



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS
PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI
COMERCIALIZA”**

AUTOR: OSCAR IVÁN POZO VILLARREAL

DIRECTOR: ECO. WINSTON OVIEDO

IBARRA – ECUADOR

2015



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 los textos completos de forma 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 DE CONTACTO			
CÉDULA DE IDENTIDAD	040131523-9		
APELLIDOS Y NOMBRES	POZO VILLARREAL OSCAR IVÁN		
DIRECCIÓN	CDLA. PADRE CARLOS – TULCÁN		
E-MAIL	os_pvilla@hotmail.com		
TELÉFONO FIJO:	062236380	TELÉFONO MÓVIL	0982573232
DATOS DE LA OBRA			
TÍTULO	“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA”		
AUTORA	POZO VILLARREAL OSCAR IVÁN		
FECHA	JULIO DEL 2015		
PROGRAMA	<input type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO		
TÍTULO POR EL QUE OPTA	INGENIERO EN SISTEMAS COMPUTACIONALES		
DIRECTOR	ECO. WINSTON OVIEDO		

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Pozo Villarreal Oscar Iván, portador de cédula de ciudadanía 040131523-9 en calidad de autor y titular de los derechos patrimoniales de la obra o Trabajo de Grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el 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.

Firma



Nombre: Pozo Villarreal Oscar Iván

Cédula: 040131523-9

Ibarra, Julio del 2015



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

Certifico que la Tesis “**SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA**” ha sido realizada en su totalidad por el señor: Oscar Iván Pozo Villarreal, portador de la cédula de identidad número: 040131523-9.

A handwritten signature in blue ink, appearing to read "Winston Oviedo", is written over a horizontal line.

Eco. Winston Oviedo
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Oscar Iván Pozo Villarreal, con cédula de identidad Nro. 040131523-9 manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, 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 mencionada, 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: Pozo Villarreal Oscar Iván

Cédula: 040131523-9

Ibarra, Julio del 2015

CERTIFICACIÓN

Tulcán, 20 de febrero del 2015

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Señor Egresado **OSCAR IVÁN POZO VILLARREAL** con CI: **0401315239** quien desarrolló su trabajo con el tema "**SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA**", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado **OSCAR IVÁN POZO VILLARREAL**. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra institución.

El egresado **OSCAR IVÁN POZO VILLARREAL** puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Ing. Alex Ramos
Coordinador U.T.I.C.
Gobierno Autónomo Descentralizado de la provincia del Carchi

Calle 10 de Agosto entre Sucre y Olmedo Teléfono: (06) 2980-302
Pbx: 593 (06) 2980-302 ext 138 Código Postal: EC040101 Mail: info@carchi.gob.ec

www.carchi.gob.ec



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

Dedico este trabajo a mis padres María, Yolanda, Oswaldo y Luis quienes fueron un pilar fundamental y quienes me apoyan día a día.

A mi esposa Estefanía por estar siempre conmigo y demostrarme siempre su cariño y apoyo incondicional.

A mi hijo Aarón quien es la razón para cada día esforzarme por seguir superándome personal y profesionalmente.

A mis hermanos quienes nunca dejaron de alentarme para seguir adelante y culminar mi trabajo de tesis.

Iván Pozo



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

Primeramente agradezco a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional.

Agradecer a mis padres que me acompañan desde mis primeros pasos y hoy ven mi meta cumplida.

A todos mis familiares y amigos que comparten cada uno de mis sueños y han apoyado mis esfuerzos durante mi vida de estudiante.

Es importante agradecer a los profesores que desde el inicio de la carrera supieron entregar sus mejores esfuerzos dedicándose a enseñar y compartir sus conocimientos.

Como siempre existe en nuestras vidas la persona que se dedica a guiar y a dar el ánimo necesario en este caso quisiera agradecer al Eco. Winston Oviedo quien se dedicó a revisar de manera continua los avances del proyecto.

Iván Pozo

RESUMEN

La Presente tesis tiene el objetivo de realizar una aplicación web con software libre, el mismo que está dirigido al Gobierno Autónomo Descentralizado de la provincial del Carchi, las herramientas a emplearse son Apache como servidor web, PHP como lenguaje de programación y PostgreSQL como base de datos, esto por ser herramientas de software libre y además como uno de los requerimiento de la institución auspiciante, el sistema está orientado a gestionar la información de los proyectos Carchi Comercializa y Carchi Seguridad Alimentaria, en donde el primer proyecto se encarga de llevar información concerniente a ingreso y egreso de productos, registro de proveedores de la provincia del Carchi, control de pedidos y control de pagos; en cuanto al segundo proyecto se encarga de llevar un registro de beneficiados con información de su localidad, actividades económicas, numero de cargas familiares y datos del conyugue, por lo cual el sistema cuenta con los módulos de registro y análisis, de control y de costos, que automatizan los procesos de los dos proyectos antes mencionados y de esta manera centralizar la información tanto de beneficiados como de proveedores en un solo servidor, evitando duplicidad en procesos y pérdida de costos y tiempo, de la misma manera facilitara la generación de reportes para una mejor toma de decisiones. Para el desarrollo de esta tesis se cuenta con varios capítulos los mismos que describiremos brevemente CAPITULO I. Se realiza una introducción del proyecto, objetivos y alcance del mismo. CAPITULO II. En esta parte se realiza la elaboración del marco teórico en donde se describe los procesos que realiza cada uno de los proyectos así como también las herramientas y metodologías para el desarrollo del proyecto. CAPITULO III. Es en esta etapa es donde se realiza el análisis, diseño construcción e implementación de la aplicación esto en base a los requerimientos de la institución auspiciante. CAPITULO IV. Se realiza las conclusiones y recomendaciones del proceso y resultados obtenidos.

SUMMARY

The present thesis aims to make a web application based on free software, the same is directed to the use of the Gobierno Provincial del Carchi, the tools used are Apache as a web server, PHP as programming language and PostgreSQL as a database data, this are free software tools and also is one of the requirements of the sponsoring institution, the system is designed to manage project information from the Carchi Markets for a Food Safety project, where the first project is responsible for carrying information concerning to entry and exit of products, registration of suppliers in the province of Carchi, control orders and payment control; as for the second project is responsible for keeping a record of beneficiaries with information from your local economic activities, number of dependents and data spouse, so the system has modules recording and analysis, control and cost to automate the processes of the two mentioned projects and the centralizing information both beneficiaries and providers on a single server, avoiding duplication of processes and loss of cost and time, in the same way facilitate the generation of reports for better decision-making. For the development of this thesis there are several chapters that describe them briefly CHAPTER I. introduction project objectives and scope. CHAPTER II. In this part of the development of the theoretical framework where processes performed by each of the projects as well as tools and methodologies for the project as performed. CHAPTER III. It is at this stage where the analysis is performed, also the construction design and implementation of this application based on the requirements of the sponsoring institution. CHAPTER IV. The conclusions and recommendations of the process and results are performed.

ÍNDICE DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN.....	II
CERTIFICACIÓN DEL ASESOR	IV
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	V
DEDICATORIA.....	VII
AGRADECIMIENTO	VIII
RESUMEN	IX
SUMMARY	X
ÍNDICE DE CONTENIDO.....	XI
ÍNDICE DE FIGURAS	XIV
ÍNDICE DE TABLA	XVII
CAPÍTULO I	1
1 INTRODUCCIÓN.....	1
1.1 PROBLEMA.....	1
1.1.1 ANTECEDENTES	1
1.1.2 PROSPECTIVA.....	2
1.1.3 PLANTEAMIENTO	3
1.2 OBJETIVOS.....	3
1.2.1 OBJETIVO GENERAL.....	3
1.2.2 OBJETIVOS ESPECÍFICOS	3
1.2.3 ALCANCE	3
1.3 JUSTIFICACIÓN	7
CAPÍTULO II	9
2 MARCO TEÓRICO.....	9
2.1 INTRODUCCIÓN	9

2.2 PROCESO DEL PROYECTO CARCHI SEGURIDAD ALIMENTARIA.....	10
2.3 PROCESO DEL PROYECTO CARCHI COMERCIALIZA.....	13
2.4 HERRAMIENTAS Y METODOLOGÍA DE DESARROLLO.....	15
2.4.1 DEFINICIONES.....	15
2.4.2 ESPECIFICACIONES.....	34
2.4.3 LIMITACIONES.....	39
CAPÍTULO III	42
3 DESARROLLO DE LA APLICACIÓN	42
3.1 FASE DE ANÁLISIS	42
3.1.1 ESPECIFICACIÓN DE REQUISITOS DE USUARIO. (PRODUCT OWNER – VOZ CLIENTE).....	42
3.1.2 REQUISITOS DE CAPACIDAD Y RESTRICCIÓN (PRODUCT BACKLOG – OBJETIVOS PRIORIZADOS).....	44
3.1.3 DEFINICIÓN DE REQUERIMIENTOS FUNCIONALES (PRODUCT OWNER – VOZ CLIENTE).....	45
3.1.4 DEFINICIÓN DE ALCANCE (PRODUCT OWNER – VOZ CLIENTE).....	47
3.1.5 DEFINICIÓN DE TAREAS POR ITERACIÓN (SPRINT BACKLOG – TAREAS DE ITERACIÓN).....	52
3.2 FASE DE DISEÑO	55
3.2.1 DISEÑO DE ARQUITECTURA PRELIMINAR.....	55
3.2.2 REVISIÓN Y ACTUALIZACIÓN DE TAREAS POR ITERACIÓN (SPRINT BACKLOG – TAREAS DE ITERACIÓN)	66
3.2.3 REQUISITOS DE INTERFAZ PRELIMINAR	70
3.2.4 ELABORACIÓN DE UN PLAN DE PRUEBAS	71
3.2.5 AJUSTE DE LA DEFINICIÓN DEL PROYECTO	75
3.2.6 AJUSTE DE LA DEFINICIÓN DEL PROYECTO	75
3.2.7 DOCUMENTACIÓN DE SEGUIMIENTO DE TAREAS DE DISEÑO	76
3.3 FASE DE CONSTRUCCIÓN Y CONTROL DE FUNCIONABILIDAD	86

3.3.1 DESARROLLO DE INTERFACES.....	86
3.3.2 DISEÑO DE FUNCIONALIDAD DEL SISTEMA.....	89
3.3.3 ELABORACIÓN DE UN PLAN DE INTEGRACIÓN DEL SISTEMA.....	90
3.3.4 DOCUMENTACIÓN DE CONSTRUCCIÓN Y CONTROL DE FUNCIONABILIDAD.....	91
3.4 FASE DE DESPLIEGUE - IMPLEMENTACIÓN.....	92
3.4.1 INSTALACIÓN DEL PRODUCTO.....	92
3.4.2 CAPACITACIÓN FINAL DE USUARIOS Y PERSONAL DE SOPORTE.....	93
3.4.3 APROBACIÓN DEL PROYECTO.....	93
3.4.4 EVALUACIÓN DEL PROYECTO.....	94
3.4.5 DOCUMENTACIÓN DE SEGUIMIENTO DE TAREAS DE IMPLEMENTACIÓN..	95
CAPÍTULO IV.....	96
4 CONCLUSIONES Y RECOMENDACIONES.....	96
4.1 CONCLUSIONES.....	96
4.2 RECOMENDACIONES.....	96
4.3 GLOSARIO DE TÉRMINOS.....	98
4.4 BIBLIOGRAFÍA.....	100
4.5 ANEXOS.....	102
ANEXO 1: SPRINT'S.....	102
ANEXO 2: CAPACITACIÓN.....	119
ANEXO 3: MANUALES.....	120

ÍNDICE DE FIGURAS

FIGURA 1: Alcance en el desarrollo de la aplicación.....	5
FIGURA 2: Proceso de Seguridad Alimentaria.....	12
FIGURA 3: Proceso de Seguridad Alimentaria.....	14
FIGURA 4: Metodología modelo de cascada.....	27
FIGURA 5: Metodología de prototipos.....	27
FIGURA 6: Metodología Incremental.....	28
FIGURA 7: Metodología Incremental.....	29
FIGURA 8: Metodología RUP.....	30
FIGURA 9: Metodología KANBAN.....	32
FIGURA 10: Metodología SCRUM.....	33
FIGURA 11: Metodología SCRUM.....	34
FIGURA 12: Ciclo Sprint.....	35
FIGURA 13: Caso de uso registro y análisis.....	47
FIGURA 14: Caso de uso gestión de beneficiario.....	47
FIGURA 15: Caso de uso análisis de prioridad.....	48
FIGURA 16: Caso de uso análisis de capacidad.....	48
FIGURA 17: Caso de uso gestión de beneficio.....	49
FIGURA 18: Caso de uso costos.....	49
FIGURA 19: Caso de uso gestión de productos.....	50
FIGURA 20: Caso de uso gestión de clientes.....	50
FIGURA 21: Caso de uso gestión de proveedores.....	51
FIGURA 22: Caso de uso gestión de proveedores.....	51
FIGURA 23: Caso de iteración registro y análisis.....	52
FIGURA 24: Caso de iteración costos.....	52

FIGURA 25: Modelo Vista Controlador.	56
FIGURA 26: Arquitectura del lenguaje.	56
FIGURA 27: Arquitectura del sistema.	57
FIGURA 28: Base de datos.	58
FIGURA 29: Diagrama Proyecto Seguridad Alimentaria.	59
FIGURA 30: Diagrama Proyecto Carchi Comercializa	60
FIGURA 31: Diagrama lenguaje de interface	61
FIGURA 32: Funcionalidad estándar de pantallas.	62
FIGURA 33: Funcionalidad estándar de pantallas.	62
FIGURA 34: Gestión beneficiarios.	63
FIGURA 35: Ingreso al sistema.	63
FIGURA 36: Gestión Proveedores.	64
FIGURA 37: Gestión Productos.	64
FIGURA 38: Gestión Compras	65
FIGURA 39: Gestión Ventas.	65
FIGURA 40: Gestión Reportes.	66
FIGURA 41: Ciclos de un proceso.	66
FIGURA 42: Interface Seguridad Alimentaria.	70
FIGURA 43: Interface Carchi Comercializa.	71
FIGURA 44: Interface Ingreso al sistema.	76
FIGURA 45: Interface Validación de usuario.	77
FIGURA 46: Interface gestión Beneficiarios.	78
FIGURA 47: Interface Análisis beneficio.	79
FIGURA 48: Interface asignación de bono.	80
FIGURA 49: Interface Ingreso al sistema.	81
FIGURA 50: Interface compra de productos.	82
FIGURA 51: Interface venta de productos.	83

FIGURA 52: Interface ingreso de proveedores.....	84
FIGURA 53: Interfaz de diseño.....	86
FIGURA 54: Diseño de usabilidad	87
FIGURA 55: Diseño de funcionalidad sistema SACC.	89
FIGURA 56: Pantalla de ingreso al sistema SACC.....	92

ÍNDICE DE TABLA

TABLA 1: Clasificación de Bonos, Condiciones, Montos.	6
TABLA 2: Cuadro comparativo lenguajes del lado cliente.	17
TABLA 3: Cuadro comparativo lenguajes del lado servidor.	20
TABLA 5: beneficios de la metodología SCRUM.	36
TABLA 6: Roles metodología SCRUM.	42
TABLA 7: Sprint general - SACC	43
TABLA 8: Requisitos funcionales SACC.....	45
TABLA 9: Requisitos no funcionales SACC.....	46
TABLA 10: Requisitos de implementación SACC.	46
TABLA 11: Documentación de Sprint de análisis.	53
TABLA 12: Spring Parametrización de datos iniciales del sistema perfiles y usuarios. .	67
TABLA 13: Spring Ingreso de proveedores.	68
TABLA 14: Spring Ingreso de catálogo de productos.	69
TABLA 15: Prueba.	74
TABLA 16: Prueba caso de uso.	75
TABLA 17: Prueba ingreso al sistema.	76
TABLA 18: Prueba validación de usuario.	77
TABLA 19: Prueba gestión beneficiarios.	78
TABLA 20: Prueba análisis beneficio.	79
TABLA 21: Prueba asignación Bono.	80
TABLA 22: Prueba ingreso al sistema.	81
TABLA 23: Prueba validación usuarios.	81
TABLA 24: Prueba compra de productos.	82
TABLA 25: Prueba ingreso a inventario.....	82
TABLA 26: Prueba venta de productos.	83

TABLA 27: Prueba ingreso de proveedores.....	84
TABLA 28: Documentación de Sprint de diseño.	85
TABLA 29: Estructura diseño del sistema.....	89
TABLA 30: Cuadro de plan de integración del sistema.....	90
TABLA 31: Documentación de Sprint de construcción y control de funcionalidad.	91
TABLA 32: Cronograma de capacitación.....	93
TABLA 33: Cronograma de aprobación del proyecto.	93
TABLA 34: Análisis costo beneficio.....	94
TABLA 35: Documentación de Sprint de tareas de implementación.	95

CAPÍTULO I

1 INTRODUCCIÓN

1.1 PROBLEMA

1.1.1 ANTECEDENTES

A nivel nacional las provincias fronterizas mantiene en sus territorios gran cantidad de personas refugiadas y este es el caso de la provincia fronteriza del Carchi. Por la situación territorial en la cual la provincia se encuentra, tiene como una de sus competencias y cumpliendo con una parte de las obligaciones de los Proyectos del Buen Vivir la administración para el registro y control de aportes que realiza el Programa Mundial de Alimentos.

El Gobierno Autónomo Descentralizado de la Provincia del Carchi inicio con el Proyecto de Ayuda Humanitaria en lo referente al denominado eslogan “Carchi come bien vive bien” (Seguridad Alimentaria PMA, 2011) en el cual involucra a las personas favorecidas con los Bonos Alimenticios así como también pensando en las personas de la producción minorista, los mismos que son Asociaciones o personas naturales procedentes de la Provincia norteña.

Tomando en cuenta estos aspectos se crean los Proyectos denominados Carchi Seguridad Alimentaria y Carchi Comercializa (Seguridad Alimentaria PMA, 2011) los que permiten hacer la adquisición de productos aceptados en la canasta de aportes alimenticios a ser entregadas a las personas favorecidas adquiriendo productos aptos para el consumo humano los mismos que son de nuestros agricultores y productores carchenses.

SITUACIÓN ACTUAL

Actualmente la información de los beneficiarios se la lleva en formularios, registros que se efectúan manualmente por los funcionarios del GAD¹ de la provincia del Carchi, que como contra parte apoyan al proyecto de ayuda humanitaria que mantiene convenios con varias provincias del Ecuador.

¹ GAD: Gobierno Autónomo Descentralizado.

El proyecto enfocado a la ayuda humanitaria, realiza la invitación a las personas que deseen ser favorecidos con estos planes y bonos solidarios a que se acerquen a la institución, para el ingreso de información con la finalidad de que en fechas posteriores técnicos del GAD de la provincia del Carchi realicen el análisis y coordinen las visitas para la evaluación y verificación de las personas refugiadas o desplazadas de la frontera Colombo-Ecuatoriana.

De igual manera las visitas son efectuadas luego de una clasificación de manera manual para determinar la ubicación de los usuarios en los diferentes cantones de la provincia.

La sectorización de los beneficiarios tanto refugiados y asociaciones se mantiene de manera manual, de igual forma las relaciones existentes entre beneficiarios y sus respectivos cónyuges no se identifica inmediatamente, impidiendo la clasificación y limitación de beneficios por familias.

La entrega de los Beneficios se llevan en un control de libro diario el mismo que para efectuar los pedidos de ingreso y emitir las entregas se realiza el registro en hojas de cálculo de Excel generando inconsistencia en los cálculos de precios y costos, emitiendo datos erróneos en los reportes de periodos emitidos, falta de seguridad en la información existente por alteraciones voluntarias o involuntarias de los datos.

1.1.2 PROSPECTIVA

Considerando que no se realice la digitación de los registros de cada uno de los beneficiarios, datos como localidad, actividades económicas del refugiado, número de cargas familiares, relaciones conyugales, etc. La institución se aglomeraría de información difícil de manejar y administrar de manera manual, tomando en cuenta que el número de refugiados anualmente incrementa y por ende la cantidad de registros.

De igual manera el Proyecto Carchi Comercializa mantiene de manera constante el ingreso y egreso de productos, la comunicación con proveedores, el registro y control de pagos, reporte de beneficiados, los mismos que de mantenerse de manera manual conllevarían a una pérdida involuntaria de información y dinero.

1.1.3 PLANTEAMIENTO

El registro de información de las personas refugiadas y la entrega de los beneficios se la lleva de manera manual, lo que genera pérdida de información, pérdida de tiempo al momento de registrar, reportes y duplicidad de datos, impidiendo una acertada toma de decisiones.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Implementar un sistema informático para apoyar los procesos de los proyectos Carchi Seguridad Alimentaria y Carchi Comercializa, mejorando la gestión de dichos proyectos.

1.2.2 OBJETIVOS ESPECÍFICOS

- Analizar y determinar las funciones de registro, toma de decisiones de los refugiados y abastecimiento de beneficios por el Proyecto Carchi Seguridad Alimentaria y Carchi Comercializa.
- Registro de información y requerimientos a ser utilizados para el desarrollo del sistema.
- Diseño y estructura del modelo del sistema informático.
- Desarrollo del proyecto de software para la gestión de proyectos Carchi Seguridad Alimentaria y Carchi Comercializa utilizando la metodología SCRUM, y herramientas de desarrollo PHP² y PostgreSQL.

1.2.3 ALCANCE

Tener al alcance la información de los registros existentes en el Proyecto Carchi Seguridad Alimentaria y Carchi Comercializa referente a los datos de los diferentes ciudadanos beneficiados, así como también de las Asociaciones proveedoras de productos.

² PHP: Acrónimo de Hipertext Pre-procesor (Hipertexto pre-procesado).

Se mantendrá un historial de la información ingresada por cada periodo o en un tiempo determinado ya que los beneficiados al ser ciudadanos no natos podrían regresar a su país o cambiar su residencia.

La automatización de este proceso beneficiaría al Proyecto de Ayuda Humanitaria ya que al contar con la centralización de la información ayudaría a las consultas y verificación de los datos de registros y entrega de beneficios.

El sistema presentaría información sectorizada es decir podremos observar el número de usuarios existentes por cantón, por parroquias, de esta manera tener una idea del número total de refugiados en nuestra provincia.

No solamente el Proyecto Carchi Seguridad Alimentaria se involucrará en el sistema ya que al generar diariamente el ingreso de nuevos beneficiados se emite información al área de Carchi Comercializa el que contará con la herramienta en donde se llevará un registro de todos los cambios efectuados en el primer Proyecto y el abastecimiento necesario para las entregas.

Los funcionarios de los Proyectos Carchi Seguridad Alimentaria y Carchi Comercializa serán los responsables directos de la manipulación de este software con eficiencia y eficacia profesional, los mismos que mantendrán información correcta de los datos con los cuales se va a realizar el registro, análisis, sectorización, aprobación y negación de solicitudes; así como también la vinculación con las asociaciones proveedoras, los cuales generan ingreso de información a ser controlada por la Institución.

El administrador del programa es la persona que genera los permisos pertinentes de quienes van a utilizar este programa.

En cuanto a las herramientas de desarrollo a utilizarse serán:

Base de datos PostgreSQL.

Código para el desarrollo WEB PHP.

La metodología para el desarrollo de la aplicación será Metodología Ágil SCRUM donde permite realizar trabajo en equipo conjuntamente usuario propietario y desarrollador, entregando avances parciales y regulares del producto de interacción final.

Para una mejor comprensión de cómo será la arquitectura funcional del sistema lo ilustramos en el siguiente gráfico.

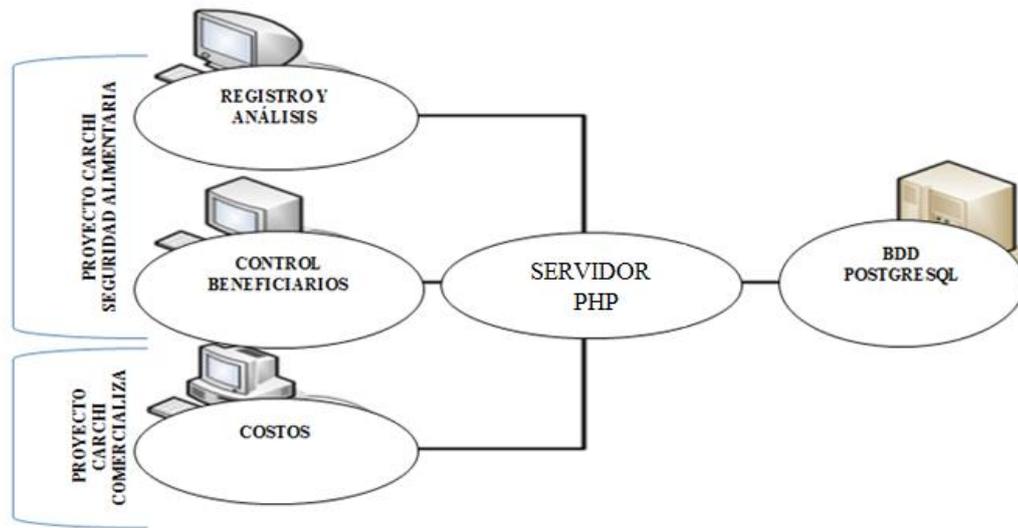


FIGURA 1: Alcance en el desarrollo de la aplicación.

Fuente: Autoría propia a partir de información recopilada.

PROYECTO CARCHI SEGURIDAD ALIMENTARIA

MÓDULO DE REGISTRO Y ANÁLISIS

- Creación de registros.
- Modificación de registros.
- Eliminación de registros.
- Análisis de prioridad.
- Análisis de capacidad.
- Análisis de beneficio.

TABLA 1: Clasificación de Bonos, Condiciones, Montos.

TIPO DE BENEFICIO	CONDICIONES PAREJA	CANTIDAD BONO	DURACIÓN
PRRO Vulnerables ³	1 colombiana (1 o 2)	\$ 20	12 meses
	1 colombiana (3)	\$30	
	1 colombiana (4 o más)	\$40	
PRRO Socorro ⁴	2 personas colombianas	\$ 20 - \$120 -40	12 meses
CLOSAN ⁵	1 jefe de hogar - Niños de 6 meses a 3 años de edad SOLO ECUATORIANAS	\$ 40	12 meses

Fuente: Autoría propia a partir de información recopilada.

- Reporte de especificación y aprobación.

MÓDULO DE CONTROL

- Creación de registros por beneficio.
- Análisis de direccionamiento del beneficio.
- Depuración de datos.
- Control de beneficios no dúplex.
- Reporte general de usuarios más beneficio.
- Cálculo de montos por beneficiado.

³ **PRRO Vulnerables:** - Proyecto de Recuperación en Operaciones Vulnerables.

⁴ **PRRO Socorro:** - Proyecto de Recuperación en Operaciones Prolongadas de Socorro.

⁵ **CLOSAN-** Capacidades Locales en Alimentación y Nutrición.

PROYECTO CARCHI COMERCIALIZA

MÓDULO DE COSTOS

- Creación de proveedores – asociaciones.
- Creación planillas de emisión.
- Análisis de costos.
- Ingreso y egreso de productos.
- Cálculo de costos subtotal, total.
- Registro entrega y recibí conforme.

1.3 JUSTIFICACIÓN

Importancia social

El Proyecto Carchi Seguridad Alimentaria en los 8 meses de ejecución registra aproximadamente 1400 familias beneficiadas y Carchi Comercializa en los 6 meses registra alrededor de 70 proveedores beneficiados quienes serán los principales favorecidos por el sistema, al poder llevar una debida gestión de actividades existentes en los proyectos y a su vez tener a mano un registro del total de proveedores, así como también el total de personas que perciben el bono de ayuda humanitaria.

Importancia técnica

La elaboración de Modelos y Programas Informáticos con alta capacidad en el manejo de los datos puede constituirse en vehículos de socialización, no sólo de la propia información generada, sino de las herramientas adecuadas que faciliten la automatización de los procesos de registros de refugiados y control de beneficios.

La metodología a ser utilizada en el desarrollo del proyecto deberá ser ágil adaptable a los cambios y variantes existentes en el transcurso de la elaboración del software.

Importancia económica

La definición de reglas de registro y control a implementar en el sistema y cumpliendo con los requerimientos que exige el convenio con la entidad como contraparte GAD de la provincia del Carchi - PMA⁶ nos lanzaremos por la utilización de software de desarrollo gratuito (free), evitando de esta manera que instituciones que están en pleno desarrollo como es el caso del Gobierno Autónomo Descentralizado de la Provincia del Carchi no realice gastos en la adquisición de licencias por software propietario, de la misma manera al realizar la digitalización de los procesos se mejoraría la mano de obra en lo referente a costo hombre provocando la reestructuración del personal en otras actividades.

Importancia en la gestión operativa

En la actualidad los sistemas informáticos son la base fundamental del accionar de una empresa o institución ya sea esta pública o privada debido a la necesidad de que todas las actividades que se desarrollen se digitalicen mejorando los procesos en tiempo de ejecución, resultados confiables y seguros, información oportuna e inmediata.

⁶ **PMA:** Programa Mundial de Alimentos.

CAPÍTULO II

2 MARCO TEÓRICO

2.1 INTRODUCCIÓN

El Gobierno Autónomo Descentralizado de la provincia del Carchi (GAD de la provincia del Carchi) al ser una institución autónoma descentralizada tiene autonomía política, administrativa y financiera.

Basándose en los principios de solidaridad, la Prefectura Carchi tiene proyectos enfocados a mejorar las condiciones de vida de los habitantes de la provincia, entre estos tenemos proyectos de capacitación en superación y administración de sus actividades agrícolas, capacitación en actividades comerciales, actividades de producción, formación de asociaciones productoras entre otras; es así como se crea el proyecto Carchi Seguridad Alimentaria en cooperación con PMA.

El Proyecto Carchi Seguridad Alimentaria tiene como objetivo mejorar las condiciones de alimentación de los habitantes sean estos habitantes locales como también personas colombianas que se encuentren radicadas en el territorio carchense (Seguridad Alimentaria PMA, 2011).

La prefectura teniendo presente los esfuerzos de los agricultores y asociaciones por mejorar su economía, calidad de vida y producción identifico que una de las debilidades de los agricultores y las asociaciones carchenses era la competencia y falta de lanzamiento de sus productos, con estos antecedentes se crea el Proyecto Carchi Comercializa cuyo objetivo es ofertar productos elaborados por habitantes carchenses (Carchi Comercializa, 2010).

Se logra unificar los Proyectos Seguridad Alimentaria y Carchi Comercializa de ahí que la administración de la información generada en cada uno de estos proyectos es de difícil manejo y administración de forma física viendo la necesidad de la creación de un aplicativo para poder digitalizar los procesos de cada uno de estos proyectos permitiendo mantener una base de datos de información de beneficiarios así como también de las asociaciones proveedoras.

Al tener gran cantidad de usuarios que desean formar parte de los beneficios que tiene cada uno de los proyectos y considerando las distancias existentes para la presentación de su respectiva documentación se ve la necesidad de crear un aplicativo web de fácil manejo y utilización para el ingreso y postulación a mencionados proyectos.

2.2 PROCESO DEL PROYECTO CARCHI SEGURIDAD ALIMENTARIA

El denominado Proyecto Carchi Seguridad Alimentaria fue creado en el año 2011 el mismo que tuvo mucho éxito y acogida por parte de los grupos vulnerables de la provincia como es el caso de las personas refugiadas del hermano país de Colombia.

El objetivo principal de este proyecto es el mejorar la calidad de vida de los habitantes así como también brindar a los usuarios un desarrollo personal e intelectual utilizando para esto el convenio con el Programa Mundial de Alimentos que tiene como misión crear proyectos de desarrollo, para las personas que son refugias en un país por un periodo de 12 meses, es decir periodos de larga duración por parte de las personas desplazadas.

En el convenio se establece que las provincias fronterizas deberían evaluar la calidad y aceptación del proyecto, en caso de ser favorable durante el periodo de aceptación que es un límite de 12 meses, el proyecto se mantendrá en auge de igual manera por un lapso de 12 meses, incrementales.

Con esta breve explicación se puede indicar que los términos del convenio y los puntos vulnerables se establecen según la ubicación geográfica del territorio con el cual se crea el convenio.

Siendo así Carchi una provincia fronteriza donde aloja a gran cantidad de refugiados se ha identificado los posibles beneficiarios tales como:

Víctimas de desastres naturales (temblores, terremotos, inundaciones, huracanes, sequias, entre otros).

Personas desplazadas por conflictos armados, damnificados, pobreza.

El denominado sexo débil como es la mujer es uno de los puntos prioritarios del convenio PMA-GAD de la provincia del Carchi el cual identifica y denomina como jefe de hogar a una sola persona.

Para obtener los resultados esperados en el Proyecto de Seguridad Alimentaria se ha establecido el siguiente proceso (Seguridad Alimentaria PMA, 2011):

Recepción de postulantes.

Registros de datos personales.

Sectorización y asignación de técnicos.

Visitas postulantes.

Informe de visitas y análisis de aprobación.

Clasificación y acceso a los beneficios.

Asignación de bono:

PRRO Vulnerables.

PRRO Socorro.

CLOSAN.

Entrega de Tarjetas con el bono respectivo.

Firma de aceptación.

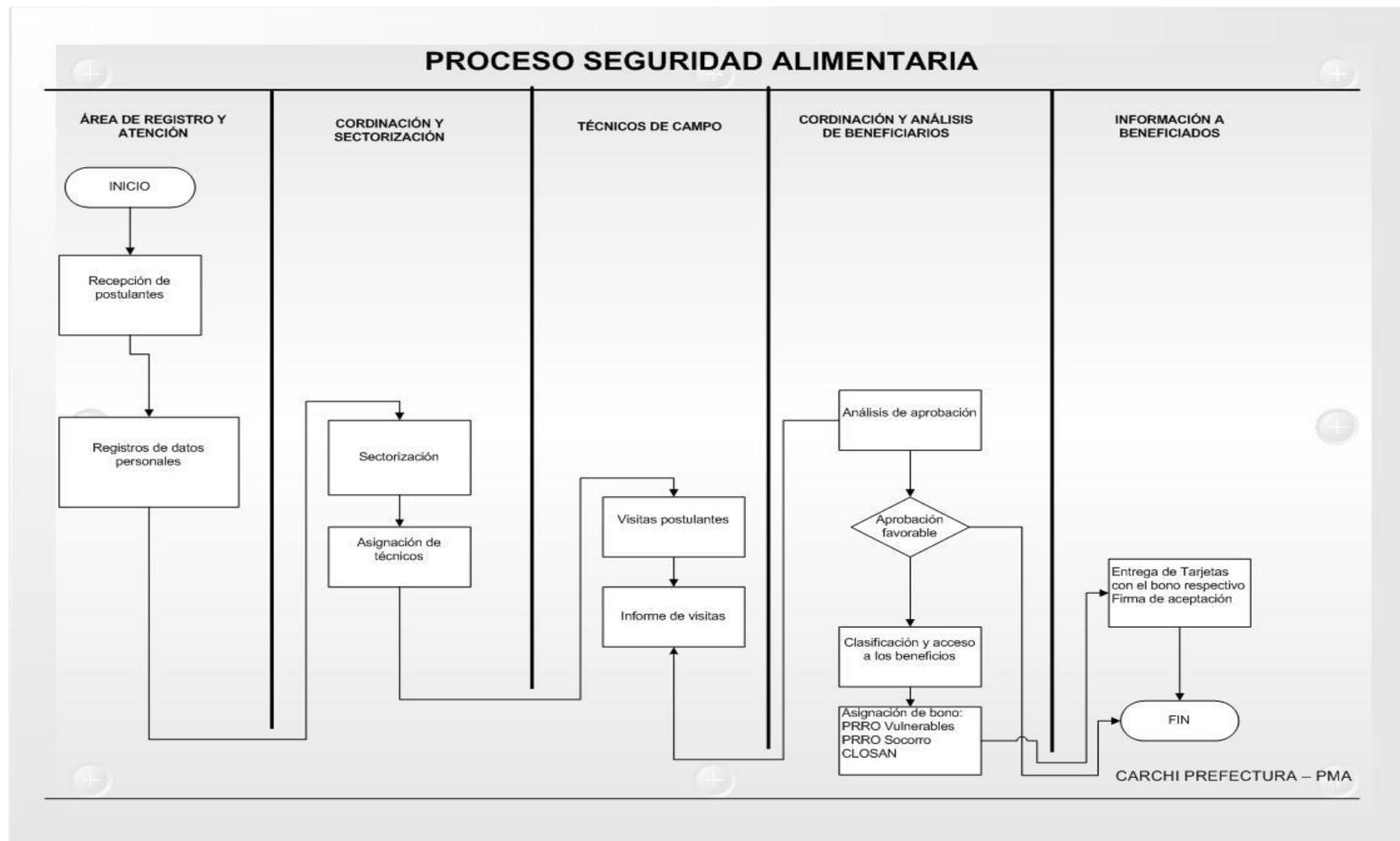


FIGURA 2: Proceso de Seguridad Alimentaria

Fuente: Convenio Carchi Prefectura – PMA.

2.3 PROCESO DEL PROYECTO CARCHI COMERCIALIZA

El GAD de la provincia del Carchi tiene gran cantidad de proyectos nuevos con nuevas metas y miras a un desarrollo provincial.

Es así que se ha dedicado a la formación de grupos de pequeños comerciantes y productores utilizando para ello el análisis de sector, productividad, zonas climáticas y varios factores que han ayudado a ver las posibles soluciones a desarrollo de la comunidad.

Tiene proyectos enfocados a la industrialización de lácteos, producción y agricultura, micro empresas, y más.

Carchi Comercializa (Carchi Comercializa, 2010) es el proyecto que incentiva a las asociaciones de productores carchenses a presentar sus productos y a ofertar al mercado de la provincia.

Con el objetivo ya trazado que es dar un mayor impulso a la producción, las asociaciones tienen como propósito incrementar el número de personas asociadas fortaleciendo así el proceso de Seguridad Alimentaria y de Carchi Comercializa.

Actividades para formar una asociación Carchense (D.E.L., 2008):

- Identificación de actividades por zonas.
- Capacitación e incentivo a la producción.
- Formación de asociaciones.
- Capacitación e incentivo a la permanencia en las asociaciones.
- Capacitación a la mejora de cultivos u otras actividades que desempeñen como asociación.
- Capacitación a la comercialización.
- Capacitación a la venta y oferta.
- Producción por entregas.
- Entrega del producto.

Proceso Carchi Comercializa

- ❖ Recepción de proveedores – asociaciones Carchenses.
- ❖ Clasificación de proveedores según ofertas.
- ❖ Registro de productos a ofertarse.
- ❖ Actualización de inventarios y comprobación de existencias.
- ❖ Pedidos de abastecimiento.
- ❖ Ingreso de productos según pedido.
- ❖ Venta al público.
- ❖ Creación de clientes.
- ❖ Cierres diarios.



FIGURA 3: Proceso de Seguridad Alimentaria

Fuente: Proyecto Carchi Comercializa.

2.4 HERRAMIENTAS Y METODOLOGÍA DE DESARROLLO

2.4.1 DEFINICIONES

Software Libre

Son programas de ordenador que pueden ser utilizados, copiados, modificados y redistribuidos libremente por sus usuarios.

WEB⁷ (WordPress, 2008-2014)

Es aquella que consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil como un smartphone o un monitor de computadora.

Servidor Web

Sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP⁸.

HTTP

Es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

Lenguajes de ejecución en el servidor

Son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

⁷ **WEB**: Es un vocablo inglés que significa "red", "telaraña" o "malla".

⁸ **HTTP**: Protocolo de transferencia de hipertexto.

Lenguajes de ejecución en el cliente

Son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento; entre los cuales no sólo se encuentra el HTML⁹ (LibrosWeb.es, 2015) sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML.

Lenguaje de desarrollo WEB

El tiempo ha ido evolucionando e incrementando las necesidades de los usuarios en automatizar diferentes procesos por tal motivo a un inicio de la programación se creaba sistemas locales con lenguajes estáticos.

Tipado estático (Se dice de un lenguaje de programación que usa un tipado estático cuando la comprobación de tipificación se realiza durante la compilación, y no durante la ejecución).

En la actualidad, la programación y las tecnologías han cambiado permitiendo acortar distancias y tiempos de respuesta pero de igual manera surgen más problemas e inconvenientes a los cuales se debe buscar una solución, es así como utilizando la red y el famoso Internet se crean los lenguajes de programación para la web.

Existen gran cantidad de lenguajes que permiten realizar esta actividad de una programación dinámica y web entre los tipos de lenguajes web y su clasificación tenemos: ¹⁶

⁹ **HTML:** HyperText Markup Language («lenguaje de marcas de hipertexto»).

Lenguajes del lado cliente

Lenguaje de lado cliente Son aquellos lenguajes que son asimilados Directamente por el navegador y no necesitan pre tratamiento.

TABLA 2: Cuadro comparativo lenguajes del lado cliente.

LENGUAJE	DEFINICION	VENTAJAS	DESVENTAJAS
	<p>Este lenguaje envía información al navegador indicando la ubicación de cada uno de los objetos utilizados tales como texto, imágenes, videos y su forma en la visualización de la página (Marcos, 2014).</p>	<ul style="list-style-type: none"> • De fácil manejo para la vinculación de texto e imágenes y programación. • Presentación amigable del texto. • Generación de archivos pequeños. • Velocidad aceptable. • Admisible con todos los navegadores, exploradores. 	<ul style="list-style-type: none"> • Lenguaje fijo, estático. • La comunicación con los navegadores puede ser diferente. • Conserva etiquetas y dificulta la corrección. • Diseño lento, etiquetas limitadas.
	<p>Es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.</p> <p>Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.</p>	<ul style="list-style-type: none"> • Seguro, confiable, de lento procesamiento. • Se ejecuta directamente en el cliente. 	<ul style="list-style-type: none"> • La visualización es accesible para todos. • No mantiene seguridad en la descarga del código. • Se debe combinar con otros lenguajes en lo referente a diseño.

	<p>Es otra manera de incluir código a ejecutar en los clientes que visualizan una página web. Se trata de pequeños programas hechos en Java, que se transfieren con las páginas web y que el navegador ejecuta en el espacio de la página.</p> <p>Los applets de Java están programados en Java y precompilados, es por ello que la manera de trabajar de éstos varía un poco con respecto a los lenguajes de script como Javascript. Los applets son más difíciles de programar que los scripts en Javascript y requerirán unos conocimientos básicos o medios del lenguaje Java.</p>	<ul style="list-style-type: none"> • Ofrecer una gran cantidad de contenidos interactivos. • Con la programación independiente de la plataforma ofrece una serie de beneficios tanto para programadores y usuarios. 	<ul style="list-style-type: none"> • Más difíciles de programar que los scripts. • Requieren conocimientos básicos o medios del lenguaje Java.
	<p>Es un lenguaje de programación de scripts del lado del cliente, pero sólo compatible con Internet Explorer. Es por ello que su utilización está desaconsejada a favor de Javascript.</p> <p>Está basado en Visual Basic, tanto su sintaxis como su entorno de trabajo. Sin embargo, no todo lo que se puede hacer en Visual Basic lo podremos hacer en Visual Basic Script, pues este último es una versión reducida del primero.</p>	<ul style="list-style-type: none"> • Es un lenguaje muy extendido. • El código en Visual Basic es fácilmente migrable a otros lenguajes. 	<ul style="list-style-type: none"> • No es Multiplataforma. • Es un lenguaje basado en objetos pero no implementa por completo la filosofía de Orientación a Objetos.

	<p>Flash es una tecnología, y un programa, para crear efectos especiales en páginas web. Con Flash también conseguimos hacer páginas dinámicas del lado del cliente. Flash en realidad no es un lenguaje; Sin embargo, si tuviéramos que catalogarlo en algún sitio quedaría dentro del ámbito de las páginas dinámicas de cliente (S.L, 2011).</p>	<ul style="list-style-type: none"> • Las Web's se vuelven atractivas. • Compatibilidad con todos los navegadores. 	<ul style="list-style-type: none"> • Más demora en tiempo de cargar las páginas. • Para su funcionamiento requiere la instalación de un Plug-in¹⁰.
	<p>CSS¹¹, es una tecnología que nos permite crear páginas web de una manera más exacta. Gracias a las CSS somos mucho más dueños de los resultados finales de la página, pudiendo hacer muchas cosas que no se podía hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores... Incluso podemos definir nuestros propios estilos en un archivo externo a nuestras páginas; así, si en algún momento queremos cambiar alguno de ellos, automáticamente se nos actualizarán todas las páginas vinculadas de nuestro sitio (Lazaro, 2001).</p>	<ul style="list-style-type: none"> • Con una Hoja de Estilo podemos modificar la presentación de cada elemento sin modificar el código HTML. • Con CSS se evita tener que recurrir a "trucos" para conseguir algunos efectos. 	<ul style="list-style-type: none"> • Si hay problemas de compatibilidades, el navegador aplicará el formato predeterminado. • Algunas propiedades de las CSS pueden provocar que una parte del contenido de nuestra página resulte inaccesible desde algunos navegadores.

Fuente: Autoría propia a partir de información recopilada.

¹⁰ **Plug-in:** Programa que puede anexarse a otro para aumentar sus funcionalidades.

¹¹ **CSS:** Cascading Style Sheets - Hojas de Estilo.

Lenguajes del lado servidor

Son aquellos lenguajes que se ejecutan por el propio servidor y son enviados al cliente en un formato claro para él.

TABLA 3: Cuadro comparativo lenguajes del lado servidor.

LENGUAJE	DEFINICION	VENTAJAS	DESVENTAJAS
	<p>Es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Actualmente se encuentra un poco desfasado por diversas razones entre las que destaca la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta.</p> <p>Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos.</p>	<ul style="list-style-type: none">• Independiente del lenguaje.• Interfaz sencilla: no hace falta utilizar una librería o API específicos.• Basta con utilizar entrada/salida estándar y variables de entorno.	<ul style="list-style-type: none">• Por cada petición HTTP se crea un nuevo proceso en el servidor.• Sobrecarga de la interpretación del script.• Las aplicaciones CGI son dependientes de la plataforma.

	<p>Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP.</p> <p>Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl.</p>	<ul style="list-style-type: none"> • Es un lenguaje que se le puede considerar como un lenguaje que no tiene fronteras. • Este lenguaje se desarrolló pensando en que el lenguaje fuera fácil de usar, eficiente, y completo. • Es multiplataforma. 	<ul style="list-style-type: none"> • Aunque es fácil de usar, es difícil de aprender. • Hay aplicaciones que requieren mucha rapidez en las cuales es mejor utilizar otros lenguajes de programación que no sean interpretados
	<p>ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).</p>	<ul style="list-style-type: none"> • Completamente orientado a objetos. • División entre la capa de aplicación o diseño y el código fuente. • Facilita el mantenimiento de grandes aplicaciones. • Incremento de velocidad de respuesta del servidor. 	<ul style="list-style-type: none"> • Mayor consumo de recursos. • Es un software propietario de código no abierto. • Se requiere servidores de gran capacidad.



PHP Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

	<p>JSP¹² Es una tecnología orientada a crear páginas web con programación en Java.</p> <p>Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.</p>	<ul style="list-style-type: none"> • Crea páginas del lado del servidor. • Es Multiplataforma. • Código bien estructurado. <p>Integridad con los módulos de Java.</p> <ul style="list-style-type: none"> • La parte dinámica está escrita en lenguaje Java. • Permite la utilización de servlets y su ejecución es rápida. 	<ul style="list-style-type: none"> • Complejidad de aprendizaje. • Para alojar el portal web necesita tener instalado un servidor Tomcat.
---	--	---	---

Fuente: Autoría propia a partir de información recopilada.

¹² **JSP:** Es un acrónimo de Java Server Pages, (Páginas de Servidor Java).

Lenguajes del lado cliente-servidor

LENGUAJE	DEFINICION	VENTAJAS	DESVENTAJAS
<p>dhtml</p>	<p>DHTML¹³ prácticamente no es un lenguaje de programación. Se trata de una nueva capacidad de la que disponen los navegadores modernos, para tener un mayor control sobre la página. DHTML realiza efectos en el navegador por los cuales se pueden mostrar y ocultar elementos de la página, se puede modificar su posición, dimensiones, color, etc.</p> <p>DHTML nos da más control sobre la página, gracias a que los navegadores modernos incluyen una nueva estructura para visualizar en páginas web denominada capa. Las capas se pueden ocultar, mostrar, desplazar, etc.</p> <p>En la actualidad, DHTML también puede englobar la programación en el servidor.</p>	<ul style="list-style-type: none"> • Es útil para desarrollar aplicaciones web en las que los usuarios pueden interaccionar con páginas o con aplicaciones que ofrezcan servicios tales como buscadores, gestores web de correo. 	<ul style="list-style-type: none"> • El desarrollar de este lenguaje es muy complejo. • Es de difícil depuración. • Pierde su funcionalidad de un navegador a otro.

¹³**DHTML:** Dinamic HyperText Markup Language (Lenguaje dinamico de hipertexto de marcado).

	Dentro del concepto de DHTML se engloban también las Hojas de Estilo en Cascada o CSS (Cascade Style Sheets), que veremos a continuación.		
	<p>XML¹⁴ es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores.</p> <p>XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes.</p>	<ul style="list-style-type: none"> • Diseñado para ser utilizado en cualquier lenguaje. • De fácil procesamiento. 	<ul style="list-style-type: none"> • Incompatibilidad en versiones. • Su sintaxis es redundante por que puede afectar la eficiencia.

Fuente: Autoría propia a partir de información recopilada.

¹⁴ **XML:** Extensible Markup Language (lenguaje de marcas extensible).

METODOLOGÍA DE DESARROLLO DE SOFTWARE

A nivel general una metodología es un camino a seguir para el cumplimiento de un objetivo y una meta.

En programación una metodología de desarrollo de software es el medio que se usa para la estructuración, la planificación y el desarrollo de un sistema informático.

Existen varias metodologías las mismas que se han creado con el pasar del tiempo y han surgido la modificación según las necesidades, cada metodología tiene sus fortalezas y sus debilidades.

Una metodología no es siempre aplicable para cada tipo de proyecto a desarrollarse, lo recomendable es que cada proyecto según su necesidad debe adaptarse a la metodología.

Cada metodología tiene diferente enfoque para el desarrollo de software de tal manera las metodologías de desarrollo se clasifican en:

Metodología Modelo Cascada

En este modelo se crea una serie de escalones o gradas en el desarrollo según el avance en las diferentes fases.

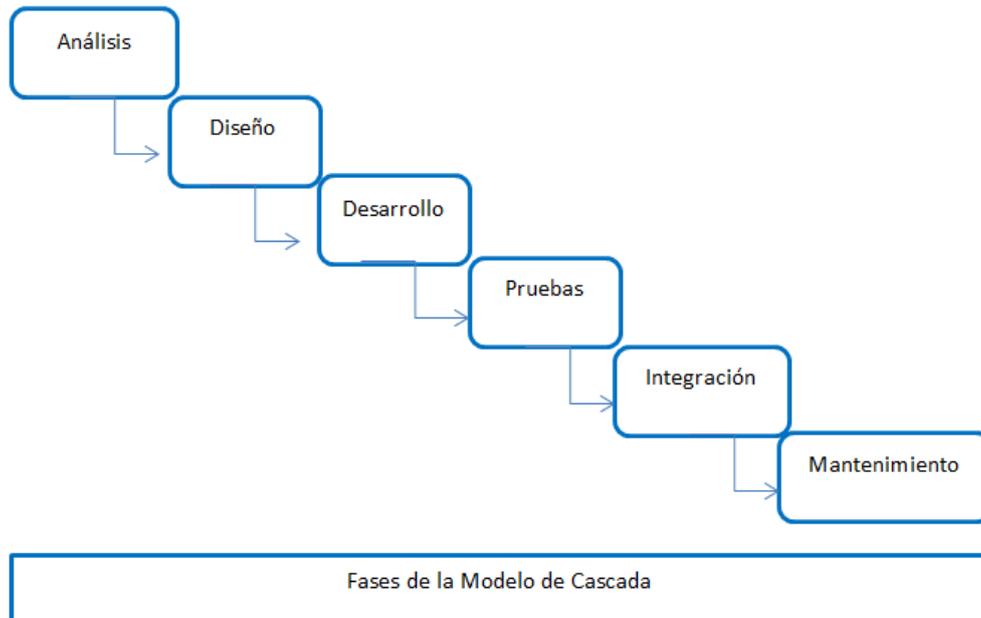


FIGURA 4: Metodología modelo de cascada.

Fuente: Autoría propia a partir de información recopilada.

Metodología de Prototipos

La metodología de prototipos se basa fundamentalmente en la creación de un modelo prototipo del original con versiones incompletas para su validación y aceptación por parte del usuario.

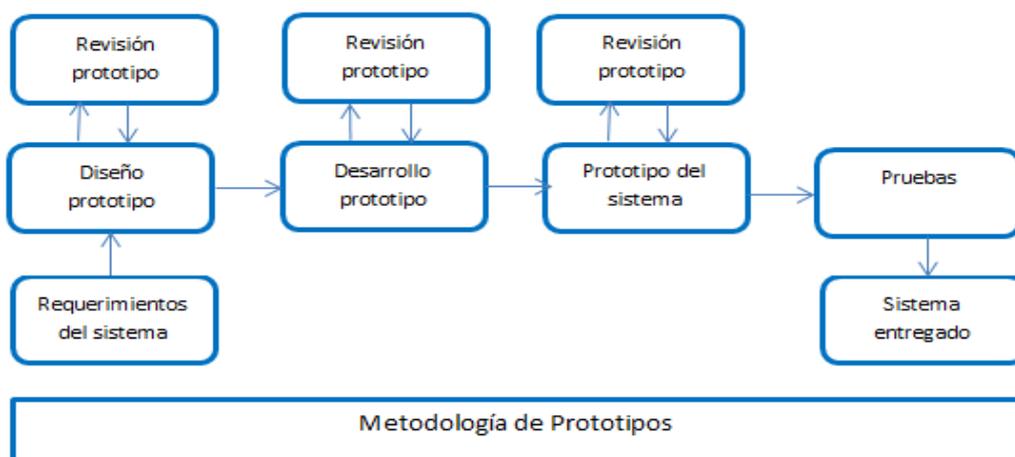


FIGURA 5: Metodología de prototipos.

Fuente: Autoría propia a partir de información recopilada.

Metodología Incremental

La metodología incremental combina métodos iterativos y líneas con el objeto de reducir tiempos, riesgos posibles durante el desarrollo, y generando pequeños módulos de desarrollos los mismos que deberán incrementar sus soluciones en cada iteración presentada al usuarios, de tal manera que si el usuario solicita un cambio durante un módulo permite el cambio en esta fase.

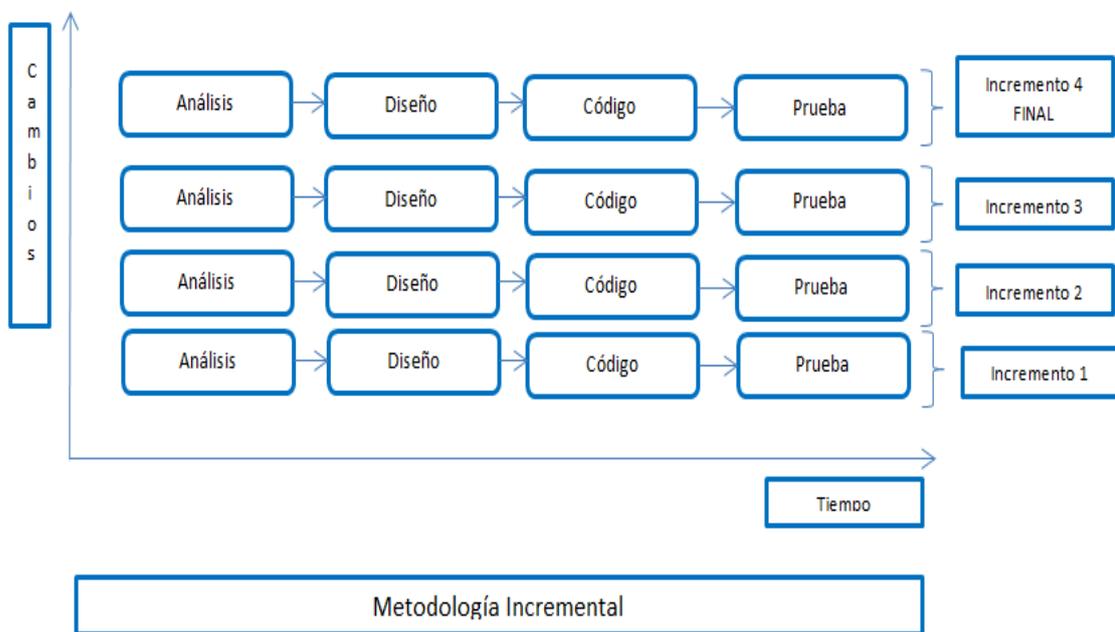


FIGURA 6: Metodología Incremental.

Fuente: Autoría propia a partir de información recopilada.

Metodología en Espiral

Es una metodología que está formada por ciclos, los mismos que se van repitiendo en forma espiral hasta obtener la meta o el producto desarrollado iniciando desde el centro con todos los errores encontrados hasta entregar el sistema en su último ciclo y con la validez aprobada. En cada ciclo se tiene 4 actividades a desarrollarse.

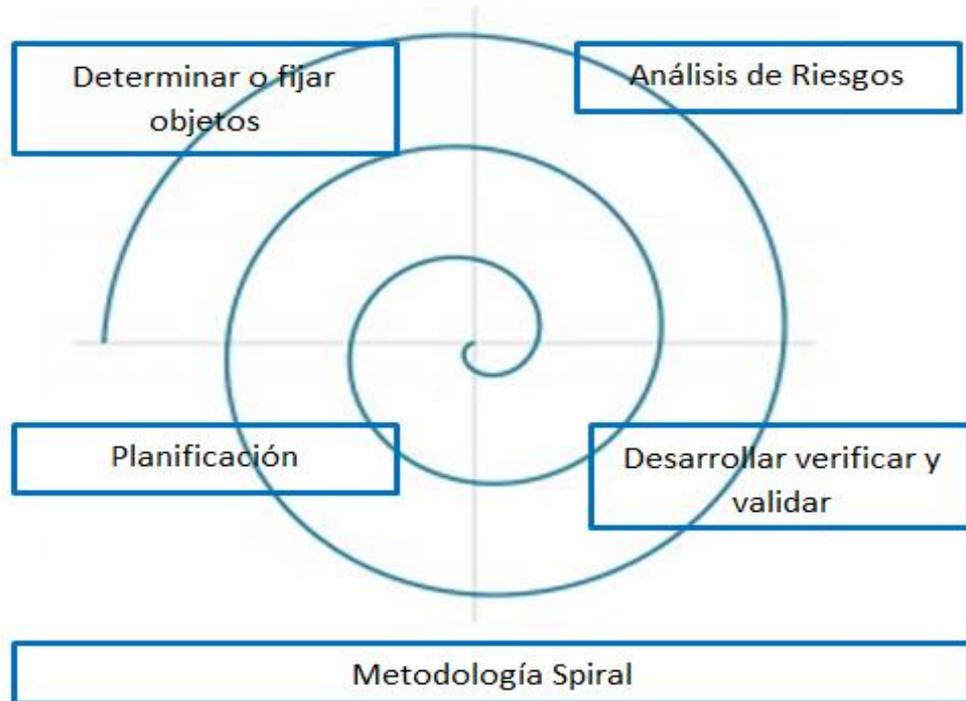


FIGURA 7: Metodología Incremental.

Fuente: Autoría propia a partir de información recopilada.

Metodología RUP¹⁵

Esta metodología estándar es la más utilizada en el desarrollo de un sistema, no es una metodología cerrada permite adaptarse a las necesidades de cada uno de los proyectos y de las organizaciones auspiciantes.

Esta metodología entrega productos tangibles de los procesos como de los modelos de casos de uso, clases, código fuente entre otra información efectuada durante el proceso de desarrollo.

Fases de desarrollo:

- Iniciación.
- Elaboración.
- Construcción.
- Transición.

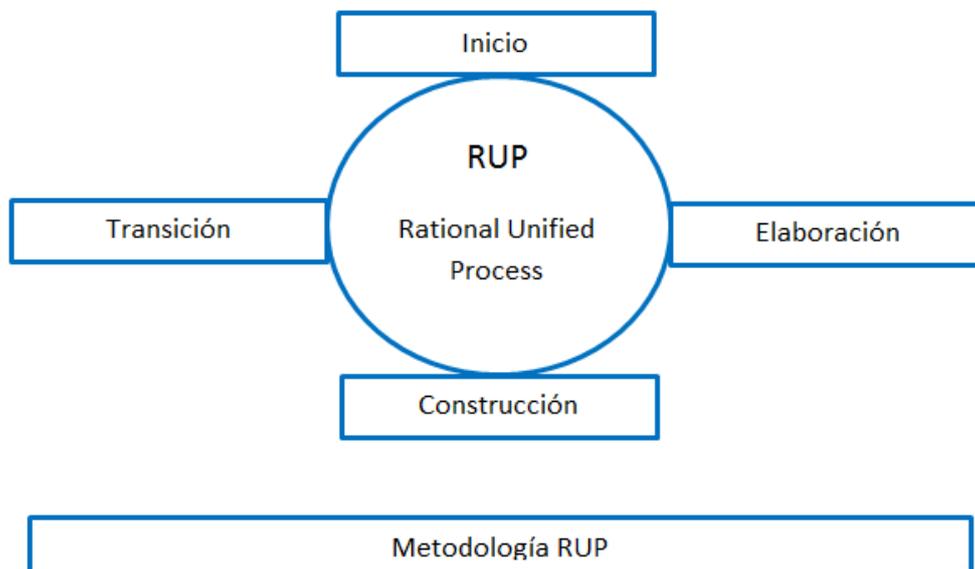


FIGURA 8: Metodología RUP.

Fuente: Autoría propia a partir de información recopilada.

¹⁵ RUP Proceso Unificado de Rational.

Metodologías RAD

Este tipo de metodología permite la creación de prototipos permitiendo reducir requerimientos según el proyecto sin aumentar tiempos, aquí se involucran directamente los usuarios por medio de sesiones de trabajo.

Etapas para el desarrollo RAD.

- Definición.
- Diseño funcional.
- Desarrollo.
- Despliegue.

Metodologías de desarrollo orientado a objetos

Este modelo incluye seis diagramas.

- Clases.
- Objetos.
- Transición y estados.
- Interacción.
- Módulos.
- Procesos.

Metodología de proceso unificado Unified Process

Basado en UML, organizado en cuatro fases.

- Inicio.
- Elaboración.
- Construcción.
- Transición.

Una de las versiones más populares es el RUP (Rational Unified Process)

METODOLOGÍAS AGILES

Estas metodologías son adaptables al cambio durante el desarrollo de un sistema, se caracterizan por definir al momento de la inicialización del proyecto el alcance, los costos en su elaboración y los tiempos de desarrollo, estableciendo la manera de monitorización y control de avances para cumplir con tiempos y respuestas de desarrollo.

Entre las metodologías ágiles tenemos:

- KANBAN.
- Extreme Programming (XP).
- SCRUM.

KANBAN

En esta metodología se trata de mostrar al usuario el proceso del desarrollo, optimizando el flujo de trabajo, indicando el análisis, el desarrollo, las pruebas, la implementación y aceptación del producto, utilizando para ello una matriz de información.

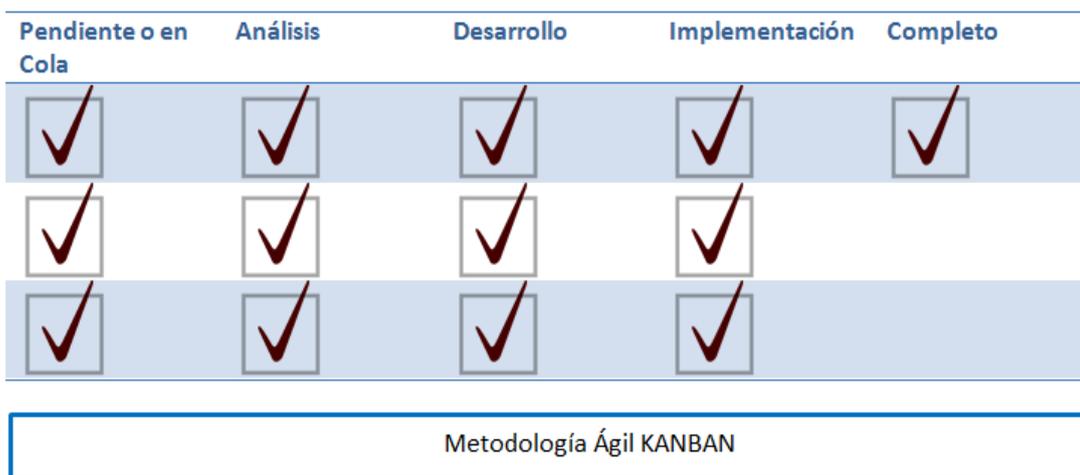


FIGURA 9: Metodología KANBAN.

Fuente: Autoría propia a partir de información recopilada.

Extreme Programming (XP)

Otra metodología ágil es la XP que permite un ágil desarrollo y se caracteriza por su fácil adaptabilidad a los cambios existente en el proceso.

Permite el cambio de diseño forma y resultados, ya que como debilidad es la definición de los requisitos al momento de iniciar el proyecto de desarrollo.

Fases de desarrollo:

- Planificación.
- Diseño.
- Codificación pruebas.

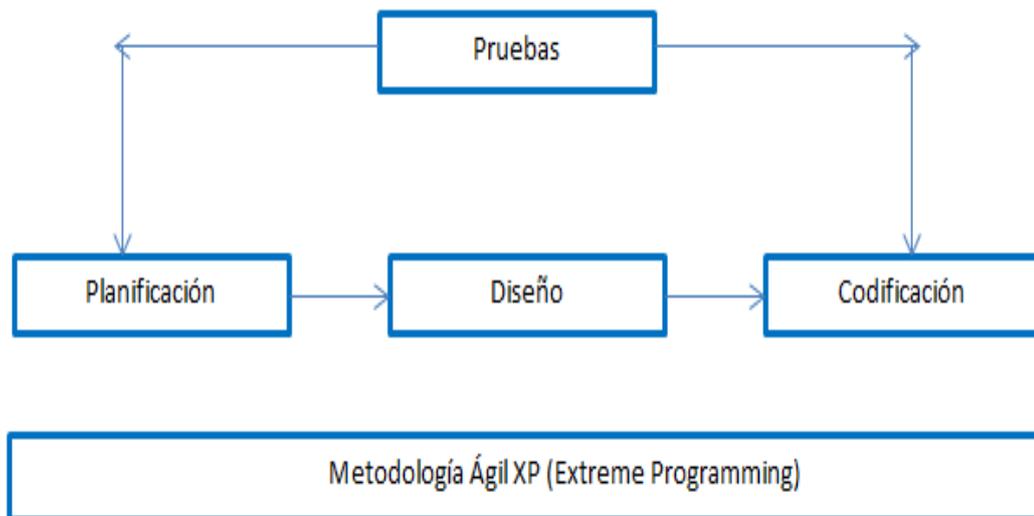


FIGURA 10: Metodología SCRUM.

Fuente: Autoría propia a partir de información recopilada.

2.4.2 ESPECIFICACIONES

METODOLOGÍA A EMPLEARSE EN EL DESARROLLO

METODOLOGÍA SCRUM

Esta metodología permite la gestión ágil de los proyectos a desarrollarse donde su principal objetivo es priorizar las necesidades del cliente, aceptando cambios durante el proceso de desarrollo y a su vez emitiendo resultados frecuentes en un lapso de tiempo establecido.

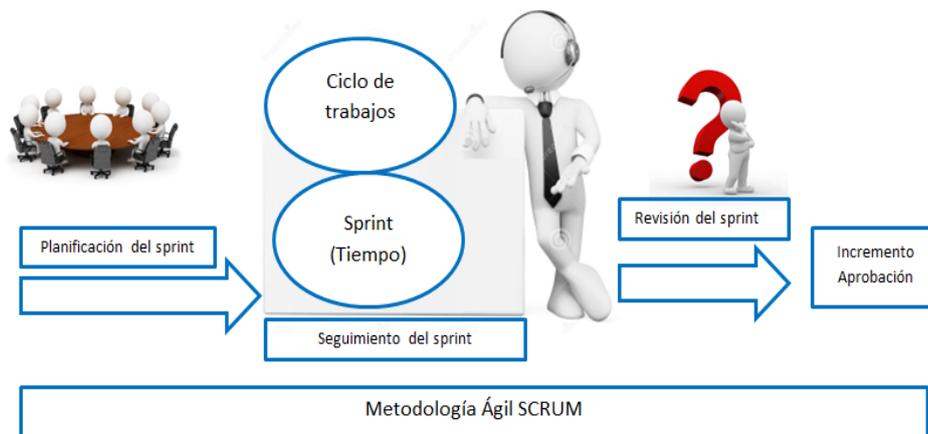


FIGURA 11: Metodología SCRUM.

Fuente: Autoría propia a partir de información recopilada.

De igual manera es un conjunto de resultados enfocados a emitir un resultado general, es adaptable a las organizaciones según las necesidades del propietario del producto, trabaja con equipos organizados y tareas distribuidas.

Terminología

SCRUM utiliza una serie de términos que permiten realizar las actividades de avance en el desarrollo del sistema con la intervención de propietario y su respectiva aprobación.

Sprint

Es la denominación utilizada para un ciclo de trabajo en donde se establece fecha inicio y fecha de entrega, generando un incremento funcional en el desarrollo del sistema.

Incremento

Es el resultado emitido por cada sprint, el mismo que es parte importante para el inicio del siguiente sprint.

Backlog

Es el conjunto de requisitos establecido entre el usuario cliente y el desarrollador.

Roles Principales

- Product Owner: representa la voz del cliente.
- ScrumMaster (o Facilitador): es el que hace que las reglas se cumplan.

Roles Auxiliares

Los roles auxiliares en Scrum son aquellos que no tienen un rol formal y no se involucran con frecuencia en el proceso Scrum, pero deben ser tomados en cuenta.

Stakeholders (Clientes, Proveedores, Vendedores, etc).

Managers (Administradores).

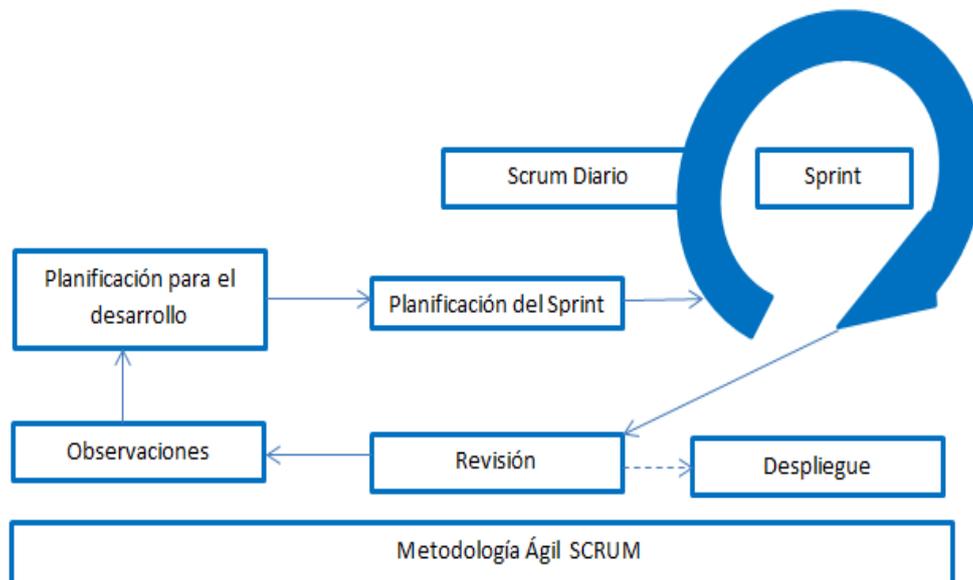


FIGURA 12: Ciclo Sprint.

Fuente: Autoría propia a partir de información recopilada.

Beneficios de SCRUM (agiles, 2015)

TABLA 5: beneficios de la metodología SCRUM.

BENEFICIO	COMO LOGRARLO?
Gestión regular de las expectativas del cliente.	Lista de requisitos priorizada. Demostración de los resultados de proyecto en cada iteración.
Resultados anticipados ("time to market").	Priorización de requisitos por valor y coste.
Flexibilidad y adaptación.	Replanificación en el inicio de cada iteración.
Retorno de inversión (ROI).	Priorización de requisitos por valor.
Mitigación de riesgos.	Desarrollo iterativo e incremental.
Productividad y calidad.	Mejora continua. Comunicación diaria del equipo. Cada actividad de Scrum siempre tiene la misma duración. Equipo multidisciplinar. Estimación de esfuerzo conjunta. Compromiso del equipo. Demostración de resultados preparados para ser utilizados y velocidad sostenida.
Alineamiento entre cliente y equipo.	Cliente y equipo trabajando "en equipo".
Equipo motivado.	Equipo auto gestionado. Demostración.

Fuente: (Agiles, 2015)

Lenguaje de programación

PHP (Blogger, 2012)

PHP identifica a un lenguaje de programación que nació como Personal Home Page (Página personal). Fue desarrollado por el programador de origen danés Rasmus Lerdorf en 1994 con el propósito de facilitar el diseño de páginas web de carácter dinámico.

Sin embargo PHP en la actualidad está vinculado Hypertext Pre-Processor (Hipertexto pre-procesado). El lenguaje es desarrollado hoy en día por The PHP Group, aunque carece de una normativa formal. La Free Software Foundation, por lo tanto, considera la licencia PHP como parte del software libre.

Ventajas

- Una de las ventajas principales que tiene **PHP** es código libre.
- Tiene una serie de frameworks para simplificar el desarrollo web.
- Soporta la mayoría de los servidores de alojamiento (Linux, Mac, Windows).
- Otra ventaja de PHP es su integración perfecta con HTML y su fácil uso y aprendizaje del lenguaje.
- La base de datos que destaca PHP por su conectividad es MySQL y PostgreSQL, aunque cabe recalcar su capacidad de conexión con la mayoría de motores de base de datos que se utilizan en la actualidad.

Base de Datos

PostgreSQL (S.L., 2015)

Es un sistema de gestión de base de datos relacional orientada a objetos y libre. Es un programa de código abierto, por lo que está dirigido por una comunidad de desarrolladores llamada PGDG¹⁶. Comienza su desarrollo en el año 1982 con el proyecto Ingres en la Universidad de Berkeley. Entre sus principales características tenemos la alta concurrencia, la amplia variedad de tipos nativos, y diversas funciones más específicas.

Ventajas:

- Instalación ilimitada: se puede instalarlo en más ordenadores de los que la licencia permite, ya que no hay costo asociado a la licencia de *software*.

¹⁶ **PGDG**: PostgreSQL Global Development Group.

- Ahorros considerables de costos de operación: PostgreSQL ha sido diseñado para tener un mantenimiento y ajuste menor que los productos de proveedores comerciales.
- Estabilidad y confiabilidad: No se han presentado caídas de la base de datos.
- Extensible: El código fuente está disponible de forma gratuita, para que quien necesite extender o personalizar el programa pueda hacerlo sin costes.
- Diseñado para ambientes de alto volumen: Utilizando una estrategia de almacenamiento de filas llamada MVCC que permite a los accesos de solo lectura continuar leyendo datos consistentes durante la actualización de registros, permitiendo copias de seguridad en caliente.
- Herramientas gráficas de diseño y administración de bases de datos.
- Puede operar sobre distintas plataformas, incluyendo *Linux, Windows, Unix, Solaris y MacOS X*.
- Buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.
- Gran capacidad de almacenamiento.
- Buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta.

Servidor WEB

Apache (Blogsfarm, 2015)

Un servidor Web es un sistema que almacena información y la entrega, cuando es necesario. Todos los sitios web en Internet residen en servidores web. Cuando un usuario se conecta a un sitio, el servidor envía los datos que se muestran en la pantalla. El servidor Web Apache ha estado en servicio desde 1995, y es uno de los más utilizados.

Trabaja con gran cantidad de lenguajes de programación como Perl, PHP y otros lenguajes de script, Java y páginas JSP, teniendo todo el soporte que se necesita para tener páginas dinámicas.

Ventajas

- Apertura: Apache Web Server es un software de código abierto.
- Costo: El servidor web Apache es completamente gratuito y puede ser descargado por cualquier persona.
- Funcionalidad: Apache Web Server tiene un gran conjunto de funcionalidades de gran alcance.
- Soporte: cuenta con una gran comunidad de usuarios de soporte.
- Portabilidad: se puede instalar en una amplia variedad de servidores y sistemas operativos.

2.4.3 LIMITACIONES

Metodología a emplearse en el desarrollo

Metodología SCRUM

Desventajas (Minjarez, 2011)

- Si no existe una fecha definitiva de finalización del proyecto es posible que se siga solicitando, y añadiendo, nueva funcionalidad.
- Si una tarea no está bien definida, los costes de tiempo y dinero estimados del proyecto no serán demasiado exactos. En ese caso, la tarea se puede extender sobre varios Sprint.
- Si los miembros del equipo no están centrados y convencidos, el proyecto nunca se completará o incluso fallará.
- Está bien para proyectos pequeños, de rápido movimiento ya que trabaja bien solo con equipos pequeños.
- Esta metodología necesita solo miembros de equipo experimentados. Si el equipo consiste en gente que son junior, el proyecto no puede ser completado a tiempo.
- Además de los recursos sin suficiente experiencia, la falta de dirección firme pueden llevar a los proyectos a no completarse o incluso fallar.

- La metodología Scrum funciona bien cuando el scrum master confía en el equipo que lleva. Si se practican controles muy estrictos sobre los miembros del equipo, puede ser extremadamente frustrante para ellos, llevando a la desmoralización y el fallo del proyecto.
- Si algunos de los miembros del equipo se marcha durante el desarrollo puede tener un efecto negativo enorme en el desarrollo del proyecto.
- El control de la calidad del proyecto es difícil de implementar y cuantificar a menos que el equipo de test puedan llevar a cabo testeo de regresión después de cada sprint.

Lenguaje de programación

PHP

Desventajas (Ibrugor, 2015)

- La legibilidad del código puede verse afectada al mezclar HTML y PHP.
- Php es un poderoso lenguaje e interprete, ya sea incluido como parte de un servidor web en forma de modulo o ejecutando como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

Base de Datos

PostgreSQL (S.L., 2015)

Desventajas:

- En comparación con *MySQL* es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene *MySQL*.
- Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
- Consume más recursos que *MySQL*.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

Servidor WEB

Apache (LinkedIn, 2015)

Desventajas

- Formatos de configuración no estándar.
- Falta de integración.
- No posee buena administración.
- No existe aún configuración gráfica oficial.

CAPÍTULO III

3 DESARROLLO DE LA APLICACIÓN

3.1 FASE DE ANÁLISIS

3.1.1 ESPECIFICACIÓN DE REQUISITOS DE USUARIO. (PRODUCT OWNER – VOZ CLIENTE)

Requisitos de usuario.- Se entiende por requisitos de usuarios en un sistema a los beneficios, los servicios, las necesidades a cubrir por medio del sistema informático, a desarrollarse.

Se debe establecer las restricciones y límites en el desarrollo del sistema informático.

Requerimientos funcionales y no funcionales

Los requisitos funcionales indican a los clientes lo que debe hacer el sistema, su función en el entorno, su estado.

Los requisitos no funcionales indican como debe ser el sistema, su rendimiento, la interfaz, los procesos de desarrollo, las herramientas de desarrollo, tiempos de entrega, tiempos de respuesta.

Los requerimientos son:

TABLA 6: Roles metodología SCRUM.

ROLES		
NOMBRE	ROL EMPRESA	SCRUM ROLES
Coord. Seguridad Alimentaria	Gerente	Product owner
Coord. Carchi Comercializa	Gerente	Product owner
Desarrollador		Scrum master

Fuente: Autoría propia a partir de información recopilada.

TABLA 7: Sprint general - SACC¹⁷

SPRINT:	1 (GENERAL)		INICIO	DURACIÓN
			Septiembre	6 Meses
SISTEMA:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA			
BLACKLOG ID	TAREA	TIPO	ESTADO	RESPONSABLE
Módulo de Registro y Análisis	• Creación de registros.	Análisis y Diseño	Fase Inicial	Desarrollador
	• Modificación de registros.			
	• Eliminación de registros.			
	• Análisis de prioridad.			
	• Análisis de capacidad.			
	• Análisis de beneficio.			
	• Reporte de especificación y aprobación.			
Módulo de Control	• Creación de registros por beneficio.	Análisis y Diseño	Fase Inicial	Desarrollador
	• Análisis de direccionamiento del beneficio.			
	• Depuración de datos.			
	• Control de beneficios no dúplex.			
	• Reporte general de usuarios más beneficio.			
	• Cálculo de montos por beneficiado.			
Módulo de Costos	• Creación de proveedores – asociaciones.	Análisis y Diseño	Fase Inicial	Desarrollador
	• Creación planillas de emisión.			
	• Análisis de costos.			
	• Ingreso y egreso de productos.			
	• Cálculo de costos subtotal, total.			
	• Registro entrega y recibí conforme.			
Roles involucrados:	Desarrollador	Oscar Iván Pozo Villarreal		
	Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi. Representante de la empresa: Ing. Alex Ramos.		

Fuente: Autoría propia a partir de información recopilada.

¹⁷ SACC Seguridad Alimentaria Carchi Comercializa.

3.1.2 REQUISITOS DE CAPACIDAD Y RESTRICCIÓN (PRODUCT BACKLOG – OBJETIVOS PRIORIZADOS)

Cada módulo contara con límites en su creación y/o desarrollo:

Módulo de Registro y Análisis

Para poner en marcha este módulo es necesario crear los registros, información que será depurada y procesada según las necesidades en el sistema.

- Creación de registros.

Se crea un registro con la información solicitada por el Proyecto Seguridad Alimentaria, información que deberá cumplir los estándares de validación siguientes:

- Campos numéricos.
- Campos de textos.
- Cédulas.
- Duplicidad de información.
- Nacionalidad.
- Sectorización.

Los registros ingresados en la BDD permitirán:

Modificar, eliminar (es un pasivo de la información registrada), realizar búsquedas con diferentes criterios, y la creación de nuevos registros.

Módulo de Control

La información de los beneficiarios favorecidos llegara al módulo de control el mismo que se encargara de asignar un código dentro de cada tipo de beneficio.

Según la información registrada en el módulo anterior se realizara el análisis del bono al cual aplica.

Módulo de Costos

Este módulo deberá solicitar la creación o registro de proveedores, la creación de productos a ser ofertados, calcular los costos de ingresos y ventas generadas en un periodo de tiempo identificando los responsables de cada actividad y mantener un inventario tiempo real.

De igual manera se llevara un registro de los beneficiarios del Proyecto Seguridad Alimentaria quienes son los directos favorecidos de los productos ofertados en el Proyecto Carchi Comercializa.

3.1.3 DEFINICIÓN DE REQUERIMIENTOS FUNCIONALES (PRODUCT OWNER – VOZ CLIENTE)

El sistema informático deberá llevar un registro de los movimientos generados como ingresos de postulantes - proveedores, aprobación en estado activo y pasivo de documentación a ser analizada, reportes por diferentes criterios de búsqueda, exportación de los reportes en *.xls , *.pdf, reportes de información de compras, ventas, por privilegios de usuario entre otros.

TABLA 8: Requisitos funcionales SACC.

REQUISITOS FUNCIONALES
Registrar los postulantes.
Aprobar los beneficiarios.
Sectorizar los beneficiarios.
Reportes según criterios.
Exportación de reportes en archivos .xls, .pdf.
Registrar proveedores, productos ofertados, datos de clientes.
Procesos de pedidos.
Proceso de ingreso de productos según pedido.
Reportes de inventarios de productos.
Proceso de facturación.
Realizar descuentos.
Realizar consultas sobre los productos existentes.
Generar reportes de ventas y compras por periodos, por perfiles de usuario.
Nuevo registro, editar, eliminar (activo o pasivo).
Control de acceso al sistema mediante usuario y contraseña.

Fuente: Autoría propia a partir de información recopilada.

TABLA 9: Requisitos no funcionales SACC.

REQUISITOS NO FUNCIONALES
<p>Seguridad: Para el ingreso al sistema solicitara el usuario y la contraseña lo cual indica que según los perfiles tendrá acceso a la manipulación de cada una de las opciones del sistema, indicando así la seguridad de los datos ingresados.</p> <p>Amigable: la aplicación contara con un diseño amigable de fácil aceptación para el usuario, con opciones claras y de forma ordenada.</p> <p>Confiabilidad y efectividad: Los datos ingresados en el sistema serán confiables con la validación respectiva, permitiendo realizas las búsquedas de forma ágil, y en tiempo real.</p> <p>La BDD será desarrollada en POSTGRESQL.</p> <p>El desarrollo WEB utilizando PHP con JQUERY.</p>

Fuente: Autoría propia a partir de información recopilada.

TABLA 10: Requisitos de implementación SACC.

REQUISITOS DE IMPLEMENTACIÓN
<p>La aplicación funcionara en cualquier sistema operativo que cuente con servicio de internet.</p> <p>La aplicación estará instalada en un servidor web.</p> <p>XAMPP.</p> <p>POSTGRESQL.</p>

Fuente: Autoría propia a partir de información recopilada.

3.1.4 DEFINICIÓN DE ALCANCE (PRODUCT OWER – VOZ CLIENTE)

Módulo de Registro y Análisis

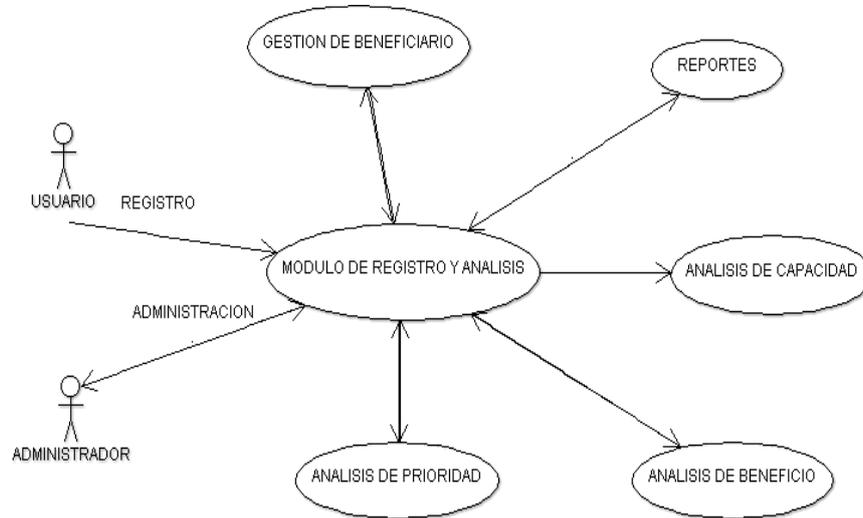


FIGURA 13: Caso de uso registro y análisis

Fuente: Autoría propia a partir de información recopilada.

Gestión de beneficiario

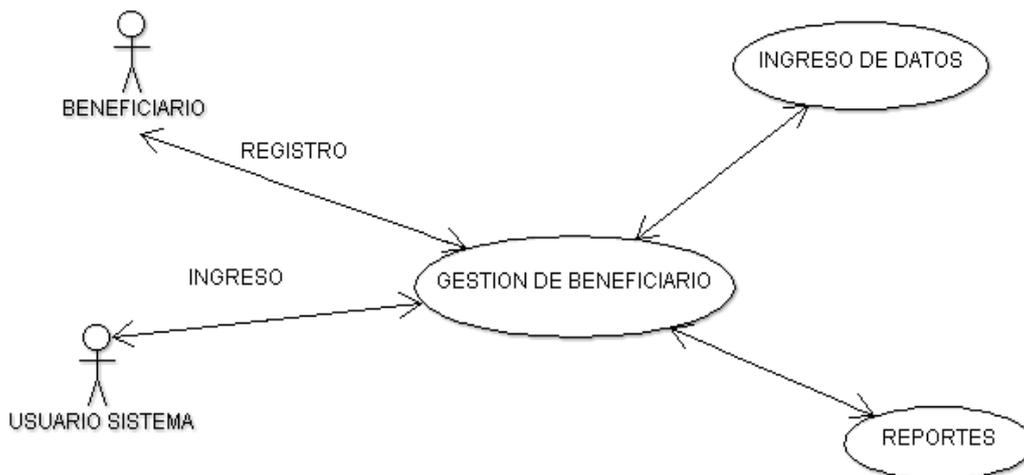


FIGURA 14: Caso de uso gestión de beneficiario.

Fuente: Autoría propia a partir de información recopilada.

Análisis de prioridad

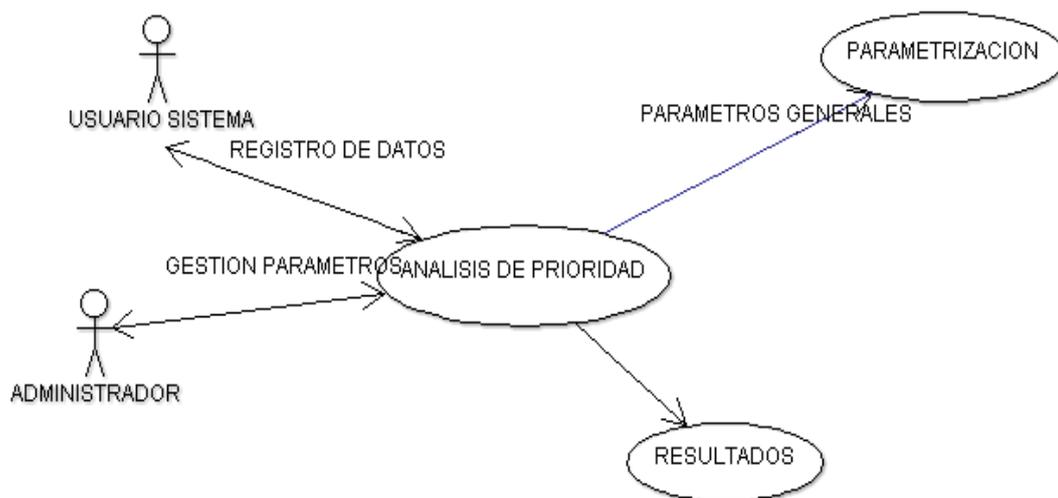


FIGURA 15: Caso de uso análisis de prioridad.

Fuente: Autoría propia a partir de información recopilada.

Análisis de capacidad

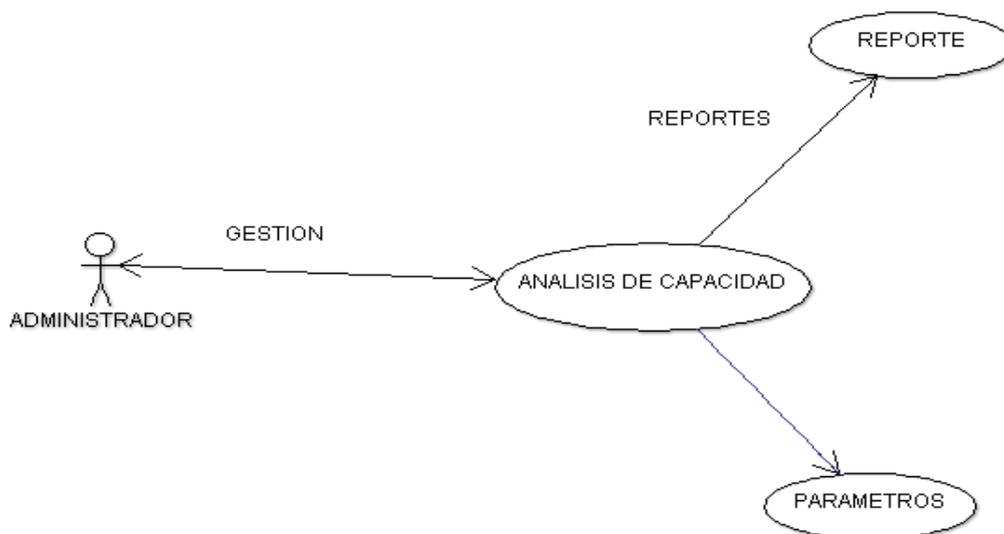


FIGURA 16: Caso de uso análisis de capacidad

Fuente: Autoría propia a partir de información recopilada.

Gestión de beneficio

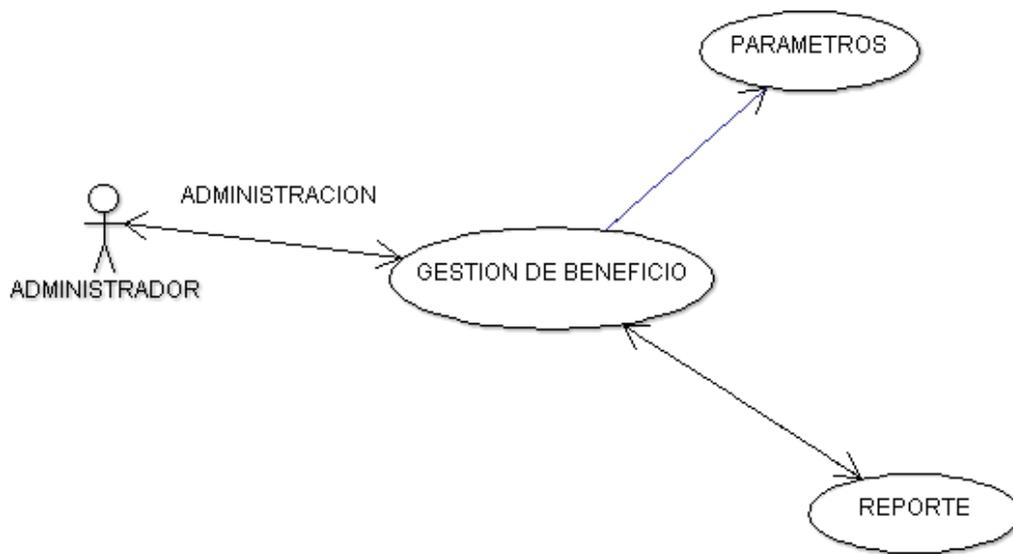


FIGURA 17: Caso de uso gestión de beneficio.

Fuente: Autoría propia a partir de información recopilada.

Costos

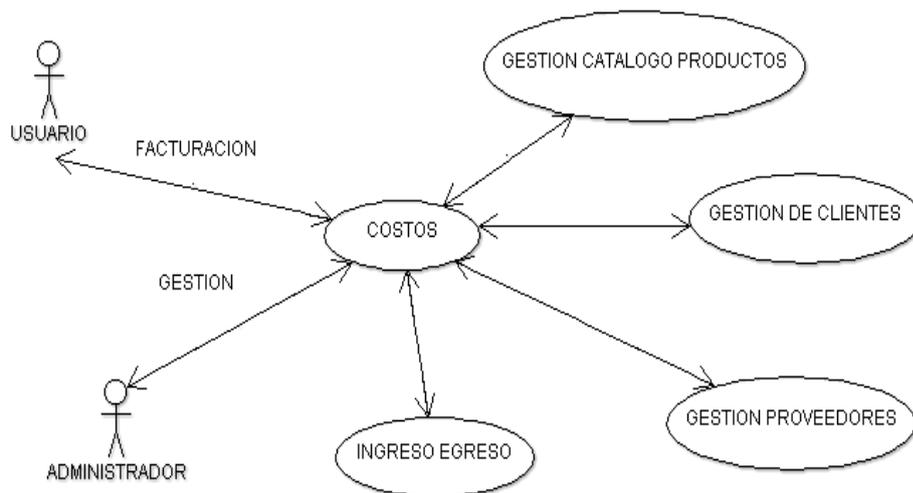


FIGURA 18: Caso de uso costos.

Fuente: Autoría propia a partir de información recopilada.

Gestión de productos

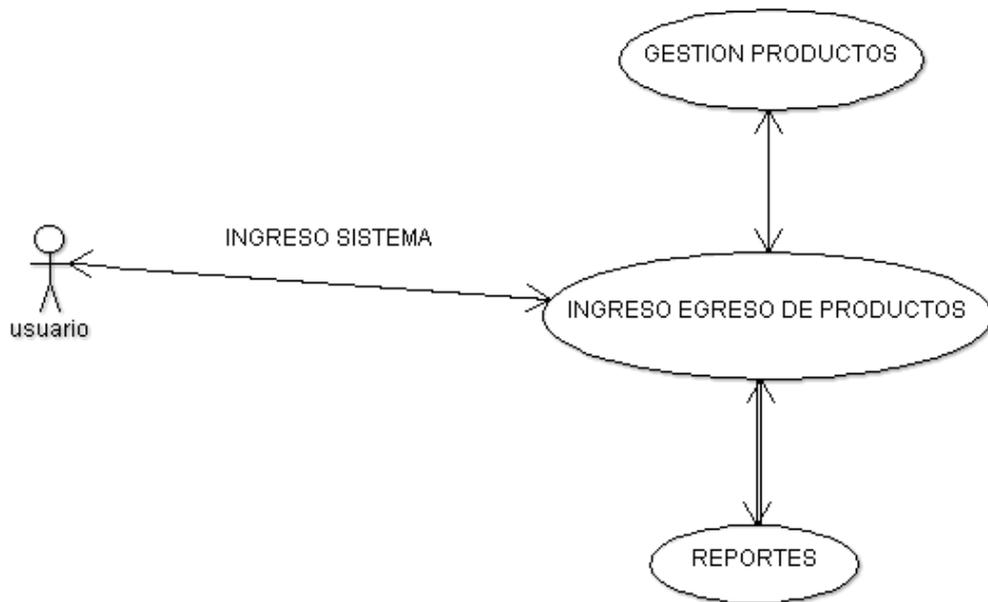


FIGURA 19: Caso de uso gestión de productos.

Fuente: Autoría propia a partir de información recopilada.

Gestión de clientes

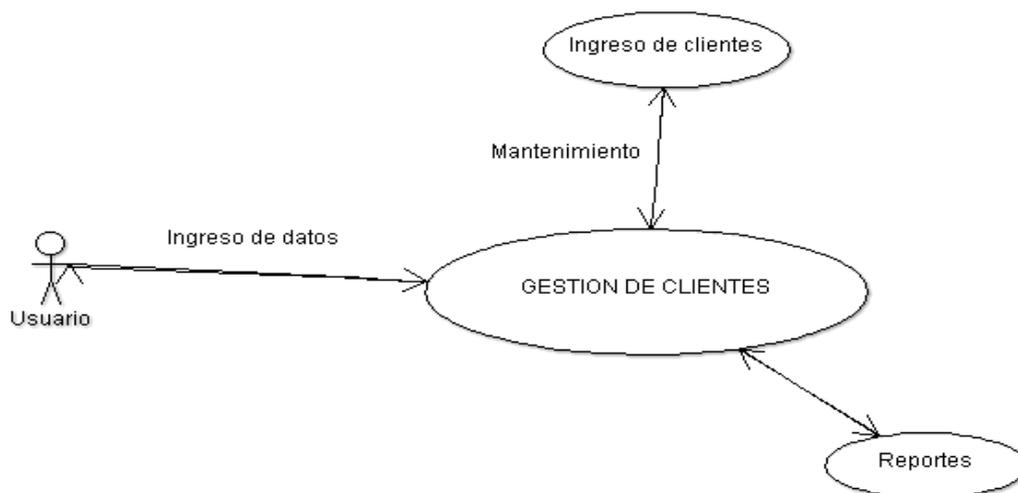


FIGURA 20: Caso de uso gestión de clientes.

Fuente: Autoría propia a partir de información recopilada.

Gestión de proveedores

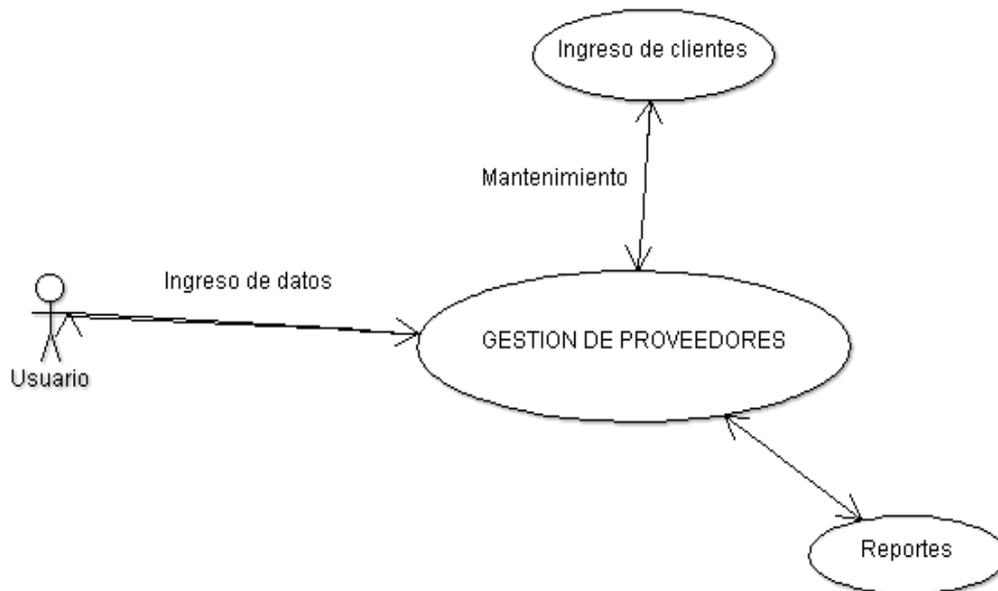


FIGURA 21: Caso de uso gestión de proveedores.

Fuente: Autoría propia a partir de información recopilada.

Ingreso de egreso de productos (Facturación)

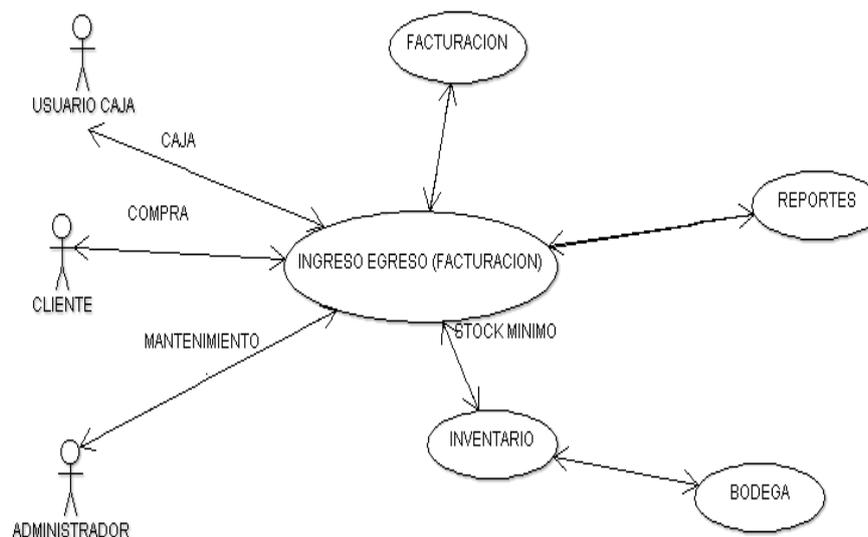


FIGURA 22: Caso de uso gestión de proveedores.

Fuente: Autoría propia a partir de información recopilada.

3.1.5 DEFINICIÓN DE TAREAS POR ITERACIÓN (SPRINT BACKLOG – TAREAS DE ITERACIÓN)

Módulo de registro y análisis

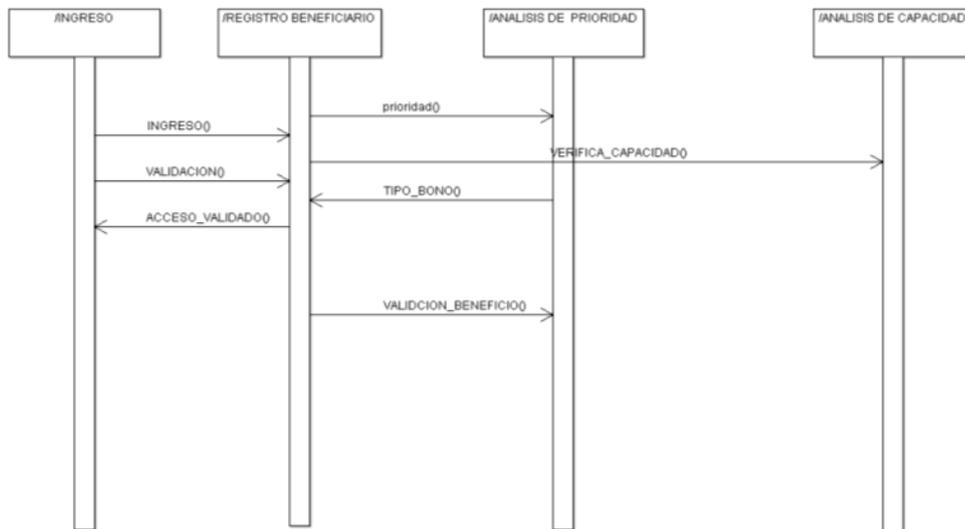


FIGURA 23: Caso de iteración registro y análisis.

Fuente: Autoría propia a partir de información recopilada.

Módulo de costos

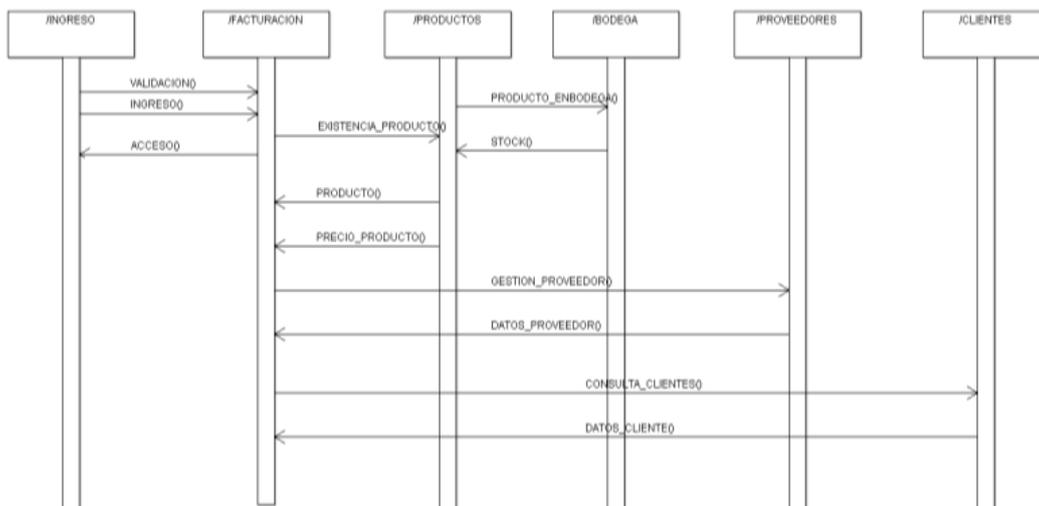


FIGURA 24: Caso de iteración costos

Fuente: Autoría propia a partir de información recopilada.

3.1.6 Documentación de seguimiento de tareas

Para cada una de las fases del desarrollo de la aplicación debemos tener.

TABLA 11: Documentación de Sprint de análisis.

Desarrollo de Sistema[Nombre de la empresa o cliente]:			Gobierno Autónomo Descentralizado de la provincia del Carchi	Versión [n.n]
Planificación:			Fase de Análisis	Fecha[01/09/2014]:
Descripción de la metodología de trabajo:			Proceso Scrum para inicio del desarrollo de la aplicación	
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
1.1	Módulo de Registro y Análisis	<ul style="list-style-type: none"> • Creación de registros. • Modificación de registros. • Eliminación de registros. • Análisis de prioridad. • Análisis de capacidad. • Análisis de beneficio. • Reporte de especificación y aprobación. 	Sprint 1.1.1 Creación de registros. Sprint 1.1.2 Modificación de registros. Sprint 1.1.3 Eliminación de registros. Sprint 1.1.4 Análisis de prioridad. Sprint 1.1.5 Análisis de capacidad. Sprint 1.1.6 Análisis de beneficio. Sprint 1.1.7 Reporte de especificación y aprobación.	Anexo: Módulo de Registro y Análisis
1,2	Módulo de Control	<ul style="list-style-type: none"> • Creación de registros por beneficio. • Análisis de direccionamiento del beneficio. • Depuración de datos. • Control de beneficios no dúplex. • Reporte general de usuarios más beneficio. • Cálculo de montos por beneficiado. 	Sprint 1.1 Creación de registros por beneficio. Sprint 1.2 Análisis de direccionamiento del beneficio. Sprint 1.3 Depuración de datos. Sprint 1.4 Control de beneficios no dúplex. Sprint 1.5 Reporte general de usuarios más beneficio. Sprint 1.6 Cálculo de montos por beneficiado.	Anexo: Módulo de Control

1.3	Módulo de Costos	<ul style="list-style-type: none"> • Creación de proveedores – asociaciones. • Creación planillas de emisión. • Análisis de costos • Ingreso y egreso de productos. • Cálculo de costos subtotal, total. • Registro entrega y recibí conforme. 	<p>Sprint 1.1 Creación de proveedores – asociaciones.</p> <p>Sprint 1.2 Creación planillas de emisión.</p> <p>Sprint 1.3 Análisis de costos</p> <p>Sprint 1.4 Ingreso y egreso de productos.</p> <p>Sprint 1.5 Cálculo de costos subtotal, total.</p> <p>Sprint 1.6 Registro entrega y recibí conforme.</p>	Anexo: Módulo de Costos
Proyecto [Nombre del sistema o proyecto]:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA			
Roles involucrados:	Desarrollador	Oscar Iván Pozo Villarreal		
	Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos		

Fuente: Autoría propia a partir de información recopilada.

3.2 FASE DE DISEÑO

3.2.1 DISEÑO DE ARQUITECTURA PRELIMINAR

En base a la primera reunión se determinó que arquitectura se usara para el desarrollo del sistema, tenemos como base los siguientes aspectos:

- Lenguaje de programación.
- Base de datos.
- Lenguaje de interface.
- Arquitectura de seguridad.

Lenguaje de desarrollo.

Como lenguaje se está usando PHP por su fácil uso y robustez, se usara el lenguaje en una estructura vista controlador lo cual nos permite separar los procesos funcionales del diseño o interface.

Modelo vista controlador (Mestras, 2008) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC¹⁸ propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

¹⁸ **MVC:** Modelo - Vista – Controlador.

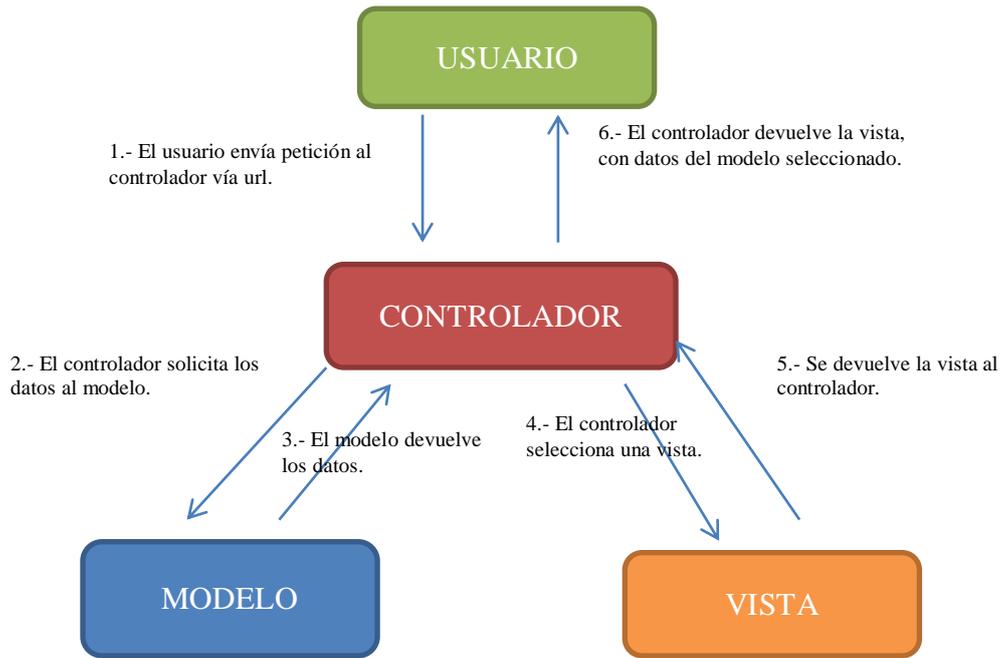


FIGURA 25: Modelo Vista Controlador.

Fuente: (Ríos, 2014)

Arquitectura de lenguaje

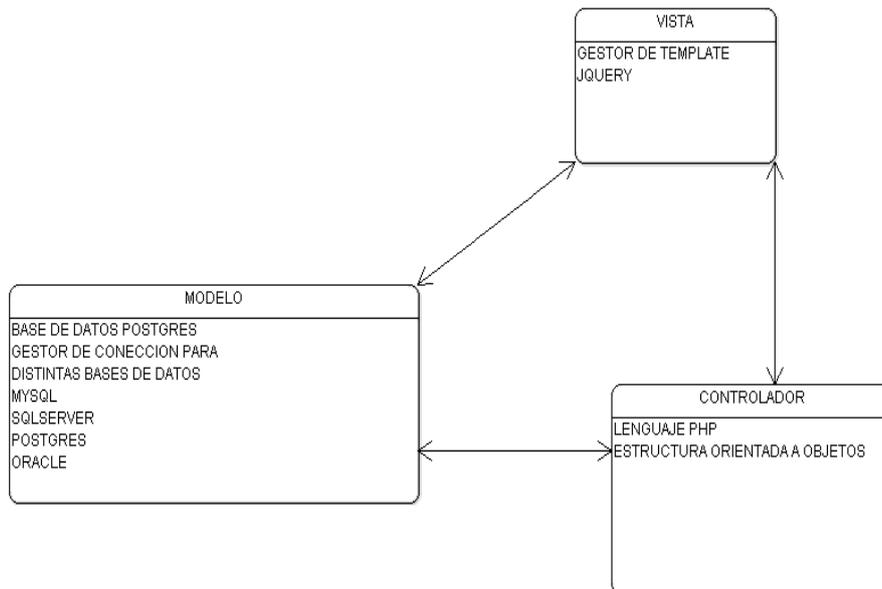


FIGURA 26: Arquitectura del lenguaje.

Fuente: Autoría propia a partir de información recopilada.

En nuestra arquitectura se establecieron varios módulos en el sistema que controlan la conexión a la base de datos lo cual permite una conexión a distintas bases este módulo es controlado por las funciones y objetos que despliegan y ejecutan los procesos y cálculos, este está estrechamente relacionado con la interface en esta parte el sistema puede manejar distintos templates o diseños.

Arquitectura del sistema

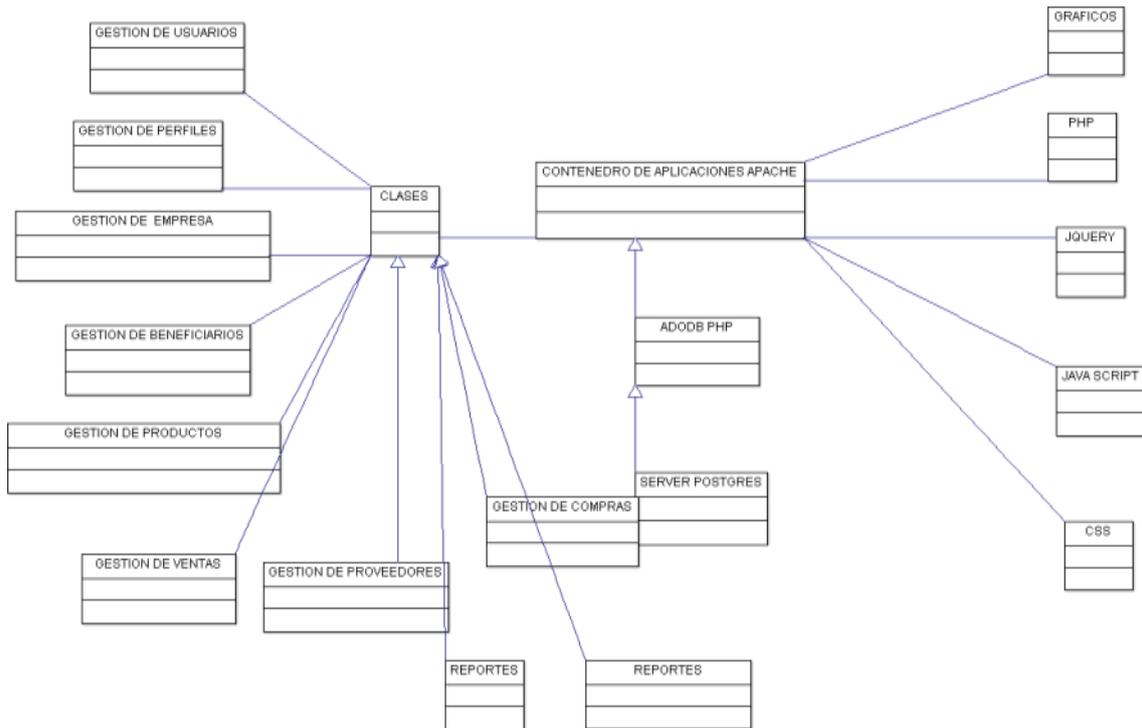


FIGURA 27: Arquitectura del sistema.

Fuente: Autoría propia a partir de información recopilada.

Base de datos

Se desarrolló la base de datos de acuerdo a los requerimientos solicitados por la empresa en la cual se implementara el sistema.

Como base de datos estamos usando PostgreSQL (Martinez, 2010) su arquitectura permite utilizar un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

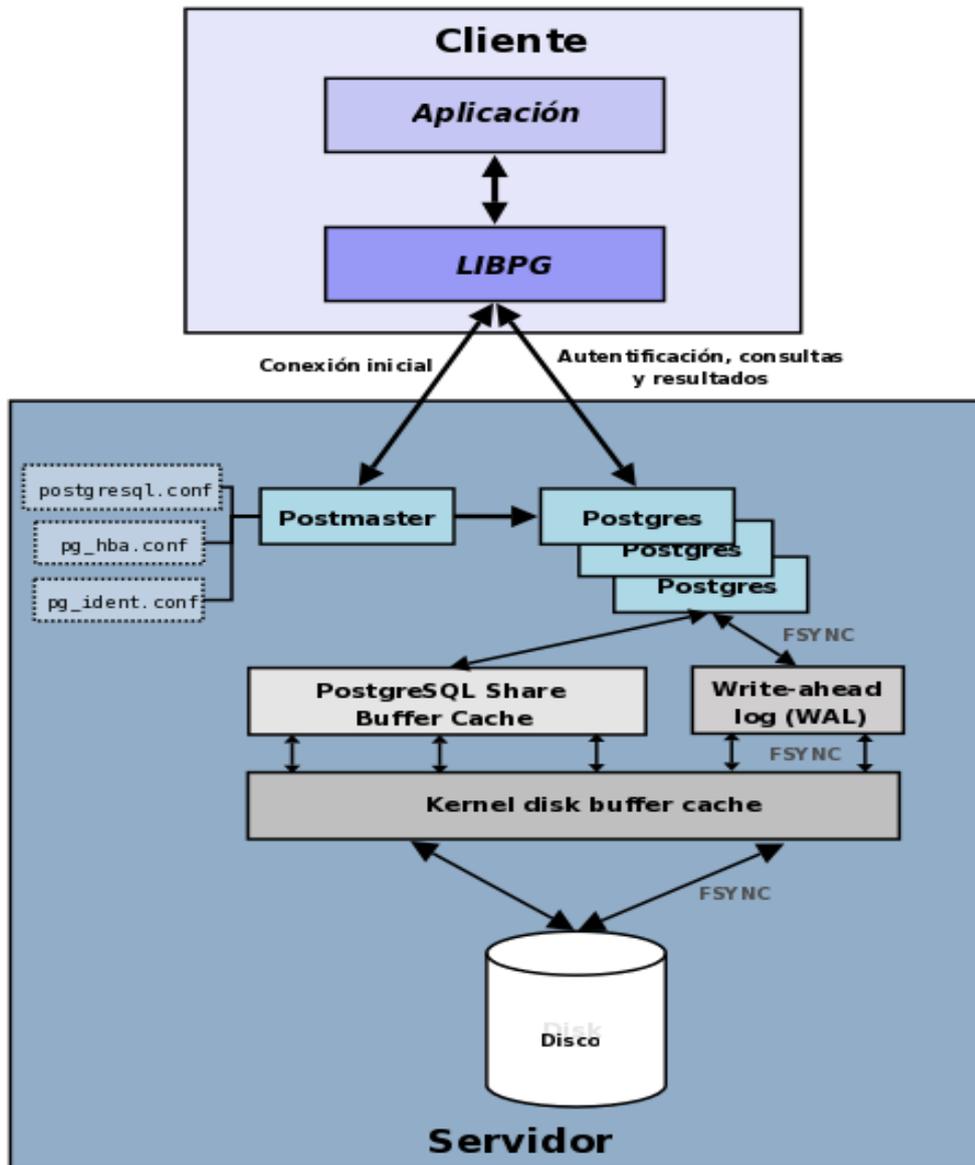


FIGURA 28: Base de datos.

Fuente: (Martinez, 2010)

Nuestro sistema maneja dos subsistemas en una misma arquitectura:

Proyecto Seguridad Alimentaria:

Este proyecto se encarga de hacer el ingreso de beneficiarios y análisis de entrega de bonos de seguridad alimentaria para este proyecto estamos usando el siguiente modelo entidad relación de la base de datos.

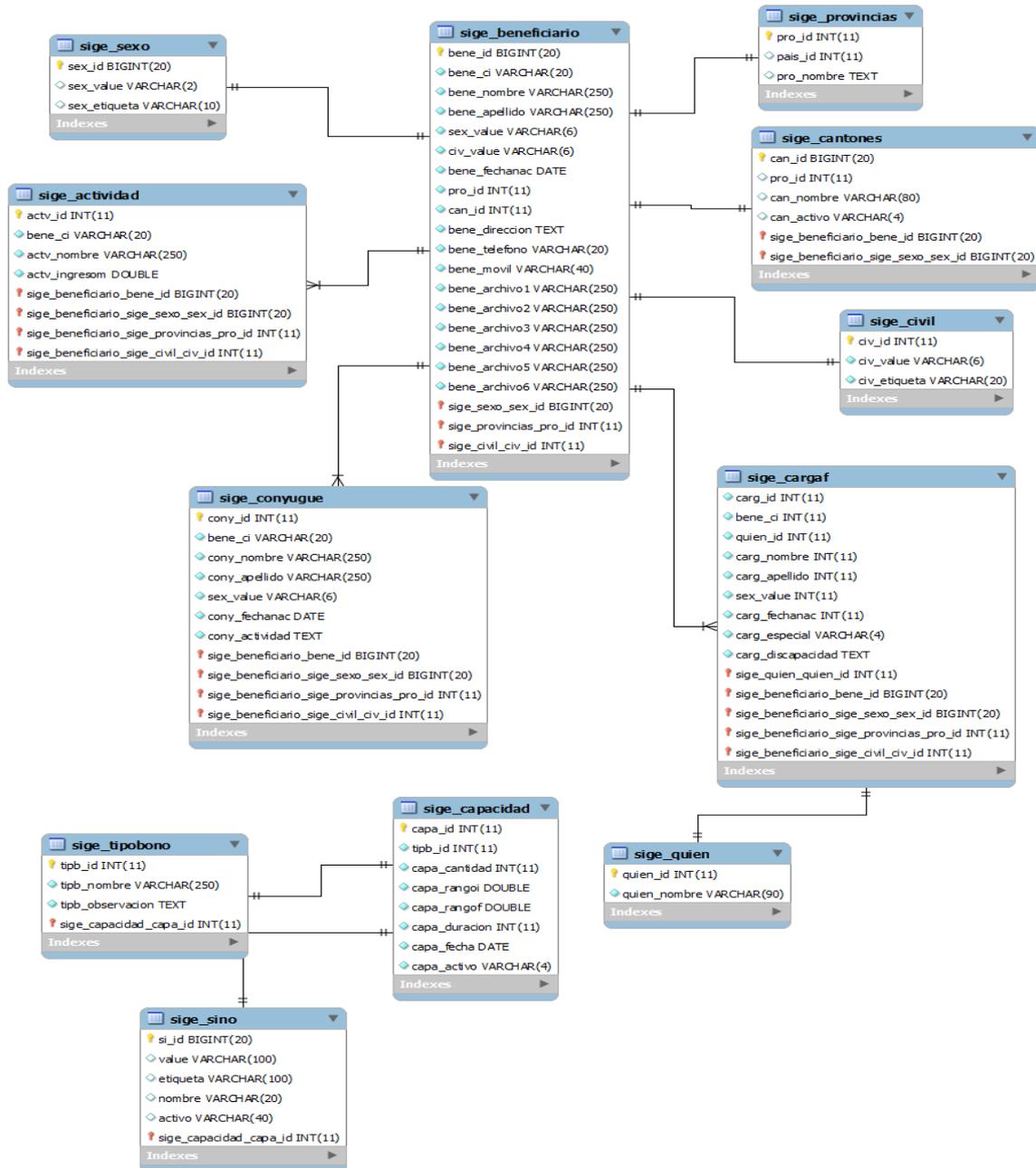


FIGURA 29: Diagrama Proyecto Seguridad Alimentaria.

Fuente: Autoría propia a partir de información recopilada.

Proyecto Carchi Comercializa:

Este módulo se encarga de la facturación y control de inventarios de los distintos productos que administra el concejo.

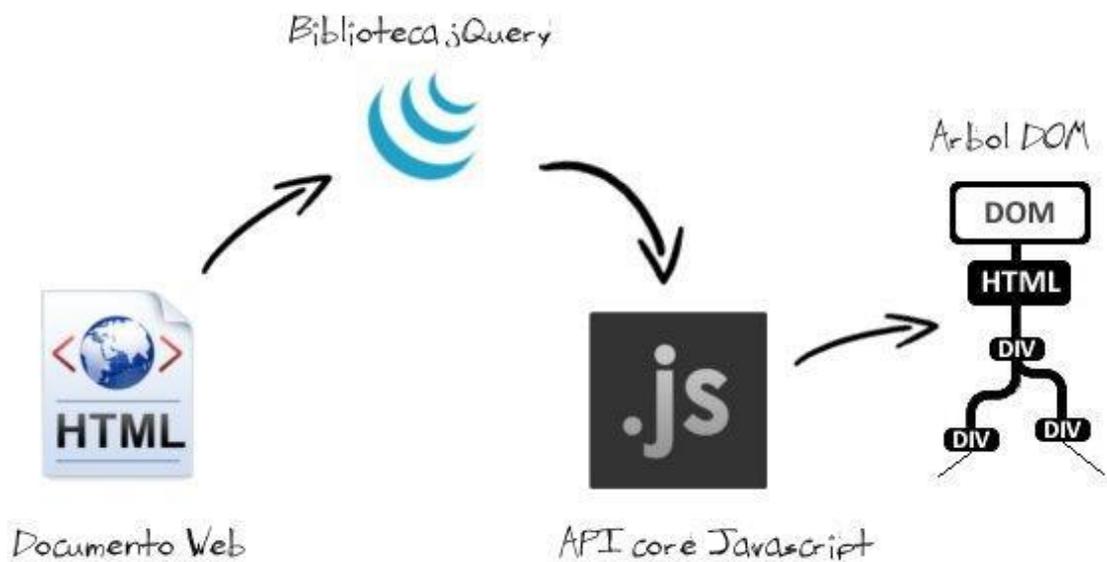


FIGURA 31: Diagrama lenguaje de interface

Fuente: (Garcia, 2012)

Arquitectura de seguridad

Un sistema completamente seguro es prácticamente un imposible, de modo que el enfoque usado con mayor frecuencia en la profesión de seguridad es uno que busque el balance adecuado entre riesgo y funcionalidad. Si cada variable enviada por un usuario requiriera de dos formas de validación biométrica (como rastreo de retinas y análisis dactilar), usted contaría con un nivel extremadamente alto de confiabilidad. También implicaría que llenar los datos de un formulario razonablemente complejo podría tomar media hora, cosa que podría incentivar a los usuarios a buscar métodos para esquivar los mecanismos de seguridad.

El sistema tiene una librería que se encarga del manejo de encriptación de datos control de referencias.

Casos de uso del sistema

Funcionalidad estándar

En el diagrama se detalla el caso de uso general de funcionalidad de todas las pantallas del sistema.



FIGURA 32: Funcionalidad estándar de pantallas.

Fuente: Autoría propia a partir de información recopilada.

Validación de usuarios

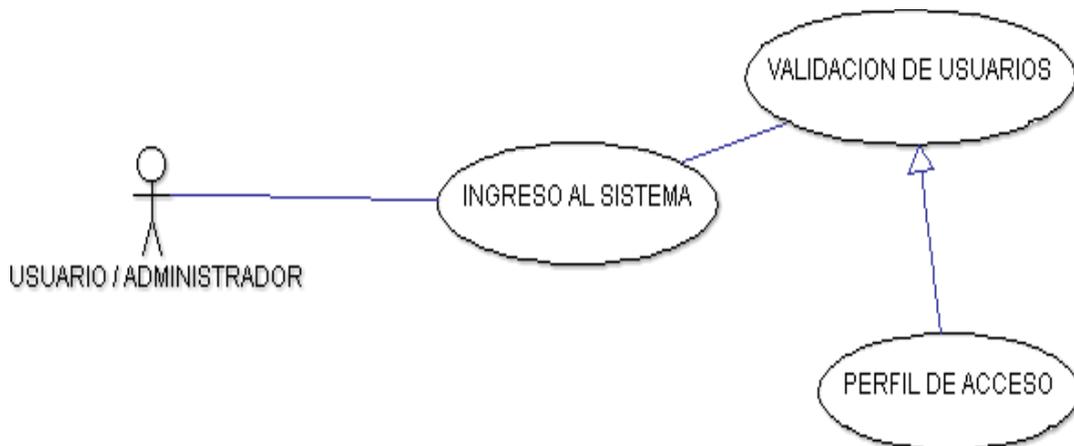


FIGURA 33: Funcionalidad estándar de pantallas.

Fuente: Autoría propia a partir de información recopilada.

Seguridad alimentaria

Beneficiarios

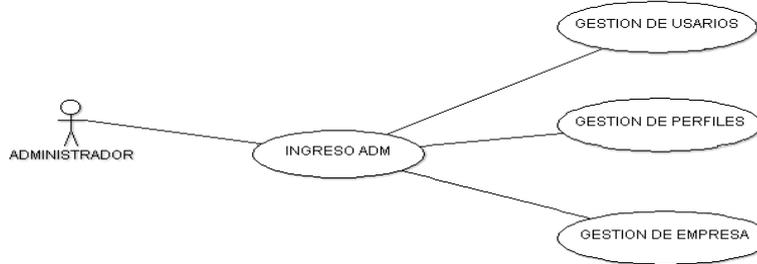
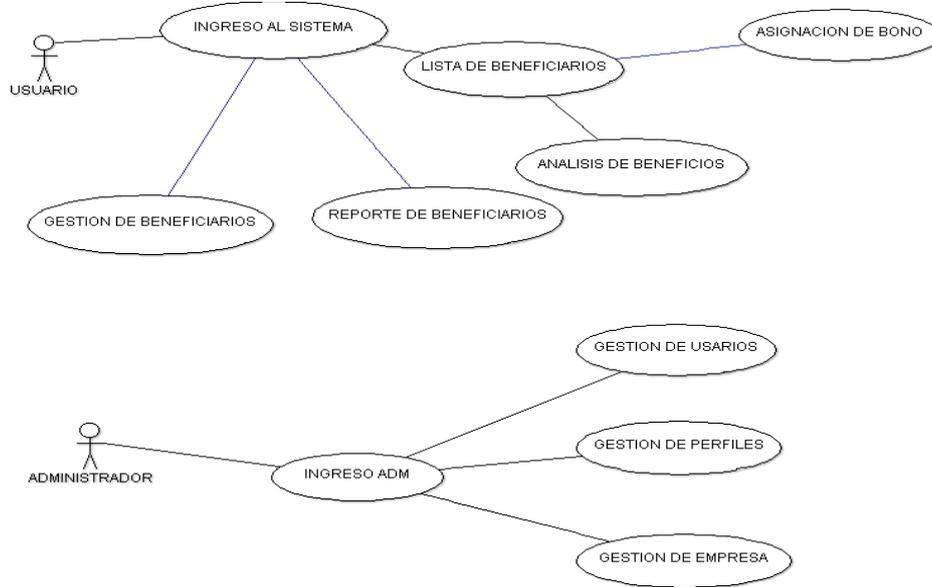


FIGURA 34: Gestión beneficiarios.

Fuente: Autoría propia a partir de información recopilada.

Carchi Comercializa

Ingreso al sistema



FIGURA 35: Ingreso al sistema.

Fuente: Autoría propia a partir de información recopilada.

Proveedores

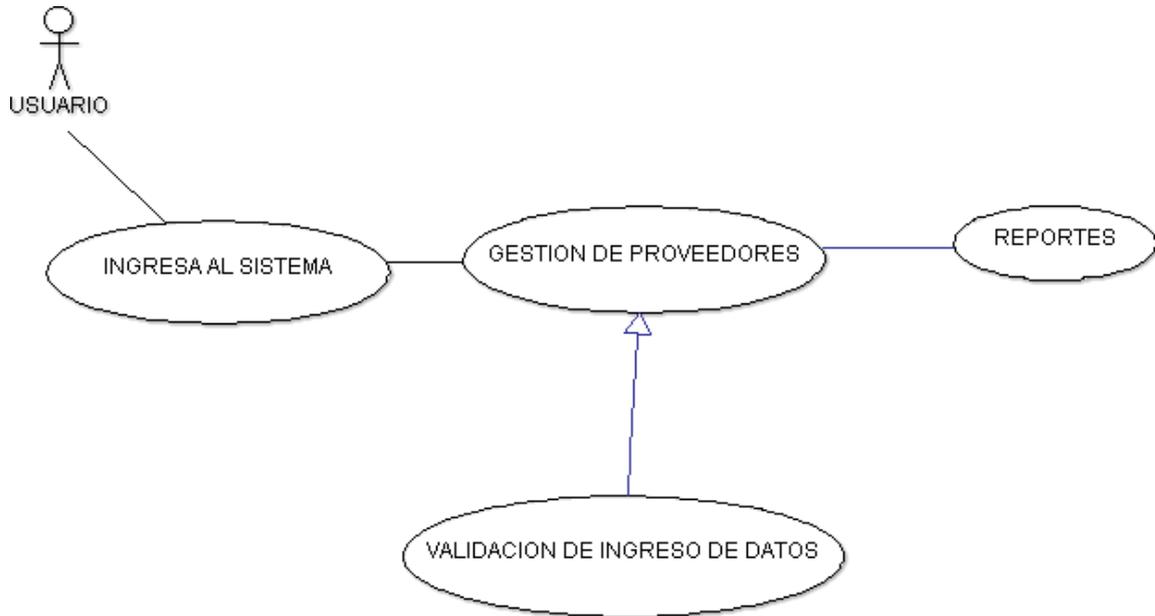


FIGURA 36: Gestión Proveedores.

Fuente: Autoría propia a partir de información recopilada.

Productos

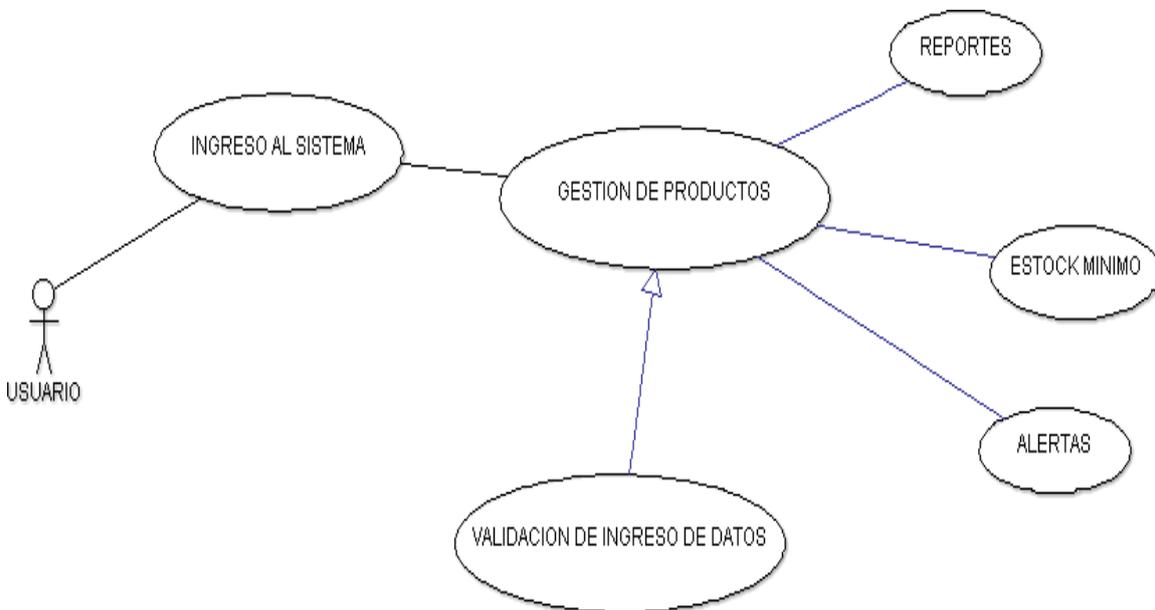


FIGURA 37: Gestión Productos.

Fuente: Autoría propia a partir de información recopilada.

Compras

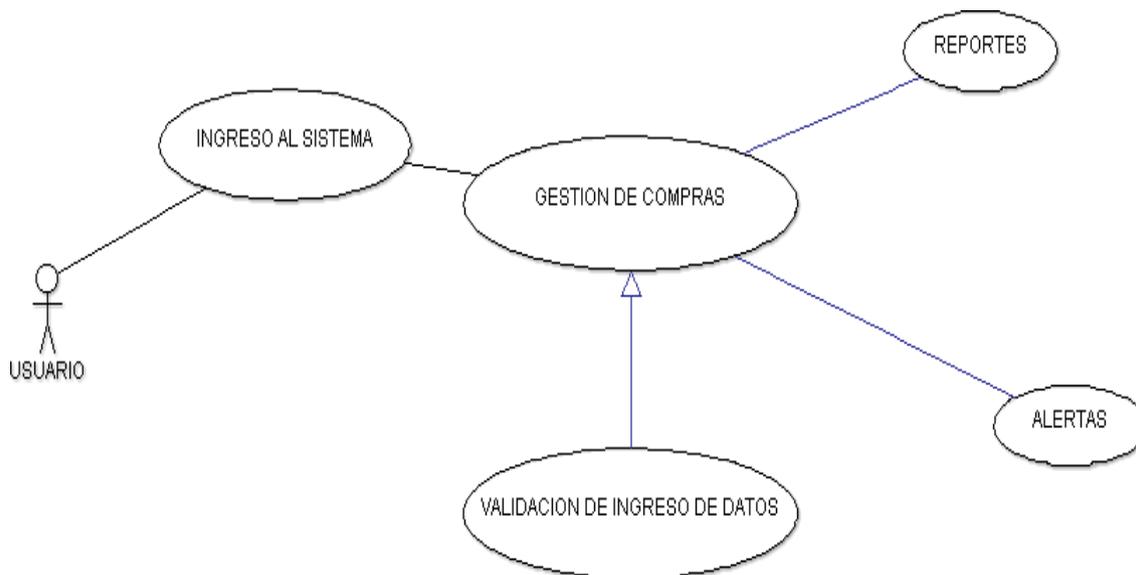


FIGURA 38: Gestión Compras

Fuente: Autoría propia a partir de información recopilada.

Ventas

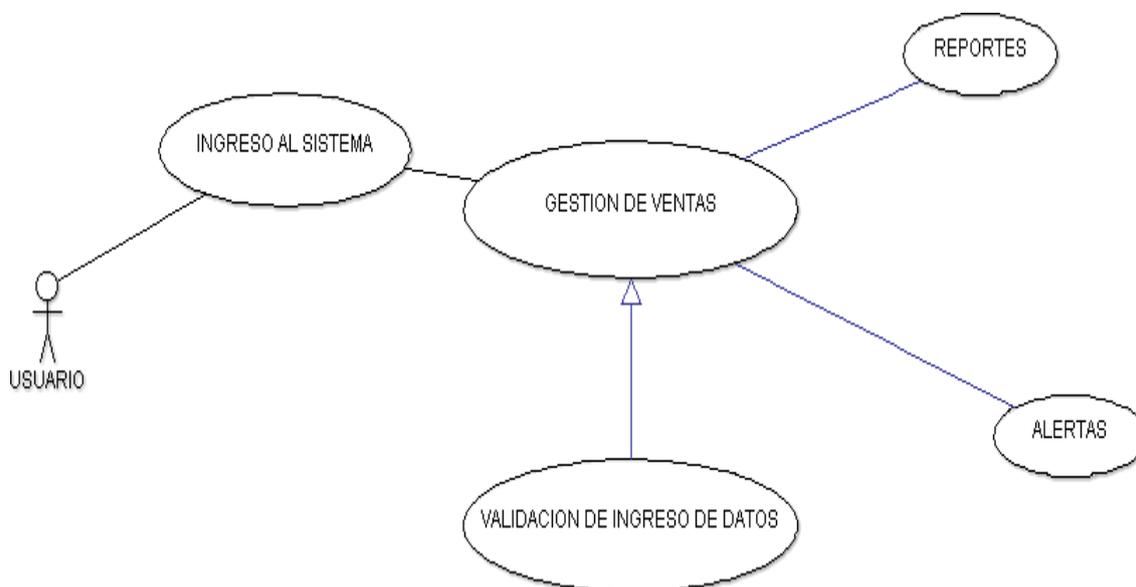


FIGURA 39: Gestión Ventas.

Fuente: Autoría propia a partir de información recopilada.

Reportes



FIGURA 40: Gestión Reportes.

Fuente: Autoría propia a partir de información recopilada.

3.2.2 REVISIÓN Y ACTUALIZACIÓN DE TAREAS POR ITERACIÓN (SPRINT BACKLOG – TAREAS DE ITERACIÓN)

SCRUM propone dividir el desarrollo en pequeñas etapas a superar. Por un lado esto hace que el desarrollo no se vea como un gigante a vencer, sino como pequeños pasos a dar hacia un destino al que nos acercamos de a poco. No desmotiva, y hace factible y visible la evolución de un producto. Estos períodos de 1, 2 o a veces 3 semanas, se llaman SPRINTS (Averbuj).



FIGURA 41: Ciclos de un proceso

Fuente: (Ortega, 2010)

TABLA 12: Spring Parametrización de datos iniciales del sistema perfiles y usuarios.

			Nivel de complejidad					
Id	Requerimientos	Responsable	0	1	2	3	4	5
1	Desarrollo del módulo de usuarios y perfiles.	Desarrollador				4		
	Interface de Ingreso.	Desarrollador				2		
	Validación de Usuarios.	Desarrollador				3		
	Administración de perfiles.	Desarrollador				4		
2	Pruebas del módulo de usuarios y perfiles.	Usuario				1		
	Ingreso de usuarios.	Usuario				1		
	Ingreso de perfiles y pruebas.	Usuario				1		
	Corrección de errores.	Desarrollador				3		

Fuente: Autoría propia a partir de información recopilada.

TABLA 13: Spring Ingreso de proveedores.

			Nivel de complejidad					
Id	Requerimientos	Responsable	0	1	2	3	4	5
1	Desarrollo del módulo de Proveedores.	Desarrollador					4	
	Interface de Ingreso.	Desarrollador					2	
	Validación de Proveedores.	Desarrollador					1	
2	Pruebas del módulo de Proveedores.	Usuario				1		
	Ingreso de Proveedores.	Usuario				1		
	Verificar validaciones.	Usuario				1		
	Corrección de errores.	Desarrollador				3		

Fuente: Autoría propia a partir de información recopilada.

TABLA 14: Spring Ingreso de catálogo de productos.

			Nivel de complejidad					
Id	Requerimientos	Responsable	0	1	2	3	4	5
1	Desarrollo del módulo de Productos	Desarrollador						5
	Interface de Ingreso	Desarrollador						2
	Validación de Productos	Desarrollador						1
2	Pruebas del módulo de Productos	Usuario				1		
	Ingreso de Productos	Usuario				1		
	Verificar validaciones	Usuario				1		
	Corrección de errores	Desarrollador				3		

Fuente: Autoría propia a partir de información recopilada.

3.2.3 REQUISITOS DE INTERFAZ PRELIMINAR

Requisitos de interface, aquí se detallara las distintas interfaces que tiene el sistema:

- Interface de Usuario.
- Interface de Administrador.

Interface de usuario.-

Mantener los colores corporativos de la institución.

Usar un estándar de pantalla igual en todos los módulos del sistema.

Manejar un menú de acceso rápido visible en todas las pantallas.

Interface de Administrador.-

Mantener los colores corporativos de la institución.

Usar un estándar de pantalla igual en todos los módulos del sistema.

Manejar un menú de acceso rápido visible en todas las pantallas.

Seguridad Alimentaria

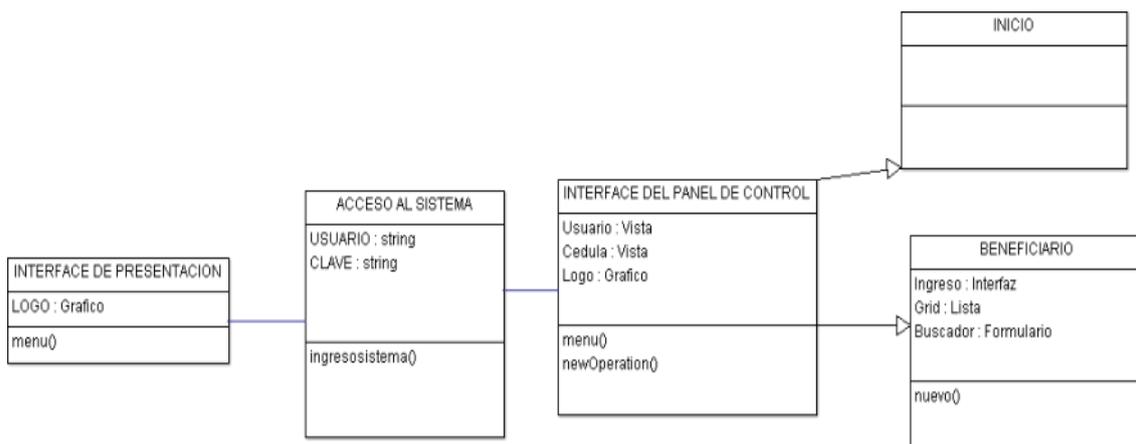


FIGURA 42: Interface Seguridad Alimentaria.

Fuente: Autoría propia a partir de información recopilada.

Carchi Comercializa

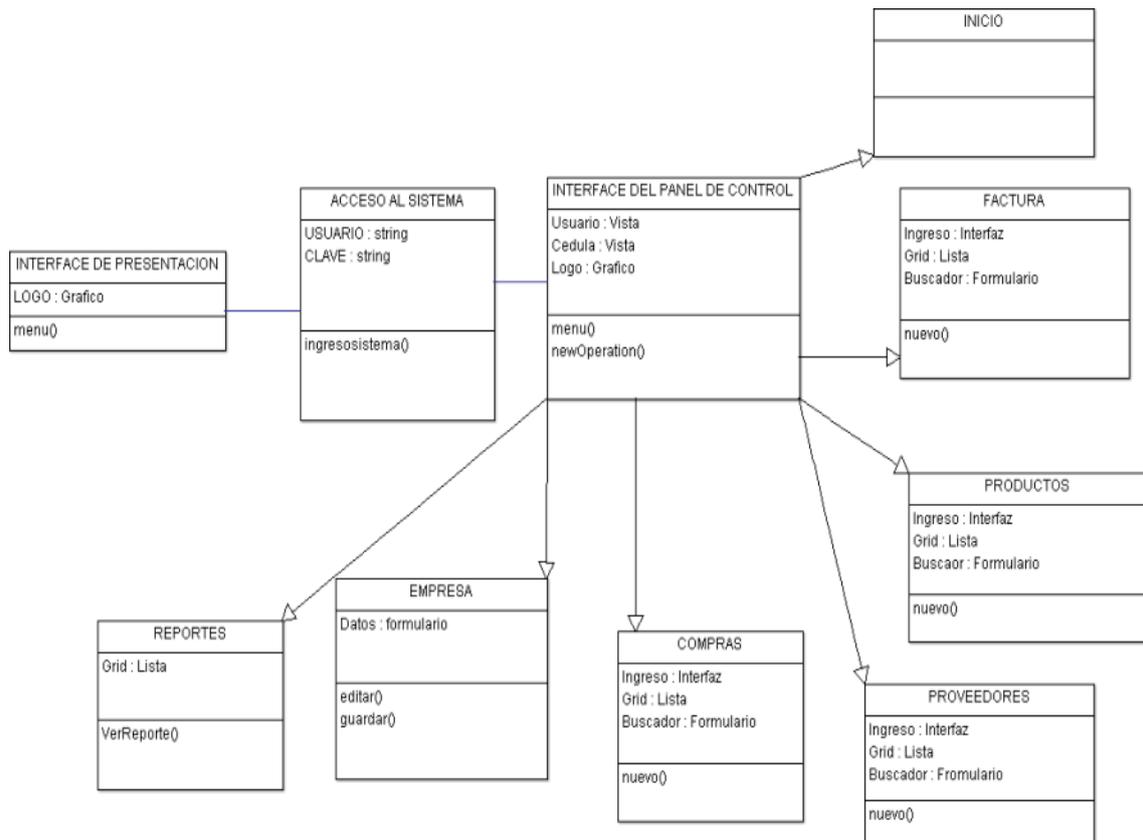


FIGURA 43: Interface Carchi Comercializa.

Fuente: Autoría propia a partir de información recopilada.

3.2.4 ELABORACIÓN DE UN PLAN DE PRUEBAS

En el Plan de Pruebas describe los pasos que se deben seguir para verificar que el sistema construido satisface los requerimientos, es uno de los planes de prueba detallados y corresponde al nivel de pruebas de aceptación del sistema o de la solución. Este plan describe clara y completamente como realizar las pruebas.

Las pruebas de aceptación, involucran al usuario final y pretenden comprobar que la solución cumple con el modelo de negocio para el que fue desarrollado. Detección de defectos del producto entregado y planes de acción para corrección de los mismos.

Alcance del plan de pruebas

Luego de finalizar las pruebas de sistema, el programa se encuentra completamente ensamblado, y se han encontrado y corregido los errores entre los módulos, métodos, clases y objetos. En este punto se comienza con la etapa de las pruebas de validación de requerimientos más conocida como pruebas de aceptación. Éstas se enfocan en las acciones que realiza el usuario además de las salidas del sistema que puedan ser reconocidas por él; dichas acciones y salidas engloban las expectativas del usuario, y están definidas en las especificaciones de los requerimientos del software.

Las pruebas de aceptación, se realizan a los requerimientos funcionales, y a los no-funcionales como facilidad de uso, recuperación, eficiencia, entre otros; y se pretende lograr: corrección, vale decir, carencia de ambigüedad; completitud, es decir, especificación completa y clara del problema; y por último pero no menos importante, consistencia, quiere decir, que no haya requisitos contradictorios.

El plan que a continuación se detalla pretende dar una visión general sobre las actividades a realizar; sobre las pruebas consideradas; además de una explicación global que se consideró para la realización de los documentos a entregar, ya que darán una mayor información relacionada a la evaluación y reportes de este tipo de pruebas.

Áreas que se revisan en el software en el plan de pruebas

- Funcionalidad especificada en el Sistema.
- Manejo de los datos y transacciones involucradas en las funcionalidades del punto anterior.
- Rendimiento al ejecutar las funcionalidades del primer punto.

Categoría de resultados

- **Éxito:** El resultado de la prueba es conforme al resultado esperado.

- **Aceptable:** El resultado de la prueba indica que el sistema difiere de la especificación aceptada pero es aceptable, no son necesarios cambios en la aplicación, pero requiriendo un cambio en la Especificación Funcional.
- **Tolerable:** El resultado de la prueba es incorrecto, la aplicación en prueba trabaja y podría ser aceptada, pero la falla deberá ser rectificada en el periodo de tiempo acordado.
- **Intolerable:** El resultado de la prueba es incorrecto, y la falla debe ser corregida antes de concluir la fase de prueba.
- **Error:** El resultado de la prueba observado es correcto, pero el resultado esperado de acuerdo a los scripts de prueba son incorrectos.

Entorno de las pruebas

- Las pruebas se realizarán principalmente en el laboratorio de Desarrollo.
- El número de computadoras con que se contará será de acuerdo al tipo de prueba, entre una y cuatro computadoras.
- En los laboratorios se cuenta con acceso al Servidor y definición de Datos del Sistema.

Lista de pruebas

Revisar la implementación del caso de uso estándar de funcionalidad.

Gestión alimentaria

Revisar la implementación del caso de uso ingreso al sistema.

Revisar la implementación del caso de uso validación de usuarios.

Revisar la implementación del caso de uso beneficiario.

Revisar la implementación del caso de uso nuevo beneficiario.

Revisar la implementación del caso de uso validación de beneficiario.

Revisar la implementación del caso de uso análisis de beneficio.

Revisar la implementación del caso de uso asignación del bono.

Carchi comercializa

Revisar la implementación del caso de uso ingreso al sistema.

Revisar la implementación del caso de uso validación de usuario.

Revisar la implementación del caso de uso compra productos.

Revisar la implementación del caso de uso ingreso a inventario.

Revisar la implementación del caso de uso venta de productos.

Revisar la implementación del caso de uso proveedores.

Revisar la implementación del caso de uso reportes.

Tipo de pruebas a realizar

Pruebas por Caso de Uso

Para las pruebas de casos de uso se probarán en el siguiente orden: Ingreso de datos, Validaciones, despliegue de información.

Pruebas de integración

Se realizarán de manera implícita al realizar las pruebas del caso de uso.

Pruebas del caso de uso

Se verificará la correcta implementación de los flujos básicos y alternativos de todos los casos de uso a implementar en la iteración.

Tablas de registro de las pruebas

TABLA 15: Prueba.

Prueba	
Objetivo Prueba:	Probar el funcionamiento del flujo básico Validar Usuario.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Logra entrar al sistema.

Fuente: Autoría propia a partir de información recopilada.

3.2.5 AJUSTE DE LA DEFINICIÓN DEL PROYECTO

Después de las pruebas y reuniones iniciales se analizarán cambios del sistema tanto en funcionalidad como en interface para una optimización del software.

3.2.6 AJUSTE DE LA DEFINICIÓN DEL PROYECTO

Después de las pruebas y reuniones iniciales se analizarán cambios del sistema tanto en funcionalidad como en interface para una optimización del software.

Todos los cambios y mejoras debe estar dentro del alcance del proyecto, en casos extremos se podría aumentar o reducir el alcance del proyecto, esto suele suceder si la funcionalidad lo requiere.

Tabla para registro de mejoras y optimización de funcionalidad.

También se lo realiza en base a los casos de uso del sistema.

TABLA 16: Prueba caso de uso.

Caso de uso	
Objetivo Prueba:	Mejorar funcionalidad.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Logra entrar al sistema.

Fuente: Autoría propia a partir de información recopilada.

3.2.7 DOCUMENTACIÓN DE SEGUIMIENTO DE TAREAS DE DISEÑO

Gestión alimentaria

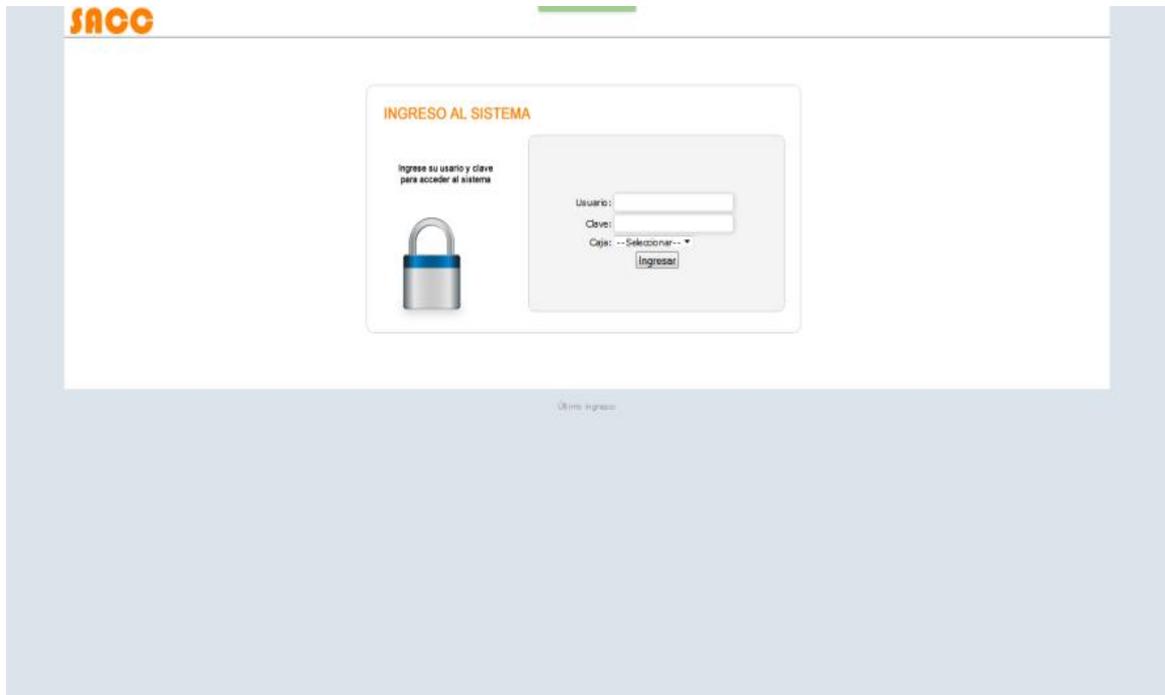


FIGURA 44: Interface Ingreso al sistema.

Fuente: Autoría propia a partir de información recopilada.

TABLA 17: Prueba ingreso al sistema.

Ingreso al sistema	
Objetivo Prueba:	Verificar la interface y diseño.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Colores y logo corporativo correctos.

Fuente: Autoría propia a partir de información recopilada.

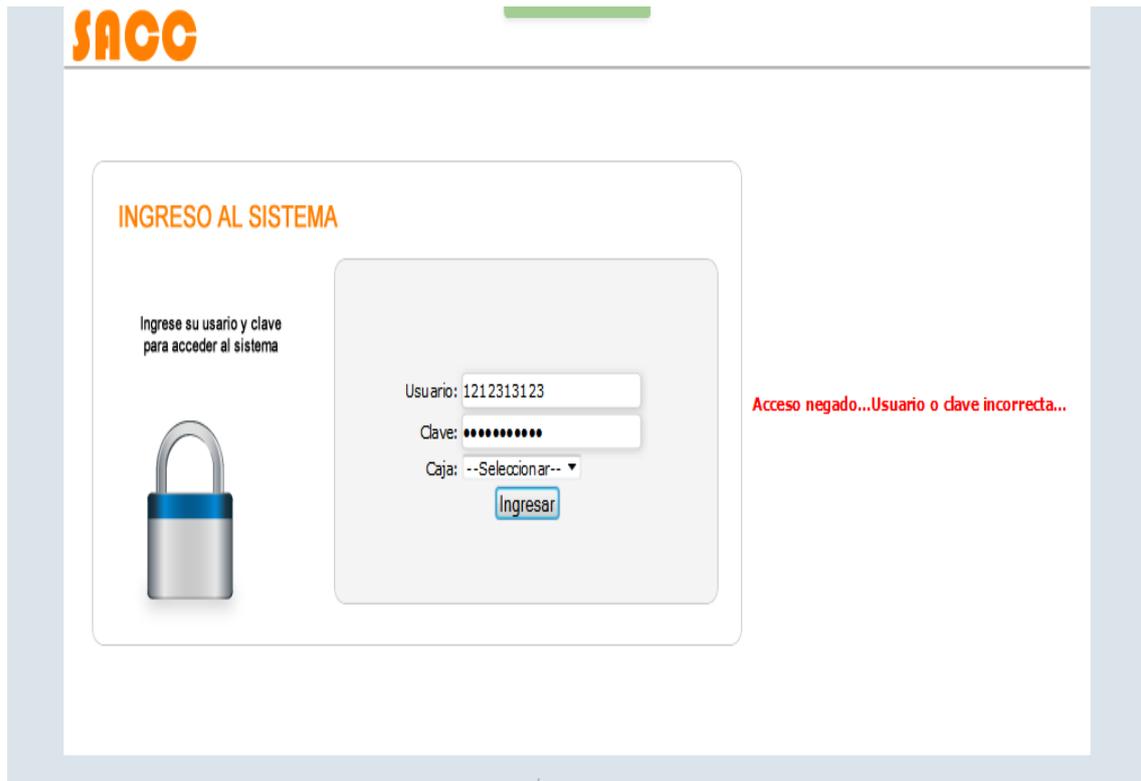


FIGURA 45: Interface Validación de usuario.

Fuente: Autoría propia a partir de información recopilada.

TABLA 18: Prueba validación de usuario.

Validación de usuarios	
Objetivo Prueba:	Probar el funcionamiento del flujo básico Validar Usuario.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Logra entrar al sistema.

Fuente: Autoría propia a partir de información recopilada.

FIGURA 46: Interface gestión Beneficiarios.

Fuente: Autoría propia a partir de información recopilada.

TABLA 19: Prueba gestión beneficiarios.

Beneficiarios	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos en beneficiarios.
Precondición:	-
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

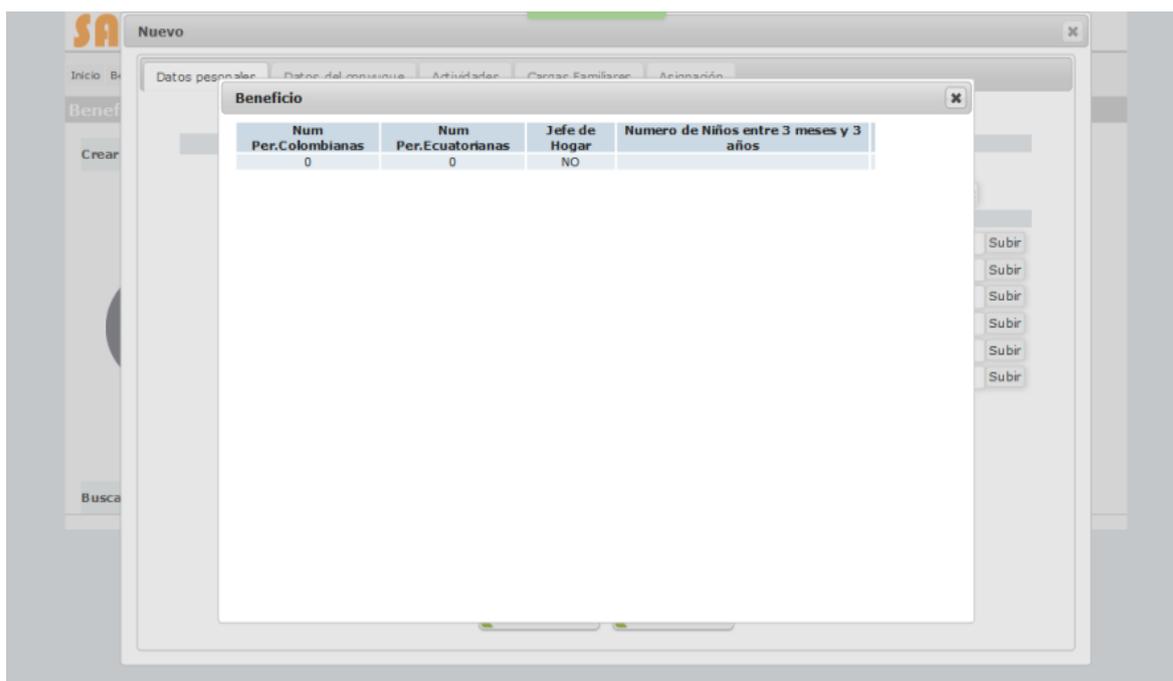


FIGURA 47: Interface Análisis beneficio.

Fuente: Autoría propia a partir de información recopilada.

TABLA 20: Prueba análisis beneficio.

Análisis de beneficio				
Objetivo Prueba:	Verificar que el análisis de beneficio este correctamente basado en los parámetros solicitados.			
Precondición:	TIPO DE BENEFICIO	CONDICIONES PAREJA	CANTIDAD BONO	DURACIÓN
	PRRO Vulnerables	1 persona colombiana	\$ 20 - \$40	12 meses
	PRRO Socorro	2 personas colombianas	\$ 20 - \$120	12 meses
	CLOSAN	1 jefe de hogar - Niños de 3 meses a 3 años de edad	\$ 40	6 meses
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema ejecuta correctamente el proceso de análisis.			
Resultados Esperados:	El sistema debe gestionar los datos correctamente.			

Fuente: Autoría propia a partir de información recopilada.

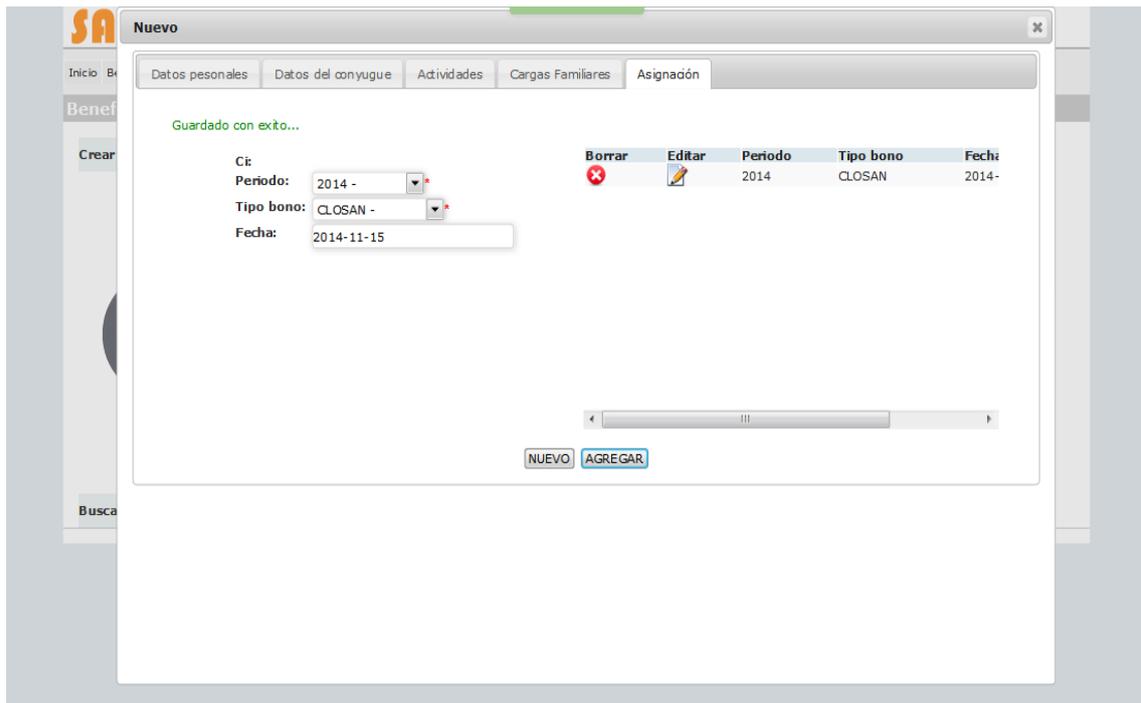


FIGURA 48: Interface asignación de bono.

Fuente: Autoría propia a partir de información recopilada.

TABLA 21: Prueba asignación Bono.

Asignación del bono	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos en asignación del bono.
Precondición:	-
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

Carchi comercializa

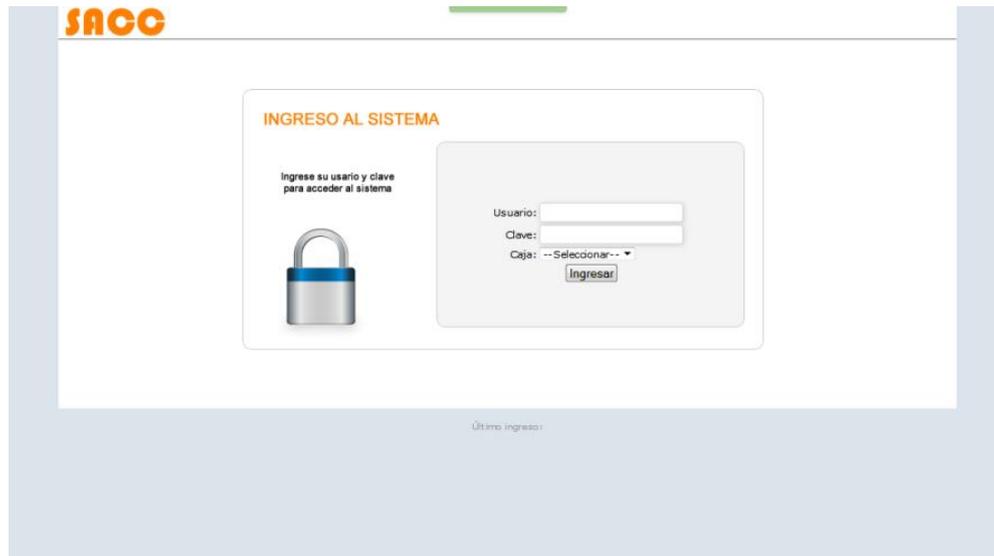


FIGURA 49: Interface Ingreso al sistema.

Fuente: Autoría propia a partir de información recopilada.

TABLA 22: Prueba ingreso al sistema.

Ingreso al sistema	
Objetivo Prueba:	Verificar la interface y diseño.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Colores y logo corporativo correctos.

Fuente: Autoría propia a partir de información recopilada.

TABLA 23: Prueba validación usuarios.

Validación de usuarios	
Objetivo Prueba:	Probar el funcionamiento del flujo básico Validar Usuario.
Precondición:	-
Descripción de la prueba:	Ingresar al sistema con su usuario y su contraseña.
Resultados Esperados:	Logra entrar al sistema.

Fuente: Autoría propia a partir de información recopilada.

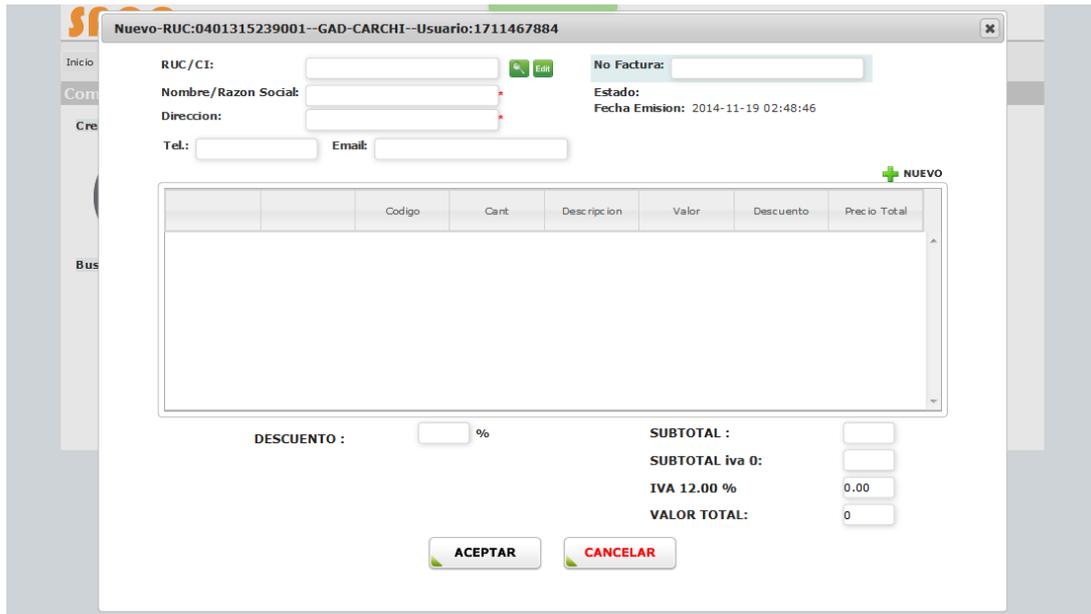


FIGURA 50: Interface compra de productos.

Fuente: Autoría propia a partir de información recopilada.

TABLA 24: Prueba compra de productos.

Compra de productos	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos.
Precondición:	Verificar el ingreso de catálogo de productos Verificar el ingreso de proveedores
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

TABLA 25: Prueba ingreso a inventario.

Ingreso a inventario	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos.
Precondición:	-
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

Nuevo-RUC:----Usuario:

FACTURA

RUC/CI: 1711467884

No Factura: 001-002-0000000232

Nombre/Razon Social: Franklin Aguas

Direccion: Cotacollao

Tel.: Email:

Forma de pago: EFFECTIVO -

Estado:

Fecha Emision: 2014-11-01

Autorizacion:

Control: 2

+ NUEVO

	Cod	Cant	Descripción	Impuesto	Precio Unitario	Descuento	Precio Total
<input type="button" value="Edit"/>	<input type="button" value="X"/>	HUV001	1	HUEVOS	1.00	0.00	1.00
<input type="button" value="Edit"/>	<input type="button" value="X"/>	HUV001	1	HUEVOS	1.00	0.00	1.00

Subtotal : \$2.00

Subtotal IVA 0: \$0.00

Subtotal no obj IVA: \$0.00

IVA 12.00 % \$0.24

Valor Total: \$2.24

FIGURA 51: Interface venta de productos

Fuente: Autoría propia a partir de información recopilada.

TABLA 26: Prueba venta de productos.

Venta de productos	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos.
Precondición:	-
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

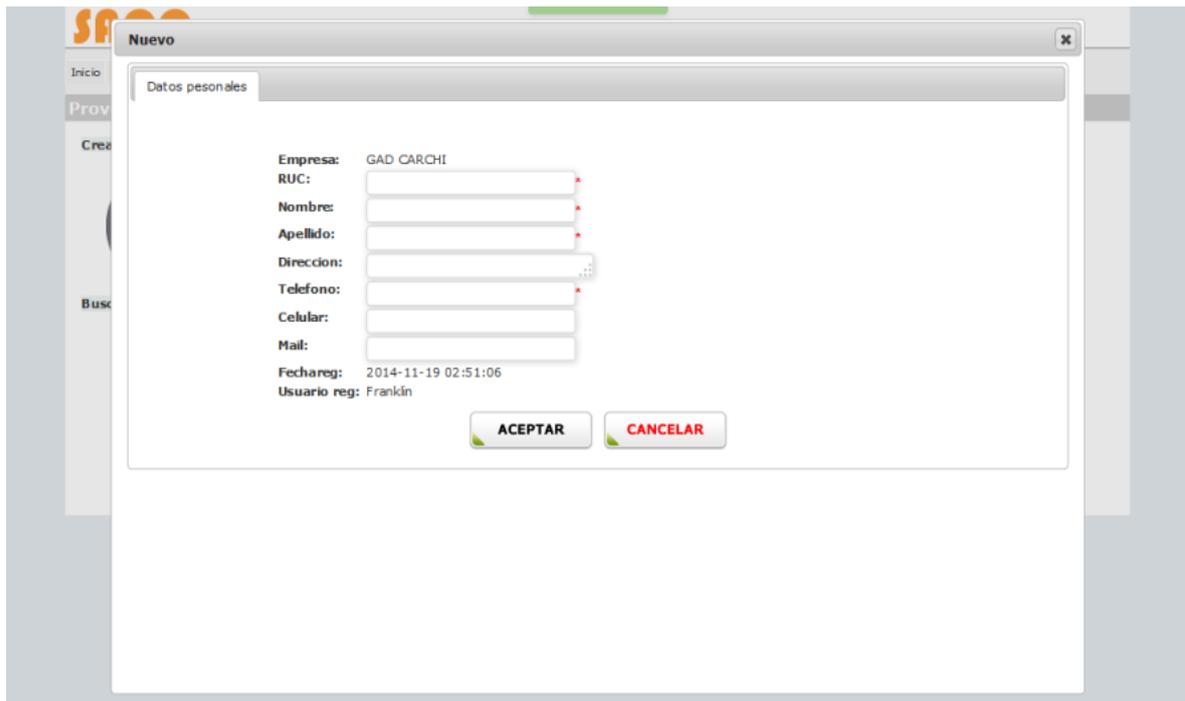


FIGURA 52: Interface ingreso de proveedores.

Fuente: Autoría propia a partir de información recopilada.

TABLA 27: Prueba ingreso de proveedores.

Proveedores	
Objetivo Prueba:	Probar el funcionamiento del flujo básico de ingreso de datos.
Precondición:	-
Descripción de la prueba:	Ingresar datos en los formularios, verificar si el sistema guarda los datos correctamente.
Resultados Esperados:	El sistema debe gestionar los datos correctamente.

Fuente: Autoría propia a partir de información recopilada.

TABLA 28: Documentación de Sprint de diseño.

Desarrollo de Sistema[Nombre de la empresa o cliente]:		Gobierno Autónomo Descentralizado de la provincia del Carchi		Versión [n.n]
Planificación:		Fase de Diseño		Fecha[dd/mm/aaaa]:01/10/2014
Descripción de la metodología de trabajo:		Proceso Scrum para diseño del desarrollo de la aplicación		
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
2.1	Diseño de arquitectura preliminar.	Lenguaje de programación. Base de datos. Lenguaje de interface. Arquitectura de seguridad.	Sprint 2.1.1 Lenguaje de programación. Sprint 2.1.2 Base de datos. Sprint 2.1.3 Lenguaje de interface. Sprint 2.1.4 Arquitectura de seguridad.	Anexos: Diseño de arquitectura preliminar
2.2	Requisitos de interfaz preliminar	Interfaz de usuario - interfaz del administrador.	Sprint 2.2.1 Interfaz de usuario - interfaz del administrador.	Anexo: Requisitos de interfaz preliminar
2.3	Pruebas diseño preliminar	Pruebas con usuarios por módulos en diseño. Categorización de los resultados obtenidos. Rectificación de las observaciones. Ajustes de las observaciones dentro del alcance del desarrollo.	Sprint 2.3.1 Pruebas con usuarios por módulos en diseño. Sprint 2.3.2 Categorización de los resultados obtenidos. Sprint 2.3.3 Rectificación de las observaciones. Sprint 2.3.4 Ajustes de las observaciones dentro del alcance del desarrollo.	Anexo: Pruebas diseño preliminar
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Fuente: Autoría propia a partir de información recopilada.

3.3 FASE DE CONSTRUCCIÓN Y CONTROL DE FUNCIONABILIDAD

3.3.1 DESARROLLO DE INTERFACES



FIGURA 53: Interfaz de diseño.

Fuente: (Marcos, 2014)

Los requerimientos de interfaz son todos aquellos elementos que debe proveer el sistema para permitir la interacción entre el usuario y las funcionalidades que este tiene, con el fin de que en el proceso de diseño se tenga claridad de las interfaces que se deben crear y la relación que debe existir entre ellas.

La interface de usuario debe tener las siguientes características (Zamora, 2013):

- La interfaz representa el punto de encuentro entre el usuario y la computadora. En esta interacción, el usuario juzga la utilidad de la interfaz; el hardware y el software se convierten en simples herramientas sobre los cuales fue construida la interfaz.

- Manipulación de archivos y directorios.
- Comunicación con otros sistemas.
- Configuración de la propia interfaz y entorno.
- Intercambio de datos entre aplicaciones.

El entorno dentro del cual se inscribe el diseño de una interfaz y la medida de su usabilidad, está dado por tres factores:

- Una persona.
- Una tarea.
- Un contexto.



FIGURA 54: Diseño de usabilidad

Fuente: (LinkedIn, 2015)

Principios para el Diseño de Interfaz de Usuario

Existen principios relevantes para el diseño e implementación de IU¹⁹, ya sea para las IU gráficas, como para la Web.

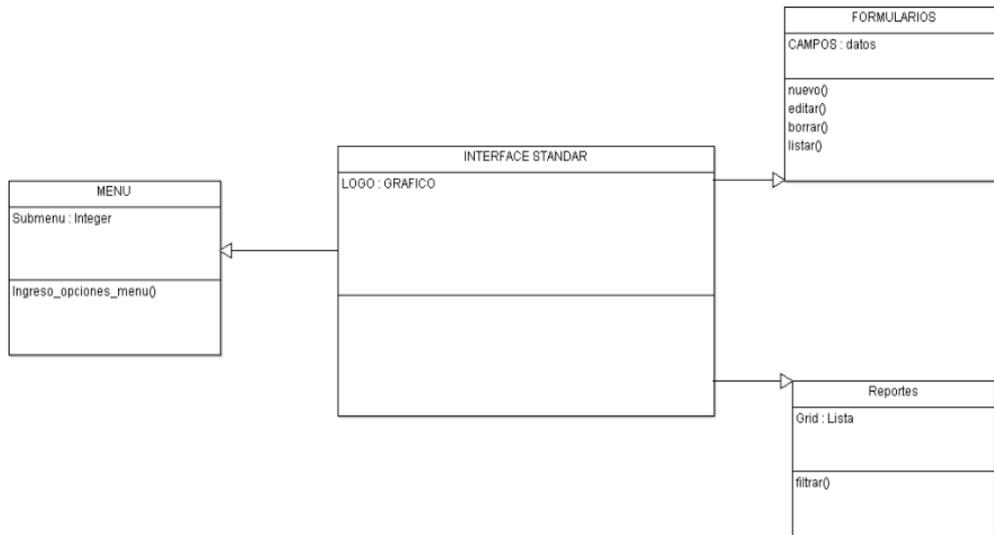
- Familiaridad del usuario: Utilizar términos y conceptos que se toman de la experiencia de las personas que más utilizan el sistema.
- Consistencia: Siempre que sea posible, la interfaz debe ser consistente en el sentido de que las operaciones comparables se activan de la misma forma.
- Mínima sorpresa: El comportamiento del sistema no debe provocar sorpresa a los usuarios.
- Recuperabilidad: La interfaz debe incluir mecanismos para permitir a los usuarios recuperarse de los errores. Esto puede ser de dos formas: Confirmación de acciones destructivas. Proveer de un recurso para deshacer.
- Guía al usuario: Cuando los errores ocurren, la interfaz debe proveer retroalimentación significativa y características de ayuda sensible al contexto.
- Diversidad de usuarios: La interfaz debe proveer características de interacción apropiada para los diferentes tipos de usuarios.

¹⁹ IU: Interfaz de Usuario.

3.3.2 DISEÑO DE FUNCIONALIDAD DEL SISTEMA

En nuestro proyecto se aplicara la siguiente estructura en cuanto a diseño.

TABLA 29: Estructura diseño del sistema.



Fuente: Autoría propia a partir de información recopilada.

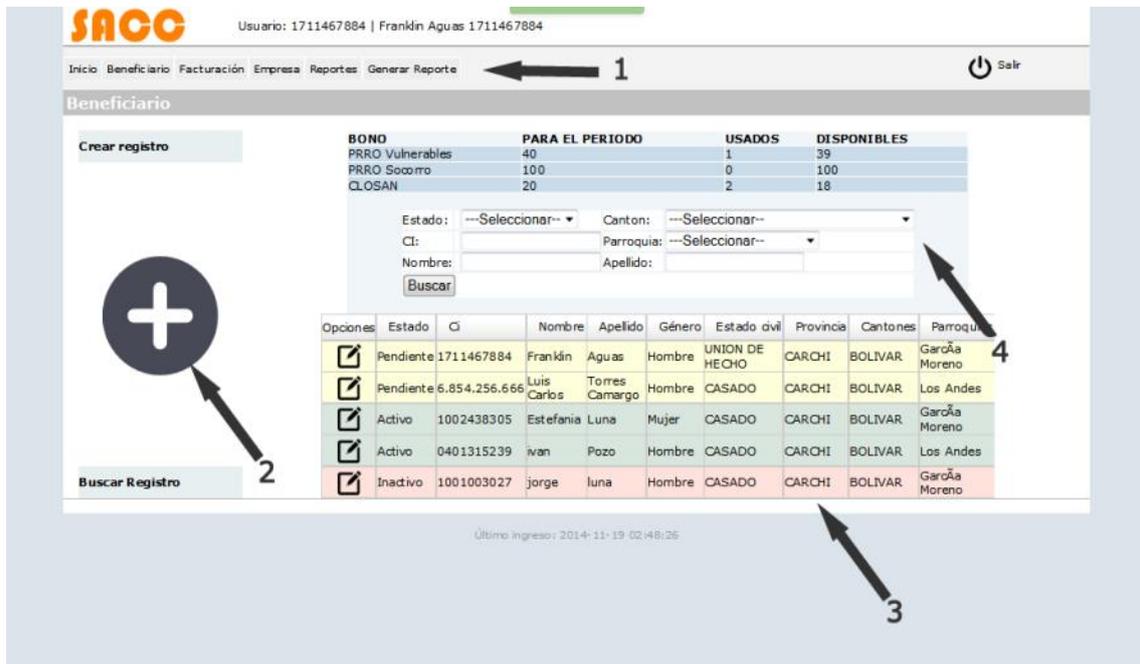


FIGURA 55: Diseño de funcionalidad sistema SACC.

Fuente: Autoría propia a partir de información recopilada.

1. Menú principal.
2. Nuevo registro.
3. Grid de datos.
4. Formulario de búsqueda.

3.3.3 ELABORACIÓN DE UN PLAN DE INTEGRACIÓN DEL SISTEMA

Plan de integración del sistema en el entorno actual de los procesos de la empresa.

TABLA 30: Cuadro de plan de integración del sistema.

TIEMPO	ACTIVIDAD	DÍAS				
		I	II	III	IV	V
1	Pruebas preliminares en el área de desarrollo.	■				
2	Pruebas con usuarios.		■			
3	Corrección de las solicitudes y errores encontrados con usuarios.			■	■	
4	Piloto en otra estación de trabajo de la empresa.					■

Fuente: Autoría propia a partir de información recopilada.

3.3.4 DOCUMENTACIÓN DE CONSTRUCCIÓN Y CONTROL DE FUNCIONABILIDAD

TABLA 31: Documentación de Sprint de construcción y control de funcionalidad.

Desarrollo de Sistema[Nombre de la empresa o cliente]:			Gobierno Autónomo Descentralizado de la provincia del Carchi	Versión [n.n]
Planificación:			Fase de Construcción y Control de Funcionabilidad	Fecha[dd/mm/aaaa] : 01 /11/2014
Descripción de la metodología de trabajo:			Proceso Scrum para diseño del desarrollo de la aplicación	
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
3.1	Diseño de interfaz	Estandarización de colores. Posicionamiento de los objetos en la pantalla.	Sprint 3.2.1 Estandarización de colores. Sprint 3.2.2 Posicionamiento de los objetos en la pantalla.	Anexo: Diseño de interfaz
3.2	Diseño de funcionalidad	Funciones estándar del sistema. Funciones extras del sistema.	Sprint 3.3.1 Funciones estándar del sistema. Sprint 3.3.2 Funciones extras del sistema.	Anexo: Diseño de funcionalidad
3.3	Pruebas	Pruebas preliminares en el área de desarrollo. Pruebas con usuarios. Corrección de las solicitudes y errores encontrados con usuarios. Piloto en otra estación de trabajo de la empresa.	Sprint 3.1.1 Pruebas preliminares en el área de desarrollo. Sprint 3.1.2 Pruebas con usuarios. Sprint 3.1.3 Corrección de las solicitudes y errores encontrados con usuarios. Sprint 3.1.4 Piloto en otra estación de trabajo de la empresa.	Anexo: Pruebas
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Fuente: Autoría propia a partir de información recopilada.

3.4 FASE DE DESPLIEGUE - IMPLEMENTACIÓN

Luego de haber realizado el análisis, diseño y construcción del “SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA”, se procederá a la instalación y entrega de la documentación del proyecto a la institución auspiciante.

3.4.1 INSTALACIÓN DEL PRODUCTO

Para la instalación del producto realizaremos los siguientes pasos:

1. Verificamos que el servidor cumpla los requisitos mínimos para la instalación, y de la misma manera verificamos la conectividad de la red.
2. Instalación y restauración de la base de datos en el servidor.
3. Instalación del servidor WEB.
4. Alojamiento de la aplicación del sistema en el directorio establecido en el servidor.
5. Iniciar los servicios.

Para comprobar que la instalación tuvo éxito procedemos a abrir un navegador en el servidor y colocamos la siguiente dirección <http://localhost/sacc/>, automáticamente veremos la pantalla de ingreso al sistema.

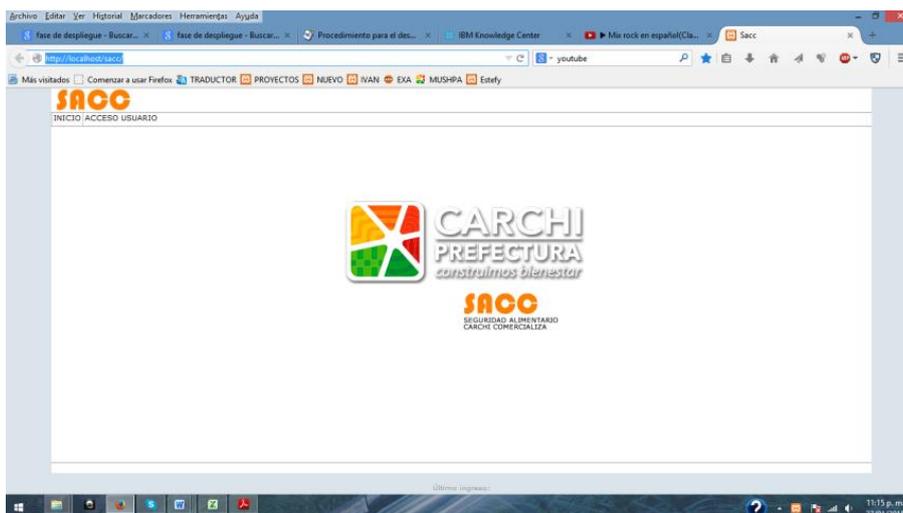


FIGURA 56: Pantalla de ingreso al sistema SACC.

Fuente: Autoría propia a partir de información recopilada.

3.4.2 CAPACITACIÓN FINAL DE USUARIOS Y PERSONAL DE SOPORTE

Después de realizar la instalación del sistema en el GAD provincial del Carchi se realizó la capacitación al personal que utilizara el sistema, esto de acuerdo al cronograma de capacitación que se indica a continuación.

TABLA 32: Cronograma de capacitación.

ID	ACTIVIDAD PREVIA A LA CAPACITACIÓN	DÍAS				
		I	II	III	IV	V
1	Elaboración de cronograma de capacitación, temario, tiempo de duración, logística, manual de usuario y manual técnico.					
2	Comunicado a o cliente					
3	Convoca a sus técnicos involucrados					
4	Comunicación y aceptación de propuesta de capacitación					
ID	ACTIVIDAD - CAPACITACIÓN	DÍAS				
		I	II	III	IV	V
5	Capacitación					

Fuente: Autoría propia a partir de información recopilada.

3.4.3 APROBACIÓN DEL PROYECTO

Como etapa final del proceso está la aprobación y entrega de la documentación del proyecto en donde la empresa auspiciante y el desarrollador se encuentran de acuerdo en los términos inicialmente establecidos.

TABLA 33: Cronograma de aprobación del proyecto.

ID	ACTIVIDAD APROBACIÓN DEL PROYECTO	DÍAS				
		I	II	III	IV	V
1	Entrega de manuales					
2	Entrega de la aplicación – código fuente					
3	Acta entrega recepción					

Fuente: Autoría propia a partir de información recopilada.

3.4.4 EVALUACIÓN DEL PROYECTO

Cuando desarrollamos un proyecto se debe considerar los aspectos para evaluar el producto creado es así que debemos contemplar aspectos como el análisis técnico costo beneficio.

TABLA 34: Análisis costo beneficio.

PARÁMETROS	DESCRIPCIÓN	VALOR REAL	VALOR CON EL PROYECTO
PHP	Lenguaje de desarrollo - Software Libre.	0\$	0\$
PostgreSQL	Base de Datos - Software Libre.	0\$	0\$
APACHE	Servidor WEB - Software Libre.	0\$	0\$
Infraestructura Hardware	1 Servidor de aplicaciones Servicio de internet	2400\$	2400\$
Implantación	1 instalación y configuración del sistema.	1500\$	0\$
Entrenamiento	Capacitación al personal de la institución.	200\$	0\$
TOTAL		4100\$	2400\$

Fuente: Autoría propia a partir de información recopilada.

Considerando el análisis de costo y beneficio podemos indicar que la implementación de este sistema beneficia a los procesos con una mínima cantidad de gasto económico.

3.4.5 DOCUMENTACIÓN DE SEGUIMIENTO DE TAREAS DE IMPLEMENTACIÓN

TABLA 35: Documentación de Sprint de tareas de implementación.

Desarrollo de Sistema[Nombre de la empresa o cliente]:		Gobierno Autónomo Descentralizado de la provincia del Carchi		Versión [n.n]
Planificación:		Fase de Despliegue - Implementación		Fecha[dd/mm/aaaa 01/01/2015
Descripción de la metodología de trabajo:		Proceso Scrum para diseño del desarrollo de la aplicación		
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
4.1	Infraestructura hardware	Servidor características mínimas Equipos a utilizarse características mínimas Conectividad en red (intranet - internet)	Sprint 4.1.1 Servidor características mínimas Sprint 4.1.2 Equipos a utilizarse características mínimas. Sprint 4.1.3 Conectividad en red (intranet - internet)	Anexos
4.2	Plataforma del sistema	Sistema Operativo Plataforma de ejecución o funcionamiento Implementación del sistema en el servidor WEB Pruebas de conectividad	Sprint 4.2.1 Sistema Operativo Sprint 4.2.2 Plataforma de ejecución o funcionamiento Sprint 4.2.3 Implementación del sistema en el servidor WEB Sprint 4.2.4 Pruebas de conectividad	Anexos
4.3	Capacitación	Elaboración de cronograma de capacitación temario tiempo de duración logística, manual de usuario y manual técnico. Comunicado a o cliente. Convoca a sus técnicos involucrados. Comunicación y aceptación de propuesta de capacitación. Capacitación.	Sprint 4.3.1 Elaboración de cronograma de capacitación temario tiempo de duración logística, manual de usuario y manual técnico. Sprint 4.3.2 Comunicado a o cliente. Sprint 4.3.3 Convoca a sus técnicos involucrados. Sprint 4.3.4 Comunicación y aceptación de propuesta de capacitación. Sprint 4.3.5 Capacitación.	Anexos
4.4	Aprobación del proyecto	Entrega de manuales. Entrega de la aplicación – código fuente. Acta entrega recepción.	Sprint 4.2.2 Entrega de manuales. Sprint 4.2.2 Entrega de la aplicación – código fuente. Sprint 4.2.2 Acta entrega recepción.	Anexos
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Fuente: Autoría propia a partir de información recopilada.

CAPÍTULO IV

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La incorporación del sistema informático permite ayudar a la gestión de cada uno de los procesos de los proyectos de ayuda humanitaria así como también en el proyecto de impulso a la comercialización.
- Se ha visto que con la utilización de la aplicación generada se ha reducido los tiempos de respuesta en la emisión de datos por cada proyecto.
- El adecuado uso de los servicios de internet permiten utilizar la información centralizada en cualquier lugar que se encuentre el usuario registrado con acceso a internet y utilice los datos existentes bajo su responsabilidad.
- El tiempo de despacho de los productos representan un problema en la atención al cliente en el proyecto de comercialización.
- En la fase de análisis se pudo detectar que la distribución de los productos era de una manera general manejando ítems sin clasificación según su clase o su tipo.
- El sistema emite información confiable de los movimientos diarios y eficiencia en la atención y distribución de los artículos.

4.2 RECOMENDACIONES

- Es importante la utilización de la aplicación desarrollada por parte de los usuarios involucrados en cada uno de los módulos existentes.
- Mantener actualizada la información desde el registro permite continuar con los procesos, emplear para ello la aplicación desarrollada.
- Los beneficios de una aplicación web es el poder ingresar al servidor desde cualquier lugar del mundo pero se debe recordar los principios de

ética profesional al manipular la información existente en una aplicación de este tipo.

- Es importante considerar la incorporación de un lector de barras para el manejo de los productos sea para el ingreso o despacho de los mismos.
- Se debe establecer una tabla de productos clasificados por grupos según su tipo, para un mejor control de bodega.
- La utilización del sistema de una manera responsable y respetando las condiciones de su funcionamiento permitirá al usuario obtener información confiable y real al momento de solicitar reportes.

4.3 GLOSARIO DE TÉRMINOS

CLOSAN

Capacidades Locales en Alimentación y Nutrición.

CSS

Cascading Style Sheets - Hojas de Estilo.

DHTML

Dinamic HyperText Markup Language (Lenguaje dinámico de hipertexto de marcado).

GAD

Gobierno Autónomo Descentralizado.

HTML

HyperText Markup Language («lenguaje de marcas de hipertexto»).

HTTP

Protocolo de transferencia de hipertexto.

IU

Interfaz de Usuario.

JSP

Es un acrónimo de Java Server Pages, (Páginas de Servidor Java).

MVC

Modelo - Vista - Controlador.

PGDG

PostgreSQL Global Development Group.

PHP

Acrónimo de Hipertext Pre-procesor (Hipertexto pre-procesado).

Plug-in

Programa que puede anexarse a otro para aumentar sus funcionalidades.

PMA

Programa Mundial de Alimentos.

PRRO Socorro

Proyecto de Recuperación en Operaciones Prolongadas de Socorro.

PRRO Vulnerables

Proyecto de Recuperación en Operaciones Vulnerables.

RUP

Proceso Unificado de Rational.

SACC

Seguridad Alimentaria Carchi Comercializa.

WEB

Es un vocablo inglés que significa “red”, “telaraña” o “malla”.

XML

eXtensible Markup Language (lenguaje de marcas extensible).

4.4 BIBLIOGRAFÍA

- Agiles, P. (26 de Enero de 2015). *http://www.proyectosagiles.org*. Obtenido de <http://www.proyectosagiles.org/beneficios-de-scrum>
- Averbuj, G. (s.f.). *http://www.gamester.com.ar*. Obtenido de http://www.gamester.com.ar/guilleaverbuj/files/backlog_y_scrum.pdf
- Blogger. (23 de Noviembre de 2012). *http://myri2a5m.blogspot.com*. Obtenido de <http://myri2a5m.blogspot.com/2012/11/la-sigla-php-identifica-un-lenguaje-de.html>
- Blogsfarm. (23 de Enero de 2015). *http://programaciondesarrollo.es*. Obtenido de <http://programaciondesarrollo.es/ventajas-del-servidor-apache/>
- Carchi Comercializa, G. (2010). Carchi Comercializa GAD Carchi. Tulcan, Carchi, Ecuador.
- D.E.L. (2008). *Reglamentos para la creación de Asociaciones Productoras*. Tulcan.
- Garcia, O. (6 de Octubre de 2012). *http://www.elclubdelprogramador.com*. Obtenido de <http://www.elclubdelprogramador.com/2012/10/06/jquery-la-biblioteca-javascript-para-un-desarrollo-mas-dinamico-y-mas-rapido/>
- Ibrugor. (27 de Enero de 2015). *http://www.ibrugor.com*. Obtenido de <http://www.ibrugor.com/blog/que-es-php-para-que-sirve/>
- Lazaro, J. M. (2001). *http://www.desarrolloweb.com*. Obtenido de <http://www.desarrolloweb.com/articulos/26.php>
- LibrosWeb.es. (07 de Enero de 2015). *http://librosweb.es/libro/xhtml/*. Obtenido de <http://librosweb.es/libro/xhtml/>
- LinkedIn. (27 de Enero de 2015). *http://es.slideshare.net*. Obtenido de <http://es.slideshare.net/klucho19/ventajas-y-desventajas-de-apache-y-iis>

Marcos, M. C. (26 de Septiembre de 2014). <http://mcmarcos.blogspot.com>.
Obtenido de <http://mcmarcos.blogspot.com/2014/09/conceptos-hci-usabilidad-arquitectura.html>

Martinez, R. (2010). <http://www.postgresql.org.es>. Obtenido de
http://www.postgresql.org.es/sobre_postgresql

Mestras, J. P. (Septiembre de 2008).
<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>. Obtenido de
<https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>

Minjarez, A. (26 de Julio de 2011). <https://plus.google.com>. Obtenido de
<https://plus.google.com/118416497256331964442/posts/iVnKbSkL49A>

Murphey, R. (24 de Noviembre de 2014). <http://librojquery.com/>. Obtenido de
<http://librojquery.com/#introducción>

Ortega, R. J. (04 de Noviembre de 2010). <http://osl.ugr.es>. Obtenido de
<http://osl.ugr.es/talleres/SCRUM/Presentacion%20SCRUM.html#slide6>

Ríos, A. (04 de Enero de 2014). <http://www.antoniorios.net>. Obtenido de
<http://www.antoniorios.net/blog/wp-content/uploads/2012/11/mvc.jpg>

S.L, a. (Marzo de 2011). <http://www.aulaclie.es>. Obtenido de
http://www.aulaclie.es/flash-cs5/t_1_2.htm

S.L., B. (23 de Enero de 2015). <http://www.empresayeconomia.es>. Obtenido de
<http://www.empresayeconomia.es/aplicaciones-para-empresas/ventajas-y-desventajas-de-postgresql.html>

Seguridad Alimentaria PMA, G. C. (7 de Octubre de 2011). Convenio de
Cooperacion Institucional entre GAD Carchi - PMA. Quito, Pichincha,
Ecuador.

WordPress. (2008-2014). <http://definicion.de>. Obtenido de
<http://definicion.de/web/>

Zamora, A. (09 de Julio de 2013). <http://es.slideshare.net>. Obtenido de
<http://es.slideshare.net/adrianazamora/diseo-de-interfaz-importancia-y-proceso-24058181>

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS

4.5 ANEXOS

ANEXO 1: SPRINT'S



SPRINT:	1 (GENERAL)			INICIO	DURACIÓN
				Septiembre	6 Meses
SISTEMA:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA				
BLACKLOG ID	TAREA	TIPO	ESTADO	RESPONSABLE	
Módulo de Registro y Análisis	• Creación de registros.	Análisis y Diseño	Fase Inicial	Desarrollador	
	• Modificación de registros.				
	• Eliminación de registros.				
	• Análisis de prioridad.				
	• Análisis de capacidad.				
	• Análisis de beneficio.				
	• Reporte de especificación y aprobación.				
Módulo de Control	• Creación de registros por beneficio.	Análisis y Diseño	Fase Inicial	Desarrollador	
	• Análisis de direccionamiento del beneficio.				
	• Depuración de datos.				
	• Control de beneficios no dúplex.				
	• Reporte general de usuarios más beneficio.				
	• Cálculo de montos por beneficiado.				
Módulo de Costos	• Creación de proveedores – asociaciones.	Análisis y Diseño	Fase Inicial	Desarrollador	
	• Creación planillas de emisión.				
	• Análisis de costos.				
	• Ingreso y egreso de productos.				
	• Cálculo de costos subtotal, total.				
	• Registro entrega y recibí conforme.				
Roles involucrados:	Desarrollador	Oscar Iván Pozo Villarreal			
	Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi.			

Para constancia de todo lo anotado anteriormente firmamos a continuación en unidad de acto.


Oscar Iván Pozo Villarreal
Desarrollador


Ing. Alex Ramos - Coordinador UTIC
Gobierno Autónomo Descentralizado de la provincia del Carchi.



METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



Desarrollo de Sistema[Nombre de la empresa o cliente]:		Gobierno Autónomo Descentralizado de la provincia del Carchi		Versión [n.n]
Planificación:		Fase de Análisis		Fecha[01/09/2014]:
Descripción de la metodología de trabajo:		Proceso Scrum para inicio del desarrollo de la aplicación		
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
1.1	Módulo de Registro y Análisis	<ul style="list-style-type: none"> • Creación de registros. • Modificación de registros. • Eliminación de registros. • Análisis de prioridad. • Análisis de capacidad. • Análisis de beneficio. • Reporte de especificación y aprobación. 	Sprint 1.1.1 Creación de registros. Sprint 1.1.2 Modificación de registros. Sprint 1.1.3 Eliminación de registros. Sprint 1.1.4 Análisis de prioridad. Sprint 1.1.5 Análisis de capacidad. Sprint 1.1.6 Análisis de beneficio. Sprint 1.1.7 Reporte de especificación y aprobación.	Revisión y aceptación: Módulo de Registro y Análisis
1.2	Módulo de Control	<ul style="list-style-type: none"> • Creación de registros por beneficio. • Análisis de direccionamiento del beneficio. • Depuración de datos. • Control de beneficios no dúplex. • Reporte general de usuarios más beneficio. • Cálculo de montos por beneficiado. 	Sprint 1.1 Creación de registros por beneficio. Sprint 1.2 Análisis de direccionamiento del beneficio. Sprint 1.3 Depuración de datos. Sprint 1.4 Control de beneficios no dúplex. Sprint 1.5 Reporte general de usuarios más beneficio. Sprint 1.6 Cálculo de montos por beneficiado.	Revisión y aceptación: Módulo de Control
1.3	Módulo de Costos	<ul style="list-style-type: none"> • Creación de proveedores – asociaciones. • Creación planillas de emisión. • Análisis de costos. • Ingreso y egreso de productos. • Cálculo de costos subtotal, total. • Registro entrega y recibi conforme. 	Sprint 1.1 Creación de proveedores – asociaciones. Sprint 1.2 Creación planillas de emisión. Sprint 1.3 Análisis de costos. Sprint 1.4 Ingreso y egreso de productos. Sprint 1.5 Cálculo de costos subtotal, total. Sprint 1.6 Registro entrega y recibi conforme.	Revisión y aceptación: Módulo de Costos
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Ciente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Para constancia de todo lo anotado anteriormente firmamos a continuación en unidad de acto.

Oscar Iván Pozo Villarreal
Desarrollador

Ing. Alex Ramos - Coordinador UTIC
Gobierno Autónomo Descentralizado de la provincia del Carchi.



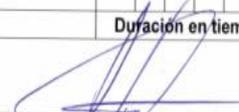
METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	1.1			Fecha Inicio : 01/09/2014	Duración: 4 semanas							
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA											
FASE:	FASE DE ANÁLISIS - Módulo de Registro y Análisis											
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD							
					0	1	2	3	4	5		
1.1.1	Creación de registros.	Desarrollador	Control de validaciones. Control de históricos en caso de eliminación de registros. Análisis y sugerencia en caso de asignación de bonos. Duplicidad en el ingreso de cedulas de beneficiarios. Control de cantidad de bonos a ser entregados en cada periodo. Reportes según las necesidades del usuario.	Según las necesidades entregadas en la reunión con el cliente se pudo constatar los inconvenientes y adicionalmente se rectificó las observaciones establecidas.			3					
1.1.2	Modificación de registros.	Desarrollador					1					
1.1.3	Eliminación de registros.	Desarrollador							1			
1.1.4	Análisis de prioridad.	Desarrollador								2		
1.1.5	Análisis de capacidad.	Desarrollador								2		
1.1.6	Análisis de beneficio.	Desarrollador								2		
1.1.7	Reporte de especificación y aprobación.	Desarrollador									2	
Duración en tiempo					Duración en tiempo (días)							
 Oscar Iván Pozo Villarreal Desarrollador					 Ing. Alex Ramallo, Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi							
ROLES INVOLUCRADOS												

ROLES:
PRODUCT OWNER (CLIENTE)

SCRUM MASTER (DESARROLLADOR)
TEAM (PERSONAL DE APOYO)

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	1.2			Fecha Inicio :	01/09/2014	Duración: 4 semanas				
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE ANÁLISIS - Módulo de Control									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
1.2.1	Creación de registros por beneficio.	Desarrollador	Cambios existentes en los beneficios el sistema debe permitir incrementar el costo del bono o su tiempo de entrega. Reportes según las necesidades del usuario.	Según las necesidades entregadas en la reunión con el cliente se pudo constatar los inconvenientes y adicionalmente se rectificó las observaciones establecidas.	2					
1.2.2	Análisis de direccionamiento del beneficio.	Desarrollador - Cliente				2				
1.2.3	Depuración de datos.	Desarrollador					3			
1.2.4	Control de beneficios no dúplex.	Desarrollador						3		
1.2.5	Reporte general de usuarios más beneficio.	Desarrollador					2			
1.2.6	Cálculo de montos por beneficiado.	Desarrollador					2			
Duración en tiempo					Duración en tiempo (días)					
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi								
ROLES INVOLUCRADOS										

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	1.3	Fecha Inicio : 01/09/2014	Duración: 4 semanas								
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA										
FASE:	FASE DE ANÁLISIS - Módulo de Costos										
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD						
					0	1	2	3	4	5	
1.3.1	Creación de proveedores – asociaciones.	Desarrollador - Cliente	En esta fase se debe trabajar conjuntamente con el desarrollador así como también con el cliente para conocer la manera de emitir resultados así como también cálculos de costos. De manera que sea interpretado por el desarrollador para ser procesado en el sistema a crearse.	Según las necesidades entregadas en la reunión con el cliente se pudo constatar los inconvenientes y adicionalmente se rectificó las observaciones establecidas.			2				
1.3.2	Creación planillas de emisión.	Desarrollador - Cliente						4			
1.3.3	Análisis de costos	Desarrollador - Cliente							4		
1.3.4	Ingreso y egreso de productos.	Desarrollador - Cliente								4	
1.3.5	Cálculo de costos subtotal, total.	Desarrollador - Cliente									4
1.3.6	Registro entrega y recibí conforme	Desarrollador - Cliente						2			
Duración en tiempo					Duración en tiempo (días)						
Oscar Iván Pozo Villarreal Desarrollador					Ing. Alex Ramos, Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi						
ROLES INVOLUCRADOS											

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



Desarrollo de Sistema[Nombre de la empresa o cliente]:		Gobierno Autónomo Descentralizado de la provincia del Carchi		Versión [n.n]
Planificación:		Fase de Diseño		Fecha[dd/mm/aaaa] 01/10/2014
Descripción de la metodología de trabajo:		Proceso Scrum para diseño del desarrollo de la aplicación		
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
2.1	Diseño de arquitectura preliminar.	Lenguaje de programación. Base de datos. Lenguaje de interface. Arquitectura de seguridad.	Sprint 2.1.1 Lenguaje de programación. Sprint 2.1.2 Base de datos. Sprint 2.1.3 Lenguaje de interface. Sprint 2.1.4 Arquitectura de seguridad.	Revisión y aceptación: Diseño de arquitectura preliminar
2.2	Requisitos de interfaz preliminar	Interfaz de usuario - interfaz del administrador	Sprint 2.2.1 Interfaz de usuario - interfaz del administrador.	Revisión y aceptación: Requisitos de interfaz preliminar
2.3	Pruebas diseño preliminar	Pruebas con usuarios por módulos en diseño. Categorización de los resultados obtenidos.	Sprint 2.3.1 Pruebas con usuarios por módulos en diseño. Sprint 2.3.2 Categorización de los resultados obtenidos. Sprint 2.3.3 Rectificación de las observaciones. Sprint 2.3.4 Ajustes de las observaciones dentro del alcance	Revisión y aceptación: Pruebas diseño preliminar
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Ciente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Para constancia de todo lo anotado anteriormente firmamos a continuación en unidad de acto

Oscar Iván Pozo Villarreal
Desarrollador

Ing. Alex Ramos - Coordinador UTIC
Gobierno Autónomo Descentralizado de la provincia del Carchi.



METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	2.1		Fecha Inicio : 01/10/2014	Duración : 4 semanas						
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE DISEÑO - Diseño de arquitectura preliminar.									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
2.1.1	Lenguaje de programación.	Desarrollador	Se solicita el desarrollo en	Se procedió al desarrollo de la aplicación con las condiciones indicadas por la institución auspiciante.	2					
2.1.2	Base de datos.	Desarrollador	software libre. BDD PostgreSQL			2				
2.1.3	Lenguaje de interface.	Desarrollador	Lenguaje de programación PHP					2		
2.1.4	Arquitectura de seguridad.	Desarrollador	Requisitos para desarrollar la aplicación.						2	
Duración en tiempo					Duración en tiempo (días)					
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi								
ROLES INVOLUCRADOS										

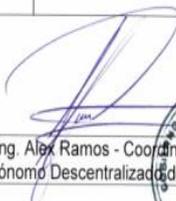
METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	2.2		Fecha Inicio: 01/10/2014	Duración: : 4 semanas						
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE DISEÑO - Requisitos de interfaz preliminar									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
2.2.1	Interfaz de usuario - interfaz del administrador.	Desarrollador	Diseño amigable y de fácil comprensión para los técnicos que se encargaran del ingreso de información y manipulación de cada uno de los módulos existentes en el sistema.	La interfaz para la utilización del proyecto en desarrollo será de fácil utilización adicionalmente se dictara capacitaciones a los usuarios con la finalidad de que su relación con el sistema sea lo más amigable posible.				4		
Duración en tiempo					Duración en tiempo (días)					
 Oscar Iván Pozo Villarreal Desarrollador			 Ing. Alex Ramos - Coordinador TIC Gobierno Autónomo Descentralizado de la Provincia del Carchi							
ROLES INVOLUCRADOS										



METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	2.3		Fecha Inicio: 01/10/2014	Duración: 4 semanas						
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE DISEÑO - Pruebas diseño preliminar									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
2.3.1	Pruebas con usuarios por módulos en diseño.	Desarrollador	Para la socialización de los diferentes módulos se solicita sea en un ambiente comprensible o de fácil entendimiento para los usuarios no emplear términos técnicos.	Se considera las observaciones del cliente y se organiza pruebas para la aprobación, utilizando para esto términos comunes.			4			
2.3.2	Categorización de los resultados obtenidos.	Desarrollador				1				
2.3.3	Rectificación de las observaciones.	Desarrollador						4		
2.3.4	Ajustes de las observaciones dentro del alcance del desarrollo.	Desarrollador						4		
Duración en tiempo					Duración en tiempo (días)					
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UIC Gobierno Autónomo Descentralizado de la provincia del Carchi								
ROLES INVOLUCRADOS										

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	3.1	Fecha Inicio : 01/11/2014	Duración: 8 Semanas							
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE CONSTRUCCIÓN - Diseño de interfaz									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
3.1.1	Estandarización de colores.	Desarrollador	Se indica el color con el cual se deberá mantener como plantilla para el diseño de la aplicación. Adicionalmente se indica el diseño con el cual se ha trabajado en programas anteriores.	El desarrollador asume las observaciones y respeta el diseño emitido por el cliente – empresa auspiciante.		2				
3.1.2	Posicionamiento de los objetos en la pantalla.	Desarrollador					3			
Duración en tiempo					Duración en tiempo (días)					
Oscar Iván Pozo Villarreal Desarrollador			Ing. Alex Ramos - Coordinador TIC Gobierno Autónomo Descentralizado de la provincia del Carchi							
ROLES INVOLUCRADOS										

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	3.2	Fecha Inicio : 01/11/2014	Duración: 8 Semanas								
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA										
FASE:	FASE DE CONSTRUCCIÓN - Diseño de funcionalidad										
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD						
					0	1	2	3	4	5	
3.2.1	Funciones estándar del sistema.	Desarrollador	Esta actividad se asigna directamente al desarrollador por el conocimiento de los temas a fines o manejo técnico de la información.	Actividades propias del desarrollador.						4	
3.2.2	Funciones extras del sistema	Desarrollador									4
Duración en tiempo					Duración en tiempo (días)						
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UTR Gobierno Autónomo Descentralizado de la provincia del Carchi									
ROLES INVOLUCRADOS											

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	3.3	Fecha Inicio : 01/11/2014	Duración: 8 Semanas							
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE CONSTRUCCIÓN - Pruebas									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
3.3.1	Pruebas preliminares en el área de desarrollo.	Desarrollador	Para la socialización de los diferentes módulos se solicita sea en un ambiente comprensible o de fácil entendimiento para los usuarios	Se considera las observaciones del cliente y se organiza pruebas para la aprobación, utilizando para esto términos comunes.				2		
3.3.2	Pruebas con usuarios.	Desarrollador						2		
3.3.3	Corrección de las solicitudes y errores encontrados con usuarios.	Desarrollador							3	
3.3.4	Piloto en otra estación de trabajo de la empresa.	Desarrollador						1		
Duración en tiempo					Duración en tiempo (días)					
Oscar Iván Pozo Villarreal Desarrollador					Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi					
ROLES INVOLUCRADOS										

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



Desarrollo de Sistema [Nombre de la empresa o cliente]:		Gobierno Autónomo Descentralizado de la provincia del Carchi		Versión [n.n]
Planificación		Fase de Despliegue - Implementación		Fecha [dd/mm/aaaa] 01/01/2015
Descripción de la metodología de trabajo:		Proceso Scrum para diseño del desarrollo de la aplicación		
SPRINT	BLACKLOG ID	TAREAS	DETALLES	HISTORIAL DE REVISIONES
4.1	Infraestructura hardware	Servidor características mínimas Equipos a utilizarse características mínimas	Sprint 4.1.1 Servidor características mínimas Sprint 4.1.2 Equipos a utilizarse características mínimas Sprint 4.1.3 Conectividad en red (intranet - internet)	Revisión y aceptación
4.2	Plataforma del sistema	Sistema Operativo Plataforma de ejecución o funcionamiento Implementación del sistema en el servidor WEB	Sprint 4.2.1 Sistema Operativo Sprint 4.2.2 Plataforma de ejecución o funcionamiento Sprint 4.2.3 Implementación del sistema en el servidor WEB Sprint 4.2.4 Pruebas de conectividad	Revisión y aceptación
4.3	Capacitación	Elaboración de cronograma de capacitación temario tiempo de duración logística, manual de Comunicado a o cliente. Convoca a sus técnicos Comunicación y aceptación de propuesta de capacitación. Capacitación.	Sprint 4.3.1 Elaboración de cronograma de capacitación temario tiempo de duración logística, manual de usuario y manual técnico. Sprint 4.3.2 Comunicado a o cliente. Sprint 4.3.3 Convoca a sus técnicos involucrados Sprint 4.3.4 Comunicación y aceptación de propuesta de capacitación. Sprint 4.3.5 Capacitación	Revisión y aceptación
4.4	Aprobación del proyecto	Entrega de manuales. Entrega de la aplicación – código Acta entrega recepción	Sprint 4.2.2 Entrega de manuales. Sprint 4.2.2 Entrega de la aplicación – código fuente. Sprint 4.2.2 Acta entrega recepción	Revisión y aceptación
Proyecto [Nombre del sistema o proyecto]:		SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA		
Roles involucrados:		Desarrollador	Oscar Iván Pozo Villarreal	
		Cliente - Propietario	Gobierno Autónomo Descentralizado de la provincia del Carchi Representante de la empresa: Ing. Alex Ramos	

Para constancia de todo lo anotado anteriormente firmamos a continuación en unidad de acto.

Oscar Iván Pozo Villarreal
Desarrollador

Ing. Alex Ramos - Coordinador UTIC
Gobierno Autónomo Descentralizado de la provincia del Carchi.



METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	4.1	Fecha Inicio : 01/01/2015	Duración : 4 semanas							
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA									
FASE:	FASE DE IMPLEMENTACIÓN - Infraestructura hardware									
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD					
					0	1	2	3	4	5
4.1.1	Servidor características mínimas.	Cliente - Desarrollador	Información ya solicitada en las reuniones de análisis - fase inicial de desarrollo.	Instalación, configuración y comunicación con el servidor de la aplicación.						
4.1.2	Equipos a utilizarse características mínimas.	Cliente - Desarrollador								
4.1.3	Conectividad en red (intranet - internet).	Cliente - Desarrollador								
Duración en tiempo					Duración en tiempo (días)					
Oscar Iván Pozo Villarreal Desarrollador		Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi								
ROLES INVOLUCRADOS										

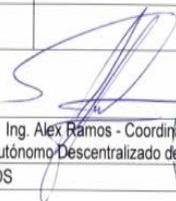
METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	4.2	Fecha Inicio : 01/01/2015	Duración : 4 semanas								
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA										
FASE:	FASE DE IMPLEMENTACIÓN - Plataforma del sistema										
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD						
					0	1	2	3	4	5	
4.2.1	Sistema Operativo	Desarrollador	Información ya solicitada en las reuniones de análisis - fase inicial de desarrollo.	Instalación, configuración y comunicación con el servidor de la aplicación.	1						
4.2.2	Plataforma de ejecución o funcionamiento (php-xampp-postgres esto es una nota).	Desarrollador			1						
4.2.3	Implementación del sistema en el servidor WEB.	Desarrollador			1						
4.2.4	Pruebas de conectividad	Desarrollador			1						
Duración en tiempo				Duración en tiempo (días)							
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UIC Gobierno Autónomo Descentralizado de la provincia del Carchi									
ROLES INVOLUCRADOS											



METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	4.3	Fecha Inicio : 01/01/2015	Duración : 4 semanas								
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA										
FASE:	FASE DE IMPLEMENTACIÓN - Capacitación										
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD						
					0	1	2	3	4	5	
4.3.1	Elaboración de cronograma de capacitación, temario, tiempo de duración, logística, manual de usuario y manual técnico.	Desarrollador	Utilizar el cronograma de desarrollo para culminar con la entrega del proyecto.	Entregar la información necesaria para culminar con el proceso de capacitación.		1					
4.3.2	Comunicado a o cliente	Desarrollador					1				
4.3.3	Convoca a sus técnicos involucrados.	Cliente							1		
4.3.4	Comunicación y aceptación de propuesta de capacitación.	Cliente								1	
4.3.5	Capacitación.	Desarrollador									1
Duración en tiempo					Duración en tiempo (días)						
 Oscar Iván Pozo Villarreal Desarrollador		 Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la Provincia del Carchi									
ROLES INVOLUCRADOS											

METODOLOGÍA ÁGIL SCRUM

DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS

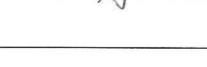


METODOLOGÍA ÁGIL SCRUM DAILY SPRINT MEETING – CICLO PARA LA CREACIÓN DE TAREAS



SPRINT:	4.4	Fecha Inicio : 01/01/2015	Duración : 4 semanas								
Nombre del Sistema:	SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROYECTOS CARCHI SEGURIDAD ALIMENTARIA Y CARCHI COMERCIALIZA										
FASE:	FASE DE IMPLEMENTACIÓN - Aprobación del proyecto										
BLACKLOG ID	TAREA	RESPONSABLE	OBSERVACIONES - CLIENTE	CAMBIOS - DESARROLLADOR	NIVEL DE COMPLEJIDAD						
					0	1	2	3	4	5	
4.4.1	Entrega de manuales.	Desarrollador					1				
4.4.2	Entrega de la aplicación – código fuente.	Desarrollador					1				
4.4.3	Acta entrega recepción.	Cliente - Desarrollador					1				
Duración en tiempo					Duración en tiempo (días)						
Oscar Iván Pozo Villarreal Desarrollador					Ing. Alex Ramos - Coordinador UTIC Gobierno Autónomo Descentralizado de la provincia del Carchi						
ROLES INVOLUCRADOS											

ANEXO 2: CAPACITACIÓN

REGISTRO DE CAPACITACIÓN		
SISTEMAS INFORMÁTICO "SACC"		
FECHA:	30 de Enero 2015	
LUGAR:	Laboratorio Informática Casa de la Juventud EAD-PC.	
NOMBRE	CARGO	FIRMA
HUBERTO KORA	PROMOTOR TÉCNICO	
MAURICIO OLAVEZ	RESPONSABLE BODEGA	
Darwin Pozo	TÉCNICO SUB-PROYECTOS	
Edison Reyes	Técnico de Escuelas	
Jefferson Alemaín R.	Técnico Apoyo Proyectos	
Manuela Jativa	Juniors I	
Liliana Cifuentes	Asistente técnico Administrativo	
Ramón Alvarado Rosero	Auxiliar Promotor Huertos	
Lenin Aceldo	Coordinador C. S. A.	
Mar Ramos B.	UTIC	

Para constancia de realizada la capacitación firmamos a continuación en unidad de acto.


Oscar Iván Pozo Villarreal
Desarrollador


Ing. Alex Ramos - Coordinador UTIC
Gobierno Autónomo Descentralizado de la
provincia del Carchi.



ANEXO 3: MANUALES

Manual módulo Proyecto Carchi Seguridad Alimentaria

Manual módulo Carchi Comercializa

Manual módulo de Administración de Usuarios y Generar Reportes