



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA
TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO.

TEMA: “DESARROLLO DE UN SISTEMA PARA EL PROCESO
DE COTIZACIÓN Y LA OPTIMIZACIÓN DE RECURSOS EN
PRODUCCIÓN, PARA ALGLASS”

AUTOR: ROGEL CONLAGO, GUILLERMO NAPOLEÓN

DIRECTOR: CAMPAÑA, MAURICIO

CODIRECTOR: DE LA TORRE, ARTURO

SANGOLQUÍ, ABRIL DEL 2016

CERTIFICACION



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "DESARROLLO DE UN SISTEMA PARA EL PROCESO DE COTIZACIÓN Y LA OPTIMIZACIÓN DE RECURSOS EN PRODUCCIÓN, PARA ALGLASS" realizado por el señor GUILLERMO NAPOLEÓN ROGEL CONLAGO ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor GUILLERMO NAPOLEÓN ROGEL CONLAGO para que lo sustente públicamente.

Sangolquí, 11 de abril del 2016

Ing. Mauricio Campaña.

DIRECTOR



AUTORIA DE RESPONSABILIDAD



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, GUILLERMO NAPOLEON ROGEL CONLAGO, con cédula de identidad N° 1716312754, declaro que este trabajo de titulación "DESARROLLO DE UN SISTEMA PARA EL PROCESO DE COTIZACIÓN Y LA OPTIMIZACIÓN DE RECURSOS EN PRODUCCIÓN, PARA ALGLASS" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 11 de abril del 2016

Guillermo Napoleón Rogel Conlago

CC. 1716312754

AUTORIZACION



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, GUILLERMO NAPOLEÓN ROGEL CONLAGO, autorizo a la Universidad De Las Fuerzas Armadas – ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "DESARROLLO DE UN SISTEMA PARA EL PROCESO DE COTIZACIÓN Y LA OPTIMIZACIÓN DE RECURSOS EN PRODUCCIÓN, PARA ALGLASS", cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 11 de abril del 2016



Guillermo Napoleón Rogel Conlago

CC. 1716312754

DEDICATORIA

Dedico esta tesis a Dios quien me ha permitido perseverar y culminar este trabajo, así como a mi esposa Ana Iza quien con su amor y apoyo incondicional me alentó y animo a pesar del tiempo que me tome para terminar este trabajo, a mis hijos Guillermo Rogel y Viviana Rogel quienes me inspiran para crecer y progresar, también la dedico a mis padres Guillermo Rogel e Ignacia Conlago que fueron un pilar y apoyo total y mis hermanas Leslie Rogel y Daisy Rogel que confiaron en mí, y a Valentina Granja por su ayuda y apoyo.

AGRADECIMIENTO

Mis agradecimientos a Dios por guiar mi camino, a mis Padres Guillermo e Ignacia por ser un apoyo moral así como económico, a mi esposa Ana Iza por su sacrificio, tiempo y estar a mi lado en cada momento mientras estudiaba, mis hijos Guillermo y Viviana que me animaban e insistían que termine, mis hermanas Daisy y Leslie que me alentaron, y amigos, profesores y todas las personas especiales que en algún momento pusieron su granito de arena para que pueda obtener conocimientos necesarios para cumplir con este trabajo y con esta carrera.

ÍNDICE

CERTIFICACION.....	II
AUTORIA DE RESPONSABILIDAD	III
AUTORIZACION	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE.....	VII
ÍNDICE DE TABLAS	XI
INDICE DE FIGURAS.....	XII
ÍNDICE DE CUADROS.....	XIV
RESUMEN	XVIII
ABSTRACT	XIX
CAPÍTULO 1.....	1
1 INTRODUCCIÓN	1
1.1 VISIÓN GENERAL	1
1.2 VISIÓN DE ALGLASS.....	1
1.3 PLANTEAMIENTO DEL PROBLEMA.....	2
1.4 JUSTIFICACIÓN	3
1.5 OBJETIVOS	4
1.5.1 <i>Objetivo general</i>	4
1.5.2 <i>Objetivos específicos</i>	4
1.6 ALCANCE.....	5
1.7 METODOLOGÍA.....	6
1.8 HERRAMIENTAS.....	6
1.9 FACTIBILIDAD.....	6
1.9.1 <i>Factibilidad Técnica</i>	6
1.9.2 <i>Factibilidad Operativa</i>	6
1.9.3 <i>Factibilidad Económica</i>	7
CAPÍTULO 2.....	8

2	MARCO TEÓRICO	8
2.1	APLICACIONES DE ESCRITORIO	8
2.2	FRAMEWORK.....	8
2.2.1	<i>Framework .NET</i>	9
2.3	LINQ.....	10
2.3.1	<i>Operadores de consulta estándar</i>	11
2.3.2	<i>Arquitectura del Linq</i>	15
2.4	HERRAMIENTAS DE DESARROLLO.....	15
2.4.1	<i>Microsoft Visual Studio community 2015. (Microsoft, 2015)</i>	15
2.4.2	<i>Visual Basic</i>	17
2.5	SQL (ESCOFET, 2007)	19
2.5.1	<i>Características generales del SQL</i>	19
2.6	MICROSOFT SQL SERVER 2014 EXPRESS (MALEMEZA, 2012)	20
2.6.1	<i>Características (Malemeza, 2012)</i>	20
2.7	METODOLOGÍA XP (EXTREME PROGRAMMING) (ECURED, 2010)	22
2.7.1	<i>Valores</i>	22
2.7.2	<i>Características fundamentales</i>	24
2.7.3	<i>Primera Fase: Planificación del proyecto</i>	25
2.7.4	<i>Segunda Fase: Diseño</i>	26
2.7.5	<i>Tercera Fase: Codificación</i>	28
2.7.6	<i>Cuarta Fase: Pruebas</i>	30
	CAPÍTULO 3.....	32
3	ANÁLISIS Y DISEÑO	32
3.1	PLANIFICACIÓN	32
3.1.1	<i>Requisitos Funcionales</i>	32
3.1.2	<i>Requisitos no funcionales</i>	33
3.2	HISTORIAS DE USUARIO	33
3.2.1	<i>MÓDULO DE SEGURIDADES</i>	34
3.2.2	<i>MÓDULO DE PRODUCTOS</i>	35
3.2.3	<i>MÓDULO DE COSTEO</i>	40
3.2.4	<i>MÓDULO DE ORDENES DE ENTREGA</i>	51
3.2.5	<i>MÓDULO DE REPORTES</i>	59

3.3	PLAN DE PUBLICACIONES.....	62
3.4	ITERACIONES.....	63
3.5	NOMENCLATURA.....	63
3.6	TARJETAS CRC	64
3.6.1	<i>Módulo de seguridades</i>	65
3.6.2	<i>Módulo de productos</i>	65
3.6.3	<i>Módulo de costeo</i>	65
3.6.4	<i>Módulos de órdenes de entrega</i>	68
3.6.5	<i>Módulo de planes de corte</i>	68
3.7	DISEÑO DE BASE DE DATOS.	69
3.8	MODELO FÍSICO.....	69
4	CODIFICACIÓN DEL SISTEMA.....	71
4.1	HISTORIA DE USUARIO 1. MANEJO DE USUARIOS	71
4.2	HISTORIA DE USUARIO 2. COLORES PARA COTIZAR.....	73
4.3	HISTORIA DE USUARIO 3. INGRESO DE TIPOS DE PRODUCTOS.....	74
4.4	HISTORIA DE USUARIO 4. INGRESO DE PRODUCTOS.....	76
4.5	HISTORIA DE USUARIO 5. INGRESO DE CLIENTES	79
4.6	HISTORIA DE USUARIO 6. MANO DE OBRA.....	81
4.7	HISTORIA DE USUARIO 7. TIPOS DE ESTRUCTURAS	82
4.8	HISTORIA DE USUARIO 8. DETALLE DE ESTRUCTURAS	86
4.9	HISTORIA DE USUARIO 9. DESCUENTOS POR PIEZAS	89
4.10	HISTORIA DE USUARIO 10. PROFORMA DE ESTRUCTURAS	90
4.11	HISTORIA DE USUARIO 11. CONTRATOS	96
4.12	HISTORIA DE USUARIO 12. LISTA DE MATERIALES.....	99
4.13	HISTORIA DE USUARIO 13. ENTREGA DE MATERIALES	103
4.14	HISTORIA DE USUARIO 14. GENERAR PLANES DE CORTE.....	107
4.15	HISTORIA DE USUARIO 15. REPORTES DE COSTOS Y FINALIZACIONES. ...	109
5	PRUEBAS Y RESULTADOS.	113
5.1	PRUEBAS DE ACEPTACIÓN	113
5.2	RESULTADOS.....	151
CAPÍTULO 6	153

6 CONCLUSIONES Y RECOMENDACIONES.....	^x 153
6.1 CONCLUSIONES.....	153
6.2 RECOMENDACIONES	154
BIBLIOGRAFÍA.....	155

ÍNDICE DE TABLAS

Tabla 1	Factibilidad económica.....	7
Tabla 2	Plan de Publicaciones.....	62
Tabla 3	Iteraciones	63

INDICE DE FIGURAS.

Figura 1 Arquitectura LINQ	15
Figura 2 Comparativa Visual Studio.....	16
Figura 3 Diseño de Base de Datos	69
Figura 4 Modelo Físico	70
Figura 5 Tabla Usuario	71
Figura 6 Formulario de Usuarios.....	71
Figura 7 Tabla Color Insumos.....	73
Figura 8 Formulario Colores	73
Figura 9 Tabla Tipo de Producto.....	74
Figura 10 Formulario ingreso de Productos.....	75
Figura 11 Tablas de Productos e Insumos	76
Figura 12 Formulario de Insumos	77
Figura 13 Tabla Cliente.....	79
Figura 14 Formulario Cliente.....	79
Figura 15 Tabla Mano de Obra	81
Figura 16 Formulario par Mano de Obra.....	81
Figura 17 Tablas Grupo de estructuras y tipos de estructuras	83
Figura 18 Formulario para grupo de estructuras.....	83
Figura 19 Formulario para tipo de estructuras	84
Figura 20 Tablas de Material Estructuras e ítems de estructuras.....	86
Figura 21 Formulario de Materiales	86
Figura 22 Formulario para Datos de Materiales.....	87
Figura 23 Formulario para Descuentos.....	90
Figura 24 Tablas para Proformas	90
Figura 25 Encabezado del formulario Proforma.....	91
Figura 26 Formulario para Estructuras de proforma	91
Figura 27 Formulario para Adicionales de la proforma	93
Figura 28 Formulario para Vidrios.....	94
Figura 29 Tabla de Contrato	97
Figura 30 Formulario de Contratos	97
Figura 31 Formulario Buscar Contrato.....	99
Figura 32 Tabla lista de Materiales	100

Figura 33 Formulario lista de materiales	100
Figura 34 Lista de materiales a Excel	102
Figura 35 Orden de entrega de materiales a Excel.....	102
Figura 36 Tabla de técnicos.....	103
Figura 37 Formulario para técnicos	103
Figura 38 Tabla de lista de materiales	105
Figura 39 Tabla para adicionales de Contrato	105
Figura 40 Formulario para órdenes de Entrega	105
Figura 41 Tabla para plan de Corte	107
Figura 42 Formulario para buscar Proforma	107
Figura 43 Formulario de plan de corte	108
Figura 44 Formulario de Reporte de Técnicos.....	109
Figura 45 Formulario de reporte final.....	110
Figura 46 Formulario Reporte de proformas.....	111
Figura 47 Formulario reporte de contrato	111
Figura 48 Resultado en cotizaciones	151
Figura 49 Obtener lista de materiales	151
Figura 50 Plan de corte.....	152

ÍNDICE DE CUADROS

Cuadro 1	Tabla de operadores SQL	11
Cuadro 2	Tarjeta CRC	28
Cuadro 3	Historia de Usuario 1	34
Cuadro 4	Tarea 1.1	34
Cuadro 5	Tarea 1.2	35
Cuadro 6	Historia de Usuario 2	35
Cuadro 7	Tarea 2.1	36
Cuadro 8	Tarea 2.2	36
Cuadro 9	Historia de Usuario 3	37
Cuadro 10	Tarea 3.1	37
Cuadro 11	Tarea 3.2	38
Cuadro 12	Historia de Usuario 4	38
Cuadro 13	Tarea 4.1	39
Cuadro 14	Tarea 4.2	39
Cuadro 15	Tarea 4.3	40
Cuadro 16	Historia de Usuario 5	40
Cuadro 17	Tarea 5.1	41
Cuadro 18	Tarea 5.2	41
Cuadro 19	Historia de Usuario 6	42
Cuadro 20	Tarea 6.1	42
Cuadro 21	Tarea 6.2	43
Cuadro 22	Historia de usuario 7	43
Cuadro 23	Tarea 7.1	44
Cuadro 24	Tarea 7.2	44
Cuadro 25	Tarea 7.3	45
Cuadro 26	Tarea 7.4	45
Cuadro 27	Historia de usuario 8	46
Cuadro 28	Tarea 8.1	46
Cuadro 29	Tarea 8.2	47
Cuadro 30	Historia de usuario 9	47
Cuadro 31	Tarea 9.1	48
Cuadro 32	Tarea 9.2	48

Cuadro 33	Historia de usuario 10	49
Cuadro 34	Tarea 10.1	49
Cuadro 35	Tarea 10.2	50
Cuadro 36	Tarea 10.3	50
Cuadro 37	Historia de usuario 11	51
Cuadro 38	Tarea 11.1	51
Cuadro 39	Tarea 11.2	52
Cuadro 40	Historia de usuario 12	52
Cuadro 41	Tarea 12.1	53
Cuadro 42	Tarea 12.2	53
Cuadro 43	Tarea 12.3	54
Cuadro 44	Historia de usuario 13	54
Cuadro 45	Tarea 13.1	55
Cuadro 46	Tarea 13.2	55
Cuadro 47	Tarea 13.3	56
Cuadro 48	Tarea 13.4	56
Cuadro 49	Historia de usuario 14	57
Cuadro 50	Tarea 14.1	57
Cuadro 51	Tarea 14.2	58
Cuadro 52	Tarea 14.3	58
Cuadro 53	Historia de usuario 15	59
Cuadro 54	Tarea 15.1	60
Cuadro 55	Tarea 15.2	60
Cuadro 56	Tarea 15.3	61
Cuadro 57	Tarea 15.4	61
Cuadro 58	Prefijos	63
Cuadro 59	Tarjeta CRC clsUsuarios	65
Cuadro 60	Tarjeta CRC clsProductos	65
Cuadro 61	Tarjeta CRC clsClientes	66
Cuadro 62	Tarjeta CRC clsEstructuras	66
Cuadro 63	Tarjeta CRC clsProforma	67
Cuadro 64	Tarjeta CRC clsContrato	67
Cuadro 65	Tarjeta CRC clsListaMateriales	68

Cuadro 66	Tarjeta CRC clsPlanCorte	68
Cuadro 67	Prueba de aceptación Registro correcto de usuario.....	113
Cuadro 68	Prueba de aceptación registro incorrecto de usuario	114
Cuadro 69	Prueba de aceptación eliminar usuario	115
Cuadro 70	Prueba de aceptación ingreso de nuevo color	116
Cuadro 71	Prueba de aceptación Ingreso erróneo de color	117
Cuadro 72	Prueba de aceptación Modificar color	118
Cuadro 73	Prueba de aceptación Ingresar tipo de producto.....	119
Cuadro 74	Ingreso incorrecto de tipo de producto	120
Cuadro 75	Prueba de aceptación modificar tipo de producto	121
Cuadro 76	Prueba de aceptación ingreso de productos	122
Cuadro 77	Prueba de aceptación ingreso erróneo de productos.....	123
Cuadro 78	Prueba de aceptación modificar producto	124
Cuadro 79	Prueba de aceptación ingreso de cliente	125
Cuadro 80	Prueba de aceptación ingreso de clientes.....	126
Cuadro 81	Prueba de aceptación modificar clientes.....	127
Cuadro 82	Prueba de aceptación ingreso valor mano de obra	128
Cuadro 83	ingreso negativo o 0 de mano de obra	129
Cuadro 84	Prueba de aceptación modificar valor mano de obra	130
Cuadro 85	Prueba de aceptación ingresar nuevo grupo de estructura.....	131
Cuadro 86	Prueba de aceptación ingresar grupo erróneo de estructura ..	132
Cuadro 87	Prueba de aceptación modificar grupo de estructura.....	133
Cuadro 88	Prueba de aceptación ingresar tipo de estructura	134
Cuadro 89	Prueba de aceptación ingresar tipo de estructura erróneo.....	135
Cuadro 90	Prueba de aceptación modificar tipo de estructura	136
Cuadro 91	Prueba de aceptación ingresar ítems de estructura.....	137
Cuadro 92	Prueba de aceptación ingresar ítems erróneos.....	138
Cuadro 93	Prueba de aceptación ingreso de descuentos para retazos....	139
Cuadro 94	Prueba de aceptación ingreso erróneo de descuentos	140
Cuadro 95	Prueba de aceptación crear una proforma	141
Cuadro 96	Prueba de aceptación crear proforma con errores.....	142
Cuadro 97	Prueba de aceptación obtener lista de materiales	143
Cuadro 98	Prueba de aceptación crear un contrato	144

Cuadro 99 Prueba de aceptación modificar un contrato	145
Cuadro 100 Prueba de aceptación obtener entrega de materiales.....	146
Cuadro 101 Prueba de aceptación plan de corte.....	147
Cuadro 102 Prueba de aceptación Reporte fin de proyecto	148
Cuadro 103 Prueba de aceptación reporte de técnicos	149
Cuadro 104 Prueba de aceptación reporte de proformas	150

RESUMEN

El desarrollo constante de la tecnología, la oferta y demanda de productos y servicios, han logrado que los sistemas informáticos formen parte activa en el mejoramiento y desarrollo de las empresas, pese a este desarrollo ALGLASS no disponía de software que le ayude en tareas como el costeo ni en producción. La tarea de cotización se hacían en forma manual o en algunos casos con la ayuda de Excel, esto estaba expuestos a errores, olvidos, pérdida de tiempo y de recursos, en el proceso de producción no había ninguna herramienta de optimización. Por ello y para ayudar a cumplir con la ambiciosa Visión de ALGLASS, que pretende ser líder en la construcción de estructuras de aluminio y vidrio, se ha desarrollado esta aplicación, que contribuirá para que el proceso de cotización y producción sea mucho más rápido, óptimo y eficiente. La metodología seleccionada es XP, que permite el desarrollo ágil de aplicaciones, admite una comunicación constante con el cliente, cambios inmediatos y se hacen entregas parciales. La herramienta seleccionada para el desarrollo es Visual Studio, con conexión a una base de datos de Microsoft SQL Server. Como resultado de la implementación se ha logrado crear plantillas dinámicas para las cotizaciones, obtener detalles de corte para la producción, y registro de los materiales utilizados. Como consecuencia la tarea de cotizar es más rápida y exacta, en el proceso de producción se logró optimizar el recurso humano y materia prima, con esto se ha logrado cumplir con las expectativas del cliente.

PALABRAS CLAVE:

- **VISUAL STUDIO,**
- **SQL SERVER,**
- **XP.**

Abstract

The constant development of technology, supply and demand for products and services, have made computer systems active part in the improvement and development of enterprises, although this development ALGLASS no software available to help you in tasks such as production or costing. The task of trading is done manually or in some cases with the help of Excel, this was exposed to mistakes, forgetfulness, loss of time and resources in the production process was no optimization tool. For this reason and to help meet the ambitious vision ALGLASS, intended as a leader in the construction of aluminum and glass, it has developed this application, which will contribute to the process of trading and production much faster, optimal and efficient. The selected methodology is XP, which enables agile application development, support constant communication with the customer, immediate changes and partial deliveries are made. The selected development tool for Visual Studio, connected to a database of Microsoft SQL Server. As a result of the implementation has managed to create dynamic templates for quotes, details cut for production, and registration materials. As a result the task of trading is faster and more accurately, in the production process was optimized human resources and raw material, it has managed to meet customer expectations.

KEYWORDS:

- **VISUAL STUDIO,**
- **SQL SERVER,**
- **XP.**

CAPÍTULO 1

1 INTRODUCCIÓN

1.1 Visión general

“ALGLASS es una empresa familiar dedicada al negocio de comercialización aluminio y vidrio desde ya hace más de 15 años, con experiencia en el trabajo de ventanería, mamparas, cubiertas de vidrio y policarbonato, vidrio de seguridad, vidrio templado, todo tipo de mamparas y cortinas de baño con asesoría técnica.

Su fundador, el Arq. José Tejada, con más de 25 años en el negocio, quien desempeña actualmente el cargo de Gerente General ha sabido imprimirle a la empresa la seriedad, honestidad, cumplimiento y calidad que hasta ahora a ha llevado a ALGLASS ser una de las empresas líderes en la instalación de sistemas de ventanería y todo lo que se refiere a componentes y acabados para la construcción en vidrio”. (Alglass, 2013)

1.2 Visión de ALGLASS

“Proveer al mercado de la construcción un producto de excelencia que brinde confort, elegancia y seguridad a las viviendas y edificaciones de los ecuatorianos, manejando principios de integridad, creatividad y honestidad con la colaboración y el esfuerzo de nuestra gente”. (Alglass, 2013)

Con tan ambiciosa visión, ALGLASS pretende estar a la vanguardia de las denominadas Tecnologías de la Información y las Comunicaciones (TIC). La convergencia tecnológica del software y las telecomunicaciones dotan de una herramienta imprescindible para el progreso interno y reconocimiento externo de la compañía.

Con la inclusión de estándares tecnológicos al crear e implementar un sistema se podrán optimizar recursos, reducir costos, establecer una estructura organizacional proactiva, mejorar el desempeño y competencia de

sus colaboradores; aprovechando el sistemas informáticos y la tecnología existente.

1.3 Planteamiento del problema

La construcción de estructuras de aluminio y vidrio requiere varios productos dependiendo el tipo como son: accesorios, vidrios, y una amplia gama de piezas de aluminio.

Actualmente para realizar una cotización la persona a cargo revisa los planos con los diseños, medidas, y hace una lista manual o en Excel de los aluminios que se necesitarían para la construcción de dicha estructura, en la parte de los accesorios solo se los considera en forma global, en el caso de ser muchas estructuras el proceso se vuelve más complicado, y con más probabilidades de errores, puesto que se puede omitir aluminios accesorios o simplemente no sumar algún ítem, además de ser tedioso el proceso requiere de mucho tiempo, en algunos casos esta demora en la cotización puede resultar en la pérdida de un cliente.

En muchos de los casos luego de una primera cotización los clientes pueden solicitar un cambio ya sea en el tipo de estructuras o en el color del vidrio, esto puede requerir en algunos casos repetir el proceso íntegramente.

Si finalmente se llega a un acuerdo en el precio con el cliente, se entrega los planos al Técnico que es quien genera la lista de los materiales y accesorios que requerirá para la fabricación de dichas estructuras.

El hecho de entregar el material de acuerdo al requerimiento del Técnico implica que a veces pidan más material del que realmente necesitan tanto en accesorios como en aluminio o viceversa.

Cada tipo de aluminio debe ser cortado con un descuento exacto para poder ser ensamblado, este descuento lo hace manualmente cada Técnico lo que requiere tiempo y está sujeto a errores.

Al cortar el aluminio pueden dejar mucho desperdicio si no se tiene un detalle de corte adecuado, ya que las piezas de aluminio vienen de medidas estándar.

Actualmente ALGLASS cuenta con un Galpón distante del Almacén, destinado para la construcción, por lo que se hace imperioso que el material que se despacha para la fabricación sea lo más exacto posible o requerirá gastos adicionales de movilización.

Durante el proceso de construcción e instalación pueden presentarse imprevistos que requerirán material adicional los mismos que deberían ser registrados para tener control del material utilizado por proyecto.

Luego de la finalización de una obra se necesita saber cuál ha sido el costo real de material de un contrato así como de la mano de obra pagada para tomar decisiones a futuro en cuanto a precios y posibles descuentos.

Los Técnicos reciben su remuneración de acuerdo a una tabla de precios por metros cuadrados, luego de la finalización de cada obra se debe hacer manualmente este cálculo.

Sería importante también obtener algunos reportes de lo contratado y de lo proformado, para análisis.

1.4 Justificación

Para ALGLASS es necesaria la implementación de un sistema que ayude a mejorar **el proceso de cotización** ya que actualmente este proceso es un proceso manual, se espera que mediante la implementación del sistema se consiga:

- Agilizar el proceso de costeo, que actualmente requiere mucho tiempo.
- Disminuir los errores en cotización por omisiones de materiales teniendo estructuras preestablecidas.

- Permitir hacer cambios en las cotizaciones fácilmente cuando el cliente requiera. (cambios en colores de aluminio, espesores de vidrios u otro tipo de especificaciones)
- Facilitar el proceso de tal manera que la persona que cotice, con conocimientos básicos pueda realizar una cotización.

También se espera que en la **ejecución de proyectos** por medio del sistema se pueda:

- Obtener una lista exacta de los materiales que se necesitarán para la fabricación de las estructuras evitando que los Técnicos la realicen.
- Disminuir costos de movilización realizando un solo envío de materiales al Galpón de construcción.
- Obtener listas de medidas a las que deben ser cortados los perfiles con sus respectivos descuentos.
- Reducir el **desperdicio** generado en el corte de las piezas de aluminio, mediante la implementación de detalles de corte que indiquen al Técnico de qué manera cortarlas.
- Obtener un costo real al finalizar un proyecto mediante el registro del **material entregado**.
- Con los datos ingresados de las estructuras se aspira a tener un detalle para el **pago de los Técnico** de acuerdo a lo realizado.

1.5 Objetivos

1.5.1 Objetivo general

Desarrollar un Sistema para realizar cotizaciones, gestionar materiales, y optimizar recursos humanos y materia prima.

1.5.2 Objetivos específicos

- Realizar el análisis de requisitos funcionales del sistema.
- Diseñar la estructura del sistema y base de datos.

- Determinar los algoritmos que permitan optimizar los materiales.
- Construir el sistema con los módulos correspondientes.
- Realizar las pruebas correspondientes del sistema.

1.6 Alcance.

En este proyecto se desarrollarán los módulos que se detallan a continuación y se llegara hasta la etapa de implementación que comprende el ciclo de desarrollo de software.

- Módulo de Seguridades.
 - Manejo de usuarios
 - Asignación de Roles y Permisos
- Módulo de productos.
 - Manejo de Productos. Por tipos y colores
- Módulo de Costeo.
 - Creación de Estructuras por Tipo, material y Color.
 - Elaboración de Cotizaciones.
- Módulo de Entrega de Materiales
 - Manejo de Clientes
 - Generación de Listas de Materiales
 - Registro de Entregas
- Módulo de Planes de Corte.
 - Creación de Tabla de Descuentos por Piezas.
 - Generación de Plan de Corte por Tipo de estructura y material.
- Módulo Reportes
 - Reportes Costo Técnicos
 - Reporte de proformas
 - Reporte de contratos

- Reportes de consumos
- Reportes varios.

1.7 Metodología.

Por el tipo de Proyecto a realizar y por la necesidad de la empresa, la metodología que se adapta de una mejor manera es **Programación Extrema (XP)**.

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck (EcuRed, 2011). Es el más destacado de los procesos ágiles de desarrollo de software.

1.8 Herramientas.

Las herramientas que se utilizarán para el presente proyecto son:

- Plataforma Windows
- Microsoft Visual Studio Community.
- Microsoft SQL Server 2014 – Management Studio

1.9 Factibilidad.

1.9.1 Factibilidad Técnica

El desarrollador posee los conocimientos necesarios acerca de las herramientas, procedimientos y el método para llegar a la culminación total del proyecto llegando hasta la implementación.

Dentro de ALGLASS existen ya redes estructuradas conectadas a Internet con un dominio con sus respectivos equipos que servirán como base de hardware para el proyecto.

1.9.2 Factibilidad Operativa

ALGLASS. Se compromete a facilitar la información necesaria para la realización de este proyecto, entregando datos acerca de los procesos

actuales así también permitirá que se realicen entrevistas con personas específicas encargadas de procesos puntuales.

De igual manera brindará el apoyo necesario para la culminación del presente proyecto dentro de los plazos establecidos.

1.9.3 Factibilidad Económica

La siguiente tabla muestra el análisis económico que tendrá el desarrollo de este sistema.

Tabla 1
Factibilidad económica.

	CANTIDAD	VALOR/U	TOTAL
Microsoft Visual Studio Community 2015.	1	\$ -	\$ -
Microsoft SQL Server 2014 EXPRESS	1	\$ -	\$ -
COMPUTADORAS (EXISTENTE)	5	\$ -	\$ -
RED (EXISTENTE)	1	\$ -	\$ -
INTERNET (EXISTENTE)	1	\$ -	\$ -
PAPELERÍA	1	\$ 300,00	\$ 300,00
RECURSO HUMANO			
PROGRAMADOR	3	\$ 700,00	\$ 2 100,00
		TOTAL	\$ 2 400,00

CAPÍTULO 2

2 MARCO TEÓRICO

2.1 Aplicaciones de escritorio

Una aplicación es un programa de computadora, que se utiliza para realizar tareas u operaciones específicas. Son utilizadas por usuarios comunes son básicas y de fácil utilización.

Principalmente el uso de estas aplicaciones está ligado a solucionar problemas o simplificar operaciones complejas o que manejan muchos datos, el resultado de su utilización es ahorrar tiempo y dinero para el usuario con aplicaciones eficientes y fáciles de operar.

Contenido de las aplicaciones.

Pantallas de bienvenida.- Al momento de arrancar una aplicación suelen cargar una serie de librerías, datos, realizar conexiones, etc., por lo que hay una pantalla de bienvenida en la que se da a un poco de información sobre la aplicación.

Diálogos de entrada.- Muchas aplicaciones presentan diálogos de entrada donde se puede preguntar un nombre de usuario y contraseña.

Avisos.- Siempre que se realice una tarea se emite al usuario mensajes o señas de lo que está pasando, o que proceso está realizando la CPU.

Es importante que se cargue en las aplicaciones, la información necesaria y no información que tal vez no se llegue a utilizar

2.2 Framework

Define a un conjunto normalizado de conceptos, prácticas y criterios para encauzar un tipo de problemática particular, que sirve de guía para enfrentar y resolver nuevos problemas de índole similar.

Desde el punto de vista del desarrollo de software, un framework o infraestructura digital, comprende una estructura conceptual y tecnológica de soporte determinado, normalmente con artefactos o módulos de software concretos, con esta base otro proyecto de software puede ser desarrollado más fácilmente y organizado. Este puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, de esa manera ayuda a la unión de los diferentes componentes y al desarrollo del proyecto.

Constituye una arquitectura de software que organiza las relaciones generales de las entidades del dominio, y proporciona una estructura y una metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

2.2.1 Framework .NET

Este framework desarrollado por Microsoft, tiene especial énfasis en la transparencia de redes, es independiente a la plataforma de hardware y ayuda que el desarrollo de aplicaciones sea más rápido. En base a esto, Microsoft trata de desarrollar una estrategia horizontal que permita la integración de todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

Según Microsoft define los siguientes objetivos:

“Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.

Proporcionar un entorno de ejecución de código que minimiza los conflictos en el despliegue y versionado de software.

Ofrecer un entorno de ejecución de código que promueva la ejecución segura del mismo, incluso del creado por terceros desconocidos o que no son de plena confianza.

Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.

Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.

Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.” (Microsoft, 2016)

2.3 LINQ

Language Integrated Query es un proyecto de Microsoft que agrega a la plataforma .NET inicialmente a Visual Basic .NET y C# consultas nativas, semejantes a las de SQL. Muchos conceptos que LINQ ha introducido fueron originalmente examinados en C ω (el antecesor de C#).

LINQ precisa operadores de consulta estándar que permiten filtrar, enumerar y crear proyecciones de varios tipos de colecciones o datos almacenados usando la misma sintaxis. Estas colecciones pueden incluir vectores, clases enumerables, XML, bases de datos relacionales y orígenes de datos de terceros.

LINQ tiene como objetivo permitir que todo el código hecho en Visual Studio (incluidas las llamadas a bases de datos, datasets, XMLs) sean también orientados a objetos. Antes de la creación de LINQ, la manipulación de datos externos era más estructurado que orientado a objetos. También LINQ trata de facilitar y estandarizar el acceso a estos objetos.

En otras palabras LINQ es una importante innovación en Visual Studio 2008 y .NET Framework que elimina la distancia que separa el mundo de los objetos y el mundo de los datos.

Antiguamente, las consultas con datos se expresaban como cadenas sencillas, sin comprobación de tipos en tiempo de compilación ni

compatibilidad con IntelliSense (Autocompletar). Además, para cada tipo de origen de datos era necesario aprender un lenguaje de consultas diferente. LINQ convierte una consulta en una construcción de lenguaje dentro de C# y Visual Basic. Las consultas se escriben para colecciones de objetos con tipos de datos establecidos, utilizando palabras clave del lenguaje y operadores con los que se está familiarizado. (Microsoft, 2007)

2.3.1 Operadores de consulta estándar

Para el manejo de datos SQL, ocupa una serie de operadores que se detallan en la tabla descrita a continuación.

Cuadro 1
Tabla de operadores SQL

CLÁUSULA	DESCRIPCIÓN
Select	Realiza una proyección de la colección y selecciona los aspectos deseados de los elementos.
Where	Permite la definición de un conjunto de predicados que son evaluados para cada objeto en la colección, de manera que los objetos que no cumplan el predicado son excluidos del resultado. El predicado se especifica al operador como un delegado.
Sum / Min / Max / Average	Estos operadores reciben un parámetro opcional lambda, que toma un valor de cada elemento en la colección y lo usa para encontrar la suma, el mínimo, el máximo o el promedio respectivamente de todos los elementos en la colección.
Aggregate	Una versión generalizada de Sum/Min/Max. Este operador recibe una expresión lambda que especifica cómo serán combinados los valores para formar un resultado intermedio o final. De manera opcional, se Continúa..

CLÁUSULA	DESCRIPCIÓN
	<p>puede especificar un valor inicial, permitiendo que el tipo de la agregación pueda ser cualquiera. Además, una función de finalización, convirtiendo el resultado de la agregación en otro valor, que puede ser especificado.</p>
Join	<p>Realiza un inner join entre dos colecciones, basado en llaves coincidentes para los objetos de cada colección. Recibe dos funciones como delegados, una para cada colección, que se ejecutará sobre cada elemento de la colección para extraer su llave.</p>
GroupJoin	<p>Al igual que el anterior pero realiza un GroupJoin.</p>
Take / TakeWhile	<p>El operador Take selecciona los n primeros objetos de una colección, mientras que el operador TakeWhile, que recibe un predicado, selecciona solo los n primeros objetos que cumplan con este.</p>
Skip / SkipWhile	<p>Son los complementos de Take y TakeWhile respectivamente, se ignoran los primeros n elementos o los primeros n que cumplan el predicado especificado.</p>
OfType	<p>Para seleccionar solo los elementos de cierto tipo.</p>
Concat	<p>Concatena dos colecciones.</p>
OrderBy / ThenBy	<p>El operador OrderBy se utiliza para especificar el ordenamiento primario de los elementos en una colección de acuerdo con alguna llave. El orden por defecto es en orden ascendente, para invertir el orden se utiliza el operador OrderByDescending. ThenBy y</p> <p>Continúa</p>

CLÁUSULA	DESCRIPCIÓN
	ThenByDescending especifica el subsiguiente orden de los elementos. La función para extraer el valor de llave del objeto debe ser especificada por el usuario como un delegado.
Reverse	El operador Reverse invierte una colección.
GroupBy	<p>El operador GroupBy toma un delegado que extrae un valor de llave y devuelve una colección de objetos de tipo IGrouping<Key, Values>, para cada valor de llave distinta.</p> <p>Los IGrouping objetos pueden entonces utilizar para enumerar todos los objetos de un valor de llave en particular.</p>
Distinct	El operador Distinct elimina instancias duplicadas de un valor de llave en una colección. La función para recuperar el valor de llave que se retornará es un delegado.
Union / Intersect / Except	Estos operadores se utilizan para realizar una operación de unión, intersección y diferencia entre dos secuencias, respectivamente.
SequenceEqual	El operador SequenceEqual determina si todos los elementos de dos colecciones son iguales y tienen el mismo orden.
First / FirstOrDefault / Last / LastOrDefault	<p>Estos operadores reciben como parámetro un predicado. El operador First devuelve el primer</p> <p>Continúa..</p>

CLÁUSULA	DESCRIPCIÓN
	<p>elemento para el cual se satisface el predicado o lanza una excepción, si este no se satisface. El operador FirstOrDefault es como el primer operador, excepto que devuelve el valor predeterminado para el tipo de elemento (por lo general una referencia null) en caso de que no se satisfaga con el predicado.</p> <p>El operador Last recupera el último elemento para el cual se satisface el predicado, o se produce una excepción, si este no se satisface. El operador LastOrDefault devuelve el valor predeterminado si no existe tal elemento.</p>
Single	<p>El operador Single recibe como parámetro un predicado y devuelve el elemento que satisface con el predicado. Se produce una excepción, si ninguno o más de un elemento satisface con el predicado.</p>
ElementAt	<p>El operador ElementAt devuelve el elemento situado en un índice especificado de una colección.</p>
Any / All / Contains	<p>El operador Any determina si ninguno de los elementos de la colección satisface con el predicado. El operador All determina, si todos los elementos de la colección satisfacen con el predicado.</p> <p>El operador Contains determina si una colección contiene un elemento especificado.</p>
Count	<p>El operador Count cuenta el número de elementos en la colección dada.</p>

Fuente: (Microsoft, 2007)

2.3.2 Arquitectura del Linq

El siguiente diagrama proporciona una visión general de cómo se relacionan las tecnologías ADO.NET LINQ con lenguajes de programación de alto nivel y orígenes de datos habilitados para LINQ.

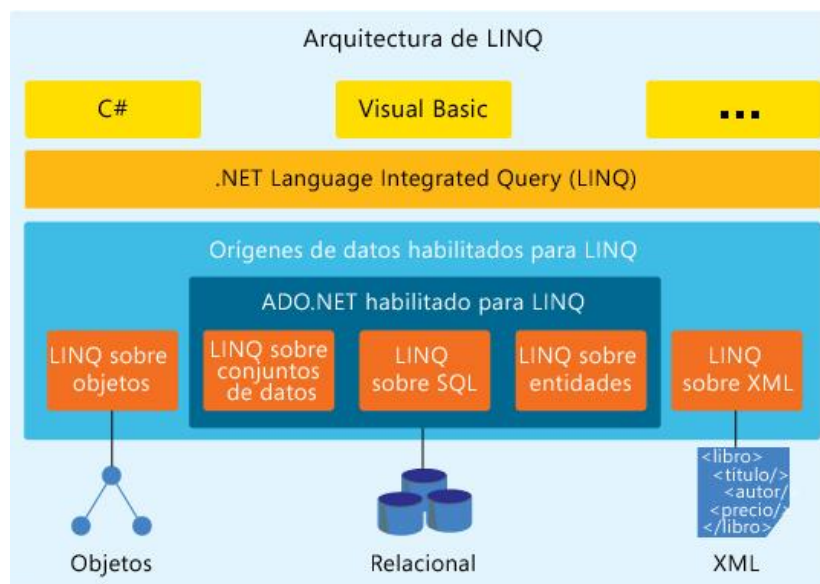


Figura 1 Arquitectura LINQ

2.4 Herramientas de Desarrollo

Las herramientas de desarrollo son los programas o aplicaciones que ayudan o facilitan el desarrollo de un sistema. La utilización de estas está dada por el conocimiento de las herramientas o la funcionalidad de estas.

Para el presente proyecto se ha seleccionado las siguientes herramientas.

2.4.1 Microsoft Visual Studio community 2015. (Microsoft, 2015)

Microsoft Visual Studio community 2015 es un completo IDE extensible y gratuito para crear aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube.

Esta edición de Visual Studio está disponible de forma gratuita para los desarrolladores individuales, desarrollo de código abierto, la investigación académica, la educación y los pequeños equipos profesionales.

En la tabla 3 se muestra un detalle de algunas de las funciones disponibles en las diferentes versiones de esta herramienta.

	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise	Visual Studio Test Professional	Plataformas de MSDN
+ Escenarios de uso admitidos	■■■	■■■	■■■	■■■	■■■
+ Depuración y diagnóstico	■■■	■■■	■■■	■■■	■■■
+ Herramientas de pruebas	■■■	■■■	■■■	■■■	■■■
+ Entorno de desarrollo integrado	■■■	■■■	■■■	■■■	■■■
+ Soporte de plataforma de desarrollo	■■■	■■■	■■■	■■■	■■■
+ Arquitectura y modelado	■■■	■■■	■■■	■■■	■■■
+ Lab Management	■■■	■■■	■■■	■■■	■■■
+ Características de Team Foundation Server	■■■	■■■	■■■	■■■	■■■
+ Herramientas de colaboración	■■■	■■■	■■■	■■■	■■■
+ Beneficios de la colaboración en equipo	Se incluye con todas las suscripciones				
+ Beneficios para los suscriptores	Se incluye con suscripciones de nube anuales y suscripciones estándar				
+ Beneficios de Visual Studio Dev Essentials	Gratis para todos los desarrolladores				

Figura 2 Comparativa Visual Studio
Fuente: (Microsoft, 2015)

Características principales de Visual Studio. (Microsoft, 2015)

Herramientas de codificación eficaces.- Codifique con confianza en su lenguaje, busque y corrija problemas de código rápidamente y refactorice con facilidad.

Depuración avanzada.- Ofrece una experiencia de depuración excelente independientemente del lenguaje que utilice (desde C#/VB y C++ hasta JavaScript, Python, XAML y HTML, todos los lenguajes admitidos son compatibles con la función de depuración.

Si su aplicación mezcla lenguajes, también puede efectuar la depuración de interoperabilidad en los límites de los lenguajes mediante la depuración de modo mixto.

Aplicaciones para dispositivos.- Independientemente de dónde se ejecute el código, Visual Studio puede depurarlo (iniciando una aplicación Windows

local en el escritorio o en el emulador de Android o adjuntando una instancia de Azure remota, un dispositivo iOS o una consola de videojuegos, o en cualquier explorador web.

Puede depurar problemas fuera de línea en su entorno de producción mediante funcionalidades como IntelliTrace y el análisis profundo de archivos de volcado.

Estado de inspección flexible.- En cuanto pause la aplicación donde desee, Visual Studio le ofrecerá muchas maneras de inspeccionar el valor de sus variables para formar o comprobar una hipótesis.

Supervise un valor mientras revisa su código; eche un vistazo a las variables locales y evalúe expresiones complejas (todo sin abandonar el depurador). Incluso puede consultar de manera interactiva niveles profundos de la estructura de datos.

Las excepciones son positivas.- Los defectos en el código y las situaciones inesperadas se manifiestan como excepciones. Las excepciones bloquearán la aplicación o simplemente se convertirán en errores difíciles de rastrear.

Durante la depuración con Visual Studio, puede configurar notificaciones cuando se produzcan excepciones e incluso elegir excepciones específicas de interés en las que centrarse. Recibirá una alerta y, desde ella, el proceso será tan sencillo como si se encontrase un punto de interrupción.

2.4.2 Visual Basic

Es un lenguaje de programación desarrollado por Alan Cooper para Microsoft. El lenguaje es un dialecto de BASIC, con incrementos importantes. Su primera versión fue presentada en 1991 el propósito fue el de simplificar la programación valiéndose de un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida la programación misma.

En 2001 Microsoft propone dejar el desarrollo basado en la API Win32 y pasar a trabajar sobre un framework o marco común de librerías independiente de la versión del sistema operativo, .NET Framework.

Visual Basic es un IDE (entorno de desarrollo integrado o en inglés Integrated Development Environment). Que está constituido por: un editor de código, un depurador (programa que corrige errores en el código fuente para que pueda ser compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina), y un Técnico de interfaz gráfica (no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

Características generales

Es un lenguaje de fácil aprendizaje tanto para principiantes como expertos, guiado por eventos, y centrado en un motor de formularios que facilita el rápido desarrollo de aplicaciones gráficas. Su sintaxis, deriva del antiguo BASIC, está ha sido ampliada al agregarse las características típicas de los lenguajes estructurados modernos.

Tiene una implementación limitada de la programación orientada a objetos (los propios formularios y controles son objetos), aunque admite el polimorfismo, no admite la herencia. No requiere de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee bibliotecas para usar bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, Dbase, Access, MySQL, SQL Server, etc) a través de ADO.

Su uso es principalmente para aplicaciones de gestión de empresas, por la rapidez con la que puede hacerse un programa que use una base de datos sencilla, a más de la abundancia de programadores en este lenguaje.

El compilador de Microsoft genera ejecutables que requieren una DLL para que funcionen. Además existen un gran número de bibliotecas (DLL) que proporcionan el acceso a muchas funciones del sistema operativo y la composición con otras aplicaciones. (EcuRed, 2016)

2.5 SQL (Escofet, 2007)

Es un lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales, que permite realizar diversos tipos de operaciones en las bases de datos. Una característica es el manejo del álgebra y el cálculo relacional que permite efectuar consultas a fin de recuperar fácilmente la información deseada de una base de datos, así como hacer cambios sobre la base de datos.

2.5.1 Características generales del SQL

El SQL es un lenguaje de acceso a bases de datos que utiliza la potencia y flexibilidad de los sistemas relacionales, admitiendo gran variedad de operaciones.

SQL es un lenguaje declarativo de “alto nivel” o “de no procedimiento”, que por su fuerte base teórica y su orientación al manejo de conjuntos de registros, permite una alta productividad en la codificación y la orientación a objetos. Es así que una sola sentencia sería equivalente a uno o más programas en un lenguaje de bajo nivel.

Como se mencionó otra de las características comunes en los lenguajes de alto nivel de acceso a bases de datos es que SQL es un lenguaje declarativo. Es decir, se debe especificar lo que se quiere y no cómo conseguirlo, por lo que una sentencia no implanta directamente un orden para la ejecución.

La eficiencia del Sistema de Gestión de la Base de datos (SGBD) puede verse gravemente afectado por el orden de ejecución interno de una sentencia, Por lo que el SGBD lleva a cabo una optimización previo a su ejecución. A veces, el uso de índices acelera una instrucción de consulta, pero hace más lenta la actualización de los datos. Dependiendo del motor de base de datos y de otros factores la optimización difiere sensiblemente.

Hay una ampliación de SQL conocida como FSQL (Fuzzy SQL, SQL difuso) que permite el acceso a bases de datos difusas, usando la lógica difusa. Este lenguaje difuso ha sido implementado en forma experimental y está rápidamente evolucionando.

2.6 Microsoft SQL Server 2014 Express (Malemeza, 2012)

“Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server es la opción de Microsoft frente a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.”

2.6.1 Características (Malemeza, 2012)

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada SQL Express Edition, que se distribuye en forma gratuita.

Para el manejo de SQL mediante líneas de comando se utiliza el SQLCMD

Microsoft lo detalla de la siguiente manera:

Motor de base de datos Es el servicio principal para almacenar, procesar y proteger los datos. Este servicio proporciona acceso controlado y

procesamiento rápido de transacciones para satisfacer los requisitos de las aplicaciones consumidoras de datos más exigentes. También proporciona compatibilidad para mantener una gran disponibilidad.

Analysis Services – Datos multidimensionales Analysis Services admite OLAP (On-Line Analytical Processing procesamiento analítico en línea) al permitir al usuario diseñar, crear y administrar estructuras multidimensionales que contienen datos agregados de otros orígenes tales como bases de datos relacionales.

Analysis Services – Minería de datos Analysis Services permite al usuario diseñar, crear y visualizar modelos de minería de datos. Estos modelos de minería de datos se construyen a partir de otros orígenes de datos por medio de una amplia variedad de algoritmos de minería de datos estándar.

Integration Services Es una plataforma para generar soluciones de integración de datos de alto rendimiento, lo que incluye paquetes que proporcionan procesamiento de extracción, transformación y carga para almacenamiento de datos.

Master Data Services Master Data Services es el origen de datos maestros para la organización. Mediante la integración de diferentes sistemas de análisis y operaciones, puede asegurarse de que todas las aplicaciones de la organización dependen de una fuente de información central y precisa. Con Master Data Services, crea un origen único de datos maestros y mantiene un registro de los datos, que cambian en el tiempo y que puede controlar.

Replicación La replicación es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de base de datos de una base de datos a otra, para luego sincronizar ambas bases de datos con el fin de mantener su coherencia.

Reporting Services Reporting Services ofrece funcionalidad empresarial de informes habilitados para Web con el fin de poder crear informes que

extraigan contenido a partir de una variedad de orígenes de datos, publicar informes con distintos formatos y administrar centralmente la seguridad y las suscripciones.

Integración de SharePoint SQL Server 2008 R2 proporciona una nueva capacidad de Business Intelligence de autoservicio a través de la integración con productos y tecnologías de SharePoint.

Service Broker Sirve a los desarrolladores de software a crear aplicaciones de base de datos escalables y seguras. Esta nueva tecnología de Motor de base de datos suministra una plataforma de comunicación basada en mensajes que permite a los componentes de aplicación independientes trabajar como un conjunto funcional.” (Microsoft, 2014)

2.7 Metodología XP (Extreme Programming) (EcuRed, 2010)

La programación extrema o eXtreme Programming es una metodología de desarrollo de software formulada por Kent Beck. En el desarrollo de software es el más destacado de los procesos ágiles. XP pone principalmente énfasis en la adaptabilidad que en la previsibilidad. A diferencia de las metodologías tradicionales los defensores de XP piensan que los cambios de requisitos son un aspecto natural, inevitable y hasta deseable en el desarrollo de proyectos. Consideran que definir todo los requisitos al inicio del proyecto es menos realista que adaptarse a los cambios en la cualquier fase del proyecto.

2.7.1 Valores

Los Valores de XP son: simplicidad, comunicación, retroalimentación (feedback), coraje y respeto

- Simplicidad:

Es la base de la programación extrema. Hay que simplificar el diseño para agilizar el desarrollo y así facilitar el mantenimiento. Lo complejo del código así como las frecuentes modificaciones de código por parte de diferentes desarrolladores hacen que la complejidad aumente por ello es necesario,

para mantener la simplicidad, refactorizar el código, que no es otra cosa que cambiar el código fuente o rescribir el mismo, sin alterar su comportamiento, ésta es la manera de mantener el código simple a medida que crece.

También la simplicidad está en la documentación, de esta manera el código debe comentarse en su justa medida, tratando que el código esté autodocumentado. Por ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres largos no disminuyen la eficiencia del código ni tampoco el tiempo de desarrollo gracias a las herramientas de autocompletado que existen actualmente. Si se usa la simplicidad se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.

- Comunicación:

Se realiza de diferentes formas. El código debe ser lo más simple posible, aunque sea complejo los programadores se deben esforzar por que sea claro. Es mejor el código autodocumentado que los comentarios ya que los comentarios pronto quedan desfasados con respecto al código a medida que este es modificado. Los comentarios deben estar en aquello que no va a variar. Hay comunicación entre los programadores gracias a la programación por parejas. Con el cliente la comunicación es fluida por ser parte del equipo de desarrollo. El cliente indica qué características tienen prioridad y para solucionar dudas debe estar disponible.

- Retroalimentación (feedback):

Con el cliente integrado en el proyecto, se conoce en tiempo real su opinión del estado del proyecto. Por ser ciclos muy cortos en los cuales se muestran resultados, es mínimo lo que hay que rehacer por no cumplir con los requisitos y por ello los programadores se pueden centrar en lo que realmente importa.

- Coraje o valentía:

La valentía implica diseñar y programar para hoy y no para mañana. Los programadores deben sentirse cómodos cuando sea necesario reconstruir el

código y esto requiere valentía. Hay que revisar el sistema existente y modificarlo si logramos que con ello los cambios futuro se hagan más fácilmente. También es necesaria la valentía para desechar un código: valentía para borrar código fuente obsoleto, sin importar el esfuerzo y tiempo que invertimos en crear ese código. Además, valentía significa persistencia: un programador en problema complejo puede permanecer por un día entero sin avanzar, y al día siguiente resolverlo rápidamente, sólo si es persistente.

- **Respeto:**

El respeto entre los miembros del equipo se debe demostrar pues los programadores no pueden hacer cambios que hagan que las pruebas existentes fallen o que demore el trabajo de sus compañeros. Los programadores deben tener respeto por su trabajo al estar esforzándose por la alta calidad, diseño óptimo y eficiente en el producto para la solución a través de la refactorización del código.

2.7.2 Características fundamentales

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** Se hacen mejoras pequeñas, unas tras otra.
- **Pruebas unitarias continuas,** frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- **Programación en parejas:** es recomendable que el desarrollo de tareas se realice entre dos en un mismo sitio. Se supone que escrito de esta manera es de mejor calidad.
- **Frecuente integración del equipo de programación con el cliente o usuario.** Es recomendable que un representante del cliente este junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir nueva funcionalidad.
- **Hacer entregas frecuentes.**

- **Refactorización del código**, es reescribir partes del código para mejorar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Es indispensable luego de ello las pruebas para evitar fallos.
- **Propiedad del código compartida**: No se desarrolla los módulos en grupos de trabajo distintos, por el contrario todo el personal puede corregir y aumentar cualquier parte del proyecto.
- **Simplicidad en el código**: La programación extrema enseña que es mejor hacer algo simple aunque esto implique un poco de trabajo adicional en posibles cambios, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son complementarias. Mientras mejor sea la comunicación es más fácil identificar que se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos necesidad habrá de comunicar sobre éste.

2.7.3 Primera Fase: Planificación del proyecto.

- **Historias de usuario**: Primero se debe definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; sin posibles algoritmos para su implementación ni de diseños de base de datos, etc. También se utilizan en la fase de pruebas, para confirmar si el programa satisface lo que especifica en las historia de usuario.
- **Release planning**: se debe crear un plan de publicaciones, en inglés “Release plan”, donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas de publicación de las versiones. Un “Release plan” es una planificación donde los desarrolladores y clientes definen los tiempo y las historias que se desarrollarán en cada versión. Después de un “Release plan” se debe tener que estar claros estos cuatro factores: los objetivos que se deben

cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado

- Iteraciones: se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al inicio de cada iteración los clientes seleccionarán las historias de usuario identificadas en el “Release planning” que serán implementadas.
- Velocidad del proyecto: La velocidad del proyecto representa la rapidez con la que se desarrolla el proyecto; estimarla es sencillo, basta con contar las historias de usuario que se pueden implementar en una iteración; de esta forma, se identificará cuántas historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos las tareas se puedan desarrollar en el tiempo del que dispone la iteración.
- Programación en pareja: Se aconseja la programación en pareja para mejorar la productividad y la calidad del software. El trabajo en pareja comprende a dos programadores trabajando en el mismo equipo; mientras
- Reuniones diarias. Es necesario reuniones diarias para que los desarrolladores expongan sus problemas, soluciones e ideas de forma conjunta. Estas reuniones deben ser fluidas y todos deben tener voz y voto.

2.7.4 Segunda Fase: Diseño.

- Diseños simples: Esta metodología sugiere que hay que conseguir diseños simples y sencillos. Se debe procurar hacer, todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que ayudara a ocupar menos tiempo y esfuerzo en el desarrollo.
- Glosarios de términos: Usar glosarios de términos y especificar en una forma correcta los nombres de métodos y clases facilitará la

comprensión del diseño y ayudará a sus posteriores modificaciones o aumentos, y a reutilizar el código.

- Riesgos: Para posibles que surjan riesgos durante el desarrollo por lo que la metodología aconseja una programación en pareja para que investiguen y minimicen al máximo el riesgo de algún problema.
- Funcionalidad extra: Un programa específico debe mantenerse sin añadiduras en la funcionalidad, puesto que ha sido analizado y creado para apegarse a las necesidades puntuales en este caso de Alglass, no se recomienda proyectarse para reutilizar o repotenciar en algo extra el programa, aunque se piense que en un futuro será utilizada. El desarrollo de funcionalidad extra es por tanto un desgaste de tiempo y recursos infructuoso a corto y largo plazo.
- Refactorizar. Se basa en la mejora y si fuere necesario modificación estructural de códigos. Para refactorizar se revisará de nuevo estos códigos procurando optimizar su funcionamiento, si solo se usa el existente sin consideraciones, podemos caer en una codificación inadecuada u obsoleta
- Tarjetas C.R.C. Son Tarjetas de Clase, Responsabilidad y Colaboración, usada para mantener una metodología centrada al desarrollo orientado a objetos, manteniendo un orden y claridad en la programación, que evita cometer errores a programadores menos experimentados.

Las tarjetas C.R.C representan objetos; indicando en la parte de arriba centrada, la clase a la que pertenece el objeto, en la columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad, según diagrama.

Cuadro 2 Tarjeta CRC

Nombre de la clase.	
Responsabilidades	Colaboradores

2.7.5 Tercera Fase: Codificación.

La codificación es el proceso de transformación de la información como los entiende el usuario a un lenguaje simbólico que será leído y entendido por el sistema. Es imprescindible que el mensaje que se transformará en un código venga de una fuente segura, y es donde el cliente forma parte del equipo de desarrollo. La presencia del cliente viene a ser trascendental a la hora de codificar una historia de usuario, estos crean dichas historias, que debemos codificar, coordinamos tiempos e implementación en las diferentes fases.

El cliente manifiesta en detalle la historia del usuario; lo que hará, su origen y finalización, esto previo a que se desarrolle la historia de usuario, para especificar los por menores de su funcionalidad. Es más, es el cliente el veedor cuando se ejecutan los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe ser técnica, ajustándose a estándares de codificación establecidos, y que den claridad y consistencia. La Programación bajo estándares, evita los enigmas en la comprensión y escalabilidad futura.

Los test se crear como una herramienta de verificación y avance seguro. Crear test que comprueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código con mayor garantía de que al integrar el sistema este sea consistente. Sin embargo es

primordial crear estos test antes de implementar un código, ya que nos ayuda a saber qué es exactamente lo que tiene que hacer el código, con la certeza de que se aprobarán los test puesto que se orientó el diseño para aprobarlos. Los test mejoran su rendimiento cuando se dividen en unidades pequeñas y que atiendan a una función, al hacerlo así, un test para cada unidad y luego desarrollar la unidad que será una pequeña tarea programada, poco a poco se irán cumpliendo las especificaciones del sistema y se va complementando la funcionalidad.

X.P, recomiendo la programación en pareja para confrontar el desafío de la singularidad y perfeccionar el sistema, ante la perspectiva de dos individuos.

X.P, además recomienda usar repositorios comunes de códigos por compañerismos, con el fin de ir aportando y validando los códigos publicados, este modelo de trabajo aporta con agilidad en la corrección de errores y en general en la mejora de calidad en pocas horas, siendo los test la forma que deben publicar adjunto al código y pasarla. Siendo la solidaridad del conocimiento una de las mayores fuentes de perfeccionamiento de los programadores, que harán uso de códigos ya trabajados y probados en las últimas versiones. Se debe tener cuidado en que este repositorio se maneje únicamente entre pareja de programadores para mantener el sistema consistente.

Es posible optar por un modelo de desarrollo colectivo, donde sean varios los programadores implicados en hacer estos aportes. Consideramos que cualquiera puede modificar o ampliar una clase de otro compañero programador si lo considerase necesario para luego subirla. De esta manera los programadores modifican códigos que no son suyos pero que deben pasar los test de funcionamiento definido a fin de que siempre se conserve la fidelidad del trabajo realizado.

Hay que estar claros que la optimización del código es de manera indiscutible necesaria para catalogar un sistema de calidad, sin embargo

esta tarea viene luego de que los requerimientos básicos estén en correcto y pleno funcionamiento.

X.P plantea la realización de un “Release plan” al cual apegarse de manera asidua para completar el proyecto. Si este requiriera más tiempo que el planificado, afirma que a menudo no logra cumplir con el proyecto a pesar de que se añada tiempo o se inviertan más recurso humano u otros recursos. Para X.P se debe volver a realizar un “Release plan” donde se proyecte concretar los nuevos tiempos de publicación y de velocidad del proyecto.

2.7.6 Cuarta Fase: Pruebas.

Pruebas o test es básico en la metodología X.P se presentan como exámenes de aprobación, es decir cada test comprueba el funcionamiento del códigos que se vaya implementando, y se realiza de manera escalonada.

Utilización de los test en X.P:

El entorno de pruebas donde se tomaran estos test debe ser aplicaciones específicas creadas para el desarrollo de pruebas.

Cada clase del sistema debe estar debidamente probado por el test cuidando que no se compruebe lo obvio.

Anteriormente explicamos la importancia de que los códigos pasen los test antes de implementarlos, siendo su creación previa al desarrollo de los códigos, de esta manera se asegura que cumplan su fin para el que fueron creados.

Los test deben ser individuales e independientes unos de otros, esto evitará que comprometamos el funcionamiento del sistema por redundancias cíclicas de error. Al crear un test aislado sin que tengan dependencia del código que en un futuro evaluará, asegurará la independencia del test respecto al código que evalúa.

Cuando los test se suben al repositorio, donde los códigos están acompañados del código que verifican, es una manera de comprobar su funcionamiento, ya que para ser publicado un código es validado mediante el test que lo acompaña. Ya estando en el repositorio, es seguro y de uso colectivo.

Los test también se usan para definir la refactorización e incluso para optimizar el código. Los test permiten identificar que un cambio en la estructura de un código no implica necesariamente cambiar su funcionamiento.

Test de aceptación. Sirven a los clientes como simuladores, los test de aceptación reflejan las distintas tareas en que se dividió una historia de usuario. Y es el usuario quien al hacer uso de este test asegurando el funcionamiento final de una determinada historia de usuario y si satisfactoriamente cumple su cometido.

Las funcionalidades de esta aplicación son específicas y concretas, por lo que se realizaran test que prueben las funcionalidades generales que debe cumplir el programa detallado en la descripción de requisitos.

CAPÍTULO 3

3 ANÁLISIS Y DISEÑO

3.1 PLANIFICACIÓN

Especificaciones de requisitos del sistema

Se ha clasificado los requerimientos proporcionados por Alglass dependiendo de la factibilidad de los mismos. De acuerdo al alcance de este proyecto tenemos lo siguiente:

3.1.1 Requisitos Funcionales

- Se necesita que el sistema tenga usuarios con diferentes accesos. Hay módulos a los que solo deben ingresar ciertos usuarios que tienen conocimiento adicional de ventanería.
- Los productos deben estar agrupados por familias. es importante que se identifique algunos productos como son los aluminios y los vidrios puesto que esto tienen un trato diferente.
- Se debe crear productos con sus propiedades, cada producto tiene sus características propias y el sistema debe permitir ingresar estos datos.
- El sistema debe proporcionar cotizaciones agrupadas por clientes, un cliente puede tener varias cotizaciones o contratos y se debe poder buscar o modificar.
- Se debe poder crear estructuras con diferentes materiales, un tipo de estructura puede ser armada con uno o más tipos de aluminios el sistema debe permitir escoger y crear estos tipos.
- A cada estructura se le debe asignar una lista de materiales, que son los necesarios para la construcción e instalación.
- El sistema debe proporcionar en base a la proforma, una lista de los materiales necesarios para la construcción de lo contratado.
- Se debe llevar un registro de los materiales entregados, el sistema debe emitir una orden para que se despachen los materiales.

- Se necesita que el sistema proporcione una lista de aluminio con sus respectivos descuentos, esto evitará que los técnicos hagan cálculos manuales.
- Se debe optimizar las piezas de aluminios por medio de detalles de corte, calculados por el sistema.
- Debe proporcionar reportes de costos totales de materiales, luego que se termine un proyecto se debe sacar un reporte de lo gastado, para realizar un análisis de costos.
- Debe actualizar sus precios por medio de listas de otro sistema contable. Se debe establecer un método para actualizar fácilmente los precios del sistema en base al sistema contable y de ventas.

3.1.2 Requisitos no funcionales

- Que el sistema se integre con el sistema contable existente Decisión Win (DW) para obtener precios. El sistema contable y de ventas está desarrollado en Fox Pro y tiene una base de datos bloqueada por el desarrollador.
- Que se haga descargos de inventario del sistema DW, debido a que la base de datos del sistema está bloqueada, esto no es factible.
- Que se redacte en un procesador de texto automáticamente una proforma detallada.

3.2 Historias de Usuario

De acuerdo a los requerimientos funcionales, se procede a elaborar con el cliente las historias de usuario.

Las historias de usuario nos darán las directrices para continuar con el desarrollo de la aplicación, y tener los parámetros para dividirla en algunas iteraciones.

3.2.1 MÓDULO DE SEGURIDADES

En este módulo se establecen los lineamientos para el ingreso al sistema.

En este módulo se tiene la siguiente historia de usuario.

Cuadro 3
Historia de Usuario 1

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre de Historia: Manejo de Usuarios	
Prioridad en Negocio: Media	Riesgo en desarrollo: Baja
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe registrar usuarios con sus datos para el ingreso al sistema.	
Observaciones: Los datos que se necesitan de los usuarios son básicos	

Cuadro 4
Tarea 1.1

Tarea	
Número: 1.1	Número de Historia: 1
Nombre de Tarea: Crear base de datos para usuarios	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para guardar información esencial de usuarios.	

Cuadro 5
Tarea 1.2

Tarea	
Número: 1.2	Número de Historia: 1
Nombre de Tarea: Crear formulario de ingreso de datos	
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para el ingreso y modificación de datos de los usuarios.	

3.2.2 MÓDULO DE PRODUCTOS

Para este módulo el cliente ha proporcionado las siguientes historias de usuario.

Cuadro 6
Historia de Usuario 2

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre de Historia: Colores para Cotizar	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe permitir poner colores a los productos para las cotizaciones.	
Observaciones: Estos datos son muy importante, ya que depende de esto las opciones.	

Cuadro 7
Tarea 2.1

Tarea	
Número: 2.1	Número de Historia: 2
Nombre de Tarea: Crear base de datos de colores para cotizar	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para guardar los colores para las cotizaciones.	

Cuadro 8
Tarea 2.2

Tarea	
Número: 2.2	Número de Historia: 2
Nombre de Tarea: Crear formulario para ingreso de colores.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para ingresar o modificar los colores.	

Cuadro 9
Historia de Usuario 3

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre de Historia: Ingreso de tipos de productos	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 1	Iteración Asignada: 1
Descripción: Cada producto está clasificado por tipo, como vidrio, aluminio, accesorios, etc.	
Observaciones: Es importante que los aluminios estén plenamente identificados para el sistema, para realizar planes de corte.	

Cuadro 10
Tarea 3.1

Tarea	
Número: 3.1	Número de Historia: 3
Nombre de Tarea: Crear base de datos para tipos de productos	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de tipos de productos. Con sus diferentes características.	

Cuadro 11
Tarea 3.2

Tarea	
Número: 3.2	Número de Historia: 3
Nombre de Tarea:	Crear formulario para ingresar los datos de los tipos de productos.
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema
Descripción: Se desarrolla un formulario para ingresar datos de los tipos de productos.	

Cuadro 12
Historia de Usuario 4

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre de Historia:	Ingreso de productos
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 3	Iteración Asignada: 1
Descripción: El sistema debe admitir el ingreso de productos con sus diferentes características. Y deben ser de diferentes colores, en algunos casos los precios difieren dependiendo del color	
Observaciones: El precio de los productos se quiere que se actualice con un reporte en Excel que proporciona el precio del sistema actual	

Cuadro 13**Tarea 4.1**

Tarea	
Número: 4.1	Número de Historia: 4
Nombre de Tarea: Crear base de datos para insumos, productos	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de insumos y productos con sus diferentes detalles.	

Cuadro 14**Tarea 4.2**

Tarea	
Número: 4.2	Número de Historia: 4
Nombre de Tarea: Crear formulario para ingresar las variables de los insumos.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para ingresar los insumos	

Cuadro 15
Tarea 4.3

Tarea	
Número: 4.3	Número de Historia: 4
Nombre de Tarea:	Crear formulario para ingresar el color y datos del producto.
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema
Descripción: Se desarrolla un formulario para ingresar los datos por productos, y un código de vinculación.	

3.2.3 MÓDULO DE COSTEO

Este es el modulo central y el más complejo, por ser de donde se parte para obtener el reto de datos o reportes.

Cuadro 16
Historia de Usuario 5

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre de Historia:	Ingreso De Clientes
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe ingresar clientes con todos los datos necesarios para crear un historial de clientes.	
Observaciones:	

Cuadro 17
Tarea 5.1

Tarea	
Número: 5.1	Número de Historia: 5
Nombre de Tarea: Crear base de datos para clientes	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de clientes.	

Cuadro 18
Tarea 5.2

Tarea	
Número: 5.2	Número de Historia: 5
Nombre de Tarea: Crear formulario para registrar clientes.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para registrar los datos de los clientes.	

Cuadro 19
Historia de Usuario 6

Historia de Usuario	
Número: 6	Usuario: Administrador
Nombre de Historia: Mano de obra	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Cada estructura tiene un costo de mano de obra que viene en función de los metros cuadrados. Este es el valor con el que se les para a los técnicos.	
Observaciones:	

Cuadro 20
Tarea 6.1

Tarea	
Número: 6.1	Número de Historia: 6
Nombre de Tarea: Crear base de datos para mano de obra	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro del costo de mano de obra.	

Cuadro 21
Tarea 6.2

Tarea	
Número: 6.2	Número de Historia: 6
Nombre de Tarea: Crear formulario para registrar mano de obra.	
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para registrar los datos del costo de mano de obra.	

Cuadro 22
Historia de usuario 7

Historia de Usuario	
Número: 7	Usuario: Administrador
Nombre de Historia: Tipos de estructuras	
Prioridad en Negocio: Media	Riesgo en desarrollo: Media
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe permitir crear estructuras con sus diferentes tipos.	
Observaciones: Una estructura es: Una ventana, una puerta, etc.	

Cuadro 23
Tarea 7.1

Tarea	
Número: 7.1	Número de Historia: 7
Nombre de Tarea: Crear base de datos para tipos de estructuras	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de estructuras con sus diferentes tipos.	

Cuadro 24
Tarea 7.2

Tarea	
Número: 7.2	Número de Historia: 7
Nombre de Tarea: Crear formulario para ingresar las estructuras con sus tipos.	
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para crear las estructuras con sus tipos.	

Cuadro 25
Tarea 7.3

Tarea	
Número: 7.3	Número de Historia: 7
Nombre de Tarea: Crear base de datos para materiales Estructuras	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de estructuras con sus diferentes tipos.	

Cuadro 26
Tarea 7.4

Tarea	
Número: 7.4	Número de Historia: 7
Nombre de Tarea: Crear formulario para ingresar materiales de las estructuras.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para crear las estructuras con sus tipos.	

Cuadro 27
Historia de usuario 8

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre de Historia: Detalle de estructuras	
Prioridad en Negocio: Alto	Riesgo en desarrollo: Media
Puntos Estimados: 3	Iteración Asignada: 1
Descripción: El sistema debe permitir crear estructuras con la lista de lo que se necesita para construirlas.	
Observaciones: Cada estructura necesita varios productos para su creación. Y cada uno con distintas cantidades	

Cuadro 28
Tarea 8.1

Tarea	
Número: 8.1	Número de Historia: 8
Nombre de Tarea: Crear base de datos para detalle de estructuras	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de la lista de materiales por estructura.	

Cuadro 29
Tarea 8.2

Tarea	
Número: 8.2	Número de Historia: 8
Nombre de Tarea:	Crear formulario para ingresar la lista de materiales por tipo de estructura.
Tipo de tarea: Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema
Descripción: Se desarrolla un formulario para crear las estructuras con sus tipos.	

Cuadro 30
Historia de usuario 9

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre de Historia:	Descuentos por piezas.
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe permitir crear tablas de descuentos por tipo de perfil.	
Observaciones: Los aluminios para ser cortados necesitan que se haga un descuento en su medida.	

Cuadro 31
Tarea 9.1

Tarea	
Número: 9.1	Número de Historia: 9
Nombre de Tarea: Crear base de datos para descuentos	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para registrar los descuentos por piezas.	

Cuadro 32
Tarea 9.2

Tarea	
Número: 9.2	Número de Historia: 9
Nombre de Tarea: Crear formulario de descuentos.	
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se crea un formulario para ingresar los descuentos de los diferentes tipos de perfiles de aluminio.	

Cuadro 33
Historia de usuario 10

Historia de Usuario			
Número:	10	Usuario:	Administrador
Nombre de Historia:		Proforma de estructuras	
Prioridad en Negocio:	Alta	Riesgo en desarrollo:	Alto
Puntos Estimados:	3	Iteración Asignada:	1
Descripción:			
El sistema debe permitir crear proformas de estructuras de aluminio.			
Observaciones:			
Las proformas constan de estructuras de aluminio así como materiales adicionales que se pueden necesitar.			

Cuadro 34
Tarea 10.1

Tarea			
Número:	10.1	Número de Historia:	10
Nombre de Tarea:		Crear base de datos para proformas	
Tipo de tarea	Diseño de BD	Puntos Estimados:	1
Fecha Inicio:		Fecha Fin:	
Responsable: Diseñador de Base de Datos			
Descripción:			
Se diseña base de datos para el registro de las proformas.			

Cuadro 35
Tarea 10.2

Tarea	
Número: 10.2	Número de Historia: 10
Nombre de Tarea: Crear formulario para crear proforma.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario para elaborar una proforma. Esta debe permitir incluir medidas de ventanas. Seleccionar color, datos de vidrios y adicionales.	

Cuadro 36
Tarea 10.3

Tarea	
Número: 10.3	Número de Historia: 10
Nombre de Tarea: Crear formulario para seleccionar materiales adicionales.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se desarrolla un formulario que permita ingresar el precio de los materiales adicionales	

3.2.4 MÓDULO DE ORDENES DE ENTREGA

Todo material para egresar de la bodega debe tener un registro y un documento

Cuadro 37
Historia de usuario 11

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre de Historia: Contratos.	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe permitir pasar una proforma a contrato.	
Observaciones: Cuando una proforma se hace contrato el sistema asigna un técnico y los materiales utilizados.	

Cuadro 38
Tarea 11.1

Tarea	
Número: 11.1	Número de Historia: 11
Nombre de Tarea: Crear base de datos para Contratos	
Tipo de tarea: Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para registrar los contratos por cliente.	

Cuadro 39
Tarea 11.2

Tarea	
Número: 11.2	Número de Historia: 11
Nombre de Tarea: Crear formulario de Contratos.	
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se crea un formulario para ingresar los datos de un contrato. Los contratos parten de una proforma.	

Cuadro 40
Historia de usuario 12

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre de Historia: Lista de materiales	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: El sistema debe generar una lista de los materiales necesarios para elaborar un proyecto.	
Observaciones: Es importante que la lista proporcionada contenga los códigos del sistema contable para poder ser descargada.	

Cuadro 41
Tarea 12.1

Tarea	
Número: 12.1	Número de Historia: 12
Nombre de Tarea:	Crear interface que permita seleccionar la proforma contratada.
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema.
Descripción:	
Se crea la interface para poder seleccionar la proforma contratada.	

Cuadro 42
Tarea 12.2

Tarea	
Número: 12.2	Número de Historia: 12
Nombre de Tarea:	Crear base de datos para las listas.
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador de BD
Descripción:	
Se crea una BD para guardar las listas de materiales.	

Cuadro 43
Tarea 12.3

Tarea	
Número: 12.3	Número de Historia: 12
Nombre de Tarea: Crear lista de materiales.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema	
Descripción: Se crea una lista de materiales a partir de una proforma especificada..	

Cuadro 44
Historia de usuario 13

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre de Historia: Entrega de materiales	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimados: 4	Iteración Asignada: 1
Descripción: El sistema debe proporcionar órdenes de entrega de materiales asignadas a un técnico.	
Observaciones: Estas órdenes se deben vincular a un contrato para poder obtener el reporte final	

Cuadro 45
Tarea 13.1

Tarea	
Número: 13.1	Número de Historia: 13
Nombre de Tarea: Crear base de datos para Técnicos	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de los técnicos.	

Cuadro 46
Tarea 13.2

Tarea	
Número: 13.2	Número de Historia: 13
Nombre de Tarea: Crear interface para Ingreso de técnicos.	
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se crea la interface ingresar datos de los técnicos que realizaran los proyectos.	

Cuadro 47
Tarea 13.3

Tarea	
Número: 13.3	Número de Historia: 13
Nombre de Tarea: Crear base de datos para Entregas de Materiales	
Tipo de tarea Diseño de BD	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador de Base de Datos	
Descripción: Se diseña base de datos para el registro de las entregas de materiales.	

Cuadro 48
Tarea 13.4

Tarea	
Número: 13.4	Número de Historia: 13
Nombre de Tarea: Crear interface para generar Órdenes de entrega.	
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se crea la interface para obtener órdenes para entregas. Se asigna a un técnico por proyecto.	

3.1.1 MÓDULO DE PLANES DE CORTE

Este módulo debe permitir obtener una lista de todos los retazos, necesarios para construir un proyecto, estos deben clasificarse por código de aluminio.

Cuadro 49
Historia de usuario 14

Historia de Usuario			
Número:	14	Usuario:	Administrador
Nombre de Historia:		Generar planes de corte.	
Prioridad en Negocio:	Alta	Riesgo en desarrollo:	Alto
Puntos Estimados:	3	Iteración Asignada:	1
Descripción:			
El sistema debe generar un detalle de corte para optimizar material.			
Observaciones:			
El sistema debe entregar un detalle de como cortar las piezas de aluminio			

Cuadro 50
Tarea 14.1

Tarea			
Número:	14.1	Número de Historia:	14
Nombre de Tarea:		Crear base de datos para detalle de corte.	
Tipo de tarea	Diseño de sistema	Puntos Estimados:	1
Fecha Inicio:		Fecha Fin:	
Responsable:		Diseñador del sistema.	
Descripción:			
Se la tabla para guardar los datos del plan de corte.			

Cuadro 51
Tarea 14.2

Tarea	
Número: 14.2	Número de Historia: 14
Nombre de Tarea:	Crear interface para seleccionar proforma para detalle de corte.
Tipo de tarea Diseño de sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema.
Descripción: Se crea una interface que permita seleccionar la estructura e ingresar medidas exactas.	

Cuadro 52
Tarea 14.3

Tarea	
Número: 14.3	Número de Historia: 14
Nombre de Tarea:	Crear detalles de corte.
Tipo de tarea Diseño del sistema.	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable:	Diseñador del sistema.
Descripción: Se crea reportes por tipos de perfiles con su detalle de corte.	

3.2.5 MÓDULO DE REPORTES

Los reportes son una herramienta importante en la toma de decisiones, por ello son necesarios los siguientes reportes:

- Reporte de Técnicos.- Este reporte contiene lo construido por un técnico en un lapso de tiempo.
- Reporte de Finalización de proyectos.- para saber si una obra fue rentable o no, se necesita un reporte que recolecte toda la información, de lo gastado en la terminación de un proyecto.
- Reporte de proformas.- Obtiene un reporte de lo proformado en un lapso de tiempo, con este reporte se puede conocer la aceptación de clientes.
- Reporte de Contratos.- Se puede saber cuánto se ha contratado en un lapso determinado.

Cuadro 53
Historia de usuario 15

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre de Historia: Reportes de costos y finalizaciones.	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimados: 4	Iteración Asignada: 1
Descripción: El sistema debe entregar reportes de técnicos, de fin de proyectos, proformas, y contratos.	
Observaciones:	

Cuadro 54
Tarea 15.1

Tarea	
Número: 15.1	Número de Historia: 15
Nombre de Tarea: Crear reportes de técnicos.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se obtiene reportes de áreas construidas por técnico.	

Cuadro 55
Tarea 15.2

Tarea	
Número: 15.2	Número de Historia: 15
Nombre de Tarea: Crear reportes de finalización de proyectos.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se obtiene reportes de costos totales de proyectos.	

Cuadro 56
Tarea 15.3

Tarea	
Número: 15.3	Número de Historia: 15
Nombre de Tarea: Crear reportes de Proformas.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se obtiene reportes de proformas realizadas en un lapso de tiempo.	

Cuadro 57
Tarea 15.4

Tarea	
Número: 15.4	Número de Historia: 15
Nombre de Tarea: Crear reportes de Contratos.	
Tipo de tarea Diseño del sistema	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Responsable: Diseñador del sistema.	
Descripción: Se obtiene reportes de contratos realizadas en un lapso de tiempo.	

3.3 Plan de publicaciones.

De acuerdo a las historias de usuario, y realizando el respectivo análisis de dificultad, se establece en el plan realizar las publicaciones de acuerdo al siguiente cuadro.

Tabla 2
Plan de Publicaciones

N°	HISTORIAS	INICIO	FIN
1	Manejo de Usuarios	09-oct	13-oct
2	Colores para cotizar	14-oct	18-oct
3	Ingreso de tipos de productos	19-oct	23-oct
4	Ingreso de productos	24-oct	01-nov
5	Ingreso De Clientes	02-nov	06-nov
6	Mano de Obra	07-nov	11-nov
7	Tipos de estructuras	12-nov	16-nov
8	Detalle de estructuras	17-nov	25-nov
9	Descuentos por piezas	26-nov	30-nov
10	Proforma de estructuras	01-dic	09-dic
11	Lista de materiales	10-dic	16-dic
12	Contratos	17-dic	22-dic
13	Entrega de materiales	23-dic	28-dic
14	Generar planes de corte	29-dic	04-ene
15	Reportes de costos y finalizaciones	05-ene	08-ene

3.4 Iteraciones

El proyecto total se divide en seis iteraciones, con los siguientes tiempos de publicación.

Tabla 3
Iteraciones

ITERACIONES	HISTORIAS	INICIO	FIN	DURACIÓN (DÍAS)
ITERACIÓN 1	MÓDULO DE SEGURIDADES	09-oct	18-oct	10
ITERACIÓN 2	MÓDULO DE PRODUCTOS	19-oct	01-nov	14
ITERACIÓN 3	MÓDULO DE COSTEO	02-nov	09-dic	38
ITERACIÓN 4	MÓDULO DE ORDENES DE ENTREGA	10-dic	28-dic	19
ITERACIÓN 5	MÓDULO DE PLANES DE CORTE	29-dic	04-ene	7
ITERACIÓN 6	MÓDULO DE REPORTES	05-ene	08-ene	4

3.5 Nomenclatura

La nomenclatura que se utilizará va de acuerdo a la siguiente tabla referencial.

Cuadro 58
Prefijos

Prefijo	Objeto	Ejemplo
cbx	Combo box and drop down list box	cboProvincias
chk	Checkbox	chkLectura
clp	Picture clip	clpToolbar

Continúa..

Prefijo	Objeto	Ejemplo
cmd	Command button	cmdOk
com	Communications	comFax
dat	Data control	datBiblio
dir	Directory list box	dirSource
dlg	Common dialog control	dlgFileOpen
fil	File list box	filSource
frm	Form	frmEntry
fra	Frame (3d)	fraStyle
grd	Grid	grdPrices
img	Image	imgIcon
lbl	Label	lblHelpMessage
lin	Line	linVertical
lst	List box	lstPolicyCodes
mnu	Menu	mnuFileOpen
pic	Picture	picVGA
spn	Spin control	spnPages
txt	Text Box	txtLastName
tmr	Timer	tmrAlarm

Fuente: (Microsoft, 2016)

3.6 Tarjetas CRC

Estas tarjetas son indispensables para determinar, en las clases que métodos serán necesarios desarrollar.

3.6.1 Módulo de seguridades

Cuadro 59
Tarjeta CRC clsUsuarios

Nombre de la Clase:	clsUsuarios
Responsabilidad	Colaboración
Cargar usuarios	funIngresarUsuario Ingresa datos del usuario
Ingresar Usuarios	
Modificar usuarios	funModificarUsuario Modificar datos del usuario
Listar Usuarios	
Asignar rol a usuarios	funEliminarUsuario Elimina usuarios
	funListarUsuarios Obtiene lista de Usuarios

3.6.2 Módulo de productos.

Cuadro 60
Tarjeta CRC clsProductos

Nombre de la Clase:	clsProductos
Responsabilidad	Colaboración
	funIngresarInsumos Ingresa datos de insumo
	funModificarInsumos modifica los datos del insumo
Cargar Tipos de Productos	
Ingresar tipos de productos	funIngresarColorInsumos Ingresa un color a los insumos
Modificar tipos de productos	
Cargar Productos	funIngresarTiposInsumos ingresa los tipos de insumos
Ingresar productos	
Modificar productos	funModificarTipoInsumos modifica los tipos de insumos

3.6.3 Módulo de costeo

Cuadro 61
Tarjeta CRC clsClientes

Nombre de la Clase:	clsClientes
Responsabilidad Cargar Clientes Ingresar Clientes Modificar Clientes Eliminar Clientes	Colaboración funIngresarClientes ingresa los datos de un cliente. funModificarClientes modifica los datos de un cliente. funlistarClientes entrega una lista de los clientes ingresados funEliminarCliente elimina un cliente.

Cuadro 62
Tarjeta CRC clsEstructuras

Nombre de la Clase:	clsEstructuras
Responsabilidad Ingresar Grupo de Estructuras Modificar Grupo de Estructuras Listar grupo de estructuras. Cargar Tipos Estructuras Ingresar Tipos Estructuras Modificar Tipos Estructuras	Colaboración funIngresarGrupoEstructuras .-Ingresa datos de un grupo de estructuras. funModificarGrupoEstructura.- modifica los datos de un grupo. funListarGrupoEstructura Lista los grupos ingresados. funIngresarTipos.- Ingresa los datos de los tipos funModificarTipos.- Modifica los datos del tipo. funlistarTipos.- Lista los tipos de estructuras.

Cuadro 63
Tarjeta CRC clsProforma

Nombre de la Clase:	clsProforma
Responsabilidad	Colaboración
Cargar Proformas	funIngresarProforma.- Ingresar el encabezado de una proforma.
Ingresar Proformas	funModificarProforma.- Modificar los datos de la proforma.
Modificar Proformas	funListarProforma.- lista las proformas creadas.
Buscar Proformas	funIngresarDetalleProforma.- ingresa el detalle de una proformas.
Ingresar los detalles de las proformas.	funModificarDetalleProforma.- Modificar el detalles de una proformas.

Cuadro 64
Tarjeta CRC clsContrato

Nombre de la Clase:	clsContrato
Responsabilidad	Colaboración
Cargar Contratos	funlistarContrato.- Ingresar los datos de un contrato.
Ingresar Contratos	funIngresarContrato.- Ingresar los datos de un contrato.
Modificar Contratos	funModificarContrato.- Modificar los datos del contrato.
Poner un contrato como inactivo.	funDesactivarContrato.- Cambia el estado de un contrato.

3.6.4 Módulos de órdenes de entrega

Cuadro 65
Tarjeta CRC clsListaMateriales

Nombre de la Clase:	clsListaMateriales
Responsabilidad	Colaboración
Generar Lista de materiales Presentar Lista de Materiales Eliminar Lista de materiales	GeneralistaDeMateriales.- Obtiene una lista de materiales de una proforma. EliminarListaMateriales.- Elimina los materiales de una proforma. funListarMateriales.- Devuelve los materiales de una proforma.

3.6.5 Módulo de planes de corte

Cuadro 66
Tarjeta CRC clsPlanCorte

Nombre de la Clase:	clsPlanCorte
Responsabilidad	Colaboración
Crear lista de retazos Eliminar lista de retazos	funlistarAluminiosparaPlan.- Obtiene una lista de todos los retazos para la contruccion. funEliminarListaEstructura.- elimina un plan de corte creado.

3.7 Diseño de base de datos.

Para cumplir con los requerimientos del sistema, se estructura la base de datos de acuerdo a la siguiente ilustración.

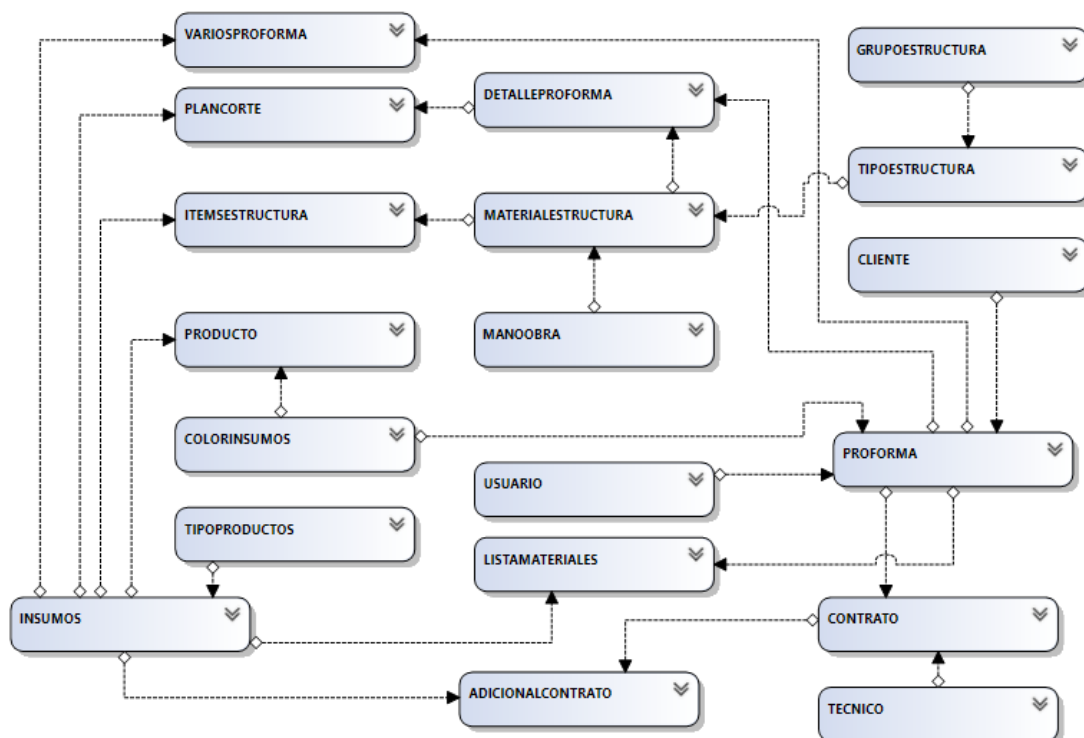


Figura 3 Diseño de Base de Datos

3.8 Modelo físico

Se diseña un modelo físico que cubre las necesidades de información, y parámetros para obtener los resultados y reportes deseados.

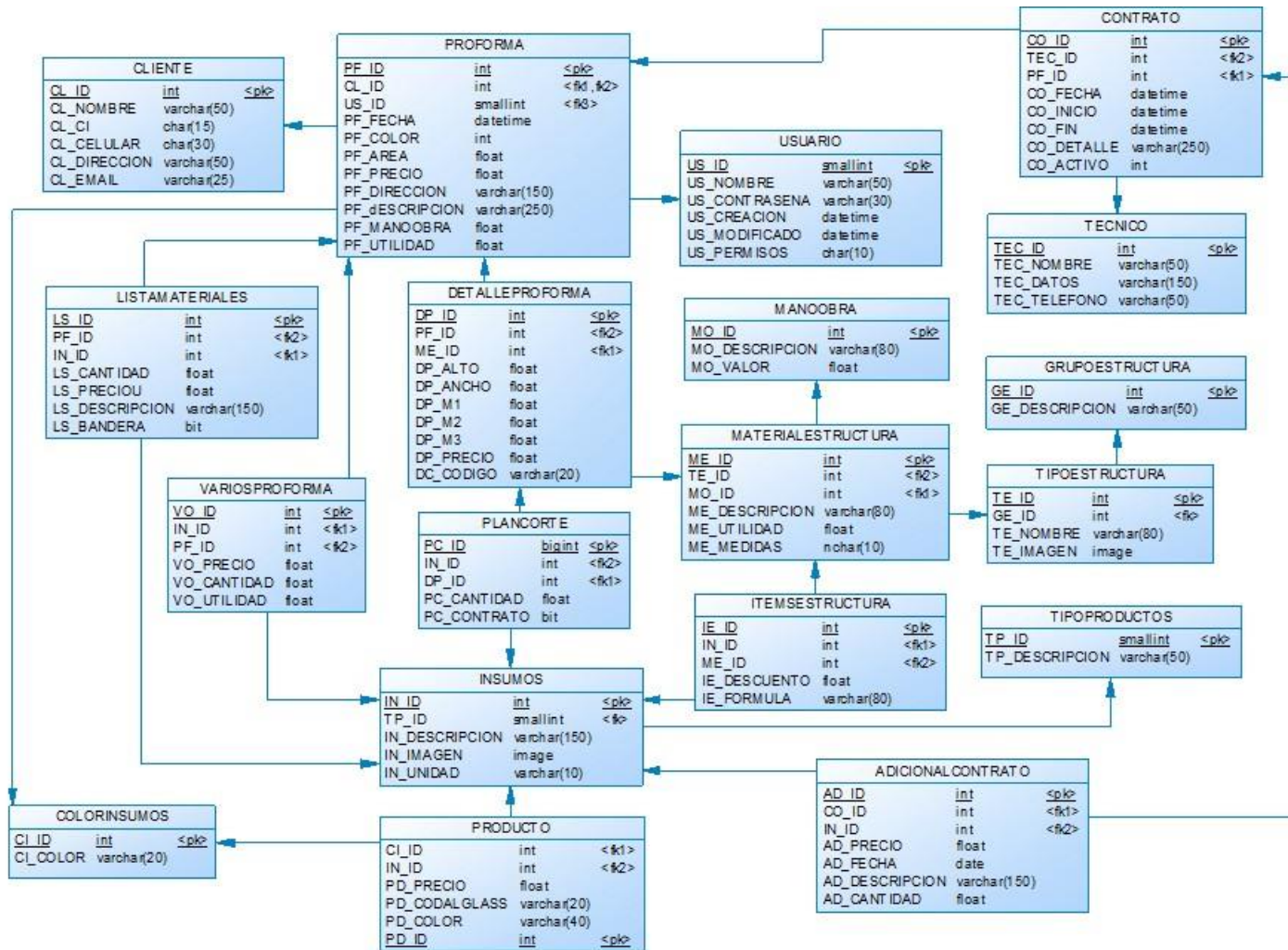


Figura 4 Modelo Físico

CAPÍTULO 4

4 CODIFICACIÓN DEL SISTEMA

4.1 Historia de Usuario 1. Manejo de Usuarios

Tarea 1.1 Crear base de datos para usuarios.

Para el registro de usuarios es necesario guardar el nombre, su contraseña y la fecha en la que se crea o se modifica su contraseña, así como los permisos que tiene este usuario.

USUARIO		
<u>US_ID</u>	smallint	<pk>
US_NOMBRE	varchar(50)	
US_CONTRASENA	varchar(30)	
US_CREACION	datetime	
US_MODIFICADO	datetime	
US_PERMISOS	char(10)	

Figura 5 Tabla Usuario

Tarea 1.2 Crear formulario de ingreso de datos

Este formulario permite el ingreso en forma explícita de los datos.

Figura 6 Formulario de Usuarios

Creación de Funciones Necesarias:

Función para el ingreso de datos de usuarios:

```
Function funIngresarUsuario(varUsuario As String, varContraseña As String, varFecha As Date, varPermisos As String) As Boolean
```

Función para modificar datos de usuarios:

```
Function funModificarUsuario(codigo As Integer, varUsuario As String, varContraseña As String, varFecha As Date, varPermisos As String) As Boolean
```

Función para Eliminar un usuario:

```
Function funEliminarUsuario(varIdUsuario As Integer) As Boolean
```

Opciones del formulario:

Nuevos usuarios

- Permitir crear nuevos usuarios
- Asigna un código y permitir ingresar los datos del usuario.
- Se debe escoger los permisos a los cuales tendrá acceso el usuario.
- Con el botón guardar ingresa los datos a la base de datos.
- Si no están llenos los campos emite un mensaje de error.

Modificar usuario

- Al hacer clic en la tabla permite modificar los datos existentes.
- Con el botón modifica los datos de la base de datos.
- Si no están llenos los campos emite un mensaje de error.

Eliminar Usuario

- Se debe seleccionar en la tabla el usuario que se va a eliminar
- Y con el botón eliminar se borra el usuario

Nota.- Si el usuario tiene registros en la base de datos no se puede eliminar.

4.2 Historia de Usuario 2. Colores para cotizar

Tarea 2.1 Crear base de datos para colores para cotizar

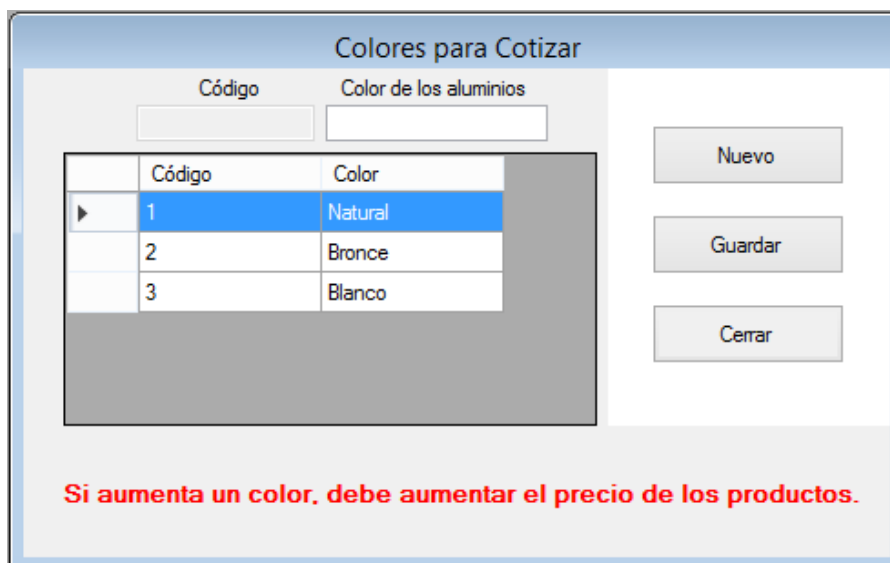
Para realizar las proformas se debe establecer los colores con los que se dispondrán para la creación de estructuras, se crea una tabla que contenga un código y el nombre del color.

COLORINSUMOS		
<u>CI_ID</u>	int	<pk>
CI_COLOR	varchar(20)	

Figura 7 Tabla Color Insumos

Tarea 2.2 Crear formulario para ingreso de colores.

En este solo se necesita ingresar el color.



Código	Color
1	Natural
2	Bronce
3	Blanco

Si aumenta un color, debe aumentar el precio de los productos.

Figura 8 Formulario Colores

Creación de Funciones Necesarias:

Función para el ingreso de color de datos de usuarios:

Function `funIngresarColorInsumos(descripcion As String) As Boolean`

Función para modificar Colores:

`Function funModificarColorInsumos(codigo As Integer, color As String) As Boolean`

Opciones del formulario:

Nuevo Color

- Opción Nuevo
- Ingrese el nuevo color
- Opción guardar ingresa los datos.

Modificar Color

- Seleccione en la tabla el color que se desea modificar
- Con Opción guardar modifica los datos existentes
- En caso de no ingresar ningún color emite un error.

4.3 Historia de Usuario 3. Ingreso de tipos de productos

Tarea 3.1 Crear base de datos para tipos de productos

Los insumos están agrupados de acuerdo a su clasificación estos pueden ser, Aluminios, vidrios, tornillos, etc.

Estos grupos se utilizan para poder manejar en forma separada el vidrio, que se debe poner al final por metro cuadrado, y el aluminio que es del que se hace los planes de corte.

La tabla contiene un código y una descripción.

TIPOPRODUCTOS		
<u>TP_ID</u>	<u>smallint</u>	<pk>
TP_DESCRIPCION	varchar(50)	

Figura 9 Tabla Tipo de Producto

Tarea 3.2 Crear formulario para ingresar los datos de los tipos de productos.

Se crea el formulario con la lista de tipos de productos.

Código	Descripción
1	Aluminios
2	Tomillería
3	Accesorios
4	Vidrios
5	Viniles

Figura 10 Formulario ingreso de Productos

Creación de Funciones Necesarias:

Función para el ingreso de Tipos de productos:

Function funIngresarTiposProductos(descripcion **As String**) **As Boolean**

Función para modificar Tipo Productos:

Function funModificarTipoProductos(codigo **As Integer**, descripcion **As String**) **As Boolean**

Opciones del formulario:

Nuevo tipo de Producto

- Opción Nuevo Tipo.
- El sistema asigna un código.
- Ingrese el nuevo Tipo.
- Opción guardar ingresa los datos.

Modificar Color

- Seleccione en la tabla el tipo que se desea modificar.
- Con Opción guardar modifica los datos existentes.

En caso de no ingresar ningún tipo de producto emite un error.

4.4 Historia de Usuario 4. Ingreso de productos

Tarea 4.1 Crear base de datos para insumos, productos

Los insumos son los materiales de los que se componen las estructuras en estos no se define ningún color ni precio solo las características del mismo.

La tabla Insumos tiene un código, el id del Tipo, una descripción la unidad de medida y una imagen.

La tabla Producto, contiene un código, el código del insumo al que pertenece, el precio, un color específico, y un id del color con el que se asociará el precio para la cotización.

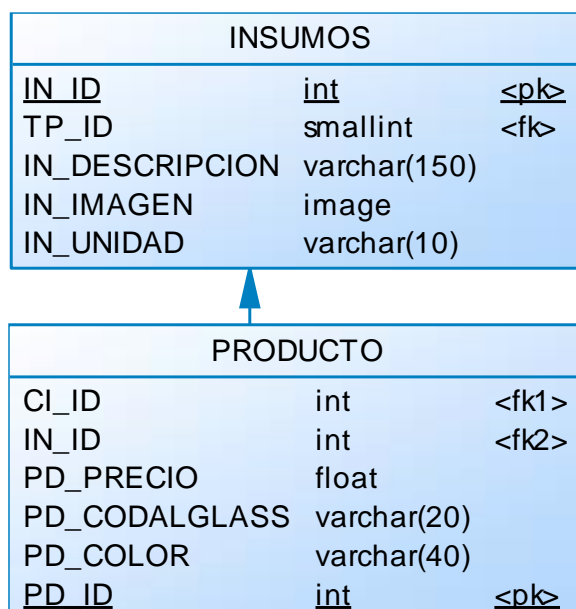


Figura 11 Tablas de Productos e Insumos

Tarea 4.2 y 4.3 Crear formulario para ingresar las variables de los insumos.

El formulario permite la inserción de los datos en las dos tablas, incluida una imagen.

Manejo de Insumos

Código del insumo: 2 << Anterior Siguiente >> Tipo de Producto: Aluminios

Descripción: Fijo (S-100) (1705-1400)FISA Unidad de Medida: ML

Colores de Aluminios						
Color del Insumo	Código Alglass	Precio	Natural	Bronce	Blanco	
BLANCO	F1705BL64	\$ 4.383	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
BRONCE	F1705BR64	\$ 4.297	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
NATURAL	F1705NA64	\$ 4.297	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Agregar Color Eliminar Color

perimetral fijo

- Ref.- 1705
- Peso.- 0.629 kg/m

Figura 12 Formulario de Insumos

En este formulario se cumplen dos tareas de las historias de usuario, en la parte superior se guarda las características del insumo y en la inferior los productos con sus colores, códigos y la asociación del color para la cotización.

Creación de Funciones Necesarias:

Función para el ingreso Insumos:

Function funIngresarInsumos(tipo As Integer, descripcion As String, unidad As String, imagen As Image) As Integer

Función para modificar Insumos:

Function funModificarInsumos(codigo As Integer, tipo As Integer, descripcion As String, unidad As String, imagen As Image) As Boolean

Función para Buscar Insumos:

`Function funBuscarInsumo(strbuscado As String) As Array`

Función para Ingresar Colores (Productos):

`Function funIngresarColorInsumos(coloreId As Integer, InsumoId As Integer, precio As Double, codigoAlglass As String, color As String, productoId As String) As Boolean`

Función para Eliminar Colores (Productos):

`Sub funEliminarProducto(CodigoInsumo As Integer)`

Opciones del formulario:

Nuevo producto.

- Seleccionar el tipo de producto en el combobox.
- Ingresar una descripción para el insumo.
- Seleccionar una unidad de medida.
- En la tabla se debe ingresar los colores que tiene ese insumo y activar el Checkbox con el que se asociará para la cotización.
- Se puede eliminar ítems o aumentar según sea necesario.
- Para ingresar una imagen del producto se debe copiar en el porta papeles y luego opción pegar.
- Y se selección la opción guardar.

Modificar insumos.

- Si ya existen insumos ingresados se puede escoger la opción buscar
- Escribir la descripción
- Seleccionar la opción aceptar.
- Se modifica los datos deseados
- Y se selección la opción guardar.

Se debe asignar un precio a cada color de las columnas, caso contrario no se guardarán los cambios.

Si se ingresa un nuevo color para cotizar se asocia un precio para cada insumo

4.5 Historia de Usuario 5. Ingreso De Clientes

Tarea 5.1 Crear base de datos para clientes

Los datos del cliente que se necesitan guardar son: el nombre su cédula o ruc, su número celular, su dirección y su dirección de correo.

CLIENTE		
<u>CL_ID</u>	int	<pk>
CL_NOMBRE	varchar(50)	
CL_CI	char(15)	
CL_CELULAR	char(30)	
CL_DIRECCION	varchar(50)	
CL_EMAIL	varchar(25)	

Figura 13 Tabla Cliente

Tarea 5.2 Crear formulario para registrar clientes.

El formulario de clientes permite insertar todos los datos del cliente. El código se pone de manera automática.

Figura 14 Formulario Cliente

Creación de Funciones Necesarias:

Función para el ingreso Clientes:

```
Function funIngresarClientes(varNombre As String, varCI As String,  
varCelular As String, varDireccion As String, varEmail As String) As Integer
```

Función para modificar Clientes:

```
Function funModificarClientes(IdCliente As Integer, varNombre As String,  
varCi As String, varCelular As String, varDireccion As String, varEmail As  
String) As Integer
```

Función para Buscar Clientes:

```
Function funBuscarClientes(strCliente As String) As Array
```

Función para Eliminar Clientes:

```
Function funEliminarCliente(varIdCliente As Integer) As Boolean
```

Opciones del formulario:

Nuevo cliente.

- El código lo asigna el sistema automáticamente.
- Se ingresa los datos solicitados en el formulario.
- Y se selección la opción guardar.

Modificar clientes.

- Si ya existen clientes ingresados se puede escoger la opción buscar
- Escribir el nombre o apellido
- Seleccionar la opción aceptar.
- Se modifica los datos deseados
- Y se selección la opción guardar.

Eliminar cliente.

- Si ya existen clientes ingresados se puede escoger la opción buscar
- Escribir el nombre o apellido
- Seleccionar la opción aceptar.
- Opción eliminar
- Esto eliminará definitivamente al cliente. Si no tiene proformas.

4.6 Historia de Usuario 6. Mano De Obra

Tarea 6.1 Crear base de datos para mano de obra

Los técnicos cobran por metro cuadrado en esta tabla se debe guardar esta información.

La tabla debe tener un código, una descripción del tipo de estructura u un precio.

MANOOBRA		
<u>MO_ID</u>	int	<pk>
MO_DESCRIPCION	varchar(80)	
MO_VALOR	float	

Figura 15 Tabla Mano de Obra

Tarea 6.2 Crear formulario para registrar mano de obra.

El formulario lista los costos ingresados y permite su inserción y modificación.

The screenshot shows a window titled "Costo de Mano de Obra". At the top, there are three input fields labeled "Código", "Descripción", and "Costo". Below these fields is a table with the following data:

	Código	Descripción	Valor
▶	1	Ventanas	\$ 12,00
	2	Puertas Corredizas	\$ 12,00
	3	Puerta Batiente	\$ 17,00
	4	Cubiertas con estructura	\$ 6,00
	5	Puerta colgante	\$ 15,00

To the right of the table are three buttons: "Nuevo", "Guardar", and "Cerrar".

Figura 16 Formulario par Mano de Obra

Creación de Funciones Necesarias:

Función para el ingreso Mano de Obra:

Function funIngresarManoObra(descripcion **As String**, valor **As Double**)
As Boolean

Función para modificar Mano de Obra:

Function funModificarManoObra(codigo **As Integer**, descripcion **As String**,
valor **As Double**) **As Boolean**

Opciones del formulario:

Nuevo Valor de mano de obra

- Opción Nuevo.
- El sistema le asigna un código.
- Ingrese la descripción y el valor.
- Opción guardar ingresa los datos.

Modificar Valor de Mano de Obra

- Seleccione en la tabla el valor o descripción a modificar.
- Con Opción guardar modifica los datos existentes

En caso de no ingresar ningún dato emite un error.

Se debe considerar que costo que se ingresa es por metro cuadrado de construcción de dicha estructura.

4.7 Historia de Usuario 7. Tipos de estructuras

Tarea 7.1 Crear base de datos para tipos de estructuras

Las estructuras están agrupadas y sub agrupadas en un grupo y tipo por ejemplo: Grupos (Puertas batientes, Puertas corredizas, ventanas fijas, etc.), Tipos.- Puertas Batientes (Sin divisiones, con división intermedia, etc.)

Grupo.- Las clasifica en forma global como: Puertas Batientes, puertas corredizas, etc.

Tipo: Constituye un sub grupo_ Puerta batiente con divisiones, puerta batiente sin divisiones, etc.

La tabla grupo estructura debe tener un id y una descripción.

La Tabla tipo estructura debe tener el id, id del grupo, un nombre y una imagen que ayude a reconocer si es la opción deseada.

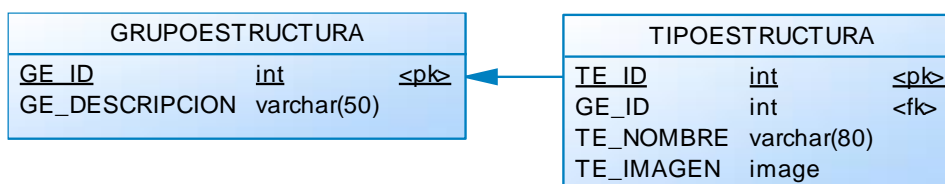


Figura 17 Tablas Grupo de estructuras y tipos de estructuras

Tarea 7.2 Crear formulario para ingresar las estructuras con sus tipos y grupos

El formulario de grupos lista los existentes y permite modificar e insertar.

El formulario 'Grupos de Estructuras' muestra un cuadro de diálogo con los siguientes elementos:

- Encabezado: 'Grupos de Estructuras' con botones de minimizar, maximizar y cerrar.
- Formulario de entrada: Campos para 'Código' y 'Descripción'.
- Tabla de datos:

	Código	Descripción
▶	1	Puertas Batientes
	2	Puertas Corredizas
	3	Ventanas Corredizas
	4	Ventanas Fijas
	5	Ventanas Proyectable
- Botones de acción: 'Nuevo', 'Guardar' y 'Cerrar'.

Figura 18 Formulario para grupo de estructuras

Figura 19 Formulario para tipo de estructuras

Creación de Funciones Necesarias:

Función para el ingresar Grupo de Estructuras:

`Function funIngresarGrupoEstructuras(descripcion As String) As Boolean`

Función para modificar Grupo de Estructuras:

`Function funModificarGrupoEstructura(codigo As Integer, Descripcion As String) As Boolean`

Función para el ingresar Tipo de Estructuras:

`Function funIngresarTipos(codigo As Integer, Grupo As Integer, descripcion As String, imagen As Image) As Boolean`

Función para modificar Tipo de Estructuras:

`Function funModificarTipos(codigo As Integer, Grupo As Integer, descripcion As String, imagen As Image) As Boolean`

Función para Buscar Tipo de Estructuras:

`Function funBuscartipo(strbuscado As String) As Array`

Opciones del formulario:

Nuevo Grupo de estructura

- Opción Nuevo Grupo.
- El sistema le asigna un código.
- Ingrese la descripción del grupo.
- Opción guardar ingresa los datos.

Modificar grupo de estructura

- Seleccione en la tabla la descripción a modificar.
- Con Opción guardar modifica los datos existentes

Nuevo Tipo de estructura

- Opción Nuevo Tipo.
- El sistema le asigna un código.
- Ingrese la descripción del tipo.
- Copia una imagen al portapapeles
- Clic en el botón Pega la imagen
- Opción guardar ingresa los datos.

Modificar grupo de estructura

- Seleccione Buscar
- Seleccione el tipo de estructura a modificar.
- Modifique los datos necesarios.
- Con Opción guardar modifica los datos existentes

Es de ayuda que en el dibujo de la estructura se ingrese las medidas que se ocuparán como se muestra en el gráfico.

4.8 Historia de Usuario 8. Detalle de estructuras

Tarea 8.1 y 9.1 Crear base de datos para detalle de estructuras

Cada estructura pertenece a un grupo y sub tipo, y estas se componen de diverso insumos.

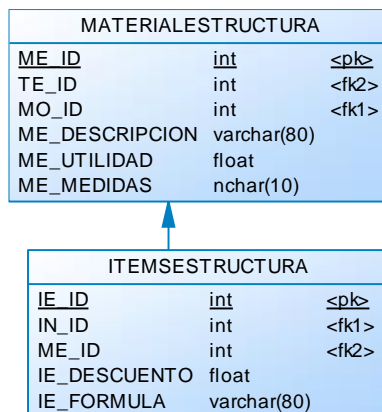


Figura 20 Tablas de Material Estructuras e ítems de estructuras

La tabla contiene un código, los id de los grupos y de los tipos, el id de la mano de obra, un porcentaje de utilidad, y un parámetro para saber qué medidas son necesarias.

Tarea 8.2 Crear formulario para ingresar la lista de materiales por tipo de estructura.

Los materiales contienen una descripción. Por ejemplo: Estándar, serie 100, serie 200, etc. Dependiendo de esto se les asocia un valor de mano de obra y una utilidad.

1) Una fija una corrediza

Código Materiales

	Código	Descripción
▶	1	ESTANDAR

Nuevo

Guardar

Cerrar

Figura 21 Formulario de Materiales

Figura 22 Formulario para Datos de Materiales

Creación de Funciones Necesarias:

Función para el ingresar Material Estructura:

```
Function funIngresarMateriales(descripcion As String, varTipoE As Integer) As Boolean
```

Función para modificar Material Estructura:

```
Function funModificarMaterialesE(codigo As Integer, varTipoE As Integer, Optional Descripcion As String = "", Optional varManoObra As Integer = vbNull, Optional varUtilidad As Double = vbNull) As Boolean
```

Función para listar Materiales Estructura:

```
Function funlistarMateriales(Tipo As Integer) As Array
```

Función para Obtener Ítems de la estructura:

```
Function funBuscarDetalleInsumo(strbuscado As String) As Array
```

Función para Ingresar ítems a las Estructuras:

```
Function funIngresarItems(insumoid As Integer, MaterialId As Integer, descuento As Double, formula As String) As Boolean
```

Función para Modificar ítems a las Estructuras:

```
Function funModificarItemsE(ItemId As Integer, insumoid As Integer, MaterialId As Integer, descuento As Double, formula As String) As Boolean
```

Opciones del formulario:

Nuevo Material para la estructura

- Se debe seleccionar el grupo
- Seccionar el tipo
- Clic botón Aumentar o Modificar material
- Se ingresa una nueva descripción.
- Se guarda los datos ingresados.

Modificar Material para la estructura

- Se debe seleccionar el grupo
- Seccionar el tipo.
- Clic botón Aumentar o Modificar material.
- Se modifica los datos.
- Se guarda los datos ingresados.

Nueva Estructura

- Se debe seleccionar el grupo
- Seccionar el tipo
- Seleccionar el Material
- Se le asigna un valor de mano de obra.
- Y se pone una utilidad para esta estructura.
- Se ingresa los ítems necesarios.
- Se guarda los datos ingresados.

Modificar Estructura

- Se debe seleccionar el grupo.
- Seccionar el tipo.
- Seleccionar el Material.

- Se modifica los datos deseados.
- Y se pone una utilidad para esta estructura.
- Se ingresa o modifica los ítems necesarios.
- Se guarda los datos ingresados.

FÓRMULAS

En el campo fórmula se debe ingresar variables como: alto, ancho, m1, m2 y m3.

Alto.- Representa la altura de una estructura

Ancho.- Representa el ancho de una estructura

M1, m2 y m3.- Son medidas adicionales que pueden ser utilizadas para la creación de una estructura

Las fórmulas pueden contener expresiones matemáticas que incluyan las variables. Por ejemplo:

“Alto/2”

“(alto+ancho)*40”

4.9 Historia de Usuario 9. Descuentos por piezas

Tarea 9.2 Crear formulario para descuentos.

Este formulario es un poco complejo y es importante que quien lo ocupe tenga conocimientos de cómo se construye y que materiales se necesitan, así también conocer que descuentos se deben hacer al momento de cortar los aluminios.

Figura 23 Formulario para Descuentos

Se da como solución crear en los ítems de la estructura un campo de descuento para cada ítem de aluminio. Estos valores deben ser ingresados en metros.

Esto evita la creación de otra tabla y más formularios, esto solo se habilita para aluminios.

4.10 Historia de Usuario 10. Proforma de estructuras

Tarea 10.1 Crear base de datos para proformas

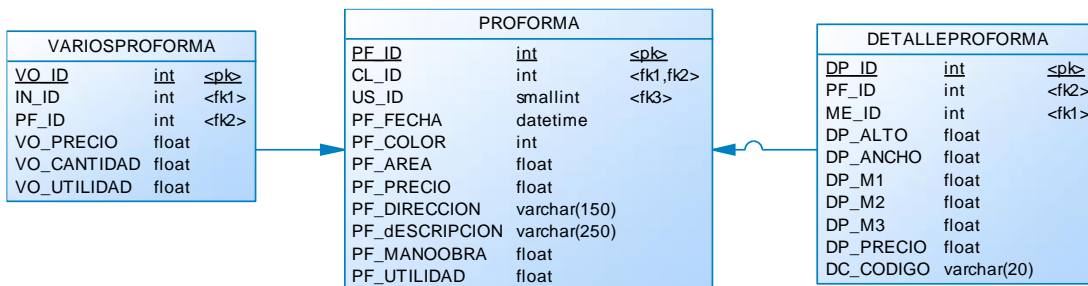


Figura 24 Tablas para Proformas

Tarea 10.2 Crear formulario para crear proforma.

Las proformas tienen un encabezado con los datos informativos como el nombre del cliente, una descripción, la dirección y la fecha.

Figura 25 Encabezado del formulario Proforma

Una proforma se puede transformar en contrato y cuando pasa esto, la proforma ya no se puede modificar.

En el encabezado se asigna un cliente para la proforma, así también se tiene una descripción y dirección de la obra que se va a cotizar.

Código	Grupo	Tipo	Material	Alto	Ancho	M1	M2	M3	Área	Precio
1	Ventanas Corredizas	Una fija una Corrediza	7 Perfiles	1,56	2	0,6	0	0	3,12	\$ 114,01
2	Ventanas Corredizas	Una fija una Corrediza	7 Perfiles	1,56	2	0,6	0	0	3,12	\$ 114,01
3	Ventanas Corredizas	Una fija una Corrediza	7 Perfiles	1,56	2	0,6	0	0	3,12	\$ 114,01
4	Ventanas Corredizas	Una fija una Corrediza	7 Perfiles	0,64	0,64	0,6	0	0	0,41	\$ 32,63
5	Ventanas Corredizas	Dos corredizas una fija Intermedia	7 perfiles	2	3,2	0	0	0	6,40	\$ 205,10
6	Puertas Corredizas	Colgante tubular	Tubular 1762	2	3,4	1	0	0	6,80	\$ 405,58
6	Ventanas Fijas	Sin Divisiones	Estándar	1,8	2	0	0	0	3,60	\$ 102,20

Figura 26 Formulario para Estructuras de proforma

Creación de Funciones Necesarias:

Función para ingresar una nueva Proforma:

```
Function funIngresarProforma(varPfld As Integer, varCiid As Integer,
varUsId As Integer, varFecha As Date, varColor As Integer, varArea As
Double, varPrecio As Double, varDescripcion As String, varDireccion As
String) As Boolean
```

Función para modificar Proforma:

```
Function funModificarProforma(varPfld As Integer, varCiid As Integer,
varUsId As Integer, varFecha As Date, varColor As Integer, varArea As
Double, varPrecio As Double, varDescripcion As String, varDireccion As
String) As Boolean
```

Función para listar Proformas:

```
Function funListarProforma(varIdProforma As Integer) As Array
```

Función para ingresar estructuras a la proforma:

```
Function funIngresarDetalleProforma(varPfld As Integer, varMeID As
Integer, varCodigo As String, varAlto As Double, varAncho As Double, varM1
As Double, varM2 As Double, varM3 As Double, varPrecio As Double) As
Boolean
```

Función para modificar las estructuras de la proforma:

```
Function funModificarDProforma(varDpld As Integer, varPfld As Integer,
varMeID As Integer, varCodigo As String, varAlto As Double, varAncho As
Double, varM1 As Double, varM2 As Double, varM3 As Double, Optional
varPrecio As Double = -1) As Boolean
```

Opciones del formulario:**Nueva Proforma.**

- Seleccione el cliente para la Proforma.
- Ingrese la Descripción y Dirección.
- Guarde los datos

Modificar Proforma.

- Buscar la proforma para modificar el cliente para la Proforma.
- Ingrese la Descripción y Dirección.
- Guarde los datos

Ingresar Estructuras a la Proforma.

- Seleccione un grupo de estructuras.
- Seleccione el tipo de estructuras.
- Seleccione el material con el que se hará la estructura
- Ingrese un código y las medidas.
- Clic en Ingresar

Modificar estructuras de una Proforma.

- Doble clic en la estructura a modificar.
- Modificar las medidas o código.
- Clic en Modificar.

Las proformas solo pueden ser modificadas mientras no sean contrato.

Tarea 10.3 Crear formulario para seleccionar materiales adicionales.

The screenshot displays a software interface for managing proforma items. At the top, there are three tabs: 'Estructuras', 'Otros Adicionales', and 'Vidrios'. Below the tabs is a search bar with the label 'Insumo' and a 'Ingresar' button. A table with the following columns is visible: 'Producto', 'Cantidad', 'Precio', and 'Utilidad'. The table contains one row: 'Entrecierre P.C. (...)' with a quantity of 3, a price of 3.9, and a utility of 30. Below the table is an 'Eliminar Item' button. At the bottom of the interface, there is a section for 'Color Estructuras' with a dropdown menu set to 'Natural', a 'Total Proforma' field showing 591.73, and an 'Actualizar Precio' button. On the right side, there is a summary table:

Estructuras	587,83
Otros	3,9
Vidrios	0
Total Proforma	591,73

Figura 27 Formulario para Adicionales de la proforma

Figura 28 Formulario para Vidrios

Creación de Funciones Necesarias:

Función para Ingresar otros ítems a la proforma:

Function funIngresarVProforma(varInId **As Integer**, varPfid **As Integer**, varPrecio **As Double**, varCantidad **As Double**, varUtilidad **As Double**) **As Double**

Función para Modificar otros ítems de la proforma:

Function funModificarVProforma(varVold **As Integer**, varInId **As Integer**, varPfid **As Integer**, **Optional** varPrecio **As Double** = -1, **Optional** varCantidad **As Double** = 0, **Optional** varUtilidad **As Double** = -1) **As Double**

Función para Eliminar otros ítems de la proforma:

Sub funEliminarVarios(varIdVario **As Integer**)

Función para Listar otros ítems de la proforma sin vidrios:

Function funListarVariosProformaSinVidrio(varPfid **As Integer**) **As Array**

Función para Listar Vidrios ítems de la proforma:

`Function funListarVidriosProforma(varPfid As Integer) As Array`

Opciones del formulario:

Ingresar productos adicionales.

- En la pestaña Otros Adicionales.
- Seleccionar Ingresar Insumo
- Ingresar los datos Cantidad y Utilidad para este producto
- Clic en Ingresar.

Modificar productos adicionales.

- En la pestaña Otros Adicionales.
- En la tabla de accesorios adicional hacer doble clic para seleccionar el Insumo a Modificar
- Modificar los datos Cantidad y Utilidad para este producto
- Clic en Modificar.

Eliminar un producto adicional.

- En la pestaña Otros Adicionales.
- En la tabla de insumos seleccionar el Insumo a Eliminar
- Clic en el botón Eliminar Ítem

Ingresar Vidrios.

- En la pestaña Vidrios.
- Seleccionar un tipo de vidrio de la lista.
- Ingresar los datos área y Utilidad para este vidrio.
- Clic en Ingresar.

Modificar productos adicionales.

- En la pestaña Vidrios.
- En la tabla de vidrios hacer doble clic para seleccionar el vidrio a Modificar
- Modificar los datos área y Utilidad para este vidrios
- Clic en Modificar.

Eliminar un Vidrio.

- En la pestaña Vidrios.
- En la tabla de vidrios seleccionar el vidrio a Eliminar
- Clic Botón Eliminar Ítem

Las opciones:

Área de vidrios.- Corresponde a las áreas de los vidrios ingresados.

Área de estructuras.- Corresponde a las áreas de todas las estructuras cotizadas

Área faltante de vidrios.- Es el área que falta de vidrios para la cotización.

4.11 Historia de Usuario 11. Contratos

Tarea 11.1 Crear base de datos para Contratos

Luego de la creación de una proforma cuando llegan a un acuerdo con el cliente, los datos de esta son parte de un contrato, con algunos datos adicionales, como:

Id del técnico que realizará el trabajo, Id de la proforma, las fechas de inicio, fin y contrato, y un detalle, y finalmente una bandera.

CONTRATO		
CO_ID	int	<pk>
TEC_ID	int	<fk2>
PF_ID	int	<fk1>
CO_FECHA	datetime	
CO_INICIO	datetime	
CO_FIN	datetime	
CO_DETALLE	varchar(250)	
CO_ACTIVO	int	

Figura 29 Tabla de Contrato

Tarea 11.2 Crear formulario de Contratos.

Este formulario permite pasar una proforma previamente realizada, a un contrato, es decir que ya fue aprobada por el cliente y se va a realizar.

Es importante que permita seleccionar la proforma de la que se va a partir, la fecha en la que se contrata, y el periodo de tiempo que abarcará.

Los valores de mano de obra y valor de contrato pueden variar del cotizado, dependiendo del descuento manual que se haga.

Para los administradores se habilitará la opción de poner al contrato como activo o inactivo.

The screenshot shows a web form titled "Contrato". The form has the following fields and values:

- Contrato Id: 3
- Fecha Inicio: miércoles, 09 de marzo de 2016
- Fecha Fin: miércoles, 09 de marzo de 2016
- Descripción: conjunto
- Técnico: 1 | Heman S
- Valor Profoma: 216,55
- Valor Contratado: 216
- M. Obra Profoma: 48
- Mano de Obra Real: 48
- Contrato Activo

On the right side of the form, there are five buttons: "Nuevo", "Guardar", "Abrir", "Eliminar", and "Cerrar".

Figura 30 Formulario de Contratos

Creación de Funciones Necesarias:

Función para Ingresar contratos:

```
Function funIngresarContrato(varTeclId As Integer, varPfId As Integer,  
varFecha As Date, varInicio As Date, varFin As Date, varDetalle As String,  
varActivo As Boolean) As Integer
```

Función para Modificar los contratos:

```
Function funModificarContrato(VarCold As Integer, varTeclId As Integer,  
varPfId As Integer, varFecha As Date, varInicio As Date, varFin As Date,  
varDetalle As String, varActivo As Boolean) As Integer
```

Función para Desactivar un contrato:

```
Function funDesactivarContrato(VarCold As Integer) As Integer
```

Función para Eliminar contrato:

```
Function funEliminarContrato(varPfId As Integer) As Boolean
```

Función para Buscar contrato:

```
Function funBuscarContrato(strCliente As String) As Array
```

Opciones del formulario:

Ingresar Contratos.

- Buscar una proforma para hacer contrato.
- Ingresar los datos Correspondientes.
- Clic en Guardar.

Modificar Contratos.

- Buscar un Contrato por el nombre del cliente.
- Clic en aceptar
- Modificar los datos Correspondientes.

- Clic en Guardar.

Eliminar Contratos.

- Buscar un Contrato por el nombre del cliente.
- Clic en Eliminar.

El estado del contrato debe estar en activo, caso contrario no aparecerá en los reportes ni en las órdenes de materiales. Estos se pueden cambiar de estado en la Opción ventas/ Contratos.

4.12 Historia de Usuario 12. Lista de materiales

A partir de la proforma se obtiene una lista de todos los materiales, que se necesitan para la fabricación de las estructuras de dicha proforma.

Tarea 12.1 Crear interface que permita seleccionar la proforma contratada.

Permite la búsqueda de los contratos.

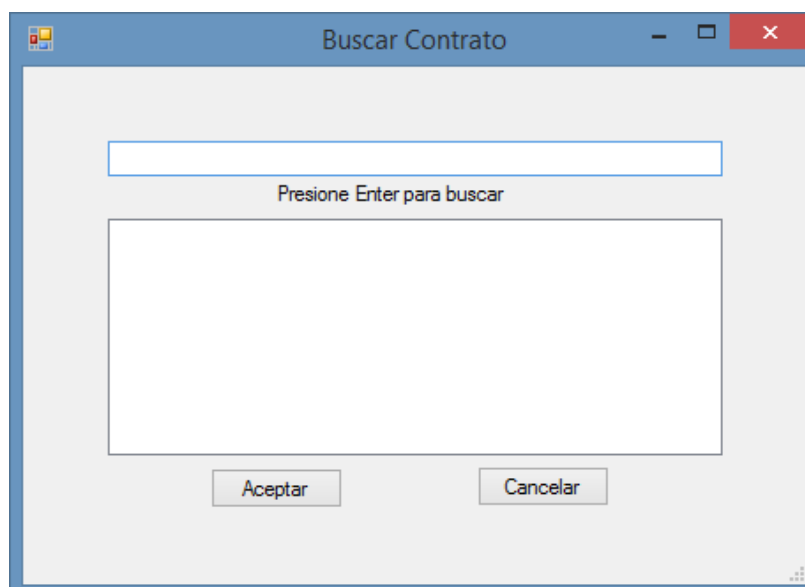


Figura 31 Formulario Buscar Contrato

Tarea 12.2 Crear base de datos para las listas.

Las listas de materiales que se generan para los contratos se guardarán las de las proformas se eliminarán, puesto que las proformas pueden ser modificadas.

LISTAMATERIALES		
<u>LS_ID</u>	int	<pk>
PF_ID	int	<fk2>
IN_ID	int	<fk1>
LS_CANTIDAD	float	
LS_PRECIU	float	
LS_DESCRIPCION	varchar(150)	
LS_BANDERA	bit	

Figura 32 Tabla lista de Materiales

Los datos de la tabla lista de materiales son los id de las proforma, insumos, las cantidades, precios y descripciones y una bandera para los contratos.

Tarea 12.3 Crear lista de materiales.

Este formulario permite obtener una lista de los materiales necesarios a partir de una proforma o un contrato.

CANTIDAD	DESCRIPCION	PRECIO	SUBTOTAL
13,29	F1108NA64 NATURAL Entrecieme P.C. (1108-196) FISA	1,00	13,29
6,65	F1065NA64 NATURAL Parante Chapa Anch P.C. (1065-14...	1,00	6,65
10,26	F1106NA64 NATURAL Peinazo Inferior P.C. (1106-142) Fl...	1,00	10,26
10,26	F1107NA64 NATURAL Peinazo Superior P.C. (1107-143) F...	1,00	10,26
13,29	F1109NA64 NATURAL Jamba P.C. (1109-147) FISA	1,00	13,29
10,20	F1111NA64 NATURAL Base P.C. (1111-149) FISA	1,00	10,20
10,65	F1116NA64 NATURAL Parante Fijo P.C. (1116-154) FISA	1,00	10,65
4,21	A101B BLANCO SILICON BLANCO	3,00	12,63
163,38	A150 GALVANIZADO Tornillo p/aglomerado 8 X1 1/2	0,01	1,63
67,38	A106 GRIS Tacos Fisher # 5	0,00	0,07
10,20	F1110NA64 NATURAL Cabezal P.C. (1110-148) FISA	2,00	20,40
22,20	F1765NA64 NATURAL Rectang Abierto Aleta 3"X1" (1765-...	5,00	111,00
27,00	F1583NA64 NATURAL Tubo Cuadrad 11/2"X11/2" Media....	8,35	225,45
1,00	A6214330 AZUL Vidrio Azul 6mm 214 X 330	11,50	11,50

Subtotal Insumos	457,27
Subtotal Utilidad	238,39
Subtotal Mano Obra	337,36
TOTAL	1033,02

Figura 33 Formulario lista de materiales

Creación de Funciones Necesarias:

Función para Eliminar listas de materiales de Proformas:

Function EliminarListaMaterialesSinContrato() **As Boolean**

Función Ingresar ítems para las listas:

Function IngresarItemListadeMateriales(varPfId **As Integer**, varInslId **As Integer**, varCantidad **As Double**, varPrecioU **As Double**, varDescripcion **As String**, varEsContrato **As Boolean**)

Función para generar lista de materiales:

Function GeneralistaDeMateriales(varPfId **As Integer**, varColor **As Integer**)
As Array

Opciones del formulario:

Lista de materiales de proforma.

- Buscar una proforma a partir del nombre del cliente
- Se obtendrá una lista de los materiales necesarios para la construcción de este proyecto.

Lista de materiales de contrato.

- Buscar un contrato a partir del nombre del cliente
- Se obtendrá una lista de los materiales necesarios para la construcción de este proyecto.

Lista de materiales a Excel.

- Luego de obtener la lista de materiales, se puede exportar a Excel para realizar análisis.

CANTIDAD	DESCRIPCION	VAL UNITAR	SUBTOTAL
6,645	F1065NA64 NATURAL Parante Chapa Anch P.C. (1065-145) F	\$ 1,00	\$ 6,65
10,26	F1106NA64 NATURAL Peinazo Inferior P.C. (1106-142) FISA	\$ 1,00	\$ 10,26
10,26	F1107NA64 NATURAL Peinazo Superior P.C. (1107-143) FISA	\$ 1,00	\$ 10,26
13,29	F1109NA64 NATURAL Jamba P.C. (1109-147) FISA	\$ 1,00	\$ 13,29
10,2	F1111NA64 NATURAL Base P.C. (1111-149) FISA	\$ 1,00	\$ 10,20
10,645	F1116NA64 NATURAL Parante Fijo P.C. (1116-154) FISA	\$ 1,00	\$ 10,65
4,21125	A101B BLANCO SILICON BLANCO	\$ 3,00	\$ 12,63
163,38	A150 GALVANIZADO Tornillo p/aglomerado 8 X1 1/2	\$ 0,01	\$ 1,63
67,38	A106 GRIS Tacos Fisher #5	\$ 0,00	\$ 0,07
10,2	F1110NA64 NATURAL Cabezal P.C. (1110-148) FISA	\$ 2,00	\$ 20,40
13,29	F1108NA64 NATURAL Entrecierre P.C. (1108-196) FISA	\$ 1,00	\$ 13,29
22,2	F1765NA64 NATURAL Rectang Abierto Aleta 3"X1" (1765-13	\$ 5,00	\$ 111,00
32,4	F1583NA64 NATURAL Tubo Cuadrad 11/2"X11/2" Media. (15	\$ 8,44	\$ 273,46
1	A6214330 AZUL Vidrio Azul 6mm 214 X 330	\$ 11,50	\$ 11,50
	SUBTOTAL		\$ 505,28
	UTILIDAD		\$ 252,79
	MANO DE OBRA		\$ 337,36
	TOTAL		\$ 1.095,43

Figura 34 Lista de materiales a Excel

Orden de Entrega a Excel.

- Luego de obtener la lista de materiales, se puede exportar a Excel la orden de entrega.

CANTIDAD	COD. ALGLASS	DESCRIPCIÓN
4	A150	GALVANIZADO Tornillo p/aglomerado 8 X1 1/2
4	F1065BL64	BLANCO Parante Chapa Anch P.C. (1065-145) FISA
1	F1106BL64	BLANCO Peinazo Inferior P.C. (1106-142) FISA
4	F1108BL64	BLANCO Entrecierre P.C. (1108-196) FISA
4	F1762BL64	BLANCO Tubo Rectang Chapero 3"X11/2" (1762-1369) FISA
Recibido Por David		
Emitido Por GUILLE		

Figura 35 Orden de entrega de materiales a Excel

4.13 Historia de Usuario 13. Entrega de materiales

Tarea 13.1 Crear base de datos para Técnicos

Los técnicos tienen su hoja de vida en otro sistema existente, por lo que aquí solo se guardan datos necesarios para identificarlos. Como son: su nombre, datos adicionales y un Número de Teléfono.

TECNICO		
<u>TEC_ID</u>	int	<pk>
TEC_NOMBRE	varchar(50)	
TEC_DATOS	varchar(150)	
TEC_TELEFONO	varchar(50)	

Figura 36 Tabla de técnicos

Tarea 13.2 Crear interface para Ingreso de técnicos.

Este formulario permite ingresar, modificar y eliminar datos de los técnicos.

Figura 37 Formulario para técnicos

Creación de Funciones Necesarias:

Función para Ingresar técnicos:

Function funIngresarTecnico(varTecNombre **As String**, varTelefono **As String**, varTecDatos **As String**) **As Integer**

Función para Modificar Técnicos:

Function funModificarTecnico(varTecId **As Integer**, varTecNombre **As String**, varTelefono **As String**, varTecDatos **As String**) **As Boolean**

Función para Eliminar técnicos:

Sub funEliminarTecnico(varTecId **As Integer**)

Opciones del formulario:

Ingresar técnico.

- Clic en Nuevo
- Ingresar los datos del técnico.
- Clic en Guardar.

Modificar Técnico.

- Seleccionar el técnico que desea modificar.
- Modificar los datos Correspondientes.
- Clic en Guardar.

Eliminar Técnico.

- Seleccionar un técnico.
- Clic en Eliminar.
Si tiene datos relacionados no se eliminará el técnico.

Tarea 13.3 Crear base de datos para Entregas de Materiales

Se guardará en esta tabla los materiales obtenidos a partir de la proforma y los adicionales que soliciten los técnicos para la construcción.

Con los datos del insumo, cantidades, precios, descripción y fecha.

LISTAMATERIALES		
<u>LS_ID</u>	int	<pk>
PF_ID	int	<fk2>
IN_ID	int	<fk1>
LS_CANTIDAD	float	
LS_PRECIOU	float	
LS_DESCRIPCION	varchar(150)	
LS_BANDERA	bit	

Figura 38 Tabla de lista de materiales

ADICIONALCONTRATO		
<u>AD_ID</u>	int	<pk>
CO_ID	int	<fk1>
IN_ID	int	<fk2>
AD_PRECIO	float	
AD_FECHA	date	
AD_DESCRIPCION	varchar(150)	
AD_CANTIDAD	float	

Figura 39 Tabla para adicionales de Contrato

Tarea 13.4 Crear interface para generar Órdenes de entrega.

Este formulario crea y registra los materiales adicionales que se les entrega a los técnicos.

Orden de Entrega para adicionales

lunes, 18 de enero de 2016

Contrato: 2 Jose Casa

Buttons: Nuevo, Guardar/Imprimir, Salir

ID	CANTIDAD	DESCRIPCION	PRECIO	TOTAL
12	4	12 Bronce Entrecierre P.C. (...)	1	4

Buttons: Insertar, Seleccionar Insumo, Eliminar Seleccionado

Figura 40 Formulario para órdenes de Entrega

Creación de Funciones Necesarias:

Función para Ingresar ordenes:

`Function funIngresarAddContrato(VarCoid As Integer, VarInId As Integer, varCantidad As Double, varPrecio As Double, varFecha As Date, varDetalle As String) As Integer`

Opciones del formulario:

Ingresar Orden de Entrega.

- Clic en Nuevo
- Seleccionar el contrato la que pertenecen los materiales.
- Seleccionar los insumos que desee
- Ingresar para cada uno una cantidad
- Clic en Ingresar
- Al finalizar Clic en Guardar

Los contratos que se listan en el combobox son solo los contratos activos, los inactivos ya no deberían ocupar materiales adicionales, por lo que ya no se listan.

Modificar ítems de la Orden de Entrega.

- Doble Clic en el insumo que se desee modificar.
- Modificar los datos deseados.
- Clic en modificar
- Al finalizar Clic en Guardar

Al momento de guardar, los datos se exportan a Excel. Para poder ser impresos.

4.14 Historia de Usuario 14. Generar planes de corte

Tarea 14.1 Crear base de datos para detalle de corte.

Cada ítem del plan de corte pertenece a una estructura de la proforma, los datos que se necesitan guardar son: id del insumo, id de la estructura, la cantidad y una bandera.

PLANCORTE		
<u>PC_ID</u>	<u>bigint</u>	<pk>
IN_ID	int	<fk2>
DP_ID	int	<fk1>
PC_CANTIDAD	float	
PC_CONTRATO	bit	

Figura 41 Tabla para plan de Corte

Tarea 14.1 Crear interface para seleccionar proforma para detalle de corte.

Este formulario busca una proforma entre las realizadas.

Figura 42 Formulario para buscar Proforma

Opciones del formulario:

Buscar Proforma

- Ingrese el nombre de la proforma a buscar.
- Escoja la proforma y Aceptar.

Tarea 14.2 Crear detalles de corte.

Plan de Corte

Código: 4 Cliente: Freddy Nuñez

Buscar Proforma: 4

Longitud Perfil: 6.35

Lista de aluminios para hacer Plan

Código	Aluminio
16	Parante Fijo P.C. (1116-154) ...
77	Parante Tubular V.C.L. (1388...
83	Base V.C.L. (1521-513) FISA
84	Cabezal V.C.L. (1523-511) Fl...
108	Jamba V.C.L. (1823-512-146...
121	Parante 7P (741-2231) FISA
122	Entrecierre Fijo 7P (744-2232)...
123	Peinazo 7P (2233-742) FISA

Detalle de Corte

Peinazo 7P (2233-742) FISA

Pz#1	60	V#2
Pz#1	120	V#2
Pz#1	120	V#2
Pz#1	145	V#4
Pz#1	145	V#4
Pz #1=	590 cm	Sobra 45 cm
Pz#2	75	V#3
Pz#2	75	V#3
Pz#2	75	V#3
Pz#2	75	V#5
Pz#2	75	V#5
Pz#2	75	V#4
Pz#2	75	V#4
Pz #2=	600 cm	Sobra 35 cm
Pz#3	75	V#7
Pz#3	75	V#7
Pz#3	75	V#6
Pz#3	75	V#6
Pz#3	70	V#6
Pz#3	70	V#6
Pz#3	70	V#7
Pz#3	70	V#7
Pz #3=	580 cm	Sobra 55 cm
Pz#4	70	V#3
Pz#4	70	V#3
Pz#4	70	V#3
Pz#4	70	V#4
Pz#4	70	V#5
Pz#4	70	V#5
Pz#4	60	V#2
Pz#4	70	V#3
Pz #4=	620 cm	Sobra 15 cm

Lista de Retazos para calcular

DC_CODIGO	PC_CANTIDAD
2	0.6
2	0.6
3	0.7
3	0.7
3	0.7
3	0.7
4	0.7

Longitud Perfil: 6.35

Limpiar Exportar para Imprimir

Pasar a lista para imprimir. ==>

Total Perfiles 4

Cerrar

Figura 43 Formulario de plan de corte

Creación de Funciones Necesarias:

Función para Ingresar datos Plan de Corte:

Function funIngresarItemsPlan(insumold As Integer, DetallePFId As Integer, Cantidad As Double, esContrato As Boolean) As Integer

Función lista los aluminios para hacer el plan:

Function funlistarAluminiosparaPlan(varpfld As Integer, varinsumo As Integer) As Array

Función eliminar planes que no sean Cotrato:

Function funEliminarPlan() As Boolean

Opciones del formulario:

Buscar Proforma

- Ingrese el nombre de la proforma a buscar.
- Escoja la proforma y Aceptar.

Detalle de Corte

- Seleccione un ítem en la lista de aluminios
- En detalle de corte aparece el plan de Corte
- Clic en ==> Para hacer una lista
- Exportar para Imprimir, pasa la lista a Excel

Cambiar tamaño de la Pieza

- Normalmente aparece 6.35
- Puede cambiar la longitud.
- Clic nuevamente en el ítem para recalcular.

4.15 Historia de Usuario 15. Reportes de costos y finalizaciones.

Tarea 15.1 Crear reportes de técnicos.

Este formulario permite seleccionar un técnico e ingresar fechas en un lapso para filtrar los trabajos realizados en cierto tiempo.

The screenshot shows a web application window titled "Reporte Técnicos". It contains the following elements:

- A dropdown menu for "Técnico" with the value "1 Herman S".
- Two date pickers: "Desde" (viernes, 01 de enero de 2016) and "Hasta" (domingo, 31 de enero de 2016).
- A button labeled "Reporte".
- A table with the following data:

	Contrato	Area	Valor_Mano_de_ob
▶	1	11,9	74,9
- A field labeled "Valor a pagar" with the value "74,9".

Figura 44 Formulario de Reporte de Técnicos

Opciones del formulario:

Reporte Técnico

- Seleccione el técnico del cual desea el reporte.
- Seleccione el rango de fechas para el reporte.
- Clic en Reporte.
- El valor a pagar es la suma de todos los contratos realizados por este técnico.

Tarea 15.2 Crear reportes de finalización de proyectos.

Este es un reporte de un proyecto sumado todo lo que se utilizó para su construcción.

The screenshot shows a window titled "Reporte Final" with the following elements:

- Contrato:** A dropdown menu showing "2 Jose Casa" and a "Contrato Terminado" button.
- Lista de Materiales:** A table with columns: CANTIDAD, DESCRIPCION, PRECIO, SUBTOTAL.

CANTIDAD	DESCRIPCION	PRECIO	SUBTOTAL
10.5	F1765NA64 NAT...	5	52.5
11.9	F1583NA64 NAT...	8.35	99.365
- Lista Adicionales:** A table with columns: Cantidad, Descripción, Precio, Subtotal.

Cantidad	Descripción	Precio	Subtotal
4	14 Bronce Cabez...	8	32
- Summary:**
 - Insumos: 183.86
 - Mano de Obra: 134.42
 - Total Contrato: 458
 - Total Gastado: 318,28
- A "Salir" button is located at the bottom right.

Figura 45 Formulario de reporte final

Tarea 15.3 Crear reportes de Proformas.

Reporte Proforma

Desde: viernes, 01 de enero de 2016
Hasta: sábado, 23 de enero de 2016

Reporte

	Proforma	Ciente	Fecha	Valor
▶	1	Freddy Nuñez	20/01/2016	395,47
	2	Juan Perez	20/01/2016	397,36
	3	Ana Iza	22/01/2016	90,73

Total Proformado: 883,56

Salir

Figura 46 Formulario Reporte de proformas

Opciones del formulario:

Reporte Proformas

- Seleccione el rango de fechas para el reporte.
- Clic en Reporte.

Si desea copiar los datos seleccione la tabla y presión ctrl + c

Tarea 15.4 Crear reportes de Contratos.

Este permite obtener un reporte de los contratos realizados.

Reporte Contratos

Desde: viernes, 15 de enero de 2016
Hasta: sábado, 23 de enero de 2016

Reporte

	Contrato	Ciente	Fecha	Valor
▶	1	Juan Perez	20/01/2016 2:09	397,36

Total Contratado: 397,36

Salir

Figura 47 Formulario reporte de contrato

Opciones del formulario:

Reporte Contratos

- Seleccione el rango de fechas para el reporte.
- Clic en Reporte.

Si desea copiar los datos seleccione la tabla y presión ctrl + c

CAPÍTULO 5

5 PRUEBAS Y RESULTADOS.

5.1 Pruebas de Aceptación

Cuadro 67

Prueba de aceptación Registro correcto de usuario

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Manejo de Usuarios
Número de Caso de prueba: 1	Número de historia de Usuario: 1
Nombre del caso de prueba:	Registro correcto de usuario
Descripción:	
Se realiza el ingreso correcto de los datos para el usuario, nombre, contraseña, roles.	
Condiciones de la ejecución:	
Los usuarios se registran para el acceso al sistema	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Manejo de Usuario 3. Ingresa los datos para un nuevo usuario 4. Selecciona los roles correspondientes 5. Da clic en guardar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarde la información 	
Evaluación:	
Los datos del usuario se ingresan correctamente.	

Cuadro 68**Prueba de aceptación registro incorrecto de usuario**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Manejo de Usuarios
Número de Caso de prueba: 2	Número de historia de Usuario: 1
Nombre del caso de prueba:	Registro incorrecto de usuario
Descripción:	
Se realiza el ingreso incompleto de los datos para el usuario, nombre, contraseña, roles.	
Condiciones de la ejecución:	
Los usuarios se registran para el acceso al sistema	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Manejo de Usuario 3. No ingresa los datos para un nuevo usuario 4. Selecciona los roles correspondientes 5. Da clic en guardar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que emita un mensaje de error 	
Evaluación:	
<p>Sale un mensaje de datos incompletos</p> <p>Los datos no se guardan.</p>	

Cuadro 69
Prueba de aceptación eliminar usuario

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Manejo de Usuarios
Número de Caso de prueba: 3	Número de historia de Usuario: 1
Nombre del caso de prueba:	Eliminación de Usuario con proforma
Descripción:	
Elimina un usuario del sistema.	
Condiciones de la ejecución:	
El usuario registra una proforma.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Manejo de Usuario 3. Se selecciona un usuario existente. 4. Da clic en Eliminar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que emita un mensaje de error 	
Evaluación:	
<p>Sale un mensaje "No se puede eliminar este usuario"</p> <p>Los datos no se eliminan.</p>	

Cuadro 70
Prueba de aceptación ingreso de nuevo color

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Colores para cotizar
Número de Caso de prueba: 1	Número de historia de Usuario: 2
Nombre del caso de prueba:	Ingreso de un nuevo color para cotizar.
Descripción:	
Los colores definen las opciones para cotizar	
Condiciones de la ejecución:	
Ingreso normal de datos	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Colores para cotizar 3. Ingresa un nuevo color 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan satisfactoriamente.	

Cuadro 71**Prueba de aceptación Ingreso erróneo de color**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Colores para cotizar
Número de Caso de prueba: 2	Número de historia de Usuario: 2
Nombre del caso de prueba:	Ingreso erróneo de un nuevo color para cotizar.
Descripción:	
Los colores definen las opciones para cotizar	
Condiciones de la ejecución:	
No se ingresa ningún color	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Colores para cotizar 3. No ingresa datos en color 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que no se guarden los datos 	
Evaluación:	
<p>Los datos no se guardan.</p> <p>Sale un mensaje de Ingrese un color.</p>	

Cuadro 72
Prueba de aceptación Modificar color

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Colores para cotizar
Número de Caso de prueba: 3	Número de historia de Usuario: 2
Nombre del caso de prueba:	Modificar un nuevo color para cotizar.
Descripción:	
Los colores definen las opciones para cotizar	
Condiciones de la ejecución:	
Se selección un color existente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Colores para cotizar 3. Selección un color existente. 4. Modifica el color. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan correctamente.	

Cuadro 73**Prueba de aceptación Ingresar tipo de producto**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de productos
Número de Caso de prueba: 1	Número de historia de Usuario: 3
Nombre del caso de prueba:	Ingreso de un nuevo tipo de producto
Descripción:	
Los productos se agrupan por tipos.	
Condiciones de la ejecución:	
Se ingresan los datos correctamente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Tipos de Productos 3. Ingresa un nuevo Tipo 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan satisfactoriamente	

Cuadro 74
Ingreso incorrecto de tipo de producto

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de productos
Número de Caso de prueba: 2	Número de historia de Usuario: 3
Nombre del caso de prueba:	Ingreso incorrecto de un nuevo tipo de producto
Descripción:	
Los productos se agrupan por tipos.	
Condiciones de la ejecución:	
Se ingresan los datos incorrectamente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Tipos de Productos 3. No ingresa datos 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que no se guarden los datos 	
Evaluación:	
<p>Los datos no se guardan</p> <p>Pide una descripción.</p>	

Cuadro 75**Prueba de aceptación modificar tipo de producto**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de productos
Número de Caso de prueba: 3	Número de historia de Usuario: 3
Nombre del caso de prueba:	Modificar un tipo de producto
Descripción:	
Los productos se agrupan por tipos.	
Condiciones de la ejecución:	
Se selecciona los datos correctamente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Tipos de Productos 3. Selección un tipo de producto existente. 4. Modifica la descripción. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos no guardan correctamente.	

Cuadro 76
Prueba de aceptación ingreso de productos

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de productos
Número de Caso de prueba: 1	Número de historia de Usuario: 4
Nombre del caso de prueba:	Ingreso correcto de productos
Descripción:	
Los insumos son los nombres generales, los productos son lo que tienen ya definido un color.	
Condiciones de la ejecución:	
Los productos son ingresados con su descripción completa.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Ingreso de Insumos 3. Ingresa una descripción 4. Selecciona un tipo y una unidad. 5. Agrega un color 6. Se pega una imagen. 7. Y los habilita para las cotizaciones en diferentes colores. 8. Clic en guardar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se ingresan satisfactoriamente.	

Cuadro 77
Prueba de aceptación ingreso erróneo de productos

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de productos
Número de Caso de prueba: 2	Número de historia de Usuario: 4
Nombre del caso de prueba:	Ingreso incompleto de datos del producto
Descripción:	
Los insumos son los nombres generales, los productos son lo que tienen ya definido un color.	
Condiciones de la ejecución:	
Los productos son ingresados con su descripción incompleta.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Ingreso de Insumos 3. No ingresa una descripción 4. Selecciona un tipo y una unidad. 5. Agrega un color 6. Y los habilita para las cotizaciones en diferentes colores. 7. Clic en guardar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que no se guarden los datos 	
Evaluación:	
<p>Los datos no se ingresan pero sale un mensaje repetitivo.</p> <p>Se realiza la corrección del error.</p>	

Cuadro 78
Prueba de aceptación modificar producto

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de productos
Número de Caso de prueba: 3	Número de historia de Usuario: 4
Nombre del caso de prueba:	Modificar producto
Descripción:	
Los insumos son los nombres generales, los productos son lo que tienen ya definido un color.	
Condiciones de la ejecución:	
Los productos son previamente ingresados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Ingreso de Insumos 3. Busca un Insumo existente 4. Modifica los precios de los productos. 5. Clic en guardar. 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se ingresan correctamente.	

Cuadro 79
Prueba de aceptación ingreso de cliente

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso De Clientes
Número de Caso de prueba: 1	Número de historia de Usuario: 5
Nombre del caso de prueba:	Ingreso correcto de cliente
Descripción:	
Los clientes tienen varios datos, cédula, nombre, teléfono, mail, etc.	
Condiciones de la ejecución:	
Se ingresan los datos correctamente.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción ventas/ manejo de clientes. 3. Ingresa los datos correspondientes. 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que se guarden los datos. 	
Evaluación:	
Los datos se guardan satisfactoriamente.	

Cuadro 80
Prueba de aceptación ingreso de clientes

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso De Clientes
Número de Caso de prueba: 2	Número de historia de Usuario: 5
Nombre del caso de prueba:	Ingreso incorrecto de cliente
Descripción:	
Los clientes tienen varios datos, cédula, nombre, teléfono, mail, etc.	
Condiciones de la ejecución:	
No se ingresa nombre al cliente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción ventas/ manejo de clientes. 3. No ingresa nombre al cliente. 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que se no guarden los datos. 	
Evaluación:	
<p>Los datos no se guardan.</p> <p>Sale un mensaje de error.</p>	

Cuadro 81
Prueba de aceptación modificar clientes

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso De Clientes
Número de Caso de prueba: 3	Número de historia de Usuario: 5
Nombre del caso de prueba:	Modificar datos del cliente
Descripción:	
Los clientes tienen varios datos, cédula, nombre, teléfono, mail, etc.	
Condiciones de la ejecución:	
Se tiene clientes ingresados previamente.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción ventas/ manejo de clientes. 3. Busca un cliente existente. 4. Modifica los datos. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que se guarden los datos. 	
Evaluación:	
Los datos se guardan correctamente.	

Cuadro 82
Prueba de aceptación ingreso valor mano de obra

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Mano de Obra
Número de Caso de prueba: 1	Número de historia de Usuario: 6
Nombre del caso de prueba:	Ingresar un valor de mano de obra
Descripción:	
Dependiendo de las estructuras tienen un costo de mano de obra.	
Condiciones de la ejecución:	
Se ingresa los datos correctamente.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/Costo mano de obra. 3. Ingresa una descripción de mano de obra. 4. Ingresa un costo de mano de obra. 5. Modifica los datos. 6. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que los datos se ingresen correctamente. 	
Evaluación:	
Los datos se guardaron en forma satisfactoria.	

Cuadro 83
ingreso negativo o 0 de mano de obra

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Mano de Obra
Número de Caso de prueba: 2	Número de historia de Usuario: 6
Nombre del caso de prueba:	Ingresar un valor de mano de obra negativo o 0
Descripción:	
Dependiendo de las estructuras tienen un costo de mano de obra.	
Condiciones de la ejecución:	
Se ingresa los datos erróneos.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/Costo mano de obra. 3. Selecciona un costo de mano de obra. 4. El valor con valor negativo. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que los datos no se guarden. 	
Evaluación:	
<p>Los datos no se guardaron.</p> <p>Emite un mensaje de error que los datos están incorrectos.</p>	

Cuadro 84**Prueba de aceptación modificar valor mano de obra**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Mano de Obra
Número de Caso de prueba: 3	Número de historia de Usuario: 6
Nombre del caso de prueba:	Modificar un valor de mano de obra
Descripción:	
Dependiendo de las estructuras tienen un costo de mano de obra.	
Condiciones de la ejecución:	
Se ingresa los datos correctamente.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/Costo mano de obra. 3. Selecciona un costo de mano de obra. 4. Modifica los datos. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Que los datos se ingresen correctamente. 	
Evaluación:	
Los datos se guardaron en forma satisfactoria.	

Cuadro 85**Prueba de aceptación ingresar nuevo grupo de estructura**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de Grupo de estructuras
Número de Caso de prueba: 1	Número de historia de Usuario: 7
Nombre del caso de prueba:	Ingreso de un nuevo Grupo de Estructura
Descripción:	
Las estructuras se clasifican por grupos en general.	
Condiciones de la ejecución:	
Se ingresan los datos correctamente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Grupo de estructuras. 3. Ingresa un nuevo Grupo 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan satisfactoriamente	

Cuadro 86**Prueba de aceptación ingresar grupo erróneo de estructura**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de Grupo de estructuras
Número de Caso de prueba: 2	Número de historia de Usuario: 7
Nombre del caso de prueba:	Ingreso de un Grupo de Estructura erróneo
Descripción:	
Las estructuras se clasifican por grupos en general.	
Condiciones de la ejecución:	
Se ingresan los datos incompletos	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Grupo de estructuras. 3. No ingresa datos 4. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que no se guarden los datos 	
Evaluación:	
<p>Los datos no se guardan.</p> <p>Sale un mensaje que falta ingresar una descripción.</p>	

Cuadro 87**Prueba de aceptación modificar grupo de estructura**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de Grupo de estructuras
Número de Caso de prueba: 3	Número de historia de Usuario: 7
Nombre del caso de prueba:	Modificar un Grupo de Estructura
Descripción:	
Las estructuras se clasifican por grupos en general.	
Condiciones de la ejecución:	
Existe un grupo ya creado de grupo de estructuras.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Grupo de estructuras. 3. Selecciona un grupo existente 4. Se modifica la descripción 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan perfectamente.	

Cuadro 88
Prueba de aceptación ingresar tipo de estructura

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de estructuras
Número de Caso de prueba: 3	Número de historia de Usuario: 7
Nombre del caso de prueba:	Ingreso de un nuevo tipo de Estructura
Descripción:	
Las estructuras se clasifican por tipos.	
Condiciones de la ejecución:	
Se ingresan los datos correctamente	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/Ingresar tipos de estructuras. 3. Selecciona el grupo. 4. Ingresa un nuevo Tipo 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan satisfactoriamente	

Cuadro 89**Prueba de aceptación ingresar tipo de estructura erróneo**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de estructuras
Número de Caso de prueba: 4	Número de historia de Usuario: 7
Nombre del caso de prueba:	Ingreso de un nuevo tipo de Estructura erróneo.
Descripción:	
Las estructuras se clasifican por tipos.	
Condiciones de la ejecución:	
Se ingresan los datos incompletos	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/Ingresar tipos de estructuras. 3. Selecciona el grupo. 4. No se ingresa una descripción. 5. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que no se guarden los datos 	
Evaluación:	
Los datos no se ingresan	
Da un mensaje de datos incompletos.	

Cuadro 90
Prueba de aceptación modificar tipo de estructura

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Ingreso de tipos de estructuras
Número de Caso de prueba: 4	Número de historia de Usuario: 7
Nombre del caso de prueba:	Modificar un tipo de Estructura
Descripción:	
Las estructuras se clasifican por tipos.	
Condiciones de la ejecución:	
Hay tipos de estructuras ya creados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Ingresar tipo de estructuras. 3. Selecciona un grupo 4. Selecciona un tipo existente 5. Se modifica la descripción 6. Clic en guardar 	
Resultado esperado:	
<ul style="list-style-type: none"> • Se espera que se guarden los datos 	
Evaluación:	
Los datos se guardan correctamente.	

Cuadro 91
Prueba de aceptación ingresar ítems de estructura

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Detalle de estructuras
Número de Caso de prueba: 1	Número de historia de Usuario: 8
Nombre del caso de prueba:	Ingresar ítems a una estructura.
Descripción:	
Cada estructura tiene varios ítems que sumados constituyen el costo de la misma.	
Condiciones de la ejecución:	
Se ingresan y seleccionan los datos correctos. Existen grupos y tipos creados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Estructuras. 3. Selecciona un grupo existente 4. Selecciona un tipo existente 5. Seleccionamos o creamos los materiales 6. Seleccionamos un valor de mano de obra 7. Ingresamos un valor para utilidad. 8. Ingresamos un producto. 9. Ingresamos una fórmula 10. Clic en modificar para actualizar en la tabla. 11. Clic en guardar 	
Resultado esperado:	
Se crea o modifica una estructura con sus ítems y cantidades.	
Evaluación:	
Los datos son ingresados en forma correcta	

Cuadro 92
Prueba de aceptación ingresar ítems erróneos

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Detalle de estructuras
Número de Caso de prueba: 2	Número de historia de Usuario: 8
Nombre del caso de prueba:	Ingresar ítems a una estructura erróneos.
Descripción:	
Cada estructura tiene varios ítems que sumados constituyen el costo de la misma.	
Condiciones de la ejecución:	
Se ingresa fórmula de la cantidad con error, y seleccionan los datos correctos. Existen grupos y tipos creados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Estructuras. 3. Selecciona un grupo existente 4. Selecciona un tipo existente 5. Seleccionamos o creamos los materiales 6. Seleccionamos un valor de mano de obra 7. Ingresamos un valor para utilidad. 8. Ingresamos un producto. 9. Ingresamos una fórmula con error 10. Clic en modificar 	
Resultado esperado:	
Mensaje de error en la fórmula.	
Evaluación:	
El campo aparece en rojo mientras la fórmula es errónea. Al tratar de guardar emite un mensaje de error.	

Cuadro 93**Prueba de aceptación ingreso de descuentos para retazos**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Descuentos por piezas
Número de Caso de prueba: 1	Número de historia de Usuario: 9
Nombre del caso de prueba:	Ingreso de descuento para piezas
Descripción:	
Los aluminios para ser cortados casi todos deben tener un descuento en su medida.	
Condiciones de la ejecución:	
Existen datos ingresados, e ítems seleccionados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Estructuras. 3. Selecciona un grupo existente 4. Selecciona un tipo existente 5. Seleccionamos un material 6. Ingresamos un valor para descuento 7. Clic en modificar 	
Resultado esperado:	
El descuento ingresado sea guardado.	
Evaluación:	
El descuento se ingresa correctamente para un ítem	

Cuadro 94
Prueba de aceptación ingreso erróneo de descuentos

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Descuentos por piezas
Número de Caso de prueba: 2	Número de historia de Usuario: 9
Nombre del caso de prueba:	Ingreso de descuento para piezas erróneo
Descripción:	
Los aluminios para ser cortados casi todos deben tener un descuento en su medida.	
Condiciones de la ejecución:	
Existen datos ingresados, e ítems seleccionados.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Estructuras/ Estructuras. 3. Selecciona un grupo existente 4. Selecciona un tipo existente 5. Seleccionamos un material que no es aluminio 6. Ingresamos un valor para descuento 7. Clic en modificar 	
Resultado esperado:	
Solo los aluminios pueden tener descuento.	
Evaluación:	
El valor ingresado se guarda.	
CORRECCIÓN.	
Se modifica el formulario para que solo cuando se seleccione aluminios se habilite el descuento.	

Cuadro 95
Prueba de aceptación crear una proforma

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Proforma de estructuras
Número de Caso de prueba: 1	Número de historia de Usuario: 10
Nombre del caso de prueba:	Crear una nueva proforma
Descripción:	
Una proforma es una cotización de una lista de estructuras con sus medidas.	
Condiciones de la ejecución:	
Se han creado clientes, y existen estructuras creadas.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Ventas/ Proforma. 3. Selecciona un cliente 4. Ingrese una descripción y una dirección. 5. Seleccione un grupo, tipo y material de una estructura 6. Ingrese las medidas requeridas 7. Clic en ingresar. 8. Ingresamos otros materiales 9. Cambiamos el color de la cotización. 	
Resultado esperado:	
Datos ingresados, una proforma.	
Evaluación:	
Los datos se ingresan correctamente, los valores son los correctos.	

Cuadro 96
Prueba de aceptación crear proforma con errores

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Proforma de estructuras
Número de Caso de prueba: 2	Número de historia de Usuario: 10
Nombre del caso de prueba:	Crear una nueva proforma con errores
Descripción:	
Una proforma es una cotización de una lista de estructuras con sus medidas.	
Condiciones de la ejecución:	
Se han creado clientes, y existen estructuras creadas.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Ventas/ Proforma. 3. Selecciona un cliente 4. Ingrese una descripción y una dirección. 5. Seleccione un grupo, tipo y material de una estructura 6. No ingrese las medidas requeridas 7. Clic en ingresar. 	
Resultado esperado:	
Datos ingresados y una proforma.	
Evaluación:	
Los datos no se ingresan si no están completas las medidas.	

Cuadro 97
Prueba de aceptación obtener lista de materiales

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Lista de materiales
Número de Caso de prueba: 1	Número de historia de Usuario: 11
Nombre del caso de prueba:	Obtener lista de materiales
Descripción:	
El sistema debe proporcionar listas de materiales a partir de una cotización.	
Condiciones de la ejecución:	
Se ha creado una proforma previamente.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Lista de Materiales. 3. Selecciona una proforma o contrato 4. Clic en exportar a Excel para imprimir. 5. Si es contrato se puede sacar una orden de entrega de materiales. 	
Resultado esperado:	
Lista de materiales y datos adicionales de los valores de la proforma	
Evaluación:	
Se obtiene una lista de materiales. El valor final es igual al valor de la proforma.	

Cuadro 98
Prueba de aceptación crear un contrato

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Contratos
Número de Caso de prueba: 1	Número de historia de Usuario: 12
Nombre del caso de prueba:	Crear un contrato
Descripción:	
<p>Cuando una proforma ha sido aceptada por el cliente, esta pasa a ser un contrato, en el que se relaciona a un técnico, y a la proforma</p>	
Condiciones de la ejecución:	
<p>Existe una proforma previamente creada.</p>	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Crear Contrato. 3. Selecciona una proforma 4. Establezca los valores para la fecha de inicio del proyecto, fin del proyecto. 5. Ingrese un detalle adicional 6. Selección el nombre del técnico que construirá el proyecto. 7. Clic en guardar 	
Resultado esperado:	
<p>Un nuevo contrato creado</p>	
Evaluación:	
<p>Los datos se guardan correctamente.</p>	

Cuadro 99
Prueba de aceptación modificar un contrato

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Contratos
Número de Caso de prueba: 2	Número de historia de Usuario: 12
Nombre del caso de prueba:	Modificar un contrato
Descripción:	
Cuando una proforma ha sido aceptada por el cliente, esta pasa a ser un contrato, en el que se relaciona a un técnico, y a la proforma	
Condiciones de la ejecución:	
Existe un contrato previamente creado.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Crear Contrato. 3. Busca un contrato. 4. Realiza las modificaciones necesarias 5. Clic en guardar 	
Resultado esperado:	
Un contrato modificado	
Evaluación:	
Los datos se guardan correctamente.	

Cuadro 100**Prueba de aceptación obtener entrega de materiales**

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Entrega de materiales
Número de Caso de prueba: 1	Número de historia de Usuario: 13
Nombre del caso de prueba:	Orden de entrega de materiales
Descripción:	
Luego de realizado un contrato, al momento de la construcción puede faltar materiales adicionales y estos deben ser registrados en el sistema.	
Condiciones de la ejecución:	
Existe un contrato creado	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Orden de entrega adicionales. 3. Selecciona un contrato (Solo se listan los contrato activos.) 4. Seleccione el o los producto que desea entregar 5. Ingrese una cantidad. 6. Clic en insertar. 7. Cuando termine clic en Guardar/imprimir. 	
Resultado esperado:	
Una orden de estrega.	
Evaluación:	
Se obtiene una lista de materiales para descargar.	
Los datos se registran correctamente en el sistema.	

Cuadro 101
Prueba de aceptación plan de corte

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Generar planes de corte
Número de Caso de prueba: 1	Número de historia de Usuario: 14
Nombre del caso de prueba:	Obtener un plan de corte a partir de una proforma
Descripción:	
Un plan de corte es un detalle de como optimizar los cortes que se deben hacer en las tiras de aluminio.	
Condiciones de la ejecución:	
Debe haber previamente una proforma cargada. Las estructuras deben estar cargadas con sus descuentos.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Planes de Corte. 3. Selecciona una proforma. 4. Seleccione el aluminio que desea calcular 5. Pase los datos a la lista para imprimir 6. Clic en Exportar para Imprimir. 	
Resultado esperado:	
Detalle de corte	
Evaluación:	
Se obtiene el resultado deseado	

Cuadro 102
Prueba de aceptación Reporte fin de proyecto

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Reportes de costos y finalizaciones
Número de Caso de prueba: 1	Número de historia de Usuario: 15
Nombre del caso de prueba:	Reporte fin de proyecto.
Descripción:	
Luego de terminado un proyecto es importante tener un reporte de los materiales utilizados para la construcción total.	
Condiciones de la ejecución:	
Para esto se debe haber hecho un contrato, y haber generado órdenes de entrega.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Reporte de finalización de proyectos. 3. Seleccionas un contrato. 4. Clic en contrato terminado para desactivar el contrato. 	
Resultado esperado:	
Reporte final	
Evaluación:	
Se obtiene un reporte con todos los datos cargados.	

Cuadro 103
Prueba de aceptación reporte de técnicos

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Reportes de técnicos.
Número de Caso de prueba: 2	Número de historia de Usuario: 15
Nombre del caso de prueba:	Reporte Técnicos.
Descripción:	
A los técnicos se les paga por proyecto terminado es importante un reporte que tenga la mano de obra realizada en un periodo de tiempo.	
Condiciones de la ejecución:	
Para esto se debe haber hecho un contrato.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Reporte por técnicos. 3. Seleccionas un técnico. 4. Selecciona un periodo de tiempo 5. Clic en Reporte. 	
Resultado esperado:	
Reporte de lo que se le debe pagar a un técnico.	
Evaluación:	
Se obtiene un reporte con todos contratos realizados por un técnico.	

Cuadro 104
Prueba de aceptación reporte de proformas

PRUEBA DE ACEPTACIÓN	
Caso de prueba:	Reportes de proformas y contratos.
Número de Caso de prueba: 3	Número de historia de Usuario: 15
Nombre del caso de prueba:	Reporte Proformas y contratos.
Descripción:	
Es importante tener un reporte de lo proformado y contratado durante un mes, con el fin de calcular la aceptación de clientes.	
Condiciones de la ejecución:	
Para esto se debe haber hecho proformas y contratos.	
Entradas:	
<ol style="list-style-type: none"> 1. El usuario ingresa con nombre y contraseña 2. Ingresa a la Opción Producción/ Reporte Proformas (Contratos). 3. Selecciona un periodo de tiempo 4. Clic en reporte. 	
Resultado esperado:	
Reporte de contratos y proformas.	
Evaluación:	
Se obtiene un reporte con todas las proformas y contratos en un lapso de tiempo.	

5.2 RESULTADOS.

Uno de los resultados que se quería obtener, es una comparación del tiempo empleado para realizar cotizaciones de forma manual y con el sistema:

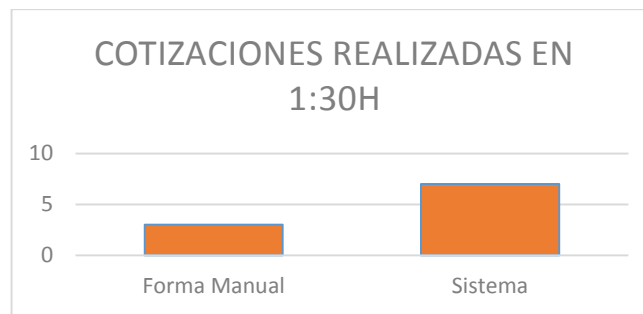


Figura 48 Resultado en cotizaciones

Como indica el gráfico, se ha disminuido en más del 50% la cantidad de tiempo empleado para realizar una cotización.

Otra de las tareas necesaria es la obtención de una lista de los materiales para realizar un proyecto.



Figura 49 Obtener lista de materiales

En este gráfico se compara el tiempo que se demora el sistema con el que se demora el técnico en obtener la lista de materiales, y el resultado es una optimización de más del 75%. Así también se observa que en la lista no hay errores mientras que en la lista del técnico se encontraron 2.

Los planes de corte son una tarea constante y delicada, en este punto se obtuvo el siguiente gráfico.



Figura 50 Plan de corte

En este proceso comparamos el tiempo utilizado por el técnico con el utilizado por el sistema, la optimización es mayor pues se disminuye en casi un 90%, así también se eliminan los errores por descuentos mal hechos.

También se puede obtener reportes de in de obras que permiten ver rápidamente cuanto se gastó en una obra, incluyendo los materiales adicionales entregados a los técnicos. No se hace una comparación por no tener antecedentes.

CAPÍTULO 6

6 CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- La metodología XP ha sido aplicada en el desarrollo de este sistema, y ha permitido con la implementación sus fases, satisfacer los objetivos planteados.
- Se analizó y validó los requisitos solicitados por la empresa, dejando como resultado solamente los funcionales. Lo que permitió obtener un alcance específico de lo que se ha desarrollado.
- Se ha estructurado y diseñado el sistema que permite realizar las tareas de cotización y producción de una forma ágil y precisa, así también se implementó la base de datos que registra y graba toda la información necesaria.
- Se ha aplicado un algoritmo que genera planes de corte para el aluminio, los mismos que son exactos, rápidos y ayudan a optimizar la materia prima.
- El alcance del proyecto propuesto por el cliente ha sido cumplido, mediante la creación de los módulos solicitados, se ha optimizado los procesos de cotización y producción; la implementación de estos satisfacen las necesidades de Alglass.
- Como todo proceso de desarrollo está expuesto a errores se han realizado las pruebas necesarias que han permitido, encontrar y corregir errores, a fin de obtener un sistema seguro y fiable.

6.2 Recomendaciones

- Para el desarrollo de este tipo de aplicaciones es recomendable la utilización de la metodología XP que permite el desarrollo ágil de aplicaciones, debido a que se necesita tener una programación organizada y una constante comunicación con el cliente.
- La herramienta Visual Studio Community, permitió cumplir con los requerimientos de Alglass, sin embargo en caso de necesitar módulos adicionales puede ser necesario la migración a una versión más completa como Visual Studio Profesional.
- Se recomienda la indexación de este sistema con el sistema DW de la empresa, con el fin de lograr la actualización automática de precios y la interacción con el inventario.
- Para requerimientos adicionales o modificaciones en los módulos desarrollados, es importante que se lea el presente documento y los comentarios en el código, para facilitar el proceso.

BIBLIOGRAFÍA

- Alglass. (01 de 01 de 2013). *Quienes somos: Alglass Ecuador*. Obtenido de <http://alglassecuador.com/>
- EcuRed. (May de 2010). *Programación Extrema*. Obtenido de [http://www.ecured.cu/Programaci%C3%B3n_Extrema_\(XP\)](http://www.ecured.cu/Programaci%C3%B3n_Extrema_(XP))
- EcuRed. (Nov de 2011). *Kent Beck*. Obtenido de http://www.ecured.cu/Kent_Beck#Bibliograf.C3.ADa
- EcuRed. (2016). *Visual Basic*. Obtenido de http://www.ecured.cu/Visual_Basic
- Escofet, C. M. (Feb de 2007). *Base de Datos*. Obtenido de http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02149.pdf
- Malemeza. (sep de 2012). *SQL Server*. Obtenido de <http://malemeza.blogspot.com/2012/09/sql-server.html>
- Microsoft. (Febrero de 2007). *LINQ .Net*. Obtenido de <https://msdn.microsoft.com/en-us/library/bb308959.aspx>
- Microsoft. (2014). *Libros en pantalla de SQL Server 2014*. Obtenido de [https://msdn.microsoft.com/es-es/library/ms130214\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ms130214(v=sql.120).aspx)
- Microsoft. (2015). *Visual Studio Community*. Obtenido de <https://www.visualstudio.com/es-es/products/visual-studio-community-vs.aspx>
- Microsoft. (2016). *INFO: Microsoft Consulting Services Naming Conventions for Visual Basic*. Obtenido de [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
- Microsoft. (2016). *Información general acerca de .NET Framework*. Obtenido de [https://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
- Microsoft. (2016). *LINQ y ADO.NET*. Obtenido de [https://msdn.microsoft.com/es-ec/library/bb399365\(v=vs.110\).aspx](https://msdn.microsoft.com/es-ec/library/bb399365(v=vs.110).aspx)