

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA – LEÓN

UNAN-LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA



DEPARTAMENTO DE COMPUTACIÓN

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMÁTICA EMPRESARIAL

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN TÉCNICA DEL CULTIVO DEL
CAMARÓN PARA LAS GRANJAS CAMARONERAS DE LA EMPRESA CAMANICA,
CHINANDEGA EN EL AÑO 2015.**

Autor:

Nery Ricardo González Castro

Tutor:

MSc. Valeria Mercedes Medina Rodriguez

Febrero 2016

Resumen

Resumen

Las granjas camaroneras de la empresa CAMANICA ZONA FRANCA, S.A. son el área productiva del cultivo del camarón. Ésta área tiene una gran importancia ya que es la productora de la materia prima que utiliza la empresa para la elaboración del producto que se comercializa y exporta.

Para el buen aprovechamiento del cultivo, éste debe ser monitoreado ya que la calidad del agua juega un papel importante en el rendimiento final del mismo. La información técnica recopilada es almacenada en documentos manuscritos y digitalizada en hojas de cálculo.

En este proyecto se desarrolla un sistema para gestionar la información técnica, debido a la importancia que tiene ésta para el cultivo del camarón por la que se utilizará como un repositorio para tener acceso a información técnica histórica.

El sistema se desarrolló con el paradigma de Programación Orientada a Objeto para la arquitectura cliente – servidor utilizando la metodología RAD con las herramientas Microsoft© Visual Studio© Express for Desktop 2013 y Microsoft© SQL Server© Standard 2012.

Dedicatoria

Dedicatoria

Tras culminar una etapa más de mi vida profesional deseo dedicar esta obra:

A nuestro Señor

Por darme la salud, la sabiduría y la fortaleza necesaria para completar esta etapa.

A mi esposa

Lic. Ana Maribel Palma Romero por su amor, tiempo y paciencia. Ayudándome día a día a tratar de ser un mejor profesional e individuo.

A mis hijos

Ricardo Javier y Andrea Fernanda por ser ambos la esperanza y horizonte en mi vida personal y profesional, quienes desapercibidamente y sin interés alguno me ayudaron a no desviarme de la meta final.

Agradecimientos

Agradecimientos

Le agradezco profundamente a nuestro Señor por permitirme la vida, darme la perseverancia suficiente para poder completar esta meta más en mi vida.

A *mi esposa* por apoyarme incondicionalmente en mis proyectos de vida y profesionales.

A *mis hijos* por prestarme parte de su tiempo para poder culminar esta meta.

A *mis madres* por haber cultivado en mí las buenas costumbres y educación que me han permitido avanzar por la vida.

A *mi amigo Lic. Carlos Enrique Gómez Mayorga* por su invaluable tiempo y opinión.

A *mi tutora Msc. Valeria Mercedes Medina Rodríguez* por su tiempo, dedicación y paciencia; piezas fundamentales, para culminar este trabajo.

A *mis catedráticos* por haber aportado lo mejor de sí en todas sus clases.

A *mis compañeros* por haberme permitido intercambiar y aumentar mis conocimientos.

Tabla de Contenido

RESUMEN	I
DEDICATORIA	II
A NUESTRO SEÑOR	II
A MI ESPOSA.....	II
A MIS HIJOS	II
AGRADECIMIENTOS	III
TABLA DE CONTENIDO	I
ÍNDICE DE TABLAS	VI
ÍNDICE DE FIGURAS	VII
CAPÍTULO 1: INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	1
ANTECEDENTES	2
JUSTIFICACIÓN.....	2
OBJETIVOS	3
<i>Objetivo General</i>	3
<i>Objetivos Específicos</i>	3
CAPÍTULO 2: MARCO TEÓRICO	4
GRANJA ACUICULTURA	4
<i>Características Generales</i>	4
<i>Diseño Estructural</i>	4
Campamento.....	5
Bodega de Almacenamiento	5
Invernadero.....	5
Reservorio	5
Bordas.....	5
Canales.....	5
Esclusas.....	5
Pilas	5
ESPECIES DE CAMARÓN	7
<i>Camarón Patiblanco (Litopenaeus vannamei)</i>	7
Países Productores	8
Hábitat	9
Técnicas de Cultivo	9
Extensivas.....	9
Semi Intensivas	9
Intensivas.....	9
Súper Intensivas.....	9
Técnicas de Cosecha	10
Manejo y Maquila.....	11
CALIDAD DEL AGUA	12

Tabla de Contenido

TIPOS DE PARÁMETROS	13
PARÁMETROS	13
<i>Parámetros Físicos</i>	13
<i>Parámetros Químicos</i>	13
<i>Parámetros Biológicos</i>	14
ANÁLISIS DE REGRESIÓN LINEAL	15
<i>Introducción</i>	15
<i>Línea de tendencia</i>	15
<i>Coefficiente de Pearson o Coeficiente de determinación R²</i>	17
ARQUITECTURA DE APLICACIONES CLIENTE – SERVIDOR	18
<i>Arquitectura de Una Capa</i>	19
<i>Arquitectura de Dos Capas</i>	20
<i>Arquitectura de Tres Capas</i>	21
<i>Arquitectura de Dos Capas vs Tres Capas</i>	22
<i>Arquitectura de N Capas</i>	23
PROGRAMACIÓN ORIENTADA A OBJETOS	23
<i>Introducción</i>	23
<i>Objeto</i>	24
Modularidad	24
Información oculta – Encapsulación	25
Reutilización del código	25
Interoperabilidad	25
<i>Mensajes</i>	25
<i>Métodos</i>	25
<i>Clases</i>	25
<i>Características</i>	26
Abstracción	26
Encapsulamiento	26
Herencia	26
Polimorfismo	27
LENGUAJE DE PROGRAMACIÓN C#	27
SEGURIDAD INTEGRADA DE WINDOWS	29
<i>Autenticación con Microsoft NTLM (Desafío/Respuesta)</i>	29
<i>Autenticación con Kerberos</i>	31
Conceptos Básicos de Autenticación	33
Llave de Autenticación	33
Mensaje del Autenticador	33
Distribución de la llave	33
Tickets de Sesión	34
Tickets del servicio de generación de tickets	35
Modelo de Kerberos	36
Centro de Distribución de Llaves	36
Base de datos de Cuentas	37
Políticas de Kerberos	38
WEB SERVICES	39
<i>Tipos de Web Services</i>	39

Tabla de Contenido

Servicios Web Completos.....	39
Servicios Web “RESTful”.....	40
SERVIDOR DE BASES DE DATOS.....	41
<i>Estructura de las bases de datos</i>	41
Sistema de gestión de archivos.....	41
Bases de datos jerárquicas.....	42
Bases de datos en red.....	44
Bases de datos relacionales.....	45
Lenguaje Estructurado de Consultas (SQL) Structured Query Language.....	45
Normalización.....	46
Transacciones.....	48
Microsoft SQL Server.....	49
CAPÍTULO 3: DISEÑO METODOLÓGICO.....	56
TIPO DE ESTUDIO.....	56
TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN.....	56
<i>Observación Directa</i>	56
<i>Revisión Documental</i>	56
<i>Entrevistas no estructuradas</i>	56
POBLACIÓN Y MUESTRA.....	56
PROCEDIMIENTO DE LA INFORMACIÓN.....	56
DESCRIPCIÓN DE LA METODOLOGÍA.....	57
<i>Objetivo y/o Propósito</i>	58
<i>Fase de Requerimientos</i>	58
<i>Fase de Diseño de Usuario</i>	58
<i>Fase de Construcción</i>	59
<i>Fase de Reemplazo</i>	59
<i>Desventajas</i>	60
ETAPAS DEL PROYECTO.....	60
<i>Flujo de proceso de la recolección de la información técnica y su frecuencia</i>	60
<i>Análisis de requerimientos</i>	61
<i>Diagrama entidad relación</i>	62
<i>Diseño arquitectónico</i>	62
<i>Desarrollo del sistema</i>	62
<i>Implementación del sistema</i>	62
MATERIALES UTILIZADOS.....	63
<i>Software</i>	63
<i>Hardware</i>	63
CAPÍTULO 4: RESULTADOS DEL PROYECTO.....	64
FUNCIONAMIENTO DEL SISTEMA.....	64
<i>Consideración General</i>	64
<i>Seguridad</i>	64
Roles.....	64
Usuarios.....	66

Tabla de Contenido

Usuarios por Roles.....	67
Opciones del sistema.....	69
Opciones del Sistema por Roles.....	71
Criptografía.....	73
Configuración.....	74
Granjas.....	74
Pilas.....	76
Ciclos.....	77
Semanas.....	79
Parámetros.....	81
Tipos de Pilas.....	82
Tipos de Parámetros.....	83
Unidades de Medidas.....	84
Proceso.....	85
Lecturas.....	85
Gráficas.....	87
CAPÍTULO 5: ASPECTOS FINALES.....	90
CONCLUSIONES.....	90
RECOMENDACIONES.....	90
<i>Recomendaciones de uso.....</i>	<i>90</i>
<i>Recomendaciones o mejoras a la aplicación.....</i>	<i>90</i>
BIBLIOGRAFÍA.....	91
ANEXOS.....	99
ANEXO 1: CRONOGRAMA DE ACTIVIDADES.....	99
ANEXO 2: REQUERIMIENTOS DEL SISTEMA.....	100
<i>Definición, Acrónimos y abreviaturas.....</i>	<i>100</i>
Sistema.....	100
Sistemas Informáticos.....	100
Métodos.....	100
Análisis de Sistemas.....	100
Análisis estructurado.....	100
Ciclo de vida de un sistema informático.....	100
Especificaciones de Requisitos de Software (ERS).....	100
Diagrama Entidad Relación (ER).....	100
Diseño de sistemas.....	100
Diseño arquitectónico.....	100
Red.....	101
LAN (Local Area Network).....	101
Clientes.....	101
Servidores.....	101
Cliente - Servidor.....	102
<i>Alcance Funcional del Sistema.....</i>	<i>102</i>
Generalidades.....	102
Alcance Funcional.....	102
Características de los usuarios.....	102

Tabla de Contenido

<i>Requerimientos</i>	102
Interfaz de usuario.....	102
Requerimientos Funcionales	103
Requerimientos no Funcionales	104
Seguridad.....	104
Disponibilidad	104
Portabilidad	105
ANEXO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	105
<i>Arquitectura del sistema</i>	105
<i>Módulos/Secciones del sistema</i>	105
Procesos.....	105
Configuración	105
Seguridad	105
<i>Diagramas de casos de uso</i>	106
Seguridad	106
Configuración	106
Proceso	107
ANEXO 4: DESARROLLO DEL SISTEMA	108
<i>Arquitectura cliente - servidor</i>	108
Proyecto de Aplicación de Windows- SITCC	109
Proyecto de Librerías de clases- SITCCDatos	111
Proyecto de Librerías de clases- SITCCUtil.....	112
Proyecto de Librerías de clases- SITCCModelos.....	113
<i>Modelo de Seguridad</i>	115
Usuario.....	116
Rol	116
Opción del Sistema	116
Usuario por Rol.....	116
Opción del Sistema por Rol	116
<i>Exportación de Datos</i>	119
<i>Control de Mapa GMap.Net (Servicio Web)</i>	121
<i>Control "Chart"</i>	122
<i>Reportes</i>	126
<i>Diagrama de clases</i>	131
<i>Diagrama entidad relación</i>	132

Índice de Tablas

Índice de Tablas

TABLA 1: PARÁMETROS FÍSICOS.....	13
TABLA 2: PARÁMETROS QUÍMICOS.....	13
TABLA 3: PARÁMETROS BIOLÓGICOS.....	14
TABLA 4: ARQUITECTURA DE DOS CAPAS VS TRES CAPAS	22
TABLA 5: VENTAJAS Y DESVENTAJAS DE LOS MODELOS JERÁRQUICO Y RED	44
TABLA 6: SOFTWARE UTILIZADO.....	63
TABLA 7: HARDWARE UTILIZADO.....	63
TABLA 8: REQUERIMIENTOS DEL SISTEMA	103

Índice de Figuras

Índice de Figuras

FIGURA 1: ESTRUCTURA DE UNA GRANJA.....	4
FIGURA 2: VISTA ÁREA DE LAS PILAS DE UNA GRANJA	6
FIGURA 3: ESPECIES QUE HABITAN AL NORTE DEL GOLFO DE SAN LORENZO, CANADÁ	7
FIGURA 4: JERARQUÍA TAXONÓMICA DE LITOPENAEUS VANNAMEI [21].....	7
FIGURA 5: CAMARÓN PATIBLANCO (LITOPENAEUS VANNAMEI) [20].....	8
FIGURA 6: PRINCIPALES PAÍSES PRODUCTORES DE CAMARÓN PATIBLANCO [20].....	8
FIGURA 7: RACEWAYS UTILIZADO PARA EL CULTIVO SÚPER INTENSIVO DEL CAMARÓN [22].....	10
FIGURA 8: COSECHA DEL CAMARÓN	11
FIGURA 9: MAQUILA DE CAMARÓN.....	12
FIGURA 10: VALOR AGREGADO AL PROCESO DE CAMARÓN	12
FIGURA 11: DIAGRAMA DE DISPERSIÓN	15
FIGURA 12: LÍNEA DE TENDENCIA.....	16
FIGURA 13: ARQUITECTURA CLIENTE - SERVIDOR	18
FIGURA 14: ARQUITECTURA UNA CAPA – APLICACIÓN MONOLÍTICA.....	19
FIGURA 15: ARQUITECTURA DE DOS CAPAS.....	20
FIGURA 16: ARQUITECTURA DE TRES CAPAS	21
FIGURA 17: ARQUITECTURA DE N CAPAS	23
FIGURA 18: REPRESENTACIÓN DE OBJETOS.....	24
FIGURA 19: REPRESENTACIÓN DE UNA CLASE	25
FIGURA 20: ABSTRACCIÓN Y ENCAPSULAMIENTO	26
FIGURA 21: HERENCIA	27
FIGURA 22: CICLO DE EJECUCIÓN C#	28
FIGURA 23: MODELO DE SEGURIDAD NT Y SSPI.....	29
FIGURA 24: ESQUEMA DE AUTENTICACIÓN NTLM [52]	30
FIGURA 25: AUTENTICACIÓN KERBEROS [52].....	32
FIGURA 26: MENSAJE SOAP	40
FIGURA 27: MENSAJE REST	41
FIGURA 28: ORGANIZACIÓN JERÁRQUICA	43
FIGURA 29: ORGANIZACIÓN DE RED	45
FIGURA 30: PROCESO DE NORMALIZACIÓN.....	48
FIGURA 31: ACID.....	49
FIGURA 32: SERVIDOR SQL.....	50
FIGURA 33: COMPONENTES / OBJETOS DE UNA BASE DE DATOS.....	51
FIGURA 34: FASES DE LA METODOLOGÍA RAD	57
FIGURA 35: METODOLOGÍA DE CASCADA VS RAD	58
FIGURA 36: FLUJO DE PROCESO DE LECTURAS DE PARÁMETROS.....	61
FIGURA 37: DISEÑO ARQUITECTÓNICO	62
FIGURA 38: ROLES DEL SISTEMA	64
FIGURA 39: IMPRESIÓN DE ROLES	65
FIGURA 40: USUARIOS DEL SISTEMA	66
FIGURA 41: IMPRESIÓN DE USUARIOS	66
FIGURA 42: USUARIOS POR ROLES	67
FIGURA 43: IMPRESIÓN DE USUARIOS POR ROLES	68

Índice de Figuras

FIGURA 44: OPCIONES DEL SISTEMA	69
FIGURA 45: IMPRESIÓN DE OPCIONES DEL SISTEMA.....	70
FIGURA 46: OPCIONES DEL SISTEMA POR ROLES	71
FIGURA 47: IMPRESIÓN DE OPCIONES DEL SISTEMA POR ROLES	72
FIGURA 48: CRIPTOGRAFÍA.....	73
FIGURA 49: CONFIGURACIÓN DE GRANJAS	74
FIGURA 50: IMPRESIÓN DE GRANJAS	75
FIGURA 51: CONFIGURACIÓN DE PILAS.....	76
FIGURA 52: IMPRESIÓN DE PILAS	76
FIGURA 53: CONFIGURACIÓN DE CICLOS DE PRODUCCIÓN.....	77
FIGURA 54: IMPRESIÓN DE CICLOS DE PRODUCCIÓN POR PILAS	78
FIGURA 55: CONFIGURACIÓN DE SEMANAS	79
FIGURA 56: IMPRESIÓN DE SEMANAS DEL AÑO.....	80
FIGURA 57: CONFIGURACIÓN DE PARÁMETROS	81
FIGURA 58: IMPRESIÓN DE PARÁMETROS.....	81
FIGURA 59: CONFIGURACIÓN DE TIPOS DE PILAS.....	82
FIGURA 60: IMPRESIÓN DE TIPOS DE PILAS	82
FIGURA 61: CONFIGURACIÓN DE TIPOS DE PARÁMETROS	83
FIGURA 62: IMPRESIÓN DE TIPOS DE PARÁMETROS	83
FIGURA 63: CONFIGURACIÓN DE UNIDADES DE MEDIDAS	84
FIGURA 64: IMPRESIÓN DE UNIDADES DE MEDIDAS.....	84
FIGURA 65: LECTURAS DE PARÁMETROS	85
FIGURA 66: GRABACIÓN DE LECTURAS DE PARÁMETROS	86
FIGURA 67: IMPRESIÓN DE LECTURAS DE PARÁMETROS	86
FIGURA 68: GRÁFICA DE LECTURAS DE UNA PILA Y LÍNEA DE TENDENCIA.....	87
FIGURA 69: GRÁFICA DE LECTURAS DE MÚLTIPLES PILAS Y LÍNEAS DE TENDENCIA	88
FIGURA 70: EXPORTACIÓN DE LECTURAS HACIA ARCHIVO DE TEXTO	88
FIGURA 71: LECTURAS IMPORTADAS DESDE ARCHIVO DE TEXTO HACIA MICROSOFT EXCEL	89
FIGURA 72: CRONOGRAMA DE ACTIVIDADES	99
FIGURA 73: DIAGRAMA DE CASO DE USO DE SEGURIDAD.....	106
FIGURA 74: DIAGRAMA DE CASO DE USO DE CONFIGURACIÓN	106
FIGURA 75: DIAGRAMA DE CASO DE USO DE PROCESO.....	107
FIGURA 76: SOLUCIÓN DE ESCRITORIO.....	108
FIGURA 77: INTERFAZ UTILIZADA EN TODOS LOS FORMULARIOS HIJOS DEL SISTEMA	109
FIGURA 78: CÓDIGO FUENTE PARCIAL DE FORMULARIO PARA LAS GRANJAS	110
FIGURA 79: CÓDIGO FUENTE PARCIAL DE LA CLASE MANEJADOR DE DATOS.....	111
FIGURA 80: CÓDIGO FUENTE PARCIAL LIBRERÍA DE CLASE UTILITARIO	112
FIGURA 81: CÓDIGO FUENTE PARCIAL DE LOS CAMPOS DE LA ENTIDAD GRANJA EN PROYECTO MODELOS	114
FIGURA 82: CÓDIGO FUENTE PARCIAL DE LOS MÉTODOS DE LA ENTIDAD GRANJA EN PROYECTO MODELOS	114
FIGURA 83: CÓDIGO FUENTE PARCIAL DEL MÉTODO CREAR DE LA ENTIDAD GRANJA EN PROYECTO MODELOS.....	115
FIGURA 84: MODELO DE SEGURIDAD	115
FIGURA 85: OPCIONES DEL SISTEMA POR ROL.....	117
FIGURA 86: CARGA INICIAL DE PERMISOS EN EL FORMULARIO PRINCIPAL	118
FIGURA 87: EVENTO "LOAD" DEL FORMULARIO GRANJA.....	118

Índice de Figuras

FIGURA 88: MÉTODO ESTABLECERPERMISOS DEL FORMULARIO GRANJA.....	119
FIGURA 89: CÓDIGO FUENTE PARCIAL FORMULARIO GRÁFICA.....	120
FIGURA 90: CÓDIGO FUENTE PARCIAL DE LA CLASE CSVEXPORT	120
FIGURA 91: CÓDIGO FUENTE PARCIAL PARA INICIALIZAR CONTROL DE MAPA.....	121
FIGURA 92: CONTROL CHART EN TIEMPO DE DISEÑO.....	122
FIGURA 93: CÓDIGO FUENTE PARCIAL PARA GRAFICAR	123
FIGURA 94: CÓDIGO FUENTE PARCIAL “FINANCIALFORMULA” DEL CONTROL CHART DE MICROSOFT	124
FIGURA 95: GESTOR DE PAQUETES NUGET PARA VISUAL STUDIO 2013.....	125
FIGURA 96: REFERENCIAS DE BUSINESS OBJECT CRYSTAL REPORTS	126
FIGURA 97: CONTROL CRYSTAL REPORTS EN TIEMPO DE DISEÑO	126
FIGURA 98: CÓDIGO FUENTE PARCIAL PARA IMPRIMIR UN REPORTE	127
FIGURA 99: CÓDIGO FUENTE PARCIAL DEL VISOR DE REPORTES	128
FIGURA 100: CÓDIGO FUENTE PARCIAL INICIO DE SESIÓN REPORTE Y SUB REPORTES	129
FIGURA 101: CÓDIGO FUENTE PARCIAL ESTABLECER PARÁMETROS DE REPORTE	130
FIGURA 102: DIAGRAMA DE CLASES	131
FIGURA 103: DIAGRAMA ENTIDAD RELACIÓN	132
FIGURA 104: DIAGRAMA ENTIDAD RELACIÓN SEGUNDA PARTE	133

Capítulo 1: Introducción

Capítulo 1: Introducción

Planteamiento del Problema

CAMANICA [1] [2] una empresa perteneciente al Grupo Pescanova [3] desde el año 2007 tiene una capacidad de 3,500 hectáreas de pilas en granjas camaroneras, un laboratorio capaz de producir 900 millones de larvas mensuales, produce 120 millones diarios de nauplios¹ y cuenta con una planta de proceso de 120 toneladas métricas diarias. [4]Actualmente en las granjas camaroneras se realiza la recolección de los parámetros técnicos sobre el cultivo de forma manual. Estos parámetros físicos, químicos, biológicos [5] son tomados de 1 a 2 veces al día por un parametrista² y la medición se realiza a cada pila sembrada en cada granja durante el ciclo productivo. Esta información es documentada en formatos de papel manualmente; posteriormente, la información se digitaliza en hojas de cálculo que luego son distribuidas a los supervisores y jefes de granjas para ser analizadas y tomar acciones inmediatas en pro del beneficio del cuidado del camarón. Esta información pierde su utilidad estadística ya que no es almacenada en ningún repositorio o sistema de información.

Es por ello que se propone el desarrollo de un Sistema de Información Técnica del cultivo del camarón en las granjas camaroneras de la empresa que permita agilizar en tiempo real la disponibilidad de la información técnica en cada granja. Así mismo, proporcionar una herramienta visual para el análisis de la información técnica y exportación de los datos recopilados.

Este sistema es un componente de un sistema de gestión de calidad que involucra todos los procesos de la producción y exportación de camarón. La empresa ha comenzado un proyecto en conjunto con Telefónica-Movistar para el desarrollo de una Interfaz USSD³ que permitirá la digitación remota y en tiempo real de la información técnica en la base de datos del sistema. Esto se contempló como una alternativa de ingreso de la información ya que no todas las granjas cuentan con una interconexión de banda ancha con la sede central. El USSD se ejecuta a través del

¹ Es el producto de la eclosión del huevo de camarón.

² Colaborador que opera los equipos de medición para la obtención de lecturas de los parámetros.

³ USSD (Unstructured Supplementary Service Data) Servicio de Datos Suplementario Sin estructura.

Capítulo 1: Introducción

canal de voz provisto por la empresa celular. No es necesaria la conexión de un plan de datos en el dispositivo móvil para que sea funcional la ejecución de la plataforma.

Antecedentes

En la empresa no existe ningún antecedente sobre la automatización de la información técnica en las granjas. Ésta información se ha almacenado tradicionalmente en hojas de cálculos que se utilizan para analizar la información. En la industria regional [6] se ha utilizado el método estadístico de regresión lineal [7] [8] [9] para el análisis de los datos relacionados al cultivo del camarón; sin embargo, estos análisis son realizados a través de software estadísticos comerciales [10].

Justificación

El monitoreo de la información técnica [5] es elemental para asegurar una buena producción al final del ciclo de crecimiento/engorde del camarón. Este aspecto de la información; define, la necesidad de diseñar un sistema de información técnica que permita tener en tiempo real e histórica los datos pertinentes a las series numéricas de los parámetros físicos y químicos relacionados con el ciclo productivo del cultivo del camarón.

Dicho sistema; en caso de ser implementado por la empresa, permitiría al supervisor y/o jefe de granja agilizar la interpretación de la información técnica actual de manera granular de los datos recopilados de su respectiva granja. De igual manera, permitiría utilizar datos históricos para extrapolar o construir modelos estadísticos basados en la información disponible. Los datos almacenados se utilizarían para aplicar el método de regresión lineal para obtener un modelo estadístico que permita establecer la relación entre los parámetros obtenidos y el crecimiento del camarón a través del tiempo. Al igual, de proporcionar gráficos estadísticos de los datos almacenados del cultivo. El sistema sería desarrollado con la tecnología actualizada utilizando una metodología RAD⁴ [11] que permitiría un desarrollo “acelerado” en comparación con metodologías tradicionales de desarrollo.

⁴ RAD (Rapid Application Development) (Desarrollo Rápido de Aplicaciones).

Capítulo 1: Introducción

Objetivos

Objetivo General

Desarrollar un sistema de información técnica del cultivo de camarón para las granjas camaroneras de CAMANICA aplicando la metodología RAD utilizando Microsoft® Visual Studio® Express for Desktop 2013 y Microsoft® SQL Server® Standard 2012 durante el año 2015.

Objetivos Específicos

- Automatizar el flujo de proceso de las lecturas de parámetros mediante la estandarización del proceso de digitación de la información técnica.
- Gestionar la información técnica del cultivo del camarón para facilitar la toma de decisiones desde el punto de vista técnico.
- Ofrecer una herramienta de análisis a través de gráficas de series de datos utilizando Microsoft® Chart para formularios Windows.
- Crear un repositorio de información técnica histórica por ciclos de producción del cultivo del camarón.

Capítulo 2: Marco Teórico

Capítulo 2: Marco Teórico

Granja Acuicultura

Características Generales

Las granjas acuícolas deben presentar condiciones de terrenos favorables para el desarrollo de la actividad acuícola. Se deben considerar varios aspectos dentro de éstos: tipo de suelo, proximidad de la fuente de agua, acceso al terreno. De igual manera, deberá contar con cierta infraestructura capaz de sobrellevar las necesidades propias del cultivo. [12] [13] [14]

Diseño Estructural

La estructura principal de la granja es la pila y con ella todo un conjunto de estructuras de soporte colateral. Estas estructuras alternas son parte fundamental para el buen funcionamiento de la pila. Dentro de estas tenemos: esclusas, reservorios, bordas, canales, sistema de bombeo, etc. [12] [13]



Figura 1: Estructura de una granja

Una granja puede contener las siguientes estructuras y/o edificaciones:

Capítulo 2: Marco Teórico

Campamento

Es el conjunto de edificios que componen las oficinas administrativas; así como, el sitio donde los trabajadores residen durante el desarrollo de sus actividades laborales.

Bodega de Almacenamiento

Son estructuras utilizadas para almacenar, proteger y controlar los diferentes insumos utilizados para el desarrollo de las actividades relacionados con el cultivo del camarón.

Invernadero

Es una estructura generalmente cerrada y climatizada; con un buen control de los factores ambientales externos, que se utiliza para albergar y climatizar a las crías o larvas que serán sembradas en las pilas. Este paso es considerado, el paso intermedio para que el organismo no se vea afectado por los cambios bruscos de los factores ambientales diferenciados entre el sitio donde se desarrolló en sus primeras etapas contra el sitio [pila] donde crecerá hasta estar listo para ser cosechado.

Reservorio

Es una estructura que se utiliza para almacenar agua que será utilizada para llenar las pilas en sus etapas iniciales de preparación del cultivo. También, es utilizado para recircular el agua almacenada en las pilas. Normalmente, los reservorios son construidos a lo largo de las pilas de una granja; también, en dependencia de las distancias y tamaños de las granjas pueden haber construidos un sistema de acueducto conectado a un sistema de bombeo de agua.

Bordas

Son estructuras que se utilizan para el desplazamiento vehicular y peatonal a través de las pilas de una granja. Estas normalmente son estructuras de tierra compactadas alrededor de las pilas que se utilizan para acceder a las mismas.

Canales

Son estructuras que permiten el flujo del agua entre el reservorio y las pilas.

Esclusas

Son compuertas que se utilizan para la cosecha del camarón.

Pilas

Son estructuras que se utilizan para criar y engordar el cultivo del camarón. Estas pueden ser de diferentes formas: rectangulares, cuadradas, redondas. En la actualidad existen pilas artificiales y naturales.

Las pilas artificiales generalmente están compuestas de contenedores plásticos interconectados con un sistema de tuberías para suministrar el flujo de agua. Estos tipos de pilas tienen un manejo mucho más fácil de controlar ya que están aisladas totalmente del entorno natural disminuyendo

Capítulo 2: Marco Teórico

de esta manera posibles contactos con factores ambientales que impacten el crecimiento del camarón.

Las pilas naturales generalmente están construidas en el suelo y se refuerzan con muros de tierra. El agua es controlada a través de canales y reservorios utilizando un sistema de tubería y bombeo. También, se puede utilizar un sistema de aireación que permita el mejor recirculamiento de las corrientes de agua y propicie un mejor ambiente para el crecimiento del camarón. Este tipo de pilas tiene un reto particular y es controlar los factores ambientales que puedan influir e impactar el crecimiento de los organismos que habitan en ellas.



Figura 2: Vista área de las Pilas de una granja

Capítulo 2: Marco Teórico

Especies de Camarón

A lo largo del mundo entero existe una gran variedad de especies de camarón. Estas se diferencian por su tamaño, color, aspecto. También, cada especie posee características particulares de adaptación al entorno en el que viven. Algunas especies, toleran bajas temperaturas, otras altas temperaturas. Algunas son de fuentes de agua dulce, otras son de agua marina. [15] [16] [17] [18]



Figura 3: Especies que habitan al norte del golfo de San Lorenzo, Canadá

Camarón Patiblanco (*Litopenaeus vannamei*)

En centro américa, especialmente, Nicaragua la especie que se cultiva en las granjas es conocida como camarón patiblanco (*Litopenaeus vannamei*; conocido anteriormente como *Penaeus vannamei*). [19] [20]

Jerarquía Taxonómica

Kingdom	Animalia – Animal, animaux, animals
Subkingdom	Bilateria
Infra kingdom	Protostomia
Superphylum	Ecdysozoa
Phylum	Arthropoda – Artrópode, arthropodes, arthropods
Subphylum	Crustacea Brünnich, 1772 – crustacés, crustáceo, crustaceans
Class	Malacostraca Latreille, 1802
Subclass	Eumalacostraca Grobben, 1892
Superorder	Eucarida Calman, 1904 – camarão, caranguejo, ermitão, lagosta, siri
Order	Decapoda Latreille, 1802 – crabs, crayfishes, lobsters, prawns, shrimp, crabes, crevettes, écrevisses, homards
Suborder	Dendrobranchiata Bate, 1888
Superfamily	Penaeoidea Rafinesque, 1815 – penaeoid shrimps, crevettes pénaéides
Family	Penaeidae Rafinesque, 1815 – penaeid shrimps, crevettes pénaéides
Genus	Litopenaeus Pérez Farfante, 1969
Species	<i>Litopenaeus vannamei</i> (Boone, 1931) – camarón patiblanco, crevette pattes blanches, whiteleg shrimp

Figura 4: Jerarquía Taxonómica de *Litopenaeus vannamei* [21]

Capítulo 2: Marco Teórico

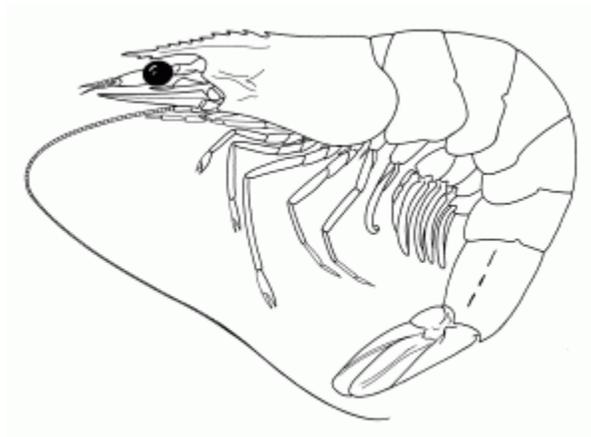


Figura 5: Camarón Patiblanco (*Litopenaeus vannamei*) [20]

Países Productores

L. vannamei es una especie que se encuentra en diversas regiones del mundo entre estas:

- Norte américa
- Centro américa
- Sur américa
- Asia
- Oceanía



Figura 6: Principales países productores de Camarón Patiblanco [20]

Capítulo 2: Marco Teórico

Hábitat

Es una especie de hábitat de agua marina y vive principalmente en aguas tropicales donde las temperaturas de sus aguas son mayores a los 20° centígrados durante todo el año. En su hábitat natural, cuando son adultos viven y desovan en mar abierto. Sin embargo, cuando están en la etapa de post larva ellos migran hacia estuarios, lagunas o manglares. Los machos se convierten en adultos después de alcanzar los 20 gramos de peso y las hembras después de los 28 gramos con un tiempo aproximado entre 6 a 7 meses. [20]

Técnicas de Cultivo

Las granjas pueden ser divididas o clasificadas en cuatro categorías diferentes: extensivas, semi intensivas, intensivas y súper intensivas. Las cuales representan una densidad de población baja, media, alta y extremadamente alta.

Extensivas

Son aquellas granjas que tienen pilas de un tamaño grande, entre 5 y 10 hectáreas algunas veces alcanzando tamaños alrededor de 30 hectáreas con profundidades entre 0.7 y 1.2 metros. Este tipo de granja son comúnmente encontradas en latino américa y éstas no necesitan mucho movimiento de agua, ni aireación. La densidad de siembra es de 4 a 10 PL/m² y se obtienen rendimientos de cosechas aproximadamente de 150 a 500 kg/ha/ciclo; con ciclos de producción de 1 a 2 por año.

Semi Intensivas

Son aquellas granjas que tienen pilas de un tamaño mediano, entre 1 y 5 hectáreas con profundidades entre 1 y 1.2 metros con aireación. La densidad de siembra es de 10 a 30 PL/m² y se obtienen rendimientos de cosechas aproximadamente de 500 a 2,000 kg/ha/ciclo; con 2 ciclos de producción por año.

Intensivas

Son aquellas granjas que tienen pilas de un tamaño pequeño, entre 0.1 y 1 hectárea con profundidades mayores a 1.5 metros normalmente son de tierra pero algunas veces se utilizan liners⁵ para mejorar la calidad del agua. La aireación es un factor muy importante para la circulación y oxigenación del agua. La densidad de siembra es de 60 a 300 PL/m² y se obtienen rendimientos de cosechas aproximadamente de 7,000 a 20,000 kg/ha/ciclo; con ciclos de producción de 2 a 3 por año.

Súper Intensivas

Son aquellas granjas que tienen raceways⁶ de un tamaño extremadamente pequeño, de 282 m². La densidad de siembra es de 300 a 400 juveniles entre 0.5 a 2 gramos/m² y se obtienen

⁵ Liners: son estructuras plásticas que se instalan al fondo de las pilas de tierra para mejorar sabor y color del camarón.

⁶ Raceways: son un sistema de canalización donde se cultiva el camarón.

Capítulo 2: Marco Teórico

rendimientos de cosechas aproximadamente de 28,000 a 68,000 kg/ha/ciclo; con ciclos de producción de 3 a 4 por año.



Figura 7: Raceways utilizado para el cultivo súper intensivo del camarón [22]

Técnicas de Cosecha

Las pilas de granjas extensivas y semi intensivas son cosechadas por medio del drenaje o vaciado del agua a través de una compuerta o esclusa durante la marea baja. En esta compuerta se coloca una red para capturar los camarones y trasladarse en contenedores con hielo.

En pilas de granjas intensivas y súper intensivas pueden utilizarse máquinas cosechadoras que succionan tanto el agua como el camarón para luego ser escurrido y trasladado a contenedores con hielo. También, pueden utilizarse la técnica de acorralar el camarón a un lado de la pila y utilizar redes o baldes perforados para capturar el camarón.

Capítulo 2: Marco Teórico



Figura 8: Cosecha del camarón

Manejo y Maquila

Al cosechar el camarón de la pila se almacena en contenedores con hielo para sacrificarlos a una temperatura aproximadamente entre 0 a 4 °C; como medida preventiva, generalmente se mezcla el camarón con metabisulfito de sodio para prevenir la melanosis y cabeza roja. El camarón se mantiene en hielo dentro de contenedores aislados y es transportado por camión hacia las plantas de proceso o mercados. En las plantas de proceso, el camarón es colocado en tolvas con hielo y se limpia y clasifica de acuerdo a los tamaños o tallas estándares de exportación. El camarón es procesado y congelado a través de proceso IQF⁷ a temperaturas de -10 °C y almacenado a temperaturas de -20 °C para ser exportado a través de barcos o avión. Debido al crecimiento de la demanda las plantas de proceso operan líneas de valor agregado para obtener márgenes mayores de ganancias.

⁷ IQF (Individually Quick Frozen) Congelado rápido individual es un proceso de congelado utilizado en diferentes industrias alimenticias.

Capítulo 2: Marco Teórico



Figura 9: Maquila de camarón



Figura 10: Valor agregado al proceso de camarón

Calidad del Agua

Es de extrema importancia cuidar y mantener observada la calidad del agua, ya que ésta, provee el ambiente necesario para el crecimiento del camarón. Los nutrientes que el camarón utiliza se encuentran presentes en este ambiente. Es por ello, que se debe tener un control estricto y riguroso de la calidad del agua donde se encuentra el camarón. La degradación de la calidad del agua es perjudicial para la salud del camarón

Para llevar a cabo éste control se deben realizar lecturas diarias y sistematizadas en las pilas donde se desarrolla el camarón. Estas lecturas ayudan a mantener un ambiente apropiado para el

Capítulo 2: Marco Teórico

crecimiento del mismo. Los parámetros y sus tipos que normalmente se utilizan; pero que no están restringidos a la lista, son:

Tipos de Parámetros

Para monitorear el desarrollo del camarón es importante tomar en cuenta diferentes factores que influyen en el crecimiento del animal. Estos factores son medidos a través de parámetros que se pueden clasificar en: físicos, químicos y biológicos. [23]

Parámetros

Se define parámetro como: “1.m Dato o factor que se toma como necesario para analizar o valorar una situación.” [24] Es por ello que se convierte en la base del monitoreo de la calidad del agua para el buen manejo del cultivo del camarón.

Parámetros Físicos-

Son aquellos parámetros que miden características o factores de fenómenos o variables naturales. Ejemplo de estos parámetros son [25]:

Tabla 1: Parámetros Físicos

Nombre del Parámetro	Descripción
Temperatura	Es la medida de la energía termal promedio de una sustancia.
Oxígeno Diluido	Es el nivel de oxígeno no compuesto libre presente en el agua u otros líquidos.
pH	Es una cifra entre 0 y 14 de una escala logarítmica que define que tan ácido o base es un cuerpo de agua.
Salinidad	Es la concentración total de todas las sales disueltas en el agua.
Luminosidad / Claridad	Está determinada por la profundidad que alcanzan los rayos solares dentro del agua.

Parámetros Químicos

Son aquellos parámetros que miden características o factores de compuestos y/o reacciones químicas. Ejemplo de estos parámetros son [26] [27] [28] [29] [30] [31] [32]:

Tabla 2: Parámetros Químicos

Nombre del Parámetro	Descripción
Amonio Tóxico (NH ₃)	Es un compuesto químico gaseoso incoloro, de olor penetrante, bastante soluble en el agua y en estado líquido fácilmente evaporable.
Cloruros	Son todas las sales presentes en el agua.

Capítulo 2: Marco Teórico

Nitritos (NO₂-N)	Son compuestos nitrogenados presentes en el agua y que afectan a los organismos fotosintetizadores.
Alcalinidad	Es la capacidad del agua para neutralizar ácidos o aceptar protones.
Dióxido de Carbono	Es un gas incoloro, inodoro, no es inflamable y ligeramente ácido.
Dureza (CaCO₃)	Es la suma de las durezas individuales debidas a los iones de calcio, magnesio, estroncio y bario en forma de carbonato o bicarbonato presentes en el agua.
Sulfatos	Son los iones de sulfatos presentes en el agua.
Fosfatos	Es la concentración de fósforo presente en el agua.

Parámetros Biológicos

Son aquellos parámetros que miden características o factores de organismos vivos. Ejemplo de estos parámetros son [33]:

Tabla 3: Parámetros Biológicos

Nombre del Parámetro	Descripción
Coliformes Totales	Son bacterias conocidas como <i>Enterobacteriaceae</i> lactosa-positivas y se caracterizan por su capacidad para fermentar la lactosa con producción de ácido y gas, más o menos rápidamente, en un período de 48 horas y con una temperatura de incubación comprendida entre 30-37°C.
Coliformes Fecales	Son un tipo de Coliformes totales conocidos como <i>Escherichia coli</i> además fermentan la lactosa con producción de ácido y gas en 24-48 horas a temperaturas comprendidas entre 44 y 45°C en presencia de sales biliares.
Conteo de Sobrevivencia	Cantidad de organismos individuales presentes.
Conteo de Enfermedades	Cantidad de organismos individuales enfermos.
Concentración de Algas	Cantidad o porcentaje de algas presente en el agua.

Capítulo 2: Marco Teórico

Análisis de Regresión Lineal

Introducción

Es una técnica estadística que se utiliza para estudiar la relación entre las variables. [9] [34] [35] Existen dos métodos de análisis de regresión: simple y múltiple. El método simple consiste en analizar la relación existente entre una variable dependiente y una independiente. En el método múltiple la relación analizada es de 2 o más variables independientes y una dependiente. [9] [36]

El ámbito de uso del análisis es muy amplio y puede utilizarse en las áreas sociales, económicas; así como, la física. En todas ellas se trata de explorar y cuantificar la relación de las variables involucradas en los fenómenos estudiados para poder predecir los resultados basados en un conjunto de datos. Tanto en la regresión simple como la múltiple la predicción se realiza basadas en 2 o más variables independientes o predictores (X_1, X_2, \dots, X_n) con una variable dependiente o criterio (Y) [9]

Línea de tendencia

Los datos de las variables pueden ser representados por un diagrama de dispersión que nos da una pista visual sobre el tipo de relación que existe entre las variables. Por ejemplo, los datos de la estatura y pesos de un grupo de personas. [37]

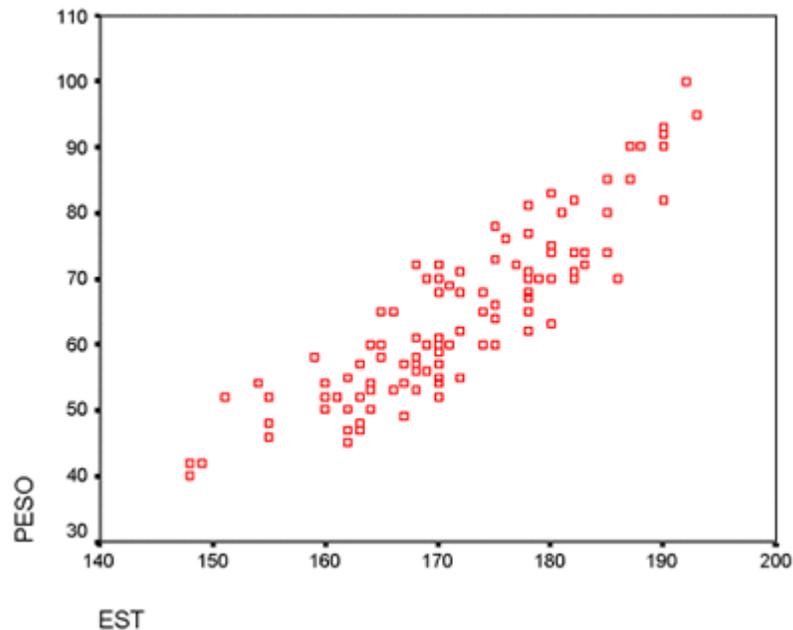


Figura 11: Diagrama de dispersión

En esta gráfica se muestran los datos de las personas estableciendo la variable “Estatura” como la variable independiente y “Peso” como la variable dependiente. La relación mostrada acá tiene una tendencia positiva; es decir, a mayor valor de la estatura habrá una estimación mayor del peso.

Capítulo 2: Marco Teórico

La tendencia de estos datos es lineal simple; ya que consta de la relación de dos variables. También, se puede observar por medio de una simple inspección visual que la figura geométrica que mejor la representa es una línea recta. Aplicando el método de regresión lineal tendríamos un resultado como el siguiente:

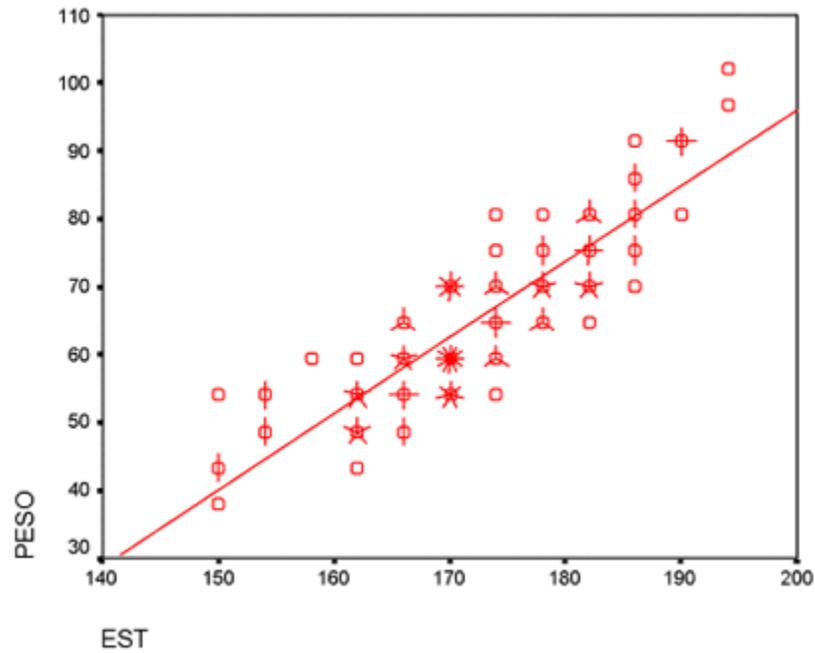


Figura 12: Línea de tendencia

Capítulo 2: Marco Teórico

La fórmula utilizada está basada en el método de los mínimos cuadrados ordinarios (MCO); la cual considera que la recta que mejor se ajusta es aquella que minimiza la suma de los cuadrados de los residuos y está expresada por la siguiente fórmula [37] [38] [8] [39] [40] [7]:

$$\hat{Y} = \beta_1 + \beta_2 X$$

Dónde:

$$\beta_1 = \bar{Y} - \beta_2 \bar{X}$$

$$\beta_2 = \frac{S_{xy}}{S_x^2}$$

Dónde:

\bar{Y} = Es la media de los valores Y

\bar{X} = Es la media de los valores X

S_{xy} = Es la covarianza de los valores X e Y

S_x^2 = Es la varianza de los valores X

Coefficiente de Pearson o Coeficiente de determinación R^2

La bondad del ajuste se evalúa por medio del coeficiente de determinación R^2 y la dispersión de los puntos alrededor de la recta estimada se mide utilizando el error típico de la estimación S_u . Estas medidas se definen como:

$$e = Y - \hat{Y}$$

$$S_u = \sqrt{\frac{\sum e^2}{n - 2}}$$

$$R^2 = \frac{SCR}{SCT}$$

Dónde:

SCR = Suma total de cuadrados. Variación total de Y en la muestra.

SCT = Suma de cuadrados de la regresión. Variación total explicada por la recta ajustada.

Capítulo 2: Marco Teórico

Por lo tanto, R^2 indica la proporción de variación total explicada por la correlación lineal que existe entre las dos variables. Puede obtener valores entre 0 y 1. Los valores de R^2 aproximados a 1 indican que existe una relación muy fuerte entre los datos; por lo tanto, la recta ajusta representa un buen modelo para explicar el comportamiento de la variable dependiente en relación de la independiente y poder realizar predicción más precisas sobre las mismas. [37] [9] [38]

Arquitectura de Aplicaciones Cliente – Servidor

Es un tipo de arquitectura utilizada principalmente en sistemas de aplicaciones distribuidas que dividen sus tareas o cargas de trabajo en diferentes equipos. Principalmente, los recursos o servicios son brindados por equipos llamados servidores. Los equipos que realizan las peticiones de estos recursos o servicios son llamados clientes. De manera general, los equipos clientes nunca comparten recursos o servicios a otros clientes. La siguiente figura muestra un esquema simplificado y generalizado de la arquitectura cliente – servidor. De acuerdo a estadísticas más del 65% de las empresas utilizan aplicaciones que corren bajo la arquitectura Cliente – Servidor. [41] [42]

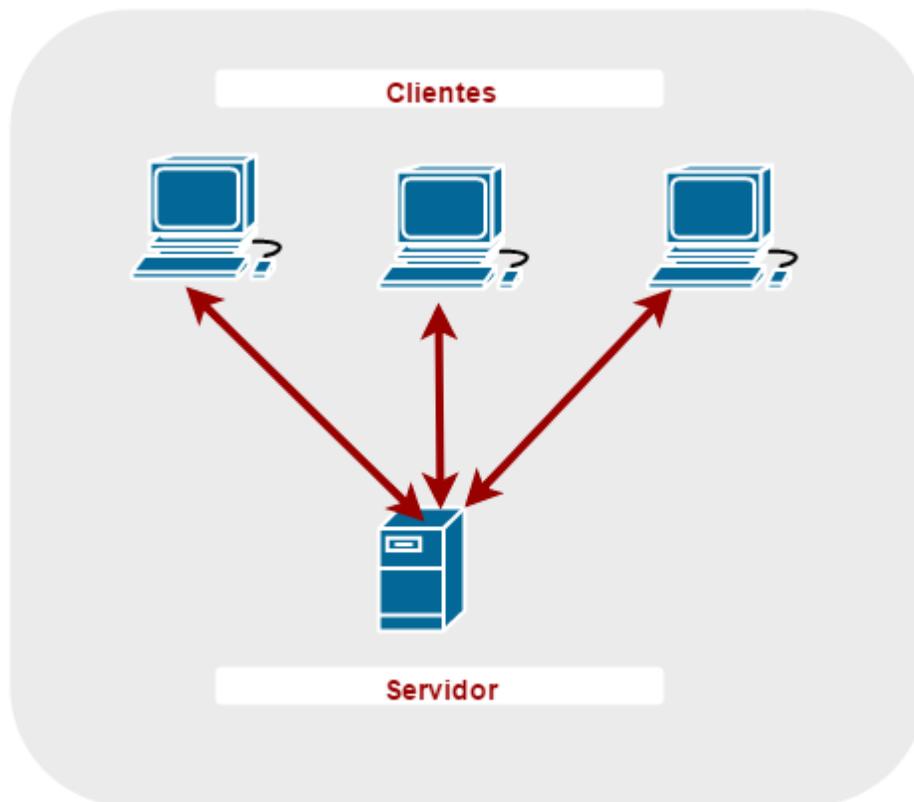


Figura 13: Arquitectura Cliente - Servidor

Capítulo 2: Marco Teórico

Arquitectura de Una Capa

Este tipo de arquitectura; hoy casi completamente en desuso, permitía la ejecución de aplicaciones sin ninguna conexión fuera del equipo donde se utilizaba. La interacción y almacenamiento de información ocurría localmente y era utilizada generalmente por sistemas desarrollados utilizando modelos de lenguajes monolíticos. Estos sistemas no eran aptos para soportar el escalamiento de usuarios. [41]

Arquitectura de Una Capa Aplicación Monolítica

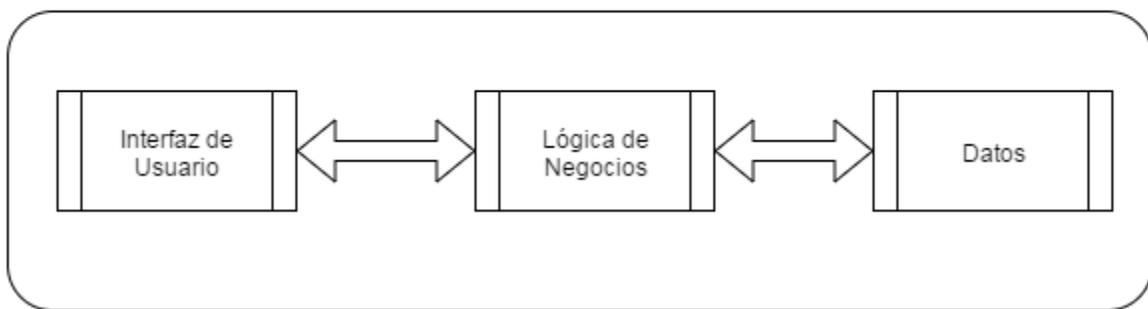


Figura 14: Arquitectura Una Capa – Aplicación Monolítica

Capítulo 2: Marco Teórico

Arquitectura de Dos Capas

Esta arquitectura permitía desarrollar aplicaciones divididas en dos capas (lógicas y físicas). Estas capas físicas residían en dos equipos diferentes: el cliente que utiliza el usuario a través de una aplicación con IGU y el servidor que ejecuta servicios; por ejemplo, de base de datos, que utiliza la aplicación cliente. En este modelo la información está centralizada y disponible para cualquier [aplicación] cliente que la consulte o consuma. A este tipo de sistemas se les conocía como modelo “fat client” ya que todo el poder de procesamiento ocurría en el equipo cliente.

Sin embargo, este modelo tiene el problema que la concentración de recursos es proporcionada por un solo servicio (servidor físico) en el cual dichos recursos se veían consumidos exponencialmente según la demanda de usuarios crece. Esto conlleva a un problema serio para empresas cuyo ritmo de crecimiento es grande y son incapaces de soportar la demanda para aplicaciones de misión crítica. [41]

Arquitectura de Dos Capas

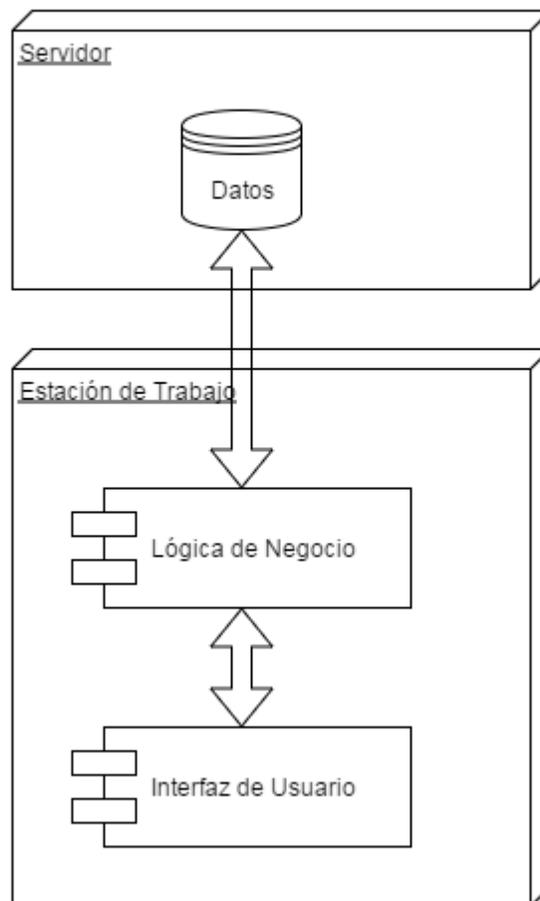


Figura 15: Arquitectura de Dos Capas

Capítulo 2: Marco Teórico

Arquitectura de Tres Capas

Esta arquitectura surge como una necesidad para superar el problema de escalabilidad que tiene la arquitectura de dos capas. En ésta se introduce una capa adicional conocida comúnmente como capa media o capa de lógica de negocios. Esta nueva capa puede ser una división lógica de servicios o física de servidores con sus servicios. En este caso la aplicación en lugar de comunicarse directamente con el servidor de datos; lo hace, con un servidor de aplicaciones que actúa como un intermediario. Este servidor se especializa en administrar y manejar los recursos que solicita al servidor de datos; así como los servicios, que sirve a los diferentes clientes. A las aplicaciones desarrolladas utilizando este tipo de arquitectura son conocidas como “thin client” ya que todo el poder de proceso es distribuido en las capas de servidores. [41]

Arquitectura de Tres Capas

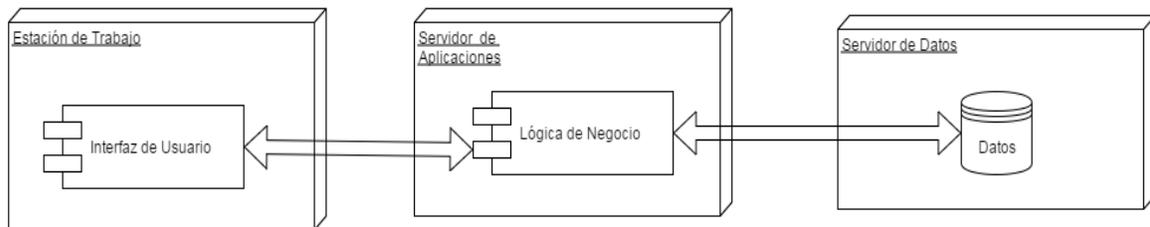


Figura 16: Arquitectura de Tres Capas

Capítulo 2: Marco Teórico

Arquitectura de Dos Capas vs Tres Capas

Las diferencias entre estas dos arquitecturas se pueden resumir según la siguiente tabla:

Tabla 4: Arquitectura de Dos Capas vs Tres Capas

	Dos Capas	Tres Capas
Administración del Sistema	Compleja (servicios dispersos en todos los clientes)	Menos compleja (servicios centralizados)
Seguridad	Baja (seguridad a nivel de datos)	Alta (seguridad a nivel del servicio)
Encapsulación de los datos	Baja (las tablas están expuestas)	Alta (el cliente invoca métodos o servicios)
Rendimiento	Pobre	Bueno
Escalabilidad	Pobre (limitación en la administración de enlaces de los clientes)	Excelente (enlaces de los clientes concentrados y se puede distribuir la carga de trabajo en múltiples servidores)
Reutilización de la aplicación	Pobre	Excelente (los servicios pueden ser reutilizables)
Facilidad de desarrollo	Alta	Baja (mejorando con los herramientas de desarrollo modernas)
Facilidad de Implementación	Alta	Baja (mayor conocimiento de administración)
Infraestructura entre servidores	No	Si (a través de capas intermedias entre servidores)
Integración con aplicaciones heredadas (Legacy)	No	Si (a través de la encapsulación de servicios o métodos)
Soporte de Internet	Pobre	Excelente
Soporte a Bases de Datos Heterogéneas	No	Si (pueden utilizar múltiples bases de datos dentro de la misma transacción de negocio)
Opciones de comunicaciones	Pobre (solamente conexiones síncronas)	Alta (soporta conexiones síncronas y asíncronas)
Flexibilidad de arquitectura de hardware	Limitada	Excelente (todas las capas pueden existir en diferentes servidores físicos)
Disponibilidad	Pobre	Excelente

Capítulo 2: Marco Teórico

Arquitectura de N Capas

Esta arquitectura se considera una evolución o extensión de la arquitectura anterior. Con ella se puede hacer separaciones adicionales a las capas existentes creando nuevas capas. La separación puede ser lógica o física y también ocurrir del lado del cliente o del servidor. Si se realiza del lado del cliente; se puede obtener una nueva aplicación con una actualización total de la interfaz de usuario, sin tener que modificar la capa de lógica de negocios. Esto permite dar soporte a varias versiones de una aplicación dándole soporte a aplicaciones anteriores.

Adicionalmente, se ha introducido el concepto de transacciones distribuidas. Estas transacciones deben coexistir en forma secuencial o de cascada. Esto permite que a que la transacción que inicia otra transacción no cierre sin que la segunda transacción haya finalizado; asegurando, el principio de consistencia. [41]

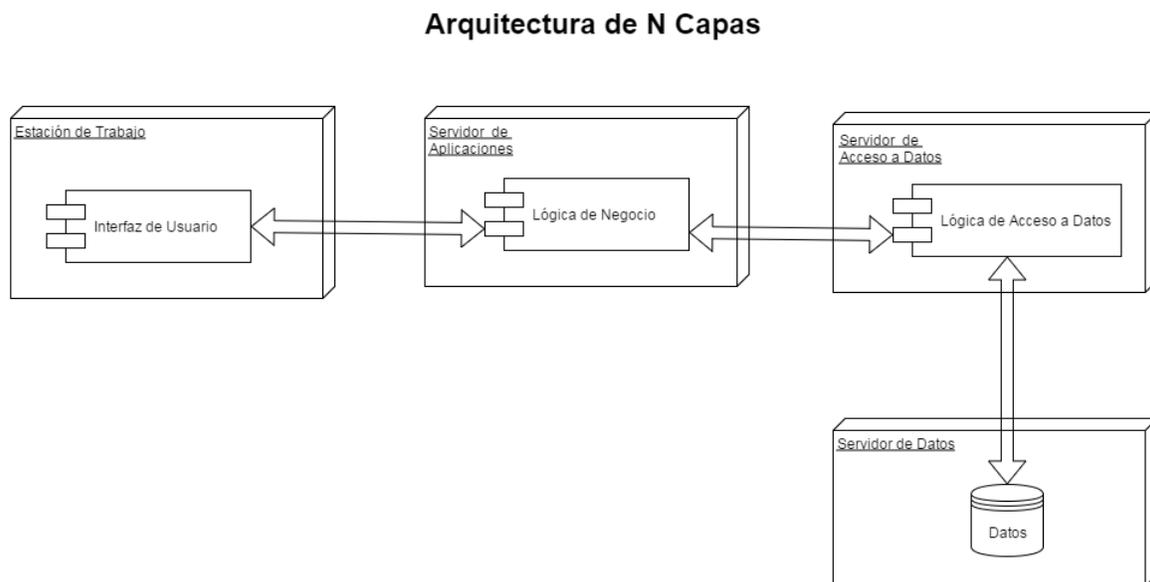


Figura 17: Arquitectura de N Capas

Programación Orientada a Objetos

Introducción

Antes de la POO el paradigma más utilizado era el de programación estructurada; donde su principal enfoque consistía en acciones y procedimientos lógicos que se utilizaban para procesar los datos. Éste enfoque suponía un problema al tratar de resolver los paradigmas de reutilización y algo más.

La POO se define como: Un modelo de lenguaje de programación enfocado en “objetos” que representan la “encapsulación” de un conjunto de datos o “estado” y métodos o

Capítulo 2: Marco Teórico

“comportamiento” que se utilizan para manipular a éstos; así como, “sucesos” que son utilizados para comunicar “mensajes” entre diferentes objetos. [43] [44] [45]

Objeto

Son estructuras utilizadas para representar los objetos que encontramos en el mundo real. Se considera como la encapsulación genérica de datos conocidos como atributos que representan las características del objeto y procedimientos o métodos que se utilizan para manipularlos. [43] [46]

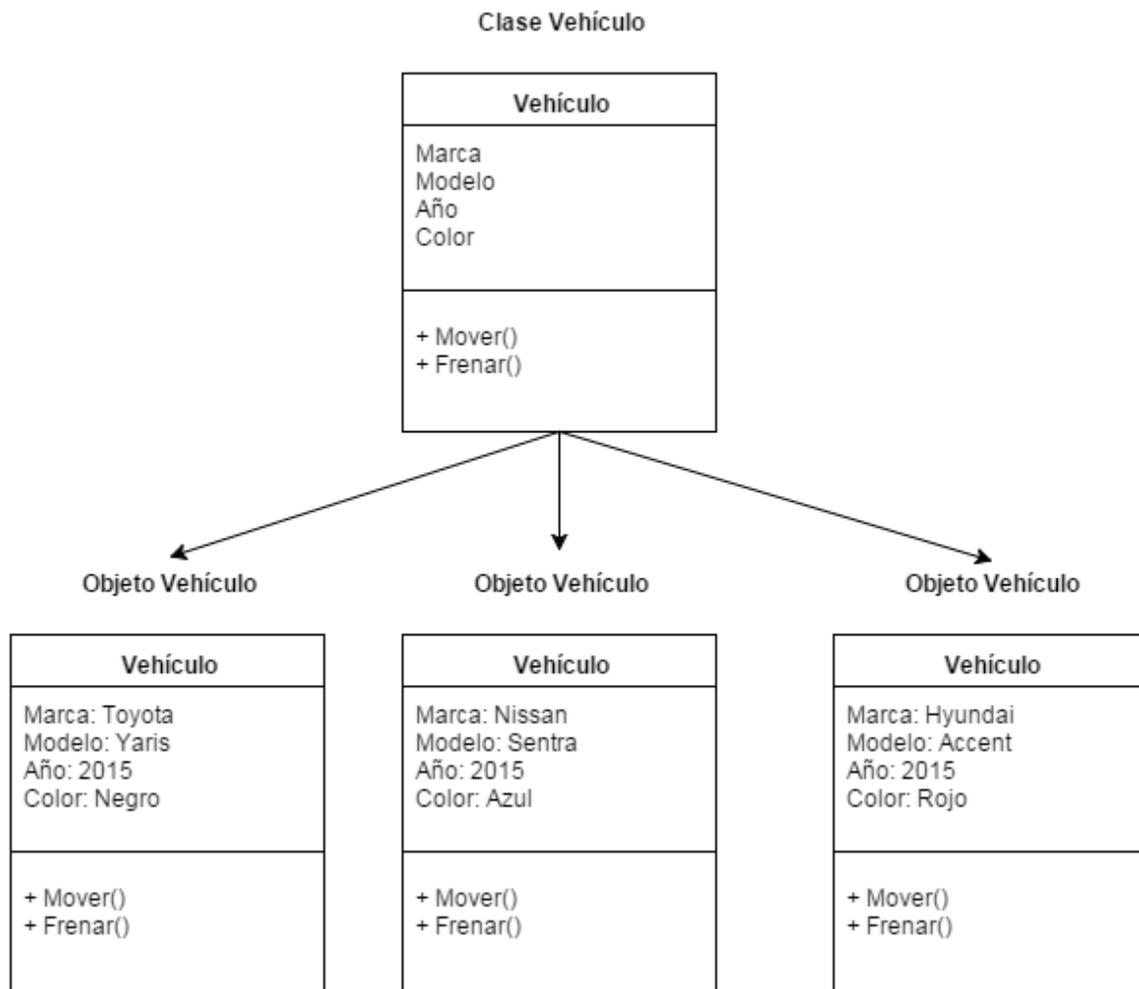


Figura 18: Representación de Objetos

Al desarrollar sistemas utilizando objetos se pueden obtener los siguientes beneficios:

Modularidad

El código fuente de un objeto puede ser escrito y mantenido separado del código fuente de otros objetos. Una vez creado un objeto puede ser utilizado a lo largo de todo el sistema.

Capítulo 2: Marco Teórico

Información oculta – Encapsulación

Al utilizar solamente los métodos expuestos en el objeto, la implementación interna del objeto permanece oculta del mundo exterior.

Reutilización del código

Si un objeto existe (aunque sea creado por otro desarrollador) éste puede ser utilizado en el sistema.

Interoperabilidad

Si un objeto presenta problemas, fácilmente puede reemplazarse por otro que realice las mismas funciones.

Mensajes

Los objetos reciben, interpretan y responden a “mensajes” de otros objetos. Esto es una característica diferenciadora entre el paradigma de POO y programación estructurada. Existe una correspondencia asociada de un mensaje y la función miembro de una clase.

Métodos

Es la definición del comportamiento que tendrá un objeto. Cada método implementado define un comportamiento específico del objeto.

Clases

Es un tipo de datos definido por el usuario que se utiliza como plantilla o plano para crear objetos o instancias de una clase. Utilizando la analogía de la construcción se puede referir a que la clase es equivalente al plano y un objeto es equivalente al edificio construido en base a ese plano. Análogamente, los objetos o instancias son construidos en base a las clases.

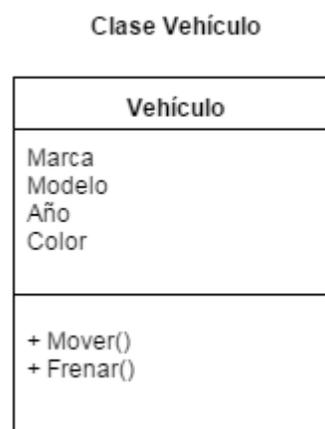


Figura 19: Representación de una Clase

Capítulo 2: Marco Teórico

Características

Las características fundamentales de la POO son: abstracción, encapsulamiento, herencia y polimorfismo.

Abstracción

Es la separación entre la especificación del objeto de dato y su implementación. Es decir, consiste en la generalización conceptual de los atributos y propiedades de un determinado conjunto de objetos.

Encapsulamiento

Es la práctica de ocultar los detalles de la implementación del objeto de datos del mundo exterior.

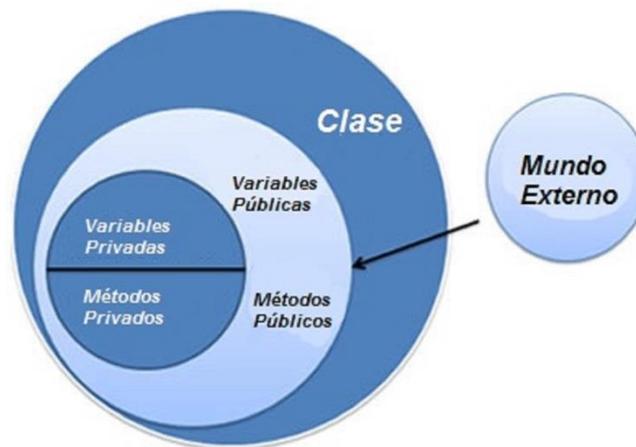


Figura 20: Abstracción y Encapsulamiento

Herencia

Es el mecanismo que se utiliza para compartir automáticamente métodos y atributos entre clases y subclases. En algunos lenguajes es conocido como la derivación de clases. Donde una clase padre o base, hereda sus atributos y métodos a una clase hija o derivada. Este concepto es la clave de la reutilización del código en POO.

Capítulo 2: Marco Teórico

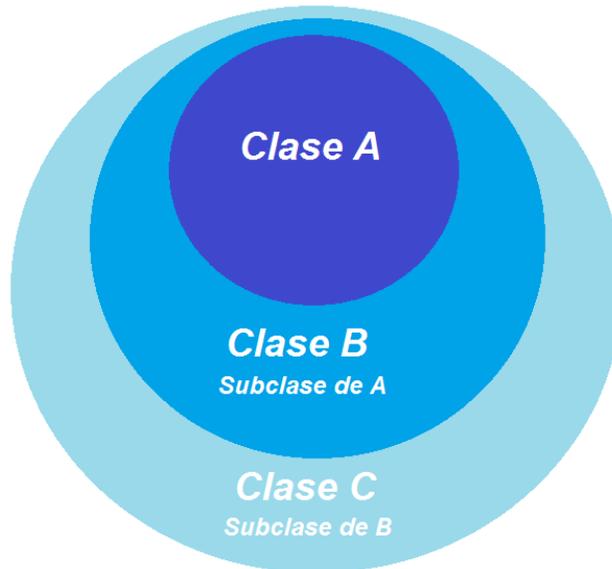


Figura 21: Herencia

Polimorfismo

Es la característica de poder implementar múltiples formas de un mismo método. Esto se refiere a la habilidad de poder sobre escribir los métodos de las clases derivadas para que soporten funcionalidad adicional en las mismas.

Lenguaje de Programación C#

Microsoft Corporation comenzó alrededor del año 2000 con el desarrollo de lenguajes de última generación llamados lenguajes gestionados. Estos lenguajes se caracterizan por ejecutarse de forma segura en una máquina virtual llamada CLR⁸ y que son parte del .Net Framework⁹ a los cuales pertenecen los siguientes lenguajes:

- Visual C#
- Visual Basic
- Visual F#
- Visual C++
- Javascript

El ciclo de ejecución de un proyecto de Visual C# se resume en la siguiente figura.

⁸ CLR: (Common Language Runtime) Lenguaje Común de Ejecución

⁹ .Net Framework: Conjunto comunes de librerías disponibles para el uso del CLR.

Capítulo 2: Marco Teórico

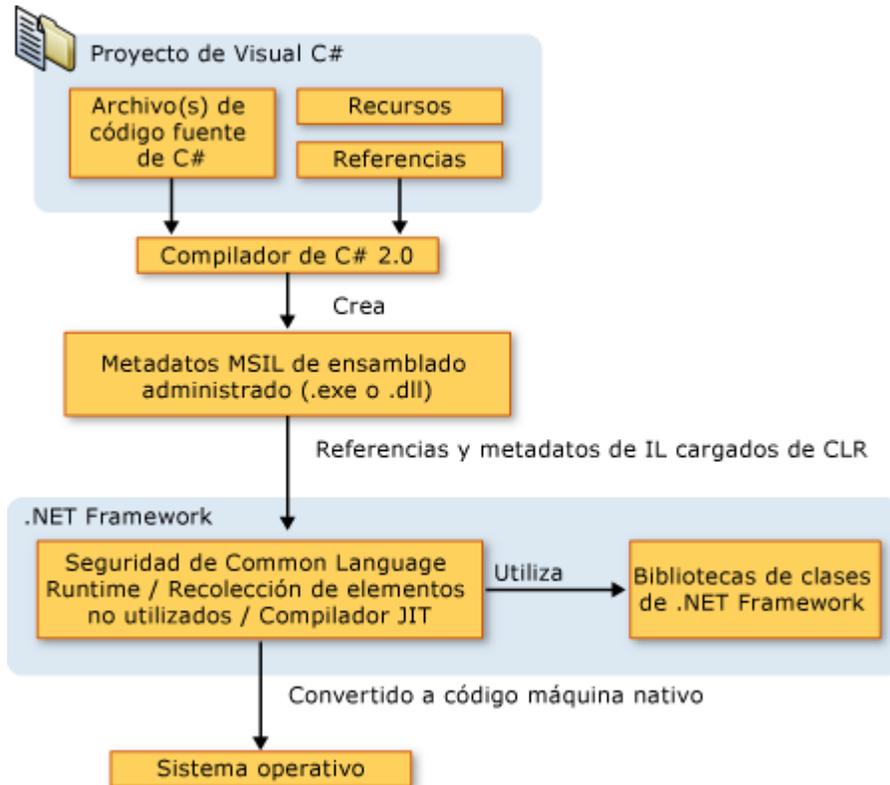


Figura 22: Ciclo de Ejecución C#

En general, todos los lenguajes gestionados utilizan el mismo ciclo de ejecución, permitiendo, la ejecución segura a través de la máquina virtual CLR. [47] [48] [49]

Capítulo 2: Marco Teórico

Seguridad Integrada de Windows

Autenticación integrada de Windows permite a los usuarios iniciar sesión con sus credenciales de dominio, utilizando Microsoft NTLM¹⁰ o Kerberos¹¹. La autenticación de Windows tiene una mayor utilidad para ambientes intranet; tales como, redes empresariales con dominio. [50]

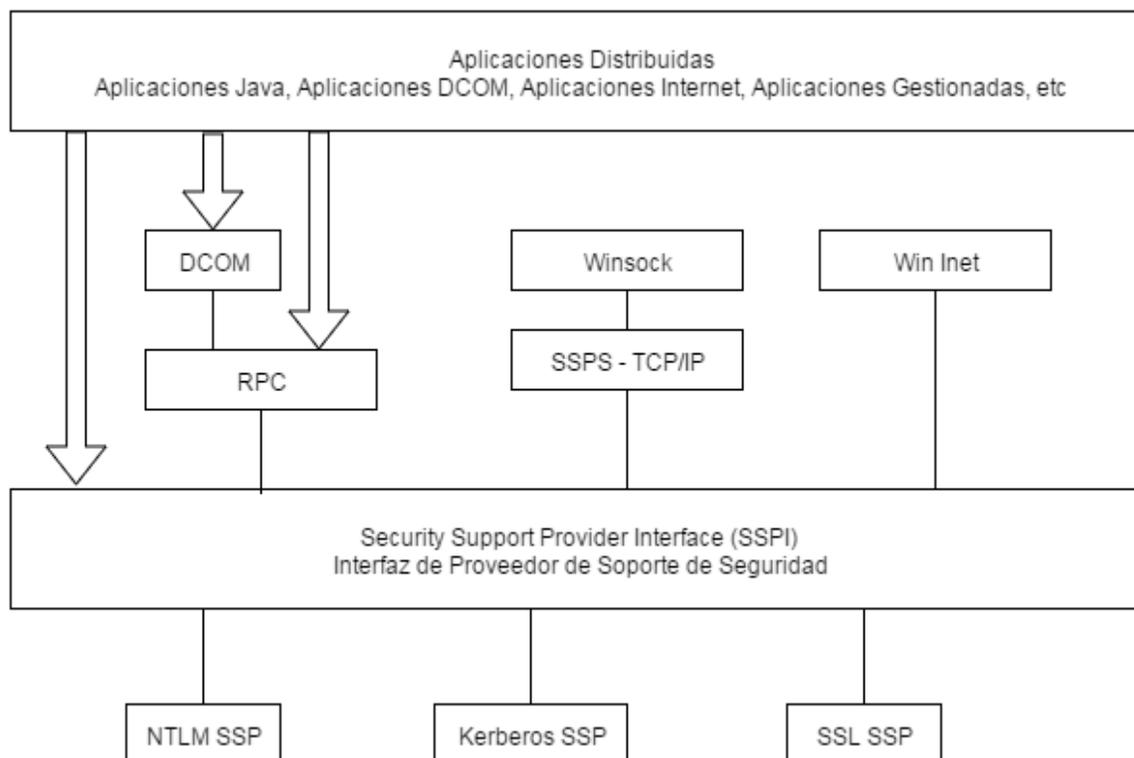


Figura 23: Modelo de Seguridad NT y SSPI

Autenticación con Microsoft NTLM (Desafío/Respuesta)

Es un protocolo de autenticación utilizado en redes que incluyen equipos corriendo sistemas operativos de Windows y en sistemas autónomos. El paquete de seguridad de Microsoft Kerberos agrega mayor seguridad que NTLM a los sistemas de la red. Aunque Microsoft Kerberos es el protocolo de elección, NTLM todavía es soportado. NTLM debe ser utilizado para el proceso de autenticación de inicio de sesión en sistemas autónomos.

Las credenciales en NTLM son basadas en datos que se obtienen durante el proceso de inicio de sesión interactivo y consiste de un nombre de dominio, nombre de usuario y el valor hash de una sola vía de la contraseña del usuario. NTLM utiliza un protocolo desafío/respuesta encriptado para autenticar a un usuario sin enviar la contraseña del usuario a través de la red. En lugar de eso, el

¹⁰ NTLM: NT LAN Manager

¹¹ Kerberos: Protocolo de autenticación de red nombrado por el carácter de la mitología griega Cerberos.

Capítulo 2: Marco Teórico

sistema que solicita la autenticación debe realizar unos cálculos para probar que tiene acceso a las credenciales NTLM aseguradas.

La autenticación NTLM interactiva sobre la red típicamente involucra dos sistemas: un sistema cliente, donde el usuario está solicitando la autenticación y un controlador de dominio, donde la información relacionada con la contraseña del usuario es guardada. Las autenticaciones no interactivas, las cuales pueden requerir que un usuario que ya inició sesión solicite el acceso a otro recurso tal como un servidor de aplicación; típicamente, involucra a tres sistemas: un cliente, un servidor y un controlador de dominio que realiza los cálculos en lugar del servidor. [51]

Autenticación Desafío/Respuesta NTLM

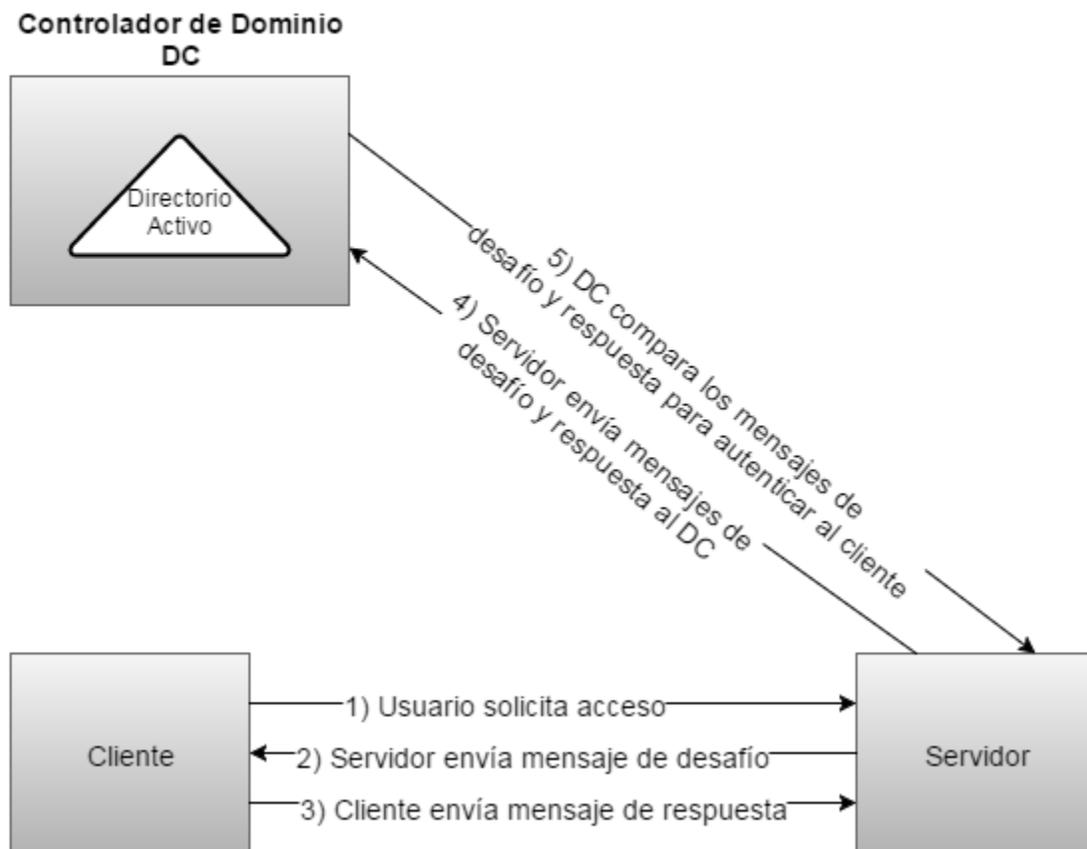


Figura 24: Esquema de Autenticación NTLM [52]

Capítulo 2: Marco Teórico

Autenticación con Kerberos

El protocolo Kerberos define como los clientes interactúan con el servicio de autenticación de red. Los clientes obtienen tickets del Centro de Distribución de Llaves de Kerberos o por nombre en inglés el “Kerberos Key Distribution Center (KDC)”, y ellos presentan los tickets a los servidores cuando se establece la conexión. Los tickets de Kerberos representan las credenciales de red de los clientes.

El protocolo de autenticación de Kerberos provee de un mecanismo para autenticación mutua entre las entidades antes de que se establezca una conexión segura de red. El protocolo asume que las transacciones entre los clientes y servidores toman lugar en una red abierta donde la mayoría de los clientes y servidores no están físicamente seguros y los paquetes viajan a través de la red y pueden ser monitoreados y modificados a voluntad.

Este ambiente asumido es como el que se tiene en Internet hoy día, donde un atacante puede fácilmente hacerse pasar tanto como cliente como un servidor y puede fácilmente intervenir las comunicaciones entre clientes y servidores legítimos. [53]

Autenticación Kerberos

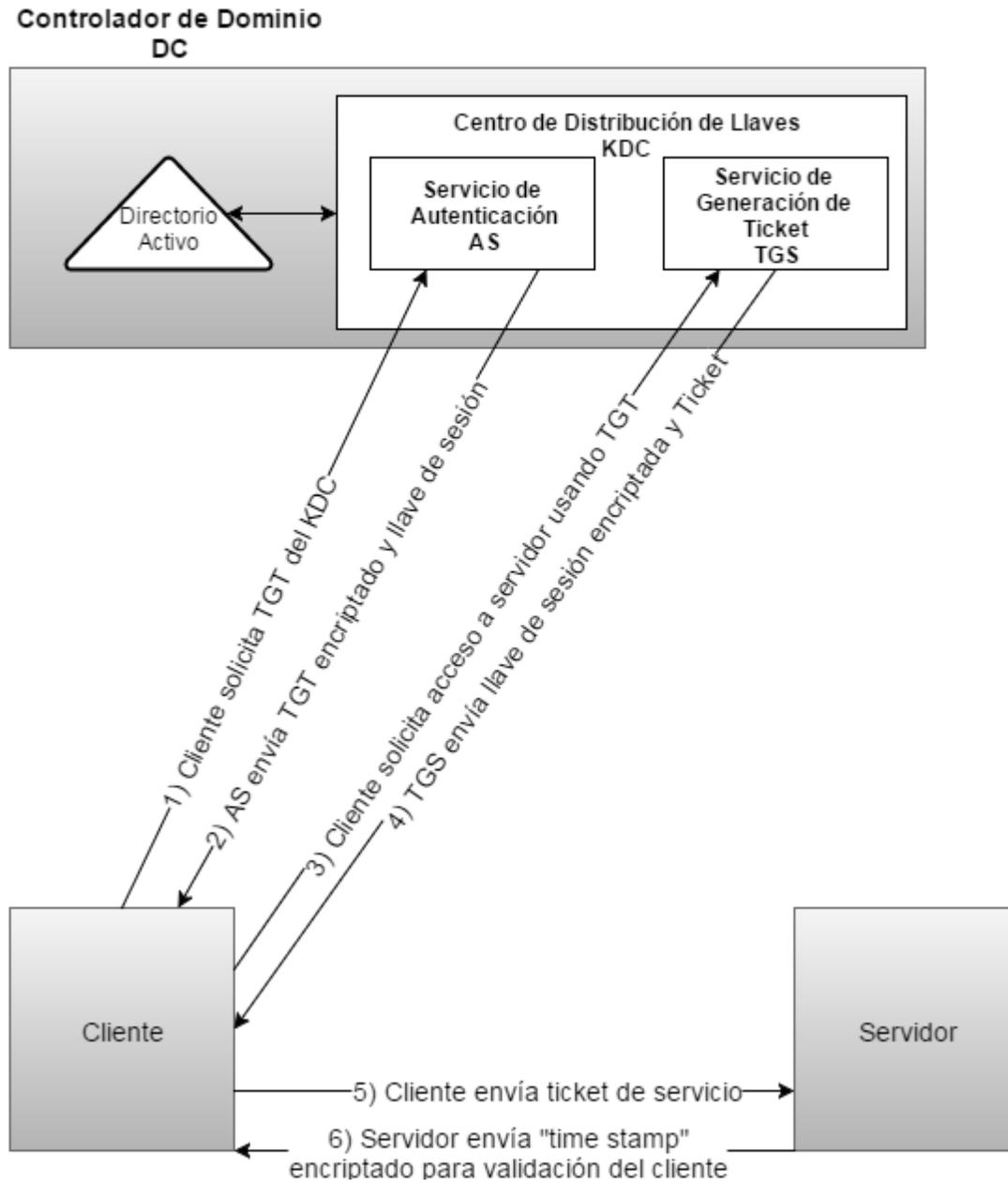


Figura 25: Autenticación Kerberos [52]

Capítulo 2: Marco Teórico

Conceptos Básicos de Autenticación

En un modelo de aplicación cliente - servidor, generalmente, los clientes inician acciones o peticiones a servidores quienes responden a estas peticiones. En el modelo de protocolo Kerberos, cada conexión cliente - servidor comienza con la autenticación. Como consecuencia, cada uno de los clientes y servidores realizarán una secuencia de acciones, designadas para la verificación de las solicitudes de cada uno de los participantes en ambos lados de la conexión. Si la autenticación es satisfactoria, la configuración de la sesión se completa y una sesión cliente - servidor segura se establece.

Llave de Autenticación

Es el mecanismo que utiliza el protocolo Kerberos para evitar revelar públicamente la contraseña. Se basa en el hecho de mantener un secreto públicamente. Para que esto funcione, las dos partes de una transacción deberán compartir una llave de sesión criptográfica la cual también es secreta conocida solamente para ambas partes. La llave es simétrica, es decir, que es una única llave utilizada para encriptar y des encriptar. Una de las partes en el proceso de autenticación prueba que tiene conocimiento de la llave al encriptar un mensaje. La otra parte prueba el conocimiento de la llave al des encriptar el mensaje.

Mensaje del Autenticador

La información en el mensaje autenticador debe ser diferente cada vez que el protocolo de autenticación es ejecutado, de no ser así, el mensaje autenticador encriptado podría ser utilizado por una entidad no autorizada. Al recibir el mensaje autenticador, el servidor des encripta el mensaje; si esto es satisfactorio revisa su contenido y envía parte del mismo encriptado al cliente. El cliente a su vez des encripta la porción del mensaje recibida y lo confirma con su mensaje original. Esto se asegura de hacer una doble verificación.

Distribución de la llave

La técnica de autenticación por llave secreta no explica de qué manera tanto el cliente como el servidor obtienen la llave de sesión que utilizan para las sesiones entre sí. Un servidor se comunica con muchos clientes y por lo tanto necesita diferentes llaves para cada uno de ellos. De igual manera, cada cliente necesita diferentes llaves para cada servidor. La distribución correcta de estas llaves se convierte en un problema.

Adicionalmente, la necesidad de almacenar y proteger muchas llaves de muchos equipos crea por sí mismo un enorme riesgo de seguridad. Kerberos o Cerberos era un carácter de la mitología Griega; un feroz perro de tres cabezas que prevenía la entrada de intrusos vivientes al inframundo. Parecido al guardián mítico, el protocolo Kerberos tiene 3 cabezas: un cliente, un servidor y una tercera parte de confianza que media entre ambos [cliente-servidor]. El intermediario de confianza en este protocolo es el Centro de Distribución de Llaves (Key Distribution Center KDC). El KDC es un servicio que corre en un servidor seguro, mantiene una base de datos con la información de las cuentas para todos los tokens de seguridad (security principals) en su dominio.

Capítulo 2: Marco Teórico

Además de almacenar información adicional de cada token de seguridad el KDC almacena una llave criptográfica conocida solamente por el token y el KDC. Esta es la llave maestra utilizada en los intercambios entre el token de seguridad y el KDC. En la mayoría de las implementaciones del protocolo Kerberos, la llave maestra es derivada utilizando una función hash de la contraseña de un token de seguridad.

Cuando un cliente desea crear una conexión segura con un servidor, el cliente comienza la conexión enviando una petición al KDC y no al servidor mismo con el que quiere conectarse. El KDC crea y envía al cliente una llave de sesión única para el cliente y un servidor; la que utilizan para autenticarse mutuamente. El KDC tiene acceso tanto a la llave maestra del cliente como a la llave maestra del servidor. El KDC encripta la copia de la llave de sesión del servidor usando la llave maestra del servidor y la copia del cliente utilizando la llave maestra del cliente. El KDC satisface el rol de intermediario de confianza porque puede enviar la llave de sesión directamente a cada token de seguridad involucrado en el proceso; sin embargo, en lugar de enviarlo directamente el KDC envía ambas llaves de sesión encriptada al cliente. La llave de sesión para el servidor es incluida en un ticket de sesión.

Tickets de Sesión

En lugar de enviar las llaves de sesión encriptadas a ambos tokens de seguridad, el KDC envía ambas copias de la llave de sesión al cliente. La copia de la llave de sesión del cliente es encriptada utilizando llave maestra del cliente; por lo tanto, no puede ser des encriptado por ninguna otra entidad. La copia de la llave de sesión del servidor es incrustado con datos de autorización acerca del cliente en una estructura de datos llamado ticket. El ticket es encriptado totalmente utilizando la llave maestra del servidor y por lo tanto no puede ser leída ni modificada por el cliente o cualquier otra entidad que no tenga acceso a la llave maestra del servidor. Es responsabilidad completa del cliente de almacenar seguramente el ticket hasta que se contacte con el servidor.

Hay que dejar claro que el KDC solamente provee de un servicio generador de tickets. Los clientes y servidores son responsables cada uno de mantener sus respectivas llaves maestras almacenadas seguramente. Cuando un cliente recibe la respuesta del KDC, extrae el ticket y la copia de su propia llave de sesión y la almacena en una memoria caché segura. Para establecer la conexión segura, el cliente envía al servidor un mensaje que consiste en el ticket; encriptado con la llave maestra del servidor y un mensaje autenticador encriptado con la llave de sesión. Ambos, el ticket y mensaje autenticador son considerados las credenciales del cliente ante el servidor.

Cuando el servidor recibe las credenciales del cliente, éste des encripta el ticket usando su llave maestra, extrae la llave de sesión y utiliza esta llave para des encriptar el mensaje autenticador del cliente. Si todo está bien, el servidor sabe que las credenciales del cliente fueron emitidas por el KDC, quien es la autoridad de confianza. Para realizar la autenticación mutua, el servidor responde encriptando la fecha y hora del mensaje autenticador del cliente utilizando la llave de sesión. Este

Capítulo 2: Marco Teórico

nuevo mensaje encriptado es enviado al cliente. El cliente des encripta el mensaje y lo compara con el valor original de la fecha y hora ubicada en el mensaje autenticador del cliente; si todo está bien, el servidor es autenticado.

Como un beneficio adicional, el servidor no necesita almacenar las llaves de sesión de sus clientes. Es responsabilidad de cada cliente de administrar el ticket para el servidor en su caché y presentar ese ticket cada vez que acceda al servidor. El cliente no necesita acceder al KDC cada vez que desee acceso a un servidor determinado. Los tickets pueden ser reutilizados; sin embargo, como precaución cada ticket tiene una fecha y hora de expiración especificado por el KDC en la estructura del ticket. La duración del ticket va en dependencia de las políticas de Kerberos en el dominio. Normalmente, los tickets tienen una duración aproximada de 8 horas, lo que corresponde con la duración normal de la sesión de un usuario. Cuando el usuario cierra sesión en su equipo, la caché de tickets de ese equipo cliente es vaciado y todos los tickets y las llaves de sesión de cliente son destruidas.

Tickets del servicio de generación de tickets

De la manera, en que el protocolo Kerberos fue originalmente diseñado, una llave maestra para un usuario era derivada de la contraseña prevista por este usuario. Cuando el usuario inicia sesión, el cliente Kerberos en el equipo del usuario acepta la contraseña del usuario y la convierte en una llave de cifrado al pasar el texto a través de una función hash de una sola vía. La cadena hash resultante es la llave maestra del usuario. El cliente [equipo] utiliza esta llave maestra para des encriptar las llaves de sesión recibidas del KDC. El problema con el diseño original es que el cliente necesita utilizar la llave maestra del usuario cada vez que des encripta la llave de sesión del KDC. Esto significa, de que el cliente debe pedir la contraseña al usuario constantemente, lo cual es engorroso o bien almacenar la llave en el equipo lo cual se convierte en un riesgo de seguridad. La solución del protocolo de Kerberos a este problema original es que el cliente obtenga una llave temporal del KDC. Esta llave temporal es válida solamente durante el tiempo del inicio de sesión.

Cuando un usuario, inicia sesión, el cliente solicita un ticket para el KDC de igual manera lo haría para cualquier otro servicio. El KDC responde creando una llave de inicio de sesión y un ticket especial para el servidor, llamado ticket de servicio completo de generación de tickets. Una copia de la llave de inicio de sesión es incrustada en el ticket y el ticket es encriptado utilizando la llave maestra del KDC. Otra copia de la llave de inicio de sesión es encriptada con la llave maestra del usuario derivada de la contraseña. Ambos, el ticket y la llave de sesión encriptada son enviadas al cliente. De esta manera, el cliente ya no necesita la llave derivada de la contraseña del usuario, porque en su lugar, utilizará la llave de inicio de sesión para des encriptar sus propias copias de cualquier llave de sesión que obtenga del KDC. El cliente almacena la llave de inicio de sesión en su caché de ticket junto con el ticket de servicio completo de generación de tickets.

El ticket de servicio completo de generación de tickets se le conoce como TGT por sus siglas en inglés (Ticket-Granting Ticket). Cuando un cliente le solicita al KDC un ticket para un servidor,

Capítulo 2: Marco Teórico

presenta sus credenciales en la forma de un mensaje autenticador y un ticket, en este caso un TGT, de igual manera, a como presentaría sus credenciales para cualquier otro servicio. El servicio de generación de ticket abre un TGT con su llave maestra, extrae la llave de inicio de sesión para este cliente y utiliza la llave de inicio de sesión para encriptar una copia de la llave de sesión del cliente para el servidor.

Modelo de Kerberos

El modelo de Kerberos está constituido por los siguientes componentes: Centro de Distribución de Llaves, Base de datos de cuentas y Políticas de Kerberos.

Centro de Distribución de Llaves

El centro de distribución de llaves o KDC por sus siglas en inglés es implementado como un servicio del dominio. Utiliza el directorio activo como su base de datos de cuentas y el catálogo global para dirigir referencias al KDC en otros dominios. En algunas implementaciones del protocolo Kerberos, el KDC es un proceso único que provee de dos servicios.

Servicio de Autenticación (AS por sus siglas en inglés)

Este servicio emite tickets de generación de tickets (TGT) para las conexiones al servicio de generación de tickets en su propio dominio o en cualquier dominio de confianza. Antes de que un cliente pueda preguntar por un ticket a otro equipo, debe solicitar un TGT del servicio de autenticación en la cuenta de dominio del cliente.

El servicio de autenticación regresa un TGT para el servicio de generación de ticket en el dominio destino del equipo. El TGT puede ser reutilizado hasta que expire, pero el primer acceso a cualquier servicio de generación de ticket del dominio requiere un contacto al servicio de autenticación de la cuenta de dominio del cliente.

Servicio de Generación de Ticket (TGS por sus siglas en inglés)

Este servicio emite tickets para conexiones de computadoras en su propio dominio. Cuando los clientes desean acceder a un equipo, ellos contactan el TGS en el dominio del equipo remoto, presentan un TGT y piden un ticket para la computadora en cuestión. El ticket puede ser reutilizado hasta que expira, pero el primer acceso a cualquier computadora siempre requiere un contacto al TGS de la cuenta de dominio del equipo remoto.

El KDC para un dominio está ubicado en el controlador de dominio, al igual que el directorio activo para ese dominio. Ambos servicios son iniciados automáticamente por la autoridad de seguridad local (LSA) del controlador de dominio y se ejecutan como parte del proceso de LSA.

Ninguno de los servicios puede ser detenido. Si el servicio de KDC no está disponible para los clientes de la red, tampoco, el Directorio Activo estará disponible y por consiguiente el controlador de dominio ya no estaría controlando el dominio. Los sistemas aseguran la disponibilidad de estos

Capítulo 2: Marco Teórico

y otros servicios de dominio al permitir que cada dominio tenga varios controladores de dominios, todos pares. Cualquier controlador de dominio puede aceptar peticiones de autenticación y peticiones de tickets dirigidos al KDC del dominio.

El token de seguridad utilizado por el KDC en cualquier dominio es "krbtgt" a como se especifica en el documento RFC 4120. Una cuenta para este token de seguridad es creada automáticamente cuando un nuevo dominio es creado. La cuenta no puede ser borrada o renombrada. Una contraseña aleatoria es asignada automáticamente por el sistema durante la creación del dominio. La contraseña para la cuenta del KDC es utilizada para derivar la llave criptográfica para encriptar y des encriptar los TGT que se emiten. La contraseña para una cuenta de confianza de dominio es utilizada para derivar una llave entre-dominio para encriptar tickets de referencia.

Todas las instancias del KDC dentro del dominio utilizan la cuenta de dominio "krbtgt" asociada con el token de seguridad. Los clientes envían sus mensajes al KDC incluyendo en éstos los nombres del token de seguridad "krbtgt" y el nombre del dominio. Ambos ítems de información también son utilizados en los tickets para identificar la autoridad emisora.

Base de datos de Cuentas

El Directorio Activo provee la base de datos de cuentas que el KDC utiliza para obtener la información acerca de los tokens de seguridad en el dominio. Cada token es representado por un objeto de cuenta en el directorio.

La llave de cifrado utilizada en la comunicación con un usuario, equipo o servicio es almacenado como un atributo en el objeto de la cuenta para el token de seguridad.

Solamente los controladores de dominio son servidores de Directorio Activo. Cada controlador de dominio mantiene una copia escribible del directorio, de esta manera, las cuentas pueden ser creadas, las contraseñas restablecidas y la pertenencia de grupo puede ser modificada en cualquier controlador de dominio. Los cambios realizados a una réplica del directorio son automáticamente propagados a todas las demás réplicas.

Windows replica la información almacenada en el directorio activo utilizando un protocolo propietario de replicación de múltiples masters que utiliza una conexión segura de RPC (remote procedure call) entre los miembros de la replicación.

La conexión utiliza el protocolo de autenticación Kerberos para proveer una autenticación y cifrado mutuo. El almacenamiento físico de los datos de las cuentas es administrado por el agente del sistema de directorio; un proceso protegido e integrado con la autoridad de seguridad local (LSA) en el controlador de dominio. Los clientes del servicio de directorio nunca pueden acceder directamente a los datos de las cuentas. Cualquier cliente, que desee acceder a la información del directorio debe conectarse con el agente del sistema de directorio y luego podrá buscar, leer y escribir objetos al directorio y sus atributos. Las peticiones para acceder un objeto o atributo en el

Capítulo 2: Marco Teórico

directorio son sujetas a la validación por parte de los mecanismos de acceso de control de Windows. Similarmente, como los archivos y carpetas en el sistema de archivo NTFS, los objetos en el Directorio Activo son protegidos por listas de control de acceso (ACL por sus siglas en inglés) las cuales especifican quien puede acceder a un objeto y de qué manera.

A diferencia de los archivos y carpetas, los objetos del directorio activo, tienen un ACL asociado a cada uno de sus atributos. De esta manera, los atributos con información sensible de una cuenta pueden protegerse de una manera más restrictiva que aquellos atributos con menor sensibilidad. La información más sensible de una cuenta, por supuesto, es su contraseña. A pesar, de que lo que se almacena en el atributo de la contraseña de una cuenta es realmente una llave encriptada derivada de la contraseña real, éste dato es tan útil para un intruso como la contraseña misma.

Por lo tanto, el acceso al atributo de contraseña de un objeto de cuenta solamente se le concede al titular de la cuenta, nunca a nadie más, ni siquiera a un administrador. Solamente, los procesos que se ejecutan bajo el contexto del SLA, se les permite leer o cambiar la información de la contraseña. Para evitar el acceso a cualquier persona a través de algún respaldo del controlador de dominio, al momento de hacerlo, éste es encriptado una segunda vez utilizando una llave del sistema. Esta llave de cifrado puede ser restaurada en un medio removible para que se pueda resguardar separadamente. Los administradores tienen la opción de seleccionar la manera de almacenar esta llave y cuál de los diferentes algoritmos de seguridad son utilizados para encriptar el atributo de la contraseña.

Políticas de Kerberos

La política de los tickets de Kerberos es definida a nivel del dominio e implementada por el Centro de Distribución de Llaves (KDC por sus siglas en inglés). La política de Kerberos es almacenada en el Directorio Activo como un sub conjunto de los atributos de la política de seguridad del dominio. De forma predeterminada, las opciones de la política pueden ser establecidas solamente por los miembros del grupo de administradores del dominio.

La política de Dominio incluye las opciones de:

- Soportan los tickets pos fechado.
- Soportan las delegaciones delimitadas.
- Soportan tickets que pueden ser reenviados.
- Soportan tickets renovables.
- Establecer la edad máxima de un ticket.
- Establecer la edad máxima de una renovación.
- Establecer la edad máxima de un ticket proxy.
- Forzar el cierre de sesión cuando los tickets expiren.

Con la delegación delimitada, a una computadora solamente se le permite poder reenviar las credenciales a solamente un lista de servicios específicos. Estos servicios deberán residir o

Capítulo 2: Marco Teórico

ejecutarse en el mismo dominio al que pertenece la computadora que está reenviando las credenciales. Bajo la delegación delimitada, los tickets ya no son enviados por los clientes al servidor. El servidor crea tickets de servicio para reenviar las veces que sean necesarias con la información utilizada al autenticar al cliente. [53]

Web Services

Son aplicaciones clientes y servidores que se comunican a través del protocolo HTTP¹² del Internet. Según la descripción del World Wide Web Consortium (W3C), los servicios web proveen de un medio estándar de interoperabilidad entre aplicaciones que se ejecutan en una variedad de plataformas y frameworks.

Los servicios web son caracterizados por su gran interoperabilidad y extensibilidad, al igual que, sus descripciones procesables por máquinas, gracias al uso de XML. Los servicios web pueden ser combinados de manera sencilla para realizar operaciones complejas. Programas proveyendo de servicios simples pueden interactuar entre sí para entregar servicios de valor agregado sofisticados. [54] [55] [56]

Tipos de Web Services

A nivel conceptual, un servicio web es un componente de software que se provee a través de un punto final accesible por la red. El consumidor y proveedor del servicio utilizan mensajes para intercambiar la petición de invocación y respuesta de la información a manera de documentos que pueden hacer pocas asunciones acerca de las capacidades técnicas del receptor. Los servicios web se pueden categorizar a nivel técnico en dos: Servicios Web completos y Servicios Web "RESTful"¹³.

Servicios Web Completos

Los servicios web completos utilizan mensajes XML que se adhieren al estándar "Simple Object Access Protocol" (SOAP por sus siglas en inglés); un lenguaje XML que define la arquitectura y formato de los mensajes. Estos sistemas, muchas veces contienen descripciones de las operaciones ofrecidas por el servicio que pueden ser leídas por las computadoras, escritas en lenguaje de descripción de servicios web (WSDL por sus siglas en inglés), un lenguaje XML utilizado para definir las interfaces sintácticamente.

El formato del mensaje SOAP y el lenguaje de definición de interfaz WSDL han adquirido una adopción amplia. Muchas herramientas de desarrollo pueden reducir la complejidad de desarrollar aplicaciones de servicios web.

Un diseño basado en SOAP debe incluir los siguientes elementos:

¹² HTTP:HyperText Transfer Protocol- Protocolo utilizado para el intercambio de información entre el cliente y servidor web.

¹³ RESTful: Representational State Transfer – Protocolo utilizado para servicios web de forma ad hoc.

Capítulo 2: Marco Teórico

- Un contrato formal debe ser establecido para describir la interfaz que ofrece el servicio web. El lenguaje WSDL puede ser utilizado para describir el detalle de dicho contrato. También, se pueden procesar mensajes SOAP sin haber publicado una definición WSDL.
- La arquitectura debe cubrir requerimientos complejos no funcionales. Muchas especificaciones de servicios web cubren estos requerimientos y establecen un vocabulario común para ellos.
- La arquitectura necesita manejar invocaciones y procesamiento asíncrono.

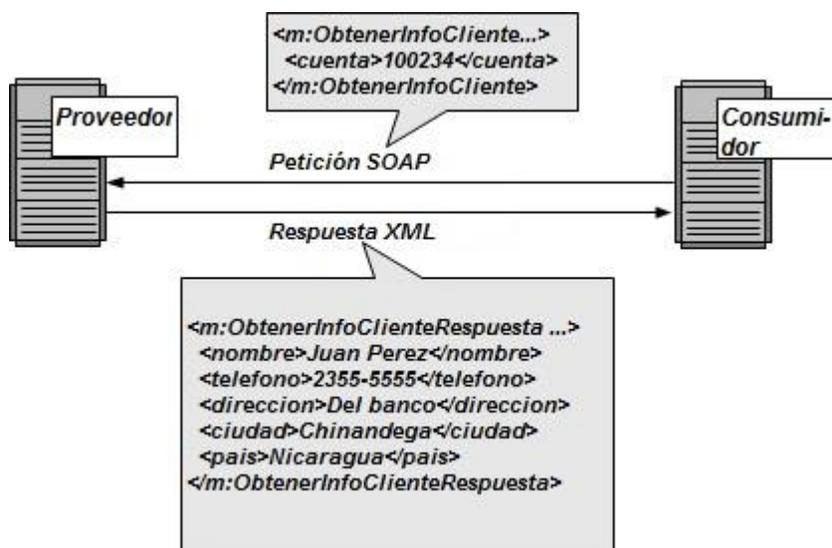


Figura 26: Mensaje SOAP

Servicios Web "RESTful"

REST está muy bien definido para escenarios básicos e integración al vuelo. Los servicios web RESTful, a menudo se integran mucho mejor con HTTP que los servicios web basados en SOAP. Éstos no requieren de mensajes XML o definiciones WSDL.

Debido a que los servicios web RESTful utilizan estándares bien conocidos de la W3C tales como, HTTP, XML, URI, MIME y tiene una infraestructura más liviana que permite crear servicios con un conjunto mínimo de herramientas, los servicios web RESTful son más baratos y por lo tanto tienen una barrera de adopción mucho más baja.

Un diseño de servicio web RESTful puede ser apropiado cuando se cumplen las siguientes condiciones:

- Los servicios web son completamente sin estado.

Capítulo 2: Marco Teórico

- Una infraestructura caché puede ser utilizada para mejorar el rendimiento.
- Tanto el consumidor como el productor del servicio tienen un entendimiento mutuo sobre el contexto y contenido que se pasa entre ambos.
- El ancho de banda es particularmente importante y necesita estar limitado.
- Cuando el despliegue del servicio web en sitios web existentes puede ser habilitado fácilmente.

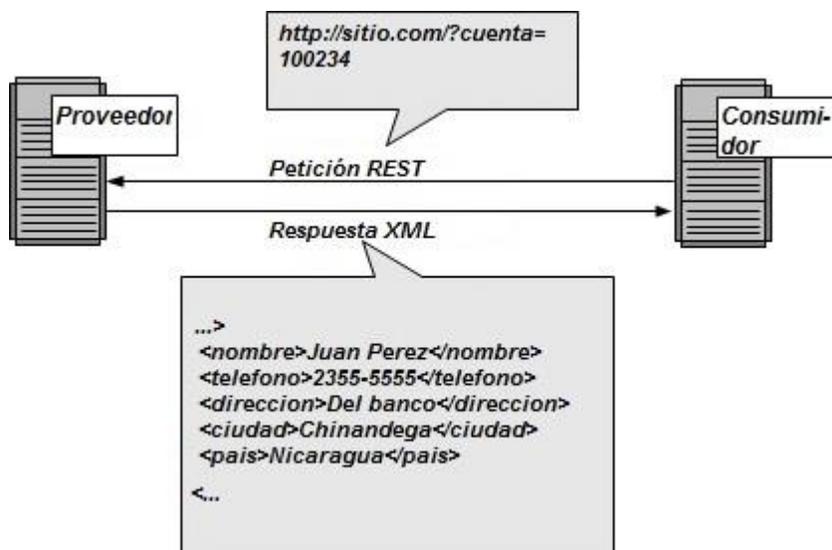


Figura 27: Mensaje REST

Servidor de bases de datos

El término base de datos aparece por primera vez alrededor de los años 60 y se utilizó para expresar un conjunto de datos relacionados entre sí, que están estructurados de tal manera que se pueden acceder a ellos automáticamente e independientemente del programa o sistema que gestiona los datos. Esta independencia está ligada a la posibilidad de modificar la estructura de los mismos sin necesidad de cambiar los programas que los gestionan y evitar problemas con la actualización de datos existentes. [57]

Estructura de las bases de datos

Los sistemas de gestión de bases de datos (SGBD o DBMS) organizan y estructuran los datos de tal manera que puedan ser recuperados y gestionados por usuarios y aplicaciones. Las estructuras de los datos y las técnicas de accesos provistas por los SGBD se le conocen como modelo de datos.

Sistema de gestión de archivos

Este modelo generalmente provisto por la funcionalidad del sistema operativo no tenía conocimiento alguno sobre el contenido de los archivos en el sistema. Esta funcionalidad se

Capítulo 2: Marco Teórico

encontraba solamente en las aplicaciones que gestionaban los archivos. Es decir, para este sistema era indistinto el contenido de archivo de texto que el de un archivo de nóminas.

Bases de datos jerárquicas

Se basa en el establecimiento de jerarquías o niveles entre los diferentes campos de los registros, basándose en el criterio de que los campos de mayor jerarquía sean los más genéricos y tienen una estructura de árbol donde los nodos del mismo nivel corresponden a los campos y cada rama representa a un registro.

Organización Jerárquica

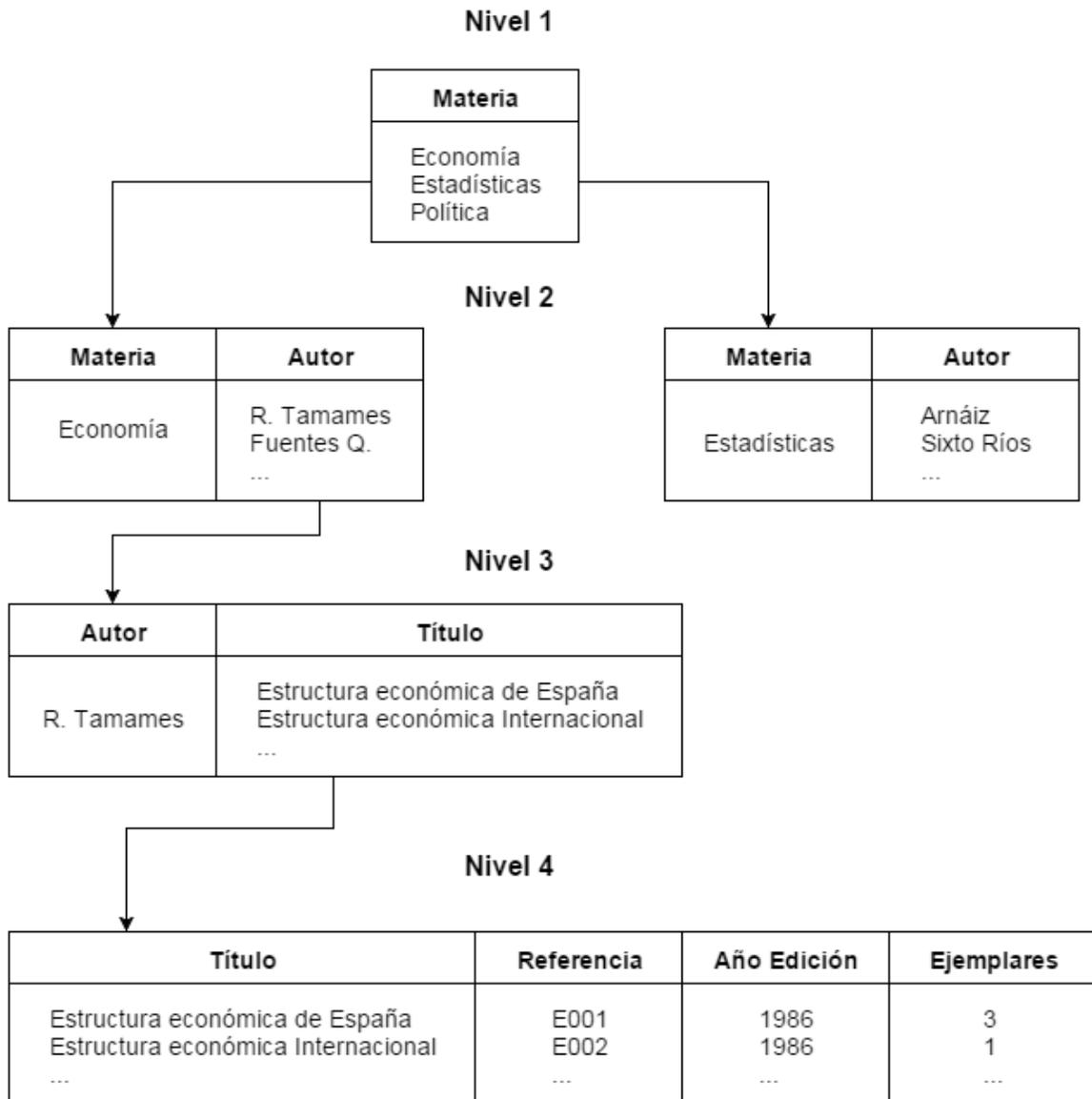


Figura 28: Organización Jerárquica

Capítulo 2: Marco Teórico

Bases de datos en red

Es una estructura de grafo donde existe una o más conexiones entre los nodos de diversos niveles. Esto permite recorrer la información por diferentes vías o rutas sin tener la necesidad de recurrir por la raíz cada vez que se desea obtener información. La flexibilidad que este tipo de estructura provee hace que desaparezca el concepto de “jerarquía” siendo esto la razón de mejora y adaptación de este tipo de base de datos sobre las bases de datos jerárquicas.

Este modelo de datos extiende el modelo jerárquico permitiendo a cualquier registro poseer múltiples relaciones padre/hijo. Estas relaciones son llamadas “conjuntos” en el modelo de red. [57] El inconveniente de este modelo es que para almacenar un registro se requería de una cantidad mayor de memoria; ya que para, poder almacenar las relaciones múltiples se debía mantener punteros de esta información hacia el siguiente nodo de la relación.

A continuación se detallan algunas ventajas y desventajas entre los dos modelos.

Tabla 5: Ventajas y desventajas de los Modelos Jerárquico y Red

	Descripción
Ventajas	Flexibilidad: Permitía modelar estructuras y relaciones muy complejas. Normalización: Adoptado el estándar CODASYL ¹⁴ que popularizó el uso de ellas. Rendimiento: Se mejoró el rendimiento en comparación al modelo jerárquico.
Desventajas	Muy rígidas Estructura y relaciones debían ser construidas de antemano Cambios costos; ya que debían reconstruir las bases de datos completa.

¹⁴ CODASYL: (Conference/Committee on Data Systems Languages). Comité formado en los años 60's para definir estándares sobre sistemas de datos.

Capítulo 2: Marco Teórico

Organización de Red

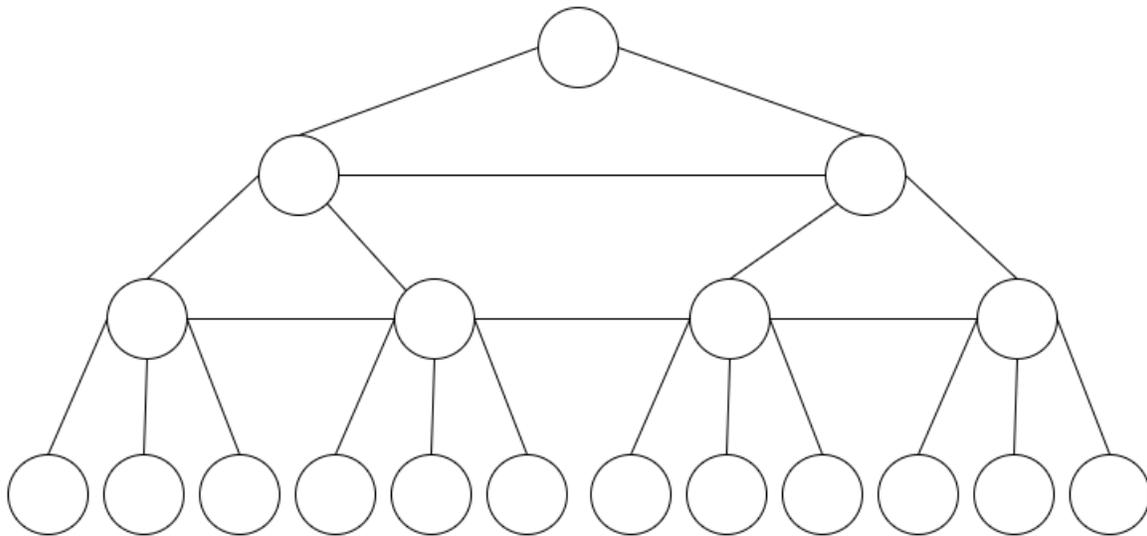


Figura 29: Organización de Red

Bases de datos relacionales

En Junio de 1970 el matemático inglés Dr. Edgar F. Codd publicó un ensayo titulado “A Relational Model of Data for Large Shared Data Banks”¹⁵ donde propone un nuevo modelo de bases de datos llamado modelo relacional. En su modelo el esquema de la base de datos u organización lógica de los datos está separado del medio de almacenaje físico de la información. Éste se convirtió en el modelo estándar utilizado por las bases de datos. [58] [59]

En los años subsecuentes aparecieron dos prototipos basados en el modelo relacional. Estos modelos son “Ingress” desarrollado por UBC y “System R” desarrollado por IBM. Ingress utilizaba un lenguaje de consultas llamado “QUEL” y esto dio lugar al desarrollo de sistemas tales como Ingress, MS SQL Server, Sybase, etc.; por otro lado, System R utilizaba un lenguaje de consultas llamado SEQUEL y dio lugar al surgimiento de sistemas tales como SQL/DBS, DB2, Oracle, etc. Es en esta época que también se popularizó el término de SGBDR¹⁶ o conocido por sus siglas en inglés como RDBMS¹⁷. [58] [60]

Lenguaje Estructurado de Consultas (SQL) Structured Query Language

SQL se convirtió en el lenguaje utilizado para consultar o interrogar un modelo relacional. Su predecesor llamado SEQUEL (Structured English Query Language) (Lenguaje de Consultas Estructurado en Inglés) fue desarrollado por IBM Corporation, Inc. para ser utilizado con el modelo relacional propuesto por el Dr. Codd. [61]

¹⁵ Un modelo relacional de datos para grandes bancos de datos compartidos.

¹⁶ SGBDR (Sistema de Gestión de Bases de Datos Relacionales)

¹⁷ RDBMS (Relational Database Management System)

Capítulo 2: Marco Teórico

Existen 3 versiones de SQL conocidos como: ANSI SQL, SQL-92 y SQL-99. La primera versión fue estandarizada por el Instituto Nacional Americano de Estándares (American National Standard Institute). Luego el lenguaje fue acogido internacionalmente por la Organización Internacional de Estándares (ISO – International Standards Organization) para el uso en bases de datos de modelo relacional. En el año de 1992 se realizó una primera revisión del lenguaje y se convirtió en la versión SQL-92. Luego, se revisó nuevamente en 1999 dando origen a la versión SQL-99; también, conocida como SQL3. [62]

Una base de datos relacional es una base de datos compuesta de objetos relacionados, principalmente compuesto de tablas. La tabla es la forma más básica de almacenar datos en una base de datos. Las tablas son conocidas como unidades lógicas y están relacionadas entre sí dentro de una base de datos. Una base de datos relaciones permite dividir o separar los datos en unidades lógicas más pequeñas y administrables que facilitan el mantenimiento y provee un mayor rendimiento óptimo de la base de datos de acuerdo al nivel de organización. Las tablas se pueden relacionar entre sí a través de llaves comunes entre ellas. [62]

Normalización

La normalización es el proceso de organizar los datos en una base de datos. Esto incluye la creación de tablas y establecer las relaciones entre estas tablas en concordancia a reglas diseñadas para proteger los datos y para hacer la base de datos más flexible al eliminar redundancia y la inconsistencia de la dependencia. Los datos redundantes crean problemas de mantenimiento así como desperdicio de espacio en disco. Si existen datos idénticos almacenados en más de un lugar y necesitan ser modificados, los datos deben ser actualizados exactamente de la misma manera en todas las ubicaciones.

Inconsistencia de dependencia es cuando un dato está almacenado en una tabla que no tiene relación alguna con la naturaleza del dato. Por ejemplo, guardar el salario de un empleado en la tabla de los clientes. La inconsistencia de dependencia puede hacer difícil el acceso a los datos ya que la ruta por la cual se busca dicho dato puede estar rota o no existir. Existen algunas reglas para la normalización de una base de datos. Estas reglas son conocidas como formas normales. Si la primer regla se aplica a una base de datos, se dice que ésta base de datos se encuentra en la primera forma normal. Si las primeras tres reglas son aplicadas, la base de datos se dice que se encuentra en la tercera forma normal. A pesar de que existen más de tres formas normales, la tercera es considerada el mayor nivel necesario para la mayoría de las aplicaciones.

A menudo, los escenarios del mundo real no permiten una aplicación perfecta de las formas normales. En general, la normalización requiere tablas adicionales; sin embargo, de no seguir con las reglas la redundancia de datos y la inconsistencia de dependencia pueden presentarse y provocar muchos problemas serios en un diseño de base de datos. [63] [64] [65]

Capítulo 2: Marco Teórico

Primera Forma Normal (1FN)

Para tener los datos de una tabla en primera forma normal se deben seguir los siguientes criterios:

- Eliminar los grupos repetidos de datos en tablas individuales.
- Crear una tabla separada para cada conjunto de datos relacionados.
- Identificar cada conjunto de datos relaciones con una llave primaria.

No utilice múltiples campos en una sola tabla para almacenar datos similares. Es mucho más difícil cambiar la estructura de columnas de una tabla que agregar un registro más a una estructura previamente definida.

Segunda Forma Normal (2FN)

Para tener los datos de una tabla en segunda forma normal se deben seguir los siguientes criterios:

- Relacionar estas tablas con una llave foránea
- Los registros no deberán depender en nada más que en la llave primaria de la tabla. (Una llave primaria puede ser compuesta).

Tercera Forma Normal (3FN)

Para tener los datos de una tabla en tercera forma normal se deben seguir los siguientes criterios:

- Eliminar campos que no dependen de la llave

Los valores en un registro que no son parte de la llave de ese registro, no pertenecen a esa tabla. En general, en cualquier momento que el contenido de un grupo de campos pueda aplicarse a más de un solo registro en la tabla, considere ubicar esos campos en una tabla separada.

Adherirse a la tercera forma normal es deseable desde un punto de vista teórico; aunque, no siempre es práctico. En teoría, la normalización se debe tratar de implementar en el diseño de una base de datos; sin embargo, muchas tablas pequeñas pueden degradar el rendimiento o bien exceder las capacidades de abrir archivos y memoria. Esto podría resultar en el diseño de una base de datos con problemas de rendimiento y acceso a la información almacenada en ella. [63]

Proceso de Normalización

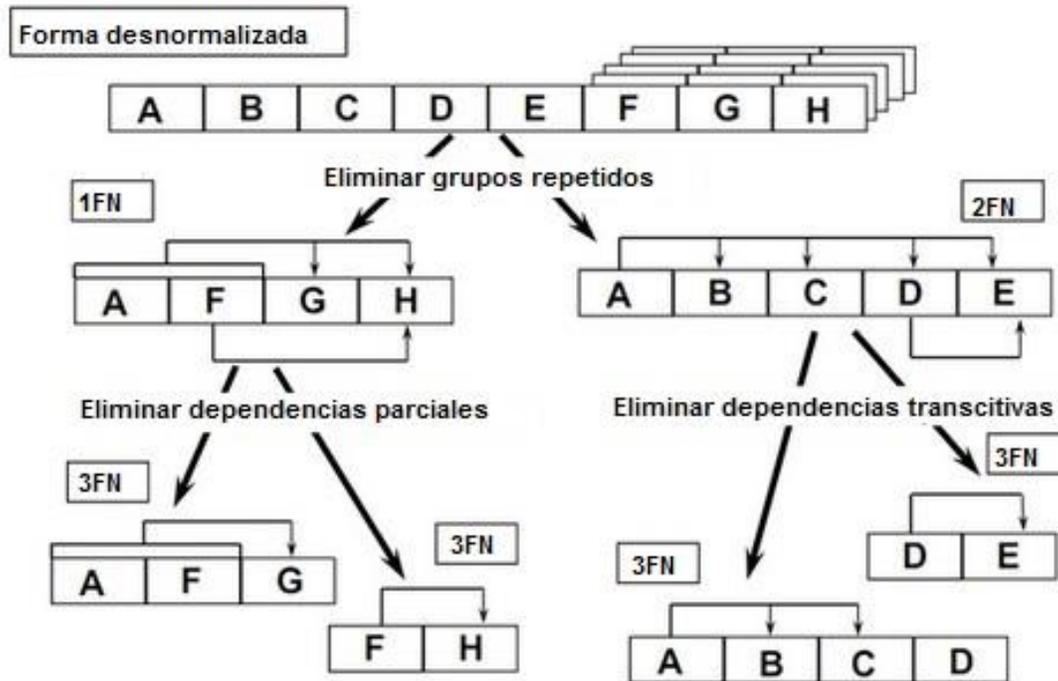


Figura 30: Proceso de Normalización

Transacciones

Un aspecto importante de las bases de datos relacionales es garantizar la calidad de los datos almacenados en sus tablas. Esta calidad se logra debido al principio "ACID" compuesto de cuatro principios: [55]

Atomicidad

La base de datos considera a todas las operaciones transaccionales como una sola unidad o átomo. Así que, cuando una base de datos procesa una transacción, ésta la completa totalmente o la aborta. En este caso, la base de datos sigue la regla del todo o nada.

Consistencia

Es la propiedad que asegura que cualquier cambio al valor en una instancia es consistente con los cambios a otros valores en la misma instancia. Una restricción de consistencia es un predicado sobre los datos que sirve como una condición previa, posterior o de transformación durante cualquier transacción.

Separación (Isolation)

Es la propiedad de separar o aislar las transacciones una de otras. Esto ayuda a garantizar la consistencia de la información. Bajo este principio, ninguna transacción es completada o abortada,

Capítulo 2: Marco Teórico

sin que la transacción anterior (si existe) haya sido completada o abortada. Esto significa que a través del tiempo se puede garantizar la consistencia de la información en la tabla. Este concepto es aplicado a la serialización de las transacciones o la ejecución de las mismas de manera secuencial o en serie.

Durabilidad

Es la habilidad de la base de datos de recuperar la última versión de los valores actualizados en la última transacción completada satisfactoriamente.



Figura 31: ACID

Microsoft SQL Server

Es un sistema de gestión de base de datos relacional (SGDBR) desarrollado por Microsoft Corporation. A como su nombre lo indica es un sistema de gestión de las base de datos. Cada servidor SQL puede alojar y/o gestionar hasta un total de 32,000 bases de datos diferentes; Aunque, esto parezca totalmente inmanejable desde el punto de vista administrativo, no existe ninguna restricción para evitarlo. [66]

Una base de datos en MS SQL Server está compuesta de al menos dos archivos físicos. El primer archivo contiene la información del esquema así como el contenido de los datos que se almacenarán en la base de datos. El segundo archivo es una bitácora de transacciones que mantiene la información de las transacciones generadas durante las operaciones de consultas para las bases de datos. El tamaño de un archivo de base de datos puede sobre pasar los 500,000 terabytes. [66]

Capítulo 2: Marco Teórico

Un servidor SQL está compuesto de las bases de datos del sistema y usuarios. Las bases de datos del sistema son cuatro: master, tempdb, msdb, model. Las bases de datos de los usuarios pueden ser de uno o más de ellas.

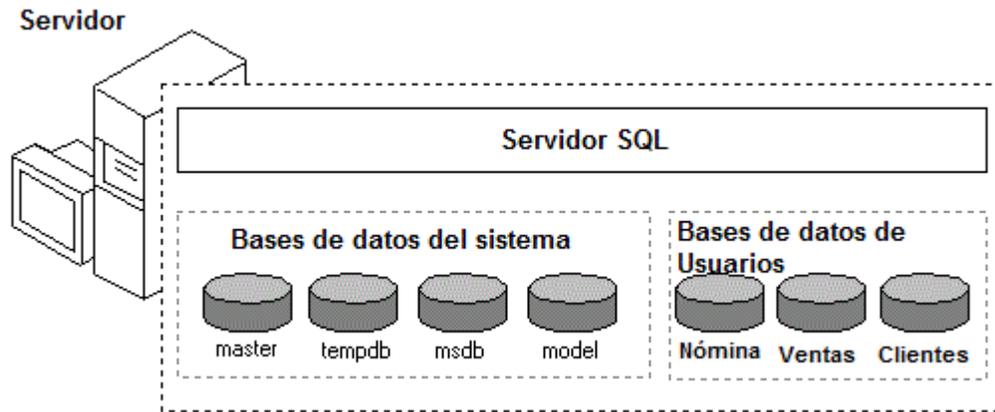


Figura 32: Servidor SQL

Capítulo 2: Marco Teórico

Arquitectura de Base de datos

Todas las bases de datos de usuario pueden contener los siguientes objetos entre otros. [67]

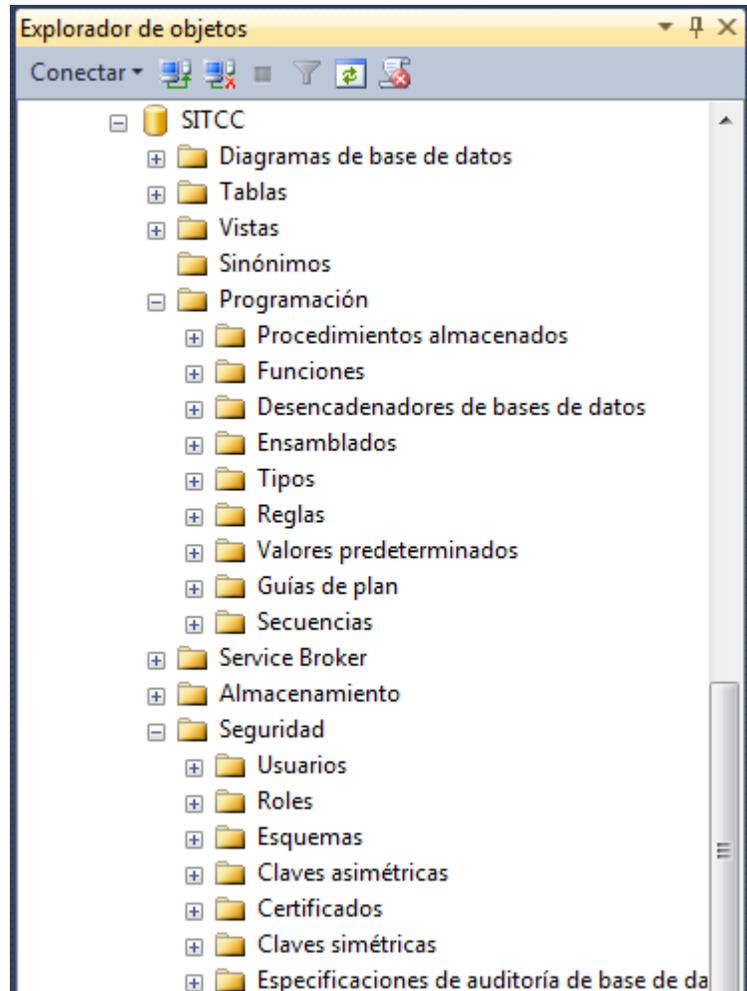


Figura 33: Componentes / Objetos de una base de datos

Tablas

Las tablas son objetos de la base de datos que contienen todos los datos en la base de datos. En las tablas, los datos son organizados de manera lógica en un formato de filas y columnas similar a una hoja de cálculo. Cada fila representa un registro único y cada columna representa un campo en ese registro. La cantidad de tablas en una base de datos está limitada solamente por el número de objetos permitidos (2, 147, 483,647) en una base de datos.

Una tabla estándar definida por el usuario puede tener hasta 1,024 columnas. El número de filas en la tabla está limitado solamente por la capacidad de almacenamiento físico del servidor. Se pueden asignar propiedades a las tablas y a cada columna en la tabla para controlar los datos que son permitidos; así como, otras propiedades. Por ejemplo, se pueden crear restricciones sobre un

Capítulo 2: Marco Teórico

campo o bien sobre una llave para que garantice las relaciones entre dos tablas. Los datos en la tabla pueden ser comprimidos ya sea por fila o por página. La compresión de datos puede permitir almacenar muchas más filas en una página. [67]

Vistas

Una vista es una tabla virtual cuyo contenido es definido por una consulta. Similar a una tabla, una vista consiste de un conjunto de datos mostrados como filas y columnas. A menos, que sea una vista indexada, las vistas no existen como un conjunto almacenado de valores de datos en la base de datos. Las filas y columnas de datos provienen de las tablas utilizadas o referenciadas en la consulta que define a la vista y son producidos dinámicamente cada vez que la vista es utilizada. Una vista actúa de forma similar a un filtro de las referenciadas en ella. La consulta que define una vista puede estar compuesta de una o más tablas o vistas de la misma base de datos o de otras.

Las consultas distribuidas también pueden ser utilizadas para definir vistas que utilizan datos de múltiples orígenes o fuentes heterogéneas. Las vistas son generalmente usadas para focalizar, simplificar y personalizar la percepción que cada usuario tiene de la base de datos. Las vistas pueden ser utilizadas como mecanismos de seguridad al permitir a los usuarios acceder a los datos a través de la vista, sin otorgar permisos al usuario directamente para acceder a las tablas que componen la vista.

Las vistas pueden ser utilizadas para proveer una interfaz compatible para emular una tabla que existía anteriormente y cuyo esquema haya cambiado. Las vistas pueden también ser utilizadas cuando copias datos hacia o desde el servidor SQL para incrementar el rendimiento. [67]

Procedimientos Almacenados

Un procedimiento almacenado es un grupo de una o más instrucciones Transact-SQL o una referencia a un método CLR del .Net Framework. Los procedimientos se asemejan a las funciones en otros lenguajes de programación, ya que estos pueden: [67]

- Aceptar parámetros de entrada y regresar valores múltiples de forma de parámetros de salida a los programas que los invocan.
- Contienen sentencias de programación que realizan operaciones en la base de datos. Esto incluye llamar otros procedimientos almacenados.
- Regresan un valor de estado al programa que lo invoca para indicar éxito o falla y la razón de la falla.

Beneficios de utilizar procedimientos almacenados.

Reduce el tráfico de red en aplicaciones cliente-servidor: Los comandos en un procedimiento son ejecutados como un solo lote de código. Esto reduce de manera significativa el tráfico entre el servidor y el cliente porque solamente el llamado a ejecutar el procedimiento es transmitido a

Capítulo 2: Marco Teórico

través de la red. Sin este nivel de encapsulación provista por el procedimiento, cada línea individual de código transaccional debería ser transmitida a través de la red entre el cliente y servidor.

Mayor seguridad: Esto se lleva a cabo al permitir que múltiples usuarios puedan realizar operaciones sobre los objetos referenciados en los procedimientos sin necesidad de otorgar el permiso directo a los usuarios sobre los mismos. El procedimiento controla que procesos y actividades son ejecutados y protege los objetos de la base de datos referenciados en ellos.

Cuando un llamado de ejecución de un procedimiento viaja sobre la red, solamente el llamado de ejecución es visible. Por lo tanto, usuarios maliciosos no pueden ver las tablas o demás objetos utilizados por el procedimiento.

Al utilizar los parámetros del procedimiento nos ayuda a protegernos contra ataques de inyección SQL. Esto se debe a que el parámetro de entrada es tratado como un valor literal y no como un código ejecutable. Esto hace más difícil para el atacante de insertar comando SQL dentro las sentencias del procedimiento y comprometer la seguridad. Los procedimientos pueden ser encriptados.

Reutilización de código: Con los procedimientos cualquier operación repetitiva es candidata a ser encapsulada en un procedimiento. Esto elimina la necesidad de reescribir el mismo código varias veces, disminuye la inconsistencia del código y permite que el código sea accesible para cualquier usuario que tenga los permisos de ejecución necesarios.

Facilidad de administración: Cuando las aplicaciones clientes mantienen las operaciones de la base de datos en la capa de datos, solamente, los procedimientos deberían ser actualizados abstrayendo a la aplicación de conocer cualquier cambio en la estructura de la base de datos, sus relaciones o procesos.

Rendimiento mejorado: De manera predeterminada, la primera vez que se ejecuta el procedimiento se compila y se crea un plan de ejecución que es reutilizado las ejecuciones subsecuentes. Ya que el procesador de consultas no debe crear un nuevo plan cada vez que se ejecuta, su tiempo de proceso se disminuye. Si existe algún cambio en la base de datos y se utiliza el mismo plan guardado, puede causar un rendimiento más lento. Sin embargo, esto se remedia al recompilar el procedimiento nuevamente.

Funciones definidas por el usuario

Al igual que las funciones de los lenguajes de programación, las funciones definidas por el usuario del servidor SQL son rutinas que aceptan parámetros, realizan una acción y regresan un resultado que puede ser un valor escalar único o bien un conjunto de datos. [67]

Los beneficios de utilizar funciones en SQL son:

Capítulo 2: Marco Teórico

- **Permiten la programación modular:** Se puede crear una función, almacenarla en la base de datos y utilizarla las veces que se necesiten desde tu programa.
- **Permiten una mayor velocidad de ejecución:** Al igual que los procedimientos almacenados las funciones disminuyen el costo de compilación almacenando los planes de ejecución y reutilizándolos durante las ejecuciones subsecuentes. Funciones de CLR proveen un mejor rendimiento sobre las funciones Transact-SQL para tareas computacionales, manipulación de cadenas y lógica de negocios.
- **Reducen el tráfico a través de la red:** Una restricción compleja de filtrado puede expresarse mejor como una función y ser utilizada como parte de la cláusula “Where” para reducir el número de filas enviadas al cliente.

Cursores

Las operaciones en una base de datos relacional actúan sobre un conjunto completo de filas. Las aplicaciones, especialmente las aplicaciones interactivas en línea no pueden trabajar efectivamente sobre un conjunto entero como una unidad. Estas aplicaciones necesitan un mecanismo para trabajar con una fila o un bloque pequeño de filas a la vez. [67]

Los cursores son una extensión a los conjuntos de datos que proporcionan estos mecanismos. Los cursores extienden el resultado del procesamiento al:

- Permitir el posicionamiento en filas específicas de un conjunto de datos.
- Recuperar una fila o bloque de filas de la posición actual del conjunto de datos.
- Soportar las modificaciones de los datos a las filas en la posición actual dentro del conjunto de datos.
- Soportar diferentes niveles de visibilidad a los cambios realizados por otros usuarios de la base datos al conjunto de datos presentado.

Desencadenadores DML¹⁸

Los desencadenadores DML es un tipo especial de procedimiento almacenado que automáticamente se ejecuta cuando un evento DML sucede en la tabla o vista definida en el desencadenador.

Los eventos DML incluyen sentencias de INSERT, UPDATE o DELETE (Insertar, actualizar o borrar). Los desencadenadores DML pueden ser utilizados para hacer cumplir las reglas de negocios y reglas de integridad de datos, consultar otras tablas e incluir sentencias complejas de Transact-SQL. El desencadenador y la sentencia que lo dispara son tratadas como una transacción simple, la cual puede ser deshecha desde el mismo desencadenador. De manera predeterminada, si un error severo se detecta toda la transacción completa automáticamente se deshace.

¹⁸ DML: (Data Manipulation Language) Lenguaje de Manipulación de Datos

Capítulo 2: Marco Teórico

Los desencadenadores DML son parecidos a las restricciones en el sentido que pueden asegurar o procurar la integridad de la entidad. En general, la integridad de la entidad debería ser asegurada al nivel más bajo permitido como los índices que son parte de las llaves primarias y restricciones de llaves únicas o bien a través de restricciones independientes. [67]

Capítulo 3: Diseño Metodológico

Capítulo 3: Diseño Metodológico

Tipo de Estudio

Se considera un tipo de estudio categorizado como no experimental y de tipo descriptivo y documental ya que permitió describir, registrar, analizar y estudiar las características fundamentales para procesar el flujo de información técnica de las granjas.

Técnicas e instrumentos de recolección de información

Se adoptaron ciertas técnicas de recolección para la recopilación de datos, tales como:

Observación Directa

Es una técnica objetiva de investigación que consiste en tener contacto directo con los elementos o caracteres en los cuales se manifiesta el fenómeno que se pretende investigar. Permite observar a las personas en la realización de su trabajo, logrando obtener datos sobre qué se está haciendo, cómo se está haciendo, quién lo hace, cuándo se lleva a cabo, cuánto tiempo dura, dónde se hace y por qué se hace.

Revisión Documental

Se realizará la revisión e investigación de los datos de ciclos o años anteriores con respecto a los procedimientos en cada etapa, con el propósito de obtener los datos y la documentación de los registros técnicos necesarios para el estudio realizado, lo cual permitirá abordar, desarrollar y alcanzar los objetivos de la investigación previamente planteados.

Entrevistas no estructuradas

Es una técnica que permite la comunicación entre el investigador y el objeto de estudio logrando facilitar la obtención de información, opiniones, conocimientos técnicos y situación actual del funcionamiento del registro de la información técnica de granja. Se entrevistó al personal técnico de granjas, jefes de granjas, coordinadores de zona y Director de granjas.

Población y muestra

La población objeto del estudio es toda la información técnica producida por las granjas durante el ciclo productivo de las mismas. La muestra se delimitó a los parámetros técnicos necesarios y sensibles para la producción exitosa de las granjas.

Procedimiento de la información

Los pasos que se realizaron en esta investigación fueron los siguientes:

1. Se diagnosticó la situación actual del flujo de información respecto al almacenamiento de los registros técnicos en las granjas.
 - a. Recorridos a las diferentes granjas para realizar las inspecciones correspondientes de las mismas.
 - b. Se realizaron entrevistas no estructuradas con el personal de granjas.

Capítulo 3: Diseño Metodológico

2. Se analizaron los fundamentos teóricos para la determinación de la metodología que permitirá la definición del registro de la información técnica de granjas.
3. Se elaboró un plan de trabajo para desarrollar un sistema de información que permitiera el registro de la información técnica de las granjas.
4. Se desarrolló un sistema de información en ambiente cliente/servidor que permitiera registrar y analizar la información técnica de granjas.

Descripción de la Metodología

RAD es una metodología formalizada en 1991 por James Martin y se “basa en la necesidad de hacer el desarrollo de software más rápido y más eficiente y a la vez creando un producto de alta calidad” [68] [69] [70] [71]. Esta metodología es parte del concepto de desarrollo de software “ágil”. Es un conjunto de conceptos que permiten agilizar el desarrollo de sistemas acortando el tiempo de desarrollo; dando lugar a la creación de nuevas metodologías tales como: SCRUM, eXtreme Programming (XP), Dynamic Systems Development Methodology (DSDM), Adaptive Software Development, Feature Driven Development, Crystal Clear Method, Rapid Application Development, Pragmatic Programming. [72]



Figura 34: Fases de la Metodología RAD

La metodología se divide en cuatro fases [73] [74]. Estas fases resultan de la compresión de las fases de la metodología tradicional de cascada [75].

Capítulo 3: Diseño Metodológico

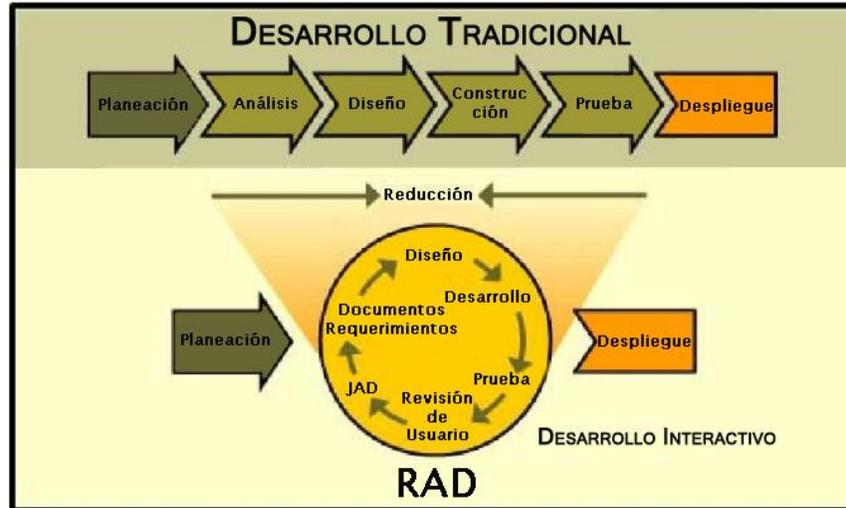


Figura 35: Metodología de Cascada vs RAD

Objetivo y/o Propósito

La metodología tiene por objetivo desarrollar sistemas con las siguientes características: **Gran Velocidad, Alta Calidad y Costos Bajos** [73].

Fase de Requerimientos

En esta fase se planifica y se determina el alcance del proyecto. Está compuesta por la formación del personal que asistirá a los JAD¹⁹. Adicionalmente, se realiza una revisión completa de las áreas involucradas en el proyecto. Se establece una definición amplia de los requerimientos del sistema basados en las funciones y/o características que tendrá o soportará el sistema. Los productos o resultados de esta fase son entre otros: los modelos de entidad relación, modelos de procesos y la definición específica del alcance del sistema [73] [75] [74].

Fase de Diseño de Usuario

En esta fase se propone el diseño de la aplicación basado en los requerimientos recolectados de la fase anterior. Se elabora un prototipo de interfaz gráfica pasando por una o más JAD. En esta fase se debe realizar un análisis detallado de las actividades del negocio relacionadas al sistema. Adicionalmente, se establecen los usuarios claves, las reuniones en talleres de trabajo, se realiza la

¹⁹ JAD (Joint Application Development) (Desarrollo de Aplicación Conjunta) Es un conjunto de reuniones y/o talleres de trabajo del grupo de trabajo de desarrolladores y usuarios donde se presentan los avances y se proponen las correcciones y/o nuevos requerimientos.

Capítulo 3: Diseño Metodológico

descomposición de las funciones del negocio, la definición de los tipos de entidades asociadas al sistema. Como resultado de esta fase se realiza la:

- Creación de diagramas de acción definiendo las interacciones entre los procesos y los datos.
- Diseño de los procedimientos del sistema.
- Elaboración de la IGU²⁰ con la propuesta preliminar del prototipo.

Una vez aprobado el prototipo pasa a la fase de construcción [73].

Fase de Construcción

En esta fase se codifica el sistema basado en el prototipo propuesto y aprobado por el JAD. El desarrollo se hace cíclico dividiendo en etapas el proyecto completo de desarrollo de esta manera se presenta en poco tiempo y en JAD la funcionalidad desarrollada. Una vez aprobado por los miembros del JAD se procede con la etapa siguiente hasta completar todas las etapas del proyecto. Es importante denotar que existe un proceso cíclico y de retroalimentación entre las fases de “Diseño de Usuario y Construcción”. Esta unión permite a la metodología a descubrir, definir y resolver problemas con el prototipo original o de sus etapas. El trabajo en conjunto con el usuario es la llave clave para el desarrollo ágil de la aplicación delimitado por los requerimientos establecidos en la fase de requerimientos. Esta delimitación permite que el proyecto se cumpla y se dé por completado; no permitiendo, el desarrollo permanente al descubrirse nuevos requerimientos en el transcurso del tiempo del proyecto en ejecución. Los resultados de esta fase en su etapa final es el sistema completo, la documentación de usuario y técnica necesaria para la implementación del mismo en un entorno de producción [73].

Fase de Reemplazo

En esta fase el sistema es implementado. La implementación conlleva a la preparación de todos los recursos necesarios para el funcionamiento del sistema. Estos recursos van desde la instalación, configuración del servidor de base de datos; así como, del servidor de aplicaciones y los equipos clientes que utilizarán el sistema. De igual manera, en esta fase se incluye la

²⁰ IGU (Interfaz Gráfica de Usuario) El método de interacción entre los humanos y las computadoras. Es la composición de los componentes del sistema, menús, formularios, iconos, botones, cajas de texto, etc.

Capítulo 3: Diseño Metodológico

capacitación pertinente de todos los usuarios involucrados en el sistema. Tanto los usuarios técnicos; aquellos responsable del o de los servidores, como del usuario final [73].

Desventajas

Debido a la naturaleza del concepto se pueden generar muchos contratiempos que se convierten en desventajas de la metodología. La falta de aplicación de un método científico estricto provoca no tener control sobre las etapas del desarrollo. Puede existir discrepancia entre los requerimientos y los resultados obtenidos entre las iteraciones de la fase de diseño y construcción. Los desarrolladores poco experimentados se sobrecargan de trabajo debido al constante cambio de los requerimientos no observados en la fase inicial; llevando al proyecto, a un ciclo infinito de corrección y desarrollo [73] [76].

Etapas del proyecto

Las etapas en las que se desarrolló el proyecto están definidas en el **Anexo 1: Cronograma de actividades**

Flujo de proceso de la recolección de la información técnica y su frecuencia

En esta etapa se realizó un primer contacto con los posibles candidatos a ser miembros de las JAD. Adicionalmente, se obtuvo el flujo de proceso de las lecturas de parámetros actual.

Capítulo 3: Diseño Metodológico

Flujo de Proceso Lectura de parámetros

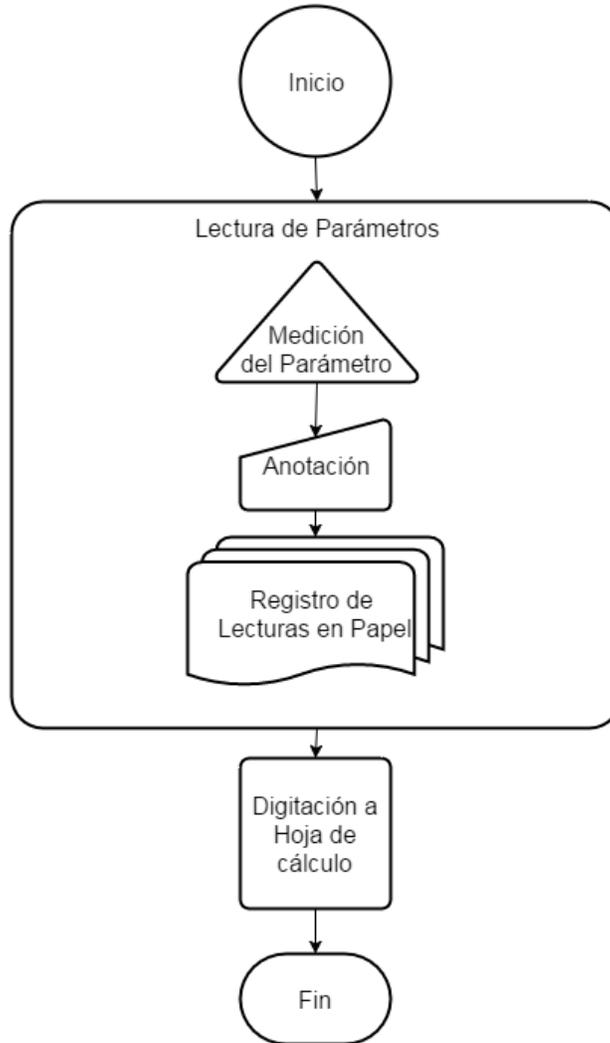


Figura 36: Flujo de proceso de lecturas de parámetros

Análisis de requerimientos

En esta etapa se definió la frecuencia de las reuniones JAD. Se establecieron los miembros de las mismas y se definió el papel que tendría cada uno de los miembros.

Las reuniones fueron implementadas con cada uno de los jefes de granjas. La frecuencia de estas reuniones era de una vez por semana. En ellas, se definieron los requerimientos comunes a cada granja. Dando como resultado, la unificación de requerimientos estándares que regirían el

Capítulo 3: Diseño Metodológico

desarrollo del sistema y que fuese aplicable a todas las granjas de la empresa. Estos requerimientos se muestran en una recopilación en el **Anexo 2: Requerimientos del sistema**

Diagrama entidad relación

En esta etapa se definió el diagrama entidad relación que utilizaría el sistema para almacenar la información de las lecturas de los parámetros que se deseaban medir. Este diagrama es presentado en el **Diagrama entidad relación**

Diseño arquitectónico

En esta etapa se definió el patrón de diseño MDI para la aplicación.

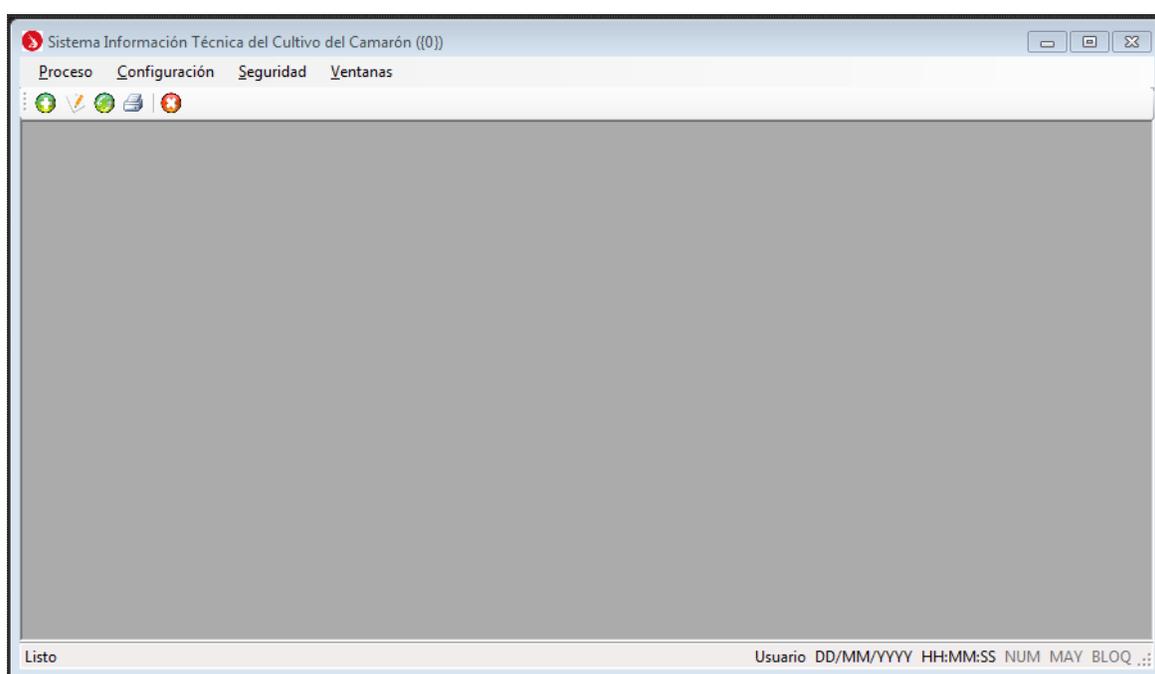


Figura 37: Diseño arquitectónico

Desarrollo del sistema

En esta etapa se llevó a cabo el desarrollo del sistema. Se utilizó el patrón de diseño MDI para una aplicación de documentos múltiples. También, se utilizó el paradigma de desarrollo de Programación Orientada a Objeto y la arquitectura cliente – servidor.

El detalle del desarrollo se muestra en el **Anexo 4: Desarrollo del sistema**

Implementación del sistema

En esta etapa se procedió al despliegue del sistema en cada uno de los equipos clientes de los usuarios que utilizarán el sistema.

Capítulo 3: Diseño Metodológico

También, se procedió con el despliegue de la base de datos en el servidor SQL.

Materiales utilizados

- Libreta de apuntes.
- Lapiceros.

Software

Para el desarrollo del sistema se utilizaron el siguiente software:

Tabla 6: Software utilizado

Nombre	Tipo
Microsoft® Visual Studio® Express for Desktop 2013	Cliente
NClass® 2.04	Cliente
StartUML® 2.5.0	Cliente
Microsoft® SQL Server® Standard 2012	Servidor

Hardware

Para el desarrollo del sistema se utilizaron el siguiente hardware:

Tabla 7: Hardware utilizado

Computadora de Escritorio	
Marca	Dell
Modelo	Optiplex 990
Arquitectura	64 bits
Memoria RAM	8 Gb
Procesador	Intel® Core i5-2400 3.10 GHz
Disco Duro	500 Gb
Sistema Operativo	Microsoft® Windows® 7 Professional
Idioma	Español

Capítulo 4: Resultados del proyecto

Capítulo 4: Resultados del proyecto

Funcionamiento del sistema

La implementación del sistema se realizó utilizando la arquitectura cliente –servidor utilizando la infraestructura de red establecida en la empresa. A partir de este punto, se describen los resultados obtenidos como consecuencia del Anexo 4: Desarrollo del sistema.

Consideración General

Con el permiso requerido el usuario podrá imprimir los datos registrados. Esta operación es totalmente contextual según el patrón de diseño de aplicaciones MDI.

Seguridad

Esta opción agrupa todos aquellos aspectos relacionados con la seguridad para garantizar los permisos de acceso indicado por cada persona que utilizará el sistema. A continuación se detalla cada uno de ellos:

Roles

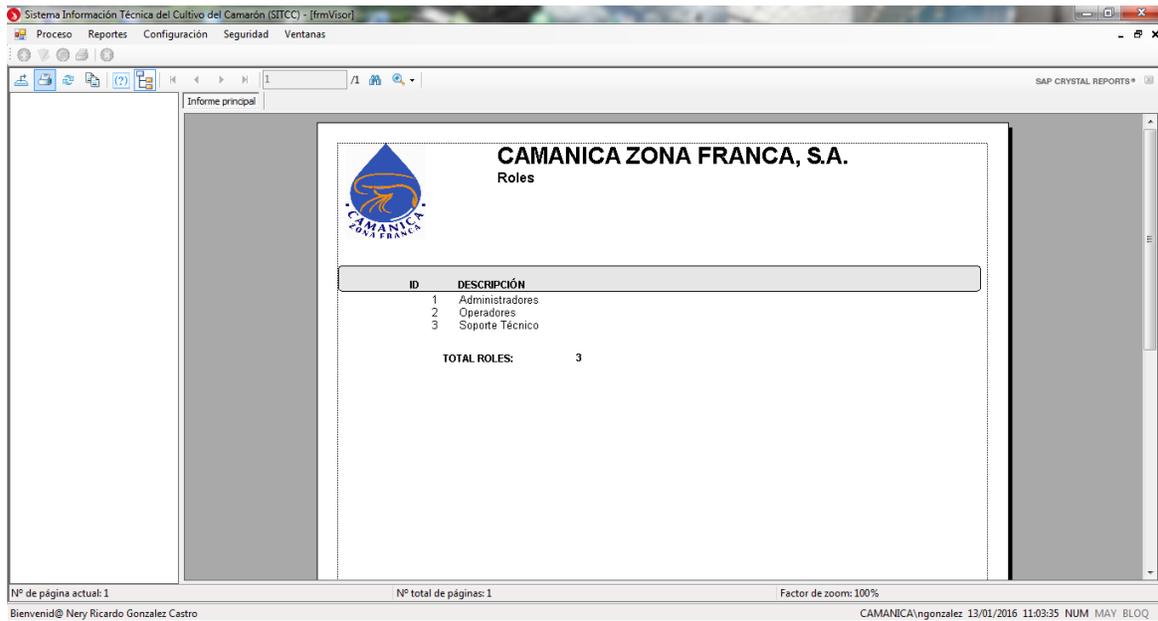
Los roles son utilizados por el administrador del sistema para permitir el acceso y los permisos necesarios para que los usuarios puedan trabajar en el sistema. Adicionalmente, se utilizan para crear una asociación o relación con las opciones del sistema y los permisos de operación que se definen para los diferentes usuarios.

Id	Descripción	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por	Actualizado En	Borrado El	Borrado Por
1	Administradores	Activo	04/06/2015 13:14:52	CAMANICA\vego...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	
2	Operadores	Activo	04/06/2015 13:14:52	CAMANICA\vego...	GRITDPWS02	17/08/2015 19:09:00	CAMANICA\vego...	GRITDPWS02	01/01/1900 00:00:00	
3	Soporte Técnico	Activo	04/06/2015 13:14:52	CAMANICA\vego...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	

At the bottom of the window, there is a status bar with the text: 'Bienvenid@ Nery Ricardo Gonzalez Castro' on the left and 'CAMANICA\ngonzalez: 13/01/2016 10:44:30 NUM MAY BLOQ' on the right.

Figura 38: Roles del Sistema

Capítulo 4: Resultados del proyecto



Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frm/visor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS*

Informe principal

CAMANICA ZONA FRANCA, S.A.
Roles

ID	DESCRIPCIÓN
1	Administradores
2	Operadores
3	Soporte Técnico

TOTAL ROLES: 3

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

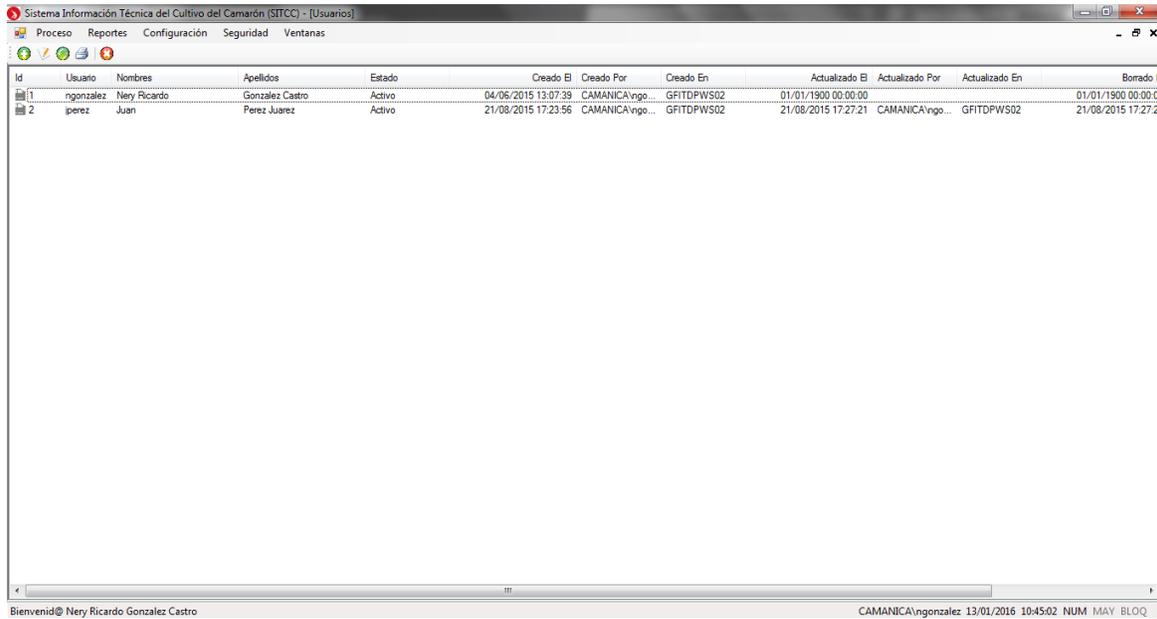
Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez: 13/01/2016 11:03:35 NUM MAY BLOQ

Figura 39: Impresión de Roles

Capítulo 4: Resultados del proyecto

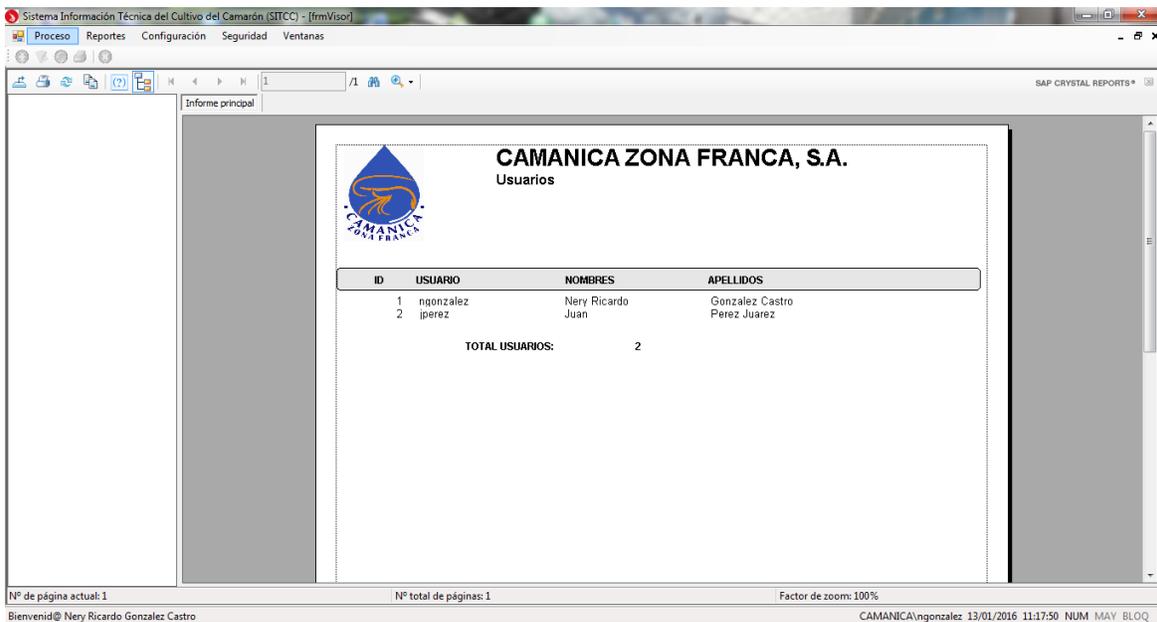
Usuarios

Los usuarios son configurados en el sistema para permitir el acceso al mismo. Los usuarios que se ingresan deben de pertenecer al Directorio Activo del dominio de la empresa. El sistema utiliza la seguridad integrada de Windows y se soporta por la misma para garantizar la seguridad de sus usuarios.



Id	Usuario	Nombres	Apellidos	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por	Actualizado En	Borrado El
1	ngonzalez	Nery Ricardo	Gonzalez Castro	Activo	04/06/2015 13:07:39	CAMANICA\ngonzalez	GFITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00
2	jperez	Juan	Perez Juarez	Activo	21/08/2015 17:23:56	CAMANICA\ngonzalez	GFITDPWS02	21/08/2015 17:27:21	CAMANICA\ngonzalez	GFITDPWS02	21/08/2015 17:27:21

Figura 40: Usuarios del sistema



ID	USUARIO	NOMBRES	APELLIDOS
1	ngonzalez	Nery Ricardo	Gonzalez Castro
2	jperez	Juan	Perez Juarez
TOTAL USUARIOS:		2	

Figura 41: Impresión de Usuarios

Capítulo 4: Resultados del proyecto

Usuarios por Roles

Los usuarios por roles definen la relación que tendrá cada usuario con el rol asignado por el administrador del sistema. Esto combinado con las opciones del sistema por rol, permite lograr la flexibilidad de controlar cual usuario tiene permiso a qué para cada opción del sistema disponible.

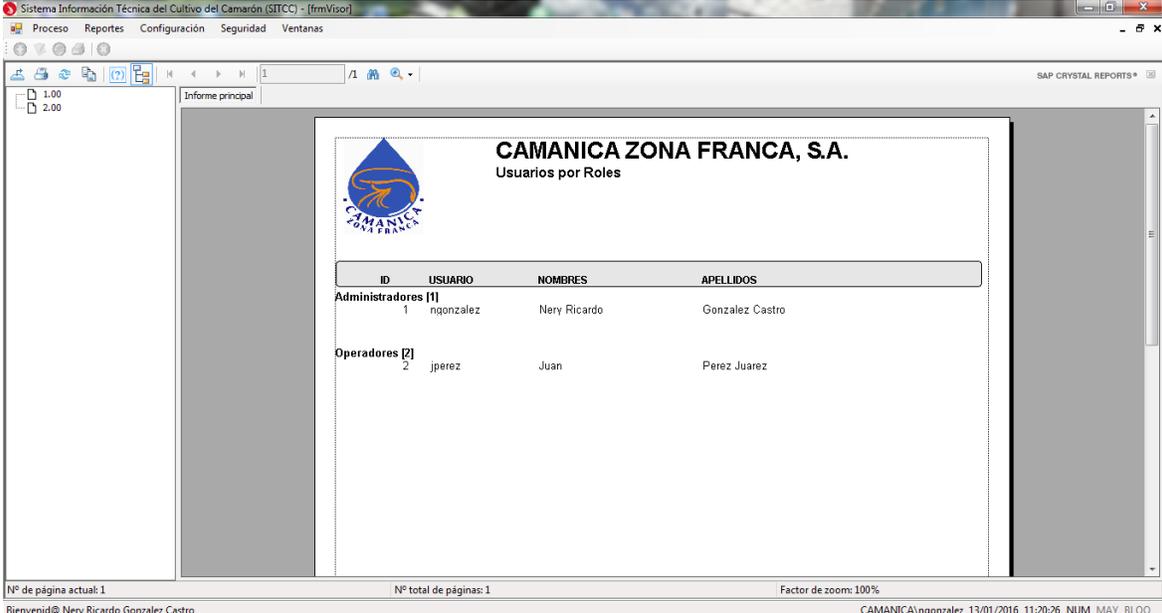
Por ejemplo, en una configuración de opciones del sistema por rol se puede establecer que un rol determinado puede leer, crear y actualizar granjas (opción del sistema). Luego, cualquier usuario asociado a ese rol, tendrá permisos de lectura, creación y actualización de datos de granjas; sin embargo, los permisos de impresión y borrados no están permitidos para esta configuración. Por lo tanto, se puede decir que el usuario no tendrá permisos a estas operaciones sobre la opción del sistema, en éste ejemplo granja.

Esta flexibilidad también infiere que la seguridad es asociativa y acumulativa. Es decir, el usuario tendrá los permisos de operación sobre cualquier opción del sistema siempre y cuando él esté asociado a los roles que tienen asociaciones directas con las opciones del sistema.

Id	Rol	Usuario	Apellidos	Nombres	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por
1	Administradores	ngonzalez	Gonzalez Castro	Nery Ricardo	Activo	04/06/2015 13:15:44	CAMANICA\ng...	GFITDPWS02	01/01/1900 00:00:00	
2	Operadores	perez	Perez Juarez	Juan	Activo	24/08/2015 09:20:55	CAMANICA\ng...	GFITDPWS02	24/08/2015 09:21:03	CAMANICA\ng...

Figura 42: Usuarios por Roles

Capítulo 4: Resultados del proyecto



Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frmVisor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS*

Informe principal

CAMANICA ZONA FRANCA, S.A.
Usuarios por Roles

ID	USUARIO	NOMBRES	APELLIDOS
Administradores [1]			
1	ngonzalez	Nery Ricardo	Gonzalez Castro
Operadores [2]			
2	jperez	Juan	Perez Juarez

NP de página actual: 1 N° total de páginas: 1 Factor de zoom: 100%

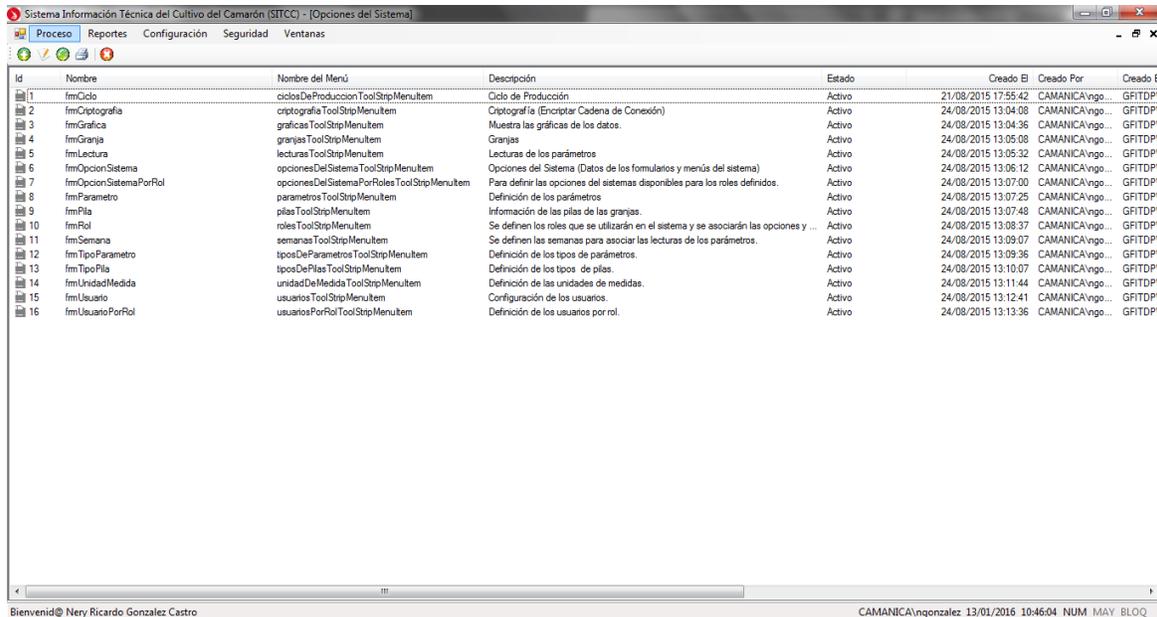
Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA/ngonzalez: 13/01/2016 11:20:26 NUM MAY BLOQ

Figura 43: Impresión de Usuarios por Roles

Capítulo 4: Resultados del proyecto

Opciones del sistema

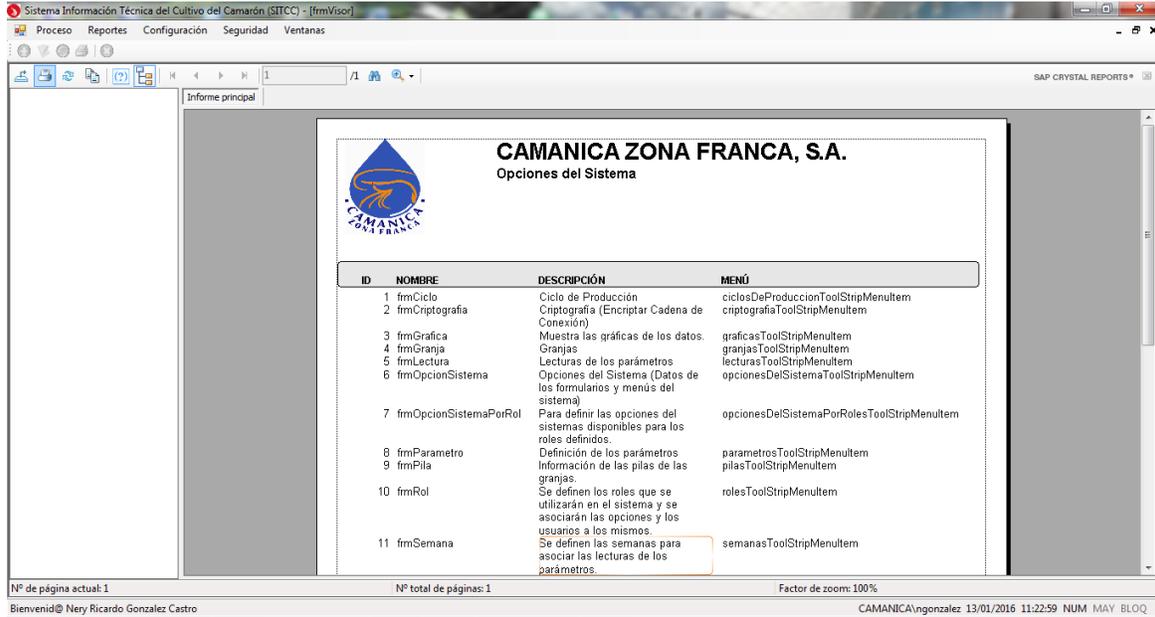
Las opciones del sistema son todas aquellas funciones disponibles a través del sistema. Entiéndase por opción, a todos los objetos o formularios que representan las diferentes entidades del sistema. Cada objeto tiene predefinida 5 funciones operativas para la gestión de los registros. Estas opciones están predefinidas ya que ellas representan los diferentes objetos físicos desarrollados en el sistema.



Id	Nombre	Nombre del Menú	Descripción	Estado	Creado El	Creado Por	Creado E
1	fmCiclo	ciclosDeProduccionToolStripMenuitem	Ciclo de Producción	Activo	21/08/2015 17:55:42	CAMANICA'ngonzalez	GFITDPI
2	fmCriptografia	criptografiaToolStripMenuitem	Criptografía (Encriptar Cadena de Conexión)	Activo	24/08/2015 13:04:08	CAMANICA'ngonzalez	GFITDPI
3	fmGrafica	graficasToolStripMenuitem	Muestra las gráficas de los datos.	Activo	24/08/2015 13:04:36	CAMANICA'ngonzalez	GFITDPI
4	fmGranja	granjasToolStripMenuitem	Granjas	Activo	24/08/2015 13:05:08	CAMANICA'ngonzalez	GFITDPI
5	fmLectura	lecturasToolStripMenuitem	Lecturas de los parámetros	Activo	24/08/2015 13:05:32	CAMANICA'ngonzalez	GFITDPI
6	fmOpcionSistema	opcionesDelSistemaToolStripMenuitem	Opciones del Sistema (Datos de los formularios y menús del sistema)	Activo	24/08/2015 13:06:12	CAMANICA'ngonzalez	GFITDPI
7	fmOpcionSistemaPorRol	opcionesDelSistemaPorRolesToolStripMenuitem	Para definir las opciones del sistemas disponibles para los roles definidos.	Activo	24/08/2015 13:07:00	CAMANICA'ngonzalez	GFITDPI
8	fmParametro	parametrosToolStripMenuitem	Definición de los parámetros	Activo	24/08/2015 13:07:25	CAMANICA'ngonzalez	GFITDPI
9	fmPila	pilasToolStripMenuitem	Información de las pilas de las granjas.	Activo	24/08/2015 13:07:48	CAMANICA'ngonzalez	GFITDPI
10	fmRol	rolesToolStripMenuitem	Se definen los roles que se utilizarán en el sistema y se asociarán las opciones y ...	Activo	24/08/2015 13:08:37	CAMANICA'ngonzalez	GFITDPI
11	fmSemana	semanasToolStripMenuitem	Se definen las semanas para asociar las lecturas de los parámetros.	Activo	24/08/2015 13:09:07	CAMANICA'ngonzalez	GFITDPI
12	fmTipoParametro	tiposDeParametrosToolStripMenuitem	Definición de los tipos de parámetros.	Activo	24/08/2015 13:09:36	CAMANICA'ngonzalez	GFITDPI
13	fmTipoPila	tiposDePilasToolStripMenuitem	Definición de los tipos de pilas.	Activo	24/08/2015 13:10:07	CAMANICA'ngonzalez	GFITDPI
14	fmUnidadDeMedida	unidadesDeMedidaToolStripMenuitem	Definición de las unidades de medidas.	Activo	24/08/2015 13:11:44	CAMANICA'ngonzalez	GFITDPI
15	fmUsuario	usuariosToolStripMenuitem	Configuración de los usuarios.	Activo	24/08/2015 13:12:41	CAMANICA'ngonzalez	GFITDPI
16	fmUsuarioPorRol	usuariosPorRolToolStripMenuitem	Definición de los usuarios por rol.	Activo	24/08/2015 13:13:36	CAMANICA'ngonzalez	GFITDPI

Figura 44: Opciones del Sistema

Capítulo 4: Resultados del proyecto



Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frm/visor]

Proceso Reportes Configuración Seguridad Ventanas

Informe principal

CAMANICA ZONA FRANCA, S.A.
Opciones del Sistema

ID	NOMBRE	DESCRIPCIÓN	MENÚ
1	frmCiclo	Ciclo de Producción	ciclosDeProduccionToolStripMenuitem
2	frmCriptografia	Criptografía (Encriptar Cadena de Conexión)	criptografiaToolStripMenuitem
3	frmGrafica	Muestra las gráficas de los datos.	graficasToolStripMenuitem
4	frmGranja	Granjas	granjasToolStripMenuitem
5	frmLectura	Lecturas de los parámetros	lecturasToolStripMenuitem
6	frmOpcionSistema	Opciones del Sistema (Datos de los formularios y menús del sistema)	opcionesDelSistemaToolStripMenuitem
7	frmOpcionSistemaPorRol	Para definir las opciones del sistemas disponibles para los roles definidos.	opcionesDelSistemaPorRolesToolStripMenuitem
8	frmParametro	Definición de los parámetros	parametrosToolStripMenuitem
9	frmPila	Información de las pilas de las granjas.	pilasToolStripMenuitem
10	frmRol	Se definen los roles que se utilizarán en el sistema y se asociarán las opciones y los usuarios a los mismos.	rolesToolStripMenuitem
11	frmSemana	Se definen las semanas para asociar las lecturas de los parámetros.	semanasToolStripMenuitem

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 11:22:59 NUM MAY BLOQ

Figura 45: Impresión de Opciones del Sistema

Capítulo 4: Resultados del proyecto

Opciones del Sistema por Roles

Las opciones del sistema por roles son las asociaciones de cada una de las opciones a los roles configurados dentro del sistema. Esto significa que el sistema tiene la flexibilidad de permitir una combinación ilimitada entre los elementos conocidos como roles y aquellos conocidos como opciones del sistema.

Esta flexibilidad permite a un conjunto de opciones tener diferentes permisos de operación (normalmente conocidas como operaciones CRUD²¹) para diferentes roles. Cada combinación tendrá la posibilidad de operar bajo la necesidad específica configurada por el administrador.

Id	Rol	Nombre	Descripción	Crear	Editar	Borrar	Seleccionar	Imprimir	Estado	Creado El	Creado Por	Creado En	Ac
1	Administradores	fmCiclo	Ciclo de Producción	Si	Si	No	Si	Si	Activo	24/08/2015 09:50:11	CAMANICA\ngo...	GRITDPWS02	09/11/20
2	Operadores	fmCiclo	Ciclo de Producción	No	Si	No	Si	Si	Activo	24/08/2015 09:56:07	CAMANICA\ngo...	GRITDPWS02	09/11/20
3	Administradores	fmCriptografia	Criptografía (Encriptar Cadena de Co...	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
4	Administradores	fmGrafica	Muestra las gráficas de los datos.	No	No	No	Si	No	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	25/08/20
5	Administradores	fmGranja	Granjas	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	25/08/20
6	Administradores	fmLectura	Lecturas de los parámetros	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
7	Administradores	fmOpcionSistema	Opciones del Sistema (Datos de los fo...	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
8	Administradores	fmOpcionSistem...	Para definir las opciones del sistemas ...	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
9	Administradores	fmParametro	Definición de los parámetros	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
10	Administradores	fmPila	Información de las pilas de las granjas.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
11	Administradores	fmRol	Se definen los roles que se utilizarán ...	Si	No	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	24/08/20
12	Administradores	fmSemana	Se definen las semanas para asociar l...	No	No	No	No	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	25/08/20
13	Administradores	fmTipoParametro	Definición de los tipos de parámetros.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
14	Administradores	fmTipoPila	Definición de los tipos de pilas.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
15	Administradores	fmUnidadMedida	Definición de las unidades de medidas.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
16	Administradores	fmUsuario	Configuración de los usuarios.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
17	Administradores	fmUsuarioPorRol	Definición de los usuarios por rol.	Si	Si	Si	Si	Si	Activo	24/08/2015 14:28:13	CAMANICA\ngo...	GRITDPWS02	01/01/19
10003	Sopote Técnico	fmParametro	Definición de los parámetros	Si	No	Si	Si	Si	Activo	18/09/2015 16:25:08	CAMANICA\ngo...	GRITDPWS02	01/01/19
10004	Sopote Técnico	fmPila	Información de las pilas de las granjas.	Si	Si	Si	Si	Si	Activo	18/09/2015 16:25:46	CAMANICA\ngo...	GRITDPWS02	01/01/19

Figura 46: Opciones del Sistema por Roles

²¹ CRUD: Create, Read, Update, and Delete. Operaciones DML que representan la creación, lectura, actualización y borrado de los datos en una entidad de una base de datos relacional.

Capítulo 4: Resultados del proyecto

Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frm/visor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS®

Informe principal

CAMANICA ZONA FRANCA, S.A.
Opciones del Sistema por Roles

ID	DESCRIPCIÓN	C	R	U	D	REPORTE
Administradores [1]						
1	Ciclo de Producción	Si	Si	Si	Si	Si
2	Criptografía (Encryptar Cadena de Conexión)	Si	Si	Si	Si	Si
3	Muestra las gráficas de los datos.	No	Si	No	No	No
4	Granjas	Si	Si	Si	Si	Si
5	Lecturas de los parámetros	Si	Si	Si	Si	Si
6	Opciones del Sistema (Datos de los formularios y menús del sistema)	Si	Si	Si	Si	Si
7	Para definir las opciones del sistemas disponibles para los roles definidos.	Si	Si	Si	Si	Si
8	Definición de los parámetros	Si	Si	Si	Si	Si
9	Información de las pilas de las granjas.	Si	Si	Si	Si	Si
10	Se definen los roles que se utilizarán en el sistema y se asociarán las opcio	Si	Si	Si	Si	Si
11	Se definen las semanas para asociar las lecturas de los parámetros.	Si	Si	Si	Si	Si
12	Definición de los tipos de parámetros.	Si	Si	Si	Si	Si
13	Definición de los tipos de pilas.	Si	Si	Si	Si	Si
14	Definición de las unidades de medidas.	Si	Si	Si	Si	Si
15	Configuración de los usuarios.	Si	Si	Si	Si	Si
16	Definición de los usuarios por rol.	Si	Si	Si	Si	Si
SUBTOTAL OPCIONES:		16				
Operadores [2]						
1	Ciclo de Producción	No	Si	Si	No	Si

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 11:25:27 NUM MAY BLOQ

Figura 47: Impresión de Opciones del Sistema por Roles

Capítulo 4: Resultados del proyecto

Criptografía

Esta opción la utiliza el administrador del sistema para configurar la cadena de conexión utilizada por el sistema. El administrador ingresa la cadena de conexión hacia el servidor SQL y esta opción cifra dicha cadena. Luego, se almacena el valor obtenido en un archivo llamado "cadena.txt" y se guarda en la carpeta raíz donde se encuentra el archivo ejecutable del sistema.

Al iniciar el sistema buscará automáticamente este archivo y obtendrá de él la cadena de conexión que utilizará para acceder a la base de datos del sistema. A como se mencionó anteriormente, esta cadena de conexión utiliza la seguridad integrada de Windows como método de autenticación entre el cliente y servidor.

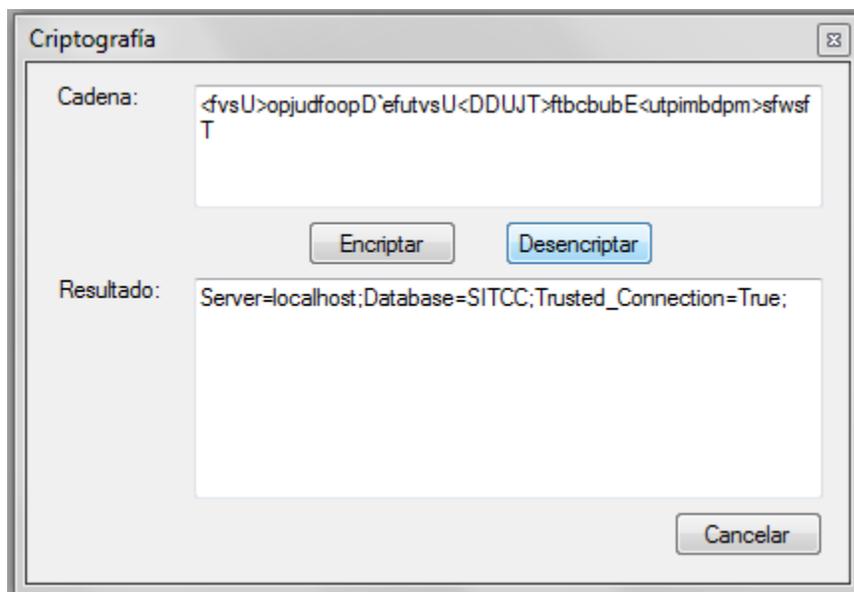


Figura 48: Criptografía

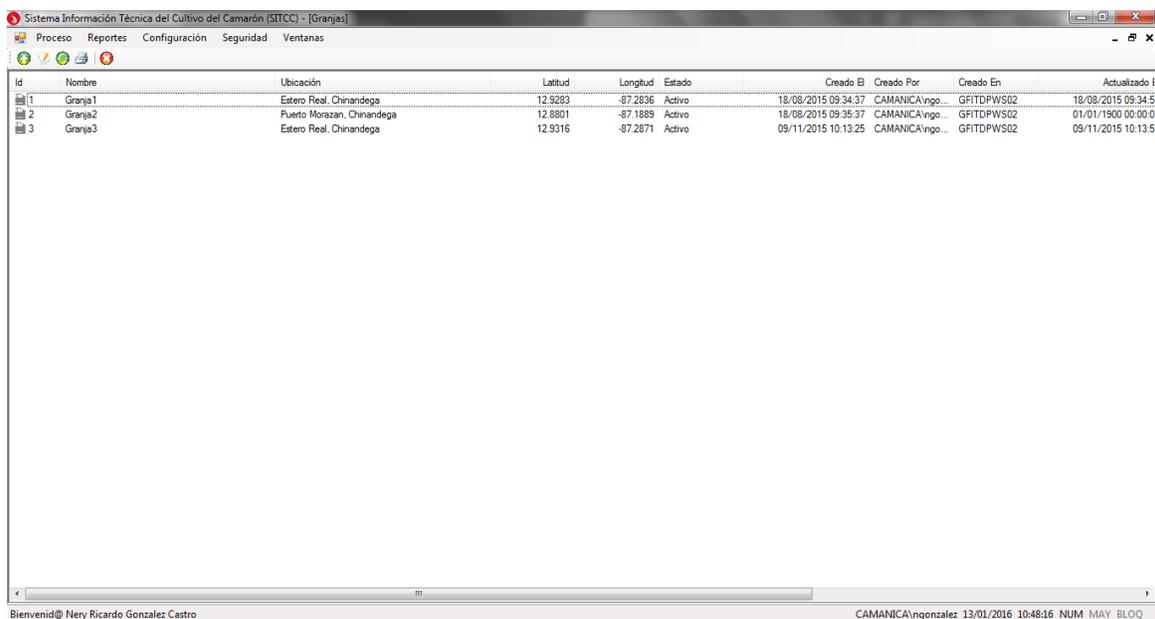
Capítulo 4: Resultados del proyecto

Configuración

Esta opción agrupa todos aquellos componentes que se utilizan en el sistema y que forman parte de la información de apoyo para los procesos definidos dentro del sistema. En otras palabras, esta sección contempla todos los catálogos utilizados por el sistema. A continuación se detalla cada uno de ellos:

Granjas

En el catálogo granjas se puede editar, eliminar, actualizar e imprimir toda la información relacionada a las granjas pertenecientes a la empresa, tales como: nombre, ubicación, latitud, longitud.



The screenshot shows a software window titled 'Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Granjas]'. The window has a menu bar with 'Proceso', 'Reportes', 'Configuración', 'Seguridad', and 'Ventanas'. Below the menu bar is a toolbar with several icons. The main area contains a table with the following data:

Id	Nombre	Ubicación	Latitud	Longitud	Estado	Creado El	Creado Por	Creado En	Actualizado E
1	Granja1	Estero Real, Chinandega	12.9283	-87.2836	Activo	18/08/2015 09:34:37	CAMANICAvingo...	GFITDPWS02	18/08/2015 09:34:5
2	Granja2	Puerto Morazan, Chinandega	12.8801	-87.1889	Activo	18/08/2015 09:35:37	CAMANICAvingo...	GFITDPWS02	01/01/1900 00:00:0
3	Granja3	Estero Real, Chinandega	12.9316	-87.2871	Activo	09/11/2015 10:13:25	CAMANICAvingo...	GFITDPWS02	09/11/2015 10:13:5

At the bottom of the window, there is a status bar with the text 'Bienvenid@ Nery Ricardo Gonzalez Castro' on the left and 'CAMANICA\ngonzalez 13/01/2016 10:48:16 NUM MAY BLOQ' on the right.

Figura 49: Configuración de Granjas

Capítulo 4: Resultados del proyecto

Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frmVisor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS*

Informe principal

 **CAMANICA ZONA FRANCA, S.A.**
Granjas

ID	NOMBRE	UBICACIÓN	LATITUD	LONGITUD
1	Granja1	Estero Real, Chinandega	12.928300	-87.283600
2	Granja2	Puerto Morazan, Chinandega	12.880100	-87.188900
3	Granja3	Estero Real, Chinandega	12.931600	-87.287100
TOTAL DE GRANJAS:		3		

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

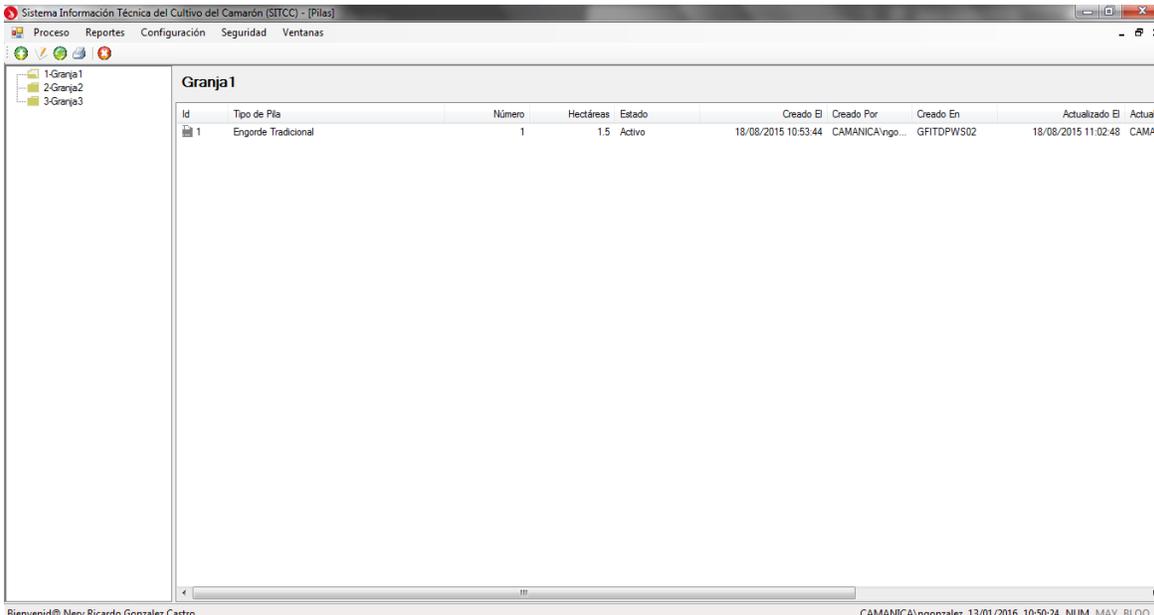
Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez: 13/01/2016 11:29:09 NUM MAY BLOQ

Figura 50: Impresión de Granjas

Capítulo 4: Resultados del proyecto

Pilas

En el catálogo pilas se puede editar, eliminar, actualizar e imprimir toda la información relacionada a las pilas pertenecientes a una granja, tales como: granja, tipo de pila, número, hectáreas.

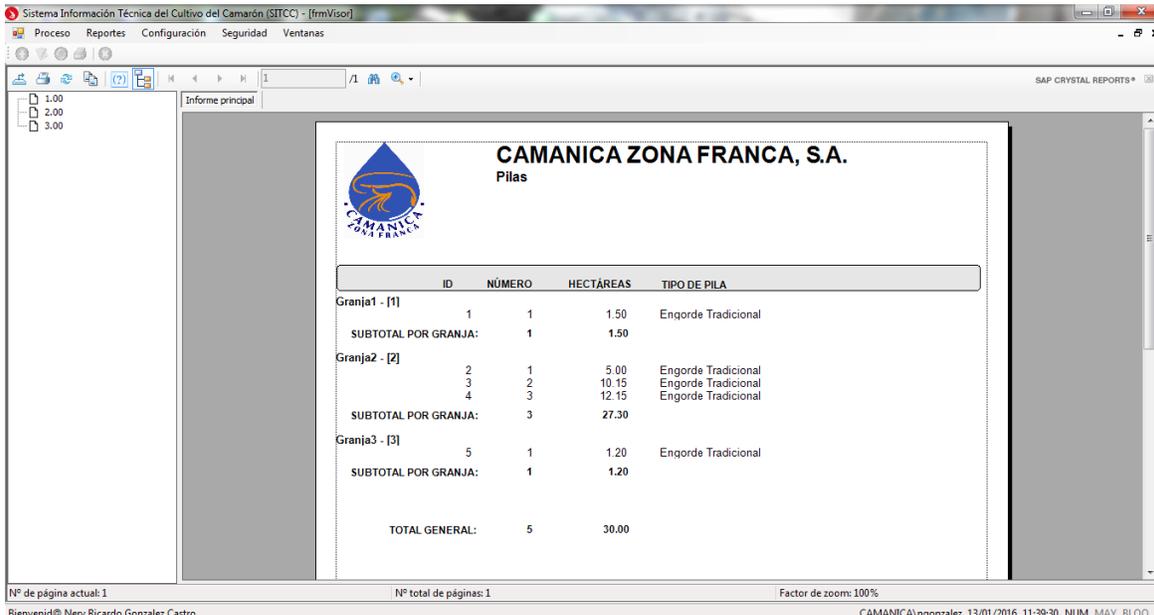


The screenshot shows a web application window titled "Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Pilas]". The main content area displays a table for "Granja 1" with the following data:

Id	Tipo de Pila	Número	Hectáreas	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por
1	Engorde Tradicional	1	1.5	Activo	18/08/2015 10:53:44	CAMANICA\ngonzalez...	GFITDPWS02	18/08/2015 11:02:48	CAMANICA\ngonzalez...

At the bottom of the window, the user is identified as "Bienvenid@ Nery Ricardo Gonzalez Castro" and the system date/time is "CAMANICA\ngonzalez: 13/01/2016 10:50:24 NUM MAY BLOQ".

Figura 51: Configuración de Pilas



The screenshot shows a printed report titled "Informe principal" for "CAMANICA ZONA FRANCA, S.A. Pilas". The report contains a table with the following data:

ID	NÚMERO	HECTÁREAS	TIPO DE PILA
Granja1 - [1]			
1	1	1.50	Engorde Tradicional
SUBTOTAL POR GRANJA:		1	1.50
Granja2 - [2]			
2	1	5.00	Engorde Tradicional
3	2	10.15	Engorde Tradicional
4	3	12.15	Engorde Tradicional
SUBTOTAL POR GRANJA:		3	27.30
Granja3 - [3]			
5	1	1.20	Engorde Tradicional
SUBTOTAL POR GRANJA:		1	1.20
TOTAL GENERAL:		5	30.00

At the bottom of the report, it shows "Nº de página actual: 1", "Nº total de páginas: 1", and "Factor de zoom: 100%". The user and system information at the bottom match the previous screenshot.

Figura 52: Impresión de Pilas

Capítulo 4: Resultados del proyecto

Ciclos

En el catálogo ciclos se puede editar, eliminar, actualizar e imprimir toda la información relacionada a los ciclos de producción de las pilas, tales como: granja, número de pila, año, número, fecha desde, fecha hasta, fecha de siembra y fecha de cosecha.

Los ciclos de producción representan rangos de fechas que pueden extenderse a lo largo de varios meses e incluso abarcar varios años. El ciclo de producción representa el ciclo de crecimiento del camarón desde su siembra hasta su cosecha.

Id	Número Pila	Año	Número	Desde	Hasta	Fecha de Siembra	Fecha de Cosecha	Estado	Creado El	Crs
2	1	2015	1	29/12/2014	01/01/2016	20/01/2015	01/01/2016	Activo	18/08/2015 11:44:45	CAI

Figura 53: Configuración de Ciclos de Producción

Capítulo 4: Resultados del proyecto

CAMANICA ZONA FRANCA, S.A.
Ciclos por Pilas

ID	AÑO	NÚMERO	DESDE	HASTA	DÍAS	COSECHA		
						INICIO	FIN	
Granja1 - [1]								
Pila N° 1	2015	1	29/12/2014	01/01/2016	368	20/01/2015	01/01/2016	
TOTAL CICLOS:					1	368		
Granja2 - [2]								
Pila N° 1	3	2015	1	21/01/2015	24/06/2015	154	27/01/2015	25/06/2015
TOTAL CICLOS:					1	154		
Pila N° 2								
Pila N° 2	4	2015	1	13/04/2015	24/09/2015	164	30/04/2015	25/09/2015
TOTAL CICLOS:					1	164		
Pila N° 3								
Pila N° 3	5	2015	1	05/05/2015	14/09/2015	132	11/05/2015	15/09/2015
TOTAL CICLOS:					1	132		

NP de página actual: 1 N° total de páginas: 1 Factor de zoom: 100%

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA/ngonzalez: 13/01/2016 11:58:56 NUM MAY BLOQ

Figura 54: Impresión de Ciclos de Producción por Pilas

Capítulo 4: Resultados del proyecto

Semanas

En el catálogo semanas se puede editar, eliminar, actualizar e imprimir toda la información relacionada a las semanas de corte de la información técnica, tales como: año, número, fecha desde, fecha hasta. Sin embargo, estas semanas tiene una característica especial donde el inicio de semana no coincide con el inicio de semana definido internacionalmente. Esto permite definir una semana totalmente configurable para abarcar los días necesarios para soportar los informes técnicos elaborados. El usuario autorizado podrá gestionar la información de las semanas.

The screenshot shows a web application window titled 'Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Semanas]'. The window contains a table with the following columns: Id, Año, Número, Desde, Hasta, Estado, Creado El, Creado Por, Creado En, Actualizado El, Actualizado Por, Actualizado En, and a final column for user/permissions. The table lists 36 rows of week configurations for the year 2015, with 'Desde' and 'Hasta' dates and 'Creado En' and 'Actualizado En' timestamps.

Id	Año	Número	Desde	Hasta	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por	Actualizado En	
1	2015	1	09/01/2015	14/01/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
2	2015	2	15/01/2015	21/01/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
3	2015	3	22/01/2015	28/01/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
4	2015	4	29/01/2015	04/02/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
5	2015	5	05/02/2015	11/02/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
6	2015	6	12/02/2015	18/02/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
7	2015	7	19/02/2015	25/02/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
8	2015	8	26/02/2015	04/03/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
9	2015	9	05/03/2015	11/03/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
10	2015	10	12/03/2015	18/03/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
11	2015	11	19/03/2015	25/03/2015	Activo	25/05/2015 20:00:25	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
12	2015	12	26/03/2015	01/04/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
13	2015	13	02/04/2015	08/04/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
14	2015	14	09/04/2015	15/04/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
15	2015	15	16/04/2015	22/04/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
16	2015	16	23/04/2015	29/04/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
17	2015	17	30/04/2015	06/05/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
18	2015	18	07/05/2015	13/05/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
19	2015	19	14/05/2015	20/05/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
20	2015	20	21/05/2015	27/05/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
21	2015	21	28/05/2015	03/06/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
22	2015	22	04/06/2015	10/06/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
23	2015	23	11/06/2015	17/06/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
24	2015	24	18/06/2015	24/06/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
25	2015	25	25/06/2015	01/07/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
26	2015	26	02/07/2015	08/07/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
27	2015	27	09/07/2015	15/07/2015	Activo	25/05/2015 20:00:26	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
28	2015	28	16/07/2015	22/07/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
29	2015	29	23/07/2015	29/07/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
30	2015	30	30/07/2015	05/08/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
31	2015	31	06/08/2015	12/08/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
32	2015	32	13/08/2015	19/08/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
33	2015	33	20/08/2015	26/08/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
34	2015	34	27/08/2015	02/09/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
35	2015	35	03/09/2015	09/09/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	
36	2015	36	10/09/2015	16/09/2015	Activo	25/05/2015 20:00:27	CAMANICA'ngonzalez	GRITDPWS02	01/01/1900 00:00:00		01/01/1	

At the bottom of the window, there is a status bar with the text: 'Bienvenid@ Nery Ricardo Gonzalez: Castro' on the left and 'CAMANICA'ngonzalez: 13/01/2016 10:51:40 NUM MAY BLOQ' on the right.

Figura 55: Configuración de Semanas

Capítulo 4: Resultados del proyecto

Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frm/visor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS*

Informe principal

CAMANICA ZONA FRANCA, S.A.
Semanas del Año

ID	NÚMERO	DESDE	HASTA
2015	1	08/01/2015	14/01/2015
	2	15/01/2015	21/01/2015
	3	22/01/2015	28/01/2015
	4	29/01/2015	04/02/2015
	5	05/02/2015	11/02/2015
	6	12/02/2015	18/02/2015
	7	19/02/2015	25/02/2015
	8	26/02/2015	04/03/2015
	9	05/03/2015	11/03/2015
	10	12/03/2015	18/03/2015
	11	19/03/2015	25/03/2015
	12	26/03/2015	01/04/2015
	13	02/04/2015	08/04/2015
	14	09/04/2015	15/04/2015
	15	16/04/2015	22/04/2015
	16	23/04/2015	29/04/2015
	17	30/04/2015	06/05/2015
	18	07/05/2015	13/05/2015
	19	14/05/2015	20/05/2015
	20	21/05/2015	27/05/2015
	21	28/05/2015	03/06/2015

Nº de página actual: 1 Nº total de páginas: 2 Factor de zoom: 100%

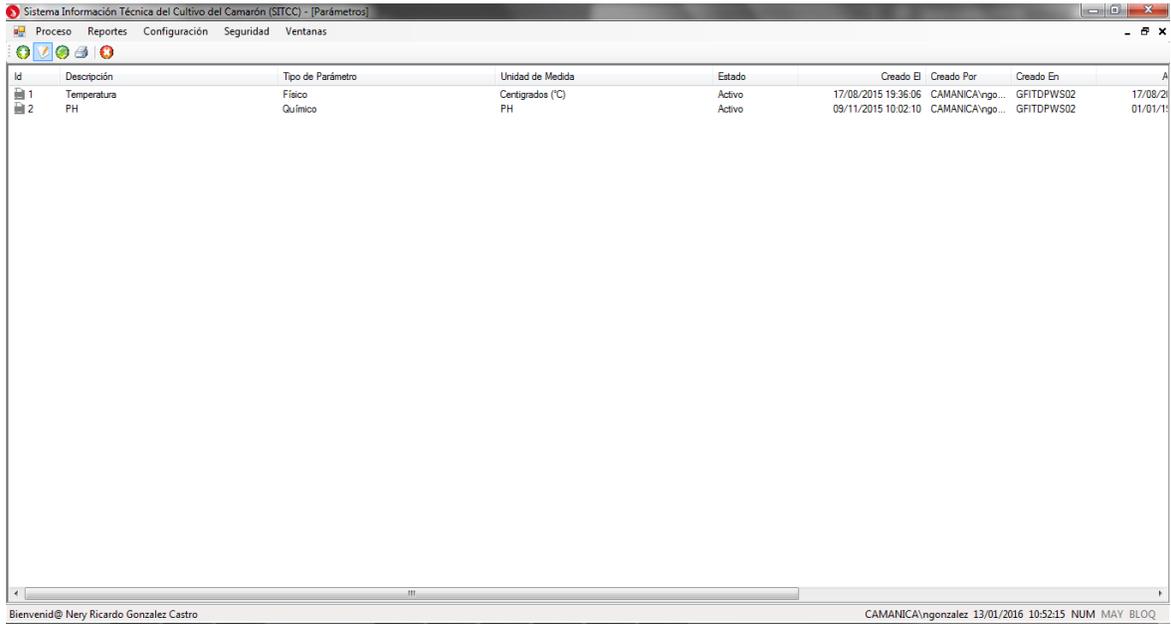
Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez: 13/01/2016 12:01:47 NUM MAY BLOQ

Figura 56: Impresión de Semanas del Año

Capítulo 4: Resultados del proyecto

Parámetros

En el catálogo parámetros se puede editar, eliminar, actualizar e imprimir toda la información relacionada a los parámetros que se miden en las granjas de la empresa, tales como: descripción, tipo de parámetro, unidad de medida.

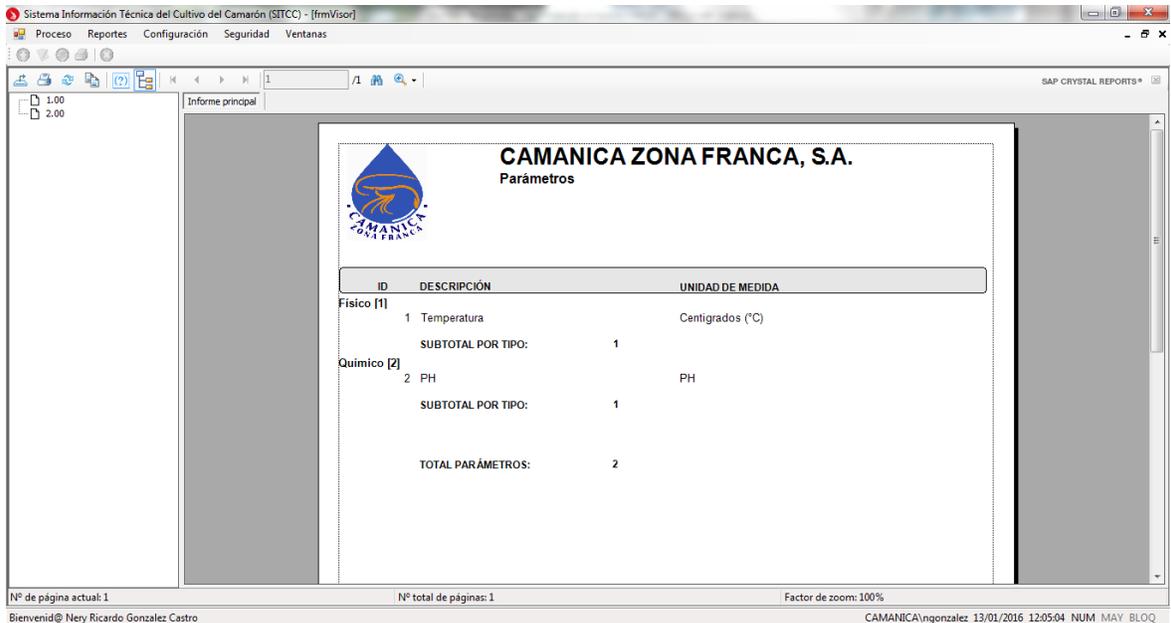


Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Parámetros]

Id	Descripción	Tipo de Parámetro	Unidad de Medida	Estado	Creado El	Creado Por	Creado En	A
1	Temperatura	Físico	Centígrados (°C)	Activo	17/08/2015 19:36:06	CAMANICA\ngonzalez...	GFITDPWS02	17/08/2015
2	PH	Químico	PH	Activo	09/11/2015 10:02:10	CAMANICA\ngonzalez...	GFITDPWS02	01/01/2016

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 10:52:15 NUM MAY BLOQ

Figura 57: Configuración de Parámetros



Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frmVisor]

Informe principal

CAMANICA ZONA FRANCA, S.A.
Parámetros

ID	DESCRIPCIÓN	UNIDAD DE MEDIDA
Físico [1]		
1	Temperatura	Centígrados (°C)
SUBTOTAL POR TIPO:		1
Químico [2]		
2	PH	PH
SUBTOTAL POR TIPO:		1
TOTAL PARÁMETROS:		2

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

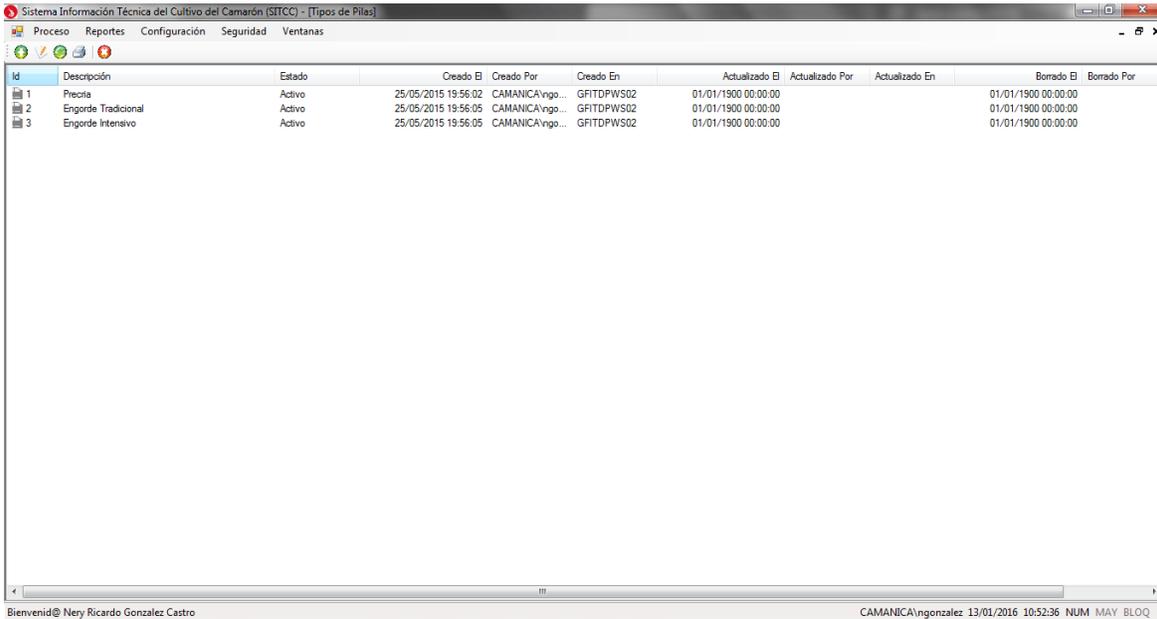
Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 12:05:04 NUM MAY BLOQ

Figura 58: Impresión de Parámetros

Capítulo 4: Resultados del proyecto

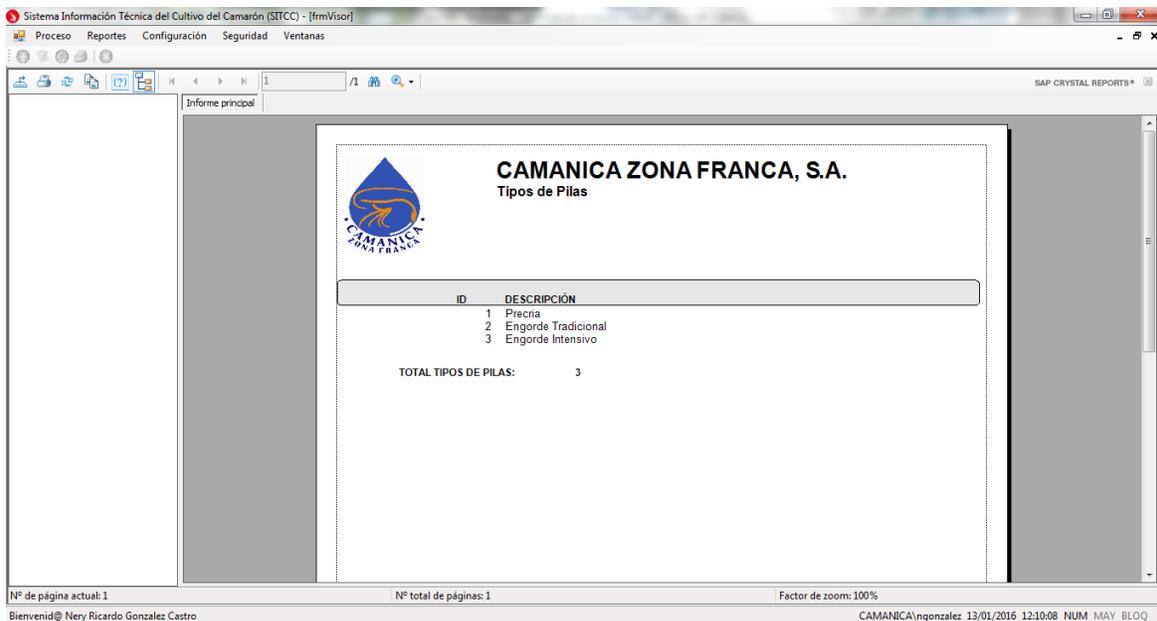
Tipos de Pilas

En el catálogo tipos de pilas se puede editar, eliminar, actualizar e imprimir toda la información relacionada a los tipos de pilas en que se clasifican las pilas de las granjas pertenecientes a la empresa, tales como: descripción.



Id	Descripción	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por	Actualizado En	Borrado El	Borrado Por
1	Precna	Activo	25/05/2015 19:56:02	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	
2	Engorde Tradicional	Activo	25/05/2015 19:56:05	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	
3	Engorde Intensivo	Activo	25/05/2015 19:56:05	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	

Figura 59: Configuración de Tipos de Pilas



Informe principal

SAP CRYSTAL REPORTS *

CAMANICA ZONA FRANCA, S.A.
Tipos de Pilas

ID	DESCRIPCIÓN
1	Precna
2	Engorde Tradicional
3	Engorde Intensivo

TOTAL TIPOS DE PILAS: 3

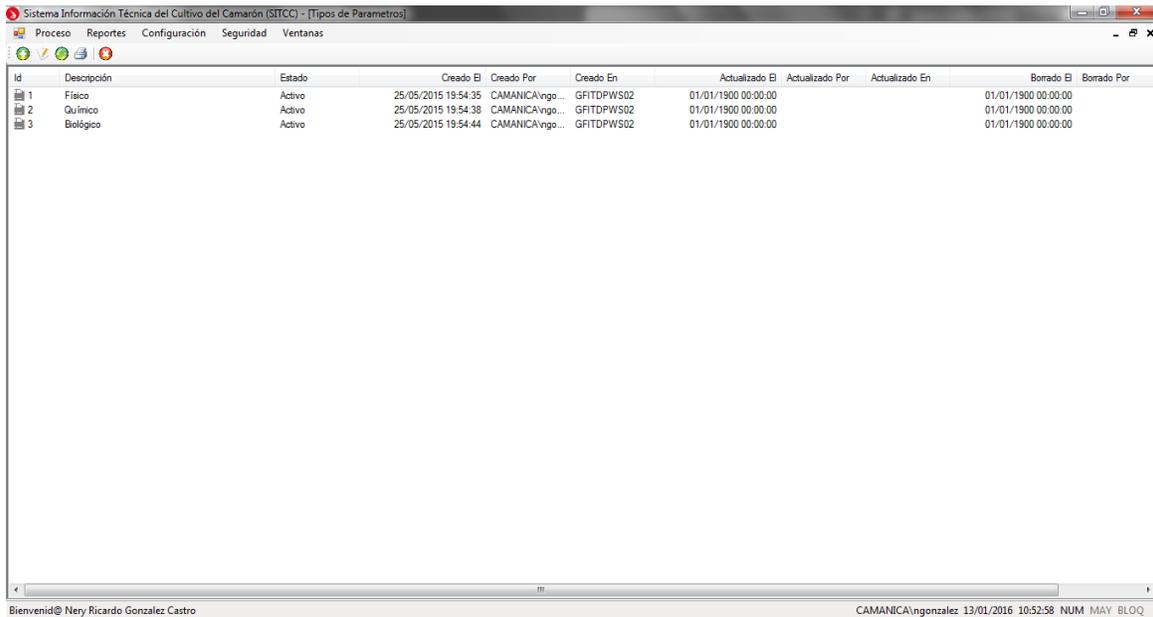
Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Figura 60: Impresión de Tipos de Pilas

Capítulo 4: Resultados del proyecto

Tipos de Parámetros

En el catálogo tipos de parámetros se puede editar, eliminar, actualizar e imprimir toda la información relacionada a los tipos de parámetros en los que se pueden clasificar los parámetros que se miden en la empresa, tales como: descripción.

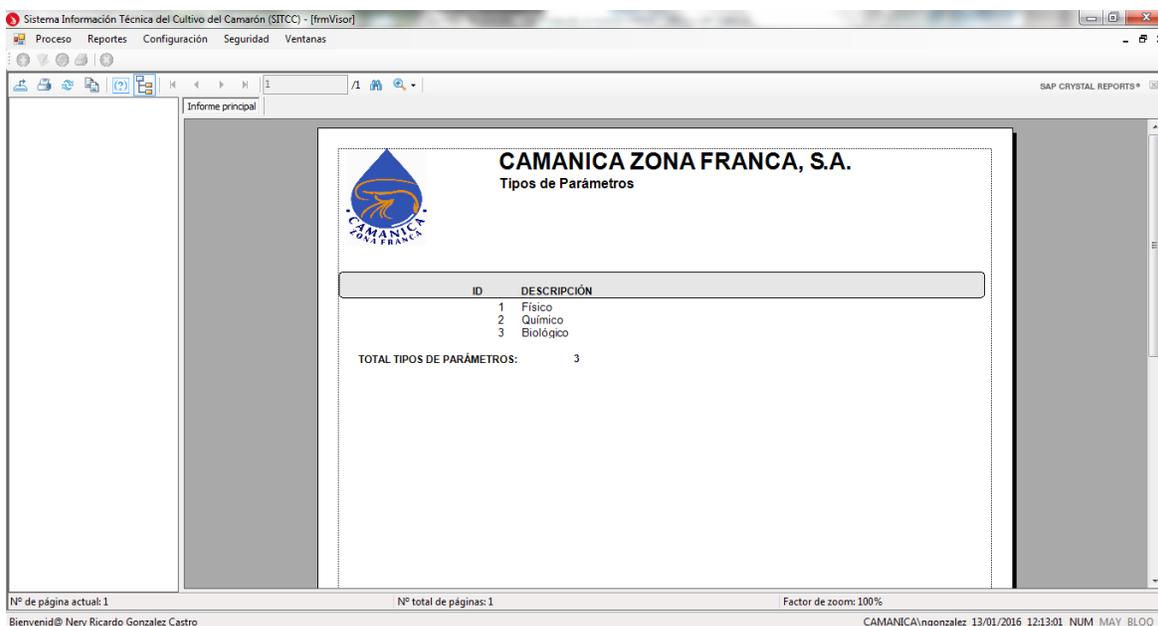


Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Tipos de Parametros]

Id	Descripción	Estado	Creado En	Creado Por	Creado En	Actualizado En	Actualizado Por	Actualizado En	Borrado En	Borrado Por
1	Físico	Activo	25/05/2015 19:54:35	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	
2	Químico	Activo	25/05/2015 19:54:38	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	
3	Biológico	Activo	25/05/2015 19:54:44	CAMANICA\ngo...	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 10:52:58 NUM MAY BLOQ

Figura 61: Configuración de Tipos de Parámetros



Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frmVisor]

Informe principal

CAMANICA ZONA FRANCA, S.A.
Tipos de Parámetros

ID	DESCRIPCIÓN
1	Físico
2	Químico
3	Biológico

TOTAL TIPOS DE PARÁMETROS: 3

NP de página actual: 1 NP total de páginas: 1 Factor de zoom: 100%

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez 13/01/2016 12:13:01 NUM MAY BLOQ

Figura 62: Impresión de Tipos de Parámetros

Capítulo 4: Resultados del proyecto

Unidades de Medidas

En el catálogo unidades de medidas se puede editar, eliminar, actualizar e imprimir toda la información relacionada a las unidades de medidas utilizadas por los parámetros que se miden en la empresa, tales como: descripción. Estas unidades de medidas definen los tipos de valores que se grabarán para los parámetros que se desean medir.

Id	Descripción	Estado	Creado El	Creado Por	Creado En	Actualizado El	Actualizado Por	Actualizado En	Borrado El	Borrado Por
1	Centigrados (°C)	Activo	17/08/2015 18:57:52	CAMANICA\ngonzalez	GRITDPWS02	17/08/2015 18:58:20	CAMANICA\ngonzalez	GRITDPWS02	01/01/1900 00:00:00	01/01/1900 00:00:00
2	Porcentaje	Activo	17/08/2015 18:58:07	CAMANICA\ngonzalez	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	01/01/1900 00:00:00
3	PH	Activo	09/11/2015 10:02:03	CAMANICA\ngonzalez	GRITDPWS02	01/01/1900 00:00:00			01/01/1900 00:00:00	01/01/1900 00:00:00

Figura 63: Configuración de Unidades de Medidas

ID	DESCRIPCIÓN
1	Centigrados (°C)
2	Porcentaje
3	PH

TOTAL UNIDAD DE MEDIDA: 3

Figura 64: Impresión de Unidades de Medidas

Capítulo 4: Resultados del proyecto

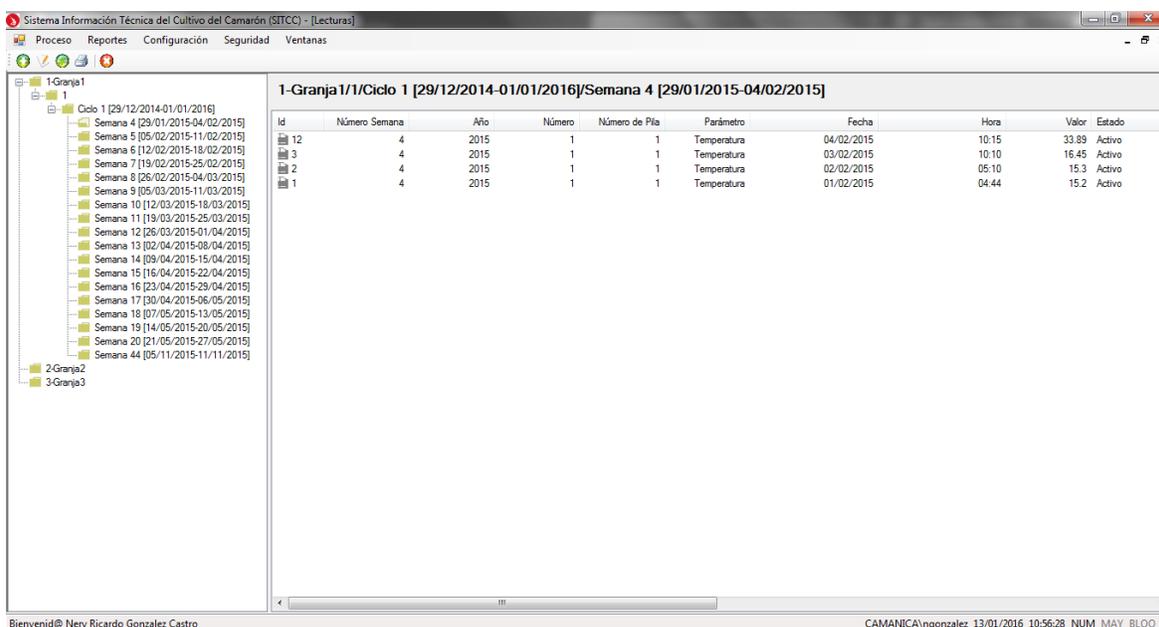
Proceso

Esta opción agrupa todos aquellos procesos operativos que se registran en el sistema. En otras palabras, esta sección contempla toda la información técnica del camarón disponible para ser analizada. A continuación se detalla cada uno de ellos:

Lecturas

Esta opción se utiliza para recopilar todas las lecturas obtenidas de los parámetros que se desean medir en las diferentes pilas de las granjas de la empresa. La información acá detallada provee una retroalimentación técnica necesaria para la toma de decisiones que permiten garantizar el término satisfactorio de la cosecha del camarón en un ciclo de producción determinado.

Esta opción es considerada la opción medular del sistema. El correcto registro de las lecturas garantizará tener resultados exitosos al término de cada ciclo de producción. De igual manera, se convierte en una base de datos históricos disponible para futuras referencia para determinar escenarios similares que se puedan presentar en una determinada granja.



The screenshot shows the 'Lecturas' window in the SITCC system. The left pane shows a tree view with '1-Granja1' expanded to show 'Ciclo 1' and its 20 weeks. The right pane shows a table of readings for 'Semana 4'.

Id	Número Semana	Año	Número	Número de Pila	Parámetro	Fecha	Hora	Valor	Estado
12	4	2015	1	1	Temperatura	04/02/2015	10:15	33.89	Activo
3	4	2015	1	1	Temperatura	03/02/2015	10:10	16.45	Activo
2	4	2015	1	1	Temperatura	02/02/2015	05:10	15.3	Activo
1	4	2015	1	1	Temperatura	01/02/2015	04:44	15.2	Activo

Figura 65: Lecturas de parámetros

En el registro de la información entran en juego todos los elementos definidos en el sistema que permite grabar la información exacta de lo sucedido en una determinada pila en determinado plano de tiempo.

El usuario con los permisos autorizados de operación podrá registrar lecturas al sistema bajo ciertas condiciones de control. Solamente se podrán grabar lecturas en pilas donde la fecha de obtención de la lectura esté relacionada a un ciclo de producción vigente para esa pila. De esta manera, si un ciclo está cerrado no se podrá grabar información relacionada a la fecha definida.

Capítulo 4: Resultados del proyecto

Sistema Información Técnica del Cultivo del Camarón (SITCC) - [Lecturas]

Proceso Reportes Configuración Seguridad Ventanas

1-Granja1

Id	Número Semana	Año	Número	Número de Pila	Parámetro	Fecha	Hora	Valor	Est
10016	44	2015	1	1	PH	07/11/2015	11:25	7	Acti
10015						08/11/2015	09:07	4	Acti
10014						09/11/2015	10:50	5.5	Acti
10013						09/11/2015	10:02	5	Acti
10012						09/11/2015	10:00	19.8	Acti
119						22/05/2015	09:09	16.26	Acti
118						21/05/2015	11:14	13.73	Acti
117						20/05/2015	09:22	18.67	Acti
116						19/05/2015	07:29	27.63	Acti
115						18/05/2015	08:26	30.07	Acti
114						17/05/2015	08:44	34.06	Acti
113						16/05/2015	07:46	31.96	Acti
112						15/05/2015	06:07	31.77	Acti
111						14/05/2015	06:54	33.47	Acti
110						13/05/2015	08:01	30.22	Acti
109						12/05/2015	11:08	19.32	Acti
108						11/05/2015	11:43	31.32	Acti
107						10/05/2015	09:03	31.67	Acti
106						09/05/2015	08:44	20.59	Acti
105						08/05/2015	10:03	28.74	Acti
104						07/05/2015	07:00	28.2	Acti
103						06/05/2015	07:55	12.23	Acti
102						05/05/2015	11:37	33.34	Acti
101						04/05/2015	10:38	31.45	Acti
100						03/05/2015	06:28	19.05	Acti
99						02/05/2015	06:37	18.76	Acti
98						01/05/2015	10:08	28.53	Acti
97						30/04/2015	09:58	32.6	Acti
96						29/04/2015	07:25	24.36	Acti
95	16	2015	1	1	Temperatura	28/04/2015	11:16	28.11	Acti
94	16	2015	1	1	Temperatura	27/04/2015	06:04	24.44	Acti
93	16	2015	1	1	Temperatura	26/04/2015	07:26	26.28	Acti

Crear Lectura

Granja: Granja1
Fecha: 13/01/2016
Parámetro: Temperatura
PH

ID	Número de Pila	Hora	Valor
*		12:17	

Aceptar Cancelar

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez: 13/01/2016 12:17:18 NUM MAY BLOQ

Figura 66: Grabación de Lecturas de parámetros

Sistema Información Técnica del Cultivo del Camarón (SITCC) - [frmVisor]

Proceso Reportes Configuración Seguridad Ventanas

SAP CRYSTAL REPORTS*

Informe principal

CAMANICA ZONA FRANCA, S.A.
Lecturas de Parámetros

ID	FECHA	HORA	VALOR
Granja1 - [1]			
Pila N° 1			
Ciclo 2015 - 1			
Semana 4			
Fisico			
Temperatura [Centigrados (°C)]			
1	01/02/2015	04:44:48	15.20
2	02/02/2015	05:10:00	15.30
3	03/02/2015	10:10:00	16.45
12	04/02/2015	10:15:00	33.89
			MINIMO: 15.20
			PROMEDIO: 20.21
			MAXIMO: 33.89
			DES. ESTÁNDAR: 9.14
			VARIANZA: 83.50
Semana 5			
Fisico			
Temperatura [Centigrados (°C)]			
13	05/02/2015	11:38:00	21.24
14	06/02/2015	06:58:00	25.64
15	07/02/2015	07:51:00	15.94
16	08/02/2015	06:13:00	18.14

Nº de página actual: 1 Nº total de páginas: 29 Factor de zoom: 100%

Bienvenid@ Nery Ricardo Gonzalez Castro CAMANICA\ngonzalez: 13/01/2016 12:18:52 NUM MAY BLOQ

Figura 67: Impresión de Lecturas de Parámetros

Capítulo 4: Resultados del proyecto

Gráficas

Esta opción se utiliza para visualizar todas las lecturas obtenidas de los parámetros que se desean medir en las diferentes pilas de las granjas de la empresa. La información recopilada previamente en el proceso de grabación de lecturas, puede ser analizada por el jefe de una granja. Acá podrá visualizar de manera dinámica la información de una pila o todas las pilas de la granja para un rango de fechas definido por él mismo.

Las gráficas disponibles son líneas de datos con serie de tiempo. De igual manera, se puede visualizar una línea de tendencia; al igual, que se obtiene la ecuación de regresión lineal utilizada por la gráfica.

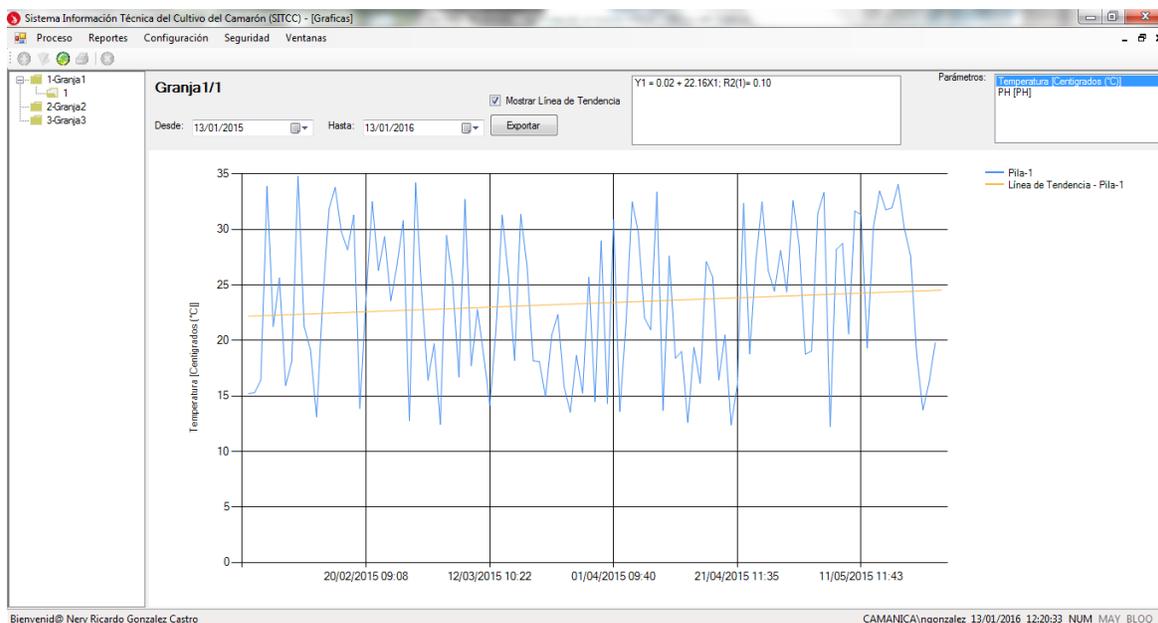


Figura 68: Gráfica de Lecturas de una pila y línea de tendencia

Capítulo 4: Resultados del proyecto

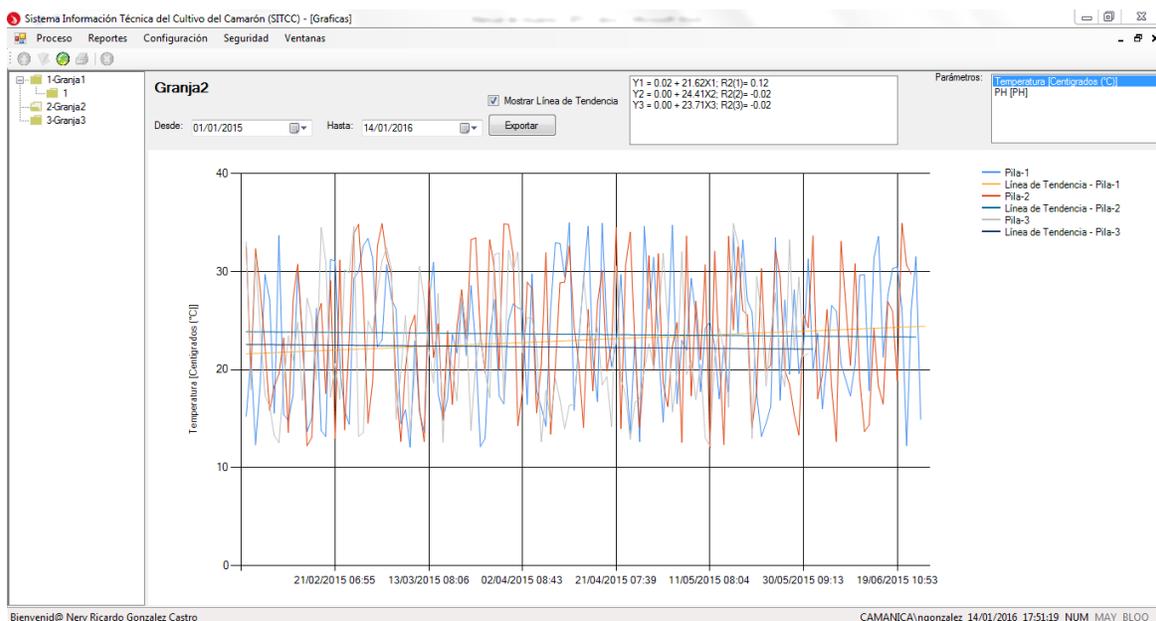


Figura 69: Gráfica de Lecturas de múltiples pilas y líneas de tendencia

Adicionalmente, la información visualizada [filtrada] puede ser exportada a un archivo de texto con formato CSV. Éste archivo puede ser utilizado para analizarlo en otros programas o software que permitan el análisis estadístico.

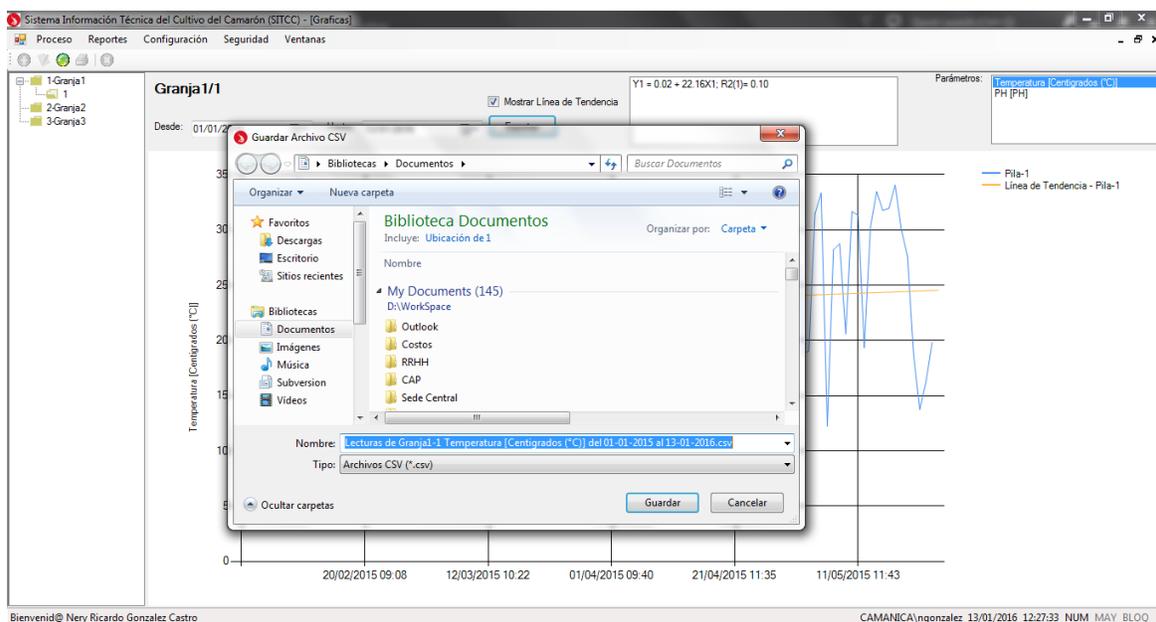
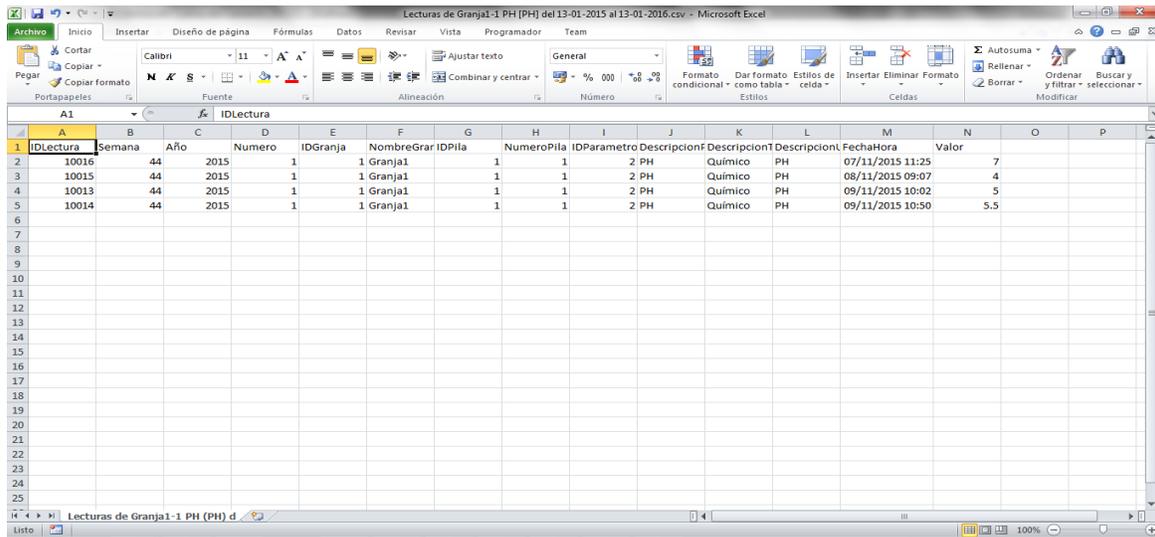


Figura 70: Exportación de Lecturas hacia archivo de texto

Capítulo 4: Resultados del proyecto



IDLectura	semana	Año	Numero	IDGranja	NombreGrar	IDPila	NumeroPila	IDParametro	DescripcionF	DescripcionT	DescripcionI	FechaHora	Valor
10016	44	2015	1	1	Granja1	1	1	2	PH	Químico	PH	07/11/2015 11:25	7
10015	44	2015	1	1	Granja1	1	1	2	PH	Químico	PH	08/11/2015 09:07	4
10013	44	2015	1	1	Granja1	1	1	2	PH	Químico	PH	09/11/2015 10:02	5
10014	44	2015	1	1	Granja1	1	1	2	PH	Químico	PH	09/11/2015 10:50	5.5

Figura 71: Lecturas importadas desde archivo de texto hacia Microsoft Excel

Capítulo 5: Aspectos Finales

Capítulo 5: Aspectos Finales

Conclusiones

Al finalizar el desarrollo del sistema se obtuvieron las siguientes conclusiones:

- Se logró una automatización del flujo de procesos con la agilización de la digitación de las lecturas de parámetros a través de la estandarización del proceso de digitación de la información técnica implementado en todas las granjas de la empresa.
- Se logró obtener de una manera más ágil la información permitiendo en menor período tener la base técnica para la toma de decisión por parte de los jefes de granjas.
- El sistema a través de sus gráficos provee a sus usuarios finales con una herramienta de fácil uso e interpretación ayudando a realizar un análisis instantáneo de la información visualizada utilizando la herramienta Microsoft® Chart para formularios Windows.
- Con la recopilación y almacenamiento de la información en una base de datos; se logra crear, un repositorio de información técnica disponible para consultas sin la restricción del ciclo de producción en el que éstas fueron recopiladas.

Recomendaciones

Recomendaciones de uso

- Establecer un procedimiento para la definición de roles y permisos que los usuarios pueden manejar.
- Conciliar periódicamente la validez del estado de los usuarios registrados en el sistema versus el Directorio Activo del dominio de la empresa.
- Ingresar información fidedigna y comprobable para establecer la validez de la misma como información histórica para usos futuros.
- Mantener actualizado y al día las configuraciones necesarias para garantizar la integridad de la información reportada.

Recomendaciones o mejoras a la aplicación

- Establecer mínimos y máximos para los valores de las lecturas obtenidas de los parámetros medidos.
- Definir un repositorio de informes dentro del sistema que sea totalmente configurable y dinámico que permitan agregar mayor flexibilidad al proceso de extraer información.
- Agregar al sistema los aspectos de costos de producción; para poder, realizar análisis costo/beneficio en base al estado actual de la información técnica.
- Desarrollar un sistema paralelo y complementario de “back office” para consultas en líneas y a través de dispositivos móviles.

Bibliografía

Bibliografía

- [1] S. N. International, «News Reports,» Bob Rosenberry, 4 Diciembre 2013. [En línea]. Available: <http://www.shrimpnews.com/FreeReportsFolder/NewsReportsFolder/NicaraguaCamanicaPescanovaReport.html>. [Último acceso: 26 Enero 2014].
- [2] B. R. A. Latina, «Perfil,» Business Reviews, 16 Septiembre 2014. [En línea]. Available: <http://www.businessreviewamericalatina.com/CAMANICA-Grupo-Pescanova/profiles/65/Excelencia-del-mar-para-el-consumidor-global>. [Último acceso: 26 Enero 2014].
- [3] G. Pescanova, «Quienes Somos,» Grupo Pescanova, [En línea]. Available: <http://www.pescanova.com/ES/content/QUIENES-SOMOS>. [Último acceso: 26 Enero 2014].
- [4] R. Castrejón, «Business Review America Latina,» Septiembre 2014. [En línea]. Available: http://issuu.com/businessreviewamericalatina/docs/bro-bral_camanica_septiembre2014_sp. [Último acceso: 22 Julio 2015].
- [5] J. Cuéllar-Anjel, C. Lara, V. Morales, A. De Garcia y O. García Suárez, «Organización Mundial de Sanidad Animal,» Julio 2010. [En línea]. Available: http://www.rr-americas.oie.int/actualizaciones%20enero_11/Manual%20de%20Buenas%20Pr%C3%A1cticas%20en%20Camarones%20OIRSA-OSPESCA%20-%202010.pdf. [Último acceso: 4 Febrero 2015].
- [6] C. Colindres, «Biblioteca Wilson Popenoe, Zamorano,» Diciembre 2009. [En línea]. Available: <http://bdigital.zamorano.edu/bitstream/11036/60/1/T2749.pdf>. [Último acceso: 04 Febrero 2015].
- [7] J. Vilar, «ESTADÍSTICA 2,» Universidade da Coruña, 2015. [En línea]. Available: http://www.udc.es/dep/mate/estadistica2/sec6_3.html. [Último acceso: 22 Julio 2015].
- [8] M. E. Ferre Jaén, «FEIR 40: Modelos de Regresión,» Universidad de Murcia, 2015. [En línea]. Available: <http://www.um.es/ae/FEIR/40/>. [Último acceso: 22 Julio 2015].
- [9] Departamento de Sociología IV, «Análisis de regresión lineal,» Universidad Complutense Madrid, 2015. [En línea]. Available: http://pendientedemigracion.ucm.es/info/socivmyt/paginas/D_departamento/materiales/analisis_datosyMultivariable/18reglin_SPSS.pdf. [Último acceso: 22 Julio 2015].

Bibliografía

- [10] J. Ruiz Velazco, «Repositorio Digital Institucional (RDI),» Instituto Politecnico Nacional, Mayo 2011. [En línea]. Available: <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/13135/ruizv2.pdf?sequence=1>. [Último acceso: 04 Febrero 2015].
- [11] C. Maker, «Case Maker,» Case Maker, [En línea]. Available: http://www.casemaker.com/download/products/totem/rad_wp.pdf. [Último acceso: 26 Enero 2014].
- [12] FAO, «Food And Agriculture Orginazation,» United Nations, 1978. [En línea]. Available: <http://www.fao.org/docrep/l8156s/l8156s00.htm>. [Último acceso: 22 Julio 2015].
- [13] P. Kungvankij y T. Chua, «Food And Agriculture Orginazation,» United Nations, Junio 1986. [En línea]. Available: <http://www.fao.org/docrep/field/003/ac210e/AC210E00.htm>. [Último acceso: 22 Julio 2015].
- [14] F. P. Veracruz, «Fundación Produce Veracruz,» [En línea]. Available: <http://www.funprover.org/formatos/cursos/Manual%20Buenas%20Practicass%20Acuicolas.pdf>.
- [15] «Planet Inverts,» Alpha Pro Breeders, 2015. [En línea]. Available: http://www.planetinverts.com/shrimp_species.html. [Último acceso: 22 Julio 2015].
- [16] «Shrimp Academy - Types of Shrimps,» American Shrimp Processors Association., 2015. [En línea]. Available: <http://www.americanshrimp.com/about-our-shrimp/shrimp-academy/>. [Último acceso: 22 Julio 2015].
- [17] «Shrimp Species,» Pet Shrimp, 2015. [En línea]. Available: <http://www.petshrimp.com/shrimpspecies.php>. [Último acceso: 22 Julio 2015].
- [18] «Shrimp Poster,» Centre for Marine Biodiversity, 2015. [En línea]. Available: <http://www.marinebiodiversity.ca/cmb/resources/posters-and-identification-guides/CMB%20Sponsored%20Posters/shrimps-of-the-estuary-and-the-northern-gulf-of-st-lawrence/shrimp.jpg/view>. [Último acceso: 22 Julio 2015].
- [19] FAO - Fisheries and Aquaculture Department, «Species Fact Sheets - Penaeus vannamei (Boone, 1931),» 2015. [En línea]. Available: <http://www.fao.org/fishery/species/3404/en>. [Último acceso: 22 Julio 2015].
- [20] FAO - Fisheries and Aquaculture Department, «Cultured Aquatic Species Information Programme - Penaeus vannamei (Boone, 1931),» 2015. [En línea]. Available:

Bibliografía

- http://www.fao.org/fishery/culturedspecies/Penaeus_vannamei/en. [Último acceso: 22 Julio 2015].
- [21] Integrated Taxonomic Information System, «Litopenaeus vannamei (Boone, 1931): Taxonomic Serial No.: 551682,» 2015. [En línea]. Available: http://www.itis.gov/servlet/SingleRpt/SingleRpt?search_topic=TSN&search_value=551682. [Último acceso: 22 Julio 2015].
- [22] U.S. Fish & Wildlife Service, «Hagerman National Fish Hatchery,» 2015. [En línea]. Available: <http://www.fws.gov/hagerman/Tour/raceways.html>. [Último acceso: 22 Julio 2015].
- [23] Conapesca, «Conapesca,» 30 Noviembre 2011. [En línea]. Available: <http://www.conapesca.sagarpa.gob.mx/work/sites/cona/dgof/publicaciones/GuiaEmpresarialTilapia.pdf>. [Último acceso: 22 Julio 2015].
- [24] Real Academia Española, «Diccionario de la lengua española,» 2015. [En línea]. Available: <http://www.rae.es/>. [Último acceso: 22 Julio 2015].
- [25] Fondriest Environmental, «Fundamentals of Environmental Measurements,» 2015. [En línea]. Available: <http://www.fondriest.com/environmental-measurements/parameters/water-quality/>. [Último acceso: 22 Julio 2015].
- [26] PRTR España, «Registro Estatal de Emisiones y Fuentes Contaminantes,» 2015. [En línea]. Available: <http://www.prtr-es.es/NH3-amoniaco,15593,11,2007.html>. [Último acceso: 22 Julio 2015].
- [27] Ambientum.com, «DETERMINACIÓN DE CLORURO,» Grupo de Tratamiento de Aguas Residuales. Escuela Universitaria Politécnica. Universidad de Sevilla., 2015. [En línea]. Available: http://www.ambientum.com/enciclopedia_medioambiental/aguas/Determinacion_de_cloruro.asp#. [Último acceso: 22 Julio 2015].
- [28] Comisión Nacional del Agua - México, «ANÁLISIS DE AGUA - DETERMINACIÓN DE CLORUROS,» 2001. [En línea]. Available: <http://www.conagua.gob.mx/CONAGUA07/Noticias/NMX-AA-073-SCFI-2001.pdf>. [Último acceso: 22 Julio 2015].
- [29] G. Goyenola, «Asociación Civil Investigación y Desarrollo,» Junio 2007. [En línea]. Available: http://imasd.fcien.edu.uy/difusion/educamb/propuestas/red/curso_2007/cartillas/tematicas/nitrato_nitrito.pdf. [Último acceso: 22 Julio 2015].

Bibliografía

- [30] A. M. Lasso Palacios, «Instituto de Hidrología, Meteorología y Estudios Ambientales - Colombia,» Febrero 2009. [En línea]. Available: <http://www.ideam.gov.co/documents/14691/38155/Nitrito+en+agua+por+Espectrofotometr%C3%ADa.pdf/4775634c-c6ba-4c95-8e98-0696ace02c03>. [Último acceso: 22 Julio 2015].
- [31] A. D. Massol, «Departamento de Biología - Universidad de Puerto Rico,» [En línea]. Available: <http://www.uprm.edu/biology/profs/massol/manual/p2-alcinidad.pdf>. [Último acceso: 22 Julio 2015].
- [32] Lenntech, «Water Treatment,» [En línea]. Available: <http://www.lenntech.es/>. [Último acceso: 22 Julio 2015].
- [33] Universidad de Salamanca - Departamento de Microbiología y Genética, «Recuento de Coliformes Totales - Filtración a través de membrana,» [En línea]. Available: http://virus.usal.es/Web/demo_fundacua/demo2/FiltraMembColiT_auto.html. [Último acceso: 22 Julio 2015].
- [34] E. U. Jiménez, «Universitat de Valencia,» 2015. [En línea]. Available: <http://www.uv.es/uriel/material/Morelisi.pdf>. [Último acceso: 22 Julio 2015].
- [35] C. S. Park, Fundamentos de ingeniería económica, México: PEARSON EDUCACIÓN, 2009.
- [36] G. Molina y M. F. Rodrigo, «Ciencias de la Salud - Universitat de Valencia,» 2010. [En línea]. Available: http://ocw.uv.es/ciencias-de-la-salud/pruebas-1/1-3/t_09nuevo.pdf. [Último acceso: 22 Julio 2015].
- [37] V. Aela Riera, M. Guillén Estany, C. Muñoz Vaquer, I. Maqueda de Anta, E. Torrelles Puig y N. Viladomiu Canela, «SPSS - Análisis de Datos,» 1 Marzo 2005. [En línea]. Available: http://www.ub.edu/aplica_infor/spss/cap3-7.htm. [Último acceso: 22 Julio 2015].
- [38] Dirección Nacional de Innovación Académica, «Métodos de Regresión,» Universidad Nacional de Colombia, 2015. [En línea]. Available: http://www.virtual.unal.edu.co/cursos/ciencias/2007315/html/un1/cont_01_01.html. [Último acceso: 22 Julio 2015].
- [39] S. García Gámez, «Departamento de Economía Aplicada - Universidad Autónoma de Madrid,» 2015. [En línea]. Available: https://www.uam.es/personal_pdi/economicas/sgarcia/regresionlinealsimple.pdf. [Último acceso: 22 Julio 2015].

Bibliografía

- [40] D. C. Stone y J. Ellis, «The Correlation Coefficient - Department of Chemistry, University of Toronto,» 29 Septiembre 2011. [En línea]. Available: <http://www.chem.utoronto.ca/coursenotes/analsci/stats/CorrCoeff.html>. [Último acceso: 22 Julio 2015].
- [41] A. Rofail and T. Martin, Building N-Tier Applications with COM and Visual Basic 6.0, Danvers: John Wiley & Sons, Inc., 1999.
- [42] J. Meier, D. Hill, A. Homer, J. Taylor, P. Bansode, L. Wall, R. Boucher Jr. y A. Bogawat, «Microsoft Application Architecture Guide, 2nd Edition,» Microsoft Corporation, Octubre 2009. [En línea]. Available: <https://msdn.microsoft.com/en-us/library/ff650706.aspx>. [Último acceso: 22 Julio 2015].
- [43] F. J. Ceballos Sierra, Programación orientada a objetos con C++, 2da edición., México: Alfaomega Grupo Editor, S.A. de C.V., 1998.
- [44] I. Horton, Beginning Visual C++ 5, Birmingham: Wrox Press, 1997.
- [45] E. Horowitz, S. Sahni and D. Mehta, Fundamentals of data structures in C++, New York: W. H. Freeman and Company, 1995.
- [46] Oracle, "The Java™ Tutorials," Oracle, 2015. [Online]. Available: <http://docs.oracle.com/javase/tutorial/java/concepts/index.html>. [Accessed 22 Julio 2015].
- [47] Microsoft Corporation, «Object-Oriented Programming (C# and Visual Basic),» Microsoft Corporation, 2015. [En línea]. Available: <https://msdn.microsoft.com/en-us/library/dd460654.aspx>. [Último acceso: 22 Julio 2015].
- [48] Microsoft Corporation, "Object-Oriented Programming (C# and Visual Basic)," Microsoft Corporation, 2015. [Online]. Available: <https://msdn.microsoft.com/en-us/library/dd460654.aspx>. [Accessed 22 Julio 2015].
- [49] Microsoft Corporation, «Visual Studio,» Microsoft Corporation, 2015. [En línea]. Available: <https://www.visualstudio.com/>. [Último acceso: 22 Julio 2015].
- [50] M. Wasson, «Integrated Windows Authentication : ASP.Net,» Microsoft Corporation, 18 Diciembre 2012. [En línea]. Available: <http://www.asp.net/web-api/overview/security/integrated-windows-authentication>. [Último acceso: 22 Julio 2015].
- [51] Microsoft Corporation, «Microsoft NTLM,» Microsoft Corporation, 2015. [En línea]. Available: <https://msdn.microsoft.com/en->

Bibliografía

- us/library/windows/desktop/aa378749(v=vs.85).aspx. [Último acceso: 22 Julio 2015].
- [52] Microsoft Corporation, "Explained: Windows Authentication in ASP.NET 2.0," Microsoft Patterns and Practices, Noviembre 2005. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ff647076.aspx>. [Accessed 22 Julio 2015].
- [53] Microsoft Corporation, «Microsoft Kerberos,» Microsoft Corporation, 2015. [En línea]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/aa378747\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa378747(v=vs.85).aspx). [Último acceso: 22 Julio 2015].
- [54] Oracle, "The Java EE 6 Tutorial," Enero 2013. [Online]. Available: <https://docs.oracle.com/javasee/6/tutorial/doc/jvaeetutorial6.pdf>. [Accessed 22 Julio 2015].
- [55] D. K. Barry, "Service Architecture," Barry & Associates, Inc., 2015. [Online]. Available: http://www.service-architecture.com/articles/web-services/web_services_explained.html. [Accessed 22 Julio 2015].
- [56] IBM, «IBM Developer Works: SOA and Web Services,» IBM, 2015. [En línea]. Available: <http://www.ibm.com/developerworks/library/ws-devaxis2part1/>. [Último acceso: 22 Julio 2015].
- [57] C. Pérez López, Domine Microsoft SQL Server 2008, México: Alfaomega Grupo Editor, S.A. de C.V., 2010.
- [58] Intuit Quickbase, "A Timeline of Database History," Intuit Quickbase, [Online]. Available: <http://quickbase.intuit.com/articles/timeline-of-database-history>. [Accessed 22 Julio 2015].
- [59] E. F. Codd, "University of Pennsylvania," Junio 1970. [Online]. Available: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>. [Accessed 22 Julio 2015].
- [60] B. Grad y T. J. Bergin, «History of Database Management Systems,» Project Muse, 2009. [En línea]. Available: https://muse.jhu.edu/journals/ieee_annals_of_the_history_of_computing/summary/v031/31.4.grad.html. [Último acceso: 22 Julio 2015].
- [61] Standford University, «Oracle® Database SQL Reference 10g Release 1 (10.1) - History of SQL,» Standfor University, 2003. [En línea]. Available: <https://web.stanford.edu/dept/itss/docs/oracle/10g/server.101/b10759/intro001.htm>. [Último acceso: 22 Julio 2015].

Bibliografía

- [62] Informit - Pearson, "SQL Definition and History," Pearson, [Online]. Available: http://www.informit.com/library/content.aspx?b=STY_Sql_24hours&seqNum=9. [Accessed 22 Julio 2015].
- [63] Microsoft Corporation, "Description of the database normalization basics," Microsoft Corporation, 12 Julio 2013. [Online]. Available: <https://support.microsoft.com/en-us/kb/283878>. [Accessed 22 Julio 2015].
- [64] J. Chomicki, "University at Buffalo The State University of New York," [Online]. Available: <http://www.cse.buffalo.edu/~chomicki/562/design.pdf>. [Accessed 22 Julio 2015].
- [65] P. D. Chen y J. Blakeley, «Dr. Peter Chen: Entity Relationship Model - Past, Present and Future,» Channel 9 - MSDN, 5 Abril 2007. [En línea]. Available: <https://channel9.msdn.com/Shows/Going+Deep/Dr-Peter-Chen-Entity-Relationship-Model-Past-Present-and-Future>. [Último acceso: 22 Julio 2015].
- [66] D. Schlichting, "What is SQL Server," Database Journal, 5 Septiembre 2008. [Online]. Available: <http://www.databasejournal.com/features/mssql/article.php/3769211/What-is-SQL-Server.htm>. [Accessed 22 Julio 2015].
- [67] Microsoft TechNet, "Books Online for SQL Server 2012," Microsoft Corporation, [Online]. Available: [https://technet.microsoft.com/en-us/library/ms130214\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/ms130214(v=sql.110).aspx). [Accessed 22 Julio 2015].
- [68] A. Pennington, PROFIT: Discover the gold mine in your IT department., Arlington Heights, IL 60004: efg Publishing, Inc., 2007.
- [69] A. Begel y N. Nagappan, «Microsoft Research,» [En línea]. Available: <http://research.microsoft.com/pubs/56015/AgileDevatMS-ESEM07.pdf>. [Último acceso: 04 Febrero 2015].
- [70] I. Casemaker, «Totem, Rapid Application Development,» 2000. [En línea]. Available: http://www.casemaker.com/download/products/totem/rad_wp.pdf. [Último acceso: 04 Febrero 2015].
- [71] S. Liang, «Sheldon Liang,» Azusa Pacific University, 1 Octubre 2009. [En línea]. Available: <http://cs-linux.cs.apu.edu/~sliang/PDF/CS470-RAD.pdf>. [Último acceso: 26 Enero 2014].
- [72] W. Anderson, «Google Books: Unwrapping the CIO,» 2006. [En línea]. Available: https://books.google.com.ni/books?id=H_NdxrCYef0C&printsec=frontcover#v=onepage&q

Bibliografía

&f=false. [Último acceso: 04 Febrero 2015].

- [73] FTMS College, [En línea]. Available: <http://www.ftms.edu.my/images/Document/IMM006%20-%20RAPID%20APPLICATION%20DEVELOPMENT/Chapter%20note.pdf>. [Último acceso: 04 Febrero 2015].
- [74] A. Dennis, B. H. Wixom y R. M. Roth, «Course Materials: Saigon Institute of Technology,» 09 Junio 2014. [En línea]. Available: http://www.saigontech.edu.vn/faculty/huynq/SAD/Systems_Analysis_Design_UML_5th%20ed.pdf. [Último acceso: 04 Febrero 2015].
- [75] P. Konstantinou, «Systems Analysis & Design: University of Houston Clear Lake,» [En línea]. Available: <https://mis.uhcl.edu/ROB/Course/SAD/Student%20Papers/RAD%20PAPER.doc>. [Último acceso: 04 Febrero 2015].
- [76] S. McConnell, «Intranet: International University of East Africa,» 1996. [En línea]. Available: <http://iclass.iuea.ac.ug/intranet/E-books/INFORMATION%20TECHNOLOGY/SOFTWARE%20ENGINEERING/Steve%20McConnell%20-%20Rapid%20Development%20-%20Taming%20Wild%20Software%20Schedules.pdf>. [Último acceso: 04 Febrero 2015].
- [77] CodePlex, «GMap.NET - Great Maps for Windows Forms & Presentation,» 1 Abril 2015. [En línea]. Available: <http://greatmaps.codeplex.com/>. [Último acceso: 22 Julio 2015].
- [78] C. Schmelkes, Manual para la presentación de anteproyectos e informes de investigación, México, 2005.
- [79] IBM, «Unified Modeling Language: IBM,» IBM, [En línea]. Available: <http://www-01.ibm.com/software/rational/uml/>. [Último acceso: 04 Febrero 2015].
- [80] OMG, «What is UML?: OMG,» OMG, Julio 2015. [En línea]. Available: http://www.omg.org/gettingstarted/what_is_uml.htm. [Último acceso: 04 Febrero 2015].
- [81] R. Hernández Sampieri, C. Fernández Collado y M. d. P. Baptista Lucio, Metodología de la Investigación, México D.F.: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V., 2010.
- [82] R. S. Pressman, Ingeniería del Software - Un enfoque práctico, México: McGraw Hill, 2000.
- [83] R. Casero, M. Cesarini y M. Monga, «Managing Code Dependencies in C# , in Journal of

Anexos

Object Technology, vol. 3, no. 2, pages 47–55,» 2003. [En línea]. Available: http://www.jot.fm/issues/issue_2004_02/article5. [Último acceso: 22 Julio 2015].

Anexos

Anexo 1: Cronograma de actividades

	Task Name	Duration	Work	Start	Finish	Predecessors
1	Sistema de Información Técnica	119 days	0 hrs	2/9/2015	7/23/2015	
2	Diseñado el flujo de proceso de la recolección de la información técnica y su frecuencia	3 days	0 hrs	2/9/2015	2/11/2015	
3	Entrevista con supervisores y jefes de granjas	1 day	0 hrs	2/9/2015	2/9/2015	
4	Elaboración del Flujo de Proceso	1 day	0 hrs	2/10/2015	2/10/2015	3
5	Aprobación del Flujo de Proceso	1 day	0 hrs	2/11/2015	2/11/2015	4
6	Elaborado el análisis de requerimientos	3 days	0 hrs	2/12/2015	2/16/2015	
7	Conformar el equipo de trabajo para las JAD	1 day	0 hrs	2/12/2015	2/12/2015	5
8	Definir del Hardware y Software necesarios	1 day	0 hrs	2/13/2015	2/13/2015	7
9	Entrevista con supervisores y jefes de granjas para recopilar los requerimientos	1 day	0 hrs	2/16/2015	2/16/2015	8
10	Elaborado el diagrama entidad relación	2 days	0 hrs	2/17/2015	2/18/2015	
11	Elaborar el diagrama ER utilizando herramienta de Modelo de Datos	1 day	0 hrs	2/17/2015	2/17/2015	9
12	Crear Modelo ER en SGBD	1 day	0 hrs	2/18/2015	2/18/2015	11
13	Elaborado el diseño arquitectónico	8 days	0 hrs	2/19/2015	3/2/2015	
14	Elaborar un prototipo de interfaz de usuario utilizando Visual Studio	7 days	0 hrs	2/19/2015	2/27/2015	12
15	Reunión JAD para presentar prototipo y obtener retroalimentación y/o requerimientos nuevos	1 day	0 hrs	3/2/2015	3/2/2015	14
16	Desarrollado el sistema	87 days	0 hrs	3/3/2015	7/1/2015	
17	Codificar prototipo de sistema	86 days	0 hrs	3/3/2015	6/30/2015	15
18	Reunión JAD para presentar prototipo y obtener retroalimentación y/o requerimientos nuevos	1 day	0 hrs	7/1/2015	7/1/2015	17
19	Implementado el sistema	9 days	0 hrs	7/2/2015	7/14/2015	
20	Planificar la instalación en los equipos de los usuarios y servidor de producción	1 day	0 hrs	7/2/2015	7/2/2015	18
21	Instalación de servidor de producción	4 days	0 hrs	7/3/2015	7/8/2015	20
22	Instalación en equipos de usuarios	4 days	0 hrs	7/9/2015	7/14/2015	21
23	Capacitado el personal de la empresa para uso del sistema	7 days	0 hrs	7/15/2015	7/23/2015	
24	Planificar capacitación a los usuarios	1 day	0 hrs	7/15/2015	7/15/2015	22
25	Capacitación de usuarios	6 days	0 hrs	7/16/2015	7/23/2015	24

Figura 72: Cronograma de actividades

Anexos

Anexo 2: Requerimientos del sistema

Definición, Acrónimos y abreviaturas

Sistema

Es un conjunto de componentes que interactúan entre sí para alcanzar un objetivo en común.

Sistemas Informáticos

Es un sistema cuyo principal objetivo es procesar información, almacenarla, producir información y generar reportes.

Métodos

Es el conjunto de actividades y tareas a seguir para completar las etapas del ciclo de vida del sistema informático.

Análisis de Sistemas

Es el proceso de clasificación e interpretación de los hechos, diagnósticos de problemas y empleo de la información para recomendar mejoras al sistema.

Análisis estructurado

Se utiliza para que las personas observen los elementos lógicos separados de los elementos físicos.

Ciclo de vida de un sistema informático

Es un conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implementar un sistema de información.

Especificaciones de Requisitos de Software (ERS)

Es el conjunto de requisitos que deben ser alcanzados o cumplimentados por un producto o proceso.

Diagrama Entidad Relación (ER)

Es un conjunto de objetos básicos llamados entidades y la relación existente entre ellos. Una entidad es un conjunto per se diferenciable y atribuible del resto del conjunto. La distinción se alcanza cuando se asocia a una entidad un conjunto de características o atributos que lo definen.

Diseño de sistemas

Es el proceso de planificar, reemplazar o complementar un sistema organizacional existente.

Diseño arquitectónico

Es una estructura de programa modular y representa las relaciones de control entre los módulos. Define las interfaces que facilitan el flujo de datos a lo largo del sistema.

Anexos

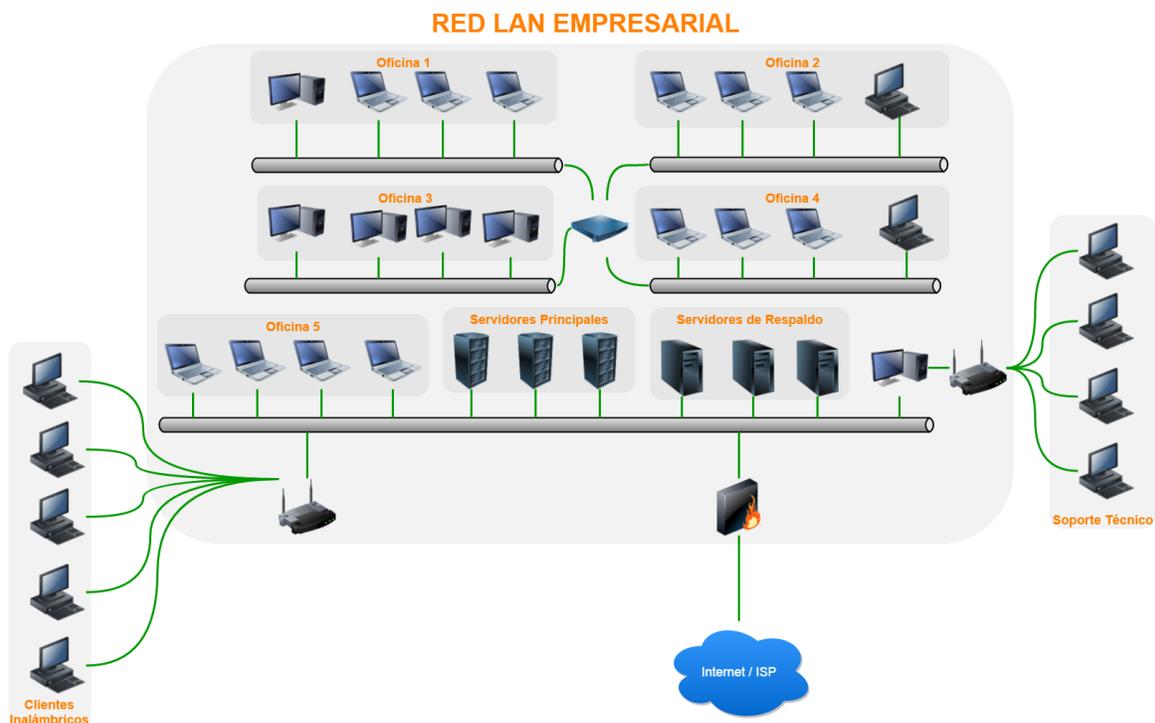
Red

Es un grupo de computadoras y/o dispositivos que comparten un enlace físico o inalámbrico de comunicaciones entre sí. Estas se clasifican según el alcance o cobertura en:

- PAN (Personal Area Network) Red de área personal.
- LAN (Local Area Network) Red de área local.
- WAN (Wide Area Network) Red de área amplia.
- MAN (Metropolitan Area Network) Red de área metropolitana.

LAN (Local Area Network)

Es un grupo de computadoras y/o dispositivos que comparten un enlace físico o inalámbrico de comunicaciones entre sí usualmente concentrados en una oficina, edificio o casa.



Clientes

Son computadoras de escritorio o portátiles que se conectan normalmente a una red de área local (LAN) y utilizan los servicios y recursos brindados por los servidores de la red.

Servidores

Son computadoras que prestan servicios y recursos a equipos clientes dentro de una red de área local (LAN).

Anexos

Ciente - Servidor

Sistemas o aplicaciones instalados en equipos o computadoras clientes conectados a su vez en una red; generalmente LAN, y que utilizan los recursos y/o servicios prestados por un servidor.

Alcance Funcional del Sistema

Generalidades

El sistema será desarrollado utilizando una arquitectura Cliente-Servidor de tres capas con el paradigma de Programación Orientada a Objetos. Será ejecutado en una red empresarial y será una aplicación de escritorio consistente con las plataformas de aplicaciones existentes en la empresa.

Alcance Funcional

El sistema tendrá las siguientes características generales. Será referenciado o conocido desde este momento como SITCC (Sistema de Información Técnica del Cultivo del Camarón):

- Gestionar la seguridad de roles o grupos de acceso para usuarios.
- Gestionar el registro de usuarios y asociarlos con los roles o grupos de acceso.
- Gestionar la información de las granjas operativas de la empresa.
- Gestionar la información de las pilas que pertenecen a las granjas de la empresa.
- Gestionar los ciclos productivos de las pilas.
- Gestionar los tipos de pilas.
- Gestionar los tipos de parámetros.
- Gestionar las unidades de medida utilizadas por los parámetros.
- Gestionar los parámetros medidos en las diferentes pilas para monitorear la calidad del agua de las mismas.
- Gestionar las lecturas de los parámetros obtenidos de las pilas.
- Generar reportes sobre la información registrada en el sistema.
- Generar gráficos utilizando las lecturas de los parámetros obtenidos de las pilas.
- Exportar las lecturas de los parámetros en archivo de texto (CSV).

Características de los usuarios

Los usuarios que utilizarán el sistema es el personal técnico laborando en las granjas y poseen conocimientos intermedios de la operación de sistemas.

Requerimientos

Aquí se definen los requerimientos del sistema que servirán como guía para el desarrollador para lograr que el sistema resultante se adhiera a éstos requerimientos.

Interfaz de usuario

El patrón de diseño utilizado para la interfaz de usuario es el de una aplicación MDI conocida como interfaz de documentos múltiples. En este diseño, la aplicación se ejecuta como un formulario

Anexos

principal que contiene las opciones disponibles del sistema a través de barra de menú y cada opción se abre como un formulario hijo del formulario principal.

Utilizando este diseño se permite trabajar con varias opciones desde la misma aplicación, ahorrándole tiempo al usuario. De igual manera, se utilizarán etiquetas bien definidas y expresadas con lenguaje coloquial concluyente siempre que sea posible. En caso contrario, se utilizará términos técnicos estrictamente relacionados al cultivo del camarón.

Los mensajes de errores serán mostrados en ventanas emergentes utilizando un lenguaje claro y sencillo donde se establece la causa específica del error. De igual manera, cualquier mensaje de confirmación de una operación o acción será mostrado utilizando la misma técnica descrita anteriormente.

Requerimientos Funcionales

Tabla 8: Requerimientos del sistema

Opciones	Código	Descripción
Seguridad	RF01	El usuario administrador podrá acceder al menú seguridad y configurar los roles que se utilizarán en el sistema.
	RF02	El usuario administrador podrá configurar los usuarios que utilizarán el sistema y asociará a cada usuario a un rol determinado.
	RF03	El usuario administrador podrá configurar las opciones del sistema.
	RF04	El usuario administrador podrá configurar la asociación entre los roles y opciones del sistema.
	RF05	El usuario administrador podrá configurar la cadena de conexión que utilizará el sistema.
Configuración	RF06	Un usuario con permiso a la opción de configuración podrá gestionar la información de las granjas de la empresa.
	RF07	Un usuario con permiso a la opción de configuración podrá gestionar la información de las pilas y las asociará a una granja de la empresa.
	RF08	Un usuario con permiso a la opción de configuración podrá gestionar la información de los ciclos de producción de las diferentes pilas de la empresa.
	RF09	Un usuario con permiso a la opción de configuración podrá gestionar la información de las semanas de un año calendario.
	RF10	Un usuario con permiso a la opción de

Anexos

		configuración podrá gestionar la información de los tipos de pilas utilizadas por las pilas.
	RF11	Un usuario con permiso a la opción de configuración podrá gestionar la información de las unidades de medidas utilizadas por los parámetros.
	RF12	Un usuario con permiso a la opción de configuración podrá gestionar la información de los tipos de parámetros utilizados por los parámetros.
	RF13	Un usuario con permiso a la opción de configuración podrá gestionar la información de los parámetros que se monitorearan en el sistema.
Proceso	RF14	Un usuario con permiso a la opción de proceso podrá gestionar la información de las lecturas obtenidas para los parámetros medidos en un ciclo productivo de una pila.
	RF15	Un usuario con permiso a la opción de proceso podrá visualizar las gráficas de las lecturas registradas de los parámetros de las pilas.

Requerimientos no Funcionales

Seguridad

- El sistema permitirá el acceso solamente a usuarios del dominio configurados por el administrador.
- El sistema no almacenará ninguna contraseña de usuario, utilizará el mecanismo de autenticación integrada de Windows.
- Se implementará de forma granular los permisos sobre las opciones a través de la asociación de opciones del sistema por rol.
- El mecanismo de autenticación integrada de Windows garantiza una comunicación segura entre el cliente y servidor.
- Se registra la bitácora de cambios en cada entidad del sistema. Permitiendo saber los cambios realizados a nivel de registro.
- Las copias de seguridad de la base de datos del sistema se registrarán por los procedimientos de respaldos establecidos por la empresa para las el servidor SQL.

Disponibilidad

El sistema estará disponible las 24 horas del día los 7 días de la semana; bajo la premisa que el servidor de dominio y SQL se encuentran 100% funcionales.

Anexos

Portabilidad

Solamente a través de sistemas operativos de Windows; como medida de estandarización de las aplicaciones de la empresa, deberá ser ejecutable para el sistema operativo Windows 7 o mayor.

Anexo 3: Análisis y diseño del sistema

Arquitectura del sistema

El sistema fue desarrollado utilizando el paradigma de Programación Orientada a Objetos, utilizando una arquitectura Cliente / Servidor de tres capas.

Módulos/Secciones del sistema

El sistema se divide en tres secciones:

Procesos

Esta sección es la encargada de gestionar la información sobre las lecturas de los parámetros obtenidos de las pilas de las granjas de la empresa. De igual manera, se puede visualizar gráficamente esa información y permite la exportación de la misma en archivo de texto con formato CSV.

Configuración

Esta sección es la que se utiliza para gestionar toda la información general o catálogos que utiliza el sistema.

Seguridad

Esta sección es utilizada por el administrador para gestionar los usuarios que podrán acceder al sistema así como los roles y opciones del sistema disponible para ellos.

Anexos

Diagramas de casos de uso

Seguridad

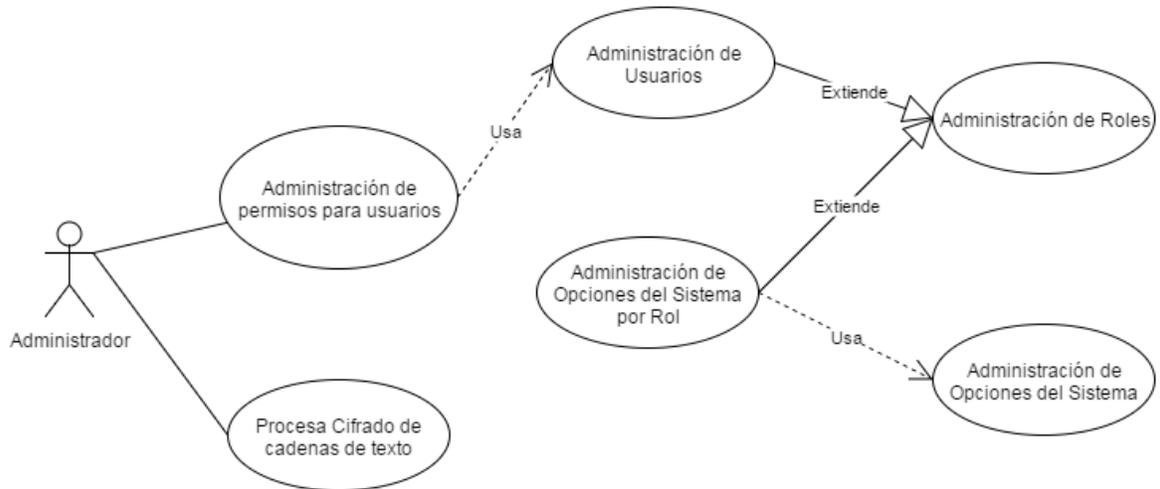


Figura 73: Diagrama de caso de uso de Seguridad

Configuración

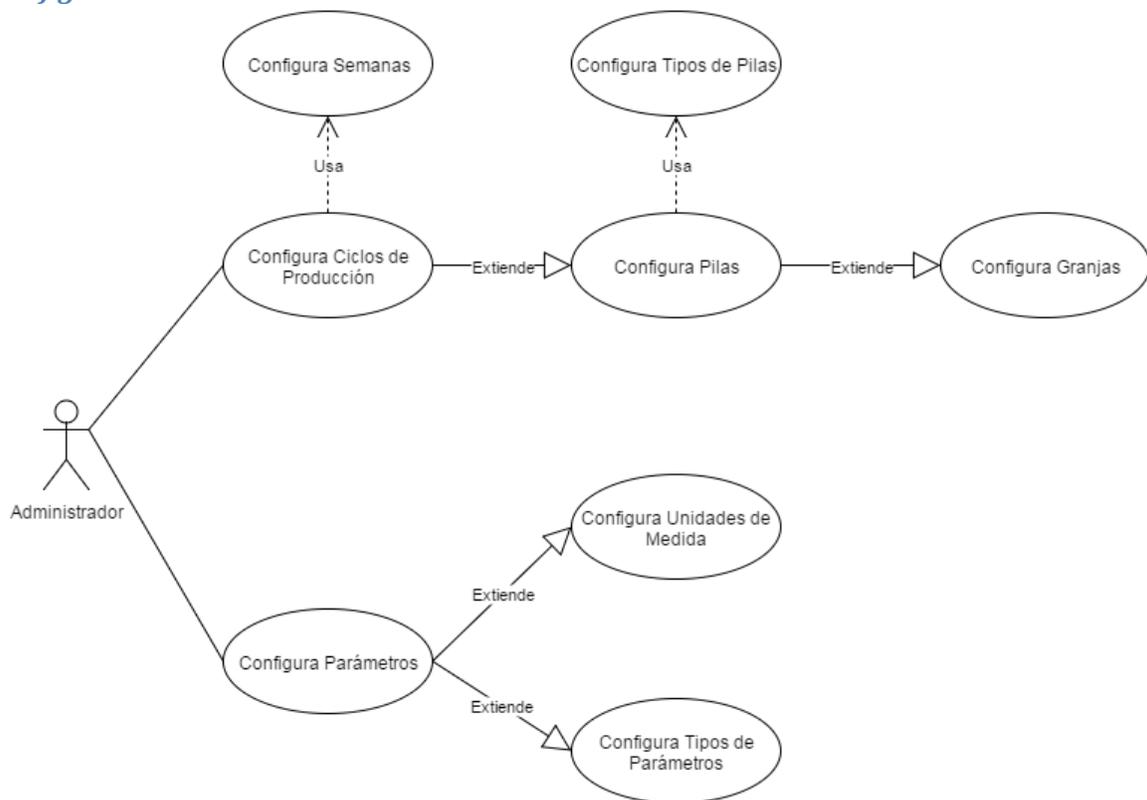


Figura 74: Diagrama de caso de uso de Configuración

Anexos

Proceso

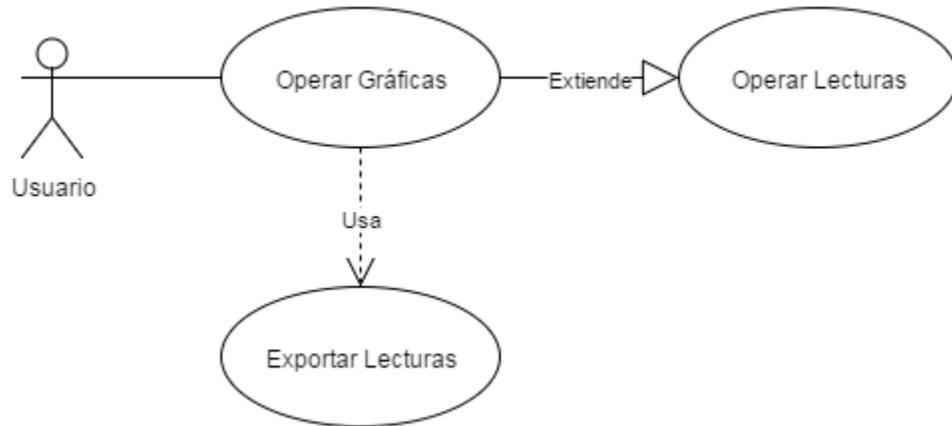


Figura 75: Diagrama de caso de uso de Proceso

Anexos

Anexo 4: Desarrollo del sistema

Arquitectura cliente - servidor

El sistema se desarrolló utilizando una arquitectura cliente – servidor. Para ello, se creó una solución de escritorio utilizando lenguaje de programación C# dividida en cuatro proyectos.

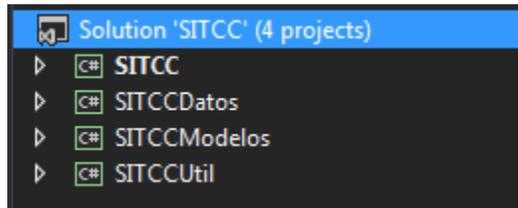


Figura 76: Solución de escritorio

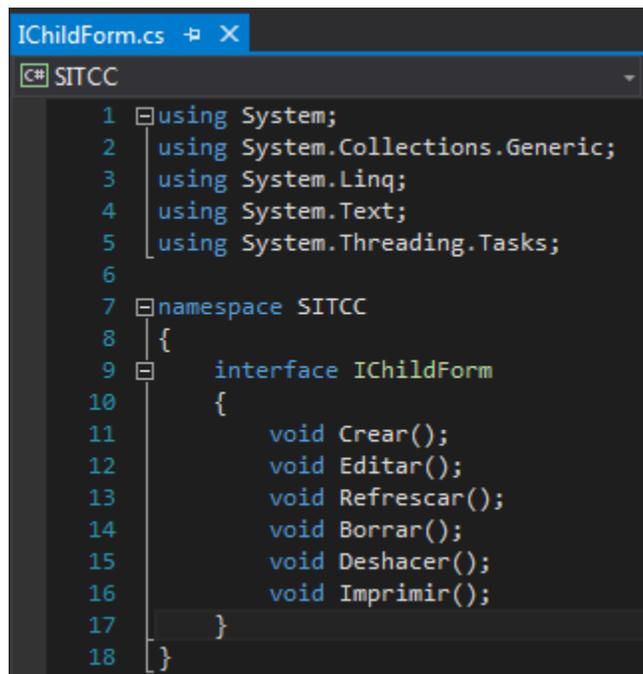
Estos proyectos dividen el sistema en capas lógicas que son reutilizables y que pueden dar servicios a otros sistemas que hagan referencias a ellas.

Anexos

Proyecto de Aplicación de Windows- SITCC

Este proyecto contiene todos los formularios que conforman el sistema. Una vez compilado, el resultado final es un archivo de tipo ejecutable que corre en un sistema operativo Windows. Éste constituye la interfaz gráfica con la cual el usuario interactúa para la gestión de la información.

Dicho proyecto, consta de un formulario principal y formularios hijos que representan cada uno de las opciones del sistema. También, consta de una “interface” que se utiliza para heredar cierta funcionalidad a todos los formularios hijos y garantizar una homogeneidad de la funcionalidad.



```
1 using System;
2     using System.Collections.Generic;
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7 namespace SITCC
8 {
9     interface IChildForm
10    {
11        void Crear();
12        void Editar();
13        void Refrescar();
14        void Borrar();
15        void Deshacer();
16        void Imprimir();
17    }
18 }
```

Figura 77: Interfaz utilizada en todos los formularios hijos del sistema

A continuación ejemplo, del uso de la interfaz para garantizar la homogeneidad de la funcionalidad; usando el concepto de polimorfismo.

Anexos

```
frmGranja.cs  X
C# SITCC SITCC.frmGranja
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Data.SqlClient;
8 using System.Text;
9 using System.Windows.Forms;
10 using SITCCModelos;
11 using SITCCUtil;
12
13 namespace SITCC
14 {
15     public partial class frmGranja : Form, IChildForm
16     {
17         private clsGranja pDatos;
18         private List<clsGranja> dtTR;
19         private ListViewItem plvitem;
20         private mdiPrincipal pPrincipal;
21         private int imageIndex;
22         private long IDGranja;
23         public frmGranja()...
24
25
26         private void frmGranja_Load(object sender, EventArgs e)...
27
28
29         private void EstablecerPermisos()...
30
31
32         public void Crear()
33         {
34             try
35             {
36                 frmCrearGranja frmCTH = new frmCrearGranja();
37                 frmCTH.EsEdicion = false;
38                 frmCTH.IDGranja = 0;
39                 frmCTH.ShowDialog();
40                 this.LlenarGranjas();
41             }
42             catch (Exception ex)
43             {
44                 MessageBox.Show(ex.ToString());
45             }
46         }
47     }
48 }
```

Figura 78: Código fuente parcial de formulario para las granjas

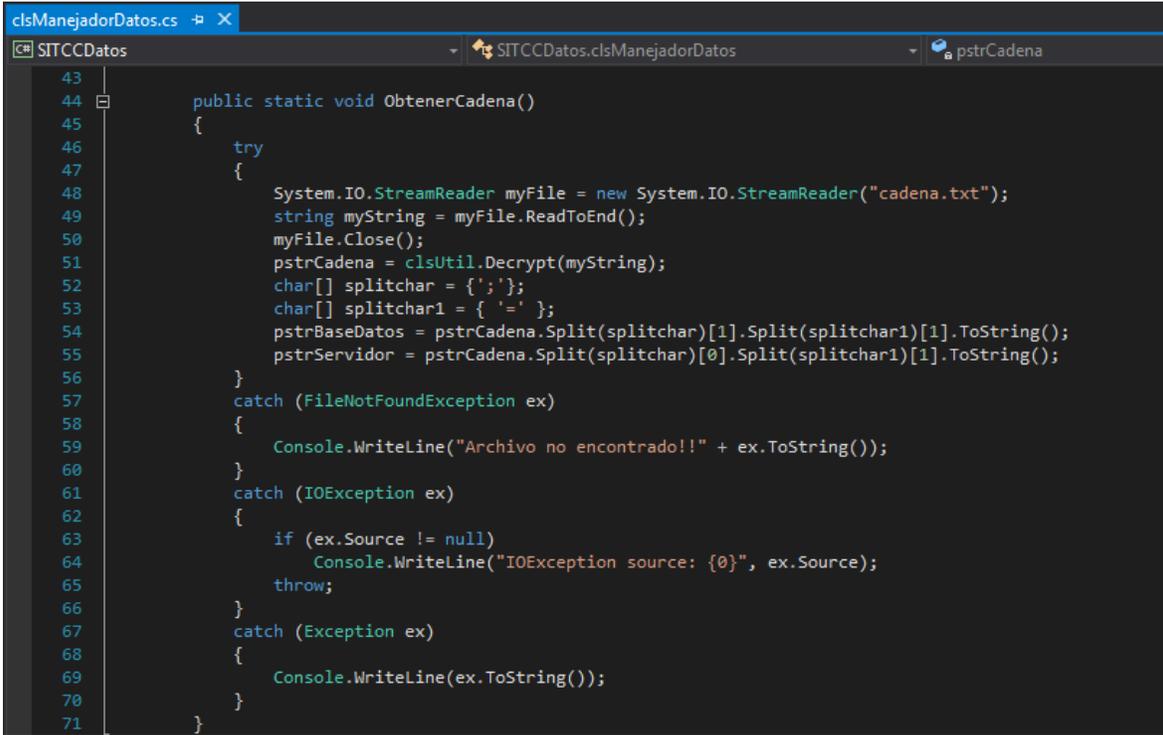
Adicionalmente, el proyecto hace referencia a dos proyectos más: Modelos y Utilitario. Estos proyectos son parte de la solución y son proyectos de librerías de clases.

Anexos

Proyecto de Librerías de clases- SITCCDatos

Es un tipo de proyecto que al compilarse se genera un archivo de tipo "Extensión de la aplicación". En el caso particular de este proyecto; éste contiene la clase que el sistema necesita para gestionar la base de datos.

El mecanismo de funcionamiento de esta librería es obtener al inicio de la aplicación la cadena de conexión con la cual se conectará a la base de datos. El usuario final de esta librería es el proyecto de Modelos de clases, que hace referencia a este recurso.



```
43
44 public static void ObtenerCadena()
45 {
46     try
47     {
48         System.IO.StreamReader myFile = new System.IO.StreamReader("cadena.txt");
49         string myString = myFile.ReadToEnd();
50         myFile.Close();
51         pstrCadena = clsUtil.Decrypt(myString);
52         char[] splitchar = {';'};
53         char[] splitchar1 = {'='};
54         pstrBaseDatos = pstrCadena.Split(splitchar)[1].Split(splitchar1)[1].ToString();
55         pstrServidor = pstrCadena.Split(splitchar)[0].Split(splitchar1)[1].ToString();
56     }
57     catch (FileNotFoundException ex)
58     {
59         Console.WriteLine("Archivo no encontrado!!" + ex.ToString());
60     }
61     catch (IOException ex)
62     {
63         if (ex.Source != null)
64             Console.WriteLine("IOException source: {0}", ex.Source);
65         throw;
66     }
67     catch (Exception ex)
68     {
69         Console.WriteLine(ex.ToString());
70     }
71 }
```

Figura 79: Código fuente parcial de la clase Manejador de datos

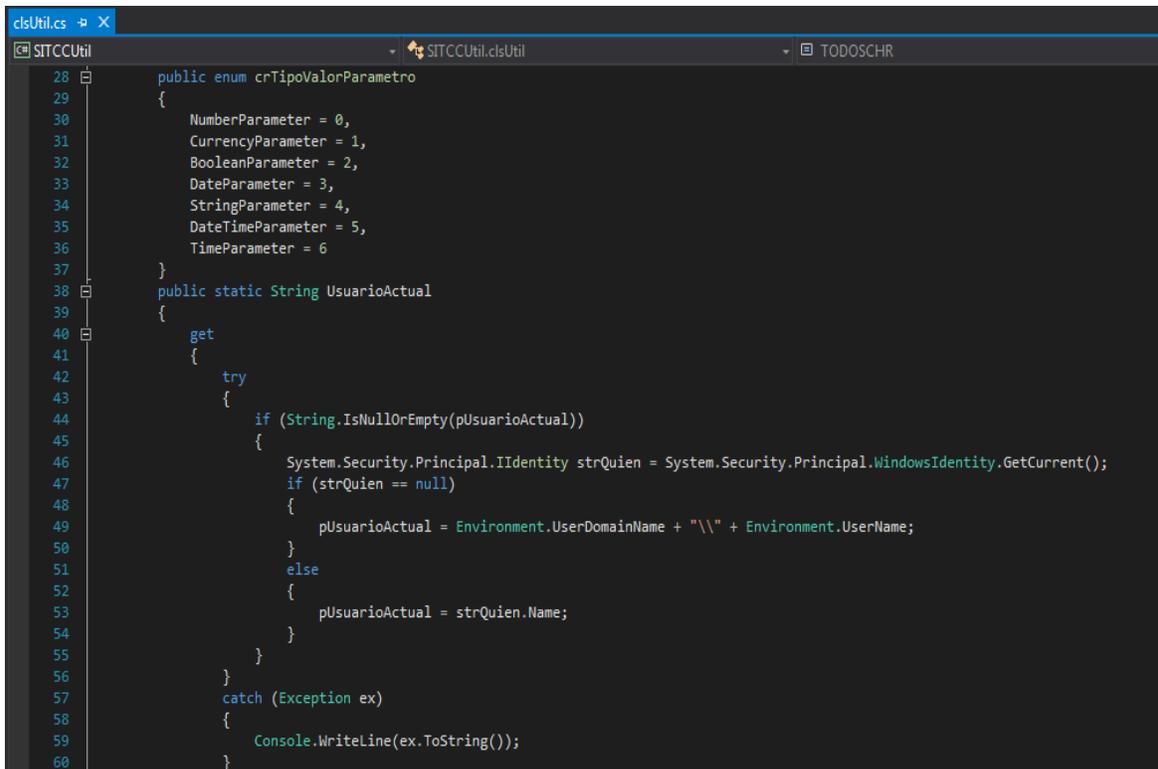
De igual manera, este proyecto hace referencia a otro proyecto de librería de clases que contiene funcionalidad genérica o utilitaria que puede ser utilizada en varios proyectos, satisfaciendo de esta manera el concepto de reutilización y encapsulación; siendo éstos pilares fundamentales del concepto de POO.

Anexos

Proyecto de Librerías de clases- SITCCUtil

Es un tipo de proyecto que al compilarse se genera un archivo de tipo “Extensión de la aplicación”. En el caso particular de este proyecto; contiene una serie de funcionalidades que pueden ser reutilizadas en los diferentes proyectos donde sea referenciado.

Dentro de estas tenemos, definición de constantes utilizadas en todo el sistema, funciones genéricas de cifrado y descripción, usuario de Windows con inicio de sesión, etc.



```
28 public enum crTipoValorParametro
29 {
30     NumberParameter = 0,
31     CurrencyParameter = 1,
32     BooleanParameter = 2,
33     DateParameter = 3,
34     StringParameter = 4,
35     DateTimeParameter = 5,
36     TimeParameter = 6
37 }
38 public static String UsuarioActual
39 {
40     get
41     {
42         try
43         {
44             if (String.IsNullOrEmpty(pUsuarioActual))
45             {
46                 System.Security.Principal.IIdentity strQuien = System.Security.Principal.WindowsIdentity.GetCurrent();
47                 if (strQuien == null)
48                 {
49                     pUsuarioActual = Environment.UserDomainName + "\\\" + Environment.UserName;
50                 }
51                 else
52                 {
53                     pUsuarioActual = strQuien.Name;
54                 }
55             }
56         }
57         catch (Exception ex)
58         {
59             Console.WriteLine(ex.ToString());
60         }
61     }
62 }
```

Figura 80: Código fuente parcial librería de clase Utilitario

Anexos

Proyecto de Librerías de clases- SITCCModelos

Es un tipo de proyecto que al compilarse genera un archivo de tipo “Extensión de la aplicación”. En el caso particular de este proyecto; contiene toda la definición de los modelos de clases utilizados por el sistema. Este proyecto puede ser reutilizado en los diferentes proyectos donde sea referenciado.

Cada clase del proyecto representa una entidad o tabla en la base de datos. Aquí se modela la estructura de la tabla y las operaciones CRUD que se ejecutan sobre la misma. Como método de estandarización y siguiendo los lineamientos y recomendaciones de desarrollo para bases de datos de servidor SQL, las operaciones CRUD en su totalidad son realizadas a través de procedimientos almacenados.

Esto significa que por cada tabla o entidad existe una vista que representa la información propia o cualquier información foránea con la cual la entidad tenga una relación. También, existen como mínimo cuatro procedimientos almacenados por cada entidad, que representan las operaciones de Crear, Leer, Actualizar y Borrar. Estas operaciones son realizadas a través de los métodos de cada clase que representa el modelo.

Anexos

```
clsGranja.cs  X
SITCCModelos  SITCCModelos.clsGranja
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Data.SqlClient;
6  using System.Data;
7  using SITCCDatos;
8
9  namespace SITCCModelos
10 {
11     public class clsGranja
12     {
13         private SqlCommand pcmd;
14
15         public clsGranja()...
16
17         public long IDGranja { get; set; }
18         public string Nombre { get; set; }
19         public string Ubicacion { get; set; }
20         public double Latitud { get; set; }
21         public double Longitud { get; set; }
22         public int Estado { get; set; }
23         public DateTime CreadoEl { get; set; }
24         public string CreadoPor { get; set; }
25         public string CreadoEn { get; set; }
26         public DateTime ActualizadoEl { get; set; }
27         public string ActualizadoPor { get; set; }
28         public string ActualizadoEn { get; set; }
29         public DateTime BorradoEl { get; set; }
30         public string BorradoPor { get; set; }
31         public string BorradoEn { get; set; }
32         public string DescripcionEstado { get; set; }
33
34     }
35 }
```

Figura 81: Código fuente parcial de los campos de la Entidad Granja en proyecto Modelos

```
clsGranja.cs  X
SITCCModelos  SITCCModelos.clsGranja  pcmd
35
36
37
38     public List<clsGranja> Obtener(long lngID = SITCCUtil.clsUtil.TODOS)...
39
40
41
42
43
44
45
46
47
48
49     private List<clsGranja> ConfigurarTabla(SqlDataReader psdr)...
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107     public Int64 Crear(string strNombre, string strUbicacion, double dblLatitud, double dblLongitud)...
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124     public Int64 Editar(Int64 intID, string strNombre, string strUbicacion, double dblLatitud, double dblLongitud)...
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195     public Int64 Borrar(Int64 intID)...
196
197
198
199
200
201
202
203
204 }
205 }
```

Figura 82: Código fuente parcial de los métodos de la Entidad Granja en proyecto Modelos

Anexos

```
clsGranja.cs x
SITCCModelos - SITCCModelos.clsGranja - pcmd
107 public Int64 Crear(string strNombre, string strUbicacion, double dblLatitud, double dblLongitud)
108 {
109     Int64 filas = 0;
110     try
111     {
112         using (SqlConnection pcn = new SqlConnection(SITCCDatos.clsManejadorDatos.strCadena))
113         {
114             pcn.Open();
115             if (pcn.State == ConnectionState.Open)
116             {
117                 SqlCommand pcmd = new SqlCommand();
118                 pcmd.Connection = pcn;
119                 pcmd.CommandText = "[dbo].[spCrearGranja]";
120                 pcmd.CommandType = CommandType.StoredProcedure;
121
122                 SqlParameter pr0 = new SqlParameter("@Nombre", SqlDbType.VarChar, 50);
123                 pr0.Value = strNombre;
124                 pcmd.Parameters.Add(pr0);
125
126                 SqlParameter pr1 = new SqlParameter("@Ubicacion", SqlDbType.VarChar, 100);
127                 pr1.Value = strUbicacion;
128                 pcmd.Parameters.Add(pr1);
129
130                 SqlParameter pr2 = new SqlParameter("@Latitud", SqlDbType.Float);
131                 pr2.Value = dblLatitud;
132                 pcmd.Parameters.Add(pr2);
133
134                 SqlParameter pr3 = new SqlParameter("@Longitud", SqlDbType.Float);
135                 pr3.Value = dblLongitud;
136                 pcmd.Parameters.Add(pr3);
137
138                 filas = Convert.ToInt64(pcmd.ExecuteScalar());
139             }
140         }
141     }
142 }
```

Figura 83: Código fuente parcial del método Crear de la Entidad Granja en proyecto Modelos

Modelo de Seguridad

El modelo de seguridad del sistema está compuesto de cinco entidades cuyas relaciones definen los permisos que un usuario puede tener sobre las demás entidades del sistema. Estos permisos pueden ser definidos de manera granular dando la posibilidad de controlar qué operaciones tienen derecho a realizar sobre cada entidad del sistema.

El modelo de seguridad general está definido por el siguiente diagrama de clases.

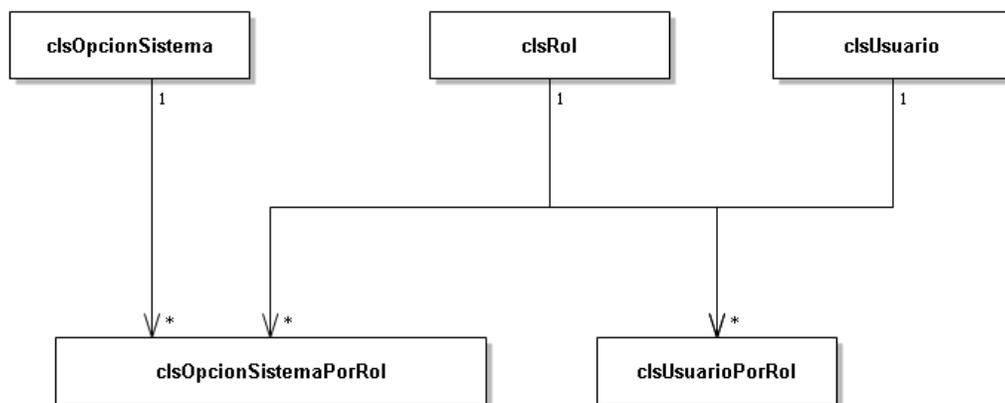


Figura 84: Modelo de seguridad

Anexos

Las entidades primarias son Usuario, Rol y Opción del sistema.

Usuario

Es la entidad que registra y gestiona los usuarios que pueden utilizar el sistema. Estos usuarios por sí mismos no son capaces de tener definición de permisos sobre las entidades. Ellos necesitan ser asociados a un rol.

Rol

Es la entidad que registra y gestiona los roles que son configurados en el sistema. Estos roles por sí mismos no son capaces de tener definición de permisos sobre las entidades. Ellos necesitan ser asociados a dos entidades: la entidad Usuario por Rol y la entidad Opción del Sistema por Rol.

Opción del Sistema

Es la entidad que registra y gestiona las opciones del sistema que componen el sistema. Estas opciones pueden aumentar o disminuir solamente por la modificación o cambio que tenga el sistema como tal. Se puede decir que esta entidad contiene la definición o metadato de la estructura funcional del sistema.

Usuario por Rol

Es la entidad que registra y gestiona los usuarios por rol que se definen o asocian en el sistema. Estos usuarios por roles por sí mismos son capaces de tener definición de permisos sobre las entidades; siempre que el rol esté asociado a una opción del sistema. De lo contrario, no hay una definición de permisos establecida.

Opción del Sistema por Rol

Es la entidad que registra y gestiona las opciones del sistema por rol que componen el sistema. Esta asociación es la que permite determinar qué rol tiene qué permiso sobre una entidad dada. Como consecuencia de la propiedad asociativa y extensiva, todos aquellos usuarios que pertenecen a un rol determinado heredarán los permisos que éste rol tiene sobre una entidad.

Anexos

clsOpcionSistemaPorRol 	
-	pcmd: SqlCommand
+	IDOpcionSistemaPorRol { get; set; } : Int64
+	IDOpcionSistema { get; set; } : Int64
+	IDRol { get; set; } : Int64
+	Rol { get; set; } : String
+	Descripcion { get; set; } : String
+	Nombre { get; set; } : String
+	NombreMenu { get; set; } : String
+	Insertar { get; set; } : Boolean
+	Actualizar { get; set; } : Boolean
+	Anular { get; set; } : Boolean
+	Seleccionar { get; set; } : Boolean
+	Listar { get; set; } : Boolean
+	Estado { get; set; } : Int32
+	CreadoEl { get; set; } : DateTime
+	CreadoPor { get; set; } : String
+	CreadoEn { get; set; } : String
+	ActualizadoEl { get; set; } : DateTime
+	ActualizadoPor { get; set; } : String
+	ActualizadoEn { get; set; } : String
+	BorradoEl { get; set; } : DateTime
+	BorradoPor { get; set; } : String
+	BorradoEn { get; set; } : String
+	DescripcionEstado { get; set; } : String
+	clsOpcionSistemaPorRol()
+	Obtener(IngID: Int64, IngIDOpcionSistema: Int64, l...
+	Obtener(strUsuario: String) : List<clsOpcionSistema...
-	ConfigurarTabla(psdr: SqlDataReader) : List<clsOpc...
+	Crear(IngIDOpcionSistema: Int64, IngIDRol: Int64, ...
+	Editar(intID: Int64, IngIDOpcionSistema: Int64, IngI...

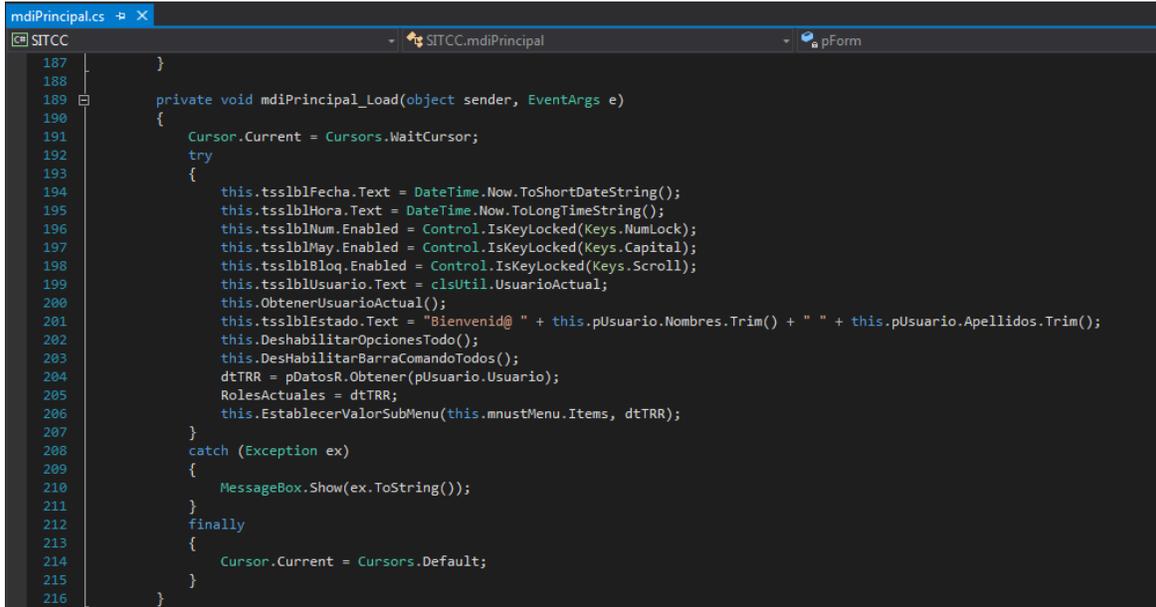
Figura 85: Opciones del Sistema por Rol

En esta entidad, se almacena la información de permisos (CRUD) que tendrá un rol sobre una opción del sistema o entidad del sistema.

Una vez guardada la información, el sistema la utiliza para conceder el uso de los permisos configurados al usuario. Esto lo realiza a través del siguiente mecanismo:

- Deshabilita todos los menús y sub menús que representan las opciones del sistema.
- Carga la definición de las Opciones del Sistema por Rol utilizando su usuario.
- Utiliza la colección obtenida para habilitar los menús y sub menús que representa las opciones del sistema sobre las cuales tiene permiso.

Anexos

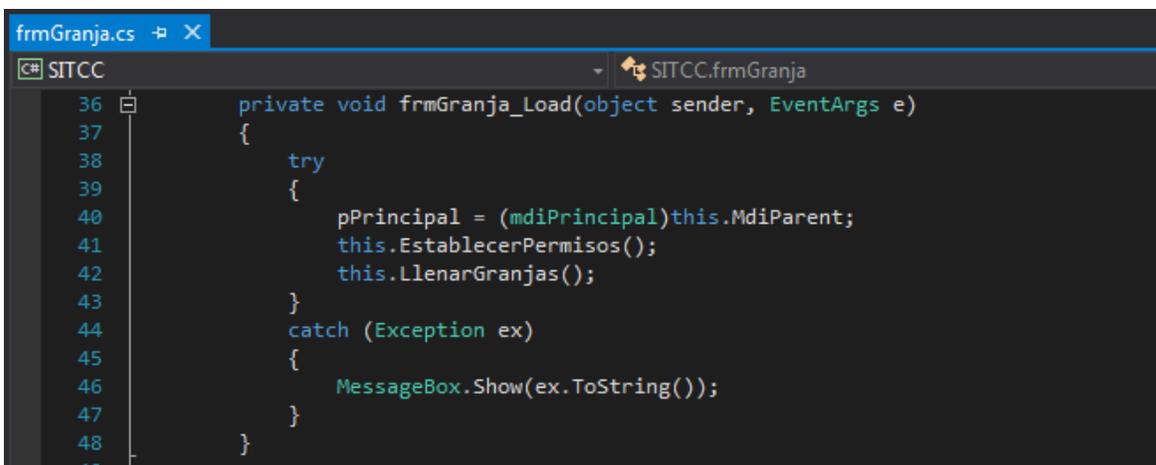


```
187     }
188
189     private void mdiPrincipal_Load(object sender, EventArgs e)
190     {
191         Cursor.Current = Cursors.WaitCursor;
192         try
193         {
194             this.tsslblFecha.Text = DateTime.Now.ToShortDateString();
195             this.tsslblHora.Text = DateTime.Now.ToLongTimeString();
196             this.tsslblNum.Enabled = Control.IsKeyLocked(Keys.NumLock);
197             this.tsslblMay.Enabled = Control.IsKeyLocked(Keys.Capital);
198             this.tsslblBloq.Enabled = Control.IsKeyLocked(Keys.Scroll);
199             this.tsslblUsuario.Text = clsUtil.UsuarioActual;
200             this.ObtenerUsuarioActual();
201             this.tsslblEstado.Text = "Bienvenid@ " + this.pUsuario.Nombres.Trim() + " " + this.pUsuario.Apellidos.Trim();
202             this.DeshabilitarOpcionesTodo();
203             this.DeshabilitarBarraComandoTodos();
204             dtTRR = pDatosR.Obtener(pUsuario.Usuario);
205             RolesActuales = dtTRR;
206             this.EstablecerValorSubMenu(this.mnustMenu.Items, dtTRR);
207         }
208         catch (Exception ex)
209         {
210             MessageBox.Show(ex.ToString());
211         }
212         finally
213         {
214             Cursor.Current = Cursors.Default;
215         }
216     }
```

Figura 86: Carga inicial de permisos en el formulario principal

Luego, al momento de que el usuario da clic sobre un menú (ejecuta una opción del sistema) el formulario hijo en cuestión realiza el siguiente mecanismo:

- En el evento "load" se invoca un método de la clase para establecer los permisos definidos para el usuario actual.

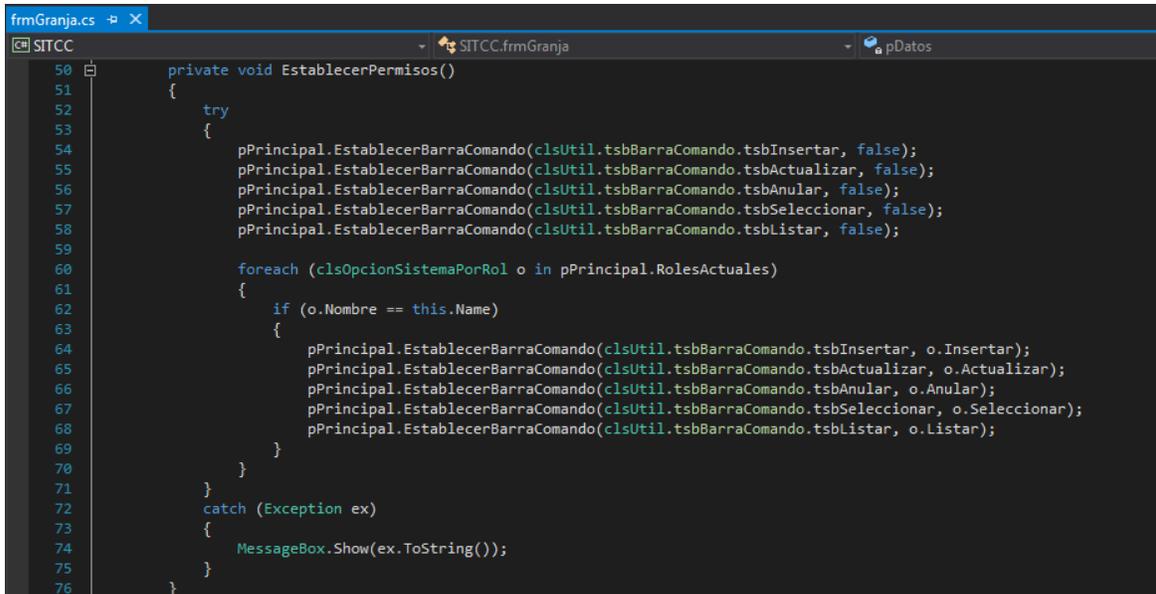


```
36     private void frmGranja_Load(object sender, EventArgs e)
37     {
38         try
39         {
40             pPrincipal = (mdiPrincipal)this.MdiParent;
41             this.EstablecerPermisos();
42             this.LlenarGranjas();
43         }
44         catch (Exception ex)
45         {
46             MessageBox.Show(ex.ToString());
47         }
48     }
```

Figura 87: Evento "Load" del formulario Granja

Anexos

- Obtiene la colección de roles asociados al usuario y recorre la colección para definir los permisos concedidos al usuario en la barra de comando que representa las operaciones CRUD.



```
50 private void EstablecerPermisos()
51 {
52     try
53     {
54         pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbInsertar, false);
55         pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbActualizar, false);
56         pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbAnular, false);
57         pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbSeleccionar, false);
58         pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbListar, false);
59
60         foreach (clsOpcionSistemaPorRol o in pPrincipal.RolesActuales)
61         {
62             if (o.Nombre == this.Name)
63             {
64                 pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbInsertar, o.Insertar);
65                 pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbActualizar, o.Actualizar);
66                 pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbAnular, o.Anular);
67                 pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbSeleccionar, o.Seleccionar);
68                 pPrincipal.EstablecerBarraComando(clsUtil.tsbBarraComando.tsbListar, o.Listar);
69             }
70         }
71     }
72     catch (Exception ex)
73     {
74         MessageBox.Show(ex.ToString());
75     }
76 }
```

Figura 88: Método EstablecerPermisos del formulario Granja

De manera similar, el mecanismo se propaga al resto de formularios hijos del sistema.

Exportación de Datos

La exportación de datos se realiza utilizando una clase genérica que convierte una colección de clases y las transforma a una cadena de texto. Delimitando cada valor por comillas dobles y separándolos por comas.

Esta clase se invoca en el evento clic del botón exportar en el formulario Gráfica. Al hacer clic, el sistema solicita al usuario una ubicación para guardar el archivo de texto; también, sugiere un nombre para el archivo utilizando los valores de los filtros aplicados por el usuario en ese formulario.

Dicha clase utiliza una técnica de programación conocida como tipo de parámetro genérico; donde, se define una estructura semántica de un método o una clase y en tiempo de ejecución puede ser sustituida por cualquier tipo base o clase pero manteniendo su funcionalidad estructural.

Anexos

```
frmGrafica.cs X
SITCC SITCC.frmGrafica btExportar_Click(object sender, EventArgs e)
430 private void btExportar_Click(object sender, EventArgs e)
431 {
432     try
433     {
434         List<clsLecturaCSV> lcsv = pDatos.ObtenerDatosCSV(this.IDGranja, this.IDPila, this.ObtenerParametroSeleccionado(), this.dtpDesde.Value.Date, this.dtpH
435         clsCsvExport<clsLecturaCSV> csv = new clsCsvExport<clsLecturaCSV>(lcsv);
436
437         this.sfdGuardar.FileName = "Lecturas de " + this.lblTitulo.Text.Trim().Replace("/", "-") + " " + ((clsParametro)this.lstParametro.SelectedItem).Descri
438         if (this.sfdGuardar.ShowDialog() == DialogResult.OK)
439         {
440             if (this.sfdGuardar.FileName.Length > 0)
441             {
442                 csv.ExportToFile(this.sfdGuardar.FileName);
443             }
444         }
445     }
446     catch (Exception ex)
447     {
448         MessageBox.Show(ex.ToString());
449     }
450 }
451 }
```

Figura 89: Código fuente parcial formulario Gráfica

```
clsCsvExport.cs X
SITCCModelos SITCCModelos.clsCsvExport<T>
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Reflection;
7 using System.IO;
8 using SITCCDatos;
9 /*
10 * Tomado y adaptado de: http://stackoverflow.com/questions/2422212/how-to-create-csv-excel-file-c/6989909#6989909
11 */
12 namespace SITCCModelos
13 {
14
15     public class clsCsvExport<T> where T : class
16     {
17         public List<T> Objects;
18
19         public clsCsvExport(List<T> objects)
20         {
21             Objects = objects;
22         }
23
24         public string Export()
25         {
26             return Export(true);
27         }
28
29         public string Export(bool includeHeaderLine)
30         {
31
32             StringBuilder sb = new StringBuilder();
33             //Get properties using reflection.
```

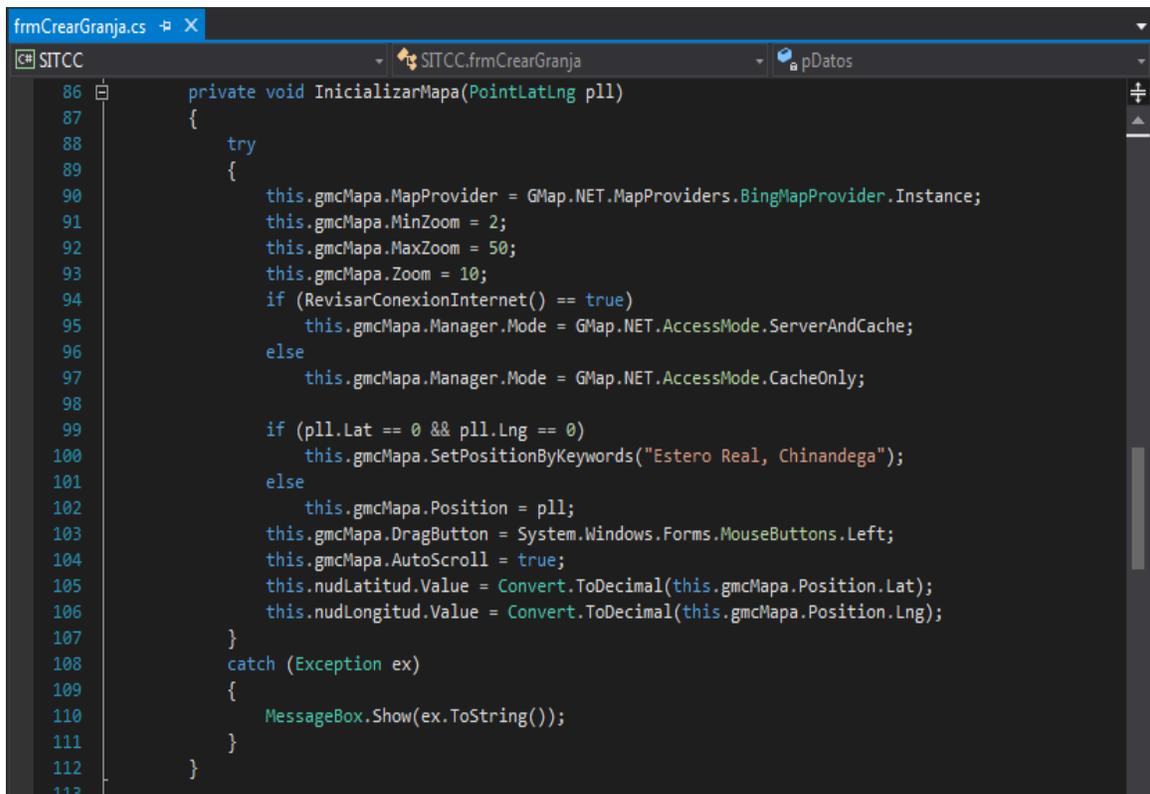
Figura 90: Código fuente parcial de la clase CsvExport

Anexos

Control de Mapa GMap.Net (Servicio Web)

Es un control para la plataforma .Net de código abierto y gratuito para formularios de Windows y se utiliza para consumir mapas de diferentes proveedores tales como: Bing, Google, Yahoo! y OpenStreet [77]. Este control se instala a través del gestor de paquetes de Visual Studio NuGet.

El gestor instala y agrega las referencias de GMap.Net para ser utilizado por los formularios de Windows. En tiempo de diseño se agrega el control al formulario. Una vez en el formulario, se pueden establecer todas las propiedades necesarias para su funcionamiento. Adicionalmente, hay que inicializar el control para que consuma un servicio web gratuito con la información geodésica necesaria para mostrar información en el mapa.



```
86     private void InicializarMapa(PointLatLng pll)
87     {
88         try
89         {
90             this.gmcMapa.MapProvider = GMap.NET.MapProviders.BingMapProvider.Instance;
91             this.gmcMapa.MinZoom = 2;
92             this.gmcMapa.MaxZoom = 50;
93             this.gmcMapa.Zoom = 10;
94             if (RevisarConexionInternet() == true)
95                 this.gmcMapa.Manager.Mode = GMap.NET.AccessMode.ServerAndCache;
96             else
97                 this.gmcMapa.Manager.Mode = GMap.NET.AccessMode.CacheOnly;
98
99             if (pll.Lat == 0 && pll.Lng == 0)
100                 this.gmcMapa.SetPositionByKeywords("Estero Real, Chinandega");
101             else
102                 this.gmcMapa.Position = pll;
103             this.gmcMapa.DragButton = System.Windows.Forms.MouseButtons.Left;
104             this.gmcMapa.AutoScroll = true;
105             this.nudLatitud.Value = Convert.ToDecimal(this.gmcMapa.Position.Lat);
106             this.nudLongitud.Value = Convert.ToDecimal(this.gmcMapa.Position.Lng);
107         }
108         catch (Exception ex)
109         {
110             MessageBox.Show(ex.ToString());
111         }
112     }
113 }
```

Figura 91: Código fuente parcial para inicializar control de mapa

Anexos

Control “Chart”

Es un control de Microsoft incluido en el .Net Framework 4.5 y se puede utilizar en formularios de Windows. Para utilizarlo una vez referenciado se agrega el control al formulario y se establecen las propiedades necesarias para su funcionamiento básico.

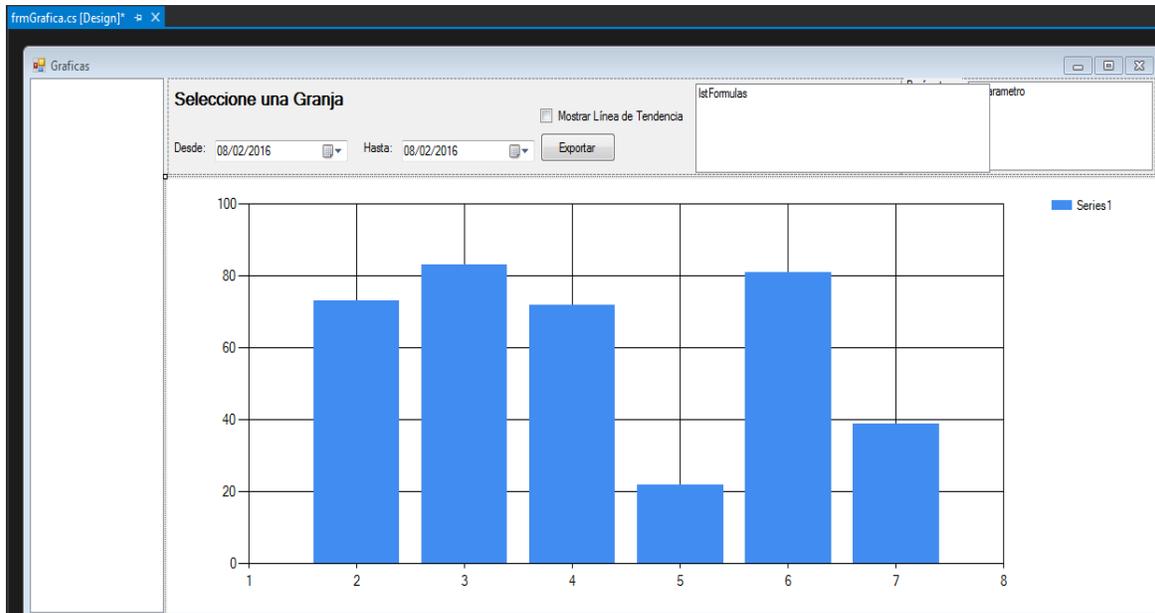


Figura 92: Control Chart en tiempo de diseño

Para mostrar información almacenada en una base de datos, primeramente se debe obtener los datos, inicializar el control y graficar los datos en el control.

Anexos

```
frmGrafica.cs  X
SITCC
SITCC.frmGrafica
tvArbol_NodeMouseDownDoubleClick(object sender, TreeNo

255 private void LlenarGraficas(long IDGranja = clsUtil.TODOS, long IDPila = clsUtil.TODOS)
256 {
257     try
258     {
259         this.chGrafico.Series.Clear();
260         String strTiempo;
261         string inputseries = "";
262         string outputseries = "";
263         String strResultado = "";
264         String strTituloP = "";
265         String strTituloL = "";
266         List<double> EjeX = new List<double>();
267         List<double> EjeY = new List<double>();
268         double x;
269         this.chGrafico.ChartAreas[0].AxisY.Title = ((clsParametro)this.lstParametro.SelectedItem).DescripcionCompuesta;
270
271         dtPL = pPila.Obtener(this.IDGranja, this.IDPila, this.ObtenerParametroSeleccionado(), this.dtpDesde.Value.Date, this.dtpHasta.Value.Date);
272         x = 1;
273         this.lstFormulas.Items.Clear();
274         foreach (clsPila p in dtPL)
275         {
276             srlaSerie = null;
277             strTituloP = "Pila-" + p.Numero.ToString();
278             srlaSerie = this.chGrafico.Series.Add(strTituloP);
279             srlaSerie.ChartType = SeriesChartType.Line;
280             srlaSerie.XValueType = ChartValueType.Auto;
281             srlaSerie.ToolTip = "Y=#VALY";
282
283             dtTR = pDatos.Obtener(p.IDGranja, p.IDPila, this.ObtenerParametroSeleccionado(), this.dtpDesde.Value.Date, this.dtpHasta.Value.Date);
284             EjeX.Clear();
285             EjeY.Clear();
286             foreach (clsLectura l in dtTR)
287             {
```

Figura 93: Código fuente parcial para graficar

El concepto básico es agregar al control chart series de datos que serán representadas gráficamente; en este caso particular, utilizando gráfica de líneas.

Para el sistema cada serie de datos la constituyen todas las lecturas de un parámetro seleccionado de una pila e un rango de fechas. Si el usuario selecciona más de una pila (selecciona la granja completa); cada conjunto de lecturas de un rango de fecha por cada pila es considerado un serie completa de datos.

De esta manera, el sistema agregará al control chart tantas series como pilas se seleccionen. Formando, cada serie de datos una línea graficada en el control.

Para crear la línea de tendencia, se utiliza la funcionalidad integrada en el gráfico llamada "FinanciamFormula". Con este método se pueden realizar predicciones sobre las series de datos del control chart. De igual manera, se pueden realizar diferentes tipos de regresiones dentro de estas se encuentra la lineal.

Anexos

```
frmGrafica.cs X
SIICC
SIICC.frmGrafica
tvArbol_NodeMouseDoubleClick(object sender, TreeNodeMouseClickEven

291 dpPunto.SetValueXY(strTiempo, l.Valor);
292 srLaSerie.Points.Add(dpPunto);
293 EjeX.Add(x++);
294 EjeY.Add(l.Valor);
295 }
296
297 if (this.ckbMostarTendencia.Checked == true)
298 {
299     string typeRegression = "Linear";
300     string forecasting = "1";
301     string error = "false";
302     string forecastingError = "false";
303     string parameters = typeRegression + ',' + forecasting + ',' + error + ',' + forecastingError;
304
305     srLaTendencia = null;
306     strTituloL = "Línea de Tendencia - Pila-" + p.Numero.ToString();
307     srLaTendencia = this.chGrafico.Series.Add(strTituloL);
308     srLaTendencia.ChartType = SeriesChartType.Line;
309     srLaTendencia.BorderWidth = 1;
310     srLaTendencia.ToolTip = "Y=#VALY";
311     inputseries = strTituloL;
312     outputseries = strTituloL;
313     this.chGrafico.DataManipulator.FinancialFormuLa(FinancialFormuLa.Forecasting, parameters, inputseries, outputseries);
314
315
316     double r2 = MathNet.Numerics.Statistics.Correlation.Pearson(EjeX, EjeY);
317     Tuple<double, double> ab = MathNet.Numerics.Fit.Line(EjeX.ToArray(), EjeY.ToArray());
318
319     strResultado = "Y" + p.Numero.ToString() + " = " + ab.Item2.ToString("0.00") + " + " + ab.Item1.ToString("0.00") + "X" + p.Numero.ToString() + "; R²";
320     this.lstFormulas.Items.Add(strResultado);
321 }
322
323
```

Figura 94: Código fuente parcial “FinancialFormuLa” del Control Chart de Microsoft

Adicionalmente, para calcular el coeficiente de determinación R^2 se utiliza una librería de código abierto llamada “Math.Net Numerics” que puede ser instalada y referenciada utilizando el gestor de paquetes NuGet.

Anexos

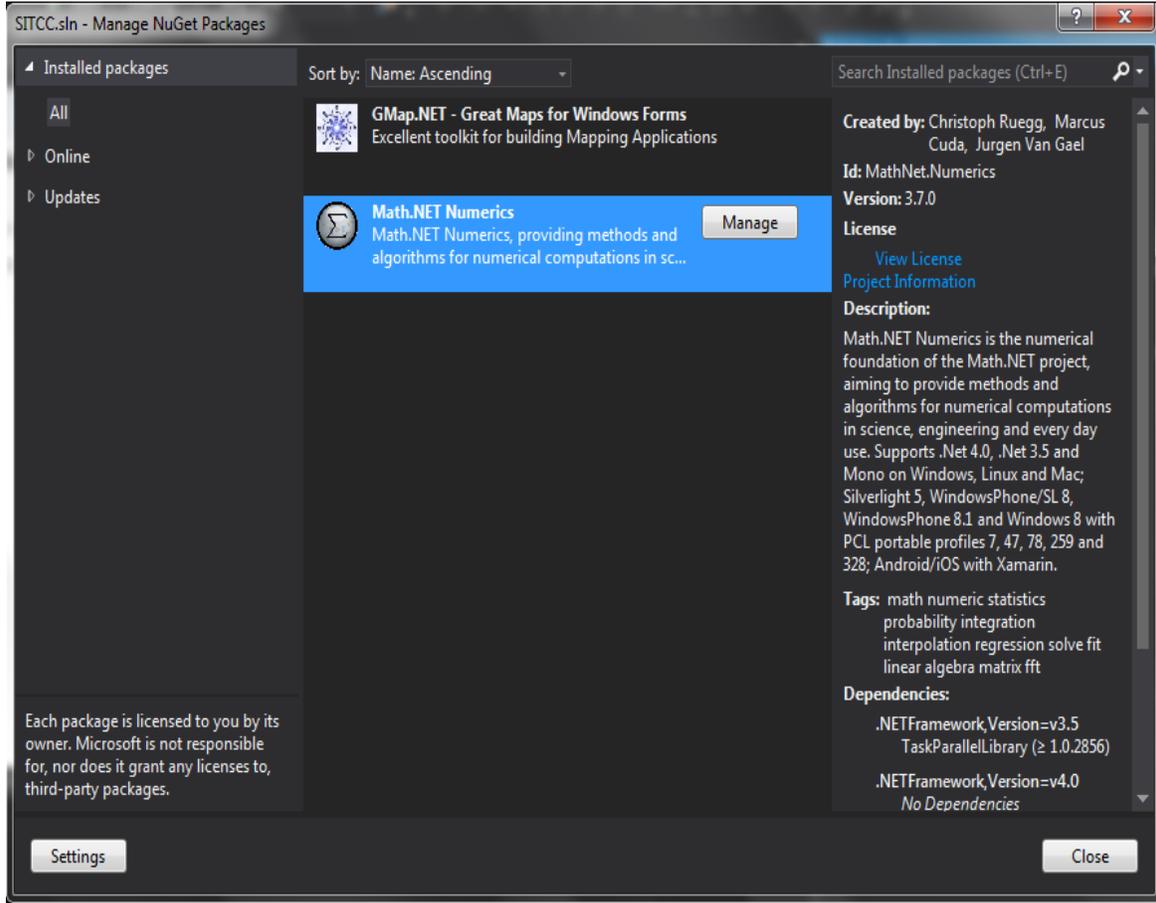


Figura 95: Gestor de paquetes NuGet para Visual Studio 2013

Esta librería contiene funcionalidad variada de cálculos entre ellas estadísticas de regresión lineal. Utilizando la librería se logra obtener los valores para generar la ecuación de la línea de tendencia. Obteniendo los valores de los coeficientes a , b y R^2 necesarios para la ecuación de la **Línea de tendencia**.

Anexos

Reportes

Los reportes en el sistema son visualizados a través del control de reportes Business Object Crystal Reports. Este control se encuentra como parte de las librerías de Crystal Reports y se utiliza en los formularios de Windows.

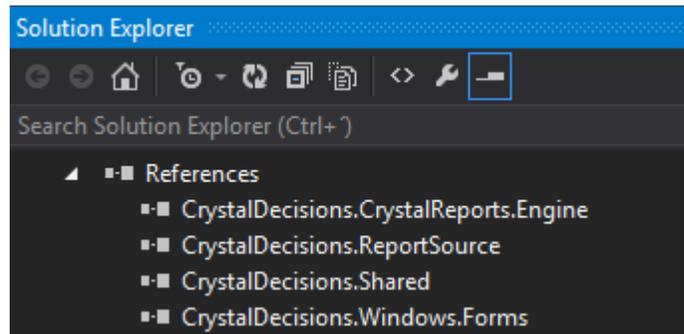


Figura 96: Referencias de Business Object Crystal Reports

Una vez referenciado, el control se coloca en un formulario que se utilizará como un visor para cargar los reportes del sistema. Una vez en el formulario se pueden establecer todas las propiedades necesarias para el uso apropiado del control.

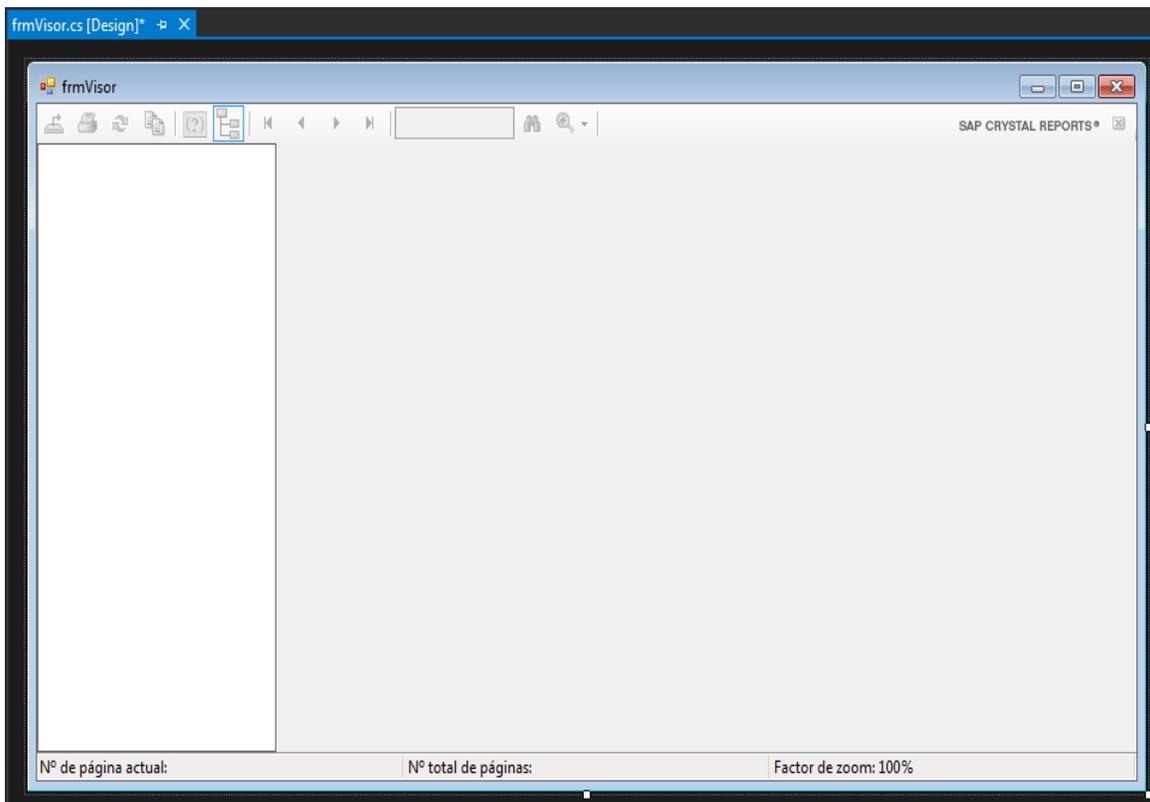
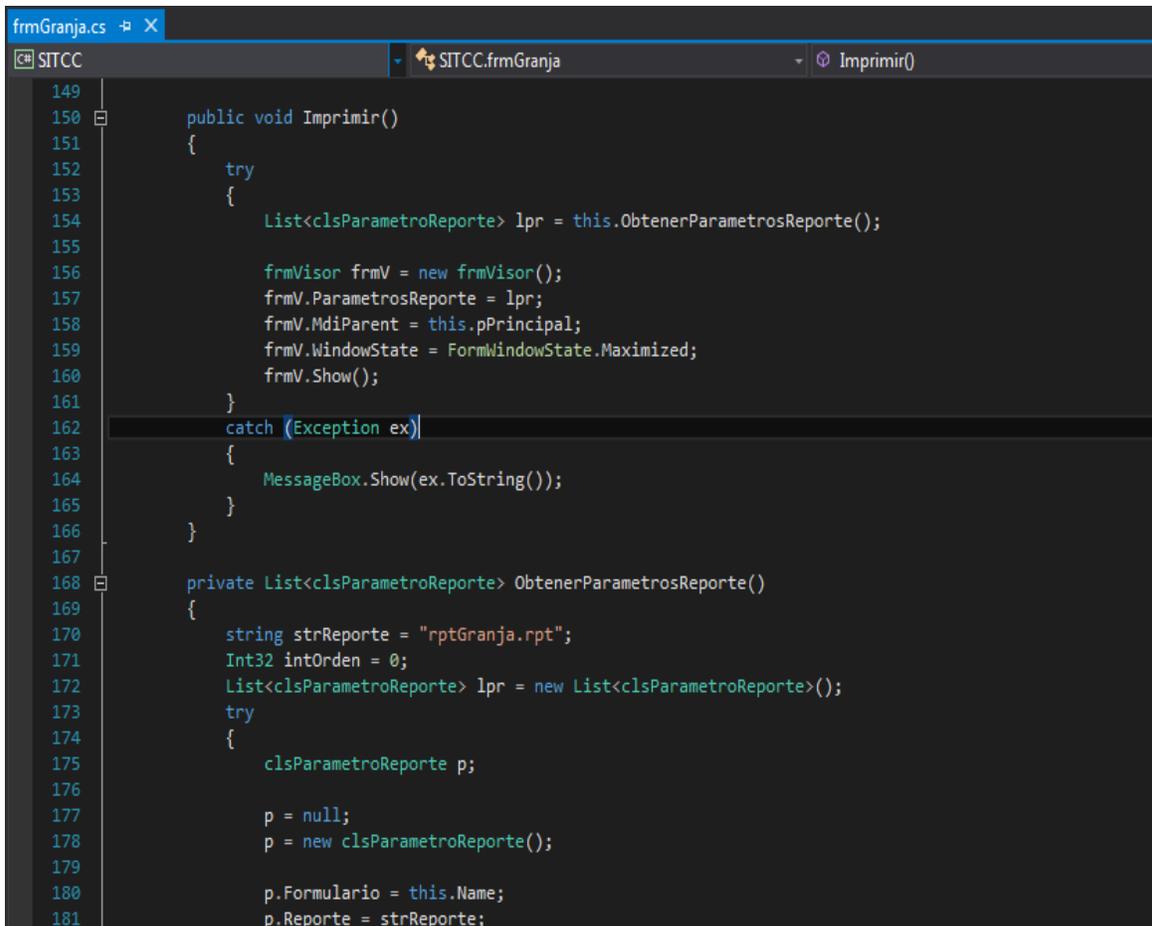


Figura 97: Control Crystal Reports en tiempo de diseño

Anexos

La carga de los reportes del sistema es totalmente contextual y dinámica. Es decir, los filtros o parámetros que necesita el reporte son tomados de los valores de las variables seleccionadas por el usuario. Cada reporte se carga dinámicamente al control visor y se realizan utilizando el siguiente mecanismo:

- Se realiza una instancia del visor y se le pasan los parámetros que utilizará el reporte para filtrar la información.

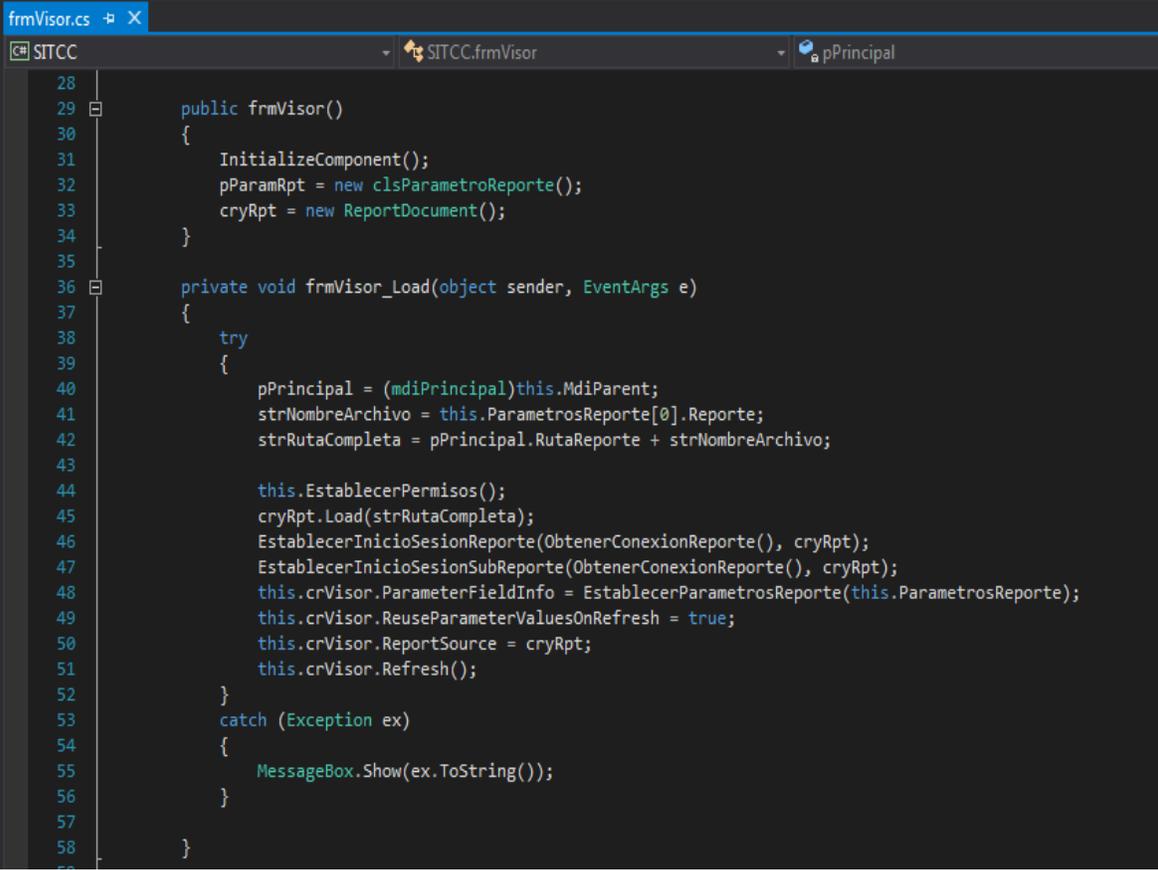


```
frmGranja.cs ▶ X
SITCC - SITCC.frmGranja - Imprimir()
149
150 public void Imprimir()
151 {
152     try
153     {
154         List<clsParametroReporte> lpr = this.ObtenerParametrosReporte();
155
156         frmVisor frmV = new frmVisor();
157         frmV.ParametrosReporte = lpr;
158         frmV.MdiParent = this.pPrincipal;
159         frmV.WindowState = FormWindowState.Maximized;
160         frmV.Show();
161     }
162     catch (Exception ex)
163     {
164         MessageBox.Show(ex.ToString());
165     }
166 }
167
168 private List<clsParametroReporte> ObtenerParametrosReporte()
169 {
170     string strReporte = "rptGranja.rpt";
171     Int32 intOrden = 0;
172     List<clsParametroReporte> lpr = new List<clsParametroReporte>();
173     try
174     {
175         clsParametroReporte p;
176
177         p = null;
178         p = new clsParametroReporte();
179
180         p.Formulario = this.Name;
181         p.Reporte = strReporte;
```

Figura 98: Código fuente parcial para imprimir un reporte

Anexos

- En el constructor del visor se instancia una clase ReportDocument, este es el documento principal que se carga en el visor. Adicionalmente, se crea una instancia de la clase parámetros del reporte que se utiliza para establecer los parámetros al objeto ReportDocument.

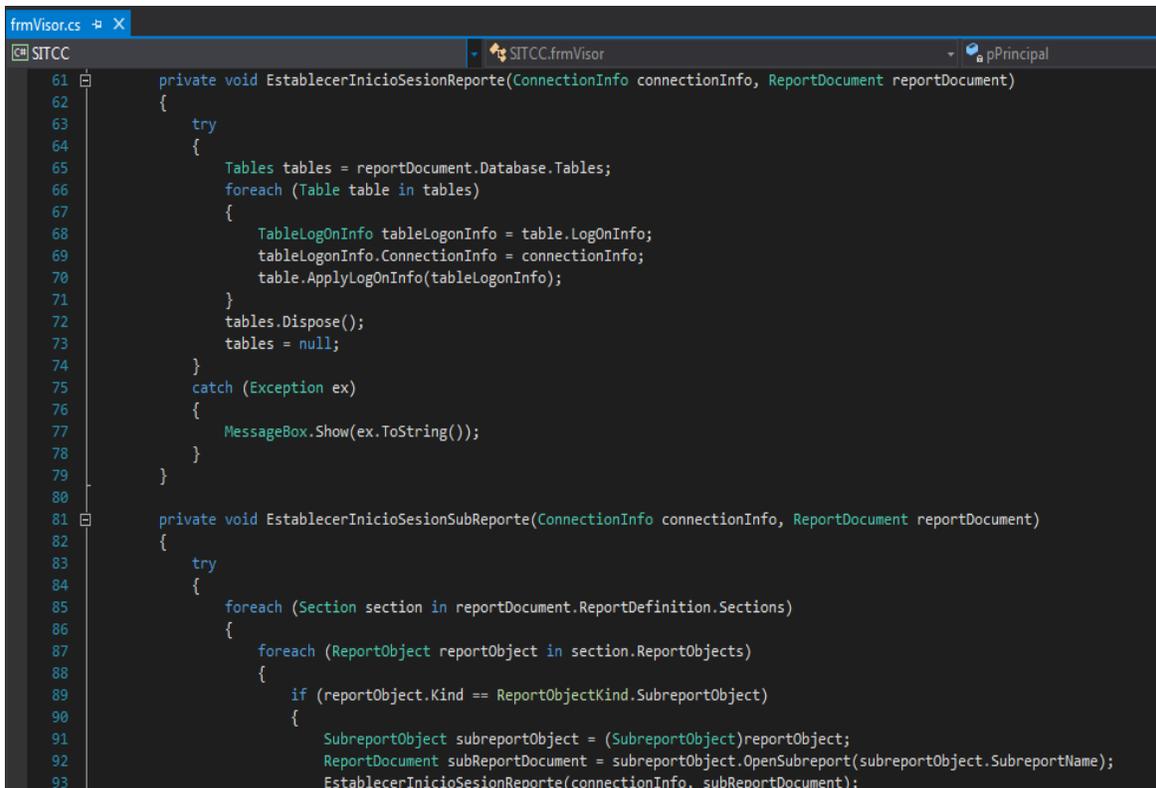


```
28
29     public frmVisor()
30     {
31         InitializeComponent();
32         pParamRpt = new clsParametroReporte();
33         cryRpt = new ReportDocument();
34     }
35
36     private void frmVisor_Load(object sender, EventArgs e)
37     {
38         try
39         {
40             pPrincipal = (mdiPrincipal)this.MdiParent;
41             strNombreArchivo = this.ParametrosReporte[0].Reporte;
42             strRutaCompleta = pPrincipal.RutaReporte + strNombreArchivo;
43
44             this.EstablecerPermisos();
45             cryRpt.Load(strRutaCompleta);
46             EstablecerInicioSesionReporte(ObtenerConexionReporte(), cryRpt);
47             EstablecerInicioSesionSubReporte(ObtenerConexionReporte(), cryRpt);
48             this.crVisor.ParameterFieldInfo = EstablecerParametrosReporte(this.ParametrosReporte);
49             this.crVisor.ReuseParameterValuesOnRefresh = true;
50             this.crVisor.ReportSource = cryRpt;
51             this.crVisor.Refresh();
52         }
53         catch (Exception ex)
54         {
55             MessageBox.Show(ex.ToString());
56         }
57     }
58 }
```

Figura 99: Código fuente parcial del visor de reportes

Anexos

- Luego se establece el inicio de sesión del reporte y sub reporte. Esto se hace para inicializar el inicio de sesión de cada objeto que contiene el reporte. Estos objetos pueden ser tablas, vistas o procedimientos almacenados. El inicio de sesión se hace tanto para las tablas contenidas en el reporte como aquellas tablas contenidas en los sub reportes que el reporte pueda contener.

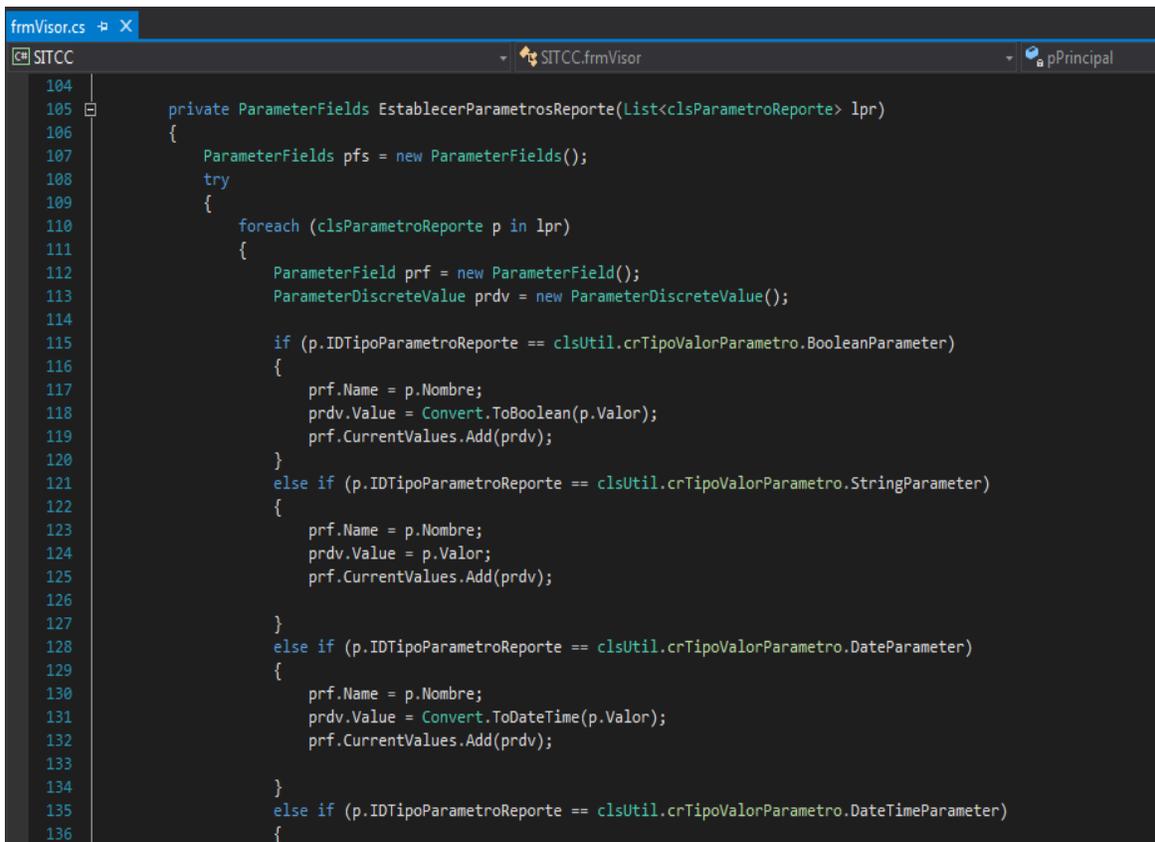


```
61 private void EstablecerInicioSesionReporte(ConnectionInfo connectionInfo, ReportDocument reportDocument)
62 {
63     try
64     {
65         Tables tables = reportDocument.Database.Tables;
66         foreach (Table table in tables)
67         {
68             TableLogOnInfo tableLogonInfo = table.LogOnInfo;
69             tableLogonInfo.ConnectionInfo = connectionInfo;
70             table.ApplyLogOnInfo(tableLogonInfo);
71         }
72         tables.Dispose();
73         tables = null;
74     }
75     catch (Exception ex)
76     {
77         MessageBox.Show(ex.ToString());
78     }
79 }
80
81 private void EstablecerInicioSesionSubReporte(ConnectionInfo connectionInfo, ReportDocument reportDocument)
82 {
83     try
84     {
85         foreach (Section section in reportDocument.ReportDefinition.Sections)
86         {
87             foreach (ReportObject reportObject in section.ReportObjects)
88             {
89                 if (reportObject.Kind == ReportObjectKind.SubreportObject)
90                 {
91                     SubreportObject subreportObject = (SubreportObject)reportObject;
92                     ReportDocument subReportDocument = subreportObject.OpenSubreport(subreportObject.SubreportName);
93                     EstablecerInicioSesionReporte(connectionInfo, subReportDocument);
94                 }
95             }
96         }
97     }
98 }
```

Figura 100: Código fuente parcial inicio de sesión reporte y sub reportes

Anexos

- Adicionalmente, es necesario establecer los valores de los parámetros que utilizará el reporte en tiempo de ejecución. De lo contrario, los parámetros definidos en el reporte serán solicitados al usuario al ejecutar el reporte. Para ello, se definió una clase que simula la estructura que utilizan los parámetros de los reportes.



```
104
105 private ParameterFields EstablecerParametrosReporte(List<clsParametroReporte> lpr)
106 {
107     ParameterFields pfs = new ParameterFields();
108     try
109     {
110         foreach (clsParametroReporte p in lpr)
111         {
112             ParameterField prf = new ParameterField();
113             ParameterDiscreteValue prdv = new ParameterDiscreteValue();
114
115             if (p.IDTipoParametroReporte == clsUtil.crTipoValorParametro.BooleanParameter)
116             {
117                 prf.Name = p.Nombre;
118                 prdv.Value = Convert.ToBoolean(p.Valor);
119                 prf.CurrentValues.Add(prdv);
120             }
121             else if (p.IDTipoParametroReporte == clsUtil.crTipoValorParametro.StringParameter)
122             {
123                 prf.Name = p.Nombre;
124                 prdv.Value = p.Valor;
125                 prf.CurrentValues.Add(prdv);
126             }
127             else if (p.IDTipoParametroReporte == clsUtil.crTipoValorParametro.DateParameter)
128             {
129                 prf.Name = p.Nombre;
130                 prdv.Value = Convert.ToDateTime(p.Valor);
131                 prf.CurrentValues.Add(prdv);
132             }
133             else if (p.IDTipoParametroReporte == clsUtil.crTipoValorParametro.DateTimeParameter)
134             {
135                 prf.Name = p.Nombre;
136                 prdv.Value = Convert.ToDateTime(p.Valor);
137                 prf.CurrentValues.Add(prdv);
138             }
139         }
140     }
141     catch { }
142     return pfs;
143 }
```

Figura 101: Código fuente parcial establecer parámetros de reporte

Anexos

Diagrama de clases

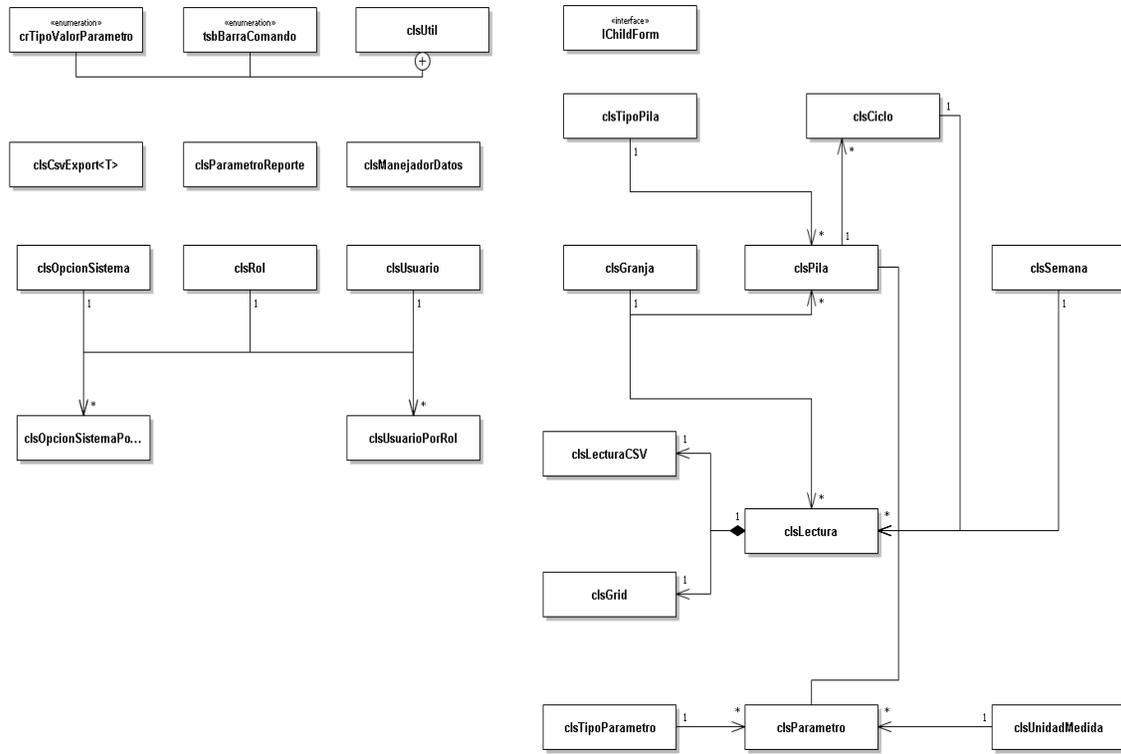
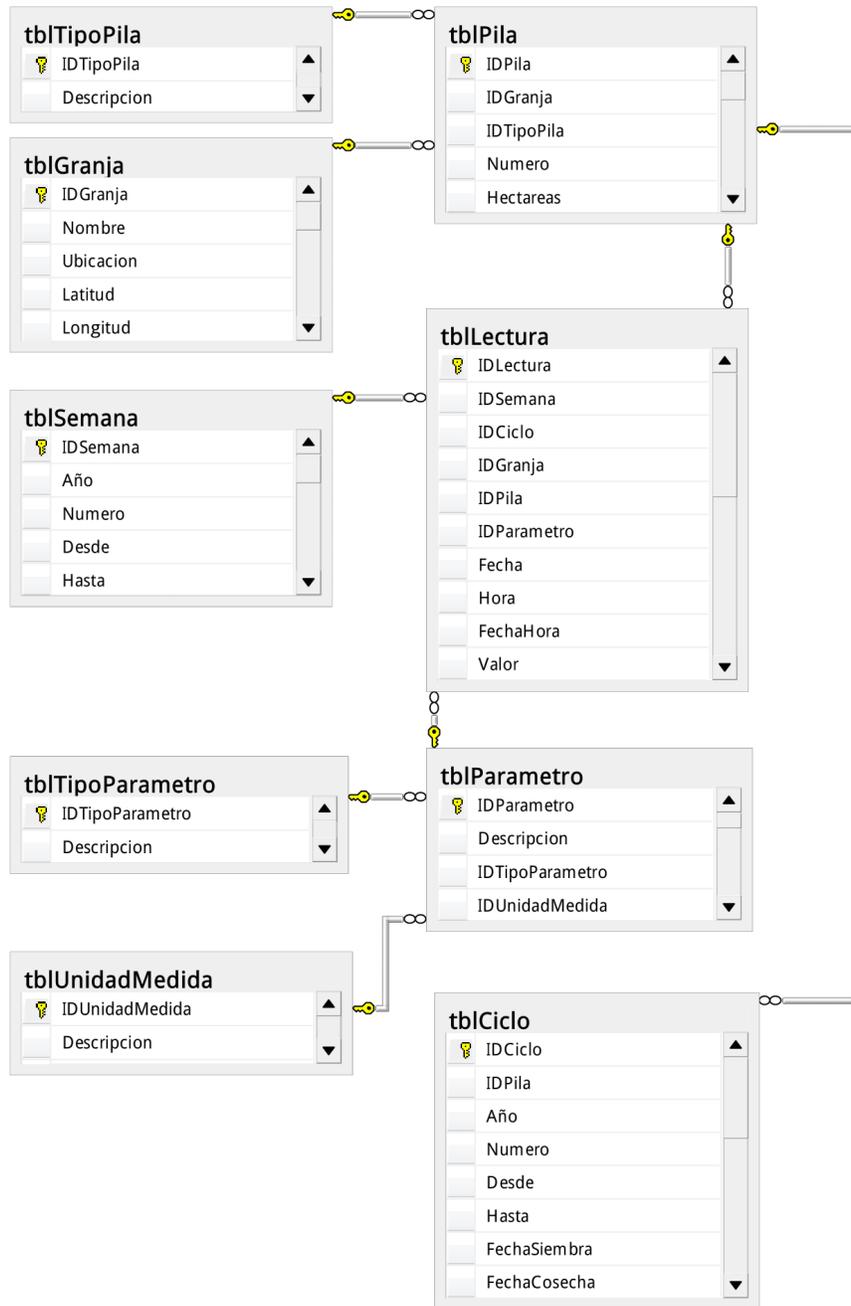


Figura 102: Diagrama de clases

Anexos

Diagrama entidad relación



1-1

Figura 103: Diagrama entidad relación

Anexos

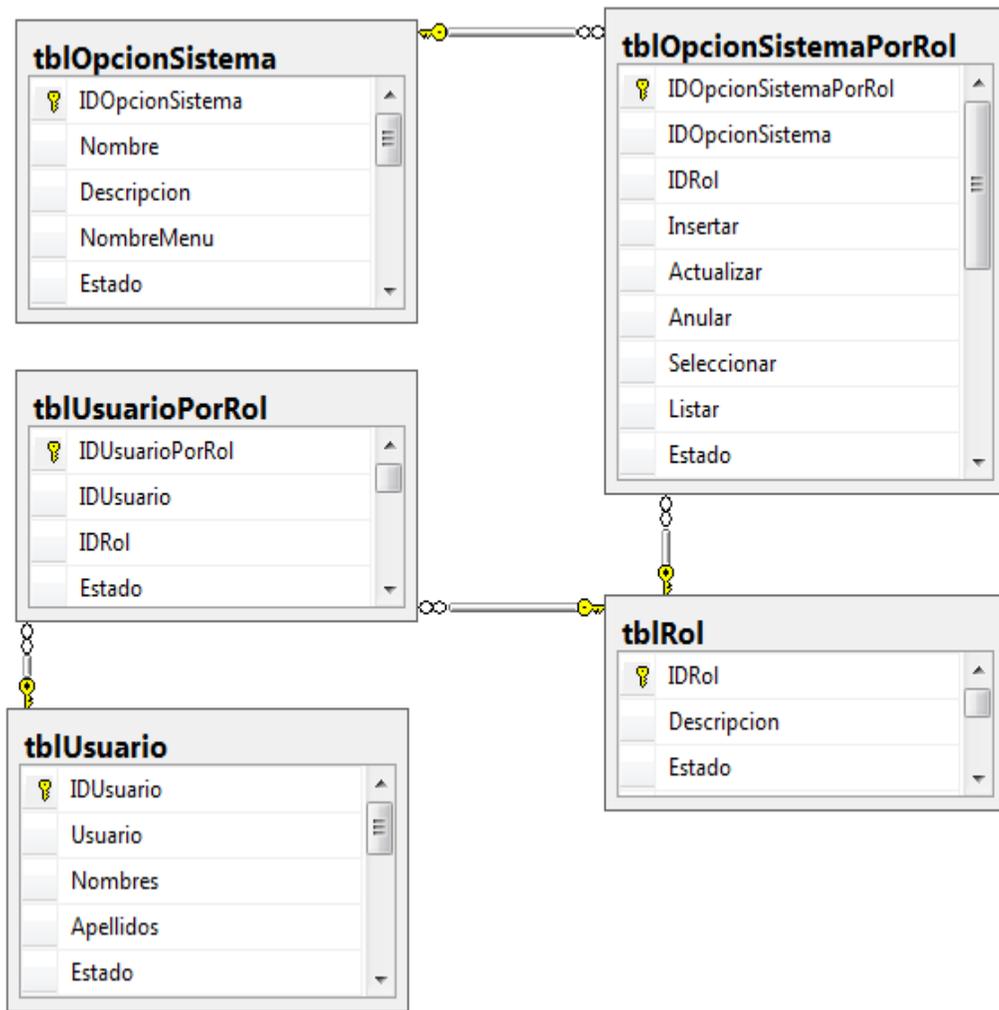


Figura 104: Diagrama entidad relación segunda parte