiento; en la actualidad existen sistemas orientados al manejo de procesos contables, auditoría, médicos, entre otros, implementados en varias empresas que realizan estas funciones. Sin embargo, en el área textil y artesanal no existen plataformas gráficas enfocadas específicamente en la presentación de diseños en confección de ternos estilo sastre para hombre conjuntamente con los procesos que intervienen en la venta de este producto.

La Microempresa SASTRERÍA ARTEAGA presenta este inconveniente debido a que está destinada a la producción de este tipo de confecciones y desea presentar sus diseños a diferentes partes del mundo cumpliendo con el objetivo en ventas de su producto, además de registrar información de sus clientes, inventariar los materiales de bodega y el pago a sus empleados, procesos que se llevan a cabo manualmente y se encuentran almacenados de forma física.

De igual forma restar el uso de recursos y tiempo, esto en varias ocasiones provoca que el cliente se sienta inconforme debido a la demora en la presentación de la información acerca de la gama de estilos de ternos que ofrece la microempresa lo que genera una desventaja frente a la competencia.

1.5. ALCANCE

El Software será desarrollado bajo la plataforma de Windows, manejando las herramientas de Eclipse con el lenguaje java, utilizando la arquitectura de JSF y para su interfaz de usuario el framework Primefaces y html5 ya que cuenta con el objeto Canvas útil para el desarrollo de la plataforma gráfica, además de que sus componentes son muy compatibles en esta estructura.

Este sistema estará basado en una de las metodologías ágiles XP. La plataforma gráfica además de llevar el registro y control sistematizado de los procesos que maneja la microempresa, proporcionará a sus clientes la oportunidad de visualizar sus prendas antes de ser confeccionadas mostrando una gama diseños de ternos al estilo sastre para hombre accesibles en diferentes partes del mundo, representando un gran valor agregado frente a la competencia y brindando sus servicios de forma rápida y eficiente a sus clientes.

1.5.1. MÓDULOS DEL SISTEMA PARA LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA”

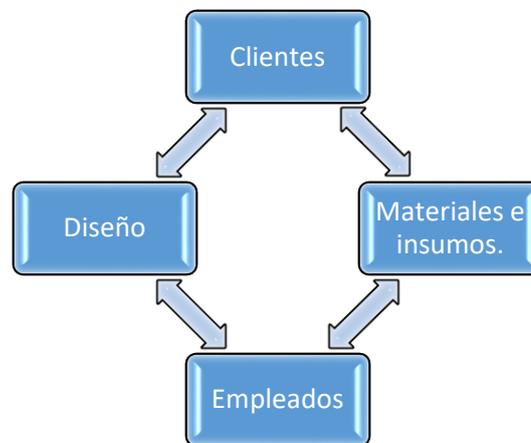


Ilustración 1: Módulos del Sistema

Fuente: Propia

Módulo de Clientes

- Registro y control de la información de los clientes.
- Citas para toma de medidas, pruebas y entrega de su prenda.

Módulo de Empleados

- Registro y control de la información de empleados.
- Préstamos.
- Horarios de trabajo.
- Pagos mensuales.

Módulo de Materiales e Insumos

- Registro de materiales e insumos (Corte, confección y telas).

Módulo Diseño

- Diseño de ternos y moldes prediseñados (Canvas).

1.5.2. ARQUITECTURA DEL SOFTWARE A DESARROLLAR PARA LA MICROEMPRESA ARTESANAL “SASTRERÍA ARTEAGA”.

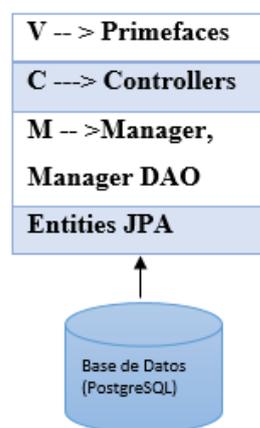


Ilustración 2: Arquitectura del aplicativo por capas

Fuente: Propia

1.5.3. ARQUITECTURA TECNOLÓGICA DE LA PLATAFORMA GRÁFICA.

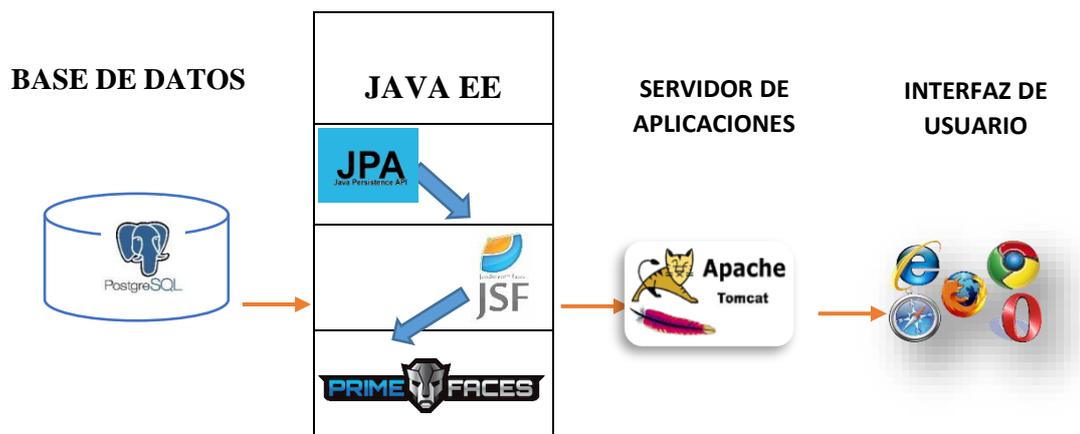


Ilustración 3: Arquitectura Tecnológica de la Aplicación

Fuente: Propia

1.6. JUSTIFICACIÓN

El desarrollo de este sistema servirá como herramienta útil para la Microempresa Artesanal “SASTRERÍA ARTEAGA” debido a que proporcionará atención de calidad y servicio eficiente a sus clientes.

Al igual que impulsará el desarrollo tecnológico tanto de las grandes empresas como de las microempresas para que independientemente del bien o servicio que ofrezcan a sus clientes, puedan hacer uso de la tecnología que les facilite tener procesos sistematizados y organizados en el desempeño de sus labores.

1.7. PROSPECTIVA

Esta plataforma gráfica permitirá dar un buen servicio por parte de la Microempresa Artesanal “SASTRERÍA ARTEAGA”, lo que impulsará la confianza de sus clientes, aumentará la mano de obra, generará oportunidades laborales, agilizará el tiempo de trabajo, permitirá llevar procesos de control y registro de clientes, empleados y materiales de bodega debidamente sistematizados y ordenados.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. DEFINICIONES

2.1.1. PLATAFORMA GRÁFICA

Se define como un software de desarrollo donde su interfaz para el usuario utiliza un conjunto de imágenes y objetos gráficos para crear un vínculo amigable, llamativo, he intuitivo que facilite la comunicación entre el usuario y el sistema, ofreciéndole varios componentes, íconos, menús y funciones que se utiliza para realizar sus respectivas actividades; como ejemplo tenemos la plataforma gráfica de Windows, contiene aplicaciones funcionales de fácil interacción con el usuario lo cual hace que el sistema no de problemas y cumpla con las necesidades del cliente.

2.1.2. EVOLUCIÓN DE LA INTERFAZ GRÁFICA DE USUARIO

En sus inicios, las computadoras digitales funcionaban utilizando dígitos de 0 y 1, según las ideas de Vannevar Bush (1937), ingeniero científico estadounidense, donde se formaron los conceptos de ordenador personal e hipertexto. En esta época y hasta el año de 1962, se interactuaba con las computadoras utilizando batch processing, nombre que caracteriza al tipo de programas que se ponían en ejecución sin el control o supervisión directa del usuario.

Con el pasar del tiempo y los años dedicados a la investigación, se estableció la verdadera relación entre el ser humano y la máquina hecho que fue realizado por el ingeniero eléctrico, Douglas Engelbart (1968), quien en su impulso de materializar la idea del científico Vennevar Bush (1968) acerca del sistema Memex (aparato que consiste en bases planas con una superficie translúcida, capaz de encontrar a alta velocidad, información almacenada en una base de datos).

Este proyecto fue apoyado por la Fuerza Aérea de Estados Unidos como una demostración de su sistema OnLine System (NLS), donde presenciaron las cámaras de televisión testificando este acontecimiento histórico.

El sistema OnLine System (NLS) trataba acerca de una presentación multimedia de líneas vectoriales y de texto. El cual era manejado por Engelbart utilizando dos teclados el primero un teclado común de una máquina de escribir convencional y el segundo un teclado de cinco letras junto con una caja pequeña que hoy se conoce como mouse, considerado como el primer mouse con tres botones.

Lo que originó que también se pueda apreciar la aparición del primer cursor, visualizado como una pequeña línea vectorial.

Tomando en cuenta esta presentación se determina que lo que el científico Engelbart (1968) mostró las ligas hipertextuales, la edición de texto, la creación colaborativa de documentos, el e-mail, el chat, e incluso el video-conferencia.

Seguido, nace la primera interfaz gráfica, en la que Xerox Palo Alto Research Center (PARC, 1973 -1981), empresa de investigación y desarrollo, propiedad de Xerox Corporation; logra un crecimiento dentro de la informática desarrollando la primera computadora personal que usó la interfaz moderna de Xerox Alto.

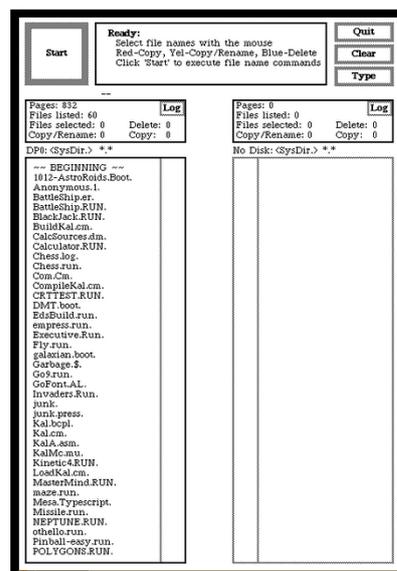


Ilustración 4: Primera interfaz gráfica de Usuario (1973-1981)- Xerox Corporation.

Fuente: (“Interfaces gráficas (GUI): Definición y evolución”,2012)

En la Ilustración 5, podemos observar el primer ordenador Xerox 8010 que introdujo la interfaz gráfica de escritorio donde se aplica el concepto de oficina virtual.

Proporcionando al usuario la facilidad representar la información que se ve en la pantalla y el resultado final que pueda tener un documento escrito tras ser impreso.

Steve Jobs-Steve Wozniak, (1976), científicos electrónicos, fundaron la compañía Apple, lanzando la computadora Apple II, con soporte para gráficos y texto, pero con interacción a partir de líneas de comando.

A continuación, en la **Ilustración 6**, se muestran avances donde se resume y se narra el proceso de desarrollo que tomó la interfaz gráfica a partir de este período.



Ilustración 6: Evolución de Interfaces

Fuente: Propia

Definiendo el diagrama la interfaz de usuario inició mediante el uso de líneas de comandos expresada por las siglas **CLI**(Command-Line Interface), esta interfaz se utilizó desde los inicios de la computación sobresalió por el uso de las tarjetas perforadas y los mecanismos que funcionaban de igual manera.

En la interfaz CLI como forma más automatizada tenemos los archivos batch que lee órdenes desde un archivo de scripts. De igual manera, es considerada parte fundamental de los shells o emuladores de terminal, apareciendo en todas las interfaces de escritorio GNOME, KDE, Microsoft Windows como un método para ejecutar aplicaciones rápidamente, conjuntamente con la interfaz de lenguajes interpretados tales como Java, Python, Ruby o Perl, entre los más conocidos. También se utilizan en aplicaciones cliente-servidor, en gestores de bases de datos, en clientes FTP. (“Interfaces gráficas (GUI): Definición y evolución”,2012)

Seguido de la interfaz de usuario ejecutada mediante línea de comandos surge la interfaz gráfica conocida como **GUI** (Graphical User Interface) que viene a ser un sistema informático que utiliza un conjunto de imágenes y objetos gráficos mejorando representación del lenguaje visual y funcionalidad además de dar como resultado una interacción amigable y de calidad para el usuario.

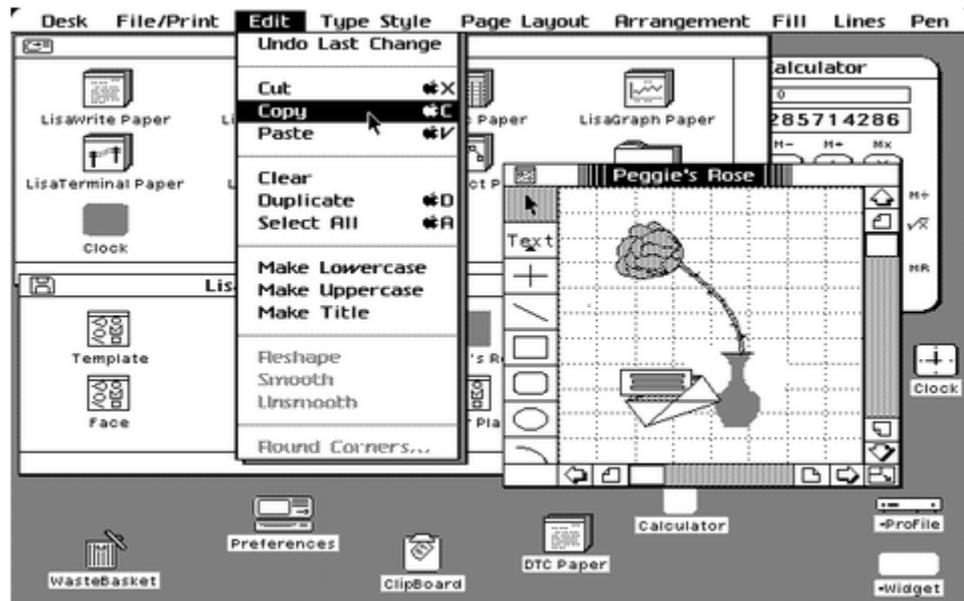


Ilustración 7: Interfaz gráfica GUI - Apple LISA

Fuente: ("Interfaces gráficas (GUI): Definición y evolución", 2012)

Un año después en 1980 se sacó la versión mejorada y más accesible al usuario conocida como Macintosh, período en el cual VisiCalc introdujo la primera hoja de cálculo llamada VisiCalc, software programado para correr en equipos PC de IBM con altos requerimientos tecnológicos lo que resultó un gran impedimento para su pronto éxito.

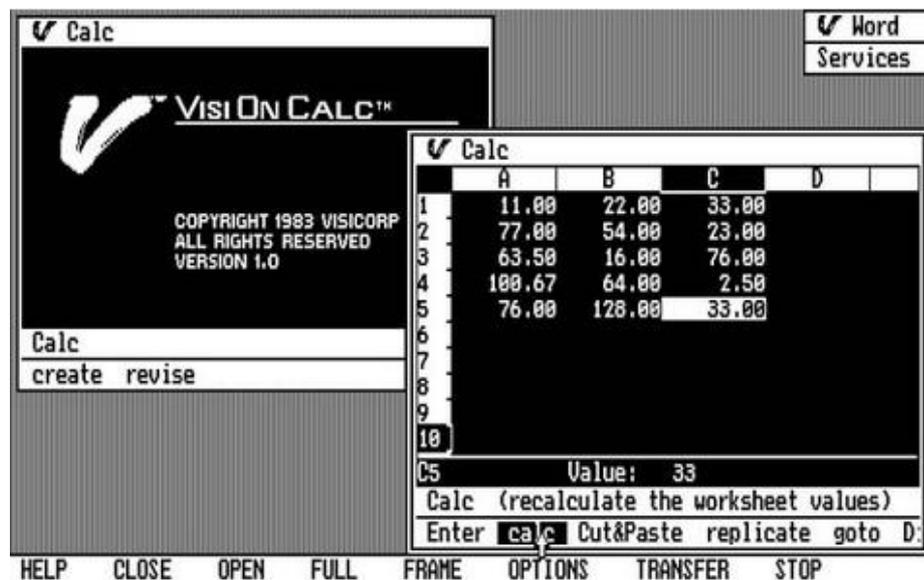


Ilustración 8: Software VisiCalc

Fuente: (“Interfaces gráficas (GUI): Definición y evolución”, 2012)

De igual forma, Bill Gates (1980) (empresario, informático estadounidense, cofundador de la empresa de software Microsoft), lanzó al mercado la primera versión de Windows. A continuación, varias empresas investigadoras y desarrolladoras contribuyeron a la competencia del mercado donde fueron mostrando sus versiones mejoras permitiendo al usuario llevar sus archivos de forma ordenada y con una visualización realmente amigable y aceptable, cumpliendo con los objetivos del producto.

Citando algunos software de acuerdo a la evolución de interfaz gráfica tenemos: Windows 3.0(1990), Workbench 2.04 (1991), Mac OS System 7 (1991), Windows 3.1 (1992), OS/2.2.0(1992), Windows 95(1995), MAC OS System 8 (1997), Windows 98(1998), KDE 1.0 (1998), BeOS 4.5(1999), GNOME 1.0(1991), Mac OS X(2001), Windows XP(2001), KDE 3(2002), Windows Vista(2007), Mac OS X Leopard (2007), GNOME 2.24 (2008), KDE 4.x (2008), Windows 7 (2009), Mac OS X Lion (2010).

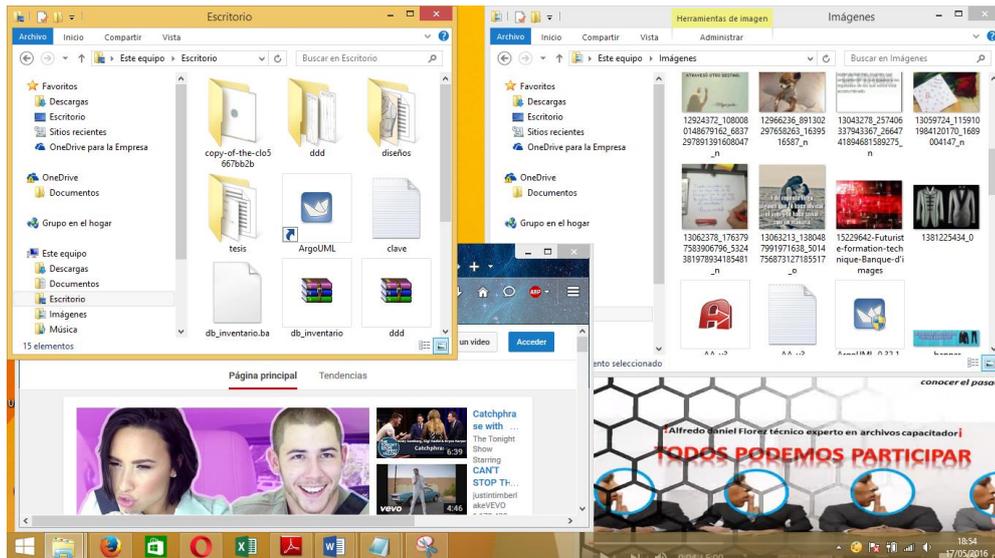


Ilustración 9: Interfaz Gráfica- Windows 8

Fuente: Propia.

Además, se debe resaltar que las interfaces de usuario (GUI), no se pueden apreciar únicamente en computadoras de escritorio y laptops sino también en dispositivos móviles donde el desarrollo solo hace tomando en cuenta que el aplicativo debe ajustarse al tamaño del dispositivo; pero que debe brindar la misma facilidad de uso al usuario al realizar sus actividades cotidianas.



Ilustración 10: Interfaz Gráfica: PC, Tablets, Dispositivos móviles

Fuente: <https://sites.google.com/site/appsdispositivosymas/contenido/pagina-cuatro>

Finalmente y con constantes investigaciones los desarrolladores han dado un gran impacto a la ciencia de la tecnología de donde surge actualmente la interfaz gráfica conocida como **NUI** (Natural User Interface), en la que el usuario interactúa con un sistema o aplicación, sin utilizar sistemas de mando o dispositivos de entrada de las GUI como sería un ratón, teclado alfanumérico, lápiz óptico, touchpad, entre otros dispositivos, los cuales son reemplazados por el uso de movimientos gestuales del cuerpo o de alguna de sus partes tales como las manos, sirviendo de mando de control. En el caso de pantallas con capacidad táctil, la ejecución de funciones se la realiza por medio de las yemas de los dedos, en uno o varios contactos y asimismo el control visual cercano a la pantalla y próximamente se está desarrollando el control de sistemas operativos por medio de la voz humana.



Ilustración 11: Interfaz Natural de Usuario

Fuente: (García, 2013)

2.1.3. HERRAMIENTAS DE DISEÑO PARA INTERFAZ GRÁFICA

En la actualidad existe una gama de herramientas libres dentro la plataforma de JAVA EE que nos permite realizar aplicaciones o sistemas a nivel empresarial, manejando un conjunto de componentes para su desarrollo, partiendo desde el

almacenamiento de información mediante la creación de base de datos cumpliendo con la arquitectura para el usuario, así tenemos:

2.1.3.1 JAVA

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995. Además, es rápido, seguro y fiable.

Tiene varios entornos de funcionamiento entre ellos tenemos: dispositivos móviles y sistemas embebidos, navegador web, sistemas de servidor, aplicaciones de escritorio y plataformas soportadas.

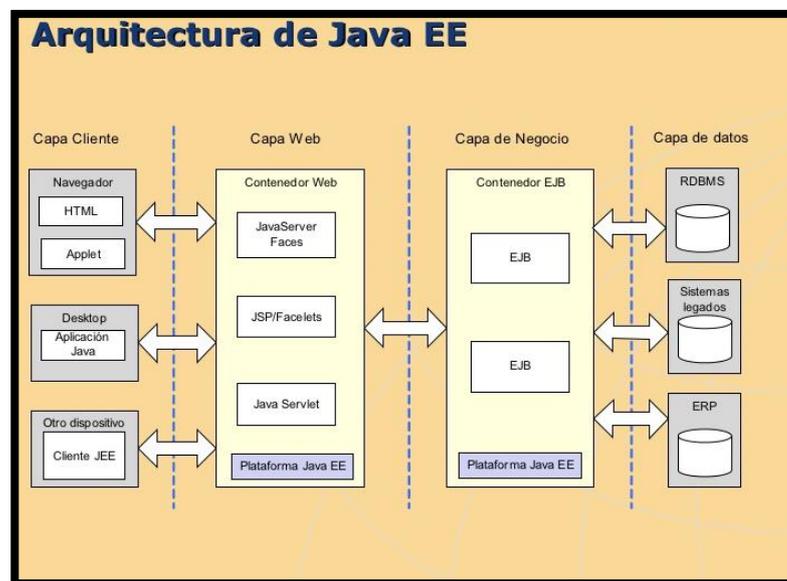


Ilustración 12: Arquitectura de Java EE.

Fuente: (Montoño, 2011).

2.1.3.2 JAVA SCRIPT

Es un lenguaje de programación, al igual que PHP, utiliza principalmente del lado del cliente, es decir, se ejecuta en nuestro ordenador, no en el servidor; permitiendo crear efectos atractivos y dinámicos en las páginas web.

La primera forma de usar JavaScript dentro de una página web es embebiendo directamente el código JavaScript dentro del código HTML.

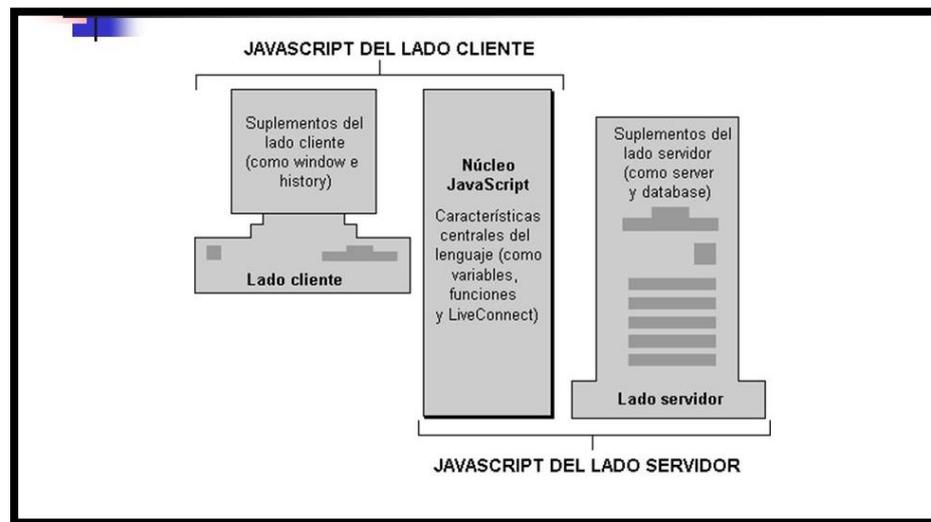


Ilustración 13: Arquitectura General JavaScript.

Fuente: (“Arquitectura general JavaScript”, n.d.)

2.1.3.3. POSTGRESQL

“Es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional”. (Acosta, Tumbarell, y otros, 2016).

Nos facilita las operaciones de consulta, inserción, actualización y eliminación de datos mediante tablas a medida que se lo requiere, realizar procedimientos almacenados, aplicar triggers, de igual forma permite ejecutar procesos de clustering, de tal forma que los datos estén accesibles para los usuarios desde diferentes ordenadores proporcionando información real, íntegra y sistematizada.

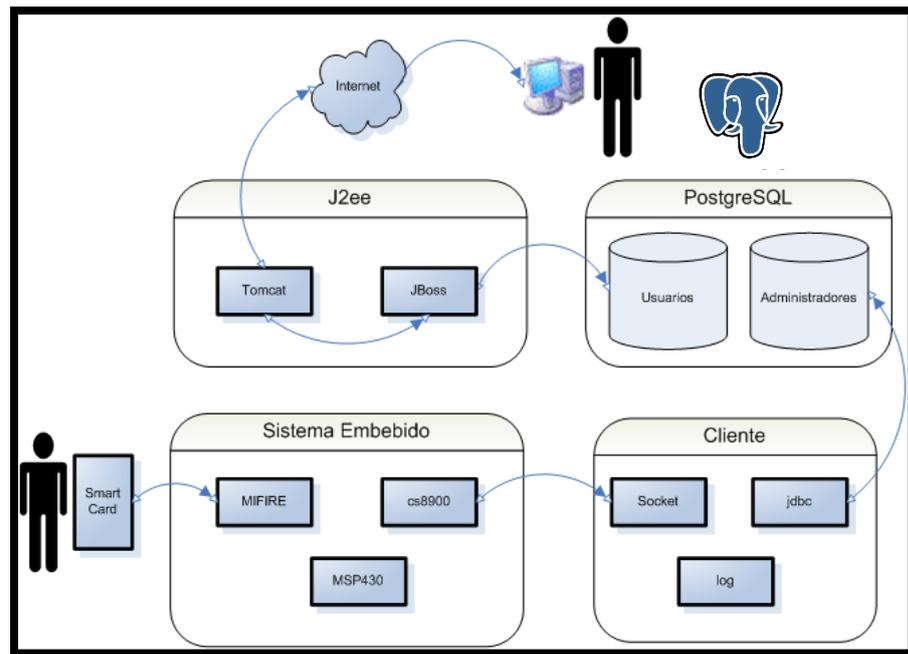


Ilustración 14: Arquitectura PostgreSQL

Fuente: (Martinez,2016)

2.1.3.4. ECLIPSE IDE

Este IDE de desarrollo es una herramienta completa que facilita el trabajo al programador o desarrollador debido a que le proporciona un conjunto de herramientas y librerías que pueden venir instaladas o se permite la configuración al implementar la librería necesaria dependiendo del aplicativo.

Mediante la arquitectura MVC (Modelo, Vista y Controlador) hacemos uso de lo que es la persistencia JPA, con la finalidad de extraer la información almacenada en tablas dentro de la base de datos, transformándola en objetos mapeados para trabajar de acuerdo a las funciones que debe llevar el sistema que se va a desarrollar.

También nos facilita la implementación de un formato de arquitectura mediante la utilización de una gama de frameworks, entre los cuales tenemos: Framework JSF, Zend, Spring, Symfony, Mylyn, EGit, (The Eclipse Foundation, 2015); entre los más conocidos; que permite

realizar cualquier tipo de aplicación que se desee programar en esta herramienta cumpliendo con su formato o estándar correspondiente.

Además, nos permite configurar ciertos servidores de aplicaciones que harán posible la ejecución y publicación de sistema para el usuario citando algunos ejemplos están: Apache Tomcat, JBoss, Widfly usados según el criterio del desarrollador.

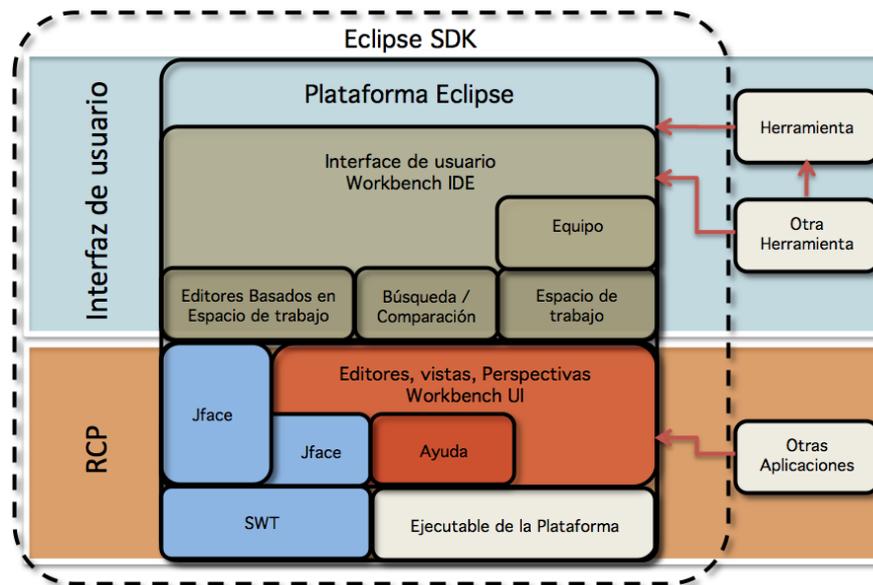


Ilustración 15: Arquitectura Eclipse IDE

Fuente: (The Eclipse Foundation, 2017)

2.1.3.5. SERVIDOR DE APLICACIONES APACHE TOMCAT

“Tomcat es un contenedor de servlets que se utiliza en la Referencia oficial de la implementación para Java Servlet y JavaServer Pages (JSP).

Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process). Apache Tomcat es desarrollado en un entorno abierto y participatorio, bajo la licencia de Apache Software License”. (Sánchez, González, Velázquez, 2016).

Considerado como un servidor de aplicaciones (Software libre) integrado en el IDE de desarrollo Eclipse, utilizado como servidor de aplicaciones Web con HTML, servlets y JSP o como complemento de servidor Apache.

Permite ejecutar aplicaciones bajo la plataforma de Java EE siendo gratuito, da la posibilidad de subdividir las aplicaciones que se encuentran en los dominios permitiendo que estas compartan librerías, logs y demás configuraciones al ser manipulados de forma particular; además de que convierte las páginas de JSP y JSF en Servlets.

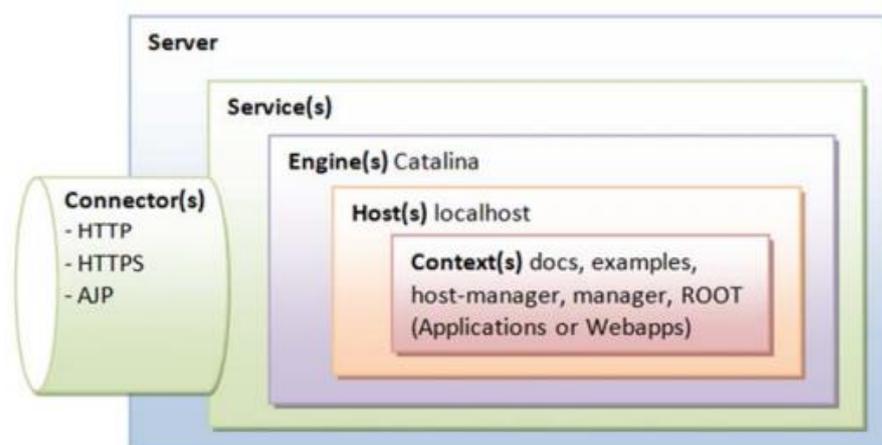


Ilustración 16: Arquitectura Jerárquica y Modular del Servidor Apache Tomcat.

Fuente: (Pavón, 2012)

2.1.3.6. ARQUITECTURA MODELO-VISTA-CONTROLADOR (MVC)

La arquitectura MVC surge con el objetivo de reducir e independizar el trabajo de los programadores y diseñadores en la implementación de sistemas múltiples, esta arquitectura divide las partes que conforman una aplicación en el modelo, las vistas y los controladores.

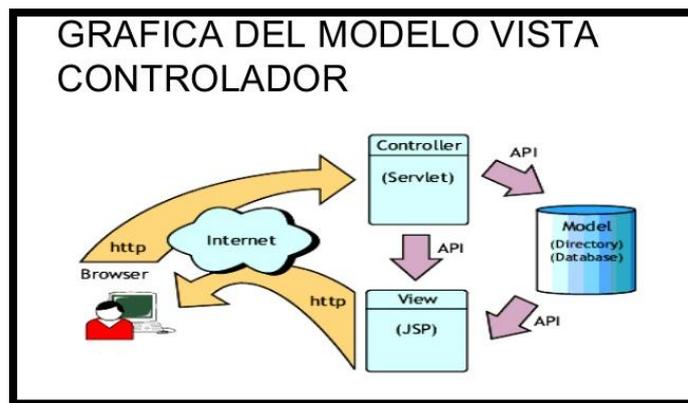


Ilustración 17: Arquitectura Modelo-Vista-Controlador.

Fuente: (Bugarín,2013).

Lo que garantiza que el mantenimiento del software sea sencillo, rápido e independiente al estar desarrollado en tres capas, de esta manera permite realizar un trabajo más eficaz y mejor organizado en el momento de la implementación. (Cajilima, y otros, 2015).

2.1.3.7. FRAMEWORK JSF

“Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones”. (Gutiérrez, 2014, pág.1).

“Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web”. (Gutiérrez, 2014, pág.1).

➤ ¿Qué es Java Server Faces?

La tecnología Java Server Faces está considerada como una estructura de componentes del lado del servidor para la creación y desarrollo de aplicaciones oficiales Java Enterprise basadas en web para simplificar los procesos de interfaces de usuario. La misma que consiste en:

- Una API (Application Programming Interface) que proporciona funciones, procedimientos para representar los componentes y administrar su estado, manejo de eventos que intervienen dentro del proceso que está enfocado el software, validación del servidor y conversión de datos.
- Usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, trabajar con archivos XML dependiendo de la funcionalidad que se aplique al desarrollo del sistema.
- Bibliotecas de etiquetas para agregar componentes a páginas web (Interfaz de Usuario) y para que estos sean conectados a objetos de lado del servidor (Mapeo de objetos y programación).
- Un conjunto de beans administrados, que son objetos ligeros y gestionados por contenedores. En una aplicación Java Server Faces, los beans administrados sirven como beans de respaldo, que definen propiedades y funciones para componentes de interfaz de usuario en una página.
- Un descriptor de despliegue web (archivo web.xml).
- Contiene uno o más archivos de recursos de configuración de la aplicación, como el archivo faces-config.xml, que se puede utilizar para definir las reglas de configuración de los beans y otros objetos personalizados, como componentes personalizados.

La **Ilustración 18**, muestra la interacción entre el cliente y el servidor en una típica aplicación de Java Server Faces. En respuesta a una petición del cliente donde una página web es representada por el contenedor web implementado por la tecnología de Java Server Faces.

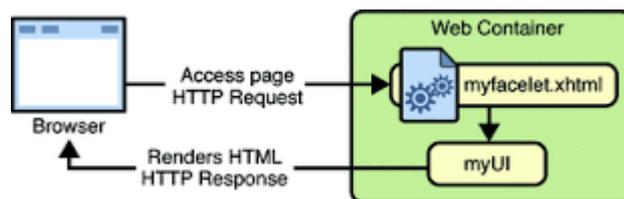


Ilustración 18: Respuesta a una solicitud del cliente para una página Java Server Faces.

Fuente: (Oracle Corporation, 2010)

➤ Características:

JSF está destinado para facilitar la construcción y desarrollo de interfaces de usuario del servidor (UI) para aplicaciones web. Las bibliotecas de etiquetas contienen controladores que se implementan en el componente proporcionando un modelo de programación bien definido y con el mínimo de esfuerzo al completar las siguientes tareas:

- Creación de páginas web, con herramientas tales como: botones, cajas de texto, tablas de datos, entre otras. Similares a objetos empleados para la generación de aplicaciones de escritorio, por ejemplo: Aplicaciones Swing.
- Genera interfaces que puedan ser consumidas por diferentes tipos de dispositivos clientes. Por ejemplo: PC de escritorio, Tablet PC o un PDA, sin tener que crear nuevas interfaces para cada uno de estos ambientes, ni desarrollar diferentes procesos.
- Permite colocar componentes en una página web agregando etiquetas de componentes funcionales y amigables para la presentación de datos al usuario final, sin tener que escribir todo el código que se necesita para generar estos componentes debido a que muchos de los IDE's actuales para el desarrollo, incorpora muchos de los controles utilizados por el estándar de manera que las interfaces o vistas de la aplicación pueden ser desarrolladas utilizando ambientes visuales.

- Conecta los componentes de una página web con los datos del servidor.
- Genera eventos por componentes en el código de las aplicaciones.
- Envía peticiones o solicitudes al servidor siempre y cuando este activa la sesión y el tiempo de respuesta de la aplicación en el server.
- Reutiliza y amplía gama componentes a través de la personalización.

➤ **¿Hacia dónde va dirigido JSF?**

Desde sus inicios, JSF estaba enfocado a proveer un estándar de controles para aplicaciones mediante un grupo de herramientas predefinidas con comportamientos embebidos que fueran capaces de manipular datos implícitamente expresados en su implementación.

El manejo de procesos, la comunicación de componentes o acceso a orígenes de datos se manejaba dentro de otros frameworks, entre ellos: Struts o WebWorks. De esta manera, JSF solo se encargaba de enmascarar la última porción del manejo de las entradas y salidas de la capa Request/Response de la comunicación HTTP de las aplicaciones basadas en ambientes Web.

Con el paso del tiempo, JSF ha ido dando soluciones para cada una de las partes involucradas en el proceso, como por ejemplo la capacidad de establecer comunicaciones con orígenes de datos que luego pueden ser manipulados desde las interfaces desarrolladas con los componentes de la aplicación.

Se considera los casos en los que se pueda integrar JSF con otros frameworks dirigidos para aplicaciones basadas en Web donde se considera el análisis de la aplicación, de su tamaño y complejidad y de las necesidades de procesamiento que presente cada una. (Loor, 2013).

➤ COMPONENTES DE JSF

• Managed Bean

Objeto identificado dentro de la aplicación en la cual se especifica una identificación, un alcance (scope) que puede ser:

Scope	Descripción
@RequestScoped	Bean vive mientras vive la respuesta HTTP. Se crea en una solicitud HTTP y se destruye cuando se termina la respuesta HTTP asociada con la solicitud HTTP.
@NoneScoped	Bean vive con una sola evaluación EL. Se crea a partir de una evaluación EL y se destruye inmediatamente después de la evaluación EL.
@ViewScoped	Bean vive mientras el usuario interactúa con la misma vista JSF en la ventana / pestaña del navegador. Se crea a través de una solicitud HTTP y se destruye una vez que el usuario postback a una vista diferente.
@SessionScoped	Bean vive mientras dure la sesión HTTP. Se crea en la primera solicitud HTTP que involucra este bean en la sesión y se destruye cuando se invalida la sesión HTTP.
@ApplicationScoped	Bean vive mientras dure la aplicación web. Se crea a partir de la primera solicitud HTTP que involucra este bean en la aplicación (o cuando la aplicación web se inicia y el atributo eager = true se establece en @ManagedBean) y se destruye cuando se cierra la aplicación web.
@CustomScoped	Bean vive mientras la entrada del bean en el Custom Map que se crea para este ámbito vive.

Tabla 1: Ámbitos en el que se coloca el bean administrado.

Fuente: (“JSF - Managed Beans”, 2017)

➤ Backing Bean

Es usualmente un Bean común de java que sirve de soporte para un objeto manejado dentro de la aplicación, por ejemplo:

- (a) La página JSP está especificada como un ManagedBean para la aplicación, con un identificador que la describe para toda la aplicación en general.
- (b) En este archivo JSP se dibujan todos los controles necesarios para proveer a la página de funcionalidad.
- (c) Esta página tiene asociado un Backing Bean que es un Bean de Java. En este Bean se codifican los comportamientos específicos asociados a cada control del Managed Bean representado por la página JSP.

La ventaja de los Backing Beans es que pueden ser compartidos por un mismo Managed Bean, de manera que para diferentes páginas se pueden agrupar comportamientos comunes en un mismo Bean que se comparte con ambos.

➤ MVC y JSF

El framework JSF al igual que otros frameworks nos permite darle un formato o una arquitectura de capas denominada **Modelo, Vista, Controlador** a nuestro sistema.

- Las entidades se generan mediante JPA que nos permite el mapeo de objetos.
- La capa de modelo en la cual se crea los Manager y Manager DAO contienen los métodos genéricos.
- La siguiente capa contiene los controladores en los cuales se realizan las operaciones del negocio, es decir los procesos que va a contener la aplicación.
- La capa de la vista utiliza componentes de librerías compatibles con JSF, entre las cuales tenemos: Primefaces que nos proporciona una gama de componentes para realizar una completa plataforma gráfica y como lenguaje estándar tenemos

HTML5 que posee su componente Canvas que realiza las funciones de diseñador mediante coordenadas (puntos, líneas, gráficos) para tener un desarrollo satisfactorio del software.

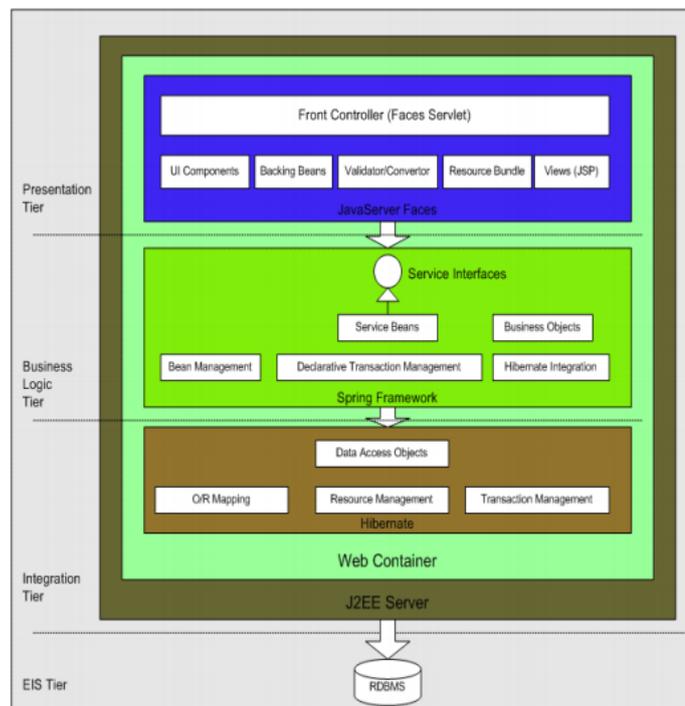


Ilustración 19: Arquitectura de JSF

Fuente: (nDeveloper Cía. Ltda., n.d.)

2.1.3.8. PRIMEFACES

Es una librería de componentes para Java Server Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes como por ejemplo RichFaces, ya que las dos bibliotecas son un conjunto de componentes ricos (editor de HTML, autocompletar, cartas, gráficas o paneles).

Además de que soporta Ajax basándose en el estándar JSF 2.0 Ajax API, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.

La principal característica de Primefaces, es que los diseños de sus componentes son realmente estables, y la complejidad para usarlos es insignificante, dando así un trabajo de calidad y sin mayor dificultad para el programador. (Caliskan, 2013)

También posee componentes para desarrollar aplicaciones web para dispositivos móviles, siendo un auge en la actualidad.



Ilustración 20: Logo de Primefaces

Fuente: (PrimeTek, 2009)



Ilustración 21: Temas y presentaciones de Primefaces

Fuente: (PrimeTek, 2009)

➤ **Características Primefaces:**

Primefaces es de origen turco, desarrollada por Prime Technology bajo la licencia de Apache License V2, al igual que presenta las siguientes características:

- **Soporte nativo de Ajax:** Se activa utilizando atributos específicos de cada uno de los componentes para lanzar un método del servidor y para indicar que componentes se van actualizar. Permite utilizar jQuery para los defectos visuales.
- **Kit de aplicaciones para móviles**
- **Compatible con RichFaces**
- **Open Source:** Cuenta con más de 100 componentes Opensource, como el Dock que es de muy alta complejidad y calidad.
- **Más de 25 temas:** Está integrado con ThemeRoller Framework CSS, el cual nos permite elegir 36 temas pre-diseñados o incluso diseñar un propio con la herramienta en línea utilizada para generar estos temas. Satisfaciendo los gustos del desarrollador en la presentación que desee darle al usuario.
- **Gran cantidad de componentes:** Es fácil y sencillo de implementar debido a que en su página oficial se presentan muchos ejemplos con su respectivo **código xhtml** y su **Bean** de Java para todos los desarrolladores tanto si son principiantes o ya tienen un nivel avanzado.

2.1.3.9. HTML 5 (CANVAS)

Canvas es un elemento o etiqueta incorporado en HTML que permite la generación de gráficos dinámicamente por medio del scripting sobre la estructura del documento.

“El código es más simple lo que permite hacer páginas más ligeras que se cargan más rápidamente favoreciendo la usabilidad y la indexación en buscadores. Ofrece una compatibilidad mayor con los navegadores de dispositivos móviles e incorpora nuevas capacidades JavaScript que aumentan la capacidad de almacenamiento”. (Arauz, Padrón, y otros, 2016).

En el cual es posible generar gráficos estáticos y animaciones donde una de las características importantes del objeto canvas es que puede ser accedido a través de JavaScript, permitiendo generar gráficos 2D, juegos, animaciones y composición de imágenes.

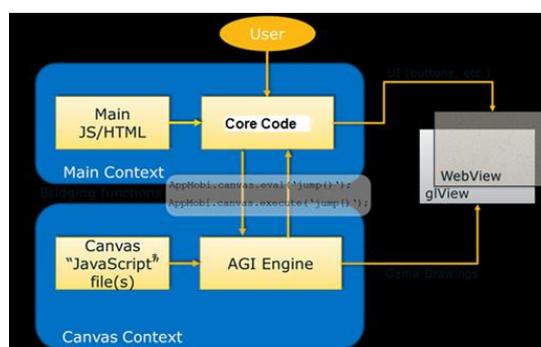


Ilustración 22: Arquitectura Canvas

Fuente: (Canvas, n.d.)

2.2. METODOLOGÍA DE DESARROLLO XP



Ilustración 23: Valores de la Metodología XP.

Fuente: (PMOinformatica.com, 2015)

La programación extrema o Extreme Programming es una metodología de desarrollo de la ingeniería de software.

Es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Uno de sus objetivos fundamentales estar en completa comunicación con el usuario de forma que se interprete todos los requerimientos de manera correcta, siendo esto un parámetro significativo para el éxito en el desarrollo del sistema.

Además, cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

De igual forma, la simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer, lo cual deduce que mientras más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

2.2.1. PROGRAMACIÓN EXTREMA

Conocido como método ágil en el desarrollo de la aplicación, es usado por programadores donde utilizan menor cantidad de procesos abiertos en un mismo tiempo; lo que quiere decir que se llevan los procesos de implementación, programación e implantación de forma rápida, inmediata donde se omiten documentación y procedimientos tradicionales que solía tomarse en cuenta antes del desarrollo del software.

Se basa en la interacción directa entre el usuario y el programador, de tal forma que permite satisfacer las necesidades por parte del usuario y el desarrollador cumple con exponer las limitaciones o los alcances que puede tener el sistema directamente con el/los involucrados.



Ilustración 24: Fases de la metodología XP

Fuente: (Instituto Tecnológico de Apizaco, Sánchez, 2013)

2.2.2. CICLO DE VIDA DE LA METODOLOGÍA XP

El ciclo de vida determina el éxito del software y para definirlo se debe aplicar los siguientes parámetros:

1. El cliente o usuario final determina el valor del negocio que se va a implementar.
2. El desarrollador se plantea un tiempo estimado en el cual realizará la implementación que requiere el usuario final.
3. El cliente expone sus necesidades en cuanto al sistema y a su tiempo en el que quiere hacer uso del mismo.
4. El desarrollador cumple con los requerimientos del sistema.
5. Se repite el ciclo desde el parámetro 1.

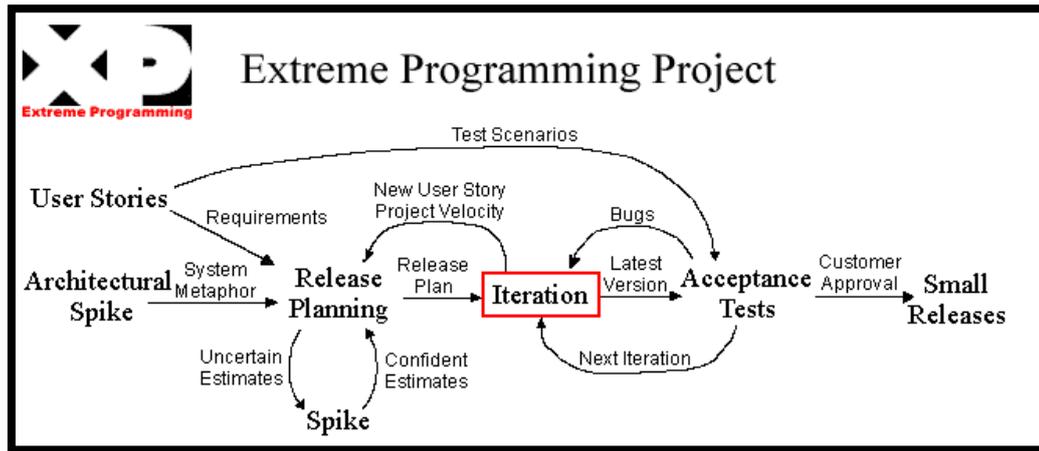


Ilustración 25: Ciclo de vida de la Metodología XP

Fuente: (Ciclo de vida de un proyecto XP, n.d.)

2.4.4. Tabla Comparativa

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos
Pocos Roles, más genéricos y flexibles	Más Roles, más específicos
No existe un contrato tradicional, debe ser bastante flexible	Existe un contrato prefijado
Cliente es parte del equipo de desarrollo (además in-situ)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Tabla 2: Diferencias entre metodologías ágiles y no ágiles.

Fuente: (Letelier, 2006)

CAPÍTULO III

3. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA UTILIZANDO LA METODOLOGÍA XP.

3.1.DEFINICIÓN DE APLICACIONES WEB

Antes de dar una definición de lo que tratan las aplicaciones web, es conveniente definir el término internet, el cual se considera “una gran red mundial de ordenadores conectados que, como todas las redes, permite compartir recursos e intercambiar información, donde los ordenadores están unidos a través de conexiones de varios tipos y para comunicarse utilizan un lenguaje o protocolo común, el TCP/IP”. (Martín, y otros, 2014, pág. 2).

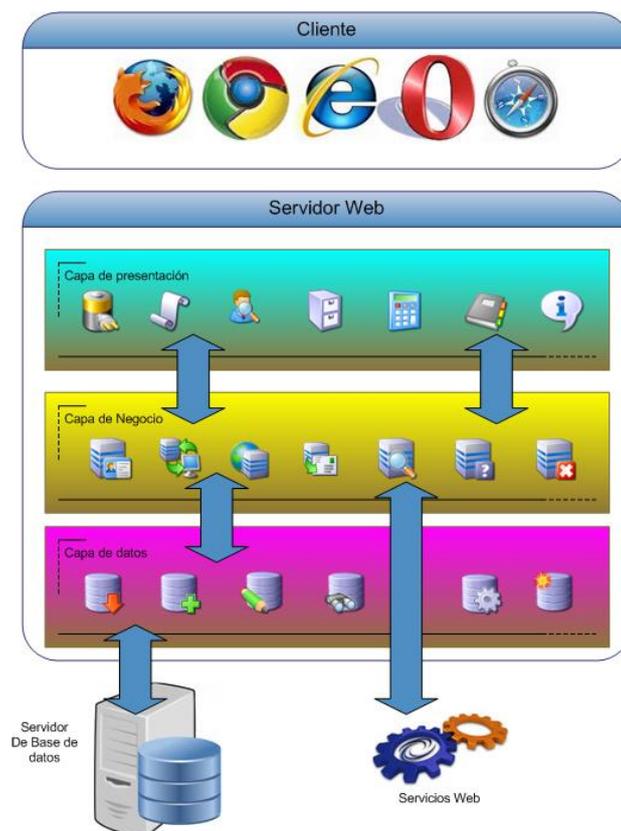


Ilustración 26: Arquitectura de Aplicaciones Web

Fuente: (Cobo, 2005)

WEB = INTERNET + HTTP

Ilustración 27: Definición de la Web

Fuente: (Martín, y otros, 2014)

Definiendo http como un protocolo que permite solicitar documentos que contengan hipertexto, como documentos en HTML (Lenguaje de Marcas de Hipertexto) que son interpretados por los navegadores como comandos de formateo de texto.

Como se observa en la **Ilustración 28**, la web atraviesa por varias etapas como son:

- **Web 1.0:** Contenido estático.
- **Web 2.0:** Contenido dinámico e interactivo.
- **Web 3.0:** Contenido colaborativo.
- **Web 4.0:** Sistema operativo establecido en la web tiene el protagonismo.

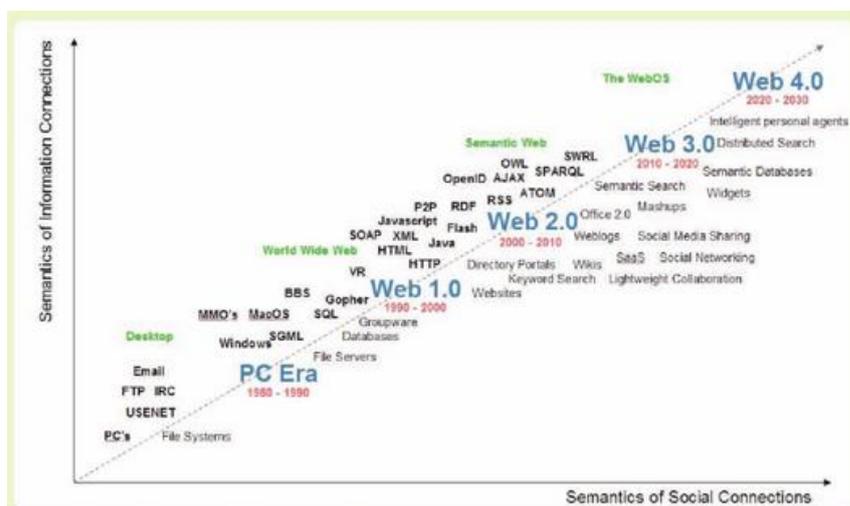


Ilustración 28: Evolución de la Web.

Fuente: (Martín, 2014)

Las **aplicaciones web** son sistemas o herramientas diseñadas para los usuarios o clientes que les facilita uno o más servicios para que ellos cumplan con sus tareas diarias de acuerdo a su área de trabajo accediendo a un servidor web a través del internet o intranet mediante un navegador.

Además, está considerado como el nivel tecnológico más avanzado debido a que varias entidades o usuarios normales pueden utilizar este servicio desde cualquier parte del mundo, mediante una dirección web como muestra la **Ilustración 29**, y que está compuesta por protocolo¹, nombre de dominio² y ruta del servidor.



Ilustración 29: Estructura de una dirección web

Fuente: (Martín, 2014)

3.2. PLANIFICACIÓN Y ESPECIFICACIONES

3.2.1 HISTORIAS DE USUARIO

La desarrolladora y analista del proyecto es: Daniela Evelyn Cabrera Pozo, teniendo como áreas de trabajo las siguientes actividades.

¹ **Protocolo:** Conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red.

² **Nombre de dominio:** es una dirección en la red en la que se ubica una página web determinada, este nombre debe ser único.

Tabla 3: Historia de Usuario 1

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 1	Nombre: Registro de datos del cliente.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Registrar los datos del cliente como número de cédula, nombres y apellidos completos, número telefónico, dirección y correo electrónico.</p>	

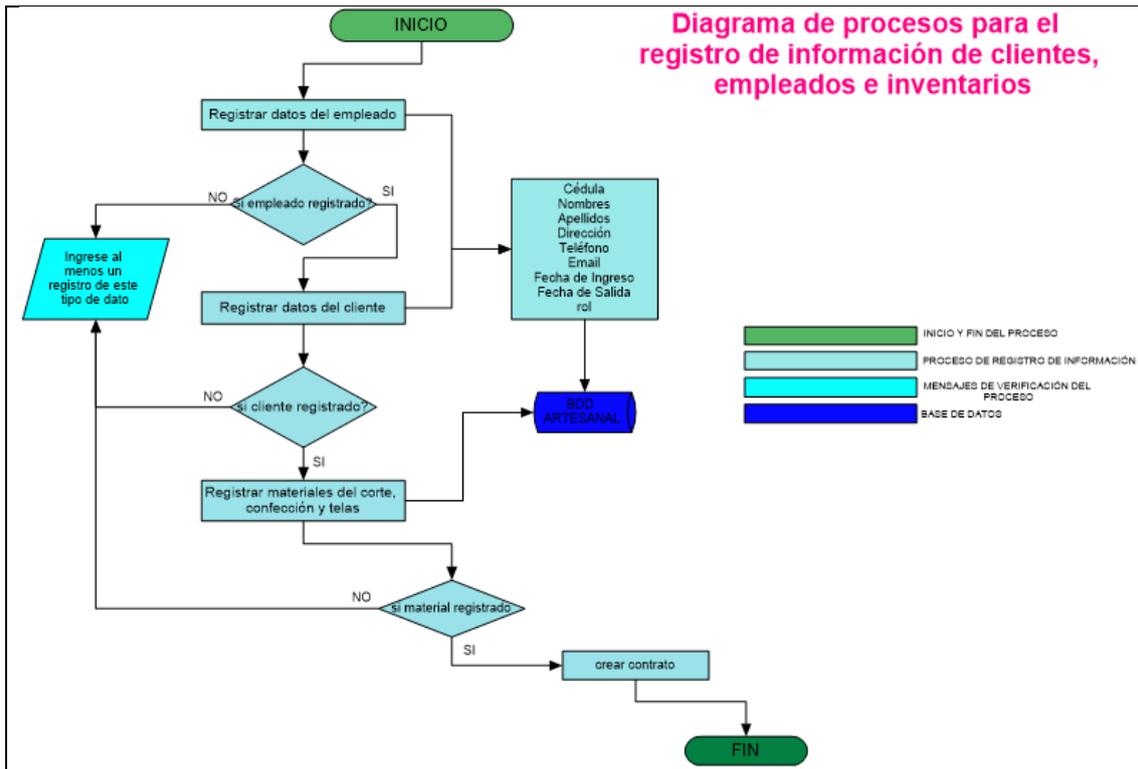


Ilustración 30: Diagrama de Procesos – Registro Clientes.

Fuente: Propia

Observaciones:

El sistema debe permitir que se ingrese la información específica del cliente correctamente.

Tabla 4: Historia de Usuario 2.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 2	Nombre: Registro de datos del empleado.
Usuario: Daniela Cabrera	

Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:

Descripción:

Registrar los datos del empleado como: número de cédula, nombres y apellidos completos, número telefónico, dirección y correo electrónico.

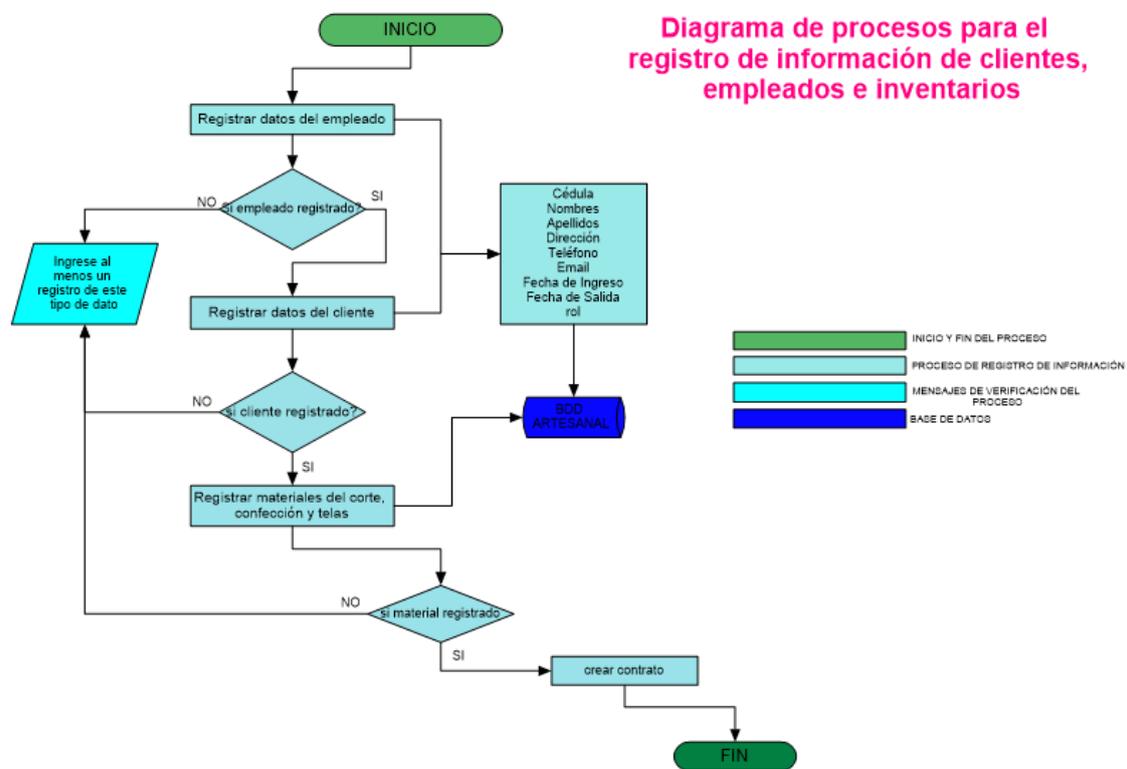


Ilustración 31: Diagrama de procesos- Registro empleados.

Fuente: Propia

Observaciones:

El sistema debe permitir que se ingrese la información del empleado correctamente.

Tabla 5: Historia de Usuario 3.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 3	Nombre: Administración de los datos del Cliente.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Permitir la creación, lectura, actualización o eliminación de los datos del cliente.</p>	
<p>Observaciones:</p> <p>El sistema debe permitir buscar al cliente o empleado por cualquier parámetro de ingresado ya sea la cédula o el nombre del cliente para cargar los datos y estos puedan ser modificados, eliminados o creados según sea el caso.</p>	

Tabla 6: Historia de Usuario 4.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 4	Nombre: Administración de los datos del empleado.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Permitir la creación, lectura, actualización o eliminación de los datos del empleado.</p>	
<p>Observaciones:</p> <p>El sistema debe permitir buscar al cliente o empleado por cualquier parámetro de ingresado ya sea la cédula o el nombre del empleado para cargar los datos y estos puedan ser modificados, eliminados o creados según sea el caso.</p>	

Tabla 7: Historia de Usuario 5.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 5	Nombre: Control de errores en el registro de información.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Se debe aplicar las respectivas validaciones a cada campo donde se registre información de tal forma que los datos que se ingresen sean correctos.</p>	
<p>Observaciones:</p> <p>El sistema debe procurar el ingreso de datos válidos del cliente y empleado, evitando se ingresen caracteres que no correspondan al registro que se solicita por el mismo</p>	

Tabla 8: Historia de Usuario 6.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 6	Nombre: Control de Inventario.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Llevar el registro de todos los materiales que se encuentran disponibles o están en stock.</p>	

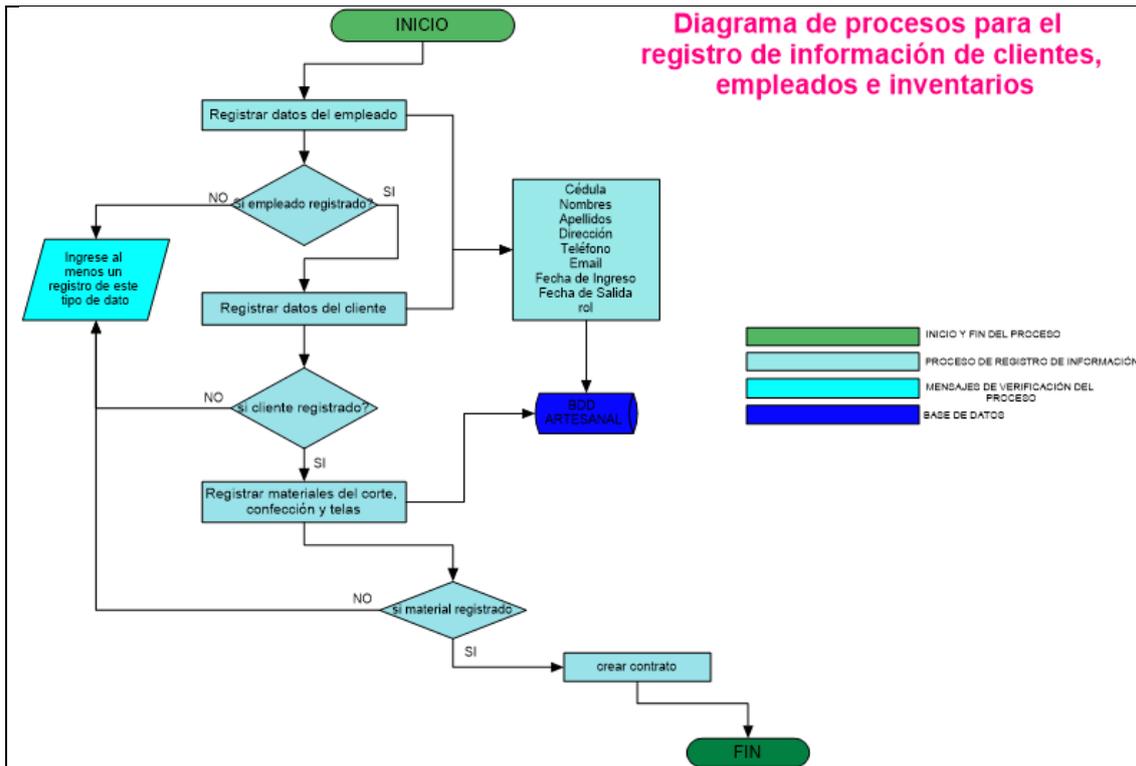


Ilustración 32: Diagrama de Procesos- Registro inventarios.

Fuente: Propia.

Observaciones:

El sistema debe presentar la información acerca de todos los materiales que se utilizan para fabricar el producto, de tal forma que se puedan analizar y conocer los materiales más vendidos.

Tabla 9: Historia de Usuario 7.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 7	Nombre: Presentación de diseños de prendas.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Pantalla que muestre a los clientes los diseños de ternos al estilo sastre para hombre (pantalón y saco) al igual que las respectivas medidas de cada prenda.</p>	

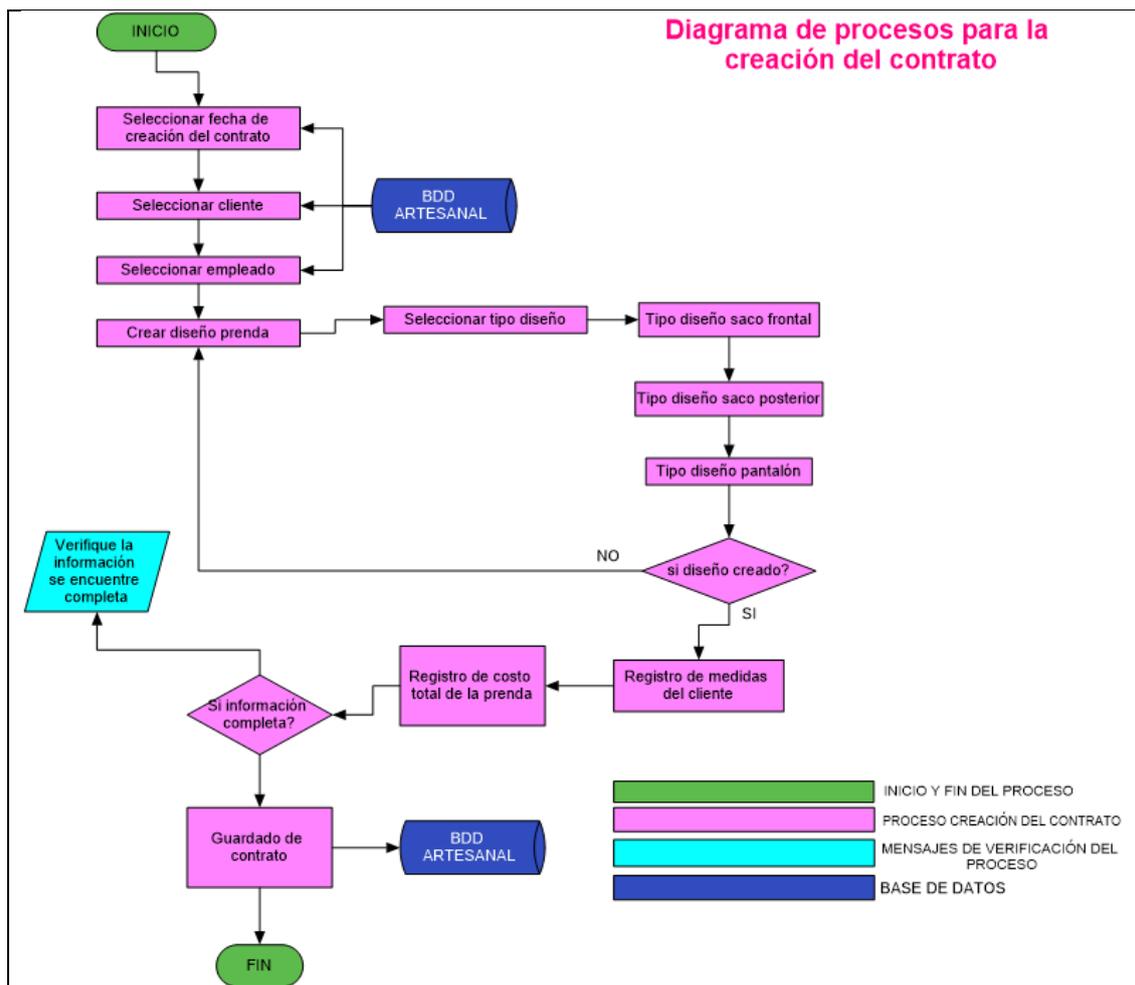


Ilustración 33: Diagrama de Procesos- Creación Contrato.

Fuente: Propia.

Observaciones:

El sistema debe presentar los diseños de tal manera que el cliente tenga la idea de cómo quedará el producto final y pueda identificar a tiempo los cambios que desea realizar antes de que los ternos sean confeccionados.

Tabla 10: Historia de Usuario 8.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 8	Nombre: Control de saldos de los clientes.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Llevar un registro de los pagos y deudas de los clientes acerca del contrato que realizan para la confección de las prendas.</p>	

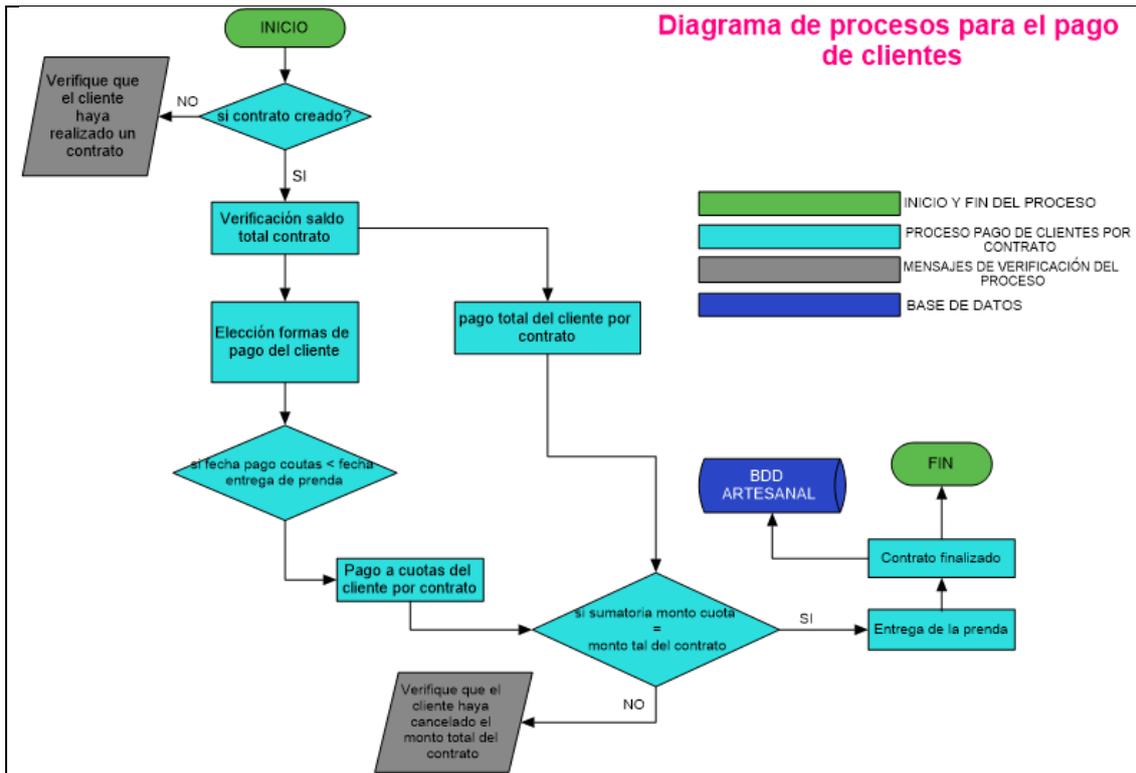


Ilustración 34: Diagrama Procesos- Pago a clientes.

Fuente: Propia.

Observaciones:

El sistema debe registrar la información en cuanto a los pagos de los clientes en cada contrato que realizar (pago en cuotas o pago total).

Tabla 11: Historia de Usuario 9.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 9	Nombre: Control de saldos de los empleados.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Llevar un registro de los pagos y deudas de empleados que confeccionan las prendas o solicitan anticipos.</p>	

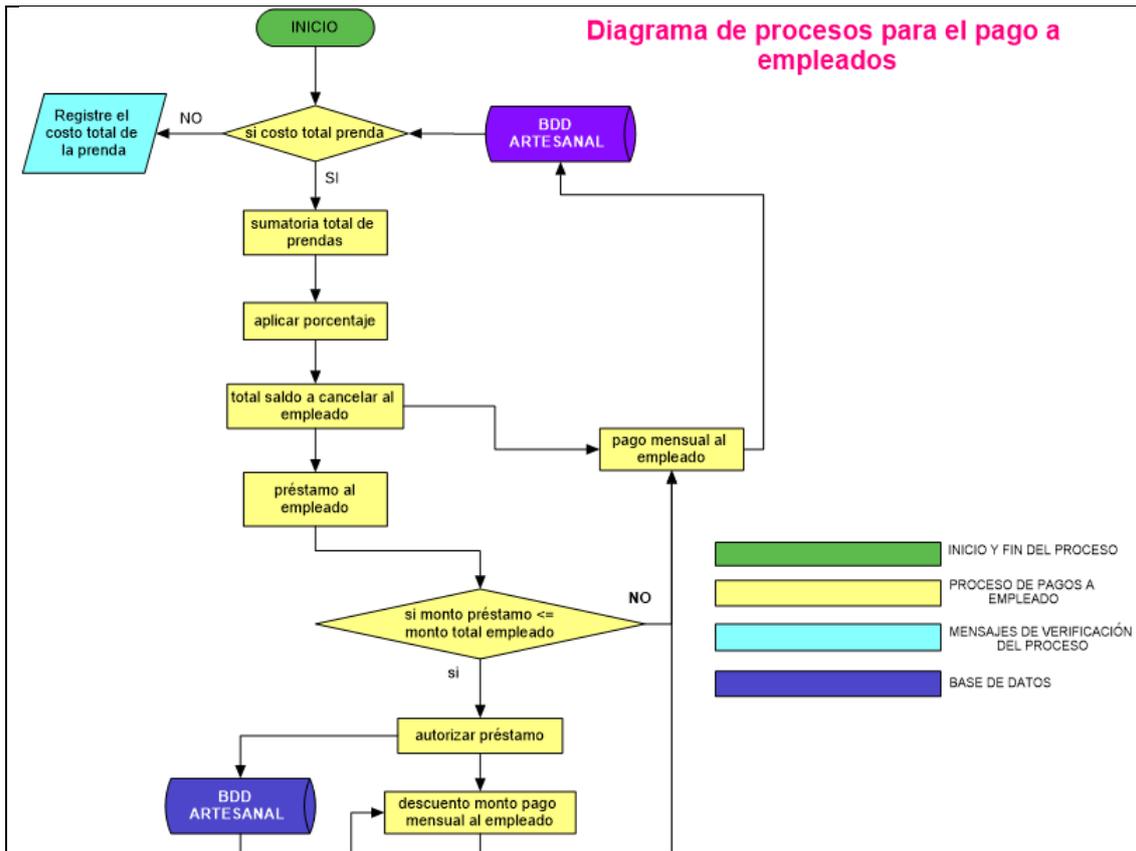


Ilustración 35: Diagrama de Procesos- Pago empleados.

Fuente: Propia.

Observaciones:

El sistema debe registrar la información en cuanto a los pagos y deudas que tiene cada empleado, con la finalidad de llevar un registro claro del manejo presupuestario.

Tabla 12: Historia de Usuario 10.

Fuente: Propia.

HISTORIA DE USUARIO	
Número: 10	Nombre: Generación de reportes.
Usuario: Daniela Cabrera	
Modificación de Historia Número:1	Iteración Asignada:1
Prioridad en Negocio: Alta	Puntos Estimados:
Riesgo de Desarrollo: Alto	Puntos Reales:
<p>Descripción:</p> <p>Presentación de reportes con información estructurada de los resultados del negocio.</p>	
<p>Observaciones:</p> <p>El sistema debe mostrar los resultados del proceso que lleva entre contratos y clientes para ser interpretados por el propietario del negocio para posible toma de decisiones o para tener constancia de los ingresos y egresos de la empresa.</p>	

3.2.2. VALORACIÓN DEL SOFTWARE

El software se encuentra valorado de acuerdo a los aspectos que se determinan en la siguiente tabla:

Tabla 13: Costo del proyecto.

Equipos			
Nro.	Descripción	Costo Estimado	Costo Real
1	Computador	1230,00	00,00
Recursos de oficina			
Nro.	Descripción	Costo Estimado	Costo Real
1	Resmas de papel	70,00	70,00
1	Impresora	120,00	0,00
6	Empastados	40,00	40,00
Gastos Personales			
Descripción		Costo Estimado	Costo Real
Movilización		120,00	0,00
Salario (8 meses de desarrollo del software)		172,50	0,00
		Total =1380,00	0,00
Material Bibliográfico			
Descripción		Costo Estimado	Costo Real
Libros		340,00	00,00
Internet		260,00	260,00
Artículos o Revistas		110,00	00,00
TOTAL		3 670,00	370,00

Fuente: Propia.

3.3. PROCESO DE DESARROLLO DE LA APLICACIÓN

3.3.1 CASOS DE USO

En la siguiente Ilustración 33, se muestran el diagrama de casos de uso de los involucrados donde se detallan cada una de las tareas que realizan dentro del sistema, para que este cumpla con todos los procesos de la lógica del negocio.

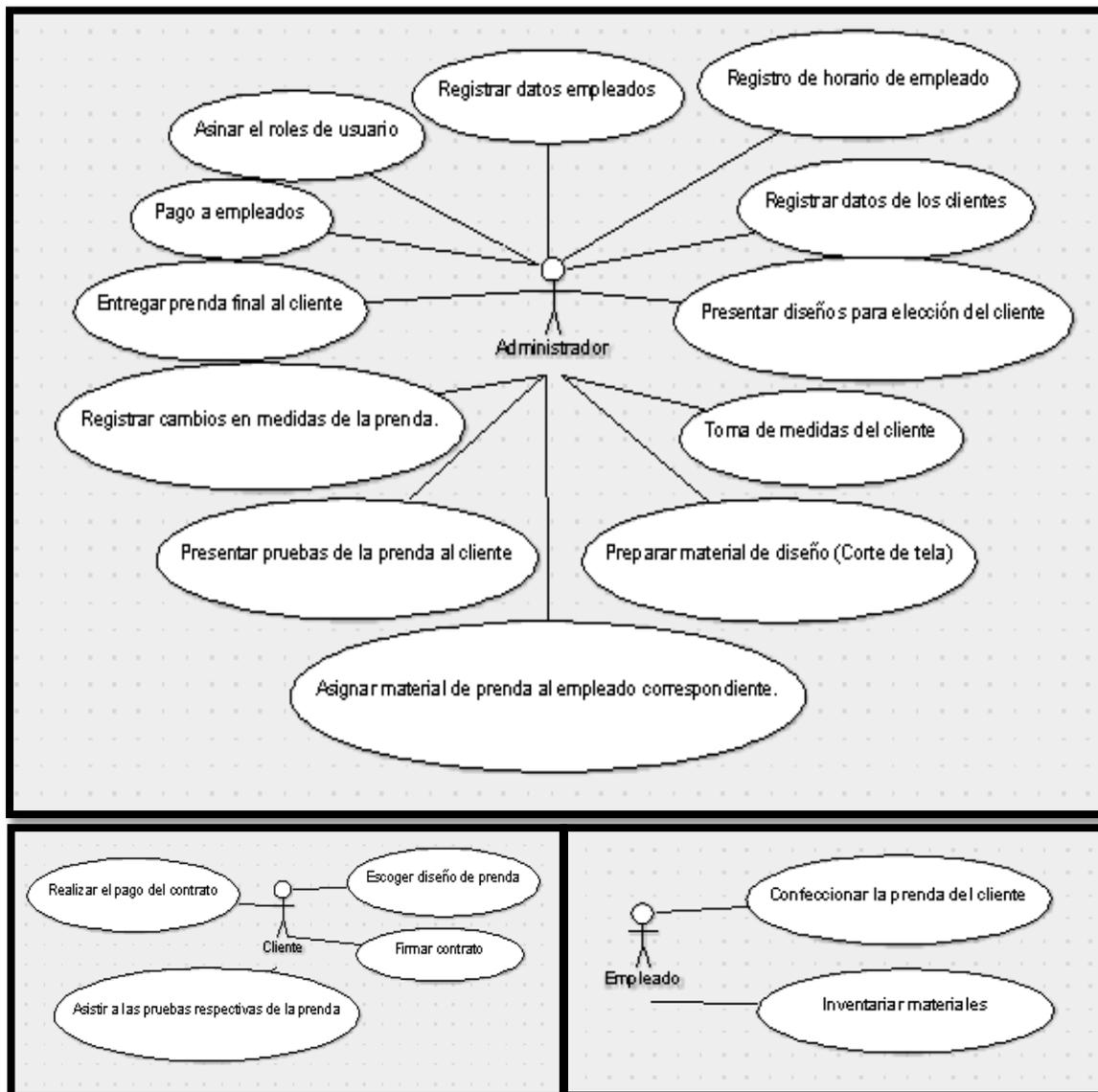


Ilustración 36: Diagrama de casos de uso

Fuente: Propia

3.3.2. DIAGRAMA DE BLOQUES

En la siguiente Ilustración 37, se muestra el diagrama de bloques del presente sistema, donde se definen que procesos se realizan en cada módulo como son Clientes, Empleados, Diseño e Insumos y Materiales.

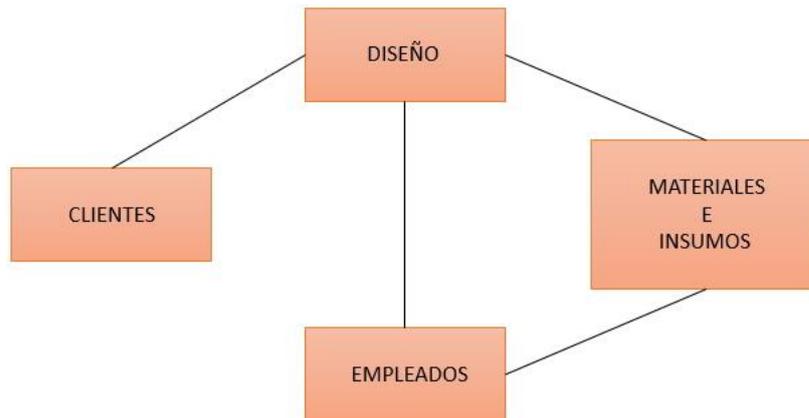


Ilustración 37: Diagrama de bloques

Fuente: Propia.

A continuación, se detallan los procesos que tiene cada módulo que intervienen en el desarrollo del sistema.

Sistema para la Microempresa Artesanal “SA”



Ilustración 38: Diagrama de bloques.

Fuente: Propia.

3.3.3 ITERACIONES

Tabla 14: Iteraciones por Actividades.

Fuente: Propia.

Nro.	ACTIVIDADES	TAREAS
1	Registro de datos del Cliente.	Diseñar la tabla para registrar los datos del cliente. Realizar la prueba del mapeo de la entidad desde la tabla del cliente para verificar la conexión entre la aplicación y la base de datos a trabajar. Realizar el prototipo de la interfaz de usuario para el registro del cliente. Presentar el prototipo creado al usuario para validar que el diseño cumpla con las necesidades del usuario. Dar funcionalidad a la interfaz de registro de los datos del cliente. Realizar pruebas de funcionalidad (programador). Realizar la presentación para pruebas de funcionalidad con el usuario final.
2	Registro de datos del Empleado.	Diseñar la tabla para registrar los datos del empleado. Realizar la prueba del mapeo de la entidad desde la tabla del cliente para verificar la conexión entre la aplicación y la base de datos a trabajar. Realizar el prototipo de la interfaz de usuario para el registro del empleado. Presentar el prototipo creado al usuario para validar que el diseño cumpla con las necesidades del usuario. Dar funcionalidad a la interfaz de registro de los datos del empleado. Realizar pruebas de funcionalidad (programador). Realizar la presentación para pruebas de funcionalidad con el usuario final.
3	Implementación de funciones para la administración de datos del Cliente.	Desarrollar el método para actualizar los datos del cliente. Desarrollar el método para la eliminación de datos del cliente. Realizar el prototipo de la nueva interfaz de usuario para la administración de los datos del cliente. Presentar al usuario el nuevo prototipo de para la administración de datos del cliente. Dar funcionalidad a la interfaz de acuerdo a las especificaciones del usuario para administrar la información del cliente. Realizar pruebas de funcionalidad (programador). Realizar la presentación para pruebas de funcionalidad con el usuario final.
4	Implementación de funciones para la administración de datos del Empleado.	Desarrollar el método para actualizar los datos del empleado. Desarrollar el método para la eliminación de datos del empleado. Realizar el prototipo de la nueva interfaz de usuario para la administración de los datos del empleado.

		Presentar al usuario el nuevo prototipo de para la administración de datos del empleado.
		Dar funcionalidad a la interfaz de acuerdo a las especificaciones del usuario para administrar la información del empleado.
		Realizar pruebas de funcionalidad (programador).
		Realizar la presentación para pruebas de funcionalidad con el usuario final.
5	Control de errores en el registro de información.	Validar interfaz de usuario para la administración de clientes.
		Validar interfaz de usuario para la administración de empleados.
		Validar la integridad y la veracidad de los datos del cliente en la base de datos.
		Validar la integridad y la veracidad de los datos del cliente en la base de datos.
6	Control de Inventario	Diseñar la tabla para registrar los datos correspondientes al inventario de los materiales empleados en el negocio.
		Mapear las clases correspondientes para continuar con el desarrollo de la aplicación.
		Verificar que la interfaz de usuario cumpla con los parámetros que se desea registrar.
		Realizar las pruebas de funcionalidad (programador).
		Realizar las pruebas de funcionalidad con el usuario.
7	Presentación de diseños de prendas.	Diseñar la tabla para almacenar los datos acerca a la confección de la prenda para el cliente.
		Mapear las clases correspondientes para continuar con el desarrollo del módulo de diseño.
		Realizar la interfaz de usuario del módulo de diseño de prendas del sistema.
		Dar funcionalidad al módulo de diseño mediante el desarrollo de método para administrar y guardar la información.
		Realizar las pruebas de funcionalidad (programador).
		Realizar las pruebas de funcionalidad con el usuario y presentación de la interfaz al mismo para verificar su validez y conformidad con los componentes implementados.
8	Control de saldos de los Clientes.	Diseñar las tablas para registrar los pagos y saldos pendientes de los clientes.
		Mapear las tablas para seguir con el desarrollo de los métodos necesarios para el registro de pagos y saldos adeudados por los clientes.
		Realizar la interfaz de usuario de los pagos de los clientes y del saldo pendiente por el contrato realizado.
		Dar funcionalidad a la interfaz de pagos de los clientes y sus saldos pendientes para llevar el registro de ésta información.
		Realizar las pruebas de funcionalidad de la interfaz (programador).
		Realizar las pruebas de funcionalidad con el usuario para determinar su validez y aprobación.
9	Control de saldos de los empleados.	Diseñar las tablas para registrar los pagos y saldos pendientes de los empleados.

		<p>Mapear las tablas para seguir con el desarrollo de los métodos necesarios para el registro de pagos y saldos adeudados a los empleados.</p> <p>Realizar la interfaz de usuario de los pagos de los clientes y del saldo pendiente por el contrato realizado.</p> <p>Dar funcionalidad a la interfaz de pagos a los empleados y sus saldos pendientes para llevar el registro de ésta información.</p> <p>Realizar las pruebas de funcionalidad de la interfaz (programador).</p> <p>Realizar las pruebas de funcionalidad con el usuario para determinar su validez y aprobación.</p>
10	Generación de reportes.	<p>Diseñar los reportes que muestren la información estructurada de los resultados del negocio.</p> <p>Realizar la implementación tecnológica para generar los reportes de tal manera que se muestren completos y amigables para el usuario del sistema.</p> <p>Verificar la validez de la información que se muestra en el reporte.</p> <p>Presentar los reportes al usuario para validar que la información que se visualiza sea la requerida por el mismo.</p>

Tabla 15: Iteraciones por Tareas.

Fuente: Propia.

Nro.	Fecha para desarrollar	TAREAS	Fecha para presentación cambios	Cambios generales por módulo
1	25/06/2015 - 15/07/2015	<p>Diseñar la tabla para registrar los datos del cliente.</p> <p>Realizar la prueba del mapeo de la entidad desde la tabla del cliente para verificar la conexión entre la aplicación y la base de datos a trabajar.</p> <p>Realizar el prototipo de la interfaz de usuario para el registro del cliente.</p> <p>Presentar el prototipo creado al usuario para validar que el diseño cumpla con las necesidades del usuario.</p> <p>Dar funcionalidad a la interfaz de registro de los datos del cliente.</p> <p>Realizar pruebas de funcionalidad (programador).</p> <p>Realizar la presentación para pruebas de funcionalidad con el usuario final.</p>	15/07/2015 - 30/07/2015	Permitir que la búsqueda de los clientes sea filtrada por todos los campos ingresados.
2	30/07/2015 - 15/09/2015	<p>Diseñar la tabla para registrar los datos del empleado.</p> <p>Realizar la prueba del mapeo de la entidad desde la tabla del cliente para verificar la conexión entre la aplicación y la base de datos a trabajar.</p> <p>Realizar el prototipo de la interfaz de usuario para el registro del empleado.</p>	17/09/2015 - 01/10/2015	Permitir que la búsqueda de los empleados sea filtrada por todos los campos ingresados.

		<p>Presentar el prototipo creado al usuario para validar que el diseño cumpla con las necesidades del usuario.</p> <p>Dar funcionalidad a la interfaz de registro de los datos del empleado.</p> <p>Realizar pruebas de funcionalidad (programador).</p> <p>Realizar la presentación para pruebas de funcionalidad con el usuario final.</p>		
3	02/10/2015 - 19/10/2015	<p>Desarrollar el método para actualizar los datos del cliente.</p> <p>Desarrollar el método para la eliminación de datos del cliente.</p> <p>Realizar el prototipo de la nueva interfaz de usuario para la administración de los datos del cliente.</p> <p>Presentar al usuario el nuevo prototipo de para la administración de datos del cliente.</p> <p>Dar funcionalidad a la interfaz de acuerdo a las especificaciones del usuario para administrar la información del cliente.</p> <p>Realizar pruebas de funcionalidad (programador).</p> <p>Realizar la presentación para pruebas de funcionalidad con el usuario final.</p>	20/10/2015 - 05/12/2015	Organizar los componentes de la interfaz para que sea amigable y de fácil de utilizar por el usuario.
4	01/01/2016 - 15/01/2016	<p>Desarrollar el método para actualizar los datos del empleado.</p> <p>Desarrollar el método para la eliminación de datos del empleado.</p> <p>Realizar el prototipo de la nueva interfaz de usuario para la administración de los datos del empleado.</p> <p>Presentar al usuario el nuevo prototipo de para la administración de datos del empleado.</p> <p>Dar funcionalidad a la interfaz de acuerdo a las especificaciones del usuario para administrar la información del empleado.</p> <p>Realizar pruebas de funcionalidad (programador).</p> <p>Realizar la presentación para pruebas de funcionalidad con el usuario final.</p>	16/01/2016 - 20/01/2016	Organizar los componentes de la interfaz para que sea amigable y de fácil de utilizar por el usuario.
5	20/01/2016 - 01/02/2016	<p>Validar interfaz de usuario para la administración de clientes.</p> <p>Validar interfaz de usuario para la administración de empleados.</p> <p>Validar la integridad y la veracidad de los datos del cliente en la base de datos.</p> <p>Validar la integridad y la veracidad de los datos del cliente en la base de datos.</p>	17/02/2016 - 31/02/2016	Implementar validaciones de acuerdo a la lógica del negocio.
6	01/03/2016 - 16/03/2016	<p>Diseñar la tabla para registrar los datos correspondientes al inventario de los materiales empleados en el negocio.</p>	16/03/2016 - 30/03/2016	Permitir el ingreso de información

		<p>Mapear las clases correspondientes para continuar con el desarrollo de la aplicación.</p> <p>Verificar que la interfaz de usuario cumpla con los parámetros que se desea registrar.</p> <p>Realizar las pruebas de funcionalidad (programador).</p> <p>Realizar las pruebas de funcionalidad con el usuario.</p>		complementaria, para ser tomada en cuenta como observaciones.
7	01/04/2016 - 18/04/2016	<p>Diseñar la tabla para almacenar los datos acerca a la confección de la prenda para el cliente.</p> <p>Mapear las clases correspondientes para continuar con el desarrollo del módulo de diseño.</p> <p>Realizar la interfaz de usuario del módulo de diseño de prendas del sistema.</p> <p>Dar funcionalidad al módulo de diseño mediante el desarrollo de método para administrar y guardar la información.</p> <p>Realizar las pruebas de funcionalidad (programador).</p> <p>Realizar las pruebas de funcionalidad con el usuario y presentación de la interfaz al mismo para verificar su validez y conformidad con los componentes implementados.</p>	19/04/2016 - 31/04/2016	Incluir en el módulo de diseño la toma de medidas con la finalidad de facilitar el proceso y manejo del sistema al usuario.
8	01/05/2016 - 15/05/2016	<p>Diseñar las tablas para registrar los pagos y saldos pendientes de los clientes.</p> <p>Mapear las tablas para seguir con el desarrollo de los métodos necesarios para el registro de pagos y saldos adeudados por los clientes.</p> <p>Realizar la interfaz de usuario de los pagos de los clientes y del saldo pendiente por el contrato realizado.</p> <p>Dar funcionalidad a la interfaz de pagos de los clientes y sus saldos pendientes para llevar el registro de ésta información.</p> <p>Realizar las pruebas de funcionalidad de la interfaz (programador).</p> <p>Realizar las pruebas de funcionalidad con el usuario para determinar su validez y aprobación.</p>	15/05/2016 - 31/05/2016	Permitir que se visualice el historial de los pagos y deudas realizado en un período específico.
9	01/06/2016 - 10/06/2016	<p>Diseñar las tablas para registrar los pagos y saldos pendientes de los empleados.</p> <p>Mapear las tablas para seguir con el desarrollo de los métodos necesarios para el registro de pagos y saldos adeudados a los empleados.</p> <p>Realizar la interfaz de usuario de los pagos de los clientes y del saldo pendiente por el contrato realizado.</p> <p>Dar funcionalidad a la interfaz de pagos a los empleados y sus saldos pendientes para llevar el registro de ésta información.</p> <p>Realizar las pruebas de funcionalidad de la interfaz (programador).</p>	10/06/2016 - 15/06/2016	Permitir que se visualice el historial de los pagos y deudas realizado en un período específico.

		Realizar las pruebas de funcionalidad con el usuario para determinar su validez y aprobación.		
10	15/06/2016 - 31/06/2016	Diseñar los reportes que muestren la información estructurada de los resultados del negocio. Realizar la implementación tecnológica para generar los reportes de tal manera que se muestren completos y amigables para el usuario del sistema. Verificar la validez de la información que se muestra en el reporte. Presentar los reportes al usuario para validar que la información que se visualiza sea la requerida por el mismo.	31/06/2016 - 12/07/2016	Incluir en el reporte cifras significativas que reflejen las ganancias del negocio.

3.3.3.1 WIREFRAMES (PROTOTIPOS DEL SISTEMA)

- El **login** del sistema permite registrar el usuario que ingresa al sistema y la contraseña con la que se encuentra registrado.

MICROEMPRESA ARTESANAL "SASTRERÍA ARTEAGA"

Ingresar al Sistema

Usuario:

Contraseña:

Ilustración 39: Login del Sistema- 1era Iteración

Fuente: Propia.

La siguiente vista corresponde a los parámetros que debe ingresar el usuario para **registrarse** y tener acceso al sistema.

Cédula:	<input type="text"/>
Nombres:	<input type="text"/>
Apellidos:	<input type="text"/>
Dirección:	<input type="text"/>
Teléfono:	<input type="text"/>
Email:	<input type="text"/>
Usuario:	<input type="text"/>
Clave:	<input type="text"/>
Vuelva a ingresar la clave:	<input type="text"/>
Rol:	Administrador ▼
<input type="button" value="Registrarse"/>	Loguearse

Ilustración 40: Registro de usuario- 1era Iteración.

Fuente: Propia.

A continuación, se muestra la pantalla de bienvenida del usuario logueado, donde puede acceder a los procesos con los que cuenta el sistema.

Bienvenido, User

Menú	Contrato	Saldos	Obras
Cambiar Contraseña	Realización Contrato	Pago Contrato	Horario Empleados
Roles de Usuarios	Reporte de contrato	Pago Empleados	Prendas Empleados
Cerrar Sesión		Saldo Adeudado Empleados	
		Saldo Adeudado Clientes	

Clientes Empleados Diseño Materiales e Insumos

Ilustración 41: Inicio del Sistema- Módulos- 2da Iteración.

Fuente: Propia.

- El primer prototipo del módulo de clientes, facilita el ingreso de la información y se muestran los parámetros que deben ser ingresados para el registro del cliente, como se observa en la **Ilustración 42**.

Administración de Clientes

Cédula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Email:

Mensaje
 Cliente ingresado correctamente

Ilustración 42: Administración de Clientes- 1era Iteración.

Fuente: Propia.

- A medida que se realizan avances del módulo de clientes, surgen formas para visualizar la información ingresada por lo que el prototipo inicial cambia para darle mayor funcionalidad para la satisfacción del cliente, como se muestran en las imágenes a continuación.

Administración de Clientes

Cédula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Email:

Mensaje
 Cliente ingresado correctamente

<input type="checkbox"/>	Cédula	Nombres	Apellidos	Dirección	Teléfono	Email	Editar	Eliminar
<input checked="" type="checkbox"/>	0401540919	Daniela Evelyn	Cabrera Pozo	calle Atahualpa	0968599871	evelyncabrera1991@gmail.com	.../	X
<input type="checkbox"/>	0401829577	Carlos Andrés	Bolaños Orbe	calle Bolívar	096618010	carlosandi@hotmail.com	.../	X

Ilustración 43: Administración de clientes- 2da Iteración.

Fuente: Propia.

Administración de Clientes

Cédula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Email:

Mensaje

Cliente editado correctamente

<input type="checkbox"/>	Cédula	Nombres	Apellidos	Dirección	Teléfono	Email	Editar	Eliminar
<input checked="" type="checkbox"/>	0401540919	Daniela Evelyn	Cabrera Pozo	calle Atahualpa	0968599871	evelyncabrera1991@gmail.com	.../	X
<input type="checkbox"/>	0401829577	Carlos Andrés	Bolaños Orbe	calle Bolívar	096618010	carlosandi@hotmail.com	.../	X

Ilustración 44: Administración Clientes - 3era Iteración

Fuente: Propia.

Editar cliente ✕

Cédula:	<input type="text" value="0401540901"/>
Nombres:	<input type="text" value="Dayerli"/>
Apellidos:	<input type="text" value="Andrade"/>
Dirección:	<input type="text" value="calle teodoro"/>
Teléfono:	<input type="text" value="0912938371"/>
Email:	<input type="text" value="dayerliandrade@hotmail.cc"/>

Ilustración 45: Edición Cliente - 4ta iteración.

Fuente: Propia.

- Como se observa en la siguiente **Ilustración 46**, el módulo de empleados también requiere ser diseñado de forma funcional antes de ser implementado y a su vez validado por el usuario del sistema.

Administración de Empleados

Cédula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Email:

Mensaje

Empleados ingresado correctamente

<input type="checkbox"/>	Cédula	Nombres	Apellidos	Dirección	Teléfono	Email	Editar	Eliminar
<input checked="" type="checkbox"/>	0401829783	Marilyn Carla	Benavides Ortiz	calle Eloy Alfaro	0983345985	marilynbenavides@gmail.com	...	X
<input type="checkbox"/>	0401829577	Sergio Armando	Tufiño Armas	calle Bolívar	096618010	sergiota123@hotmail.com	...	X

Ilustración 46: Administración empleados - 1era iteración.

Fuente: Propia.

Administración de Empleados

Cédula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Email:

Mensaje

Empleados editado correctamente

<input type="checkbox"/>	Cédula	Nombres	Apellidos	Dirección	Teléfono	Email	Editar	Eliminar
<input checked="" type="checkbox"/>	0401829783	Marilyn Carla	Benavides Ortiz	calle Eloy Alfaro	0983345985	marilynbenavides@gmail.com	...	X
<input type="checkbox"/>	0401829577	Sergio Armando	Tufiño Armas	calle Bolívar	096618010	sergiota123@hotmail.com	...	X

Ilustración 47: Administración empleados - 2da iteración.

Fuente: Propia.

- Inicialmente se muestra en la siguiente imagen, el diseño de las prendas a confeccionar correspondiente a saco y pantalón de ternos para hombre al estilo sastre, como una forma de ser visual para el usuario, a medida que se realizan las reuniones y se toma muy en cuenta la opinión del usuario, se realiza algunos cambios de tal forma que cumpla con las expectativas de las dos partes (usuario y programador).

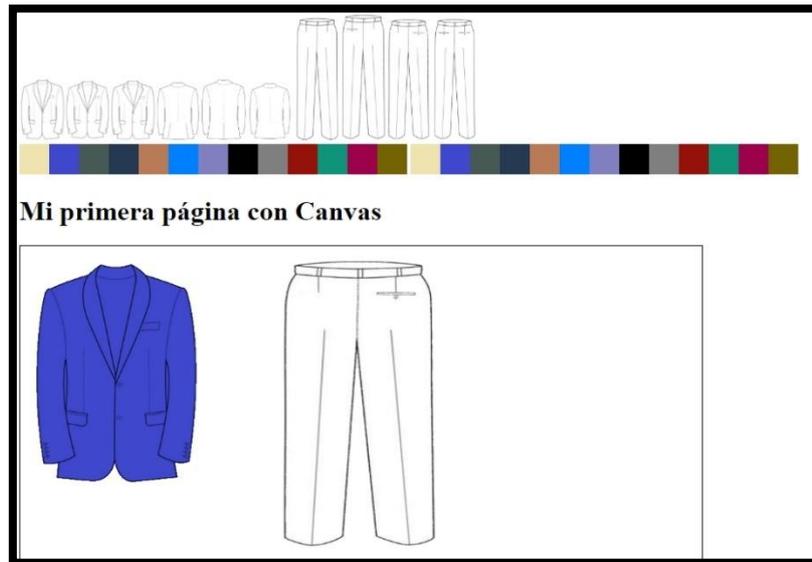


Ilustración 48: Diseño prendas - 1era Iteración.

Fuente: Propia.



Ilustración 49: Diseño prendas - 2da iteración.

Fuente: Propia.

Fecha:	<input type="text" value="02/06/16"/>				
Nombres y Apellidos:	<input type="text" value="Seleccione.."/>				
DISEÑO SACO FRONTAL		DISEÑO SACO POSTERIOR		DISEÑO PANTALÓN	
Color:	<input type="text" value="AZUL"/> <input type="checkbox"/>	Color:	<input type="text" value="ROJO"/> <input type="checkbox"/>	Color:	<input type="text" value="VERDE"/> <input type="checkbox"/>
Tipo:	<input type="text" value="Saco Cuello V"/>	Tipo:	<input type="text" value="Saco Un Partido"/>	Tipo:	<input type="text" value="Pantalon Con Bolsillo Izquier"/>
Tela:	<input type="text" value="Lana"/>	Tela:	<input type="text" value="Algodón"/>	Tela:	<input type="text" value="Algodón"/>
<input type="button" value="Diseñar"/>		<input type="button" value="Diseñar"/>		<input type="button" value="Diseñar"/>	

Ilustración 50: Elección de prendas - 1era iteración.

Fuente: Propia.

MEDIDAS PARA SACOS		Medidas para pantalón		Costo de la prenda	
Cuello:	<input type="text"/>	Largo total:	<input type="text"/>	Entrepierna:	<input type="text"/>
Espalda:	<input type="text"/>	Cintura:	<input type="text"/>	Cadera:	<input type="text"/>
Hombros:	<input type="text"/>	Muslo:	<input type="text"/>	Rodilla:	<input type="text"/>
Pecho:	<input type="text"/>	Basta:	<input type="text"/>	Precio saco:	<input type="text"/>
Mangas:	<input type="text"/>			Precio pantalón:	<input type="text"/>
Cintura:	<input type="text"/>			Precio Total Prenda:	<input type="text"/>

Ilustración 51: Registro de medidas de clientes - 1era iteración.

Fuente: Propia.

Fecha:	<input type="text" value="02/06/16"/>				
Nombres y Apellidos:	<input type="text" value="Seleccione.."/>				
DISEÑO SACO FRONTAL		DISEÑO SACO POSTERIOR		DISEÑO PANTALÓN	
Color:	<input type="text" value="AZUL"/> <input type="checkbox"/>	Color:	<input type="text" value="ROJO"/> <input type="checkbox"/>	Color:	<input type="text" value="VERDE"/> <input type="checkbox"/>
Tipo:	<input type="text" value="Saco Cuello V"/>	Tipo:	<input type="text" value="Saco Un Partido"/>	Tipo:	<input type="text" value="Pantalon Con Bolsillo Izquier"/>
Tela:	<input type="text" value="Lana"/>	Tela:	<input type="text" value="Algodón"/>	Tela:	<input type="text" value="Algodón"/>
<input type="button" value="Diseñar"/>		<input type="button" value="Diseñar"/>		<input type="button" value="Diseñar"/>	
MEDIDAS PARA SACOS		Medidas para pantalón		Costo de la prenda	
Cuello:	<input type="text"/>	Largo total:	<input type="text"/>	Entrepierna:	<input type="text"/>
Espalda:	<input type="text"/>	Cintura:	<input type="text"/>	Cadera:	<input type="text"/>
Hombros:	<input type="text"/>	Muslo:	<input type="text"/>	Rodilla:	<input type="text"/>
Pecho:	<input type="text"/>	Basta:	<input type="text"/>	Precio saco:	<input type="text"/>
Mangas:	<input type="text"/>			Precio pantalón:	<input type="text"/>
Cintura:	<input type="text"/>			Precio Total Prenda:	<input type="text"/>
<input type="button" value="✓ Guardar"/>					

Ilustración 52: Módulo diseño - 2da iteración.

Fuente: Propia.

- Finalmente, dentro de los prototipos iniciales, se muestra a continuación el módulo de registro de inventarios, el cual requiere la información que muestra en la **Ilustración 53**.

Registro de Inventarios

Fecha:

Empleado:

Tipo material:

Código material:

Cantidad:

Descripción:

<input type="checkbox"/>	Fecha	Empleado	Tipo material	Código material	Cantidad	Descripción	Editar	Eliminar
<input checked="" type="checkbox"/>	13/06/2016	Marilyn Carla Benavides Ortiz	Tela	Rusela	2m	Tela rusela confección sacos	.../	X

Ilustración 53: Registro inventarios - 3era iteración

Fuente: Propia.

CAPÍTULO IV

4. IMPLEMENTACIÓN DEL SISTEMA

4.1. FASE I: DISEÑO BASE DE DATOS

La base de datos cumple con el modelo entidad- relación, en la cual constan 25 tablas las cuales abarcan toda la información del negocio.

Esta realizado en la herramienta Dezin for Databases V6, que es un modelador de base de datos el cual genera el script de la misma para varios motores de base de datos en este caso PostgreSQL.

A continuación, se detalla el script de la base de datos del presente sistema donde se mencionan todas las tablas que intervienen en esta arquitectura.

```
/*---Creación de base de datos “bddArtesanal” del sistema---*/
```

```
CREATE DATABASE "bddArtesanal"
```

```
WITH OWNER = postgres
```

```
ENCODING = 'UTF8'
```

```
TABLESPACE = pg_default
```

```
LC_COLLATE = 'Spanish_Ecuador.1252'
```

```
LC_CTYPE = 'Spanish_Ecuador.1252'
```

```
CONNECTION LIMIT = -1;
```

```
/*---Tabla para registro de usuarios (administrador, empleados y clientes) --- */
```

```
CREATE TABLE tbl_usuarios (
```

```
usr_id integer NOT NULL,
```

```
usr_nombres character varying(50) NOT NULL,
```

```
usr_apellidos character varying(50) NOT NULL,
```

```
usr_direccion character varying(100),
```

```
usr_telefono character varying(20),
```

```
usr_email character varying(100),
```

```

usr_estado character varying(10) NOT NULL,

usr_fechaingreso timestamp without time zone NOT NULL,

usr_fechasalida timestamp without time zone,

rol_id integer NOT NULL,

usr_cedula character varying(15),

    CONSTRAINT PK_tbl_usuarios PRIMARY KEY (usr_id)
);

/*--Tabla para registro de roles de los usuarios del sistema--*/

CREATE TABLE tbl_rol (

rol_id integer NOT NULL,

rol_tipo character varying(20) NOT NULL,

rol_estado character varying(10),

    CONSTRAINT PK_tbl_rol PRIMARY KEY (rol_id)
);

/*-----Tabla para registro de prendas de los clientes-----*/

CREATE TABLE tbl_prendas (

prenda_id integer NOT NULL,

prenda_valor numeric,

usr_id integer NOT NULL,

prenda_estado character varying(10),

tm_id integer,

    CONSTRAINT PK_Prendas PRIMARY KEY (prenda_id)
);

/*-----Tabla para registro de deudas de los clientes-----*/

CREATE TABLE tbl_deudas_usr (

deu_usr_id integer NOT NULL,

deu_usr_fecha date NOT NULL,

deu_usr_valor numeric NOT NULL,

usr_id integer NOT NULL,

td_id_tipo integer NOT NULL,

```

```

deu_usr_estado character varying(10),

CONSTRAINT PK_tbl_deudas_usr PRIMARY KEY (deu_usr_id)

);

/*-----Tabla para registro de horarios de empleados-----*/

CREATE TABLE tbl_horario (

horario_id integer NOT NULL,

horario_horaentrada timestamp without time zone,

horario_horasalida timestamp without time zone,

usr_id integer NOT NULL,

horario_observaciones character varying(500),

horario_estado character varying(10),

CONSTRAINT PK_tbl_Horario PRIMARY KEY (horario_id)

);

/*-----Tabla para el registro de tipos de deudas -----*/

CREATE TABLE tbl_tipo_deuda (

td_id_tipo integer NOT NULL,

td_tipo_deuda character varying(20),

tp_estado character varying(10),

CONSTRAINT PK_tbl_tipo_deuda PRIMARY KEY (td_id_tipo)

);

/*-----Tabla para el registro de los diseños de ternos que eligen los clientes -----*/

CREATE TABLE tbl_disenio (

disenio_id integer NOT NULL,

color_id integer NOT NULL,

tela_id integer NOT NULL,

disenio_estado character varying(10),

tpd_id integer,

cont_id integer,

disenio_imagen bytea,

CONSTRAINT PK_tbl_disenio PRIMARY KEY (disenio_id)

```

);

/*---Tabla para el registro de medidas de los clientes -----*/

```
CREATE TABLE tbl_medidas (
  med_id integer NOT NULL,
  med_valor numeric,
  med_estado character varying(10),
  prenda_id integer,
  CONSTRAINT PK_tbl_medidas PRIMARY KEY (med_id)
```

);

/*-----Tabla para el registro de características de la tela que escoge el cliente -----*/

```
CREATE TABLE tbl_tela (
  tela_id integer NOT NULL,
  tela_tipo character varying(50),
  tela_estado character varying(10),
  CONSTRAINT PK_tbl_tela PRIMARY KEY (tela_id)
```

);

/*-----Tabla para el registro de tipos de diseño que propone la microempresa a sus clientes -----*/

```
CREATE TABLE tbl_tipo_disenio (
  tpd_id integer NOT NULL,
  tpd_tipo_disenio character varying(40),
  tpd_descripcion character varying(400),
  tpd_estado character varying(10),
  tp_id integer,
  CONSTRAINT PK_tbl_tipo_disenio PRIMARY KEY (tpd_id)
```

);

/*-----Tabla para el registro de materiales utilizados en la confección de prendas-----*/

```
CREATE TABLE tbl_materiales (
  mat_id integer NOT NULL,
```

```

mat_codigo character varying(20),
mat_descripcion character varying(150),
mat_cantidad numeric NOT NULL,
mat_observacion character varying(100),
tm_id integer NOT NULL,
mat_estado character varying(10),
mat_fecha date,
adm_id integer,
CONSTRAINT PK_tbl_materiales PRIMARY KEY (mat_id)
);

/*-----Tabla para el registro de tipos de materiales de corte y confección que utilizan los
empleados del área-----*/

CREATE TABLE tbl_tipo_material (
tm_id integer NOT NULL,
tm_tipo_material character varying(40) NOT NULL,
tm_estado character varying(10),
CONSTRAINT PK_tbl_tipo_material PRIMARY KEY (tm_id)
);

/*-----Tabla para el registro de colores que elige el cliente para confeccionar su prenda----
---*/

CREATE TABLE tbl_colores (
color_id integer NOT NULL,
color_nombre character varying(20),
color_estado character varying(10),
color_rgb character varying(20),
color_hex character varying(20),
CONSTRAINT PK_tbl_colores PRIMARY KEY (color_id)
);

/*-----Tabla para el registro de pagos de los usuarios-----*/

CREATE TABLE tbl_pago_usuario (
pge_id integer NOT NULL,

```

```

pge_fecha date,

pge_valor numeric,

usr_id integer NOT NULL,

tp_id integer NOT NULL,

pge_estado character varying(10),

CONSTRAINT PK_tbl_pago_usuario PRIMARY KEY (pge_id)

);

/*-----Tabla para el registro del contrato para la confección de su prenda-----*/

CREATE TABLE tbl_contrato (

    cont_id integer NOT NULL,

    cont_fecha date,

    cont_saldo_total numeric NOT NULL,

    cont_fecha_entrega date,

    cont_saldo_pagado numeric,

    cont_saldo_adeudado numeric,

    usr_id_emp integer NOT NULL,

    usr_id_cli integer NOT NULL,

    cont_estado character varying(10),

    CONSTRAINT PK_tbl_contrato PRIMARY KEY (cont_id)

);

/*-----Tabla para el registro de pagos por el contrato que realiza el cliente-----*/

CREATE TABLE tbl_pago_contrato (

    pcont_id integer NOT NULL,

    pcont_fecha date,

    pcont_valor numeric,

    tp_id integer NOT NULL,

    cont_id integer NOT NULL,

    pcont_estado character varying(10),

    CONSTRAINT PK_tbl_pago_contrato PRIMARY KEY (pcont_id)

);

```

/*-----Tabla para el registro de saldos pendientes del cliente en cada contrato -----*/

```
CREATE TABLE tbl_deudas_cont (
  dc_id integer NOT NULL,
  dc_fecha date,
  dc_valor numeric,
  cont_id integer NOT NULL,
  td_id_tipo integer NOT NULL,
  dc_estado character varying(10),
  CONSTRAINT PK_tbl_deudas_cont PRIMARY KEY (dc_id)
);
```

/*-----Tabla para el registro del tipo de pago que realizan clientes -----*/

```
CREATE TABLE tbl_tipo_pago (
  tp_id integer NOT NULL,
  tp_tipo_pago character varying(20),
  tp_estado character varying(10),
  CONSTRAINT PK_tbl_tipo_pago PRIMARY KEY (tp_id)
);
```

/*-----Tabla para el registro de las pruebas a las cuales asiste el cliente en la fecha establecida -----*/

```
CREATE TABLE tbl_pruebas (
  pb_id integer NOT NULL,
  pb_fecha date NOT NULL,
  pb_observaciones character varying(200) NOT NULL,
  usr_id integer NOT NULL,
  pb_estado character varying(10),
  CONSTRAINT PK_tbl_pruebas PRIMARY KEY (pb_id)
);
```

/*-----Tabla para el registro de medidas de las pruebas que se presenta el cliente-----*/

```
CREATE TABLE tbl_pruebas_medidas (
```

```

pm_id integer NOT NULL,

pm_observacion character varying(200),

pb_id integer NOT NULL,

med_id integer NOT NULL,

pm_estado character varying(10),

CONSTRAINT PK_tbl_pruebas_medidas PRIMARY KEY (pm_id)

);

/*-----Tabla para el registro de tipos de medida para especificar la parte del cuerpo humano
del que se extrajo la medida-----*/

CREATE TABLE tbl_tipo_medida (

tm_id integer NOT NULL,

tm_nombretipo character varying(200),

tm_estado character varying(10),

tp_id integer,

CONSTRAINT PK_tbl_tipo_medida PRIMARY KEY (tm_id)

);

/*-----Tabla para el registro de los usuarios que ingresan al sistema-----*/

CREATE TABLE tbl_administrador (

adm_id integer NOT NULL,

adm_usuario character varying(50),

adm_clave character varying(50),

adm_estado character varying(40),

usr_id integer NOT NULL,

CONSTRAINT PK_tbl_administrador PRIMARY KEY (adm_id)

);

/*-----Tabla para el registro del tipo de prenda que desea el cliente-----*/

CREATE TABLE tbl_tipo_prenda (

tp_id integer NOT NULL,

tp_descripcion character varying(500),

tp_estado character varying(40),

```

```

        CONSTRAINT PK_tbl_tipo_prenda PRIMARY KEY (tp_id)
    );

    /*-----Tabla para el registro de inventario de los materiales de confección que se utilizan
    en la microempresa artesanal -----*/

    CREATE TABLE tbl_inventario (

        inv_id SERIAL NOT NULL,

        inv_descripcion CHARACTER VARYING(150) ,

        inv_fecha TIMESTAMP ,

        usr_id INTEGER NOT NULL,

        inv_estado CHARACTER VARYING(20),

        CONSTRAINT PK_tbl_inventario PRIMARY KEY (inv_id)

    );

    /*-----Tabla para el registro del material que se desea inventariar (Materiales de corte,
    confección y telas) -----*/

    CREATE TABLE tbl_inventario_material (

        im_id SERIAL NOT NULL,

        im_cantidad CHARACTER VARYING(40) ,

        im_faltante CHARACTER VARYING(40) ,

        im_sobrante CHARACTER VARYING(40) ,

        mat_id INTEGER NOT NULL,

        inv_id INTEGER NOT NULL,

        im_estado CHARACTER VARYING(20),

        CONSTRAINT PK_tbl_inventario_material PRIMARY KEY (im_id)

    );

    /*-----Claves Foráneas-----*/

    ALTER TABLE ONLY tbl_tipo_disenio

        ADD CONSTRAINT "FK_tbl_tipo_disenio_prenda" FOREIGN KEY (tp_id) REFERENCES
tbl_tipo_prenda(tp_id);

    ALTER TABLE ONLY tbl_medidas

        ADD CONSTRAINT fk_tbl_medidas_tbl_prendas FOREIGN KEY (prenda_id)
REFERENCES tbl_prendas(prenda_id);

```

```
ALTER TABLE ONLY tbl_prendas

    ADD CONSTRAINT fk_tbl_prendas_tbl_tipo_medida FOREIGN KEY (tm_id)
REFERENCES tbl_tipo_medida(tm_id);

ALTER TABLE ONLY tbl_tipo_medida

    ADD CONSTRAINT fk_tbl_tipo_medida_tbl_tipo_prenda FOREIGN KEY (tp_id)
REFERENCES tbl_tipo_prenda(tp_id);

ALTER TABLE ONLY tbl_materiales

    ADD CONSTRAINT tbl_administrador_tbl_materiales FOREIGN KEY (adm_id)
REFERENCES tbl_administrador(adm_id);

ALTER TABLE ONLY tbl_disenio

    ADD CONSTRAINT tbl_colores_tbl_disenio FOREIGN KEY (color_id) REFERENCES
tbl_colores(color_id);

ALTER TABLE ONLY tbl_deudas_cont

    ADD CONSTRAINT tbl_contrato_tbl_deudas_cont FOREIGN KEY (cont_id) REFERENCES
tbl_contrato(cont_id);

ALTER TABLE ONLY tbl_disenio

    ADD CONSTRAINT tbl_contrato_tbl_disenio FOREIGN KEY (cont_id) REFERENCES
tbl_contrato(cont_id);

ALTER TABLE ONLY tbl_pago_contrato

    ADD CONSTRAINT tbl_contrato_tbl_pago_contrato FOREIGN KEY (cont_id)
REFERENCES tbl_contrato(cont_id);

ALTER TABLE ONLY tbl_inventario_material

    ADD CONSTRAINT tbl_materiales_tbl_inventario_material FOREIGN KEY (mat_id)
REFERENCES tbl_materiales(mat_id);

ALTER TABLE ONLY tbl_pruebas_medidas

    ADD CONSTRAINT tbl_medidas_tbl_pruebas_medidas FOREIGN KEY (med_id)
REFERENCES tbl_medidas(med_id);

ALTER TABLE ONLY tbl_pruebas_medidas

    ADD CONSTRAINT tbl_pruebas_tbl_pruebas_medidas FOREIGN KEY (pb_id)
REFERENCES tbl_pruebas(pb_id);

ALTER TABLE ONLY tbl_usuarios
```

```
ADD CONSTRAINT tbl_rol_tbl_usuarios FOREIGN KEY (rol_id) REFERENCES
tbl_rol(rol_id);
```

```
ALTER TABLE ONLY tbl_disenio
```

```
ADD CONSTRAINT tbl_tela_tbl_disenio FOREIGN KEY (tela_id) REFERENCES
tbl_tela(tela_id);
```

```
ALTER TABLE ONLY tbl_deudas_cont
```

```
ADD CONSTRAINT tbl_tipo_deuda_tbl_deudas_cont FOREIGN KEY (td_id_tipo)
REFERENCES tbl_tipo_deuda(td_id_tipo);
```

```
ALTER TABLE ONLY tbl_deudas_usr
```

```
ADD CONSTRAINT tbl_tipo_deuda_tbl_deudas_usr FOREIGN KEY (td_id_tipo)
REFERENCES tbl_tipo_deuda(td_id_tipo);
```

```
ALTER TABLE ONLY tbl_disenio
```

```
ADD CONSTRAINT tbl_tipo_disenio_tbl_disenio FOREIGN KEY (tpd_id) REFERENCES
tbl_tipo_disenio(tpd_id);
```

```
ALTER TABLE ONLY tbl_materiales
```

```
ADD CONSTRAINT tbl_tipo_material_tbl_materiales FOREIGN KEY (tm_id)
REFERENCES tbl_tipo_material(tm_id);
```

```
ALTER TABLE ONLY tbl_pago_contrato
```

```
ADD CONSTRAINT tbl_tipo_pago_tbl_pago_contrato FOREIGN KEY (tp_id)
REFERENCES tbl_tipo_pago(tp_id);
```

```
ALTER TABLE ONLY tbl_pago_usuario
```

```
ADD CONSTRAINT tbl_tipo_pago_tbl_pago_usuario FOREIGN KEY (tp_id)
REFERENCES tbl_tipo_pago(tp_id);
```

```
ALTER TABLE ONLY tbl_contrato
```

```
ADD CONSTRAINT "tbl_user_tbl_contCliente" FOREIGN KEY (usr_id_cli) REFERENCES
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_contrato
```

```
ADD CONSTRAINT "tbl_user_tbl_contEmpleado" FOREIGN KEY (usr_id_emp)
REFERENCES tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_prendas
```

```
ADD CONSTRAINT tbl_usuarios_prendas FOREIGN KEY (usr_id) REFERENCES
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_administrador
```

```
ADD CONSTRAINT tbl_usuarios_tbl_administrador FOREIGN KEY (usr_id) REFERENCES  
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_deudas_usr
```

```
ADD CONSTRAINT tbl_usuarios_tbl_deudas_usr FOREIGN KEY (usr_id) REFERENCES  
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_horario
```

```
ADD CONSTRAINT tbl_usuarios_tbl_horario FOREIGN KEY (usr_id) REFERENCES  
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_pago_usuario
```

```
ADD CONSTRAINT tbl_usuarios_tbl_pago_usuario FOREIGN KEY (usr_id) REFERENCES  
tbl_usuarios(usr_id);
```

```
ALTER TABLE ONLY tbl_pruebas
```

```
ADD CONSTRAINT tbl_usuarios_tbl_pruebas FOREIGN KEY (usr_id) REFERENCES  
tbl_usuarios(usr_id);
```

- El script anterior fue obtenido de acuerdo al siguiente modelo:

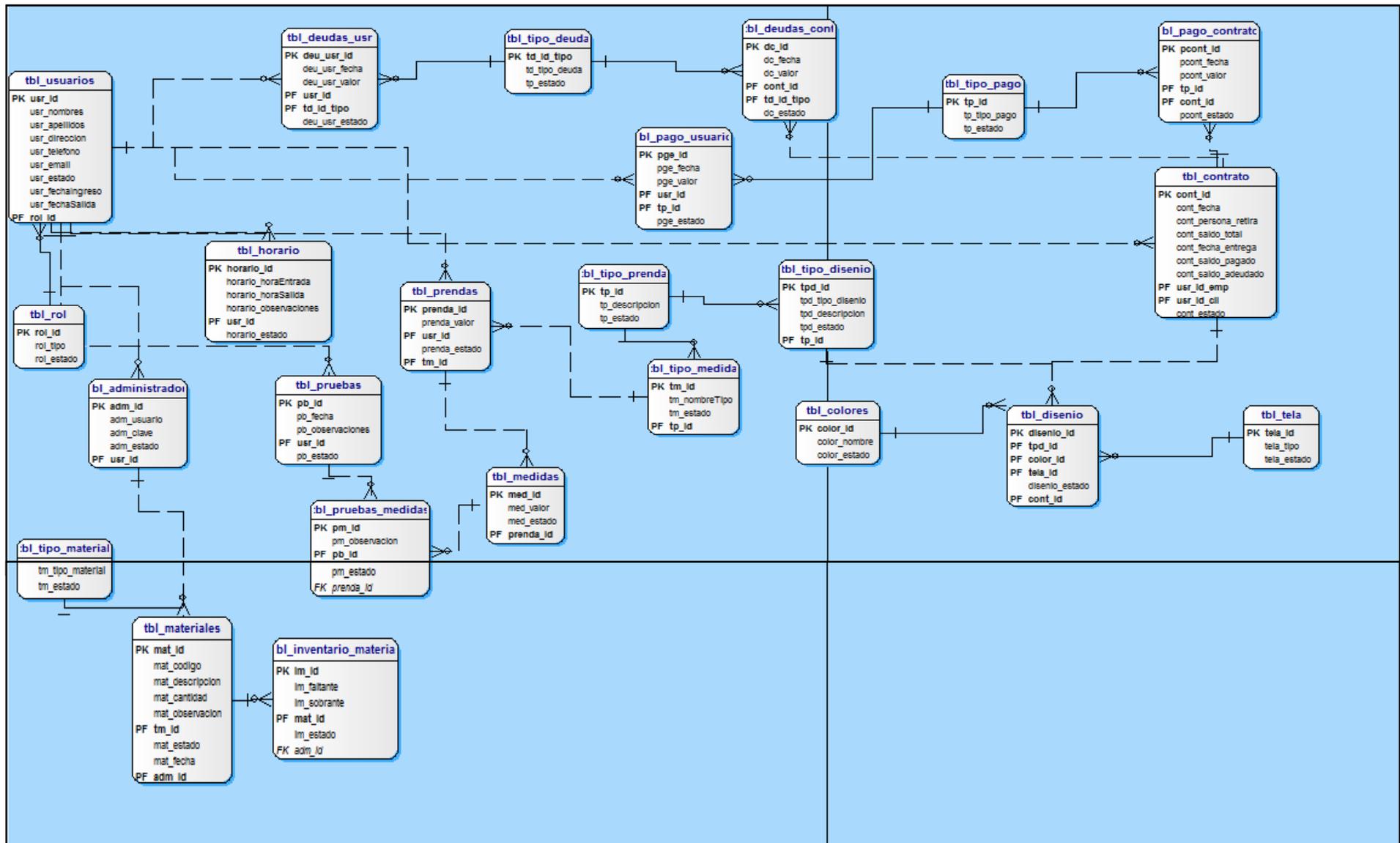


Ilustración 54: Esquema de base de datos del sistema.

Fuente: Propia.

4.2. FASE II: CREACIÓN DEL PROYECTO

Una vez generado el script de base de datos, se creó el proyecto en el IDE de desarrollo, en este caso Eclipse, con el formato de JSF (Java Server Faces), mismo que contiene cuatro capas (paquetes) para desarrollar el sistema:

- **art.model.dao.entities:** este contiene las tablas creadas en de la base de datos mapeadas, accesibles como objetos en el lenguaje en el que se desarrolla la aplicación.
- **art.model.dao.manager:** este paquete contiene los métodos genéricos, conocidos como CRUD (Create, Read, Update, Delete).
- **art.model.dao.manager:** en este paquete se encuentran las clases con los métodos respectivos de la lógica del negocio, es decir, contiene todos los procesos que debe cumplir el sistema.
- **art.model.controller:** contiene los **Session Bean** que son clases que permiten que se comunique las vistas con los métodos de la lógica del negocio.

Para mayor comprensión del lector en cuanto al formato que sigue el framework JSF en la arquitectura del sistema, se expone la definición de JSF en el capítulo II, y se refuerza con la siguiente **Ilustración 55**.

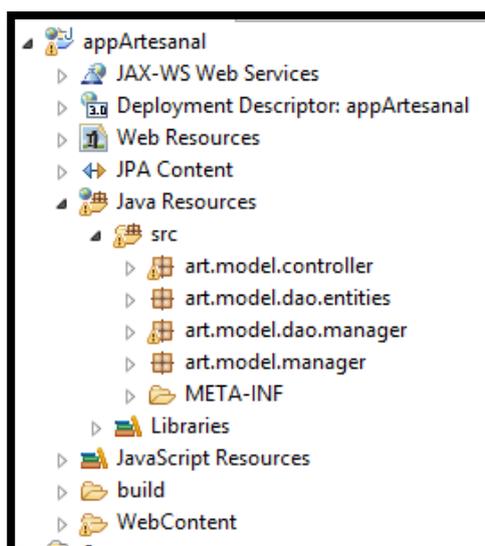


Ilustración 55: Estructura Framework JSF

Fuente: Propia.

4.3. FASE III: INTERFAZ DE USUARIO

En esta fase se proporciona la descripción precisa de la funcionalidad que tiene cada una de las pantallas de interfaz de usuario.

- En la siguiente Ilustración 56, se muestran la pantalla principal del sistema, la cual permite que el usuario que va a hacer uso del mismo, ingrese su clave y su contraseña, en este caso es el usuario **Administrador**.

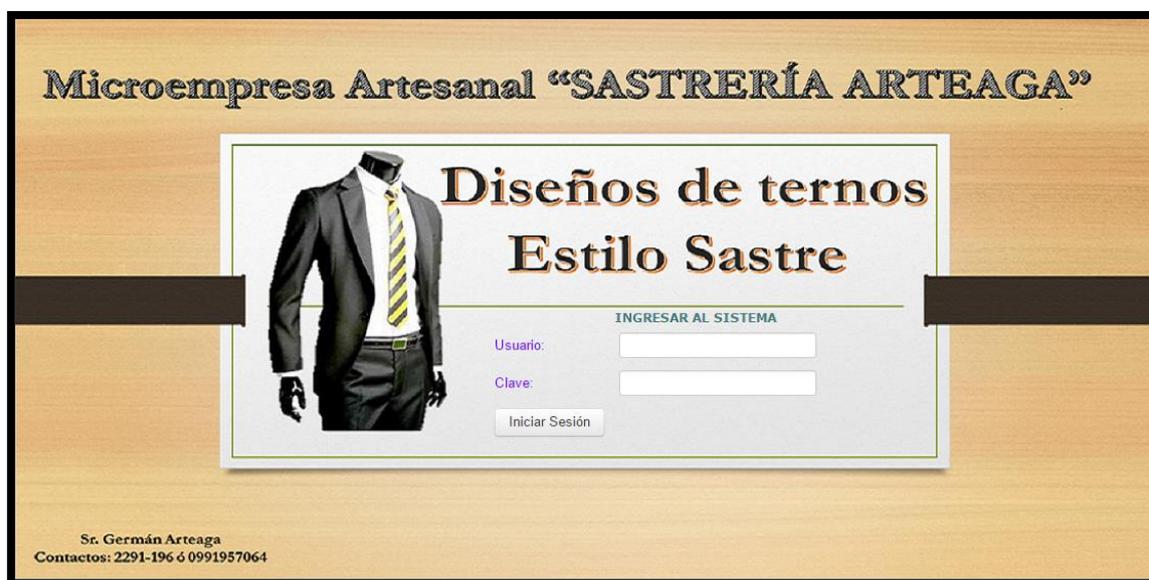


Ilustración 56: Login del Sistema.

Fuente: Propia

- Una vez que el usuario ingresa al sistema se muestra la pantalla de bienvenida al sistema, donde se observa el menú con los procesos que contiene el software y me permite realizar el proceso completo que lleva la Microempresa Artesanal "SASTRERÍA ARTEAGA".



Ilustración 57: Pantalla de inicio del sistema.

Fuente: Propia

- La siguiente **Ilustración 58**, nos muestra el proceso de creación del contrato, donde se registra el cliente, la fecha de creación del contrato, la fecha de entrega de la prenda, el empleado que la va a realizar y los diseños seleccionados al gusto del cliente.

Ilustración 58: Creación del Contrato.

Fuente: Propia

Administración - Actualizar Registros - Contrato - Inventario - Saldos - Configuración Cuenta - BIENVENIDO, gaby

Fecha: 27/12/2016
 Cliente: Eduardo Benito Lourdes
 Obra asignada a: Andrea Cevallos
 Fecha de Entrega: 27/12/2016

DISEÑO SACO FRONTAL
 Color: ROJO
 Tipo: Saco Cuello Redondo
 Tela: Algodón
 Diseñar

DISEÑO SACO POSTERIOR
 Color: ROJO
 Tipo: Saco Doble Partido
 Tela: Algodón
 Diseñar

DISEÑO PANTALÓN
 Color: VERDE
 Tipo: Pantalón Sin Bolsillos
 Tela: Algodón
 Diseñar

Costo de la prenda

Precio saco:	150
Precio pantalón:	100
Precio Total Prenda:	250.0
Precio Pagado Prenda:	100
Precio Adeudado Prenda:	150.0

Guardar

Ilustración 59: Creación contrato versión final.

Fuente: Propia.

- Una vez seleccionado el diseño, se registra las medidas de la prenda, correspondientes a cada tipo de prenda (saco frontal, posterior y pantalón).

MEDIDAS PARA SACOS

Cuello:	<input type="text"/>	Espalda:	<input type="text"/>
Hombros:	<input type="text"/>	Pecho:	<input type="text"/>
Mangas:	<input type="text"/>	Cintura:	<input type="text"/>

✓ Guardar

Ilustración 60: Registro de las medidas de la prenda.

Fuente: Propia

- Una vez obtenido la información de medidas y diseño de la prenda, se procede a valorizar la confección de la misma, como se muestra en la **Ilustración 61**.

Costo de la prenda	
Precio saco:	0,0
Precio pantalón:	0,0
Precio Total Prenda:	0,00
Precio Pagado Prenda:	0,0
Precio Adeudado Prenda:	0,0

Ilustración 61: Costo de la prenda.

Fuente: Propia

- Una vez registrado la información correspondiente se procede a guardar; para verificar que los contratos estén registrados correctamente se muestra la siguiente pantalla (**Ilustración 62**) con la lista de contratos existentes.

LISTA DE CONTRATOS					
Fecha	Empleado	Cliente	Fecha de Entrega	Saldo total	Ver Reporte
2016-06-29	Gabriela Estefania Benavides Hernández	Andrea Cevallos	2016-07-18	123.45	Ver Reporte
2016-07-05	Dayefi Andrade	Andrea Cevallos	2016-07-15	14.34	Ver Reporte

Ilustración 62: Lista de contratos realizados.

Fuente: Propia

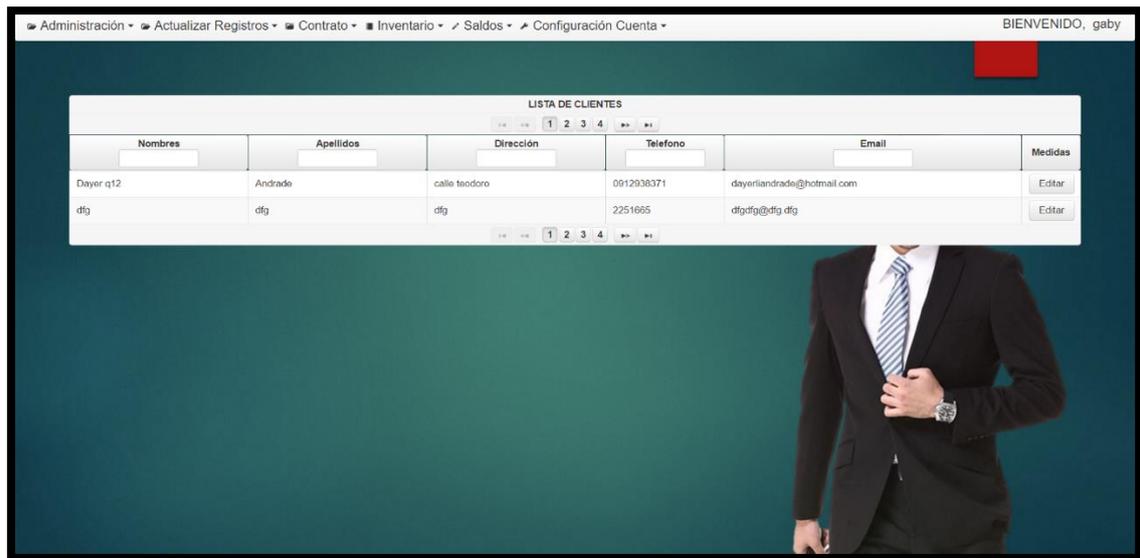


Ilustración 63: Ventana para la toma de medidas de los clientes.

Fuente: Propia.

Saco Frontal	
Cuello	50
Pecho	90
Cintura	60
Saco Posterior	
Hombros	75
Mangas	25
Espalda	35
Pantalon	
Basta	23
Muslo	30
Cintura	60
Largo Total	64
Entrepierna	35
Cadera	87
Rodilla	15
Guardar	

Ilustración 64: Ventana para el ingreso de medidas de los clientes.

Fuente: Propia.

Fecha	Empleado	Cliente	Fecha de Entrega	Saldo total	Saldo Adeudado	Pagos
2016-12-27	Andrea Cevallos	Marcelo Bolaños	2016-12-27	37.00	33.00	Pagar
2016-12-27	qwe11 qwe	qweqwe qweqwe	2016-12-27	37.00	32.00	Pagar

Ilustración 65: Lista de contratos a cancelar.

Fuente: Propia.

Pago del Contrato ✕

Tipo de Pago	Seleccione.. ▾
Valor Pago:	0
Valor Pendiente:	33.00
Pagar	

Ilustración 66: Pago de Clientes por Contrato.

Fuente: Propia.

- Seguido se presenta la sección de reportes, y en este caso se muestra el reporte de la creación de los contratos donde se indica el cliente que lo realizó, la fecha y los diseños seleccionados por el mismo.

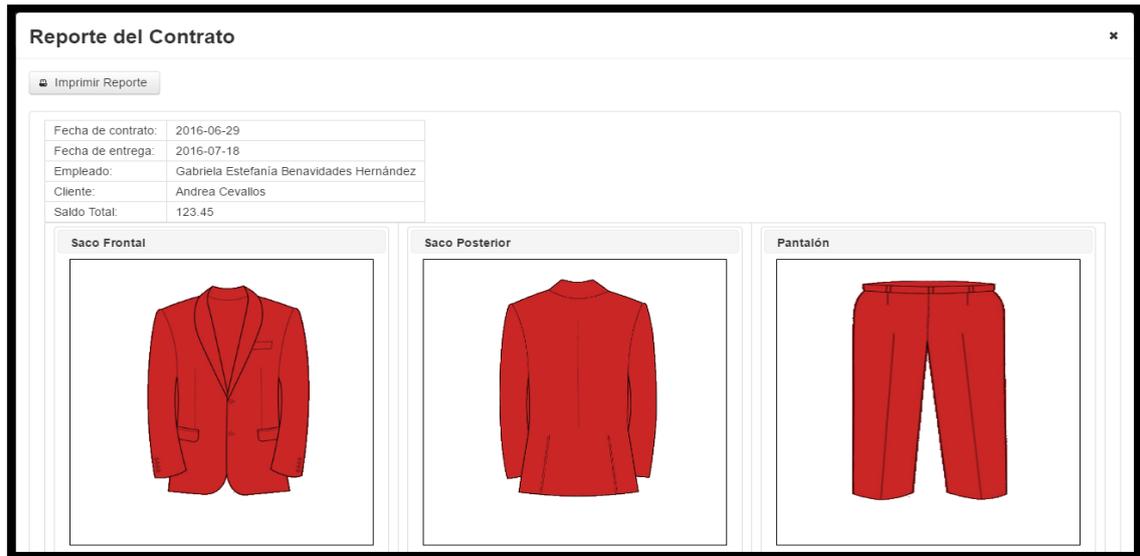


Ilustración 67: Reporte del contrato creado.

Fuente: Propia

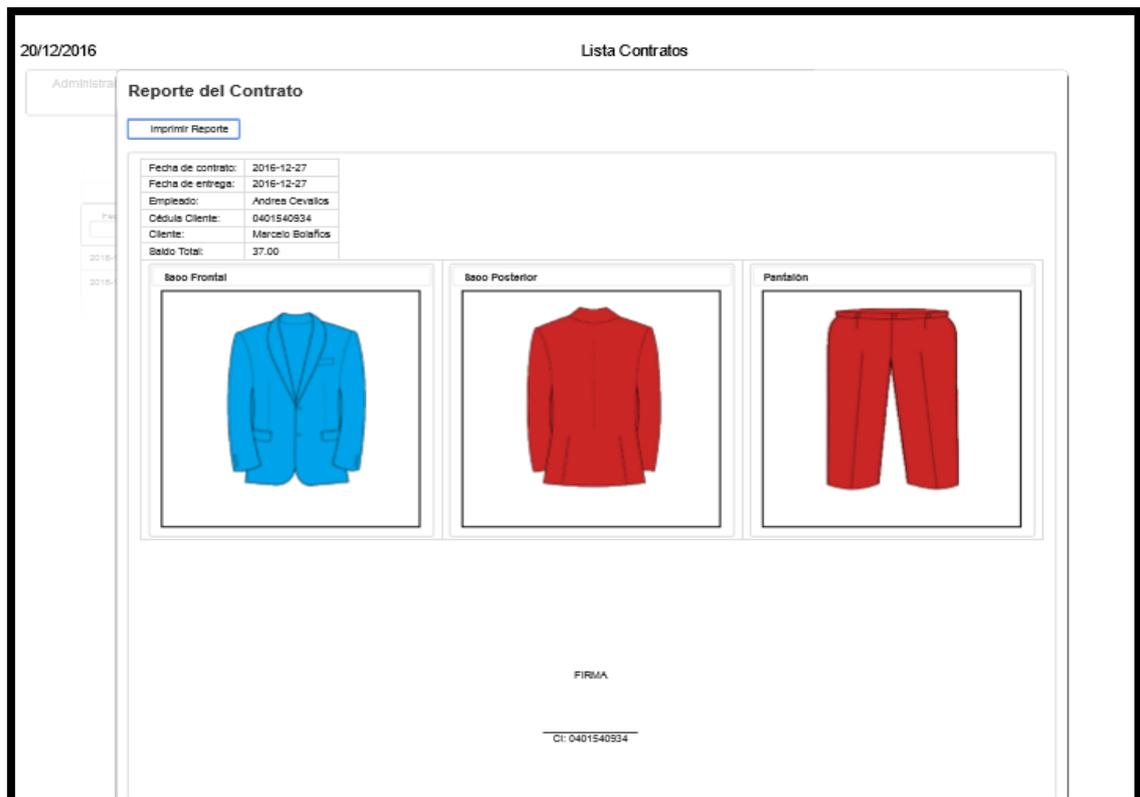


Ilustración 68: Hoja de Impresión- Contrato firmado por el cliente.

Fuente: Propia

- A continuación, se muestra el reporte que corresponde al pago de los empleados por contrato, en el que se determina el saldo total que se debe cancelar al empleado por contrato.

Fecha	Empleado	Cliente	Fecha de Entrega	Saldo total	Saldo Adeudado	Pagos
2017-02-18	David López	Gabriela Estefanía Benavidades Hernández	2017-02-26	136.00	1.00	Pagar
2017-02-17	Andrea Cabezas	Eduardo Benito Lourdes	2017-02-16	145.00	90.00	Pagar
2017-02-15	Andrea Cabezas	Eduardo Benito Lourdes	2017-02-18	845.00	800.00	Pagar
2017-02-10	JOSELITO ROLANDO CORTEZ LOPEZ	Eduardo Benito Lourdes	2017-02-10	112.00	92.00	Pagar
2017-02-10	JOSELITO ROLANDO CORTEZ LOPEZ	Daniela Cevallos	2017-02-10	62.00	42.00	Pagar
2017-02-09	David López	Eduardo Benito Lourdes	2017-02-17	48.00	36.00	Pagar

Ilustración 69: Pago empleado por contrato.

Fuente: Propia.

- En la siguiente Ilustración 70, se encuentra el formulario para el registro de materiales de corte, confección y telas, ingresados de acuerdo al código, el tipo, la cantidad, la fecha en la que se adquiere el insumo del material.

Registro de Inventarios

Fecha:

Tipo de Material:

Código material:

Cantidad:

Descripción:

Observaciones:

Fecha	Tipo material	Código	Descripción	Usuario	Cantidad
2016-12-11	Confección	CC3425	Casimir Blanco	dany	123.00
2016-12-11	Confección	AA1844	Rusela para terno	dany	123.00

Ilustración 70: Ventana para el Registro de Inventario.

Fuente: Propia.

- Para facilitar la búsqueda de los materiales ingresados y para la toma de decisiones en base a la información registrada, se facilita al usuario la siguiente pantalla como nos indica la Ilustración 71.

Fecha	Tipo material	Código	Descripción	Usuario	Cantidad
2016-12-11	Confección	CC3425	Casimir Blanco	dany	123.00
2016-12-11	Confección	AA1844	Rusela para terno	dany	123.00

Ilustración 71: Reporte de Inventario.

Fuente: Propia.

- Finalmente, como se muestra en la Ilustración 72, permite al usuario visualizar un reporte de un valor estimado invertido en cada confección realizada por contrato, de acuerdo al año y al mes que corresponden.

Cliente	# Contratos	Aproximado	Año	Mes
Andrea Cabezas	18	6	2017	Junio
David López	11	12	2017	Junio
JOSELITO ROLANDO CORTEZ LOPEZ	4	12	2017	Junio

Total Invertido: 30

Ilustración 72: Lista de Valor Estimado en metros de tela invertidos por contrato firmado en la fecha correspondiente.

Fuente: Propia.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- 1) Se permitió llevar un registro y control de información de clientes y empleados debidamente organizado, sistematizado e íntegro, facilitando a la empresa la forma rápida y eficaz de acceder a los datos, de tal forma que el proceso mejore la atención que brinda la Microempresa Artesanal “SASTRERÍA ARTEAGA”.
- 2) Se agilitó el registro de los materiales para el proceso de inventarios que maneja la microempresa, lo cual aportó a la mejora en la toma de decisiones de la misma, dando como resultado alcanzar la calidad en su producto final.
- 3) Se facilitó tener un registro de los pagos a los empleados por las prendas que confeccionan de tal manera que se valore su trabajo y puedan verificar sus saldos correspondientes en su remuneración.
- 4) El framework JSF facilitó el desarrollo del sistema al dar las soluciones esperadas, por ejemplo: manejo de eventos y listeners de forma sencilla y rápida; la efectiva comunicación entre las vistas y los controladores, opciones para convertir y validar los datos, al igual que permite implementar la internacionalización (i18n) dentro de las configuraciones básicas como aporte a la parametrización del software.
- 5) El uso de Primefaces como librería de componentes visuales para Java Server Faces (JSF) al ser de código abierto y tener una gran cantidad de componentes facilitó la creación de las páginas de forma rápida, sencilla, completa durante el desarrollo del sistema.
- 6) El desarrollo del presente sistema me ayudó a la práctica y refuerzo de conocimientos acerca del manejo de las herramientas RIA como es el uso de Primefaces, mediante la arquitectura que proporciona el framework JSF, desarrollando las destrezas necesarias y útiles en mi vida profesional.

5.2. RECOMENDACIONES

- 1) Es necesario que la Microempresa Artesanal “SASTRERÍA ARTEAGA”, registre la mayor cantidad de información en el presente sistema, de años significativos para que se muestren los datos necesarios actualmente para la misma en cuanto a clientes y empleados.
- 2) De igual forma, se tome con la importancia del caso lo referente al registro de inventario del material que se maneja para obtener resultados satisfactorios para la microempresa.
- 3) Recomiendo al administrador del sistema registre todos los saldos en cuanto a los pagos del empleado, al ser muy importante tener un control real de lo que se les cancela o lo que ellos adeudan, a fin de no perder cifras significativas para las dos partes.
- 4) Se recomienda a la Carrera de Ingeniería en Ciencias Aplicadas se incluya más materias enfocadas en el desarrollo de aplicaciones web, con herramientas libres o se refuercen las que ya existen, debido a que es el auge en la actualidad, útil para cualquier entidad.
- 5) Para el buen uso de Java Server Faces como framework de desarrollo se debe tomar en cuenta la creación de un bean por página JSF (Backing Bean), cumpliendo como intermediario entre la página xhtml y la lógica de negocio de la aplicación.
- 6) Es importante el uso de componentes que ofrece Primefaces para potencializar al framework JSF y que éste tenga versatilidad en su diseño permitiendo distintas posibilidades de agrupación de las etiquetas de desarrollo.

5.3. IMPACTOS DEL SISTEMA INFORMÁTICO

5.3.1 CARACTERÍSTICAS Y BENEFICIOS DE ESTE SISTEMA

➤ **Integridad de la información**

La información se encuentra en óptimas condiciones al estar en un medio informático seguro y diseñado especialmente para el fin establecido.

➤ **Almacenamiento ordenado de información**

Una estructura bien diseñada con el fin de tener acceso fácil y rápido a ella en cualquier momento.

➤ **Centralización de la información**

Será la fuente de información principal, haciendo que documentos, recibos y demás información sea manejada por él.

➤ **Claridad en los procesos**

Los procesos alrededor del sistema de información se tornarán más claros debido a que este será un gestor que sigue unas reglas claras y ágiles.

➤ **Confidencialidad de la información y seguridad**

Solo personal autorizado tiene acceso a determinada información.

➤ **Rapidez en la generación de reportes**

Un sistema informático debe estar en capacidad de generar reportes tanto externos (para otras entidades), como internos de forma rápida pues tiene una velocidad alta de procesamiento.

➤ **Disponibilidad de datos estadísticos**

Al contar con información debidamente organizada se pueden tener datos estadísticos en tiempo real (actualizados constantemente).

➤ **Seguimiento de indicadores de gestión y calidad.**

Es esencial para un mejoramiento continuo tener un buen esquema de indicadores de gestión y de calidad lo cual es parte integral de un sistema de información.

5.3.2. IMPACTOS

La automatización de historias del proceso de confección de prendas, así como también el registro de la información acerca de clientes, empleados, diseño y materiales e insumos, es mucho más seguro, confiable, actualizado y disponible, todo es posible porque se tiene almacenada la información en el repositorio de base de datos.

Anteriormente el registro de la información correspondiente a clientes, empleados, medidas y tipos de diseño lo realizaba de forma manual, para lo cual tardaba un lapso de tiempo considerable y molesto para el cliente.

Cuando esa parte del proceso finalizaba, continuaba con el corte del material donde registraba manualmente la parte de inventario con medidas de los cortes de piezas que empleaba para el terno.

Finalmente asignaba el material a cada empleado para la confección de la prenda (número de prendas a cada empleado), registro que lo realizaba de igual manera en forma física para cancelar en efectivo lo correspondiente a cada empleado por la obra realizada.

REFERENCIAS BIBLIOGRÁFICAS

- 1) Valarezo, E., & Luján Mora, S. (2014). Aplicaciones Web-Diseño para Aplicaciones Web. Aplicaciones Web.
- 2) Martín, A. R., & Martín, M. J. R. (2014). Aplicaciones Web. Ediciones Paraninfo, SA.
- 3) Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). Programación en Java: Algoritmos, programación orientada a objetos e interfaz gráfica de usuario. Mc Graw-Hill.
- 4) Matthew, D. (2011). HTML 5. Anaya Multimedia.
- 5) Orós Cabello, J. C. (2013). Diseño de Páginas Web con XHTML, Java Script y CSS. Alfaomega.
- 6) Gutiérrez, J. J. (2014). ¿Qué es un framework Web?. Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf Accessed May, 12.
- 7) Pardo Niebla, 2011 (Reim 2012), Guía visual de creación y diseño Web.
- 8) Kendall (2011), Análisis y diseño de sistemas.
- 9) Bell, D., & Parr, M. (2011). Java para Estudiantes. Pearson Educación.
- 10) Clavadetscher, C. (2015). Autorización en PostgreSQL.
- 11) Castellanos Veintimilla, M. D. (2015). Sistematización web para la automatización del proceso de gestión de las tesis de grado en la Facultad de Jurisprudencia de la Universidad Central del Ecuador.
- 12) Menor, P. Q. (2015). Implementación de un sistema de información bajo la metodología ágil Open UP para el control de los procesos compra, venta e inventario de la Botica Medicfarma en Tarapoto, San Martín. Revista de Investigación Ciencia, Tecnología y Desarrollo, 1(1).

- 13) Heredia, C. (2015). Integración de Spring, Hibernate y JSF en el desarrollo de aplicaciones web.
- 14) Ayala, C., Liseth, E., Hernández, L., Nelson, H., & Zelaya Guevara, H. D. (2015). Aplicación web para el control de almacén, elaboración de planillas, generación de horarios y gestión de empresas estudiantiles en el Instituto Nacional “Dr. Sarbelio Navarrete” del departamento de San Vicente (Doctoral dissertation, Universidad de El Salvador).
- 15) Acosta, C. S., Monteza, V. T., & Cabrera, I. M. (2015). ANÁLISIS COMPARATIVO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES WEB EN JAVA. *Revista Científica INGENIERÍA: Ciencia, Tecnología e Innovación*, 2(1), 60.
- 16) Palli, S. L., & Palli, S. G. (2015). Using Rich UI Frameworks for Monitoring and managing 4G Network elements. *International Journal of Advanced Studies in Computers, Science and Engineering*, 4(3), 18.
- 17) García Marro, F. (2015). Base de dades bursàtil basada en J2EE.
- 18) Chochos Ortega, D. E. (2015). Diseño, desarrollo e implementación de una aplicación web para la administración, prevención y soporte del sistema nacional de gestión de prevención de riesgos laborales SGP.