



Asignatura: Ingeniería de Software

Profesor: Conde Sergio Daniel

Alumno: Estrella Alan Leonel

Índice

Análisis de proyecto.....	Pag.4
Historial de versiones.....	Pag.4
Información del proyecto.....	Pag.4
Antecedentes del proyecto.....	Pag.4
El proyecto y su contexto.....	Pag.5
Factibilidad Técnica.....	Pag.6
Factibilidad Operativa.....	Pag.7
Tablas de Ponderación.....	Pag.8
Factibilidad Económica.....	Pag.12
Terminando la Factibilidad.....	Pag.12
Informe Final.....	Pag.13
 Consigna 1.....	 Pag.14
 Consigna 2.....	 Pag.19
2.1(Introducción).....	Pag.19
2.1.1(Propósito).....	Pag.19
2.1.2(Alcance).....	Pag.19
2.1.3(Personal involucrado).....	Pag.20
2.1.4(Definiciones, acrónimos y abreviaturas).....	Pag.21
2.1.5(Referencia).....	Pag.21
2.1.6(Resumen).....	Pag.21
2.2.1(Perspectiva del producto).....	Pag.22
2.2.2(Funcionalidad del producto).....	Pag.22
2.2.3(Características de los usuarios).....	Pag.23
2.2.4(Restricciones).....	Pag.23
2.2.5(Suposiciones).....	Pag.23
2.2.6(Requisitos futuros).....	Pag.23
2.2.7(Requisitos específicos).....	Pag.24
2.3(Requisitos funcionales).....	Pag.29
2.3.1(Requisitos comunes de las interfaces).....	Pag.29
2.3.1.1(Interfaces del administrador).....	Pag.29

2.3.1.2(Interfaz de hardware).....	Pag.29
2.3.1.3(Interfaz de software).....	Pag.29
2.3.1.4(Interfaces de comunicación).....	Pag.29
2.3.2(Requisitos funcionales en detalle).....	Pag.29
2.3.3(Requisitos no funcionales).....	Pag.30
 Consigna 3.....	 Pag.22
 Consigna 4.....	 Pag.31
 Consigna 5.....	 Pag.32
Tabla de activos de información.....	Pag.33
Valorización de activos.....	Pag.33
Sistema de valorización.....	Pag.34
Clasificación de activos.....	Pag.35
Valoración activos de información.....	Pag.36
Análisis y Estimación de riesgos.....	Pag.36
Gestión y plan de tratamiento de riesgo.....	Pag.38
Consigna 6.....	Pag.45
Consigna 7.....	Pag.47
Consigna 8.....	Pag.48
Consigna 9.....	Pag.49
Consigna 10.....	Pag.49
Consigna 11.....	Pag.54

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
04/06/2020	1.0	Marchesse Martiniano	Software S.A	Estudio de factibilidad

Información del Proyecto

Empresa	Software S.A
Proyecto	Desarrollo de software para uso interno de la empresa
Fecha de preparación	10 de Mayo de 2020
Cliente	Arregla Todo
Líder de proyecto	Marchesse Martiniano

Antecedentes del Proyecto

Breve descripción de la visión del proyecto y sus orígenes:

- Desarrollar un software para la gestión de clientes, presupuesto y control de este mismo.
- Agilización de diversas actividades que realiza la empresa.
- Actualizar los servidores de la empresa.

- Desarrollo de una página web para el uso de los clientes.
- El proyecto fue propuesto por la empresa Arregla Todo, con el fin de mejorar su performance.

El proyecto y su contexto

Descripción del proyecto

El proyecto consta de la entrega de un software cuyo fin será mejorar y agilizar diversos procesos dentro de la empresa relacionados con la gestión de clientes y presupuestos. Habrá dos fechas donde se mostrarán al cliente entregas parciales del proyecto (25 de mayo y 09 de junio) y la entrega del producto terminado el 25 de junio con su correspondiente manual de usuario.

Objetivos

El proyecto consta de los siguientes objetivos:

1. Desarrollo de software para uso interno en la empresa.
2. Creación de una página web.
3. Actualización de los servidores web.

Accionar y beneficios:

1. Desarrollado con C# se creará un software que agilice la dinámica del proceso que se verá traducido en una optimización del tiempo y por consiguiente un aumento en las ganancias de la empresa.
2. Desarrollar una página web con HTML, CSS y JavaScript con el fin de darle a los clientes de “Arregla Todo” una forma sencilla de contactar con ellos, llegando a un mayor público.
3. Realizar una búsqueda en el mercado de una gama diversa de servidores y con el estudio de factibilidad realizado, indicar al cliente (teniendo en cuenta su contexto y/o posibilidades) cual es la opción que mejor se adapta a su situación.

Contexto del proyecto

1. Finalidad: Fines económicos.
2. Sector participante: Empresa.
3. Complejidad: Simple.

4. Naturaleza: Desarrollo Tecnológico.

Factibilidad Técnica

Desarrollo de página web

Ítem	Descripción	Coste
1	Desarrollo en base a una plantilla	\$11.625,92
2	Página web profesional standard	\$42.836,44
3	Página web a medida	\$60.567,93

Ítem 1 = costos: licencia para pack de plantillas + dominio (.com) por 1 año.

Ítem 2 = costos: dominio (.com) por 1 año

Ítem 3 = costos: dominio (.com) por 1 año

*El costo de desarrollo se encontrará en la factibilidad operativa.

*No habrá necesidad de contratar un servicio de hosting ya que se utilizarán servidores propios de la empresa.

Servidores

Ítem	Descripción	Cantidad	Costo
1	Servidor Dell PowerEdge T440 Xeon 4208 32gb Oficial Torre	1	\$497.042,44
2	Dell PowerEdge T140 Server Torre Xeon 32gb 2x1tb	1	\$248.805,44
3	HPE ProLiant ML110 Gen10 Intel Xeon 4208/16GB	1	\$181.569,22

Ítem 1 = costos: \$462.551 (servidor) + \$34.491,44 (Licencia Windows Server Essencial)

Ítem 2 = costos: \$214.314 (servidor) + \$34.491,44 (Licencia Windows Server Essencial)

Ítem 3 = costos: \$117.678,15 (servidor) + \$24.056,07 (Licencia Red Hat Enterprise Linux Server) + \$39.835 (Disco 2 Tb)

Factibilidad Operativa

RR. HH para el proyecto

Ítem	Descripción	Costo
A	1 Diseñador semi.senior 1 Desarrollador frontend semi senior 2 Desarrolladores backend semi senior 1 Analista senior	\$ 550.000
B	1 Diseñador semi senior 1 Desarrollador frontend semi senior 2 Desarrolladores backend (semi senior y junior) 1 Analista semi senior	\$ 440.000
C	1 Diseñador junior 1 Desarrollador frontend junior 2 Desarrollador backend (junior y trainee) 1 Analista junior	\$ 300.000

Ítem A = Costo: Diseñador semi senior (\$ 100.000) + Desarrollador frontend semi senior (\$ 100.000) + 2 Desarrolladores backend semi senior (\$ 200.000) + Analista senior (\$ 150.000)

Ítem B = Costo: Diseñador semi senior (\$ 100.000) + Desarrollador frontend semi senior (\$ 100.000) + Desarrollador backend semi senior (\$ 100.000) + Desarrollador backend junior (\$ 60.000) + Analista semi senior (\$ 100.000)

Ítem C = Costo: Diseñador junior (\$ 60.000) + Desarrollador frontend junior (\$ 60.000) + Desarrollador backend junior (\$ 60.000) + Desarrollador backend trainee (\$ 40.000) + Analista junior (\$ 80.000)

Capacitación en la herramienta

Ítem	Descripción	Costo
A	Capacitación sobre la herramienta (2 días de 4 horas)	\$ 600
B	Curso sobre la herramienta (5 días de 3 horas)	\$ 800
C	Curso sobre la herramienta (7 días de 2 horas)	\$ 1.300

Tablas de ponderación

Elección de desarrollo web

Atributos	Peso	Diseño en base a plantilla	Diseño profesional standard	Diseño web a medida
Interfaz amigable	25%	50	60	100
Interacción con el usuario	25%	30	70	100
Velocidad de carga de la página	25%	90	80	70
Seguridad	20%	60	75	75
Optimización web	5%	60	90	100

*Optimización web se refiere a mejorar la compatibilidad del sitio a los motores de búsqueda con el fin de aumentar al máximo la visibilidad de palabras clave.

- Web por plantilla = $25*50 + 25*30 + 25*90 + 20*60 + 5*60 = 5750$
- Web profesional estándar = $25*60 + 25*70 + 25*80 + 20*75 + 5*90 = 7200$
- Web a medida = $25*100 + 25*100 + 25*70 + 20*75 + 5*100 = 8750$

Informe de decisión

Se eligió diseñar una página web a medida del cliente ya que un diseño estándar o uno basado en plantillas podría no terminar de cumplir con sus expectativas y a lo que apunta la empresa. Poder desarrollador teniendo en todo momento las ideas del cliente nos permite obtener un producto que dejará totalmente cubierta sus necesidades, al mismo tiempo que, se creará una interfaz más llamativa y que siga los ideales de la empresa. Por lo tanto, el poder decidir cada aspecto en cuanto al diseño permite salirse de la típica página dentro de su rubro y ser otra forma de atraer potenciales clientes a la hora de elegir con quien repararan sus electrodomésticos.

Elección de servidor

Atributo	Peso	PowerEdge T440	PowerEdge T140	HPE Proliant
Compatibilidad con diversos Sistemas Operativos	10%	80	80	40
Escalabilidad	10%	60	60	60
Memoria RAM	30%	75	70	65
Tamaño (físico)	5%	40	55	85
Velocidad de procesamiento	30%	70	65	60
Capacidad de almacenamiento	15%	90	50	85

- Servidor Poweredge T440 = $10*80 + 10*60 + 30*75 + 5*40 + 30*70 + 15*90 = 7300$
- Servidor Poweredge T140 = $10*80 + 10*60 + 30*70 + 5*55 + 30*65 + 15*50 = 6475$

- Servidor HPE Proliant= $10*40 + 10*60 + 30*65 + 5*85 + 30*60 + 15*85 = 6450$

Informe de decisión

El servidor PowerEdge T440 tiene una gran compatibilidad con muchos sistemas operativos, se puede optar por utilizar uno de Linux, aunque para este caso y cliente se utilizará Windows Server, pero esta cualidad puede ser aprovechada en un futuro para reducir costes de licencia. En lo que respecta a velocidad de procesamiento es ligeramente superior a los otros servidores de la comparativa y a esto sumarle que el servidor con mayor cantidad de memoria RAM. En cuanto a su tamaño físico no es la mejor opción, pero tampoco es un atributo con mayor relevancia a la hora de tomar la decisión. Por otro lado, cuenta con una excelente capacidad para el almacenamiento de grandes volúmenes de datos. Si bien este servidor puede exceder lo que hoy en día la empresa puede necesitar por su condición de Pyme, gracias a su escalabilidad y las características que presenta el producto base, es una gran inversión a futuro y una preocupación menos para la empresa por los siguientes años.

Elección de RR. HH

Atributo	Peso	Equipo A	Equipo B	Equipo C
Motivación	15%	10	55	100
Experiencia	15%	80	70	60
Profesionalidad	25%	80	70	50
Velocidad del desarrollado	20%	70	70	65
Calidad del desarrollo	15%	80	70	50
Comunicación con la empresa	5%	80	80	50
Organización dentro del equipo	5%	80	80	45

- Equipo A = $15*10 + 15*80 + 25*80 + 20*70 + 15*80 + 5*80 + 5*80 = 6750$

- $\text{Equipo B} = 15 \cdot 55 + 15 \cdot 70 + 25 \cdot 70 + 20 \cdot 70 + 15 \cdot 70 + 5 \cdot 80 + 5 \cdot 80 = 6875$
- $\text{Equipo C} = 15 \cdot 100 + 15 \cdot 60 + 25 \cdot 50 + 20 \cdot 70 + 15 \cdot 50 + 5 \cdot 50 + 5 \cdot 45 = 6275$

Informe de decisión

La mejor decisión son los recursos del equipo B, si bien no parece haber gran diferencia entre equipos B y C, la experiencia es un factor importante a la hora de desarrollar un proyecto. Creemos que a su vez un equipo lleno de experiencia como el A es excesivo para un desarrollo simple como es el caso y un gasto innecesario. Por otro lado, la calidad del código y velocidad de desarrollo es parecida entre equipos A y B, pero la mayor motivación de estos últimos podrá desembocar en un mejor proyecto por el entusiasmo de los recursos. Por último, tanto la organización dentro del equipo B y su comunicación están a la altura de la profesionalidad que necesita este proyecto.

Elección de la capacitación

Atributo	Pesos	2 Días 4 Hs (A)	5 Días 3 Hs (B)	7 Días 2 Hs (C)
Comprensión de la herramienta	30%	80	90	100
Menor carga horaria	15%	40	60	80
Menor agotamiento mental	20%	50	70	80
Tiempo para seguir en su puesto	35%	20	45	70

- $\text{Capacitación A} = \text{Costo: } 30 \cdot 80 + 15 \cdot 40 + 20 \cdot 50 + 35 \cdot 20 = 4700$
- $\text{Capacitación B} = \text{Costo: } 30 \cdot 90 + 15 \cdot 60 + 20 \cdot 70 + 35 \cdot 45 = 6575$
- $\text{Capacitación C} = \text{Costo: } 30 \cdot 100 + 15 \cdot 80 + 20 \cdot 80 + 35 \cdot 70 = 8250$

Informe de decisión

El software creado es de una complejidad simple, por lo tanto, no se requiere una gran inversión de tiempo para la capacitación del personal en cuanto a su uso. Esto es un beneficio ya que no representará un gran costo, además de que, no se deberá de prescindir de personal esencial durante la cantidad de horas que dure la capacitación. Por último, respecto a la comprensión, es preferible dedicar más días para tener un personal con gran conocimiento y dominio de la herramienta, pensando que a futuro dentro de la misma compañía podrán capacitarse a los futuros empleados.

Factibilidad económica

La empresa Arregla Todo es una pyme y por lo cual cuenta con un presupuesto acotado de \$1.000.000 (pesos un millón). Con el estudio previamente realizado le brindaremos al cliente una serie de alternativas en caso de no poder costear la elección óptima según las ponderaciones, pero siempre teniendo presente que el proyecto se va a realizar y finalizar.

Terminando la Factibilidad

Factibilidad		Descripción	Costo Total
Técnica (Servidores)		PowerEdge T440	\$497.042,44
Técnica (Desarrollo web)		Página web a medida	\$60.567,93
Operativa	RR. HH	Equipo B	\$440.000
	Capacitación	C	\$1.300
		Costo total Factibilidad	\$998.940,37

Informe Final

Tras realizar las distintas Factibilidades técnicas con sus respectivas tablas de ponderación se llegó a distintas elecciones. En lo que respecta a servidores se optó por el Dell PowerEdge T 440, ya que, cuenta con adaptabilidad para varios sistemas operativos, buena relación entre lo que la empresa necesita y sus prestaciones generales (RAM, procesador y almacenamiento base). Por último, cuenta con una escalabilidad aceptable pensando en el crecimiento de la empresa.

La página web será diseñada a medida de lo que la empresa requiere, esto permitirá que cada necesidad sea cubierta y, por otro lado, su interfaz totalmente personalizada atraerá potenciales cliente, que se traducirá en mayores ganancias.

Una vez definidas las factibilidades técnicas, se asignó un equipo para este proyecto. Será un mix entre personas con experiencia y otros recursos que sean relativamente nuevos en cuanto a experiencia, esto se debe a que sin perder de vista la seriedad del proyecto, su realización es de una complejidad simple. Al momento de capacitar a los empleados de Arregla Todo se busca que tengan un gran dominio de la herramienta y que por mas que dure más días será dosificada la carga horaria para no interrumpir abruptamente su labor cotidiano.

- 1) Definir Metodología Scrum contemplando dos Sprint para desarrollar el Proyecto.

Sprint N°: 1

Fecha de inicio: 10 de Mayo del 2020 - Fecha de finalización: 25 de Mayo del 2020

Product Backlog – Primer Sprint

Backlog Item	Estimación
Reuniones con el cliente	15
Relevamiento requisitos funcionales	4
Relevamiento requisitos no funcionales	4
Casos de uso	6
Diseño del UML	4
Replanificar lógica del sistema por cambios del cliente	10
Diseño de interfaz de las pantallas	8
Documentar	6

Sprint Backlog – Primer Sprint (primera semana)

Tareas	L	M	M	J	V
Reunión con el cliente	5				5
Relevamiento requisitos funcionales	3	1			
Relevamiento requisitos no funcionales		4			
Casos de uso		3	3		
Diseño del UML			3	1	
Documentar				5	1

Sprint Backlog – Primer Sprint (segunda semana)

Tareas	L	M	M	J	V
Analizar cambios surgidos por petición cliente	8	2			
Incorporar nuevos requisitos funcionales		4			
Incorporar nuevos casos de uso		2	2		
Modificar UML			2	2	
Reunión con el cliente			2.5		2.5
Documentar				3	3

*Se estima que el cliente siempre va a sugerir un cambio y/o incorporación luego de la primera presentación del sistema, por lo que respecto a la segunda semana del primer sprint se dedicará tiempo a dichos cambios.

Sprint N°: 2

Fecha de inicio: 26 de Mayo del 2020 - Fecha de finalización: 09 de Junio del 2020

Product Backlog - Segundo Sprint

Backlog Item	Estimación
Diseño de base de datos	5
Implementar diseño de base de datos	15
Registrar cliente	4
Registrar presupuesto	4
Modificar cliente	4
Modificar presupuesto	4
Consultar cliente	4
Consultar presupuesto	4
Anular e imprimir cliente	4
Anular e imprimir presupuesto	4
Manejo de excepciones	10
Documentar	8

Sprint Backlog - Segundo Sprint (primera semana)

Tareas	L	M	M	J	V
Diseño de base de datos	5				
Codificar base de datos		5	5	5	
Registrar cliente	2	2			
Registrar presupuesto			2		2
Modificar cliente		2	2		
Modificar presupuesto				2	2
Consultar cliente	2	2			
Consultar presupuesto				2	2
Documentar		2	3		3

Sprint Backlog - Segundo Sprint (segunda semana)

Tareas	L	M	M	J	V
Anular cliente	2	2			
Anular presupuesto			2	2	
Imprimir cliente	2	2			
Imprimir presupuesto				2	2
Testear módulo clientes	3			2	
Testear módulo presupuesto		3	2		
Manejo de excepciones	3	1	3		3
Documentar		2	2	1	3

Sprint N°: 3

Fecha de inicio: 10 de Junio del 2020 - Fecha de finalización:25 de Junio del 2020

Product Backlog - Tercer Sprint

Backlog Item	Estimación
Diseñar interfaz de pantallas	8
Desarrollo de interfaz visual del sistema	24
Corregir errores	10
Finalizar redacción de la documentación	6

Sprint Backlog Tercer Sprint (primera semana)

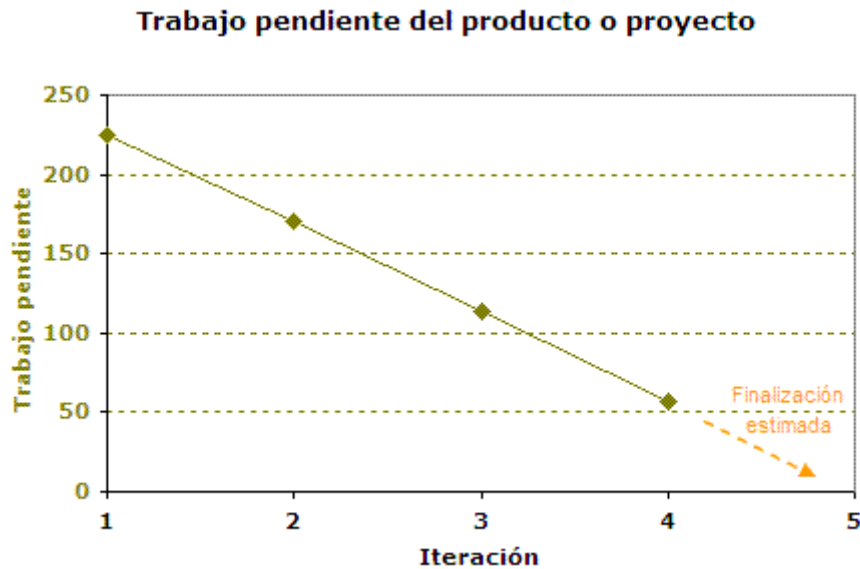
Tareas	L	M	M	J	V
Diseñar interfaz de pantallas	8				
Desarrollo de interfaz visual		6	6	6	6
Testear interfaz visual		2	2	2	1
Documentar		2	2	2	2

Sprint Backlog Cuarto Sprint (segunda semana)

Tareas	L	M	M	J	V
Prueba de la aplicación en general	4	4	4	4	
Finalizar redacción de documentación		1	2	1	2

Gráfico Burdown

Para medir que el progreso a lo largo del sprint vaya de manera correcta se utilizará un gráfico Burdown Chart. Este será actualizado constantemente por el ScrumMaster con el fin de que su seguimiento detalle si se están cumpliendo los plazos establecidos de cada sprint. Para saber que el proyecto está marchando correctamente la línea de trabajo pendiente debe estar cercana a la línea de iteración, en caso de sobrepasarla significa que el equipo está dedicando mas horas de lo que fue estipulado, por lo cual, se subestimó al sprint. En contraparte, si se encuentra muy por debajo denota que el sprint fue sobreestimado.



- 2) Definir Requerimientos Funcionales y no Funcionales utilizando el modelo SRS-830 y desarrollar los Caso de Uso que correspondan (los casos de uso se encuentra contenidos en el modelo SRS-830).

2.1 Introducción

Este documento es una Especificación de Requisitos de Software (ERS) para el sistema de gestión de procesos y control tanto de clientes como presupuestos. Esta especificación se estructura conforme el estándar dado por la IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

2.1.1 Propósito

El actual documento tiene como fin definir requisitos funcionales, no funcionales para el desarrollo de un software que se encargará de la gestión de distintos participantes y/o acciones dentro de la empresa de reparaciones. Este software será utilizado por personal administrativo de la empresa “Arregla Todo”.

2.1.2 Alcance

Esta especificación de requisitos está dirigida para el o los usuarios del sistema, cuyo objetivo es el control y agilización de diversas actividades que realiza la empresa (gestión de clientes, presupuesto y control de este mismo).

2.1.3 Personal involucrado

Nombre	Solis Marco Antonio
Rol	Product Owner
Responsabilidad	Representa al cliente.
Información de contacto	Marco.antonio@gmail.com

Nombre	Godoy Jose
Rol	Scrum master
Responsabilidad	Gestión del proyecto. Resolver impedimentos. Asegurar un equipo funcional y productivo. Promover prácticas y valores de Scrum.
Información de contacto	G.jose@gmail.com

Nombre	Carrizo Daiana
Rol	Desarrolladora
Responsabilidad	Desarrollo.
Información de contacto	D.M.Carrizo@gmail.com

Nombre	Estrella Alan Leonel
Rol	Desarrollador
Responsabilidad	Desarrollo.
Información de contacto	ealanleonel@gmail.com

Nombre	Franco German
Rol	Desarrollador
Responsabilidad	Desarrollado.
Información de contacto	Franco.German@gmail.com

2.1.4 Definiciones, acrónimos y abreviaturas

Usuario	Persona que utilizará el software
ERS	Especificación de Requisitos Software
RF	Requisito Funcional
RNF	Requisito No Funcional
Input	Pulsación de tecla

2.1.5 Referencia

Título del Documento	Referencia
Standard IEEE 830 – 1998	IEEE

2.1.6 Resumen

El documento tendrá tres apartados bien diferenciados. El primero se centra en presentar este mismo y una ligera introducción respecto al software a desarrollar.

El segundo apartado se realizará una introducción del sistema en general, para poder remarcar cuáles serán sus principales funciones, datos que referencien a este y factores, restricciones, supuestos y dependencias que pueden afectar a la hora de realizar el desarrollo.

La última se centrará en exponer los requisitos considerados primordiales para el sistema en cuestión.

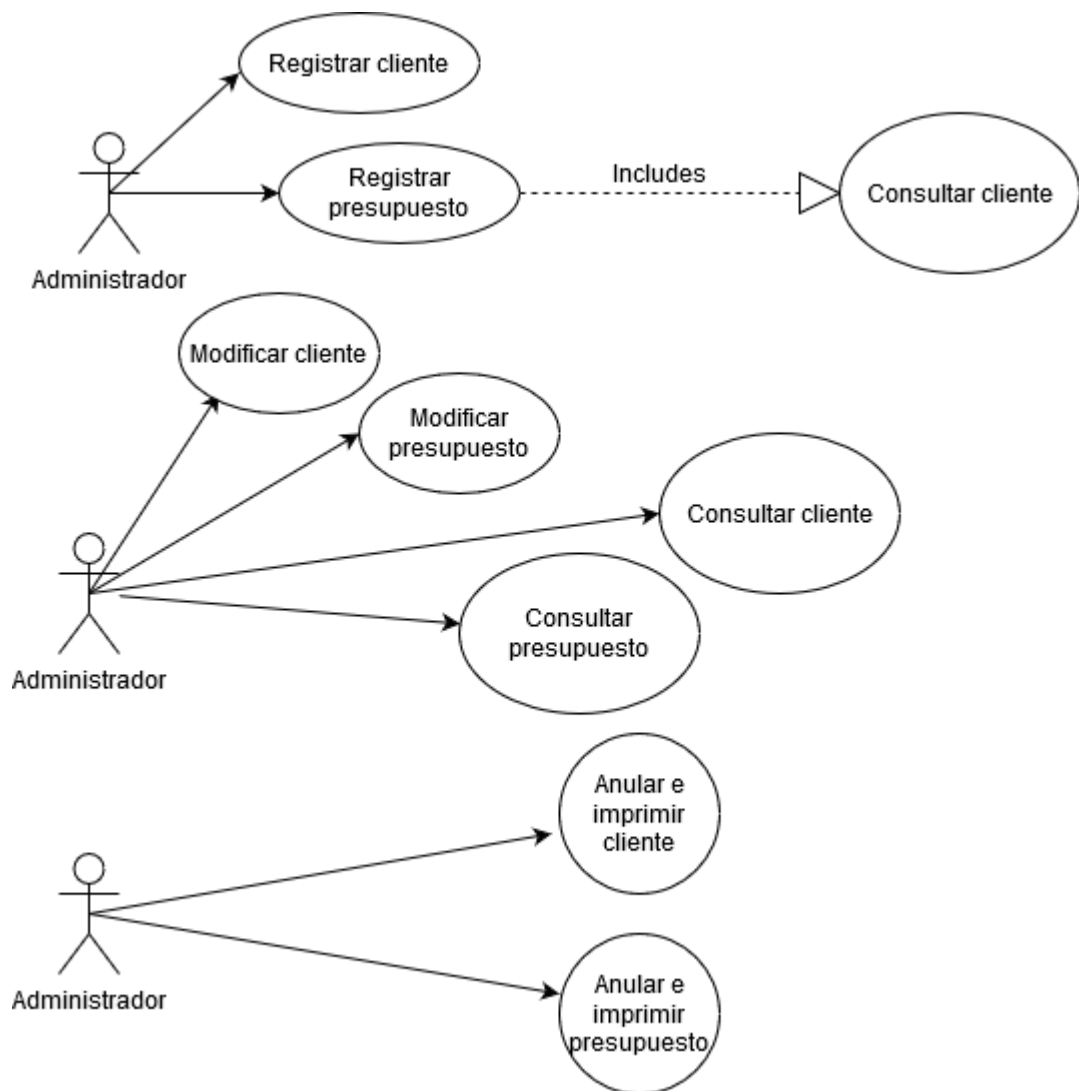
2.2 Descripción general

2.2.1 Perspectiva del producto

El software destinado a la empresa “Arregla Todo” podrá ser usado tanto en Windows como Linux y su fin será agilizar las tareas de gestión y control dentro de la empresa que, hasta este momento, se realizaban de forma manual.

2.2.2 Funcionalidad del producto

Se encuentra separados a fines prácticos de una mejor visualización.



2.2.3 Características de los Usuarios

Tipo de usuario	Administrador
Actividades	Control y manejo del sistema en general

2.2.4 Restricciones

- Los servidores deben atender consultas concurrentes.
- Diseño e implementación sencilla.
- Interfaz para ser usada en una pantalla de 1366x768.
- Modelo cliente/servidor.
- Servidor “ON” todo el día.

2.2.5 Suposiciones

- Se da por supuesto que el o los equipos que utilicen el software contarán con el hardware correspondiente para su buen funcionamiento.

2.2.6 Requisitos Futuros

- Cobrar con débito
- Cobrar con crédito
- Confeccionar factura

2.3 Requisitos específicos

Requerimientos funcionales

Identificación del requerimiento	RF01
Nombre del requerimiento	Registrar Cliente
Características	El administrador deberá registrar los clientes para poder recibir el servicio de la empresa.
Descripción del requerimiento	El sistema permite al administrador registrar clientes. El cliente suministrará sus datos personales: apellido, nombre, DNI, teléfono y dirección.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF02
Nombre del requerimiento	Modificar Cliente
Características	El administrador podrá modificar los clientes.
Descripción del requerimiento	El sistema permite al administrador modificar clientes. Cualquiera de sus datos personales podrá ser modificados, ya sea: apellido, nombre, DNI, teléfono y dirección.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF03
Nombre del requerimiento	Consultar Cliente
Características	El administrador podrá consultar al cliente.
Descripción del requerimiento	El sistema permite al administrador consultar la información del cliente en cualquier momento.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF04
Nombre del requerimiento	Anular e Imprimir Cliente
Características	El administrador podrá anular al cliente o imprimir información de este.
Descripción del requerimiento	El sistema permite al administrador dar de bajar a un cliente o varios (anular) o si es requerido en algún momento imprimir la información relacionada al cliente.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF05
Nombre del requerimiento	Registrar Presupuesto
Características	El administrador deberá registrar los presupuestos.
Descripción del requerimiento	El sistema permite al administrador registrar presupuestos. Luego de esto se deberá consultar el cliente (RF01).
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF06
Nombre del requerimiento	Modificar Presupuesto
Características	El administrador podrá modificar los presupuestos.
Descripción del requerimiento	El sistema permite al administrador modificar presupuestos. Cualquiera de sus datos, ya sean: precio del arreglo, cantidad de piezas usadas, costo de mano de obra y personal que lo realizó.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF07
Nombre del requerimiento	Consultar Presupuesto
Características	El administrador podrá consultar los presupuestos.
Descripción del requerimiento	El sistema permite al administrador consultar presupuestos.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RF08
Nombre del requerimiento	Anular e Imprimir Presupuesto
Características	El administrador podrá anular el presupuesto o imprimir información de este.
Descripción del requerimiento	El sistema permite al administrador dar de bajar a un presupuesto (anular) o si es requerido en algún momento imprimir la información relacionada al presupuesto.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF01
Nombre del requerimiento	Interfaz del Sistema
Características	La interfaz deberá ser sencilla.
Descripción del requerimiento	El software deberá ser de uso intuitivo para el administrador.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF02
Nombre del requerimiento	Idioma
Características	La interfaz deberá estar en español.
Descripción del requerimiento	El software deberá tener versión en español.
Prioridad del requerimiento	Media

Identificación del requerimiento	RNF03
Nombre del requerimiento	Tiempo de Respuesta
Características	Tiempo que tarda el software en responder.
Descripción del requerimiento	Tiempo que le llevara al programa responder ante los distintos inputs del administrador.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF04
Nombre del requerimiento	Consumo de RAM
Características	Recurso consumido por el software.
Descripción del requerimiento	El software deberá evitar consumir gran cantidad de memoria RAM.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF05
Nombre del requerimiento	Base de Datos
Características	Base de datos utilizada.
Descripción del requerimiento	Que base de datos estará vinculada al software para las distintas operaciones que se realizarán.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF06
Nombre del requerimiento	Impresora
Características	Impresora.
Descripción del requerimiento	Que impresora se utilizará para la impresión.
Prioridad del requerimiento	Baja

Identificación del requerimiento	RNF07
Nombre del requerimiento	Diseño de las Impresiones
Características	Formato de impresiones.
Descripción del requerimiento	Formato de las impresiones de clientes y presupuesto.
Prioridad del requerimiento	Baja

Identificación del requerimiento	RNF08
Nombre del requerimiento	Seguridad y Lógica de Datos
Características	Seguridad y Lógica de Datos.
Descripción del requerimiento	Seguridad del software para el manejo de los datos.
Prioridad del requerimiento	Alta

Identificación del requerimiento	RNF09
Nombre del requerimiento	Lenguaje de Programación
Características	Lenguaje para programar.
Descripción del requerimiento	En que lenguaje se programará el software.
Prioridad del requerimiento	Baja

2.3.1 Requisitos Comunes de las Interfaces

2.3.1.1 Interfaces de administrador

La interfaz consiste en un conjunto de ventanas con botones, listas y campos de textos. Deberá ser construida en función de este software a desarrollar.

2.3.1.2 Interfaz de hardware

Será primordial contar con los siguientes requisitos básicos para el correcto funcionamiento del programa:

- Adaptadores de Red
- Procesador I3
- Memoria RAM de 2 GB
- Mouse
- Teclado

2.3.1.3 Interfaces del software

- Sistema operativo Windows o Linux

2.3.1.4 Interfaces de comunicación

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet.

2.3.2 Requisitos funcionales en detalle

2.3.2.1 Requisito funcional 01

- Registro de Cliente: El administrador del sistema podrá registrar clientes.

2.3.2.2 Requisito funcional 02

- Modificar Cliente: El administrador podrá modificar datos de los clientes.

2.3.2.3 Requisito funcional 03

- Consultar Cliente: El sistema administrará información general del cliente.

2.3.2.4 Requisito funcional 04

- Anular e Imprimir Cliente: Dar de baja un cliente o realizar una impresión de la información general de este.

2.3.2.5 Requisito funcional 05

- Registro de Presupuesto: El administrador del sistema podrá registrar presupuestos.

2.3.2.6 Requisito funcional 06

- Modificar Presupuesto: El administrador podrá modificar datos del presupuesto.

2.3.2.3 Requisito funcional 07

- Consultar Presupuesto: El sistema administrará información general del presupuesto.

2.3.2.4 Requisito funcional 08

- Anular e Imprimir Presupuesto: Dar de baja un presupuesto o realizar una impresión de la información general de este.

2.3.3 Requisito no funcional

2.3.3.1 Fiabilidad

- La interfaz que utilizará el administrador será lo más intuitiva posible.
- La interfaz del software deberá estar en español.

2.3.3.2 Requisitos de rendimiento

- Tiempo de respuesta rápido del software.
- Contar con la RAM necesaria para su correcto funcionamiento.

2.3.3.3 Disponibilidad

- Servidor activo los 7 días durante 24 horas.
- El tipo de base de datos no es primordial.

2.3.3.4 Seguridad

- Usuario y contraseña para ingresar al sistema.
- Cifrado de datos.
- Seguridad y facilidad respecto al manejo de datos.

2.3.3.5 Miscelánea

- Que marca de impresora se utilizará.
- Formato de cómo serán las impresiones de los clientes y presupuesto.
- Lenguaje que se utilizará para la programación del sistema.

4) Defina y justifique el Ciclo de vida a utilizar en el desarrollo del Proyecto.

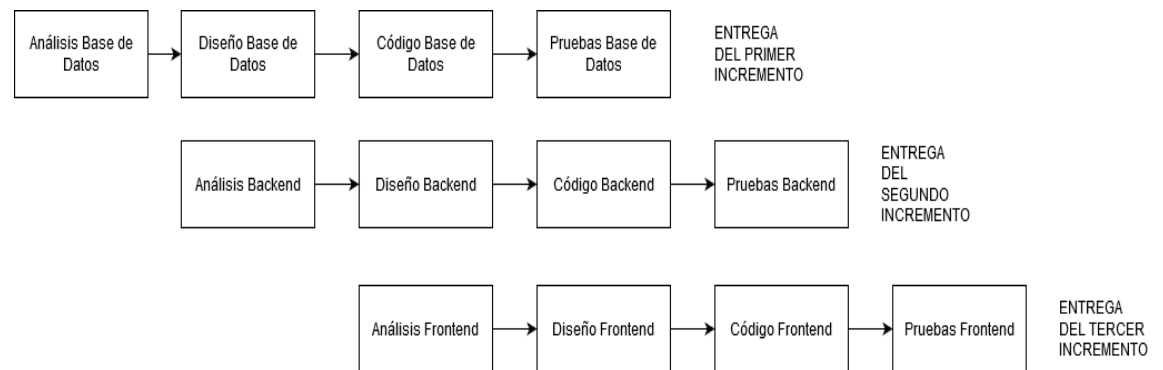
Para este proyecto la metodología de trabajo que se usará es Scrum, donde hay una muestra de avances al final de cada sprint y es el momento donde el cliente

sugiere cambios. Esta característica se acopla correctamente al ciclo de vida incremental en la cual se presenta

al usuario las implementaciones para obtener feedback y resolver o exponer nuevas dudas y deseos respecto al sistema. Aunque en este proyecto los requisitos ya se encontrarán documentados, existe la posibilidad de añadir nuevos requisitos propuestos por el usuario al finalizar el sprint (punto donde Scrum y el desarrollo incremental coinciden).

Se logra justificar el uso modelo incremental con desarrollo exploratorio y la

metodología de trabajo Scrum porque realizando los ajustes pertinentes, ambas tienen una visión parecida a la hora de organizar el trabajo dentro del proyecto.



5. Elaborar un Plan de Riesgos aplicando el modelo PDCA con las acciones a tomar. Debe haber un mínimo de 5 riesgos.

Tabla de activos de información

ID	Activo	Descripción	Propietario
1	Servicio Telefónico	Comunicación	Proveedor
2	Servicio Eléctrico	Electricidad	Proveedor
3	Servicio de Internet	Internet	Proveedor
4	RR.HH	Personal	Gerencia de RR.HH
5	Lugar Físico	Ámbito laboral	Depto. Servicios Generales
6	Sistema de Aplicación	Software aplicativo	Depto. Tecnología
7	Equipamiento Informático	PCs, impresoras, servidores	Depto. Tecnología
8	Información Digital	Información en soporte o formato electrónico	Depto. Atención a Clientes
9	Información en papel	Información en formato papel o soporte tangible	Depto. Atención a Clientes
10	Instalación Eléctrica	Infraestructura eléctrica	Depto. Servicios General
11	Instalación de Internet	Infraestructura de las redes	Depto. Servicios General
12	Identidad Empleados/Clientes	Validación	Depto. Atención a Clientes
13	Página Web	Página de la empresa	Depto. Tecnología

Valorización de los activos

Se detalla cual será el sistema para la valorización de los activos de la empresa según la importancia de estos para la organización y su potencial daño en caso de que se vieses afectados. Con el fin de quien lea el documento comprenda que significan las determinaciones tomadas.

Sistema de valoración

Valorización de la disponibilidad

Valor	Criterio
0	No aplica o no es relevante
1	Debe estar disponible al menos el 10% del tiempo
2	Debe estar disponible al menos el 50% del tiempo
3	Debe estar disponible al menos el 99% del tiempo

Valoración de la integridad

Valor	Criterio
0	No aplica o no es relevante
1	No es relevante los errores que tenga o información que falte
2	Tiene que estar correcto y completo al menos un 50%
3	Tiene que estar correcto y completo al menos un 95%

Valoración de la confidencialidad

Valor	Criterio
0	No aplica o no es relevante
1	Daños muy bajos, el incidente no trascendería del área afectada
2	Serían relevantes, el incidente implicaría otras áreas
3	Los daños serían catastróficos, la imagen y reputación de la organización se verían afectadas

Clasificación de Activos

ID	Activo	Confidencialidad	Integridad	Disponibilidad
1	Servicio Telefónico	1	3	3
2	Servicio Eléctrico	1	3	3
3	Servicio de Internet	1	3	3
4	RR.HH	2	3	3
5	Lugar Físico	2	3	3
6	Sistema de Aplicación	2	3	3
7	Equipamiento Informático	2	3	3
8	Información Digital	3	3	3
9	Información en papel	3	3	3
10	Instalación Eléctrica	1	3	3
11	Instalación de Internet	1	3	3
12	Identidad Empleados/Clientes	2	3	3
13	Página Web	3	3	3

Valoración de Activos de Información

ID	Activo	Valor (A; M; B)
1	Servicio Telefónico	ALTO
2	Servicio Eléctrico	ALTO
3	Servicio de Internet	ALTO
4	RR.HH	ALTO
5	Lugar Físico	MEDIO
6	Sistema de Aplicación	ALTO
7	Equipamiento Informático	ALTO
8	Información Digital	ALTO
9	Información en papel	ALTO
10	Instalación Eléctrica	ALTO
11	Instalación de Internet	ALTO
12	Identidad Empleados/Clientes	ALTO
13	Página Web	ALTO

Análisis y Estimación de Riesgos

ID	Activo	Amenaza	Clasificación	P.O.	Vulnerabilidad	Impacto
1	Servicio Telefónico	Corte Suministro	Tecnología	B	Falta Plan “B”	MA
2	Servicio Eléctrico	Corte Suministro	Tecnología	B	Falta Plan “B”	MA
3	Servicio de Internet	Corte Suministro	Tecnología	B	Falta Plan “B”	MA
4	RR.HH	Divulgación de Información	Humana	B	Falta de Acuerdo de Confidencialidad	MA

5	Lugar Físico	Deterioro Edificio	Física	MB	Falta de Mantenimiento	M
6	Sistema de Aplicación	Caída del Sistema	Tecnología	B	Problema Hard/Soft	MA
7	Equipamiento Informático	Robo y Fuga de Información Robo y hurtos de equipos	Tecnología	B	Sin bloqueos Puerto USB/Acceso web a discos virtuales	A
8	Información Digital	Pérdida de Información	Tecnología	B	Falta de Backup	MA
9	Información en papel	Pérdida de Información	Operacional	B	Proceder inadecuado en archivado y manejo de los papeles	MA
10	Instalación Eléctrica	Deterioro Infraestructura Cableado Eléctrico	Tecnología	M	Falta de Mantenimiento	A
11	Instalación de Internet	Deterioro infraestructura Cableado Internet	Tecnología	B	Falta de Mantenimiento	A
12	Identidad Empleados/Clientes	Robo de Identidad	Operacional	B	Falta de Validación	MA
13	Página Web	Intrusión, Fuga de Información y Pérdida de Información	Tecnología	B	No contar con Firewall No validar Inputs	MA

Gestión y Plan de Tratamiento de Riesgo

Activo de Información	Servicio Telefónico
Amenaza Identificada	Corte Suministro
Vulnerabilidad que podía explotar	Falta Plan “B”
Valor Activo de Información	ALTO
Accionar	
Mitigar	
Transferir	X
Aceptar	
Como	Contratar otro servicio telefónico aparte
Quién	Depto. Servicio Generales
Cuando	Junio 2020
Recursos	Proveedor

Activo de Información	Servicio Eléctrico
Amenaza Identificada	Corte Suministro
Vulnerabilidad que podía explotar	Falta Plan “B”
Valor Activo de Información	ALTO
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Grupo Electrónico
Quién	Depto. Servicio Generales
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Servicio Internet
Amenaza Identificada	Corte Suministro
Vulnerabilidad que podía explotar	Falta Plan “B”
Valor Activo de Información	ALTO
Accionar	
Mitigar	
Transferir	X
Aceptar	
Como	Tener un segundo servicio de internet
Quién	Depto. Servicios Generales
Cuando	Junio 2020
Recursos	Proveedor

Activo de Información	RR.HH
Amenaza Identificada	Divulgación de Información
Vulnerabilidad que podía explotar	Falta Acuerdos de Confidencialidad
Valor Activo de Información	ALTO
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Acuerdo de Confidencialidad
Quién	Gerencia RR.HH
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Lugar Físico
Amenaza Identificada	Deterioro Edificio
Vulnerabilidad que podía explotar	Falta de Mantenimiento
Valor Activo de Información	M
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Destinar personal al mantenimiento del edificio
Quién	Depto. Servicio Generales
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Sistema de Aplicación
Amenaza Identificada	Caída del Sistema
Vulnerabilidad que podía explotar	Problema de Hardware/Software
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Mantenimiento Hardware/Software
Quién	Depto. Tecnología
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Equipo Informático
Amenaza Identificada	Robo y fuga de información / Robo y hurto de equipos
Vulnerabilidad que podía explotar	Puertos USB abiertos, falta de seguro en los equipos informáticos en caso de robo
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	X
Aceptar	
Como	Bloquear puertos USB / Contratar aseguradora
Quién	Depto. Tecnología / Empresa aseguradora
Cuando	Junio 2020
Recursos	Propios / Aseguradora

Activo de Información	Información Digital
Amenaza Identificada	Pérdida de Información
Vulnerabilidad que podía explotar	Falta de backup
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Generar copias de seguridad cada semana y tener estos fuera del edificio
Quién	Depto. Tecnología
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Información en Papel
Amenaza Identificada	Pérdida de Información
Vulnerabilidad que podía explotar	Proceder inadecuado en archivado y manejo de los papeles
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Política de manejo de la información, comprar máquina para triturar papeles
Quién	Depto. Tecnología
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Instalación Eléctrica
Amenaza Identificada	Deterioro infraestructura cableado eléctrico
Vulnerabilidad que podía explotar	Falta mantenimiento
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Acuerdo Nivel de Servicio
Quién	Depto. Servicios Generales
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Instalación de Internet
Amenaza Identificada	Deterioro infraestructura cableado de internet
Vulnerabilidad que podía explotar	Falta mantenimiento
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Acuerdo Nivel de Servicio
Quién	Depto. Servicios Generales
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Identidad Empleados/Clientes
Amenaza Identificada	Robo de Identidad
Vulnerabilidad que podía explotar	Falta de Validación
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Usuario y contraseña, capacitación sobre seguridad de la información
Quién	Depto. Tecnología
Cuando	Junio 2020
Recursos	Propios

Activo de Información	Página Web
Amenaza Identificada	Intrusión, Fuga de Información y Pérdida de Información
Vulnerabilidad que podía explotar	No contar con Firewall/ No validar Inputs
Valor Activo de Información	A
Accionar	
Mitigar	X
Transferir	
Aceptar	
Como	Implementar Firewall y validar inputs
Quién	Depto. Tecnología
Cuando	Junio 2020
Recursos	Propios

6. Aplique en el Diseño del Sistema:

6.1. El Modelo de Repositorio detallando como va a utilizar cada componente

Para la realización del proyecto el equipo de desarrollo utilizará como repositorio GitHub. Esta es una herramienta cuya función es gestionar un proyecto y a su vez, tener un control de versión de código basándose en un sistema centralizado. Cuenta con ventajas que refuerzan esta elección:

- Permite compartir grandes cantidades de datos sin tener que transmitir explícitamente de un sistema a otro.
- Obliga a un compromiso entre los subsistemas y la base de datos general.
- No es necesario conocer la producción y distribución de los datos ajenos a nuestro sistema.
- Tanto copia, seguridad, protección y control de acceso están centralizadas.
- Facilita el trabajo en equipo.
- Es compatible independientemente del sistema operativo.

6.2. El Modelo Cliente – Servidor

Arregla Todo solicitó la compra un servidor para su empresa, por lo cual, es lógico la implementación de un modelo Cliente – Servidor para sacar el mayor beneficio a este mismo. Como primera medida habrá que acondicionar un lugar específico dentro de la fachada del edificio para mantener resguardado el servidor de algún daño o inconveniente a nivel hardware. Por otro lado, utilizar este modelo nos permite separar todo lo relacionado a peticiones (impresiones, correo electrónico, web de Arregla Todo, consultar información de clientes, etc.) y así agilizar al mismo tiempo que se centraliza la gestión de la información. Con lo cual se facilita y clarifica el diseño del sistema.

Cuenta con otras ventajas también:

- Escalabilidad: Permite en cualquier momento y fácilmente el crecimiento horizontal (mayor número de servidores con las mismas prestaciones) o vertical (comprar equipo con mayores prestaciones) del modelo. Esto resulta útil teniendo en mente que la empresa eventualmente podría crecer en número de clientes, servicios prestados, etc.
- Fácil mantenimiento: Permite reemplazar, reparar o actualizar sin afectar o de forma mínima a los clientes.
- Seguridad: Esta asegurada la seguridad de las transacciones, las redes Cliente – Servidor evitan por parte del cliente el acceso a las IP's por lo que se dificulta el rastreo y/o hackeo de los usuarios.

6.3. El Modelo de Capas.

Para garantizar aún más la seguridad, integridad y controlar quienes acceden a los datos, se dividirá el sistema en un modelo de capas. En este existirán distintos roles y usuarios con privilegios variados y un sistema de validación, según a la información y datos que vayan a acceder o modificar. Por esto mismo, si algún usuario quiere acceder dentro del sistema a un área sin contar con los permisos para esto, su solicitud será rechazada. Esto garantizará una mayor robustez. Se detallarán las distintas capas y su función a fin de que se comprenda el porque de esta determinación.

Capa de las configuraciones del sistema

Esta capa será de acceso exclusivo para el administrador del sistema, donde se encontrará todo lo relacionado a la configuración del sistema: configuración del dispositivo de arranque en la BIOS, creación o no de particiones en los discos duro, configuraciones a nivel general, actualizaciones, instalación y configuración de aplicaciones, etc. El usuario con permiso de administrador es el único que tiene acceso total al sistema y es capaz de manipular cualquier registro.

Capa de Red

En caso de que por alguna eventualidad se desee acceder de forma remota al sistema de Arregla Todo, deberá de ser por medio de una VPN que utilice la red de la empresa. Esto para garantizar de que si se hace algún tipo de transferencia de datos desde el servidor a una maquina externa o viceversa, sea por medio de la red empresarial. Debido a que la red interna que utiliza Arregla Todo se encuentra asegurizada a un mayor nivel que una red doméstica común que cualquiera de los empleados pueda poseer.

Capa de Base de Datos

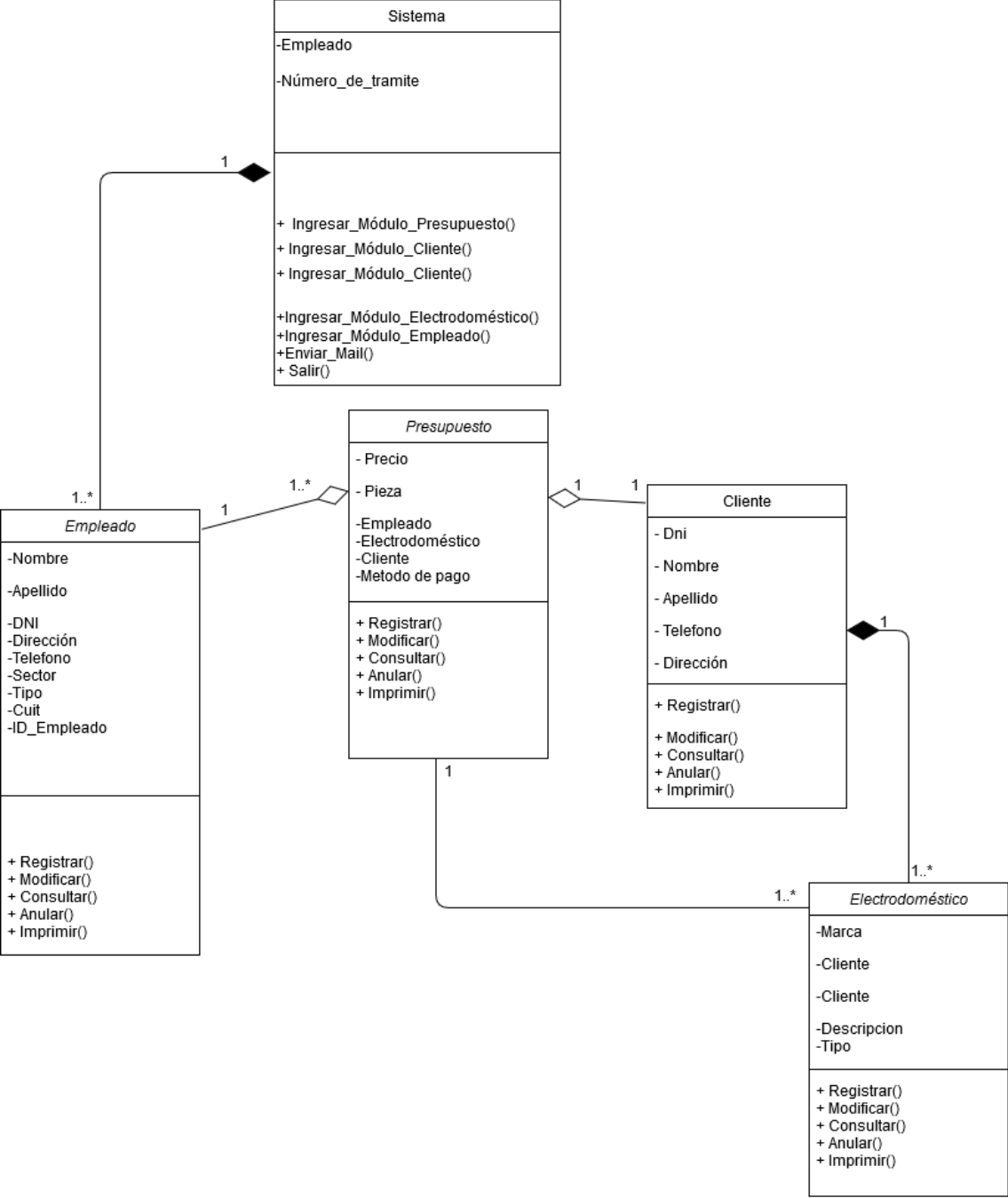
Esta capa contará con dos tipos de usuarios. Por un lado el administrador, el cual será capaz de crear tablas, borrar datos y cualquier otra acción que altere el diseño de la base. En contraparte, los usuarios de tipo empleado solo podrán realizar cargas, actualizar y consultar información de la base de datos. Esto es así porque si bien los empleados manipulan datos, no es conveniente darle capacidad de alterar la lógica de la base de datos.

Capa del Sistema Operativo

Los empleados utilizarán las distintas aplicaciones para su trabajo, pero tendrán permisos limitados. No podrán instalar, descargar ni modificar nada relacionado a las aplicaciones y herramientas usadas para su trabajo. Asimismo, la navegación a ciertos sitios webs se verá limitada con el fin de preservar la seguridad e integridad de la

información de la empresa ante cualquier tipo de malware de una página no segura. En resumen, sus permisos para manipular el sistema operativo serán limitados a lo que resulte conveniente a nivel seguridad para la empresa.

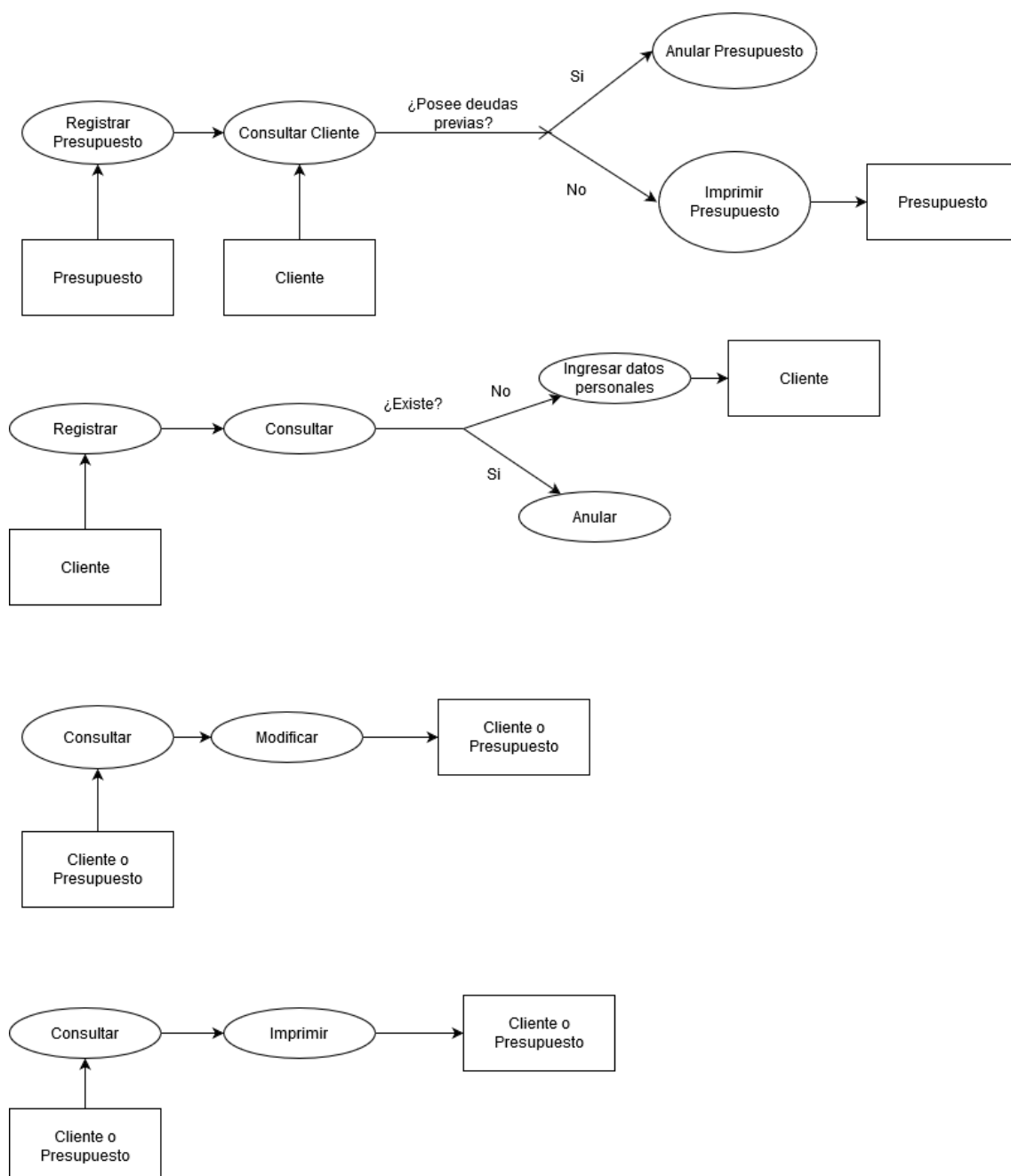
7. Desarrollo un Diseño utilizando la Descomposición Orientada a Objetos. (Desarrolle un Diagrama de Clases completo con atributos y métodos).



Aclaración: Cuando se ingresa a cualquiera de los módulos existirá un menú interactivo en el cual se podrá realizar las acciones pertinentes (registrar, consultar, etc.) tanto del lado del presupuesto como el cliente. Se menciona esto aquí porque solo con la descomposición orientada a objetos no llega a ser claro del todo.

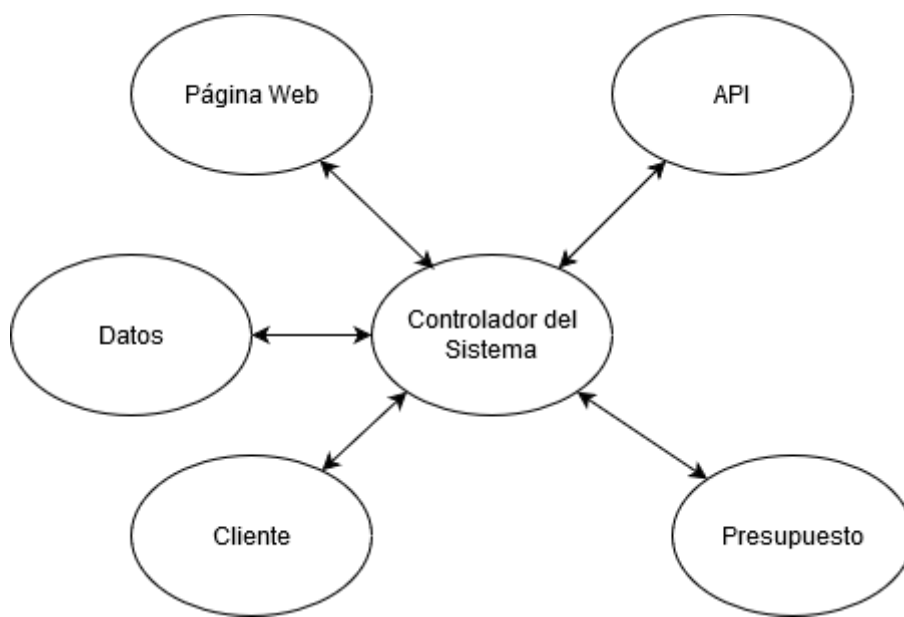
8. Aplique el modelo de descomposición Orientada a Flujo de funciones.

Por fines prácticos y de espacio aquellos flujos de funciones similares entre Cliente o Presupuesto se pusieron en una sola tabla, ya que, no hay ninguna diferencia en como sufren las transformaciones en sus respectivas clases gracias a los métodos.



9. Aplique el Modelo de Gestor.

Con el fin de tener un mejor control del sistema se implementará un modelo de gestor que actuará de controlador central para gestionar la ejecución de los procesos pertenecientes al sistema. Este controlador ayudará a decidir cuando empezar o terminar los procesos dentro del programa. A su vez, comprueba que procesos produjeron información para ser procesada o para enviarles información a otro proceso y que este la procese.



10. Desarrolle un Plan de Pruebas y Testing detallados.

Para probar el sistema se realizarán en primera instancia las siguientes pruebas:

- **Prueba de integración:** Esta se basará en según los errores que surjan en el sistema se modificará o agregará código al código fuente del sistema de gestión de Arregla Todo con el fin de solventar el problema. Para la continua integración de componentes al sistema se utilizará la continua, es decir, se añadirán componentes de infraestructura y componentes funcionales. A su vez, se generarán pequeños datos de ejemplos para ir comprobando que las nuevas integraciones al sistema van respondiendo como se espera. Por esto, se harán testeos continuos durante la

integración y se partirá de un sistema pequeño, a la vez que sencillo, hasta tener todos los módulos implementados y funcionales.

- Pruebas de entregas: Del lado del usuario final del software se testeará si el sistema cumple con sus expectativas y si es confiable. Si se acepta la entrega actual del sistema, el cliente podrá aceptarla para su uso, ya que, es el software que se le pretende entregar. Para que esto ocurra debe demostrarse que cumpla con la funcionalidad esperada, rendimiento y confiabilidad, además de no fallar durante su uso normal.
- Prueba de rendimiento: Luego que el se haya integrado por completo el sistema será momento de realizarle pruebas con el fin de saber si opera correctamente con la carga de datos esperada. Si este test es pasado correctamente se procede a sobrecargar el sistema realizando un test de estrés para intentar descubrir si al sobrepasar los límites de este existe algún tipo de peligro en cuanto a la corrupción de datos o pérdida de servicios del lado del usuario. También ayuda a descubrir defectos que bajo circunstancias de uso normal no aparecen. Evaluando la gravedad que genera en el sistema de Arregla Todo la sobrecarga de este (en caso de esto ocurra), se optará por realizar las modificaciones pertinentes.
- Prueba de componentes: Se accederán a todas las clases y métodos dentro del sistema de Arregla Todo con el fin de encontrar problemas en los componentes. Se procederá a testear cada objeto dentro del sistema en sus distintas variantes, es decir, accediendo a los métodos dentro del mismo, instanciando sus atributos y obteniéndolos, su correcta función al utilizarse en conjunto con los demás objetos y simular sus posibles cambios de estado durante la ejecución del sistema.
- Prueba de interfaces: Una vez testeados los componentes hay que realizar lo mismo en aquellos objetos que trabajen de forma conjunta como su funcionamiento (ejemplo de esto es, presupuesto, piezas y cliente). Es igualmente de importante realizar esta prueba tanto como la de componentes, ya que, a diferencia de la anterior se podrá apreciar si existen errores al interaccionar entre las diferentes partes y posteriormente en caso de existir serán solucionados por el equipo de desarrollo.
- Desarrollo dirigido de Pruebas:

1. Se comienza por identificar el incremento de funcionalidad requerido.
 2. Éste usualmente debe ser pequeño y aplicable en pocas líneas del código.
 3. Se escribe una prueba para esta funcionalidad y se implementa como una prueba automatizada.
 4. Esto significa que la prueba puede ejecutarse y reportarse, sin importar si aprueba o falla.
 5. Luego se corre la prueba, junto con todas las otras pruebas que se implementaron.
 6. Luego se implementa la funcionalidad y se opera nuevamente la prueba.
 7. Una vez puestas en funcionamiento con éxito todas las pruebas, se avanza a la implementación de la siguiente funcionalidad.
- Pruebas de Usuario:
 1. Definir criterios de aceptación.
 2. Establecer Plan de Prueba.
 3. Ejecutar las pruebas de aceptación
 4. Ver resultados de las pruebas de aceptación.
 5. Realizar reportes de la prueba de aceptación

Testing

Se realizarán pruebas en el software de Arregla Todo con el fin de obtener pruebas de acerca de su calidad. Los objetivos de este proceso son:

- Encontrar defectos o bugs para corregir.
- Corregir estos errores aumenta la confianza en cuanto a la calidad del software que se le entregará a la empresa.
- Evita que aparezcan nuevos defectos en función de bugs anteriores si no se realiza las pertinentes pruebas.

Se realizarán dos tipos de pruebas:

- Estáticas: Son aquellas realizadas sin necesidad de ejecutar el código. Mediante la revisión estática del código fuente se buscarán defectos.
- Dinámica: Ejecutar el sistema y realizar pruebas.

Debido a que no es un sistema de gran tamaño, las pruebas serán hechas de forma manual. Se comprobará que todo dentro de la aplicación funcione, ya sean: menús, botones, etc.

Habr  pruebas funcionales, tales como: unitarias, de integraci n, aceptaci n y regresi n. Las primeras tres fueron especificadas en cuanto a su metodolog a en la secci n de Plan de Pruebas. Mientras que la prueba de regresi n existir  en caso de que se considere  adir (luego de terminado alg n sprint) una funcionalidad que no estaba prevista y constar  que al agregar esta funci n nueva, no afecte el correcto funcionamiento de lo anterior.

Tambi n realizaremos pruebas no funcionales con el fin de asegurar la calidad del software que se entregar . Las pruebas ser n las siguientes:

- De seguridad: Mediante hacking  tico encontrar cualquier tipo de vulnerabilidad producto (puertos que utilice el software, acceso a la informaci n, etc.)
- De usabilidad: Que tan intuitiva resulta para el usuario la aplicaci n.
- De accesibilidad: Correcta visualizaci n de los elementos tanto en la p gina web como en el sistema.

Fallas en el Desarrollo

Estas ser n algunas de las fallas en el desarrollo del c digo fuente que se intentarn encontrar para su posterior correcci n:

- Fallas de datos: Variables usadas antes de inicializaci n. Variables declaradas, pero nunca usadas. Variables asignadas dos veces, pero nunca usadas entre asignaciones. Posibles violaciones de l mites de arreglo. Variables no declaradas.
- Fallas de control: C digo inalcanzable. Ramas incondicionales en ciclos.
- Fallas entrada/salida: Variables de salida dobles sin intervenci n de asignaci n. Comprende utilizar variables que no cumplen ninguna funci n.
- Fallas de interfaz:

Incompatibilidad del tipo de par metro.

Incompatibilidad del n mero de par metro.

No se usan resultados de funciones.

Funciones y procedimientos no llamados.

Para complementar el testing también se probará cada una de las posibles variantes que podrían generarse durante la ejecución del sistema. A continuación, se detallarán algunas:

- Registrar un cliente o presupuesto e intentar volver a registrarlo con los mismos datos.
- Anular un presupuesto o cliente que no existe.
- Asignar el mismo especialista a múltiples tareas.
- Control de excepciones al ingresar un dato nulo o mal.
- Utilizar en cualquier orden los métodos.
- Utilizar los métodos sin tener la información necesaria.
- Intentar cargar datos erróneos (un número en vez de nombre, etc.)

Habrà testing también en la página web. Estas son algunas de las pruebas que se realizarán:

- Verificar que el usuario haya iniciado sesión pueda realizar un pago.
- Comprobar si se puede registrar.
- Comprobar el funcionamiento de los elementos desplegables y botones de la página.
- Comprobar funcionamiento del método de login.
- Verificar que el usuario que no haya iniciado sesión no pueda realizar un pago.
- Poder acceder a todas las secciones y subsecciones.
- Verificar que los clics funcionen.

Por último, hay que aclarar que ante cualquier tipo de bug o falla en el código a la hora de ejecutar el software, se documentará con el fin de tener una constancia de el cambio de versionado del sistema. A continuación, se detallará mediante un esquema como serán las etapas del proceso de testing:

Ciclo de Pruebas de Calidad de Software



11 Desarrolle un esquema de integración teniendo en cuenta su Propuesta y explique el esquema detalladamente.

Primero se comenzará por la implementación de Cliente y Empleado, estas clases pueden ser testeadas de modo aislado sin tener ninguna otra implementada debido a que sus métodos no dependen de variables o cambios hechos sobre algún objeto en otra parte del sistema. Es decir, es una integración mínima del sistema y se probará la correcta función de sus métodos.

Se agregará la clase Electrodoméstico al sistema, la cual cuenta con métodos similares a los anteriores, pero uno de sus atributos depende de la clase cliente, ya que, el electrodoméstico tiene un cliente asociado.

En caso de que no haya ningún fallo entre la interfaz de Cliente y Electrodoméstico, la integración continuará en Presupuesto, donde se probarán sus métodos y comprobarán la correcta función de la interfaz Empleado-Cliente-Electrodoméstico-Presupuesto, ya que, Presupuesto depende de las tres clases anteriormente implementadas. Por último Sistema,

que tendrá los métodos para acceder a las distintas clases del sistema (mediante un menú) y realizar otro tipo de acciones asociadas al negocio.

Este esquema de integración permite hacer énfasis en desarrollar una parte del software e inmediatamente testear su funcionamiento antes de continuar con la siguiente integración. Permitiendo así, detectar de forma mas sencilla y aislada que podría estar generando fallos en el sistema luego de agregar una funcionalidad nueva. En cada caso se buscará comprobar el correcto funcionamiento de la clase con sus métodos, ya que, esto también nos permitirá saber si es correcta la carga, modificación u eliminación de atributos y objetos.

El equipo de desarrollo trabajará con un repositorio virtual (Git Hub) y luego de pasada la fase de testing correctamente, se hará un commit al repositorio en una rama de común acceso al equipo denominada “Dev”. Cada nueva integración seguirá este proceso y una vez que todo el sistema este integrado y pasadas las distintas etapas de pruebas, se subirá la versión 1.0 del software a la rama “Master” para ya pasarlo a producción. Esta será la versión final para uso de Arregla Todo.

