



UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



TESIS

**“OPTIMIZACIÓN CON PATRONES DE DISEÑO DE SOFTWARE PARA
LA SISTEMATIZACIÓN DEL TRÁMITE DE DOCUMENTOS EN
SERVICIOS GENERALES BARBOSA”**

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMÁTICA

PRESENTADO POR LAS BACH.:

IRIS ESTHER NAUTINO SALAS.

NELLY ELENA PÉREZ CÁRDENAS.

ASESORA:

ING. GRECIA MILAGROS BARRERA ORTIZ

IQUITOS – PERÚ

2019



UNIVERSIDAD NACIONAL DE LA AMAZONIA PERUANA
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA

ACTA DE SUSTENTACION DE TESIS

Siendo las 13:00 horas del día 13 de MARZO del 2019, en la Instalación del Auditorio de esta Facultad, se ha constituido el jurado examinador integrado por los siguientes miembros:

Presidente : Dr. Ángel Enrique López Rojas
Primer Miembro : Ing. Alejandro Reátegui Pezo
Segundo Miembro : Ing. José Edgar García Díaz

Acto seguido, se recibió al Tesisista **IRIS ESTHER NAUTINO SALAS**, quien sustentó el Proyecto de Tesis: **"OPTIMIZACION CON PATRONES DE DISEÑO DE SOFTWARE PARA LA SISTEMATIZACION DEL TRAMITE DE DOCUMENTOS EN SERVICIOS GENERALES BARBOZA"**, para optar el Título Profesional de Ingeniero de Sistema e Informática, de acuerdo a lo establecido en el Reglamento de Grados y Títulos de la Universidad Nacional de la Amazonia Peruana.

Posteriormente, al Acto de sustentación de la Tesis, se procedió al cálculo de Calificación obteniéndose el siguiente resultado:


	Calificaciones	
	En número	En letras
La Ejecución de la Tesis	18	DIECIOCHO
La Sustentación de la Tesis.	56	DIECISEIS
Promedio Final	57	DIECISIETE


De la evaluación se desprende que el resultado es:


APROBADO

Con las siguientes observaciones:

Se da por concluido el acto de sustentación a las 13:50 horas, procediéndose a firmar el Acta.


Dr. Ángel Enrique López Rojas
Presidente


Ing. Alejandro Reátegui Pezo
Primer Miembro


Ing. José Edgar García Díaz
Segundo Miembro



UNIVERSIDAD NACIONAL DE LA AMAZONIA PERUANA
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA

ACTA DE SUSTENTACION DE TESIS

Siendo las 13 horas del día 13 de MARZO del 2019, en la Instalación del Auditorio de esta Facultad, se ha constituido el jurado examinador integrado por los siguientes miembros:

- Presidente : Dr. Ángel Enrique López Rojas
Primer Miembro : Ing. Alejandro Reátegui Pezo
Segundo Miembro : Ing. José Edgar García Díaz

Acto seguido, se recibió al Tesista **NELLY ELENA PÉREZ CARDENAS**, quien sustentó el Proyecto de Tesis: **“OPTIMIZACION CON PATRONES DE DISEÑO DE SOFTWARE PARA LA SISTEMATIZACION DEL TRAMITE DE DOCUMENTOS EN SERVICIOS GENERALES BARBOZA”**, para optar el Título Profesional de Ingeniero de Sistema e Informática, de acuerdo a lo establecido en el Reglamento de Grados y Títulos de la Universidad Nacional de la Amazonia Peruana.

Posteriormente, al Acto de sustentación de la Tesis, se procedió al cálculo de Calificación obteniéndose el siguiente resultado:

	Calificaciones	
	En número	En letras
La Ejecución de la Tesis	<u>18</u>	<u>Dieciocho</u>
La Sustentación de la Tesis.	<u>16</u>	<u>Dieciséis</u>
Promedio Final	<u>17</u>	<u>Diecisiete</u>

De la evaluación se desprende que el resultado es:

APROBADO

Con las siguientes observaciones:

Se da por concluido el acto de sustentación a las 13:50 horas, procediéndose a firmar el Acta.


Dr. Ángel Enrique López Rojas
Presidente


Ing. Alejandro Reátegui Pezo
Primer Miembro


Ing. José Edgar García Díaz
Segundo Miembro

**“OPTIMIZACION CON PATRONES DE DISEÑO DE SOFTWARE PARA
LA SISTEMATIZACION DEL TRÁMITE DE DOCUMENTOS EN
SERVICIOS GENERALES BARBOSA”**

PRESENTADO POR LOS BACH.:
IRIS ESTHER NAUTINO SALAS.
NELLY ELENA PÉREZ CÁRDENAS.



ING. GRECIA MILAGROS BARRERA ORTIZ

ASESORA

DEDICATORIA

A mis padres que me fortalecen todos los días con su amor incondicional, su comprensión, su valor para luchar por mis sueños y que me inspiran para ser mejor cada día.

A nuestros docentes por compartir su conocimiento con nosotras para ser competitivas en mundo cada vez más cambiante y que nos inculcan a seguir .progresando y estudiando siempre.

Bach: Iris Esther Nautino Salas.

Esta tesis está dedicado en primer lugar, a Dios por permitirme cumplir mis objetivos y mis metas; a mi adorada mamá Leila Cárdenas Tananta por su apoyo incondicional, a mi papito Ramón Pérez Ríos, que aunque ya no este entre nosotros sigue vivo en mis pensamientos y mi corazón; y que su estímulo y sus buenos consejos fueron mi impulso para llegar al final, a mi hermano, a mi hermoso hijo Criss Uriel por su amor, paciencia y comprensión, por ser mi motor y motivo para seguir adelante profesionalmente y ser mejor persona.

Bach: Nelly Elena Pérez Cárdenas.

AGRADECIMIENTO

“A Dios, por derramar sobre nuestros corazones y sobre nuestra familia la paz, el amor, y la salud”

A nuestro asesora Ing. Grecia Milagros Barrera Ortiz, y Asesor Interno ING. Carlos González Aspajo Mtr. Por sus infinita paciencia y fuerzas en el desarrollo de este proyecto para poder realizar esta tesis.

ÍNDICE

CONTENIDO	Página
PORTADA	i
ACTAS	ii
ACTAS	iii
ASESORA	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE	1
INDICE DE FIGURAS	4
INDICE DE TABLAS	5
RESUMEN	6
ABSTRACT	7
INTRODUCCION	8
CAPITULO I: MARCO TEORICO	9
1.1. ANTECEDENTES	9
1.2. BASES TEÓRICAS	13
1.2.1. PATRÓN DE DISEÑO	13
1.2.2. PATRONES CREACIONALES	14
1.2.3. PATRONES ESTRUCTURALES	14
1.2.4. PATRONES DE COMPORTAMIENTO.....	15
1.2.5. RATIONAL UNIFIED PROCESS (RUP)	17
1.2.6. VISUAL ESTUDIO COMMUNITY 2015.....	18
1.2.7. MYSQL.....	19
1.2.8. SQL	20
1.2.9. RATIONAL SOFTWARE ARCHITEC 8.01.....	20
1.2.10. MICROSOFT OFFICE WORD	20
1.2.11. SIGNALR	20
1.3. DEFINICIÓN DE TÉRMINOS BÁSICOS	22
CAPITULO II: HIPOTESIS Y VARIABLES	24
2.1 FORMULACION DE HIPOTESIS	24
2.2. VARIABLES Y OPERACIONALIDAD	24
2.2.1. Variable independiente:.....	24
2.2.2. Variable dependiente:.....	24

CAPITULO III: METODOLOGIA	26
3.1. TIPO DE INVESTIGACION	26
3.3. DISEÑO MUESTRAL	26
3.2.2. POBLACION	26
3.2.2. MUESTRA	26
3.3. PROCEDIMIENTOS DE RECOLECCION DE DATOS	26
3.3.1. Técnica de recolección de datos	26
3.3.2. Instrumento de recolección de datos	26
3.4. PROCESAMIENTO Y ANALISIS DE DATOS	27
3.5. DISEÑO DE LA SOLUCIÓN	27
3.5.1. REQUERIMIENTOS	28
3.5.2. MODELO DE CASOS DE USO	29
3.5.3. APLICACIÓN DE PATRONES	30
3.5.3.1. SINGLETON:	30
3.5.3.2. FACTORY METHOD:	34
3.5.3.3. FACTORY ABSTRAC:	37
3.5.3.4. PROXY:	39
3.5.3.5. MVC:	42
3.6. ANÁLISIS	43
3.7. VISIÓN GENERAL	47
3.8. MODELO DE NEGOCIO	48
3.8.1. REALIZACION DE LOS CUN	52
3.8.2. CUN REGISTRAR DOCUMENTO EXTERNO	54
3.8.3. CUN CONSULTAR DOCUMENTO EXTERNO	56
3.8.4. CUN CONSULTAR DOCUMENTO INTERNO	58
3.8.5. DIAGRAMA E-R (MODELADO DE DATOS)	60
CAPITULO IV: RESULTADOS	61
CAPITULO V: DISCUSIONES	69
CAPITULO VI: CONCLUSIONES	72
CAPITULO VII: RECOMENDACIONES	73
CAPITULO VIII: FUENTES DE INFORMACIÓN	74
ANEXOS	76
ANEXO 1: MATRIZ DE CONSISTENCIA	77
ANEXO 2: PROTOTIPOS NO FUNCIONALES	78
ANEXO 3: DICCIONARIO DE DATOS	83

ANEXO 4: CODIGO DE PROGRAMA	88
ANEXO 5: MANUAL DE USUARIO	117

INDICE DE FIGURAS

Figura 01: fases y disciplinas del RUP.....	18
Figura 02: Versiones de MySQL.....	19
Figura 02: Diagrama UML de una clase que implementa el patrón singleton.	30
Figura 03: Implementación del Patrón Singleton.	31
Figura 04: Formulario Área.....	31
Figura 05: Implementación del Patrón Singleton en la creación de un nuevo formulario	32
7Figura 06: Lista de selección para asignar a una nueva área.	32
Figura 07: Uso del Patrón SINGLETON.	33
Figura 08: Lista de clases de la lógica de Negocios	34
Figura 09: Una de las listas de Lógica de Negocios.	34
Figura 10: Código Fuente de la Implementación del Patrón Factory en la Lógica de Negocios.	35
Figura 11: Uso del Patrón Factory.	36
Figura 12: Uso del Patrón Factory.	36
Figura 13: Diagrama UML del Patrón Factory Abstrac	37
Figura 14: Diagrama UML del Patrón Factory Abstrac.	38
Figura 15: Diagrama de UML del Patrón Proxi.	40
Figura 16: Diagrama de UML del Patrón Proxi.	41
Figura 17: Capas con Patrón.	42
Figura 18: Diagrama De Caso De Uso De Negocio.	48
Figura 19: Diagrama de trabajadores de Negocio.	49
Figura 20: Diagrama de Entidades de Negocio.	49
Figura 21: Diagrama de Actividades detallado para registrar documento interno.	52
Figura 22: Diagrama de Objeto para consultar documento inteno.....	53
Figura 23: Diagrama de Actividad Detallado para registrar documentos externo.	54
Figura 24: Diagrama de Objeto para registrar documentos externos.....	55
Figura 25: Diagrama de Actividad Detallado para consultar documentos externos.....	56
Figura 26: Diagrama de Objeto para consultar documentos externos.....	57
Figura 27: Diagrama de Actividad Detallado para consultar documentos internos.	58
Figura 28: Diagrama de Objeto para consultar documentos interno.....	59
Figura 29: Diagrama de modelado de datos. Fuente.....	60

INDICE DE TABLAS

Tabla 1: Características de los patrones	14
Tabla 2: Descripción de variables	24
Tabla 3: Descripción de versiones.....	43
Tabla 4: Cuadro de Resumen de las Versiones del Sistema.	44
Tabla 5: Cuadro de Requerimientos.....	45
Tabla 6: Cuadro de Usuarios del Sistema.....	47
Tabla 7: Cuadro de Descripción para Registrar documento Interno. Fuente: Elaboración Propia.....	50
Tabla 8: Cuadro de Descripción para Registrar documento Externo. Fuente: Elaboración Propia.....	50
Tabla 9: Cuadro de Descripción para Consultar Documento Externo. Fuente: Elaboración Propia.....	51
Tabla 10: Cuadro de Descripción para Consultar Documento Interno. Fuente: Elaboración Propia.....	51

RESUMEN

La investigación tuvo como objetivo la Optimización del proceso de desarrollo de software con la utilización de Patrones de Diseño, para la Sistematización del Trámite de Documentos en Servicios Generales Barbosa. La metodología de investigación se caracteriza por ser de tipo aplicada y el diseño de investigación se ubica en el diseño pre-experimental de comparación estática, con pos prueba únicamente y grupos intactos porque no son asignados al azar, ni emparejados, se intenta solucionar los problemas detectados, sentando las bases para la obligación inmediata de los resultados obtenidos y modificar así la situación descrita.

Se desarrollaron dos aplicativos para la comparación, uno con patrones de diseño y el otro sin patrones de diseño, se plantearon cinco incrementos evolutivos del aplicativo de software con once requerimientos de desarrollo.

Se recopila un conjunto de conocimientos teóricos-conceptuales y metodológicos acerca de la gestión documental. Se determina a través de la metodología Patrón de diseño, obteniendo las fortalezas y debilidades que ayudaron a conocer científicamente, las necesidades de la institución en este elemento.

PALABRAS CLAVES: Patrones de diseño, desarrollo de software.

ABSTRACT

The objective of the research was Optimization of the software development process with the use of Design Patterns, for the Systematization of the Documents Processing in General Services Barbosa. The research methodology is characterized by being of applied type and the research design is located in the pre-experimental design of static comparison, with only post-test and intact groups because they are not randomized, nor paired, they try to solve the problems detected, laying the groundwork for the immediate obligation of the results obtained and thus modify the situation described.

Two applications were developed for comparison, one with design patterns and the other without design patterns, five evolutionary increments of the software application with eleven development requirements were considered.

A set of theoretical-conceptual and methodological knowledge about document management is collected. It is determined through the methodology Design pattern, obtaining the strengths and weaknesses that helped to know scientifically, the needs of the institution in this element.

KEYWORDS: Design patterns, software development.

INTRODUCCION

El presente proyecto de tesis propuso la evaluación del proceso de desarrollo de software con patrones de diseño y sin patrones de diseño y toma como escenario los requerimientos operativos de un software de trámite documentario en la Empresa Servicios Generales Barbosa, para mejorar los tiempos en la búsqueda de documentos. Por ello este documento de investigación sustenta el camino para llegar a la solución; iniciando dicha propuesta desde la identificación de la problemática, luego objetivos así hasta llegar a los indicadores, conclusiones y recomendaciones, en la cual se podrá demostrar que el requerimiento y expectativas de los usuarios se puede lograr en un tiempo menor a sus procedimientos actuales.

En presente proyecto presenta como tema de investigación la optimización con patrones de diseño para el desarrollo de aplicaciones de software por lo que se analizaran los patrones de diseño: Singleton, Factory, Factory Abstrac, Proxy y Mvc.

La investigación comprende los siguientes capítulos:

Capítulo I: Planteamiento del problema

Capítulo II: Marco Teórico que abarca: antecedentes del estudio, Planteamientos teóricos; Justificación.

Capítulo III: Metodología

Capitulo IV: Resultado y discusión.

Capítulo V: Conclusiones

Capítulo VI: Recomendaciones

Capitulo VII: Bibliografía

Anexos

CAPITULO I: MARCO TEORICO

1.1. ANTECEDENTES

- **Ponce Romero, Jhadira Marilú; Valderrama Bacilio, Larry Billy; 2013. (Ponce Romero, y otros, 2013) Tesis Desarrollo De Un Sistema Basado En Tecnología Web Usando El Patrón De Diseño Modelo Vista Controlador Para Mejorar La Actualización Del Inventario De Equipos Dslam De Telefónica Del Perú; Universidad Nacional de Trujillo; Trujillo.** Resumen: En el presente trabajo, proponemos el desarrollo de un sistema con tecnología web para mejorar la actualización del inventario de equipos dslam de Telefónica del Perú, basado patrones de diseño modelo vista controlador enfocado principalmente para realizar actualizaciones de los datos en tiempo real, de tal manera que se obtenga mejoras en los resultados al realizar cualquier tipo de actualización en el inventario. También se ha tomado como estudio los procesos del desarrollo del software que constan en: Análisis, diseño, implementación y pruebas la cual nos permitirá mejorar la actualización del inventario realizando cualquier tipo de consultas como agregar, eliminar y consultar la información en la base de datos de nuestro inventario centrándonos en la etapa del diseño utilizando la metodológica RUP el cual es un proceso de desarrollo utilizando como herramienta el lenguaje unificado de modelado(UML) para obtener los resultados propuestos.
- **Edson Bryan Castañeda Rojas, 2016 (Rojas, 2016). Tesis: “Propuesta De Patrón De Diseño De Software Orientado A Prevenir La Extracción Automatizada De Contenido Web”; Universidad Pontificia Universidad Católica Del Perú; Lima.** Resumen: Web scraping o extracción de datos Web es el proceso de recolección de información de uno o más sitios Web de manera automatizada, emulando la interacción entre un usuario y un servidor, dicho proceso se basa en el análisis de estructuras HTML y no requiere la autorización de los propietarios. El uso de estructuras repetitivas o plantillas, facilita el funcionamiento de un

programa informático que extrae contenido Web, dicha intrusión genera un incremento considerable en el uso de recursos, considerando la permanente ejecución de instrucciones para obtener tanto contenido como sea posible. Con la finalidad de reducir la vulnerabilidad de los sitios Web frente a procesos de extracción de contenido masivo, en el presente trabajo se planteó un patrón de diseño de software tomando como referencia el patrón Template View de Martin Fowler, ;al cual se agregó una capa de aleatorización que permita generar estructuras HTML no predecibles. Mediante la aplicación de una herramienta de extracción de contenido a un sitio Web de prueba, cuya capa de presentación se desarrolló tomando en cuenta el patrón de diseño propuesto, se logró verificar una reducción considerable de la cantidad de datos extraídos.

- **Maricruz de Lourdes Acosta Yerovi, 2013 (Yerovi, 2013). Tesis: “Estudio De Patrones De Diseño En Plataforma Java Enterprise Edition Versión 6 Para El Desarrollo De Aplicaciones Web”;** **Universidad Técnica Del Norte; Ecuador.** Resumen: En la búsqueda constante por mejorar la calidad y el diseño del software, técnicas y tecnologías han evolucionado. Se han desarrollado diversas estrategias y habilidades, varias son producto de mejoras realizadas sobre estrategias anteriores. Los patrones de diseño probablemente son las estrategias más versátiles, consisten en la reutilización de soluciones, no solamente de código fuente, sino que constituyen soluciones integrales a problemas repetitivos en el desarrollo del software. Es importante señalar que los patrones de diseño JEE, recopilan un conjunto de buenas prácticas que se han venido desarrollando en los últimos años para el desarrollo de sistemas web. La presencia de patrones de diseño conduce a soluciones estándares, fácilmente comprensibles y mantenibles por parte de los desarrolladores. El enfoque principal de este trabajo es el conocimiento de los patrones de diseño JEE 6, ya que a pesar de los beneficios que su implementación genera, son tratados a menudo como patrones J2EE, obstruyendo la funcionalidad de los patrones de diseño y de la plataforma en sí.

- **Cristian Giovanni Martínez Rodríguez, 2012; (RODRÍGUEZ, 2012).**
Tesis: “Aplicación De Patrones De Diseño En El Diseño De Arquitecturas Orientadas A Servicios”; **Universidad De San Carlos De Guatemala; Guatemala;** Resumen: La evolución de la tecnología y la incorporación de nuevas tecnologías dan a las industrias nuevas posibilidades, a su vez generan nuevas necesidades, una de estas necesidades es la de abandonar el esquema de operación en la que el software funciona aislado. La necesidad es entonces de plantear un nuevo escenario en donde el software conoce su entorno, se integra a él y responde a necesidades tanto de un usuario como de otro componente de software, sin embargo, esto presenta el inconveniente de software desarrollado con distintas tecnologías y lograr la interacción de diferentes tecnologías con el menor costo y esfuerzo posible. Es entonces necesario conocer la alternativa de la utilización de una arquitectura orientada a servicios para facilitar la interacción entre los distintos sistemas que forman el entorno empresarial, que puede estar compuesto por sistemas desarrollados con diferentes tecnologías y épocas; y conseguir aún la integración de los mismos en un entorno que permita formar un sistema mayor. Por otro lado, el desarrollo de soluciones a problemas recurrentes ha permitido la definición de patrones, mismos que pueden ser aplicados con el fin de aplicar a un problema conocido una solución de antemano conocida y validada para facilitar así el desarrollo de soluciones efectivas en menor tiempo y con menor esfuerzo. Resulta entonces posible y útil combinar los patrones de diseño y la arquitectura orientada a servicios.
- **Ing. Darío Yorio, 2006 (Yorio, 2006).** Tesis: **“Identificación Y Clasificación De Patrones En El Diseño De Aplicaciones Móviles”.** **Universidad Nacional de La Plata, Argentina.** Resumen: A lo largo de la presente Tesis qse presentó un catálogo de patrones para aplicaciones móviles, haciendo la salvedad que al momento no se encuentra completo, pero podrá ir creciendo con trabajos futuros. Este catálogo muestra una

posible clasificación basada en los distintos conocimientos propios del dominio, e incluye al menos un patrón de cada una de clases.

1.2. BASES TEÓRICAS

1.2.1. PATRÓN DE DISEÑO

Según la información de algunos autores encontrada en la web (Gama, y otros, 2018). Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Los patrones de diseño se pueden definir como esquemas predefinidos o conjunto de estrategias aplicables en diversas situaciones, de forma que el analista se asegura de que el diseño que está aplicando tiene ciertas cualidades que le proporcionan calidad.

El libro Design Patterns propone la siguiente definición para patrón de diseño: “Los patrones de diseño son descripciones de objetos y clases comunicándose, que son adaptadas para resolver un problema de diseño general en un contexto particular” (Gamma, 1994). Por lo tanto, los patrones se centran en la microestructura de las aplicaciones, es decir, en sus clases y objetos.

Cada patrón de diseño trata de agrupar la experiencia en diseño de tal forma que los desarrolladores puedan utilizarlos con efectividad. Por eso se han documentado los más importantes patrones de diseño y presentado en catálogos.

Para que una solución sea considerada un patrón debe pasar por pruebas y poseer ciertas características. Según (Jim Coplien, 1996) La tabla 1, indica las características que los buenos patrones deben tener, de acuerdo al criterio de Jim Coplien:

Característica	Punto de vista
Solucionar un problema	Los patrones capturan soluciones, no sólo principios o estrategias abstractas.
Ser un concepto probado	Los patrones capturan soluciones demostradas, no teorías o especulaciones.
La solución no es obvia	Los mejores patrones generan una solución a un problema de forma indirecta.
Describe participantes y relaciones entre ellos.	Los patrones no sólo describen módulos sino

	estructuras del sistema y mecanismos más complejos.
El patrón tiene un componente humano significativo	El Software proporciona a los seres humanos confort o calidad de vida (estética y utilidad).

Tabla 1: Características de los patrones

Fuente: (Jim Coplien, 1996)

1.2.2. PATRONES CREACIONALES

Antonio Leiva (Antonio, 2016); indica que los patrones creacionales más conocidos son:

- Abstract Factory: Nos provee una interfaz que delega la creación de un conjunto de objetos relacionados sin necesidad de especificar en ningún momento cuáles son las implementaciones concretas.
- Factory Method: Expone un método de creación, delegando en las subclases la implementación de este método.
- Builder: Separa la creación de un objeto complejo de su estructura, de tal forma que el mismo proceso de construcción nos puede servir para crear representaciones diferentes.
- Singleton: limita a uno el número de instancias posibles de una clase en nuestro programa, y proporciona un acceso global al mismo.
- Prototype: Permite la creación de objetos basados en “plantillas”. Un nuevo objeto se crea a partir de la clonación de otro objeto.

1.2.3. PATRONES ESTRUCTURALES

Son patrones que nos facilitan la modelización de nuestro software especificando la forma en la que unas clases se relacionan con otras.

Estos son los patrones estructurales que definió la Gang of Four:

- **Adapter:** Permite a dos clases con diferentes interfaces trabajar entre ellas, a través de un objeto intermedio con el que se comunican e interactúan.

- **Bridge:** Desacopla una abstracción de su implementación, para que las dos puedan evolucionar de forma independiente.
- **Composite:** Facilita la creación de estructuras de objetos en árbol, donde todos los elementos emplean una misma interfaz. Cada uno de ellos puede a su vez contener un listado de esos objetos, o ser el último de esa rama.
- **Decorator:** Permite añadir funcionalidad extra a un objeto (de forma dinámica o estática) sin modificar el comportamiento del resto de objetos del mismo tipo.
- **Facade:** Una facade (o fachada) es un objeto que crea una interfaz simplificada para tratar con otra parte del código más compleja, de tal forma que simplifica y aísla su uso. Un ejemplo podría ser crear una fachada para tratar con una clase de una librería externa.
- **Flyweight:** Una gran cantidad de objetos comparte un mismo objeto con propiedades comunes con el fin de ahorrar memoria.
- **Proxy:** Es una clase que funciona como interfaz hacia cualquier otra cosa: una conexión a Internet, un archivo en disco o cualquier otro recurso que sea costoso o imposible de duplicar.

1.2.4. PATRONES DE COMPORTAMIENTO

Los patrones de comportamiento gestionan algoritmos, relaciones y responsabilidades entre objetos.

En este último grupo se encuentran la mayoría de los patrones, y se usan para gestionar algoritmos, relaciones y responsabilidades entre objetos.

Los patrones de comportamiento son:

- **Command:** Son objetos que encapsulan una acción y los parámetros que necesitan para ejecutarse.
- **Chain of responsibility:** se evita acoplar al emisor y receptor de una petición dando la posibilidad a varios receptores de consumirlo. Cada receptor tiene la opción de consumir esa petición o pasárselo al siguiente dentro de la cadena.

- **Interpreter:** Define una representación para una gramática, así como el mecanismo para evaluarla. El árbol de sintaxis del lenguaje se suele modelar mediante el patrón Composite.
- **Iterator:** Se utiliza para poder movernos por los elementos de un conjunto de forma secuencial sin necesidad de exponer su implementación específica.
- **Mediator:** Objeto que encapsula cómo otro conjunto de objetos interactúan y se comunican entre sí.
- **Memento:** Este patrón otorga la capacidad de restaurar un objeto a un estado anterior.
- **Observer:** Los objetos son capaces de suscribirse a una serie de eventos que otro objetivo va a emitir, y serán avisados cuando esto ocurra.
- **State:** Permite modificar la forma en que un objeto se comporta en tiempo de ejecución, basándose en su estado interno.
- **Strategy:** Permite la selección del algoritmo que ejecuta cierta acción en tiempo de ejecución.
- **Template Method:** Especifica el esqueleto de un algoritmo, permitiendo a las subclases definir cómo implementan el comportamiento real.
- **Visitor:** Permite separar el algoritmo de la estructura de datos que se utilizará para ejecutarlo. De esta forma se pueden añadir nuevas operaciones a estas estructuras sin necesidad de modificarlas.

1.2.5. RATIONAL UNIFIED PROCESS (RUP)

Jesús Martínez, en su publicación RUP (Martinez, 2012). Indica que el RUP es un proceso de ingeniería de software orientado a objetos. Consiste en un conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de software. Está especializado para diversos tipos de software de sistemas, diversas áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos.

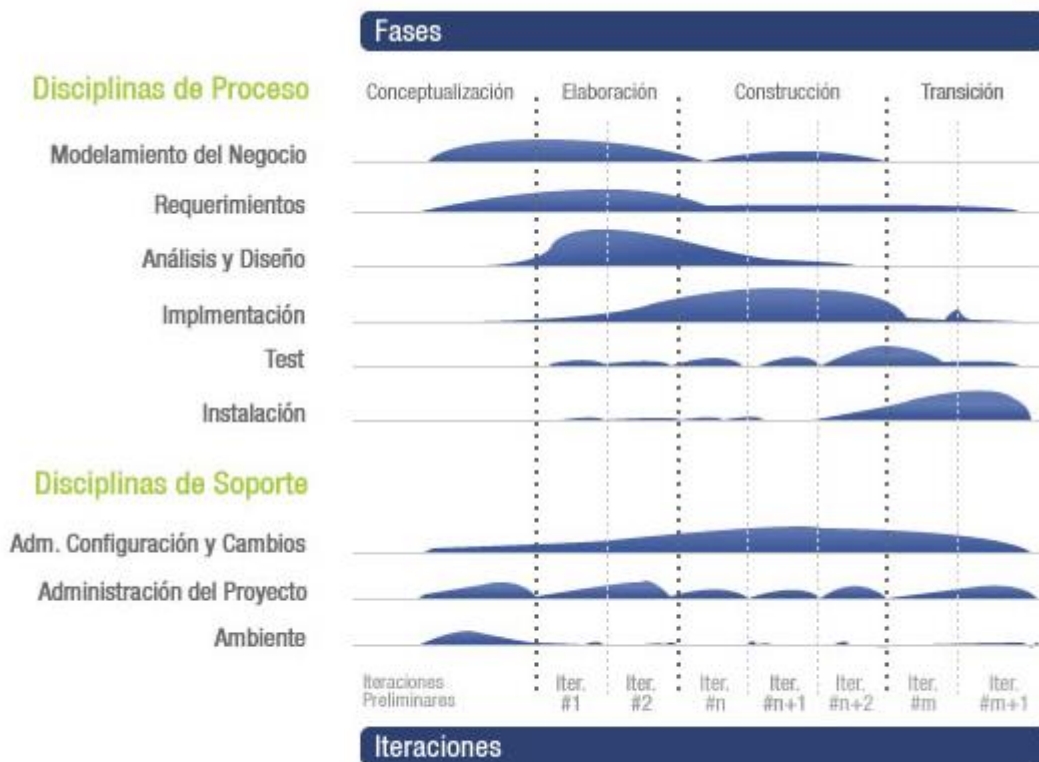
El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

El Proceso Unificado se basa en componentes (component-based), lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas (well-defined interfaces).

El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par.

Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave: dirigido por casos de uso (use-case driven), centrado en la arquitectura (architecture-centric), iterativo e incremental. Esto es lo que hace único al Proceso Unificado que permite comunicar claramente requerimientos, arquitectura y diseños.

RUP se estructura en forma bidimensional, en un eje vertical y en uno horizontal. En el eje vertical tenemos las distintas etapas de las “cascadas” de desarrollo, que se denominan “Core Workflows”, las que son acompañadas de las tareas de apoyo que aseguran el éxito del proyecto denominadas “Workflows de apoyo”. En el eje horizontal tenemos la evolución en el tiempo, que se da en cuatro fases.



El Proceso Unificado de Rational (RUP)

*Gráfico 1: fases y disciplinas del RUP.
Fuente: Internet*

1.2.6. VISUAL ESTUDIO COMMUNITY 2015

Según la información encontrada en el sitio web del Grupo de Usuario Software Libre, 2018; Visual Studio Community es una versión gratuita y completa de Visual Studio, la cual puede descargarse ya, que da posibilidades de un ecosistema extensible y soporte para cualquier plataforma, prácticamente, desde dispositivos hasta las máquinas de escritorio, desde la plataforma web hasta la nube.

El IDE de esta versión de Visual Studio soporta múltiples tipos de proyectos en un solo archivo solución dentro del IDE y tiene todas las características de productividad y extensibilidad del IDE, lo que significa que se puede usar Xamarin, ReSharper, VsVim y otras extensiones VSIX, que los desarrolladores aman de Visual Studio, a decir del propio Microsoft.

1.2.7. MYSQL

Según la información encontrada en el sitio web MySQL (Fundación Wikimedia, 2019) es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

El siguiente gráfico provee un resumen de varias versiones de MySQL y sus estados actuales de desarrollo

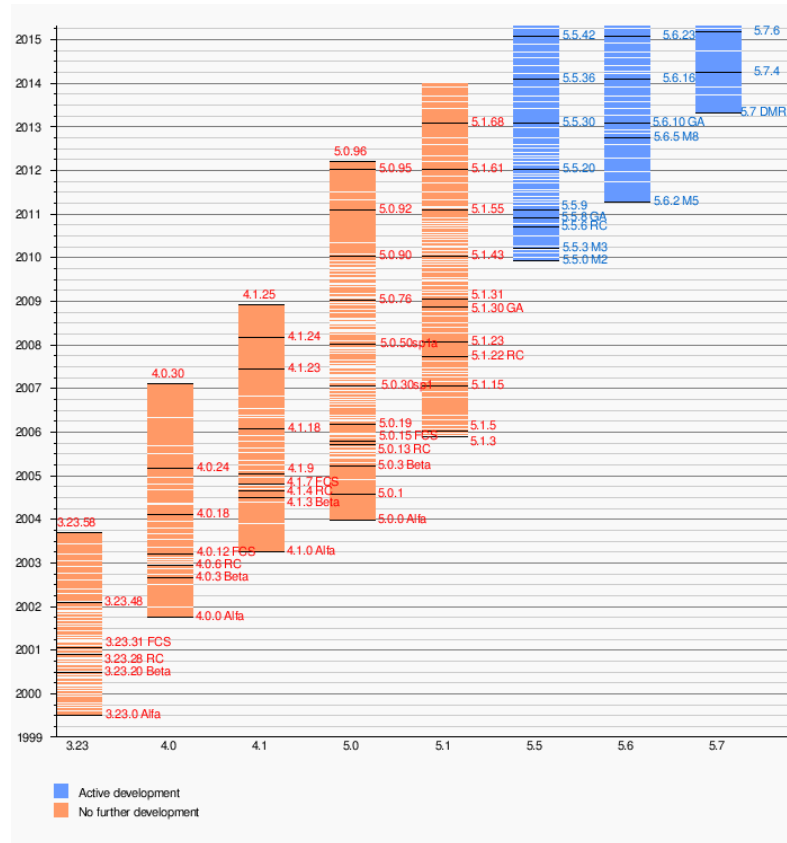


Gráfico 2: Versiones de MySql.
Fuente: Internet

1.2.8. SQL

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

1.2.9. RATIONAL SOFTWARE ARCHITECT 8.01

Joe Martínez en su publicación Software utilizados para modelar UML (Martinez, 2017). Indica que IBM Rational Software Architect Designer (RSAD, y antes denominado RSA) es una completa herramienta de diseño, modelado y desarrollo para la entrega de software de extremo a extremo. Utiliza lenguaje de modelado unificado (UML) para diseñar aplicaciones Java de empresa y servicios web. Rational Software Architect Designer se basa en la infraestructura de software de código abierto Eclipse y se puede ampliar con varios plug-ins de Eclipse. También puede mejorar la funcionalidad adecuándola a sus requisitos específicos con ampliaciones de Rational adquiridas por separado.

1.2.10. MICROSOFT OFFICE WORD

Renán Gonzales en su Proyecto de Investigación (Gonzales, 2015). Indica que Microsoft Office Word es un procesador de textos de Microsoft Office. Ofrece nuevas funciones que facilitan la creación, el uso compartido y la lectura de los documentos. Las funciones de revisión y anotación se han mejorado para solucionar las múltiples formas en que es posible controlar los cambios y administrar los comentarios.

1.2.11. SIGNALR

En la Publicación; Introducción a SignalR en el año 2013 Andrés Iturralde (Iturralde, 2013). Indica que SignalR es una biblioteca desarrollada por Damian Edwards y David Fowler, miembros del equipo de ASP .NET y de hecho el código fuente es Open Source SignalR forma parte de la familia ASP .NET y es útil para crear aplicaciones que respondan en tiempo real.

SignalR provee una capa de abstracción sobre algunos de los transportes requeridos para hacer aplicaciones de tiempo real. Estas aplicaciones empiezan directamente sobre HTTP pero en caso de que se tenga soporte para Web Sockets, dicha conexión será promovida a que use Web Sockets. Todas las conexiones son administradas por SignalR y permite transmitir los cambios a todos los clientes conectados como si fuese una sala de chat o en caso de que sea necesario podemos notificar a un cliente específico.

La conexión entre el cliente y el servidor es persistida, a diferencia de HTTP en donde se envía un request y al momento de recibir el response esta se destruye para posteriormente de cero. La capa de abstracción de SignalR simula esa persistencia a diferencia de las conexiones hechas con Web Sockets las cuales son verdaderamente persistentes.

La conexión entre el cliente y el servidor es persistida, a diferencia de HTTP en donde se hace una llamada se destruye y se empieza nuevamente. La capa de abstracción de SignalR da la sensación de que es persistente y las conexiones de Web Sockets son verdaderamente persistentes.

1.3. DEFINICIÓN DE TÉRMINOS BÁSICOS

- Documento: Es el ente que contiene información relevante o asunto del trámite, las personas, áreas u oficinas
- Recibir Documento: Es la acción que consiste en registrar en el sistema los datos que contiene un documento ya sea de fuente interna (empresa) o externa (de la calle).
- Derivar Documento: Es la operación que consiste en registrar en el sistema el área de destino a donde va dirigido el documento, es decir a quien va dirigido (usuario, área u oficina y cargo).
- Proveer Documento: Es la tarea que consiste en registrar en la base de datos el número de proveído o número de registro que corresponde al documento que se va a responder o simplemente un nuevo documento que será enviado interna (empresa) o externa (de la calle).
- Archivar documento: Es el ejercicio que reside en registrar en el sistema los documentos con la denominación de archivados, aquellos que ya tienen una respuesta o rechazo según su petición.
- Persona: Es el sujeto que contiene información del personal de las diversas áreas que usarán el sistema informático, en la cual se harán rutinas como nuevo, listar, guardar, modificar y eliminar registro.
- Área: Contiene la información referente a las áreas u oficinas, en la que se ejecutarán las programaciones como nuevo, listar, guardar, modificar y eliminar registro.
- Referencia: Es el número de registro que corresponde de donde viene el documento, que tiene los datos del área de referencia.
- Hoja de Ruta: Indica el listado de las áreas por donde pasaron el documento, su respectiva fecha, su operación con el personal.
- Cuaderno de Documentos Recibidos: Consiste en el registro manual de los documentos (interno o externo) recibidos en un libro, donde se ingresa el número de registro por orden de llegada, fecha, asunto.

- Cuaderno de Documentos Enviados: Consiste en el registro manual de los documentos (interno o externo) enviados en un libro, donde se ingresa el número de registro por orden de llegada, fecha, asunto.

CAPITULO II: HIPOTESIS Y VARIABLES

2.1 FORMULACION DE HIPOTESIS

Hipótesis General:

H_i: La identificación de patrones de diseño mejora el proceso de actualización de nuevos requerimientos para la sistematización del Trámite de Documentos en Servicios Generales Barbosa.

Hipótesis Nula:

H₀: La identificación de patrones de diseño no mejora el proceso de actualización de nuevos requerimientos para la sistematización del Trámite de Documentos en Servicios Generales Barbosa.

2.2. VARIABLES Y OPERACIONALIDAD

2.2.1. Variable independiente:

Variables	Descripción	Definición
Variable independiente (x):	Identificación de patrones de diseño.	Busca dar soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Tabla 2: Descripción de variables

Fuente: Elaboración propia

2.2.2. Variable dependiente:

Variable dependiente (y):	Proceso de actualización de nuevos requerimientos.	Busca determinar que hará el Software, definiendo las relaciones de su operación y su implementación,
---------------------------	--	---

Tabla 3: Descripción de la variable

Fuente: Elaboración propia

VARIABLES	DESCRIPCIÓN	DEFINICIÓN	DIMENSIÓN	INDICADORES	ÍNDICES
Independiente	Identificación de patrones de diseño	Busca dar soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.	Tiempo de desarrollo	horas /hombres	Tiempo utilizado en las actividades de codificación con patrones de diseño.
			Clases y métodos.	Reutilización de clases y métodos.	Tiempo utilizado en las actividades de codificación sin patrones de diseño
Dependiente	Proceso de actualización de nuevos requerimientos.	Busca determinar que hará el Software, definiendo las relaciones de su operación y su implementación	Código de desarrollo	líneas de código	Código de desarrollo números de líneas

CAPITULO III: METODOLOGIA

3.1. TIPO DE INVESTIGACION

El presente estudio reúne las condiciones de una investigación tipo Aplicativa y por sus características es de nivel descriptivo.

3.3. DISEÑO MUESTRAL

El diseño de la investigación será el no experimental de corte transversal o transeccional.

Los estudios no experimentales son aquellos donde no hay manipulación de las variables en su análisis, Gómez (2013)

3.2.2. POBLACION

La población de estudio son 25 trabajadores de la empresa

3.2.2. MUESTRA

La muestra para la presente investigación estuvo constituida 10 trabajadores de la empresa la selección de la muestra se efectuó en forma no probabilística intencionada.

3.3. PROCEDIMIENTOS DE RECOLECCION DE DATOS

3.3.1. Técnica de recolección de datos

Para la presente investigación se empleó como técnica de recolección de datos fue la observación directa, para reunir datos con un propósito específico.

3.3.2. Instrumento de recolección de datos

Para la presente investigación se empleó como instrumento de recolección de datos una hoja electrónica de Excel.

3.4. PROCESAMIENTO Y ANALISIS DE DATOS

El análisis de datos significa como van a ser tratados los datos recolectados en sus diferentes etapas. Para ello se emplea la estadística como una herramienta. “Esta fase se presenta posterior a la aplicación del instrumento y finalizada la recolección de los datos. Se procederá a aplicar el análisis de los datos para dar respuesta a las interrogantes de la investigación” (Hevia, 2001, p. 46)

3.5. DISEÑO DE LA SOLUCIÓN

El objetivo general de esta fase es plantear la arquitectura para el ciclo de desarrollo del producto. Desde el inicio del proyecto, después de los requerimientos, se crearán prototipos de las interfaces de la captura de requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan del modelo del negocio.

Los principales objetivos para esta fase son:

- Crear una planificación fiable para la fase de codificación
- Demostrar que el Patrón de Diseño, permite realizar fácilmente los objetivos deseables.

En esta fase se analizará el dominio del problema, planteado en el modelamiento del negocio, y se establecerá los cimientos del patrón de diseño.

Los artefactos que se presentará en esta fase serán:

- Modelos de Casos de Uso de Requerimiento
- Diagrama de Comunicación
- Diagrama de Clases

3.5.1. REQUERIMIENTOS

La etapa de requerimientos es el siguiente flujo de trabajo de acuerdo a una metodología de desarrollo, la cual consiste en definir los servicios que el sistema debe proveer y las restricciones bajo la cual debe operar.

El objetivo primordial de esta disciplina es establecer las funciones que se quiere que satisfaga a nivel del sistema que se va a desarrollar.

En esta línea los requerimientos son exigencias que se deben de cumplir, de modo que los usuarios finales tienen que comprender y validar todas sus peticiones.

Los principales objetivos de esta disciplina son:

- Definir el alcance del sistema.
- Precisar la interface de usuario para el sistema, enfocado a las necesidades y objetivos del usuario.
- Instaurar y mantener el acuerdo entre clientes y los involucrados sobre lo que el sistema debería hacer.
- Contar con una base para estimar recursos y tiempo de desarrollo del sistema.

Los requisitos serán divididos en funcionales, que describirán las funciones que el software va a ejecutar; y los no funcionales, que especificarán criterios que puedan usarse para las operaciones.

3.5.2. MODELO DE CASOS DE USO

Los Modelos de Caso de Uso o historias de usuarios; representan los requerimientos del usuario, describe a través de dibujos la realidad de los procesos de trabajo y describe al sistema y su ambiente

3.5.3. APLICACIÓN DE PATRONES

3.5.3.1. SINGLETON:

Con el patrón SINGLETON se creó un objeto una sola vez. Al tener un problema de crear un objeto y dicho objeto se utilizara en diferentes partes del sistema, es innecesario crear el objeto varias veces.

Donde se aplicara es en la parte de usuario de inicio de sesión el cual podrá realizar alguna actividad y su logueo tiene que ejecutarse una sola vez. Y este se maneja mediante el patrón SINGLETON, que será creado una sola vez. También estará en la lógica de negocio. El cual estará implementado con el patrón Factory method.

La lógica del patrón SINGLETON consiste en tener un objeto estático de la clase propia y un método estático que devuelva la clase propia, en el método que devuelve la misma clase se evalua si existe en memoria que retorna lo que existe y si no que lo cree.

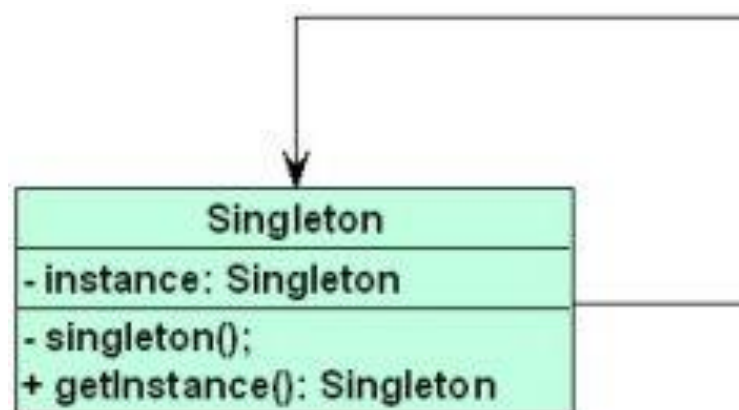


Figura 02: Diagrama UML de una clase que implementa el patrón singleton.

Fuente: Internet

```

} public class FabricaLN
{
    private static FabricaLN instancia = null;
    24 referencias
    public static FabricaLN _instancia()
    {
        if (instancia == null)
        {
            instancia = new FabricaLN();
        }
        return instancia;
    }
}

```

Figura 03: Implementación del Patrón Singleton.

Fuente: sistema

También se está aplicando en el pase de información entre formularios de forma más fluida.

Tenemos el formulario área, para asignar a un trabajador a su área respectiva, el trabajador para seleccionar en otro formulario, lista de trabajadores.

The screenshot shows a web application window titled 'Area'. At the top right, there is a status indicator 'Procesando'. The main content area contains a form with the following elements:

- Form fields: 'Id Area: ID', 'Area: [input]', and 'Trabajador: metroLabel1'.
- A button labeled 'Seleccionar Trabajador'.
- A search bar with the placeholder text 'uscar:' and a search icon.
- A table with the following columns: 'IdArea', 'Area', 'IdTrabajador', and 'Trabajador'. The table is currently empty.

On the right side of the form, there is a vertical sidebar with four buttons: 'AGREGAR' (green), 'LIMPIAR' (orange), 'ACTUALIZAR' (red), and 'ELMININAR' (purple).

Figura 04: Formulario Área.

Fuente: Elaboración Propia

```

9 referencias
public partial class FormArea : Mantenimiento
{
    List<Area> list;
    public int idTrabajador=0;
1 referencia
    public FormArea()
    {
        InitializeComponent();
    }
    public static FormArea intansia;
2 referencias
    public static FormArea GetInstancia() {
        if (intansia == null)
        {
            intansia = new FormArea();
        }
        return intansia;
    }
}

```

Figura 05: Implementación del Patrón Singleton en la creación de un nuevo formulario

Fuente: Elaboración propia

Tiene un atributo id del trabajador donde se guardara el id del trabajador seleccionado

IdPersona	DNI	Nombres	Apellidos	FechaNacimiento	EstadoCivil	Direccion
5	76028998	Franz	Mariño Torres	16/11/2017	Soltero	Libertad # 46
6	7894512	Heckerling	Aguilar	23/11/2017		Grau # 1567

Figura 06: Lista de selección para asignar a una nueva área.

Fuente: Elaboración propia

Al hacer doble click en el trabajador llamara al objeto formArea el cual está en memoria y asignara a su atributo Id.

```
private void ListaGrid_DoubleClick(object sender, EventArgs e)
{
    switch (NombreForm)
    {
        case "FormPersona": {

            break; }
        case "FormArea": {
            FormArea s = FormArea.GetInstancia();
            Persona p = ((Persona)ListaGrid.CurrentRow.DataBoundItem);
            s.TrabajadorLabel.Text = p.ToString();
            s.idTrabajador = p.IdPersona;
            this.Close();
        }
    }
}
```

Figura 07: Uso del Patrón SINGLETON.

Fuente: Elaboración Propia

Así es como se implementó el patrón SINGLETON.

3.5.3.2 FACTORY METHOD:

Consiste en una clase constructora de objetos. Tenemos el problema en la lógica de negocio donde están todas las clases que se emplearan en el sistema, tener crear uno por uno, con el patrón mencionado agilizaremos esos pasos teniendo una clase principal y referenciando a los objetos a crear; solo bastara con utilizar la clase fabricaLN el cual también se instanciara una sola vez con el patrón SINGLETON.

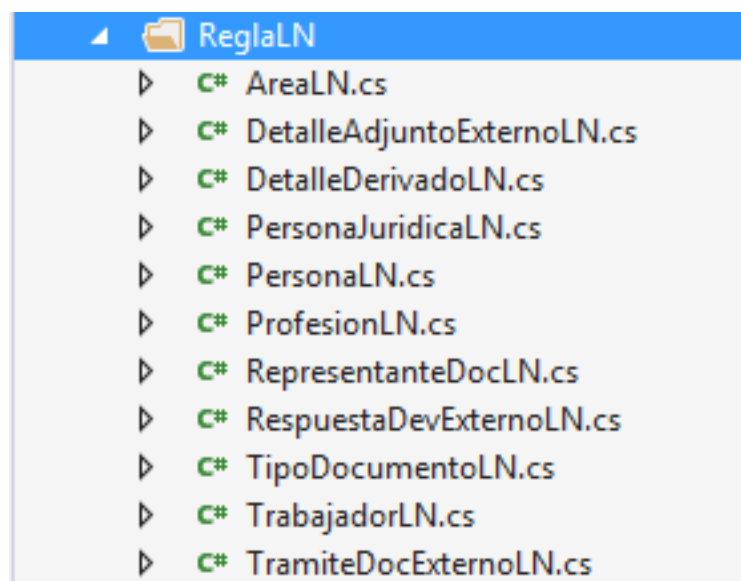


Figura 08: Lista de clases de la lógica de Negocios

Fuente: Elaboración Propia

Estos son los objetos el cual crearemos su fábrica.

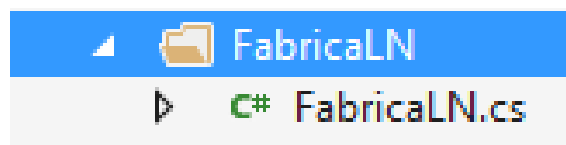


Figura 09: Una de las listas de Lógica de Negocios.

Fuente: Elaboración Propia


```

27 referencias
public class FabricaLN
{
    private static FabricaLN instancia = null;
    24 referencias
    public static FabricaLN _instancia()
    {
        if (instancia == null)
        {
            instancia = new FabricaLN();
        }
        return instancia;
    }

    9 referencias
    public IPersonaM Persona()
    {
        return new PersonaLN();
    }

    5 referencias
    public IProfesionM Profesion()
    {
        return new ProfesionLN();
    }

    4 referencias
    public ITrabajadorM Trabajador()
    {
        return new TrabajadorLN();
    }

    5 referencias
    public IAreaM Area()
    {
        return new AreaLN();
    }

    0 referencias
    public IDetalleAdjuntoExternoM DetalleAdjuntoExterno()
    {
        return new DetalleAdjuntoExternoLN();
    }
}

```

Figura 10: Código Fuente de la Implementación del Patrón Factory en la Lógica de Negocios.

Fuente: Elaboración Propia

Ya implementada la fábrica de lógica de negocio, se podrá crear los objetos solo con llamar a la fabricaLN

```

34 void cargarGrilla()
35 {
36     try
37     {
38         list = FabricaLN._instancia().Profesion().Listar();
39         profesionMetroGrid.DataSource = list;
40     }
41     catch (Exception ex)
42     {
43         Utilitario.ErrorDialog(ex.Message, this);
44         progEnd();
45     }
46 }
47
48
49

```

Figura 11: Uso del Patrón Factory.

Fuente: Elaboración Propia.

```

1 referencia
public class AreaP : IAreaM
{
    UsuarioP U = UsuarioP.GetInstancia();
    Exception F= new Exception("NO TIENE ACCESO, CONTACTESE CON EL ADMINISTRADOR");
7 referencias
    public bool Actualizar(Area p)
    {
        if (U.Rol == "")
        {
            return FabricaLN._instancia().Area().Actualizar(p);
        }
        else
        {
            throw F;
        }
    }
}

```

Figura 12: Uso del Patrón Factory.

Fuente: Elaboración Propia

3.5.3.3. FACTORY ABSTRAC:

Tenemos la problemática en los sistemas de información que muchas veces dependen del gestor de base de datos; en muchos caso la empresa necesita cambiar de gestor de base de datos, adaptar el sistema de información al nuevo gestor de base datos tomaría muchos recursos para la re-programación para adaptar al nuevo gestor de base de datos.

Se investigó y se logró analizar el patrón FACTORY ABSTRAC, el cual dice que es fábrica de fábricas.

Se creó una fábrica para SQL y otra fábrica para MYSQL; como ejemplo de dos objetos persona y trabajador:

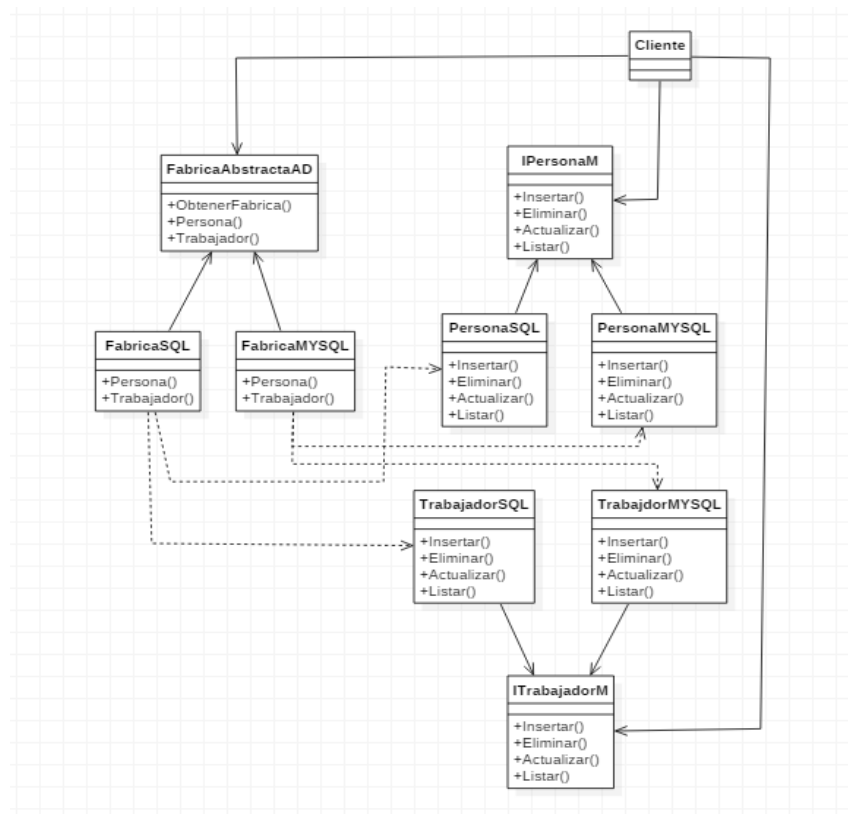


Figura 13: Diagrama UML del Patrón Factory Abstrac

. Fuente: Elaboración Propia.

El diagrama mencionado se obtuvo al análisis del diagrama oficial de FACTORY ABSTRAC

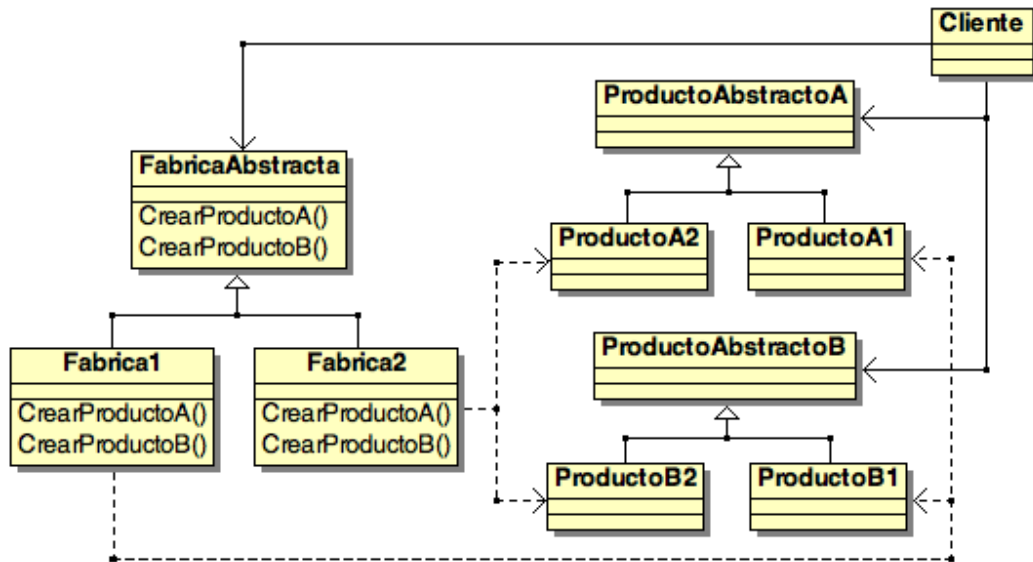


Figura 14: Diagrama UML del Patrón Factory Abstrac.

Fuente: Internet

3.5.3.4. PROXY:

Tenemos el problema en la validación de que el usuario pueda realizar alguna actividad. Ver si es un administrador, un usuario normal, etc.

Dichas validaciones se le puede hacer en la capa de presentación pero si en un futuro se cambia de arquitectura se tendría que volver a realizar las mismas validaciones; para una solución se tiene al patrón proxy, que servirá como intermediario para la comunicación en la capa de lógica de negocio con la capa de presentación, el patrón proxy se divide en 3:

- **PROXY REMOTO:** responsable de codificar una petición y sus argumentos, y de enviarla al objeto remoto.
- **PROXY VIRTUAL:** puede hacer caché de información del objeto real para diferir en lo posible el acceso a este.
- **PROXY DE PROTECCIÓN:** comprueba que el cliente tiene los permisos necesarios para realizar la petición.

El **PROXY DE PROTECCIÓN** es el que se aplicara en el sistema para eso se tendrá otra capa **PROXY**, en el cual se realizaran todas las validaciones del sistema, teniendo un código bien estructurado.

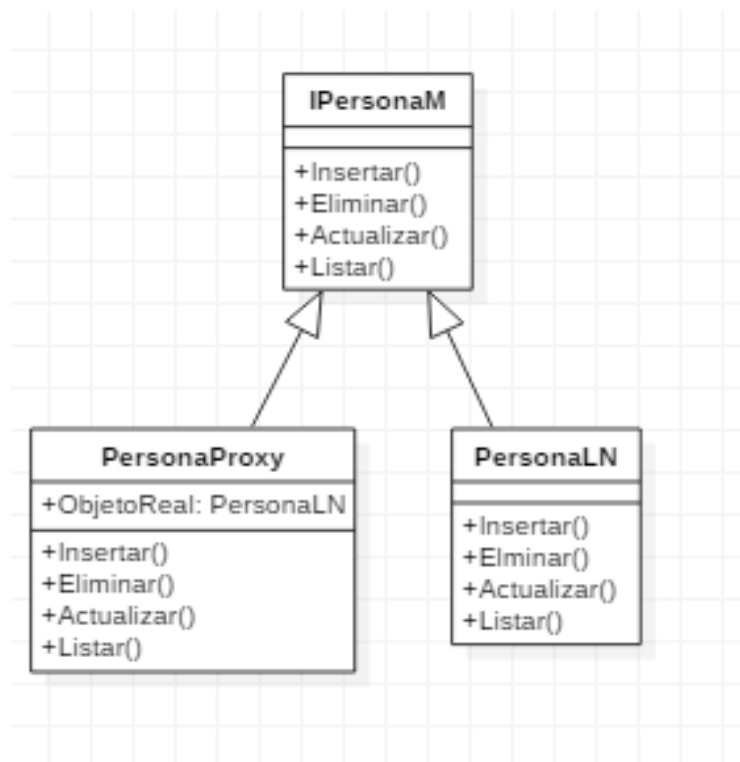


Figura 15: Diagrama de UML del Patrón Proxi.

Fuente: Elaboración Propia

Se tiene una interfaz IPersonaM en la cual están los métodos básicos que se realizara en ese objeto, tenemos el objeto real que se encuentra en la lógica de negocio PersonaLN el cual heredaremos a la interfaz IPersonaM; como dice la teoría del patrón proxi sirve como un intermediario; tenemos la clase PersonaProxy, que tiene como atributo un objeto PersonaLN, el cual también hereda a la interfaz IPersonaM. Al realizarse la implementación de la interfaz, es donde en los métodos de PersonaProxy, realizaremos las validaciones, si se eliminara, actualizara, insertara, o listara; eso dependerá de la lógica que se le aplique.

Esto nos basamos en el diagrama oficial del patrón proxy

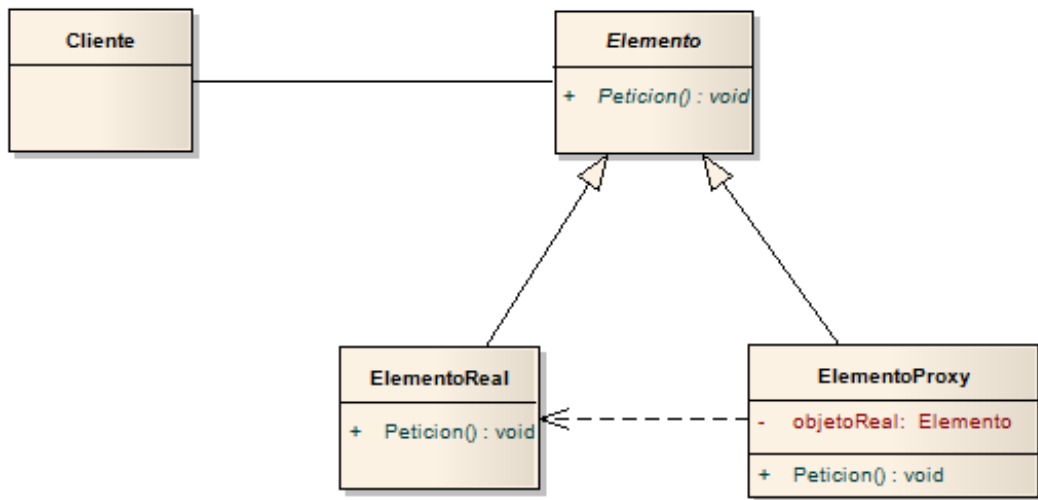


Figura 16: Diagrama de UML del Patrón Proxi.

Fuente: Internet

3.5.3.5. MVC:

El sistema tiene la necesidad de migrar a web, se necesita explotar toda la lógica que ya se implementó y solo crear una capa de presentación quedando la arquitectura de la siguiente manera:

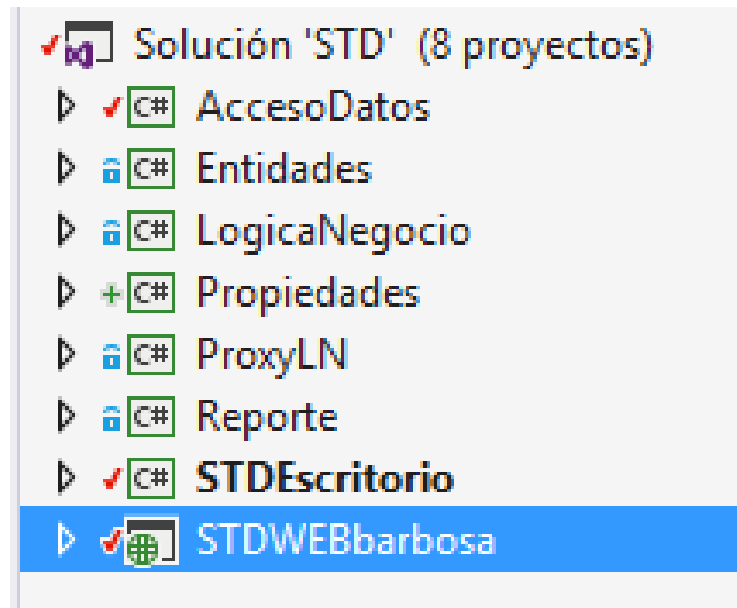


Figura 17: Capas con Patrón.

Fuente: Elaboración Propia.

3.6. ANÁLISIS

El proyecto de tesis denominado “Optimización con Patrones de Diseño de Software Para La Sistematización del Trámite de Documentos en Servicios Generales Barbosa”. Consiste en dar solución a un problema de control de los trámites documentarios que pasa por las distintas áreas de la empresa. Como parte del requerimiento del cliente, se determinó que el software contaría con 5 Versiones.

DESCRIPCIÓN DE LAS VERSIONES

VERSION	DESCRIPCION
Versión 1.0	Esta versión, estará en plataforma escritorio y que permitirá el manejo de usuarios; de los distintos trámites documentarios que realiza la entidad.
Versión 1.1	Consiste en la migración de todos los servicios de trámite a la plataforma web.
Versión 1.2	Consiste en la creación de un archivo adjunto, como cargo del proceso de tramite realizado
Versión 1.3	Migración a otro sistema gestor de base de datos. Con la finalidad de probar los patrones de diseño como parte de la implementación de buenas prácticas.
Versión 1.4	Permitirá poder realizar consultas y seguimiento de algún trámite realizado. Así como la integración del sistema de trámite web, para las distintas oficinas de la empresa, la cual contara con una herramienta de chat (Grupal e Individual) para la mayor interacción y coordinación de las actividades entre los usuarios.

Tabla 4: Descripción de versiones.

Fuente: Elaboración Propia

CUADRO DE RESUMEN DE LAS VERSIONES DEL SISTEMA

V 1.0	V 1.1	V 1.2	V 1.3	V 1.4
R1 Registro de nuevo de trámite externo.	R4 Migración Web		R9 Migración a otro gestor de base datos SQL SERVER.	R10 Consulta de trámite en web.
R2 Derivación de un trámite externo.	R5 Registro de nuevo trámite interno.	R8 Adjuntar un archivo PDF como copia del documento original.		R11 Chat Interno
R3 Respuesta de un trámite externo	R6 Derivación de un trámite interno.			
	R7 Respuesta de un trámite interno.			

Fuente: Elaboración Propia

REQUERIMIENTOS

NUMERO	REQUERIMIENTO	DESCRIPCIÓN
R1	Registro de nuevo de trámite externo.	Inicia en mesa partes, en donde se recibe y registra un documento.
R2	Derivación de un trámite externo.	El representante de oficina, tiene la opción de derivar el documento a otra área.
R3	Respuesta de un trámite externo	Respuesta a la derivación de trámite realizado.
R4	Migración web	Consiste en migrar los servicios de trámite

		documentario desde la plataforma de escritorio, a la plataforma web.
R5	Registro de nuevo trámite interno.	Se realiza el registro y evaluación de un trámite interno.
R6	Derivación de un trámite interno.	Se deriva a una oficina interna para la posterior atención de trámite.
R7	Respuesta de un trámite interno.	Se da respuesta a la solicitud de trámite.
R8	Adjuntar un archivo PDF como copia del documento original.	Se adjunta un archivo en formato pdf como cargo de un trámite realizado
R9	Migración a otro gestor de base datos SQL SERVER.	Migración a otro gestor de base de datos para mejorar el rendimiento y eficiencia del sistema a implementar.
R10	Consulta de trámite en web.	El solicitante de trámite externo puede consultar el estado de un trámite realizado.
R11	Chat Interno	Es el chat corporativo para los trabajadores de la empresa Barbosa, en dos niveles: Grupal e Individual. Para mejorar la interactividad y comunicación de los mismo.

Tabla 6: Cuadro de Requerimientos.

Fuente: Elaboración Propia

Es muy frecuente escuchar entre los conocedores del desarrollo de software, que un gran número de los proyectos de software fracasan por no realizar una adecuada definición, especificación, y administración de los requerimientos. Dentro de esa mala administración se pueden encontrar factores como la falta de participación del usuario, requerimientos incompletos y el mal manejo del cambio a los requerimientos.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que se enfoca un área fundamental: la definición de lo

que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas.

Estas son algunas definiciones de requerimientos:

De acuerdo a la IEEE es una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste.

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos.

A menudo, los requerimientos de sistemas software se clasifican en funcionales y no funcionales, o como requerimientos del dominio:

3.7. VISIÓN GENERAL

Nuestro objetivo comprende desarrollar un sistema de trámite documentario basado en módulos configurables, que permita automatizar parte del proceso generado por un usuario.

Además existe la posibilidad de mantener y manejar la información de los documentos, El administrador tiene la posibilidad de modificar toda esta información. Existirán al menos 3 tipos de usuarios a saber:

Nombre	Descripción
Mesa Partes	Denominada “Unidad de Recepción Documental”, por finalidad la recepción de documentos que son presentados a la entidad de la cual depende
Administrador	Es el encargado de sistema que tendrá el control total del sistema.
Representante de oficina	Donde se derivara el documento o el que enviara el documento.

Tabla 7: Cuadro de Usuarios del Sistema.

Fuente: Elaboración Propia

3.8. MODELO DE NEGOCIO

DIAGRAMA DE CUN (DIAGRAMA DE CASO DE USO DE NEGOCIO)

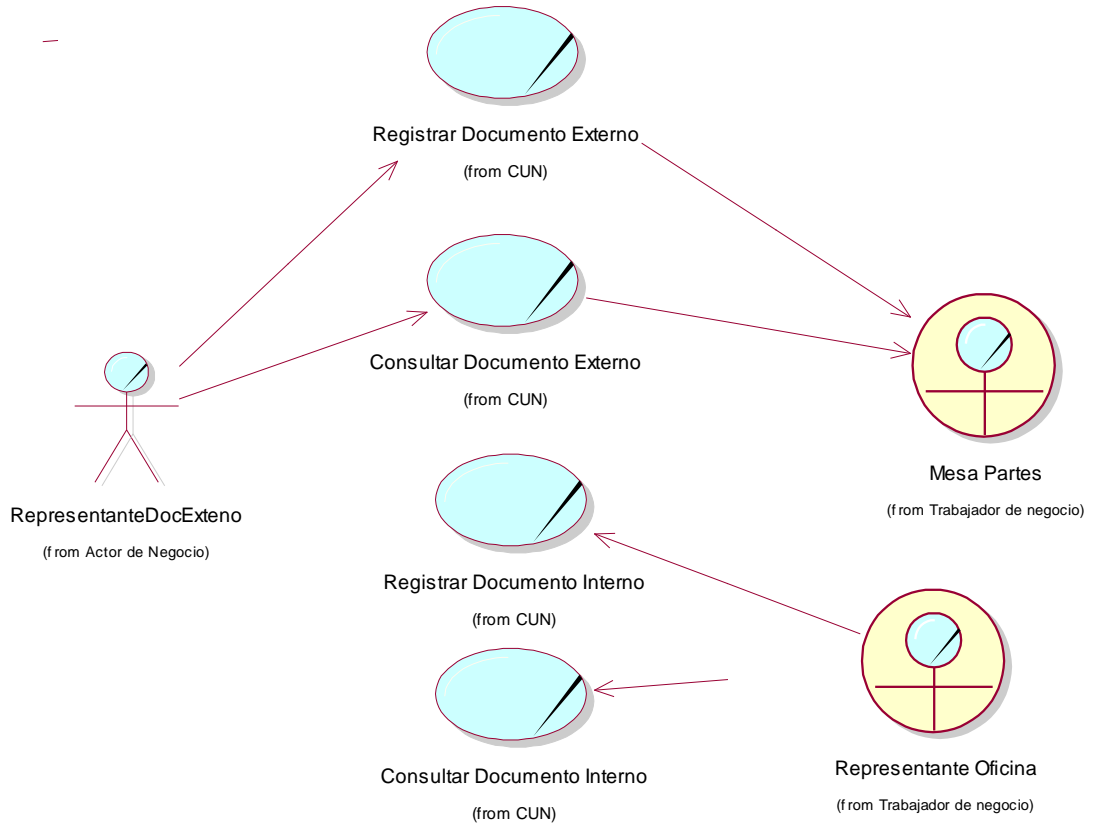


Figura 18: Diagrama De Caso De Uso De Negocio.

Fuente: Elaboración Propia.

TRABAJADORES DE NEGOCIO

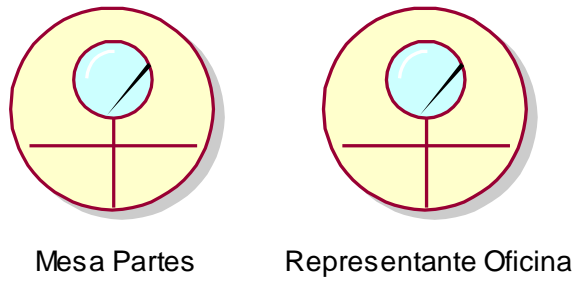


Figura 19: Diagrama de trabajadores de Negocio.

Fuente: Elaboración Propia

ENTIDADES DE NEGOCIO

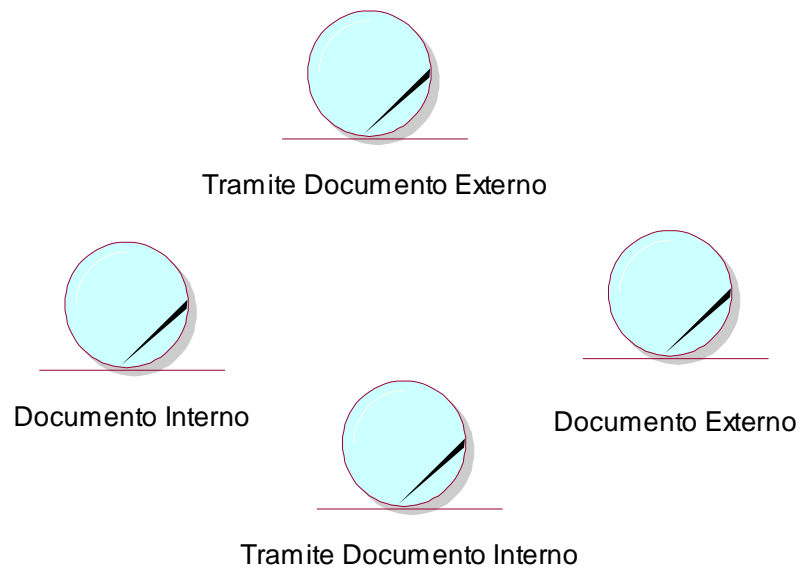


Figura 20: Diagrama de Entidades de Negocio.

Fuente: Elaboración Propia

CUN Registrar documento Interno

DESCRIPCION	Permite realizar el trámite documentario entre las oficinas de la empresa
FLUJO BASICO	El representante de la oficina desear enviar documento a otra oficina tendrá que escanear el documento como backup y detallar todos las caracteriza del documento para luego él representate de la oficina destino aceptara el documento verificando y terminando con el tramite documentario.
FLUJO ALTERNATIVO	-

Tabla 8: Cuadro de Descripción para Registrar documento Interno. Fuente: Elaboración Propia.

CUN Registrar Documento Externo

DESCRIPCION	Se recibirá documento externos para cualquier tipo de los casos, estos documentos vendrán con algún representante y serán derivados a alguna oficina de la empresa.
FLUJO BASICO	El representante del documento externo se dirige a mesa de partes solicitando que quiere realizar un trámite, mesa de partes pide todos los datos necesarios, verifica el documento si todo es correcto se deriva a la oficina destino del documento.
FLUJO ALTERNATIVO	Si al documento le faltan algunos datos o no se puede tramitar, se rechaza y se anula el trámite.

Tabla 9: Cuadro de Descripción para Registrar documento Externo. Fuente: Elaboración Propia.

CUN Consultar Documento Externo

DESCRIPCION	Se verifica el estado del trámite del documento.
FLUJO BASICO	El representante del documento se acerca a mesa de partes con un código para solicitar el estado de su documento, si tiene alguna respuesta, mesa de partes verifica el estado de su documento y comunica al representante del documento
FLUJO ALTERNATIVO	-

Tabla 10: Cuadro de Descripción para Consultar Documento Externo. Fuente: Elaboración Propia.

CUN Consultar Documento Interno

DESCRIPCION	El representante de una oficina accederá al estado de su trámite de documento para que confirme si aceptaron o no.
FLUJO BASICO	El representante da su código de seguimiento del documento, el cual obtendrá los datos necesarios del documento, como se encuentra actualmente.
FLUJO ALTERNATIVO	-

Tabla 11: Cuadro de Descripción para Consultar Documento Interno. Fuente: Elaboración Propia.

3.8.1. REALIZACION DE LOS CUN

CUN REGISTRAR DOCUMENTO INTERNO

DIAGRAMA DE ACTIVIDAD DETALLADO

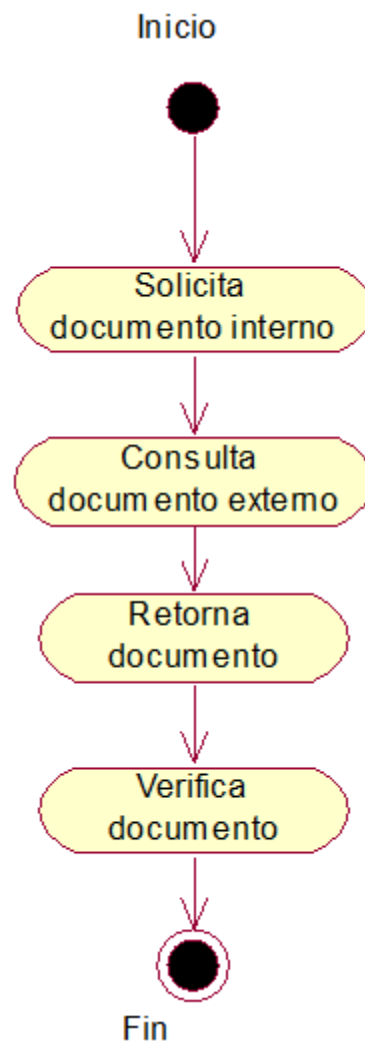


Figura 21: Diagrama de Actividades detallado para registrar documento interno.

Fuente: Elaboración Propia.

DIAGRAMA DE OBJETO

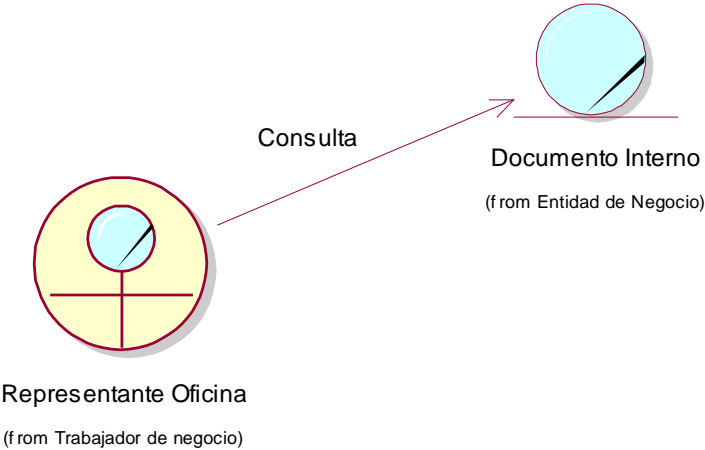


Figura 22: Diagrama de Objeto para consultar documento interno.

Fuente: Elaboración Propia

3.8.2. CUN REGISTRAR DOCUMENTO EXTERNO

DIAGRAMA DE ACTIVIDAD DETALLADO

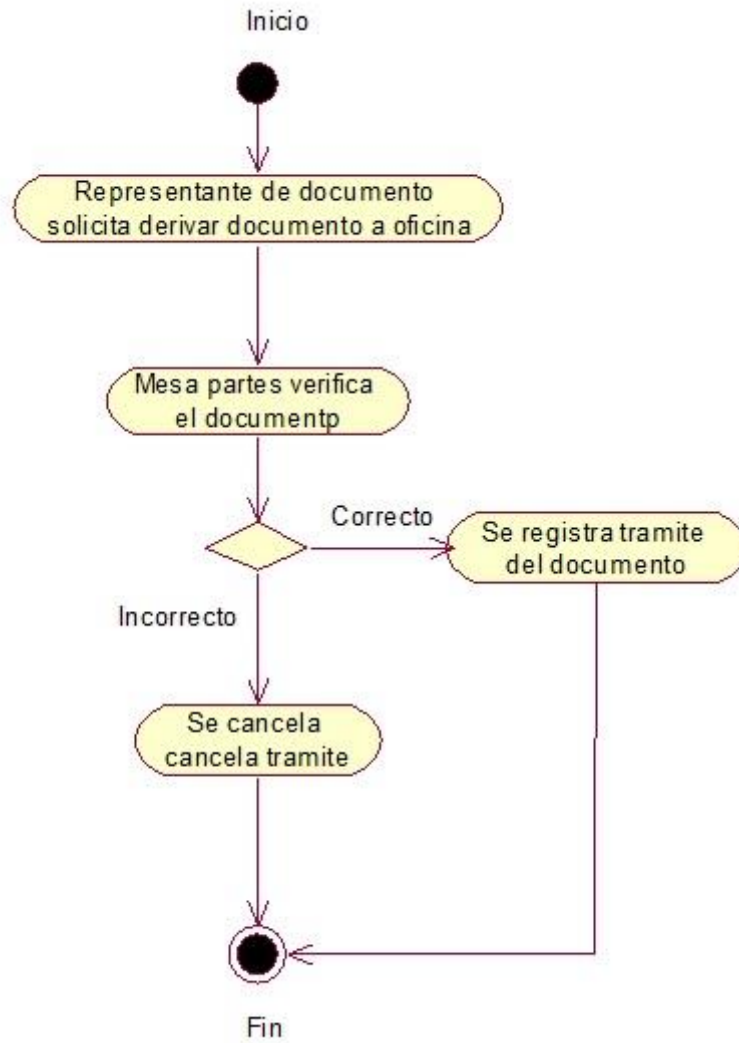


Figura 23: Diagrama de Actividad Detallado para registrar documentos externo.

Fuente: Elaboración Propia.

DIAGRAMA DE OBJETO

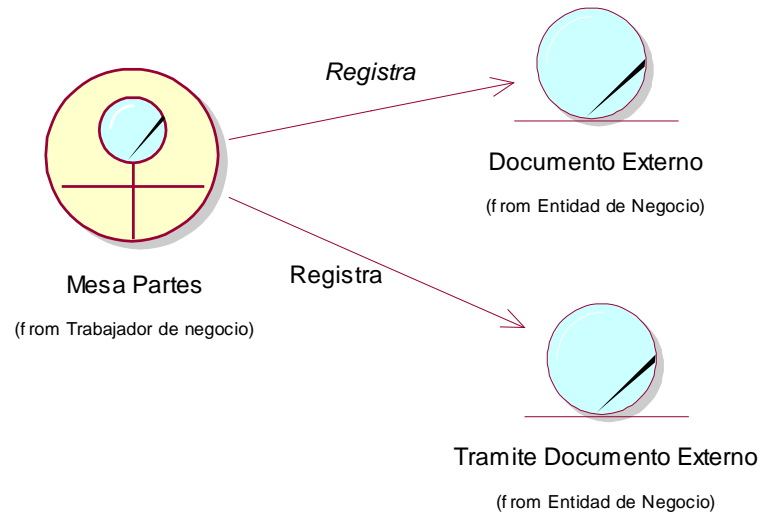


Figura 24: Diagrama de Objeto para registrar documentos externos.

Fuente: Elaboración Propia.

3.8.3. CUN CONSULTAR DOCUMENTO EXTERNO

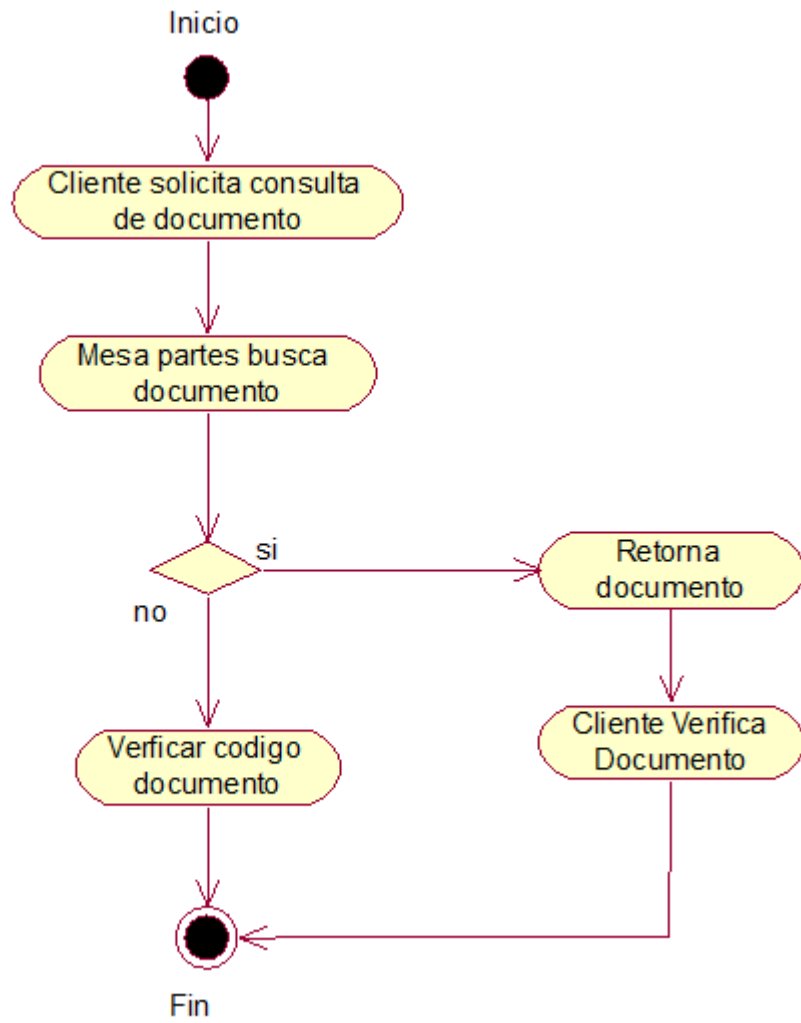


DIAGRAMA DE ACTIVIDAD DETALLADO

Figura 25: Diagrama de Actividad Detallado para consultar documentos externos.

Fuente: Elaboración Propia.

DIAGRAMA DE OBJETO

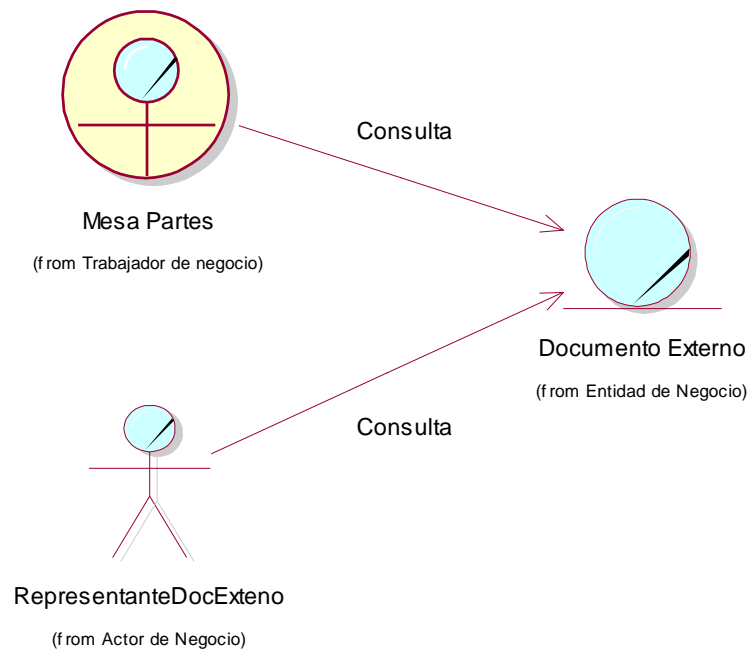


Figura 26: Diagrama de Objeto para consultar documentos externos.

Fuente: Elaboración Propia.

3.8.4. CUN CONSULTAR DOCUMENTO INTERNO

DIAGRAMA DE ACTIVIDAD DETALLADO

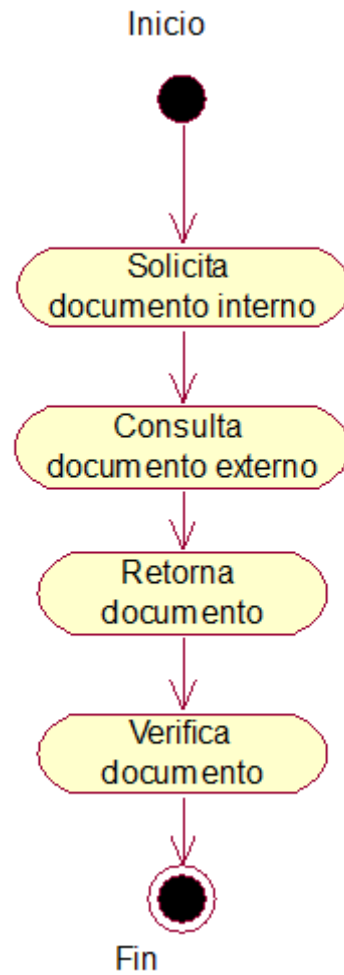


Figura 27: Diagrama de Actividad Detallado para consultar documentos internos.

Fuente: Elaboración Propia.

DIAGRAMA DE OBJETO

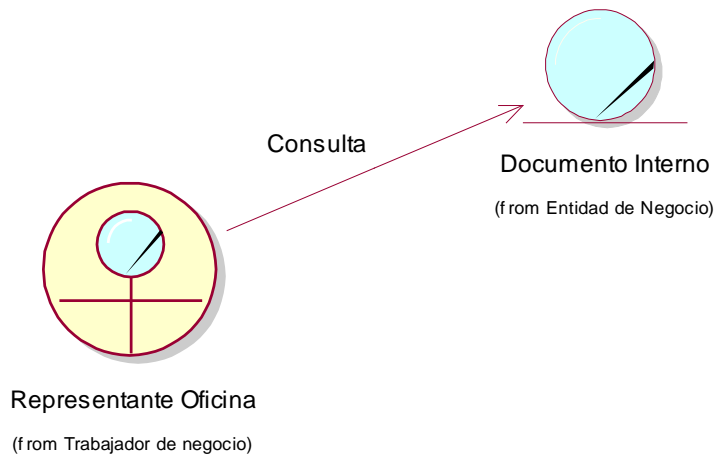


Figura 28: Diagrama de Objeto para consultar documentos interno.

Fuente: Elaboración Propia.

3.8.5. DIAGRAMA E-R (MODELADO DE DATOS)

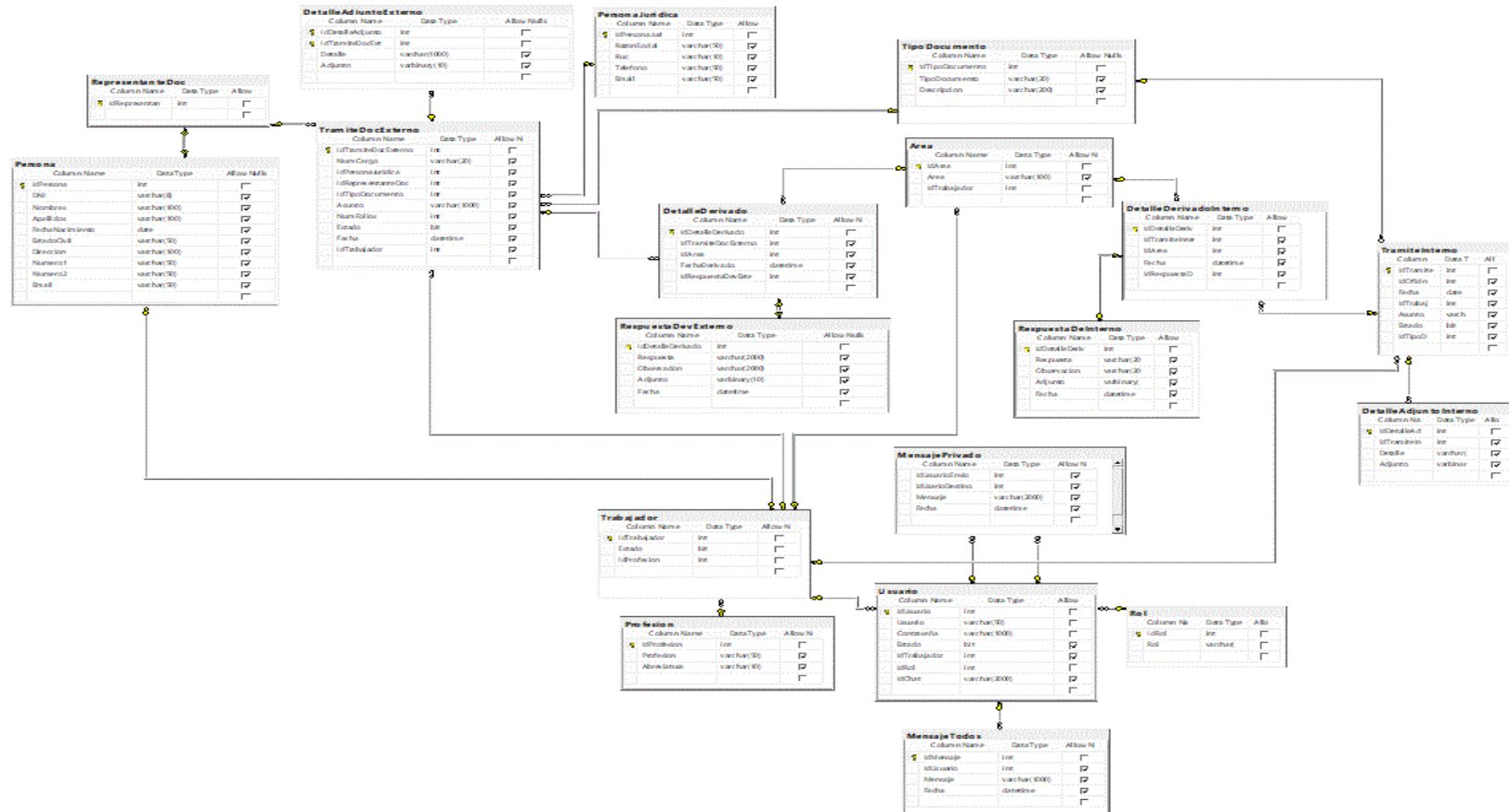


Figura 29: Diagrama de modelado de datos. Fuente

Elaboración Propia.

CAPITULO IV: RESULTADOS

TABLA N° 01: Las etapas en función del tiempo del ciclo de vida.

ETAPAS	Tiempos del ciclo de vida sin Patrones(días)	Tiempos del ciclo de vida con Patrones(días)
Análisis	5	5
Diseño	11	6
Implementación	7	5
PROMEDIO	7.67	5.33

Fuente: Elaboración Propia

La tabla N° 1, nos muestra que en las etapas de análisis, diseño e Implementación tiene un menor tiempo del ciclo de vida con patrones que en promedio es de 5.33 días mientras Tiempos del ciclo de vida sin Patrones en promedio es de 7.67 días, Concluyendo que las etapas en función del tiempo del ciclo de vida con patrones son más eficientes en un 69.5% ; de esta manera damos respuesta al primer objetivo específico de la tesis que es : Determinar la eficiencia de las etapas en función del tiempo del ciclo de vida con o sin Patrones.

TABLA N° 2. Uso de patrones de diseño Y el tiempo en las actividades de métricas de Diseño

SOFTWARE BASE CON PATRONES DE DISEÑO		ACTIVIDADES DE METRICA DE DISEÑO						
R Q	DETALLE	1.- Identificación de Patrones a utilizar		2.-IMPLEMENTAR Patrones en el módulo requerido		3.- ADICIONAR CLASES Y METODOS		
		PATRON DE DISEÑO	TIEMP O/DIAS	PATRON DE DISEÑO	TIEMP O/DIAS	PATRON DE DISEÑO	TIEMP O/DIAS	
		1	Tramite externo	RESTRICCIONES	PROXY	1	PROXY	1
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	1	NG	1	NG	0	
	CREACION DE							
	PERSISTENCIA DE							
	DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	IMPLEMENTACION							
	WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
2	Respuesta de tramite	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	1	NG	1	NG	0	
	CREACION DE							
	PERSISTENCIA DE							
	DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	IMPLEMENTACION							
	WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
3	Derivación de tramite	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	1	NG	1	NG	0	
	CREACION DE							
	PERSISTENCIA DE							
	DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	IMPLEMENTACION							
	WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
4	Autenticación de usuario	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	1	NG	1	NG	0	
	CREACION DE	NINGUNO	0	NINGUNO	0	NINGUNO	0	

PERSISTENCIA DE DATOS					
ABSTRER DB IMPLEMENTACION	NINGUNO	0	NINGUNO	0	NINGUNO 0
WEB	NINGUNO	0	NINGUNO	0	NINGUNO 0
		8	8		0

Fuente: elaboración propia

TABLA N° 3. Uso sin patrones de diseño Y el tiempo en las actividades de métricas de Diseño.

SOFTWARE BASE SIN PATRONES DE DISEÑO		ACTIVIDADES DE METRICA DE DISEÑO						
R Q	DETALLE	1.- Identificación de Patrones a utilizar		2.-IMPLEMENTAR Patrones en el módulo requerido		3.- ADICIONAR CLASES Y METODOS		
		PATRON DE DISEÑO	TIEMPO /DIAS	DISEÑO	TIEMPO/ DIAS	DISEÑO	TIEMPO /DIAS	
1	Tramite externo	RESTRICCIONES	PROXY	0	PROXY	0	PROXY	0
	MANEJO DE USUARIOS	SINGLETON	0	SINGLETON	0	SINGLETON	0	
	CREACION DE PERSISTENCIA DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRER DB IMPLEMENTACION	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
2	Respuesta de tramite	RESTRICCIONES	PROXY	0	PROXY	1	PROXY	0
	MANEJO DE USUARIOS	SINGLETON	0	SINGLETON	1	SINGLETON	0	
	CREACION DE PERSISTENCIA DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	

	ABSTRAER DB IMPLEMENTACION WEB	NINGUNO	0	NINGU NO	0	NING UNO	0
3	Derivación de tramite	RESTRICCIONES	PROXY	0	PROXY	1	PROX Y
	MANEJO DE USUARIOS	SINGLETO	NG	0	SINGLE TONG	1	SING LETO NG
	CREACION DE PERSISTENCIA DE DATOS	NINGUNO	0	NINGU NO	0	NING UNO	0
	ABSTRAER DB IMPLEMENTACION WEB	NINGUNO	0	NINGU NO	0	NING UNO	0
			NINGUNO	0	NINGU NO	0	NING UNO
4	Autenticación de usuario	RESTRICCIONES	PROXY	0	PROXY	1	PROX Y
	MANEJO DE USUARIOS	SINGLETO	NG	0	SINGLE TONG	1	SING LETO NG
	CREACION DE PERSISTENCIA DE DATOS	NINGUNO	0	NINGU NO	0	NING UNO	0
	ABSTRAER DB IMPLEMENTACION WEB	NINGUNO	0	NINGU NO	0	NING UNO	0
			NINGUNO	0	NINGU NO	0	NING UNO
			0		6		0

Fuente: elaboración propia

Las tablas N° 2 y N° 3, reportan que el Uso de patrones de diseño el tiempo en las actividades de métricas de Diseño es de 8 días mientras el Uso sin patrones de diseño el tiempo en las actividades de métricas de Diseño es de 6 días, entonces podemos determinar que el Uso sin patrones de diseño el tiempo en las actividades de métricas de Diseño es menor que el Uso de patrones de diseño el tiempo en las actividades de métricas de Diseño ; de esta manera damos respuesta

al objetivo específico 2: Determinar el Uso con y sin patrones de diseño Y el tiempo en las actividades de métricas de Diseño.

Tabla N° 4. Actualizaciones de los patrones sin diseño en relación a las actividades de métrica de diseño.

ACTUALIZACION 1 SIN PATRONES DE DISEÑO		ACTIVIDADES DE METRICA DE DISEÑO						
RQ	DETALLE	1.- Identificación de Patrones a utilizar		2.-IMPLEMENTAR Patrones en el módulo requerido		3.- ADICIONAR CLASES Y METODOS		
		PATRON DE DISEÑO	TIEMP O/DIA S	PATRON DE DISEÑO	TIEMP O/DIA S	PATRON DE DISEÑO	TIEMP O/DIA S	
1	Tramite interno	RESTRICCIONES	PROXY	0	PROXY	0	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	0	NG	0	NG	0	
	CREACION DE							
	PERSISTENCIA							
	DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
IMPLEMENTACION WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0		
2	Respuesta de tramite INTERNO	RESTRICCIONES	PROXY	0	PROXY	1	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	0	NG	1	NG	0	
	CREACION DE							
	PERSISTENCIA							
	DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
IMPLEMENTACION WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0		
3	Derivación de tramite INTERNO	RESTRICCIONES	PROXY	0	PROXY	1	PROXY	0
	MANEJO DE	SINGLETO		SINGLETO		SINGLETO		
	USUARIOS	NG	0	NG	1	NG	0	
	CREACION DE							
	PERSISTENCIA							
	DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	
	ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
IMPLEMENTACION WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0		

N WEB								
4	Migración Web	RESTRICCIONES	PROXY	0	PROXY	1	PROXY	0
		MANEJO DE	SINGLETO		SINGLETO		SINGLETO	
		USUARIOS	NG	0	NG	1	NG	0
		CREACION DE						
		PERSISTENCIA						
		DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0
		ABSTRAER DB	NINGUNO	0	NINGUNO	0	NINGUNO	0
		IMPLEMENTACION WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0
			0	6		0		

Fuente: elaboración propia

Tabla N° 5. Actualizaciones de los patrones con diseño en relación a las actividades de métrica de diseño.

ACTUALIZACION 2 CON PATRONES DE DISEÑO		ACTIVIDADES DE METRICA DE DISEÑO						
R Q	DETALLE	1.- Identificación de Patrones a utilizar		2.-IMPLEMENTAR Patrones en el módulo requerido		3.- ADICIONAR CLASES Y METODOS		
		PATRON DE DISEÑO	TIEMP O/DIAS	PATRON DE DISEÑO	TIEMP O/DIAS	PATRON DE DISEÑO	TIEMP O/DIAS	
		1	Tramite interno	RESTRICCIONES	PROXY	1	PROXY	1
	MANEJO DE		SINGLETO		SINGLETO		SINGLETO	
	USUARIOS		NG	1	NG	1	NG	0
	CREACION DE							
	PERSISTENCIA							
	DE DATOS		NINGUNO	0	NINGUNO	0	NINGUNO	0
	ABSTRAER DB		NINGUNO	0	NINGUNO	0	NINGUNO	0
	IMPLEMENTACION WEB		NINGUNO	0	NINGUNO	0	NINGUNO	0
2	Respuesta de tramite INTERNO	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
		MANEJO DE	SINGLETO		SINGLETO		SINGLETO	
		USUARIOS	NG	1	NG	1	NG	0
		CREACION DE						
	PERSISTENCIA							
	DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0	

	ABSTRAER DB IMPLEMENTACI ON WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0	
3	Derivación de tramite INTERNO	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
		MANEJO DE USUARIOS	SINGLETO		SINGLETO		SINGLETO	
		CREACION DE PERSISTENCIA	NG	1	NG	1	NG	0
		DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0
		ABSTRAER DB IMPLEMENTACI ON WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0
			NINGUNO	0	NINGUNO	0	NINGUNO	0
			NINGUNO	0	NINGUNO	0	NINGUNO	0
4	Migración Web	RESTRICCIONES	PROXY	1	PROXY	1	PROXY	0
		MANEJO DE USUARIOS	SINGLETO		SINGLETO		SINGLETO	
		CREACION DE PERSISTENCIA	NG	1	NG	1	NG	0
		DE DATOS	NINGUNO	0	NINGUNO	0	NINGUNO	0
		ABSTRAER DB IMPLEMENTACI ON WEB	NINGUNO	0	NINGUNO	0	NINGUNO	0
			NINGUNO	0	NINGUNO	0	NINGUNO	0
			NINGUNO	0	NINGUNO	0	NINGUNO	0
			8			8	0	

Fuente: elaboración propia

- Las tablas 4 y 5 muestran las actualizaciones de los patrones sin diseño el tiempo es menor relacionado con la actualización de los patrones con diseño que son de 6 y 8 días respectivamente; concluyendo Las Actualizaciones de los patrones con diseño en relación a las actividades de métrica de diseño son más eficientes en un 75% que las Actualizaciones de los sin patrones con diseño en relación a las actividades de métrica de diseño con esto se da respuesta al objetivo específico 3: Determinar la eficiencia de las actualizaciones de los patrones con y sin diseño en relación a las actividades de métrica de diseño.

Con los resultados de las tablas 1,2,,3,4 y 5 de esta tesis donde hay un elevado porcentaje de eficiencia tanto en la identificación de patrones de diseño como en

el proceso de actualización de nuevos requerimientos para sistematizar el del Trámite de Documentos en Servicios Generales Barbosa, esto implica que hay una mejora en los procesos ; esto significa que aceptamos la hipótesis de Investigación : La identificación de patrones de diseño mejora el proceso de actualización de nuevos requerimientos para la sistematización del Trámite de Documentos en Servicios Generales Barbosa.

CAPITULO V: DISCUSIONES

- Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

- Ponce Romero (2013) en su tesis propone el desarrollo de un sistema con tecnología web para mejorar la actualización del inventario de equipos dslam de Telefónica del Perú, basado patrones de diseño modelo vista controlador enfocado principalmente para realizar actualizaciones de los datos en tiempo real, de tal manera que se obtenga mejoras en los resultados al realizar cualquier tipo de actualización en el inventario. También ha tomado como estudio los procesos del desarrollo del software que constan en: Análisis, diseño, implementación y pruebas la cual los permitirá mejorar la actualización del inventario realizando cualquier tipo de consultas como agregar, eliminar y consultar la información en la base de datos de nuestro inventario centrándonos en la etapa del diseño utilizando la metodológica RUP el cual es un proceso de desarrollo utilizando como herramienta el lenguaje unificado de modelado(UML) para obtener los resultados propuestos.
- Edson Bryan Castañeda Rojas, 2016 (Rojas, 2016). En su tesis propone el uso de estructuras repetitivas o plantillas, que facilita el funcionamiento de un programa informático que extrae contenido Web, dicha intrusión genera un incremento considerable en el uso de recursos, considerando la permanente ejecución de instrucciones para obtener tanto contenido como

sea posible. Con la finalidad de reducir la vulnerabilidad de los sitios Web frente a procesos de extracción de contenido masivo, se planteó un patrón de diseño de software tomando como referencia el patrón Template View de Martin Fowler, al cual se agregó una capa de aleatorización que permita generar estructuras HTML no predecibles. Mediante la aplicación de una herramienta de extracción de contenido a un sitio Web de prueba, cuya capa de presentación se desarrolló tomando en cuenta el patrón de diseño propuesto, se logró verificar una reducción considerable de la cantidad de datos extraídos.

- Maricruz de Lourdes Acosta Yerovi, 2013 (Yerovi, 2013). En su tesis propuso la búsqueda constante por mejorar la calidad y el diseño del software, técnicas y tecnologías han evolucionado. Se desarrolló diversas estrategias y habilidades, varias son producto de mejoras realizadas sobre estrategias anteriores. Los patrones de diseño probablemente son las estrategias más versátiles, consisten en la reutilización de soluciones, no solamente de código fuente, sino que constituyen soluciones integrales a problemas repetitivos en el desarrollo del software. Es importante señalar que los patrones de diseño JEE, recopilan un conjunto de buenas prácticas que se han venido desarrollando en los últimos años para el desarrollo de sistemas web. La presencia de patrones de diseño conduce a soluciones estándares, fácilmente comprensibles y mantenibles por parte de los desarrolladores. El enfoque principal de este trabajo es el conocimiento de los patrones de diseño JEE 6, ya que a pesar de los beneficios que su implementación genera, son tratados a menudo como patrones J2EE, obstruyendo la funcionalidad de los patrones de diseño y de la plataforma en sí.
- Cristian Giovanni Martínez Rodríguez, 2012; (RODRÍGUEZ, 2012). En su tesis propone plantear un nuevo escenario en donde el software conoce su entorno, se integra a él y responde a necesidades tanto de un usuario como de otro componente de software, sin embargo, esto presenta el

inconveniente de software desarrollado con distintas tecnologías y lograr la interacción de diferentes tecnologías con el menor costo y esfuerzo posible. Es entonces necesario conocer la alternativa de la utilización de una arquitectura orientada a servicios para facilitar la interacción entre los distintos sistemas que forman el entorno empresarial, que puede estar compuesto por sistemas desarrollados con diferentes tecnologías y épocas; y conseguir aún la integración de los mismos en un entorno que permita formar un sistema mayor.

- Ing. Darío Yorio, 2006 (Yorio, 2006). En su tesis propone identificación y Clasificación de Patrones En El Diseño De Aplicaciones Móviles

CAPITULO VI: CONCLUSIONES

- Las etapas en función del tiempo del ciclo de vida con patrones son más eficientes en un 69.5%.
- En el uso sin patrones de diseño el tiempo en las actividades de métricas de Diseño es menor que el Uso de patrones de diseño el tiempo en las actividades de métricas de Diseño
- Las Actualizaciones de los patrones con diseño en relación a las actividades de métrica de diseño son más eficientes en un 75% que las Actualizaciones de los sin patrones con diseño en relación a las actividades de métrica de diseño

CAPITULO VII: RECOMENDACIONES

Se recomienda la utilización de patrones de diseño en la etapa de diseño del ciclo de vida del software para optimizar los tiempos de implementación de los requisitos.

CAPITULO VIII: FUENTES DE INFORMACIÓN

Antonio, Leiva. 2016. Patrones de diseño de software. Patrones Creacionales. 2016.

Fundación Wikimedia, Inc. 2019. MySQL. MySQL. 2019.

Gama, Erich, y otros. 2018. Gang of Four. Design Patterns. s.l. : Wikipedia, 2018.

Gamma, Erich. 1994. repositorio utn. [En línea] 1994. [Citado el: 29 de 10 de 2016.] <http://repositorio.utn.edu.ec>.

Gonzales, Renan. 2015. Microsoft Office Word. Sistema de Tramite Documentario. 2015.

Iturralde, Andrés. 2013. SignalR. Introducción a SignalR. 2013.

Jim Coplien. 1996. Repositorio UTN. [En línea] 1996. [Citado el: 29 de 10 de 2016.] <http://repositorio.utn.edu.ec>.

Martínez, Jesus. 2012. PROCESO ÚNICO DE DESARROLLO DE SOFTWARE. RUP. 2012.

Martínez, Joe. 2017. Rational Software Modeler . Software utilizados para modelar UML. 2017.

Ponce Romero, Jhadira Marilú y Valderrama Bacilio, Larry Billy. 2013. Repositorio Institucional INUTRU. Repositorio Institucional INUTRU. [En línea] 2013. [Citado el: 20 de OCTUBRE de 2016.] <http://dspace.unitru.edu.pe/handle/UNITRU/8330>.

RODRÍGUEZ, CRISTIAN GIOVANNI MARTÍNEZ. 2012. Biblioteca USAC. Biblioteca USAC. [En línea] 2012. [Citado el: 22 de octubre de 2016.] http://biblioteca.usac.edu.gt/tesis/08/08_0584_CS.pdf.

Rojas, Edson Bryan Castañeda. 2016. Repositorio PUCP. Repositorio PUCP. [En línea] 2016. [Citado el: 22 de Octubre de 2016.] <http://tesis.pucp.edu.pe/repositorio/handle/123456789/7513>.

Yerovi, Maricruz de Lourdes Acosta. 2013. Repositorio UTN. Repositorio UTN. [En línea] 2013. [Citado el: 20 de octubre de 2016.]

<http://repositorio.utn.edu.ec/bitstream/123456789/2787/1/04%20ISC%20266%20TESIS.pdf>.

Yorio, Ing. Darío. 2006. Catalogo Info UNLP. Catalogo Info UNLP. [En línea] 2006. [Citado el: 22 de Octubre de 2016.] catalogo.info.unlp.edu.ar/meran/getDocument.pl?id=1289.

ANEXOS

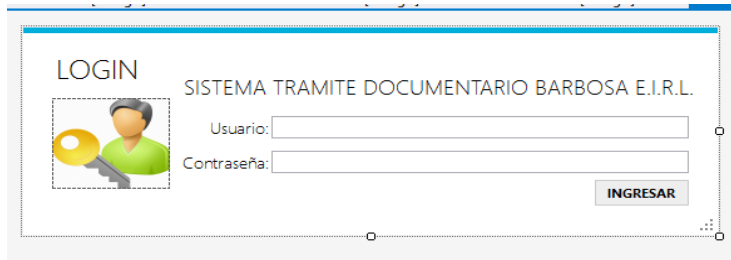
ANEXO 1: MATRIZ DE CONSISTENCIA

Planteamiento del Problema	Objetivo	Hipótesis	Variables	Indicadores
<p>Problema General</p> <p>¿La inclusión de nuevos requisitos demanda esfuerzo y tiempo adicionales en la etapa de diseño del Sistema de Trámite Documentario Barbosa?</p> <p>Problema Especifico</p> <p>¿Los patrones de diseño afectan el tiempo de desarrollo en las actividades de codificación?</p>	<p>Objetivo General</p> <p>Determinar si el uso con patrones de diseño optimiza el tiempo demandado para la inclusión de nuevos requisitos en la etapa de Diseño del Sistema de Trámite Documentario.</p> <p>Objetivo Especifico</p> <ul style="list-style-type: none"> • Determinar la eficiencia de las etapas en función del tiempo del ciclo de vida con o sin Patrones. • Determinar el Uso de patrones de diseño Y el tiempo en las actividades de métricas de Diseño. • Determinar la eficiencia de las actualizaciones de los patrones con y sin diseño en relación a las actividades de métrica de diseño. 	<p>H1 La identificación de patrones de diseño mejora el proceso de actualización de nuevos requerimientos para la sistematización del Trámite de Documentos en Servicios Generales Barbosa.</p> <p>Ho La identificación de patrones de diseño no mejora el proceso de actualización de nuevos requerimientos para la sistematización del Trámite de Documentos en Servicios Generales Barbosa.</p>	<p>VI: Identificación de patrones de diseño</p> <p>VD: Proceso de actualización de nuevos requerimientos.</p>	<p>Tiempo utilizado en la etapa de Diseño sin patrones y con patrones.</p> <p>Horas /hombres.</p> <p>Líneas de código</p> <p>Reutilización de clases y códigos.</p>

ANEXO 2: PROTOTIPOS NO FUNCIONALES

PLATAFORMA DE ESCRITORIO

LOGIN



The image shows a login form titled 'LOGIN' for the 'SISTEMA TRAMITE DOCUMENTARIO BARBOSA E.I.R.L.'. On the left, there is an icon of a person with a yellow key. To the right of the icon are two input fields: 'Usuario:' and 'Contraseña:'. Below these fields is a button labeled 'INGRESAR'.

a) Actores

- Representante doc.
- Mesa de Partes
- Representante Oficina

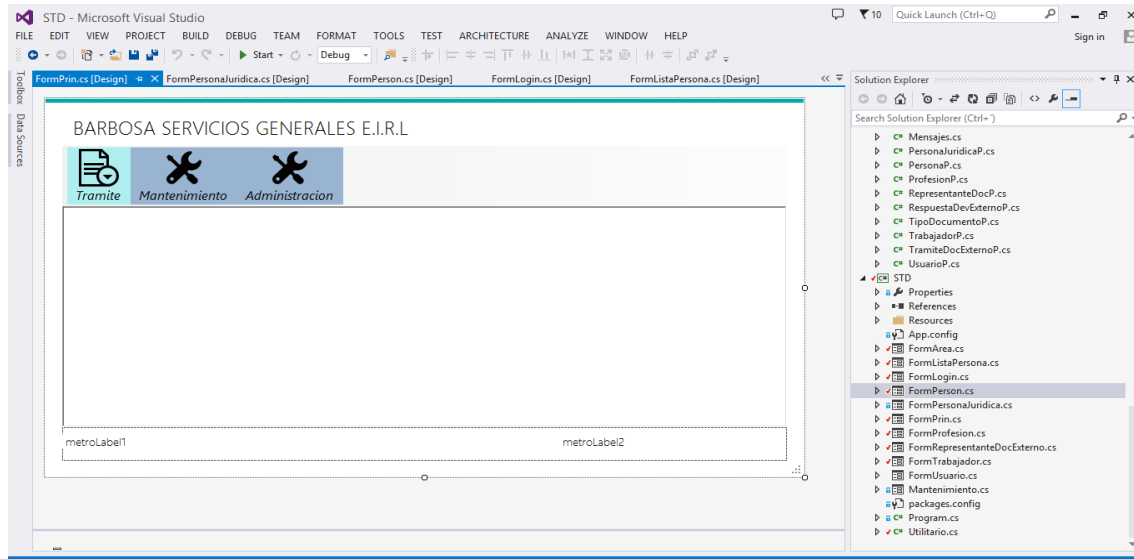
b) Breve Descripción

Permite registrar y validar los datos del usuario en el sistema y realizar el proceso según sea su función.

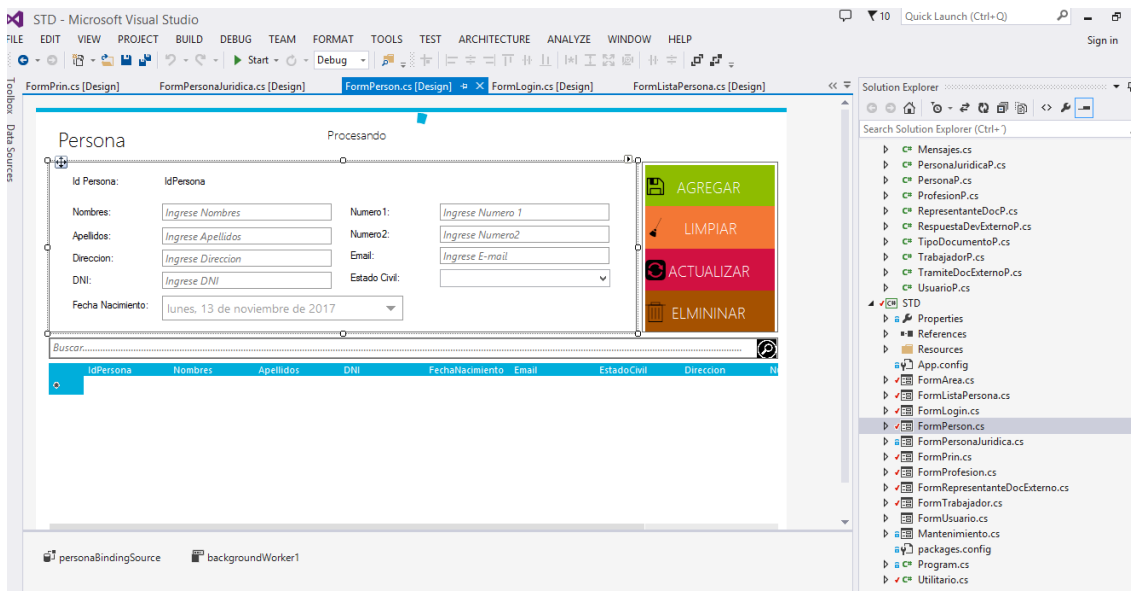
c) Flujo Básico de Eventos

- El En el Sistema muestra el interfaz, los campos de Usuario y password así como los Boton “Ingresar”.
- El usuario ingresa sus datos
- El usuario en el sistema registra los datos solicitados, luego selecciona el botón “Ingresar”.
- El sistema valida los datos ingresados del usuario.
- El sistema graba y valida datos del usuario
- El usuario ingresa a la opción según sea el requerido, realiza dicha función y guarda datos.
- El usuario para finalizar dicha función y desea cerrar sesión, realiza clic en la opción salir.

PAGINA PRINCIPAL DEL SISTEMA



MANTENIMIENTO DE PERSONA

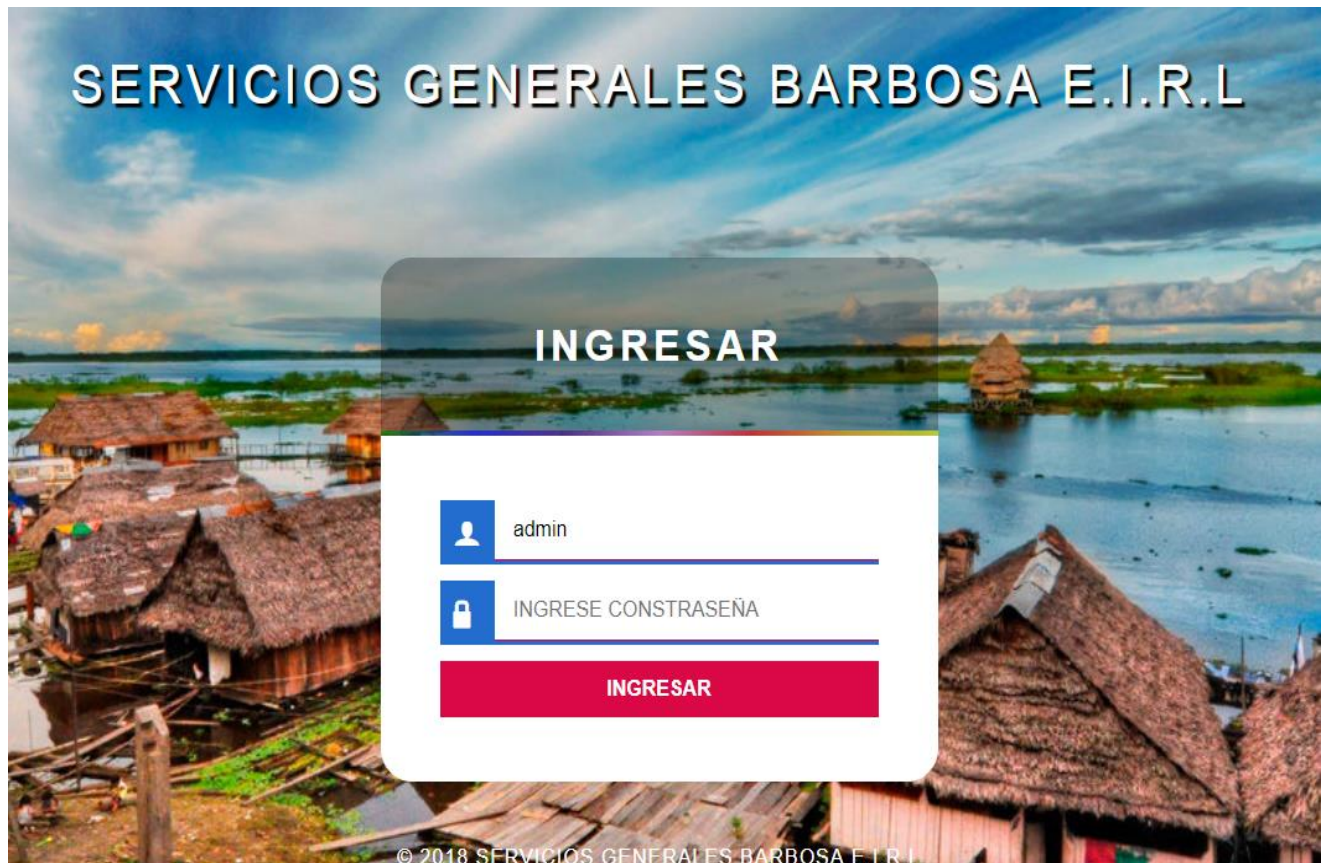


PLATAFORMA WEB



Portal Web





Vista Principal del Sistema de Tramite web

The screenshot shows a web browser window with the URL `192.168.1.33:45459/Home/Principal`. The browser's address bar and tabs are visible at the top. The page has a blue header with the text "SISTEMA TRAMITES" on the left and a user profile icon on the right. Below the header, the main content area is titled "Principal" and contains the text "© 2018 - SERVICIOS GENERALES BARBOSA E.I.R.L.". On the left side, there is a sidebar menu with the following items: "Panel" (with a monitor icon), "INICIO" (with a grid icon), "ADMINISTRACION" (with a clock icon), and "MANTENIMIENTO" (with a key icon). The user profile information in the header indicates that the user is an administrator and does not have any workers assigned.

SISTEMA TRAMITES

EL USUARIO NO TIENE
RELACION CON NINGUN
TRABAJADOR
Administrador

Panel

INICIO

ADMINISTRACION

MANTENIMIENTO

Principal

© 2018 - SERVICIOS GENERALES BARBOSA E.I.R.L

ANEXO 3: DICCIONARIO DE DATOS

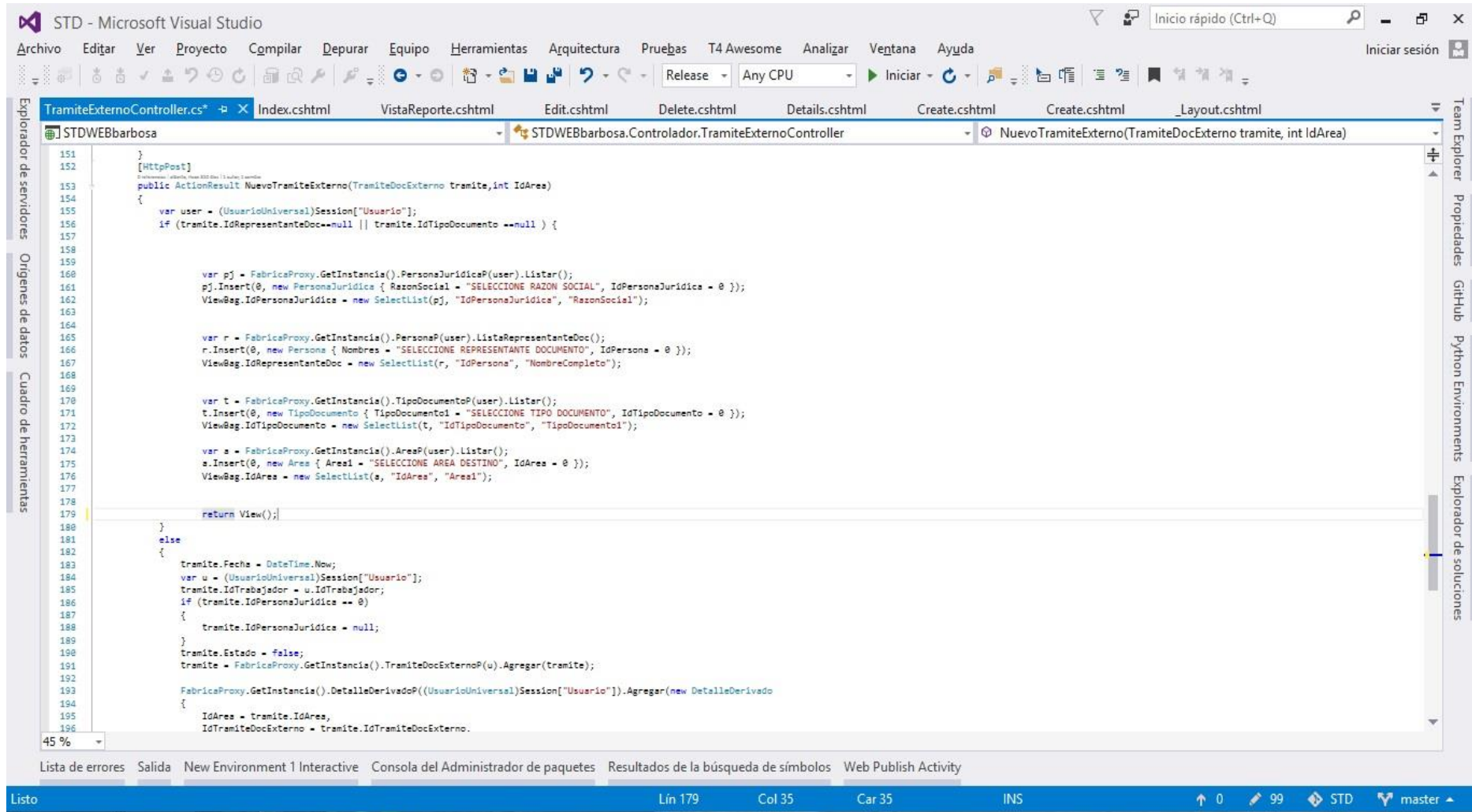
object_id	table	column	type	Precision	max_length	Permite Nulls	Es Autonumerico	Descripcion	
1	610101214	Area	IdArea	int	NULL	4	NO	SI	NULL
2	610101214	Area	Area	var...	NULL	100	SI	NO	NULL
3	610101214	Area	IdTra...	int	NULL	4	NO	NO	NULL
4	11221030...	DetalleAdj...	IdDet...	int	NULL	4	NO	SI	NULL
5	11221030...	DetalleAdj...	IdTra...	int	NULL	4	NO	NO	NULL
6	11221030...	DetalleAdj...	Detalle	var...	NULL	1000	SI	NO	NULL
7	11221030...	DetalleAdj...	Adjun...	var...	NULL	10	SI	NO	NULL
8	606625204	DetalleAdj...	IdDet...	int	NULL	4	NO	SI	NULL
9	606625204	DetalleAdj...	IdTra...	int	NULL	4	SI	NO	NULL
10	606625204	DetalleAdj...	Detalle	var...	NULL	50	SI	NO	NULL
11	606625204	DetalleAdj...	Adjun...	var...	NULL	1000	SI	NO	NULL
12	885578193	DetalleDeri...	IdDet...	int	NULL	4	NO	SI	NULL
13	885578193	DetalleDeri...	IdTra...	int	NULL	4	SI	NO	NULL
14	885578193	DetalleDeri...	IdArea	int	NULL	4	SI	NO	NULL
15	885578193	DetalleDeri...	Fecha...	dat...	NULL	8	SI	NO	NULL
16	885578193	DetalleDeri...	IdRes...	int	NULL	4	SI	NO	NULL
17	654625375	DetalleDeri...	IdDet...	int	NULL	4	NO	SI	NULL
18	654625375	DetalleDeri...	IdTra...	int	NULL	4	SI	NO	NULL
19	654625375	DetalleDeri...	IdArea	int	NULL	4	SI	NO	NULL
20	654625375	DetalleDeri...	Fecha	dat...	NULL	8	SI	NO	NULL
21	654625375	DetalleDeri...	IdRes...	int	NULL	4	SI	NO	NULL
22	398624463	MensajePr...	IdMen...	int	NULL	4	NO	SI	NULL
23	398624463	MensajePr...	IdUsu...	int	NULL	4	SI	NO	NULL
24	398624463	MensajePr...	IdUsa...	int	NULL	4	SI	NO	NULL
25	398624463	MensajePr...	Mens...	var...	NULL	2000	SI	NO	NULL

26	398624463	MensajePr...	Fecha	dat...	NULL	8	SI	NO	NULL
27	110623437	MensajeTo...	IdMen...	int	NULL	4	NO	SI	NULL
28	110623437	MensajeTo...	IdUsu...	int	NULL	4	SI	NO	NULL
29	110623437	MensajeTo...	Mens...	var...	NULL	1000	SI	NO	NULL
30	110623437	MensajeTo...	Fecha	dat...	NULL	8	SI	NO	NULL
31	18099105	Persona	IdPer...	int	NULL	4	NO	SI	NULL
32	18099105	Persona	DNI	var...	NULL	8	SI	NO	NULL
33	18099105	Persona	Nomb...	var...	NULL	100	SI	NO	NULL
34	18099105	Persona	Apelli...	var...	NULL	100	SI	NO	NULL
35	18099105	Persona	Fecha...	date	NULL	3	SI	NO	NULL
36	18099105	Persona	Estad...	var...	NULL	50	SI	NO	NULL
37	18099105	Persona	Direc...	var...	NULL	100	SI	NO	NULL
38	18099105	Persona	Nume...	var...	NULL	50	SI	NO	NULL
39	18099105	Persona	Nume...	var...	NULL	50	SI	NO	NULL
40	18099105	Persona	Email	var...	NULL	50	SI	NO	NULL
41	13975800...	PersonaJu...	IdPer...	int	NULL	4	NO	SI	NULL
42	13975800...	PersonaJu...	Razo...	var...	NULL	50	SI	NO	NULL
43	13975800...	PersonaJu...	Ruc	var...	NULL	10	SI	NO	NULL
44	13975800...	PersonaJu...	Telefo...	var...	NULL	50	SI	NO	NULL
45	13975800...	PersonaJu...	Email	var...	NULL	50	SI	NO	NULL
46	501576825	Profesion	IdProf...	int	NULL	4	NO	SI	NULL

47	501576825	Profesion	Profe...	var...	NULL	50	SI	NO	NULL
48	501576825	Profesion	Abrev...	var...	NULL	10	SI	NO	NULL
49	12055793...	Represent...	IdRep...	int	NULL	4	NO	NO	NULL
50	510624862	Respuesta...	IdDet...	int	NULL	4	NO	NO	NULL
51	510624862	Respuesta...	Resp...	var...	NULL	2000	SI	NO	NULL
52	510624862	Respuesta...	Obser...	var...	NULL	2000	SI	NO	NULL
53	510624862	Respuesta...	Adjun...	var...	NULL	200	SI	NO	NULL
54	510624862	Respuesta...	Fecha	dat...	NULL	8	SI	NO	NULL
55	10741028...	Respuesta...	IdDet...	int	NULL	4	NO	NO	NULL
56	10741028...	Respuesta...	Resp...	var...	NULL	2000	SI	NO	NULL
57	10741028...	Respuesta...	Obser...	var...	NULL	2000	SI	NO	NULL
58	10741028...	Respuesta...	Adjun...	var...	NULL	10	SI	NO	NULL
59	10741028...	Respuesta...	Fecha	dat...	NULL	8	SI	NO	NULL
60	10261026...	Rol	IdRol	int	NULL	4	NO	SI	NULL
61	10261026...	Rol	Rol	var...	NULL	50	NO	NO	NULL
62	13655799...	TipoDocu...	IdTipo...	int	NULL	4	NO	SI	NULL
63	13655799...	TipoDocu...	TipoD...	var...	NULL	20	SI	NO	NULL
64	13655799...	TipoDocu...	Descr...	var...	NULL	200	SI	NO	NULL
65	19381059...	Trabajador	IdTra...	int	NULL	4	NO	NO	NULL
66	19381059...	Trabajador	Estado	bit	NULL	1	NO	NO	NULL
67	19381059...	Trabajador	IdProf...	int	NULL	4	NO	NO	NULL
68	402100473	TramiteDo...	IdTra...	int	NULL	4	NO	SI	NULL
69	402100473	TramiteDo...	Num...	var...	NULL	20	SI	NO	NULL
70	402100473	TramiteDo...	IdPer...	int	NULL	4	SI	NO	NULL
71	402100473	TramiteDo...	IdRep...	int	NULL	4	SI	NO	NULL
72	402100473	TramiteDo...	IdTipo...	int	NULL	4	SI	NO	NULL
73	402100473	TramiteDo...	Asunto	var...	NULL	1000	SI	NO	NULL
74	402100473	TramiteDo...	NumF...	int	NULL	4	SI	NO	NULL
75	402100473	TramiteDo...	Estado	bit	NULL	1	SI	NO	NULL
76	402100473	TramiteDo...	Fecha	dat...	NULL	8	SI	NO	NULL
77	402100473	TramiteDo...	IdTra...	int	NULL	4	SI	NO	NULL
78	238623893	TramiteInt...	IdTra...	int	NULL	4	NO	SI	NULL
79	238623893	TramiteInt...	IdOfic...	int	NULL	4	SI	NO	NULL
80	238623893	TramiteInt...	Fecha	date	NULL	3	SI	NO	NULL
81	238623893	TramiteInt...	IdTra...	int	NULL	4	SI	NO	NULL
82	238623893	TramiteInt...	Asunto	var...	NULL	200	SI	NO	NULL
83	238623893	TramiteInt...	Estado	bit	NULL	1	SI	NO	NULL
84	238623893	TramiteInt...	IdTipo...	int	NULL	4	SI	NO	NULL
85	318624178	Usuario	IdUsu...	int	NULL	4	NO	SI	NULL
86	318624178	Usuario	Usuar...	var...	NULL	50	NO	NO	NULL
87	318624178	Usuario	Contr...	var...	NULL	1000	NO	NO	NULL
88	318624178	Usuario	Estado	bit	NULL	1	SI	NO	NULL
89	318624178	Usuario	IdTra...	int	NULL	4	SI	NO	NULL
90	318624178	Usuario	IdRol	int	NULL	4	NO	NO	NULL
91	318624178	Usuario	IdChat	var...	NULL	2000	SI	NO	NULL

NULL	NULL	NULL
NULL	NULL	NULL
FK_Repr...	Persona	IdPersona
FK_Resp...	DetalleDerivadoInte...	IdDetalleDerivadoInte...
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
FK_Resp...	DetalleDerivado	IdDetalleDerivado
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
FK_Traba...	Persona	IdPersona
NULL	NULL	NULL
FK_Traba...	Profesion	IdProfesion
NULL	NULL	NULL
NULL	NULL	NULL
FK_Trami...	PersonaJuridica	IdPersonaJuridica
FK_Trami...	RepresentanteDoc	IdRepresentanteDoc
FK_Trami...	TipoDocumento	IdTipoDocumento
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
FK_Trami...	Trabajador	IdTrabajador
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
FK_Trami...	Trabajador	IdTrabajador
NULL	NULL	NULL
NULL	NULL	NULL
FK_Trami...	TipoDocumento	IdTipoDocumento
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
NULL	NULL	NULL
FK_Usua...	Trabajador	IdTrabajador
FK_Usua...	Rol	IdRol
NULL	NULL	NULL

ANEXO 4: CODIGO DE PROGRAMA

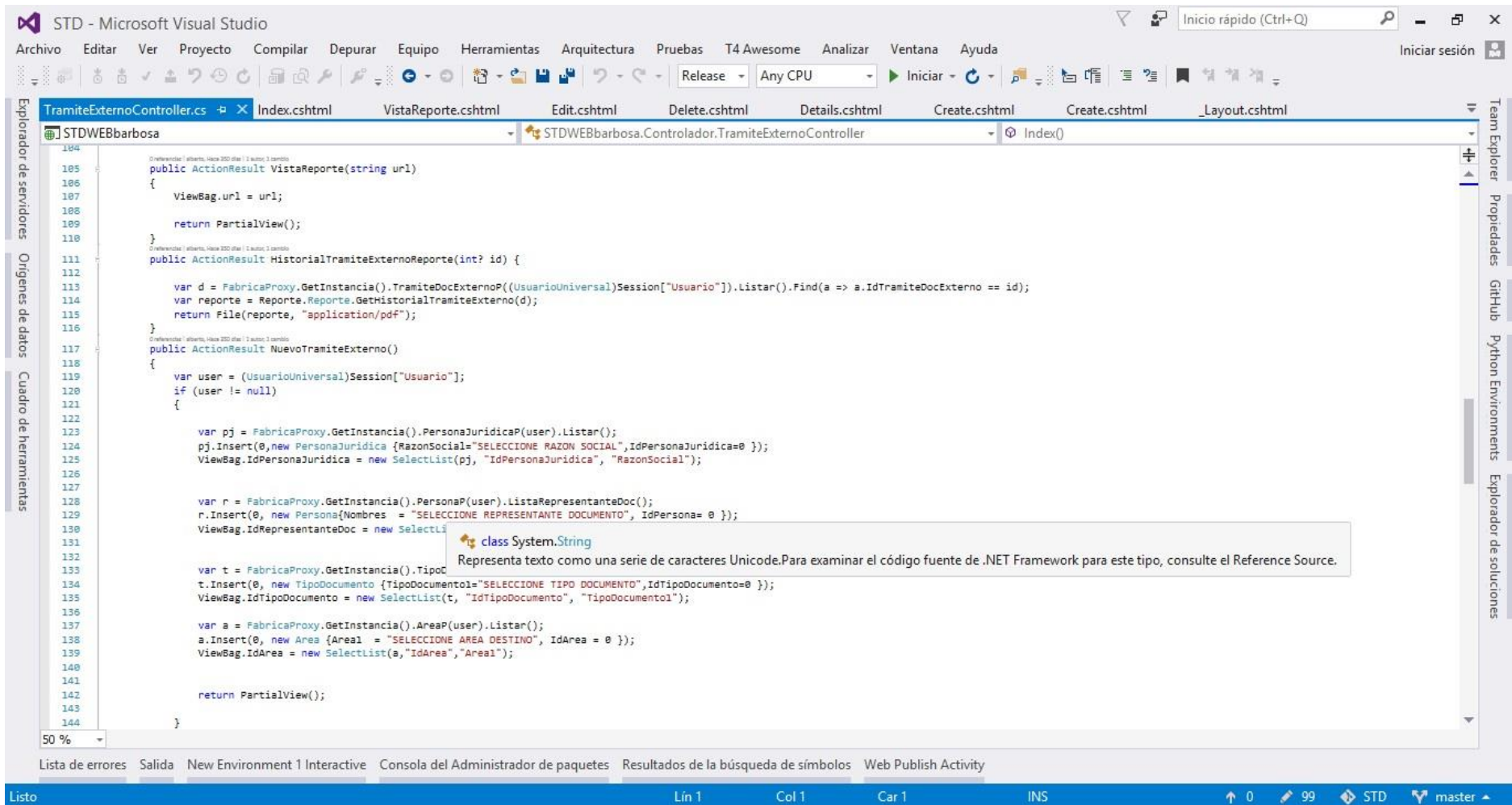


The image shows a screenshot of the Microsoft Visual Studio IDE. The main window displays the source code for the `NuevoTramiteExterno` method in `TramiteExternoController.cs`. The code is written in C# and includes several database operations using `FabricaProxy` to interact with `PersonaJuridica`, `Persona`, `TipoDocumento`, and `Area` entities. It also includes logic for setting the `Tramite` object's properties and saving it to the database.

```
151     }
152     [HttpPost]
153     public ActionResult NuevoTramiteExterno(TramiteDocExterno tramite, int IdArea)
154     {
155         var user = (UsuarioUniversal)Session["Usuario"];
156         if (tramite.IdRepresentanteDoc == null || tramite.IdTipoDocumento == null) {
157
158
159
160             var pj = FabricaProxy.GetInstancia().PersonaJuridicaP(user).Listar();
161             pj.Insert(0, new PersonaJuridica { RazonSocial = "SELECCIONE RAZON SOCIAL", IdPersonaJuridica = 0 });
162             ViewBag.IdPersonaJuridica = new SelectList(pj, "IdPersonaJuridica", "RazonSocial");
163
164
165             var r = FabricaProxy.GetInstancia().PersonaP(user).ListaRepresentanteDoc();
166             r.Insert(0, new Persona { Nombres = "SELECCIONE REPRESENTANTE DOCUMENTO", IdPersona = 0 });
167             ViewBag.IdRepresentanteDoc = new SelectList(r, "IdPersona", "NombreCompleto");
168
169
170             var t = FabricaProxy.GetInstancia().TipoDocumentoP(user).Listar();
171             t.Insert(0, new TipoDocumento { TipoDocumento1 = "SELECCIONE TIPO DOCUMENTO", IdTipoDocumento = 0 });
172             ViewBag.IdTipoDocumento = new SelectList(t, "IdTipoDocumento", "TipoDocumento1");
173
174             var a = FabricaProxy.GetInstancia().AreaP(user).Listar();
175             a.Insert(0, new Area { Area1 = "SELECCIONE AREA DESTINO", IdArea = 0 });
176             ViewBag.IdArea = new SelectList(a, "IdArea", "Area1");
177
178
179             return View();
180         }
181         else
182         {
183             tramite.Fecha = DateTime.Now;
184             var u = (UsuarioUniversal)Session["Usuario"];
185             tramite.IdTrabajador = u.IdTrabajador;
186             if (tramite.IdPersonaJuridica == 0)
187             {
188                 tramite.IdPersonaJuridica = null;
189             }
190             tramite.Estado = false;
191             tramite = FabricaProxy.GetInstancia().TramiteDocExternoP(u).Agregar(tramite);
192
193             FabricaProxy.GetInstancia().DetalleDerivadoP((UsuarioUniversal)Session["Usuario"]).Agregar(new DetalleDerivado
194             {
195                 IdArea = tramite.IdArea,
196                 IdTramiteDocExterno = tramite.IdTramiteDocExterno.

```

The status bar at the bottom indicates the current position: `Lín 179 Col 35 Car 35`. The bottom right corner shows the solution explorer with `STD` and `master` branches.



STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Release Any CPU Iniciar

Index.cshhtml VistaReporte.cshhtml Edit.cshhtml Delete.cshhtml Details.cshhtml Create.cshhtml Create.cshhtml _Layout.cshhtml VistaReporte.cshhtml chathub.cs

```
1 @model IEnumerable<Entidades.TramiteDocExterno>
2
3 @{
4     ViewBag.Title = "Index";
5 }
6
7 <h2>LISTA DE TRAMITES</h2>
8 <br />
9 <a href="@Url.Action("NuevoTramiteExterno")" data-modal="" class="mdl-button mdl-js-button mdl-button--fab mdl-js-ripple-effect botonColor botonCrear" data-modal="" data-toggle="modal" data-target=".bd-example-modal-lg">spa
10 <br />
11 <br />
12 <div class="tablaScroll">
13 <table class="mdl-data-table mdl-js-data-table">
14     <thead>
15         <tr>
16             <th>
17                 @Html.DisplayName("N° CARGO")
18             </th>
19             <th>
20                 @Html.DisplayName("REALIZADO POR")
21             </th>
22             <th>
23                 @Html.DisplayName("REPRESETANTE DOCUMENTO")
24             </th>
25             <th>
26                 @Html.DisplayName("FUNCIONES")
27             </th>
28         </tr>
29     </thead>
30     <tbody>
31         @foreach (var item in Model)
32         {
33             <tr>
34                 <td class="mdl-data-table__cell--non-numeric">
35                     @Html.DisplayFor(modelItem => item.NumCargo)
36                 </td>
37                 <td class="mdl-data-table__cell--non-numeric">
38                     @Html.DisplayFor(modelItem => item.Trabajador.Persona.NombreCompleto)
39                 </td>
40                 <td class="mdl-data-table__cell--non-numeric">
41                     @Html.DisplayFor(modelItem => item.Trabajador.Persona.NombreCompleto)
42                 </td>
43                 <td class="mdl-data-table__cell--non-numeric">
44                     @Html.DisplayFor(modelItem => item.Trabajador.Persona.NombreCompleto)
45                 </td>
46             </tr>
47         }
48     </tbody>
49 </table>
50 </div>
```

50 % | alberto, Hace 350 días | 2 autores, 3 cambios

Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo | Lín 22 Col 18 Car 18 | INS | 0 | 99 | STD | master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Release Any CPU Iniciar

Explorador de servidores Orígenes de datos Cuadro de herramientas

```
7 <h2>EDITAR DATOS DE LA PERSONA JURIDICA</h2>
8
9 @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14         <hr />
15         @Html.ValidationSummary(true)
16
17
18         <div class="form-group">
19             @Html.LabelFor(model => model.IdPersonaJuridica, new { @class = "control-label col-md-2" })
20             <div class="col-md-10">
21                 @Html.EditorFor(model => model.IdPersonaJuridica, new { htmlAttributes = new { @class = "form-control", @readonly = "readonly" } })
22                 @Html.ValidationMessageFor(model => model.IdPersonaJuridica)
23             </div>
24         </div>
25
26         <div class="form-group">
27             @Html.LabelFor(model => model.RazonSocial, new { @class = "control-label col-md-2" })
28             <div class="col-md-10">
29                 @Html.EditorFor(model => model.RazonSocial, new { htmlAttributes = new { @class = "form-control" } })
30                 @Html.ValidationMessageFor(model => model.RazonSocial)
31             </div>
32         </div>
33
34         <div class="form-group">
35             @Html.LabelFor(model => model.Ruc, new { @class = "control-label col-md-2" })
36             <div class="col-md-10">
37                 @Html.EditorFor(model => model.Ruc, new { htmlAttributes = new { @class = "form-control" } })
38                 @Html.ValidationMessageFor(model => model.Ruc)
39             </div>
40         </div>
41
42         <div class="form-group">
43             @Html.LabelFor(model => model.Telefono, new { @class = "control-label col-md-2" })
44             <div class="col-md-10">
45                 @Html.EditorFor(model => model.Telefono, new { htmlAttributes = new { @class = "form-control" } })
46                 @Html.ValidationMessageFor(model => model.Telefono)
47             </div>
48         </div>
49
50         <div class="form-group">
```

50 % | alberto, Hace 350 días | 1 autor, 1 cambio

Lista de errores Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 1 Col 1 Car 1 INS ↑ 0 99 STD master

Team Explorer Propiedades GitHub Python Environments Explorador de soluciones

STG - Microsoft Visual Studio

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Release Any CPU Iniciar

Team Explorer Propiedades GitHub Python Environments Explorador de soluciones

```
7 <h2>CREAR NUEVA PERSONA JURIDICA</h2>
8
9
10 @using (Html.BeginForm())
11 {
12     @Html.AntiForgeryToken()
13
14     <div class="form-horizontal">
15         <hr />
16         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
17
18         <div class="form-group">
19             @Html.LabelFor(model => model.RazonSocial, htmlAttributes: new { @class = "control-label col-md-2" })
20             <div class="col-md-10">
21                 @Html.EditorFor(model => model.RazonSocial, new { htmlAttributes = new { @class = "form-control" } })
22                 @Html.ValidationMessageFor(model => model.RazonSocial, "", new { @class = "text-danger" })
23             </div>
24         </div>
25
26         <div class="form-group">
27             @Html.LabelFor(model => model.Ruc, htmlAttributes: new { @class = "control-label col-md-2" })
28             <div class="col-md-10">
29                 @Html.EditorFor(model => model.Ruc, new { htmlAttributes = new { @class = "form-control" } })
30                 @Html.ValidationMessageFor(model => model.Ruc, "", new { @class = "text-danger" })
31             </div>
32         </div>
33
34         <div class="form-group">
35             @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })
36             <div class="col-md-10">
37                 @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })
38                 @Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })
39             </div>
40         </div>
41
42         <div class="form-group">
43             @Html.LabelFor(model => model.Telefono, htmlAttributes: new { @class = "control-label col-md-2" })
44             <div class="col-md-10">
45                 @Html.EditorFor(model => model.Telefono, new { htmlAttributes = new { @class = "form-control" } })
46                 @Html.ValidationMessageFor(model => model.Telefono, "", new { @class = "text-danger" })
47             </div>
48         </div>
49
50         <div class="form-group">
```

50 % | alberto, Hace 350 días | 2 autores, 3 cambios

Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 13 Col 5 Car 5 INS ↑ 0 ↻ 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda

Release Any CPU Iniciar

Layout.cshtml VistaReporte.cshtml chathub.cs NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño]

```

540     '<div class="right-chat">'+
541     '<span class="spanNombre">'+usa.nombrequerorigen+'</span>'+
542     '<p>'+
543     '    usa.mensaje +
544     '</p>'+
545     '</div> </li>';
546     $('#mensajes'+iduserdestino).append(is);
547     }
548
549     });
550     $('#cont'+iduserdestino).scrollTop("9999");
551     };
552
553     chatHub.client.llenarchatprivado=function(mensprivados,iduserdestino,nombre){
554
555
556     var idu="@id";
557     $('#mensajes'+iduserdestino).empty();
558     mensprivados.forEach(function (usa) {
559     usa.iduserorigen
560     if(idu==usa.iduserorigen){
561     var de='<li>'+
562     '<div class="left-chat">'+
563     '<span class="spanNombre">'+@nombre+'</span>'+
564     '<p>'+
565     '    usa.mensaje +
566     '</p>'+
567     '</div>'+
568     '</li>';
569     $('#mensajes'+iduserdestino).append(de);
570     }
571     else{
572     var is='<li>'+
573     '<div class="right-chat">'+
574     '<span class="spanNombre">'+usa.nombrequerorigen+'</span>'+
575     '<p>'+
576     '    usa.mensaje +
577     '</p>'+
578     '</div> </li>';
579     $('#mensajes'+iduserdestino).append(is);
580     }
581
582
583

```

50 % < Paulo Rengifo, Hace 344 días | 2 autores, 6 cambios

Lista de errores Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lin 74 Col 15 Car 15 INS ↑ 0 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Release Any CPU Iniciar

Layout.cshtml VistaReporte.cshtml chathub.cs NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño]

```
483 chatHub.client.llenarmensajegrupo=function(mensajestodos){
484     $("#mensajes0").empty();
485     var idu="ok";
486     var n=0;
487     mensajestodos.forEach(function (usua) {
488
489         if(n==0){
490             idu=usua.iduser;
491             n=1;
492         }
493         var de='<li>'+
494             '<div class="left-chat">'+
495                 '<span class="spanNombre">'+usua.nombre+'</span>'+
496                 '<p>'+
497                     usua.mensaje+
498                 '</p>'+
499             '</div>'+
500         '</li>';
501
502         var is='<li>'+
503             '<div class="right-chat">'+
504                 '<span class="spanNombre">'+usua.nombre+'</span>'+
505                 '<p>'+
506                     usua.mensaje+
507                 '</p>'+
508             '</div> </li>';
509         if(idu==usua.iduser){
510             $("#mensajes0").append(de);
511             nu=false;
512         }
513         else{
514             $("#mensajes0").append(is);
515             nu=true;
516         }
517     })
518     $("#cont0").scrollTop("9999");
519 };
520 chatHub.client.llenarchatprivado2=function(mensprivados,iduserdestino,nombre){
521
522     $("#mensajes"+iduserdestino).empty();
523     var idu="@id";
524     mensprivados.forEach(function (usa) {
525
526
```

50 % < Paulo Rengifo, Hace 344 días | 2 autores, 6 cambios

Lista de errores Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 74 Col 15 Car 15 INS ↑ 0 ↗ 99 STD ↻ master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Release Any CPU Iniciar

Layout.cshtml VistaReporte.cshtml chathub.cs NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño]

```
411 }
412 chatHub.client.recibimensajegrupo=function(mensajes){
413     $("#mensajes0").empty();
414     var idu="ok";
415     var n=0;
416
417     mensajes.forEach(function (usua) {
418
419         if(n==0){
420             idu=usua.iduser;
421             n=1;
422         }
423         var de='<li>'+
424             '<div class="left-chat">'+
425                 '<span class="spanNombre">'+usua.nombre+'</span>'+
426                 '<p>'+
427                     usua.mensaje+
428                 '</p>'+
429             '</div>'+
430         '</li>';
431
432         var is='<li>'+
433             '<div class="right-chat">'+
434                 '<span class="spanNombre">'+usua.nombre+'</span>'+
435                 '<p>'+
436                     usua.mensaje+
437                 '</p>'+
438             '</div> </li>';
439         if(idu==usua.iduser){
440             $("#mensajes0").append(de);
441             nu=false;
442         }
443         else{
444             $("#mensajes0").append(is);
445             nu=true;
446         }
447     })
448     $("#cont0").scrollTop("9999");
449 }
450
451 chatHub.client.llenarmensajes=function(mensajespri,iduserdestino){
452     $("#mensajes"+iduserdestino).empty();
453     var idu="@id";
454     mensajespri.forEach(function (usa) {
```

50 % < Paulo Rengifo, Hace 344 días | 2 autores, 6 cambios >

Lista de errores Salida New Environment 1 Interactiva Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 74 Col 15 Car 15 INS ↑ 0 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Layout.cshtml VistaReporte.cshtml chathub.cs NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño]

```
378     var nombre= ((Entidades.UsuarioUniversal)Session["Usuario"]).NombreCompleto;
379     }
380
381     //funcion cuando se logea
382     chatHub.client.llenarusuario = function (usu2) {
383         rol2=usu2.Rol;
384         idusuario2=usu2.IdUser;
385
386         $("#userid").empty();
387         var templ =
388
389         '<a href="#" class="Notification" id="notifation-unread-1">'+
390         '<div class="Notification-icon"><i class="zmdi zmdi-accounts-alt bg-success"></i></div>'+
391         '<div class="Notification-text">'+
392         '<p>'+
393
394             '<i class="zmdi zmdi-circle"></i>'+
395             '<strong>'+usu2.Nombre+'</strong>'+
396             '<br>'+
397             '<small>'+usu2.Rol+'</small>'+
398         '</p>'+
399         '</div>'+
400         '<div class="mdl-tooltip mdl-tooltip--left" for="notifation-unread-1" data-upgraded="MaterialTooltip" style="top: 85px; margin-top: -15px; left: 936px;">Notification as UnRead</div>'+
401     '</a>';
402     console.log(templ);
403     $("#userid").append(templ);
404     id= usu2.IdUser;
405 }
406 chatHub.client.llenarusuarios = function (UsuariosChat) {
407     actualizarUsuarios(UsuariosChat);
408 }
409 chatHub.client.Actualizarusuarios = function (UsuariosChat) {
410     actualizarUsuarios(UsuariosChat);
411 }
412 chatHub.client.recibimensajegrupo=function(mensajes){
413     $("#mensajes0").empty();
414     var idu="ok";
415     var n=0;
416
417     mensajes.forEach(function (usua) {
418
419         if(n==0){
420             idu=usua.iduser;
421             n=1;
422         }
423     });
424 }
```

50 % Paulo Rengifo, Hace 344 días | 2 autores, 6 cambios

Lista de errores Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 74 Col 15 Car 15 INS ↑ 0 ↗ 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda

Release Any CPU Iniciar

VistaReporte.cshhtml chathub.cs NuevoTramiteExterno.cshhtml Index.cshhtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño] FormArea.cs [Diseño]

```
1 <div class="col-12" style="height:500px" id="example1">
2
3
4     <h2>CARGAVO</h2>
5 </div>
6
7 <script src="../js/pdfobject.js"></script>
8 <script>
9     var options = {
10         fallbackLink: "<p>TU NAVEGADOR NO SOPORTA VISTA DE PDF .POR FAVOR DESCARGA EL PDF: <a href='[url]'>DESCARGAR PDF</a></p>"
11     };
12     PDFObject.embed("@Url.Action("HistorialTramiteExternoReporte","TramiteExterno",new {id=10})", "#example1",options));
13 </script>
14
```

50 % alberto, Hace 350 días | 1 autor, 1 cambio

Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lin 14 Col 1 Car 1 INS 0 99 STD master

STD - Microsoft Visual Studio

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda

chathub.cs + x NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño] FormArea.cs [Diseño]

STDWEBbarbosa STDWEBbarbosa.chathub Conectar(string usuario, string rol)

```

13 namespace STDWEBbarbosa
14 {
15     public class chathub : Hub
16     {
17         static List<Usuario> Usuarios = new List<Usuario>();
18         static List<UsuarioChat> UsuariosChat = new List<UsuarioChat>();
19         static List<MensajesGrupo> mensajesTodos = new List<MensajesGrupo>();
20         public void Conectar(string usuario, string rol)
21         {
22             var usu = new UsuarioUniversal() { Rol = rol, IdUsuario = Convert.ToInt32(usuario) };
23
24             Usuarios = FabricaProxy.GetInstancia().UsuarioP(usu).Listar();
25             mensajesTodos = new List<MensajesGrupo>();
26             var mens = FabricaProxy.GetInstancia().MensajeTodos(usu).Listar();
27             foreach (var item in mens)
28             {
29                 mensajesTodos.Add(new MensajesGrupo()
30                 {
31                     fecha = item.Fecha.Value.ToShortDateString(),
32                     iduser = (int)item.IdUsuario,
33                     mensaje = item.Mensaje,
34                     nombre = item.Usuario.Trabajador.Persona.NombreCompleto,
35                     rol = item.Usuario.Rol.Roll
36                 });
37             }
38
39             foreach (var item in Usuarios)
40             {
41                 string nom="";
42
43                 if (item.Trabajador != null)
44                 {
45                     nom = item.Trabajador.Persona.NombreCompleto;
46                 }
47
48                 if(UsuariosChat.FindAll(df=>df.IdUser==item.IdUsuario).Count==0)
49                 UsuariosChat.Add(new UsuarioChat()
50                 {
51                     Nombre = nom,

```

50 %

Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lín 25 Col 59 Car 59 INS ↑ 0 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño] FormArea.cs [Diseño] HistorialReporte.rdlc [Diseño] Permiso.cs TrabajadorP.cs*

```
4 ViewBag.Title = "Index";
5 }
6
7 <h2>TIPO DE DOCUMENTO</h2>
8
9 <a href="@Url.Action("Create")" data-modal="" class="mdl-button mdl-js-button mdl-button--fab mdl-js-ripple-effect botonColor botonCrear" data-toggle="modal" data-target=".bd-example-modal-lg"><span cla
10 <br />
11
12 <br />
13
14 <div class="tablaScroll">
15 <table class="mdl-data-table mdl-js-data-table">
16 <thead>
17 <tr>
18 <th class="mdl-data-table__cell--non-numeric">
19 @Html.DisplayNameFor(model => model.IdTipoDocumento)
20 </th>
21 <th class="mdl-data-table__cell--non-numeric">
22 @Html.DisplayNameFor(model => model.TipoDocumento1)
23 </th>
24 <th class="mdl-data-table__cell--non-numeric">
25 @Html.DisplayNameFor(model => model.Descripcion)
26 </th>
27 <th>FUNCIONES</th>
28
29 </tr>
30 </thead>
31 <tbody>
32
33 @foreach (var item in Model)
34 {
35 <tr>
36 <td class="mdl-data-table__cell--non-numeric">
37 @Html.DisplayFor(modelItem => item.IdTipoDocumento)
38 </td>
39 <td class="mdl-data-table__cell--non-numeric">
40 @Html.DisplayFor(modelItem => item.TipoDocumento1)
41 </td>
42 <td class="mdl-data-table__cell--non-numeric">
43 @Html.DisplayFor(modelItem => item.Descripcion)
44 </td>
45 <td class="mdl-data-table__cell--non-numeric">
46 <a href="@Url.Action("Details",new {id= item.IdTipoDocumento})" class="iconoVista" data-modal="" data-toggle="modal" data-target=".bd-example-modal-lg"><span class="icon-eye">
47 <a href="@Url.Action("Edit",new {id= item.IdTipoDocumento})" class="iconoEditar" data-modal="" data-toggle="modal" data-target=".bd-example-modal-lg"><span class="icon-pencil">
48
```

50 % | alberto, Hace 350 días | 1 autor, 1 cambio

Listado de errores Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo 0 99 STD master

STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda

Release Any CPU Iniciar

NuevoTramiteExterno.cshtml Index.cshtml ProfesionController.cs Mantenimiento.cs [Diseño] FormProfesion.cs [Diseño] FormArea.cs [Diseño] HistorialReporte.rdlc [Diseño]

```

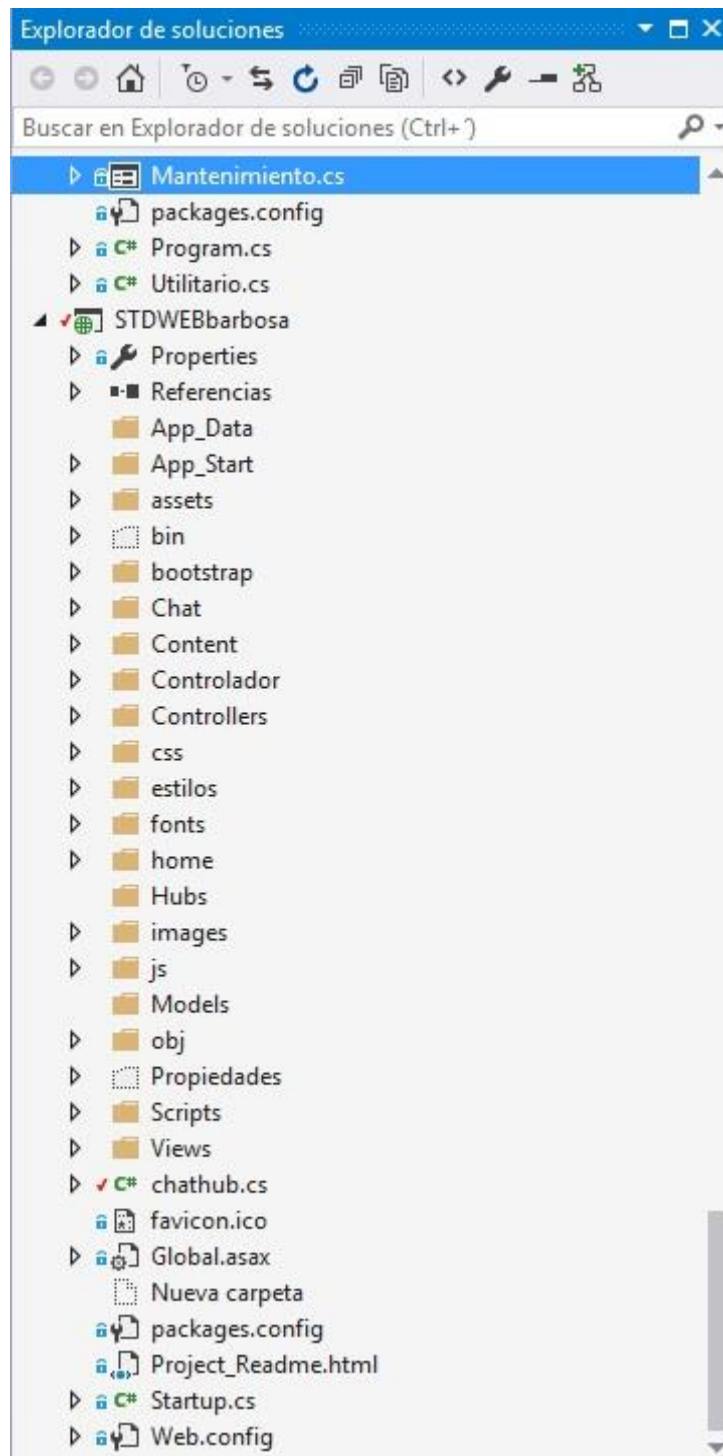
7 <h2>NUEVO TRAMITE DOCUMENTO EXTERNO</h2>
8
9 @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14
15         <hr />
16         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
17
18
19
20
21         <div class="form-group">
22             @Html.Label("RAZON SOCIAL", htmlAttributes: new { @class = "control-label col-12" })
23             <div class="col-md-10">
24                 @Html.DropDownList("IdPersonaJuridica", null, new { @class = "form-control" })
25                 @Html.ValidationMessageFor(model => model.IdPersonaJuridica, "", new { @class = "text-danger" })
26             </div>
27         </div>
28
29         <div class="form-group">
30             @Html.Label("REPRESENTANTE DOCUMENTO", htmlAttributes: new { @class = "control-label col-12" })
31             <div class="col-md-10">
32                 @Html.DropDownList("IdRepresentanteDoc", null, new { @class = "form-control" })
33                 @Html.ValidationMessageFor(model => model.IdRepresentanteDoc, "", new { @class = "text-danger" })
34             </div>
35         </div>
36
37         <div class="form-group">
38             @Html.Label("TIPO DE DOCUMENTO", htmlAttributes: new { @class = "control-label col-12" })
39             <div class="col-md-10">
40                 @Html.DropDownList("IdTipoDocumento", null, new { @class = "form-control" })
41                 @Html.ValidationMessageFor(model => model.IdTipoDocumento, "", new { @class = "text-danger" })
42             </div>
43         </div>
44
45         <div class="form-group">
46             @Html.LabelFor(model => model.Asunto, htmlAttributes: new { @class = "control-label col-md-2" })
47             <div class="col-md-10">
48                 @Html.EditorFor(model => model.Asunto, new { htmlAttributes = new { @class = "form-control" } })
49                 @Html.ValidationMessageFor(model => model.Asunto, "", new { @class = "text-danger" })
50             </div>
51         </div>
52     </div>
53 }

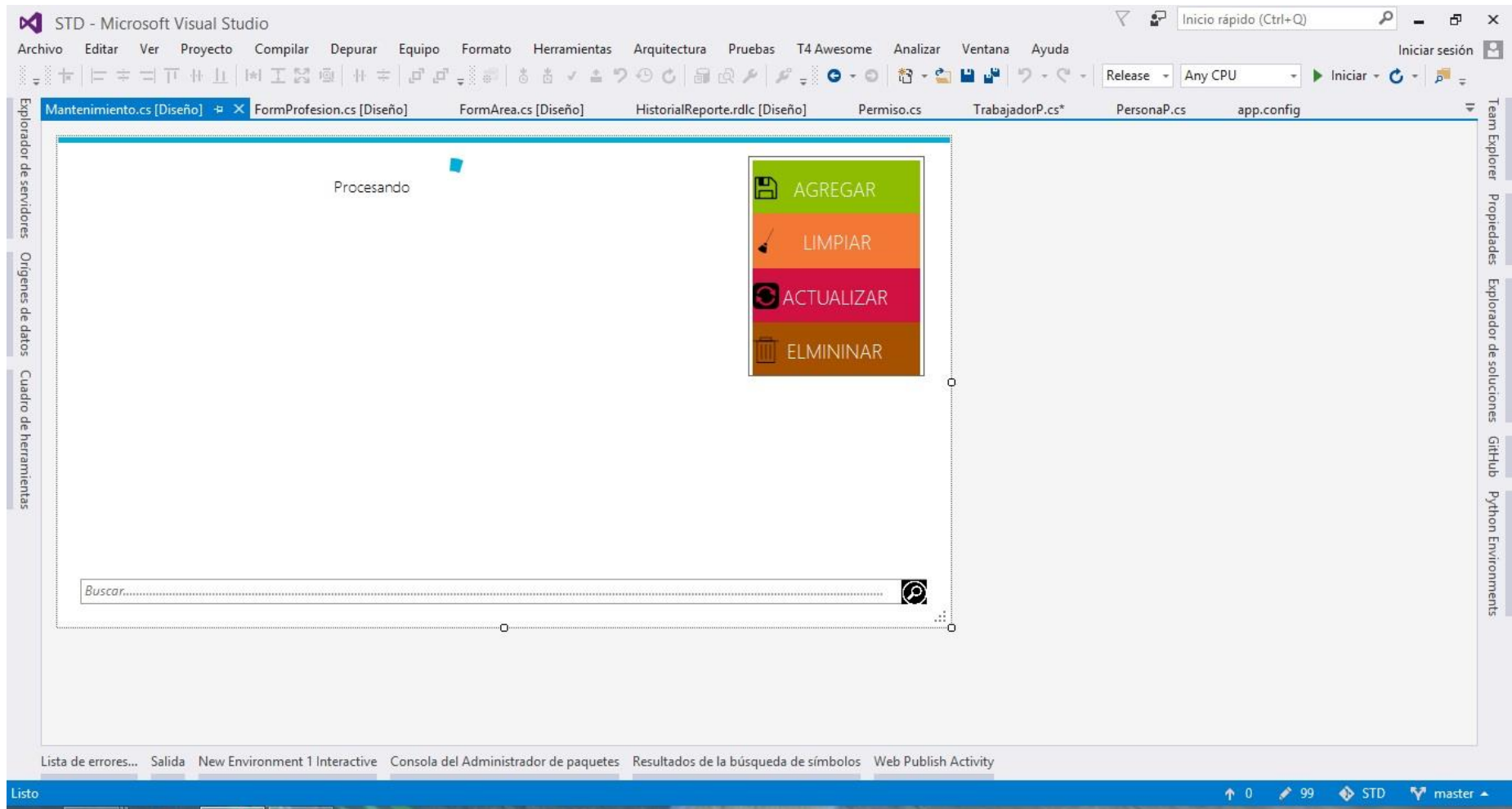
```

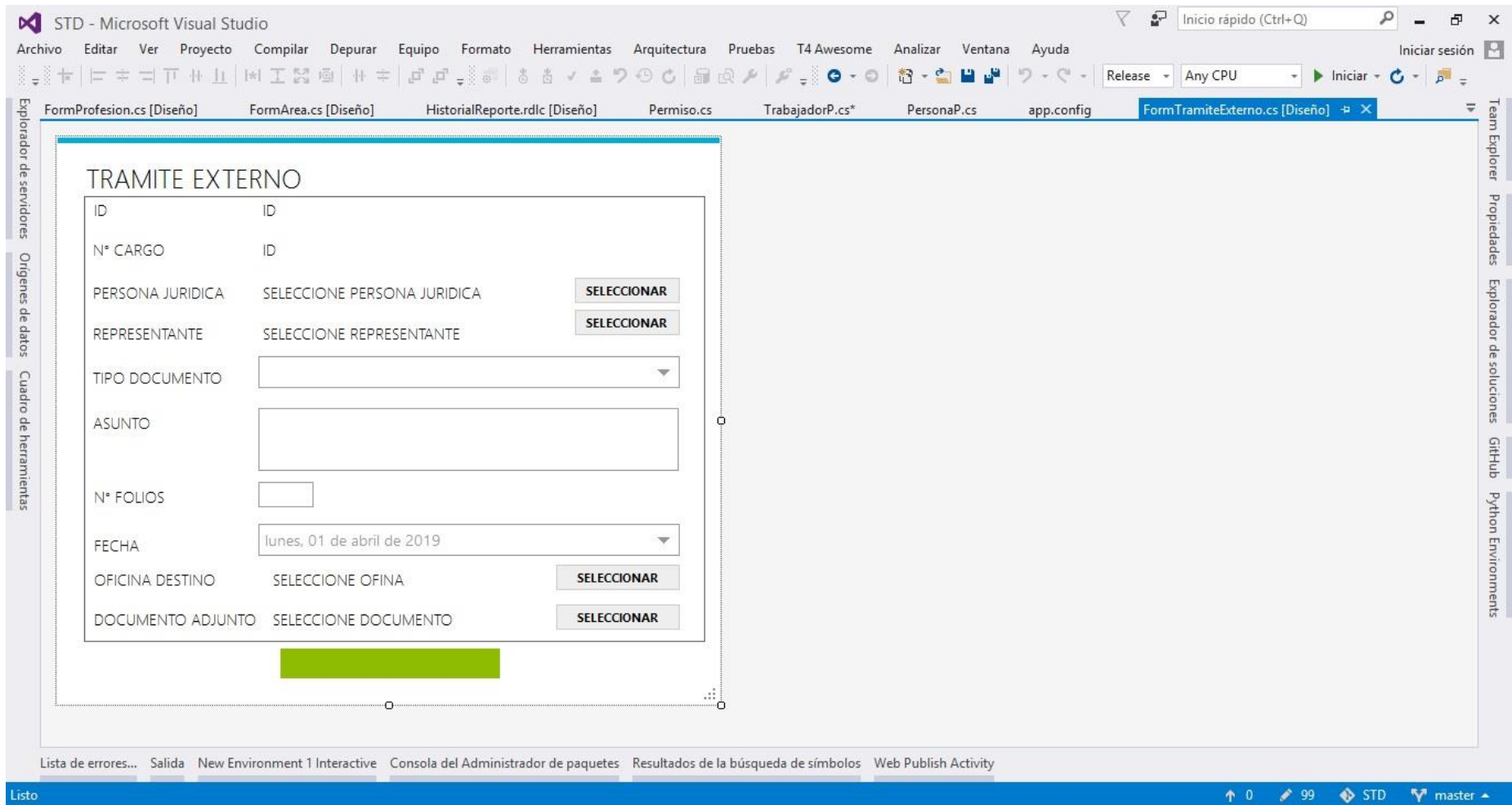
50 % | alberto, Hace 350 días | 2 autores, 3 cambios

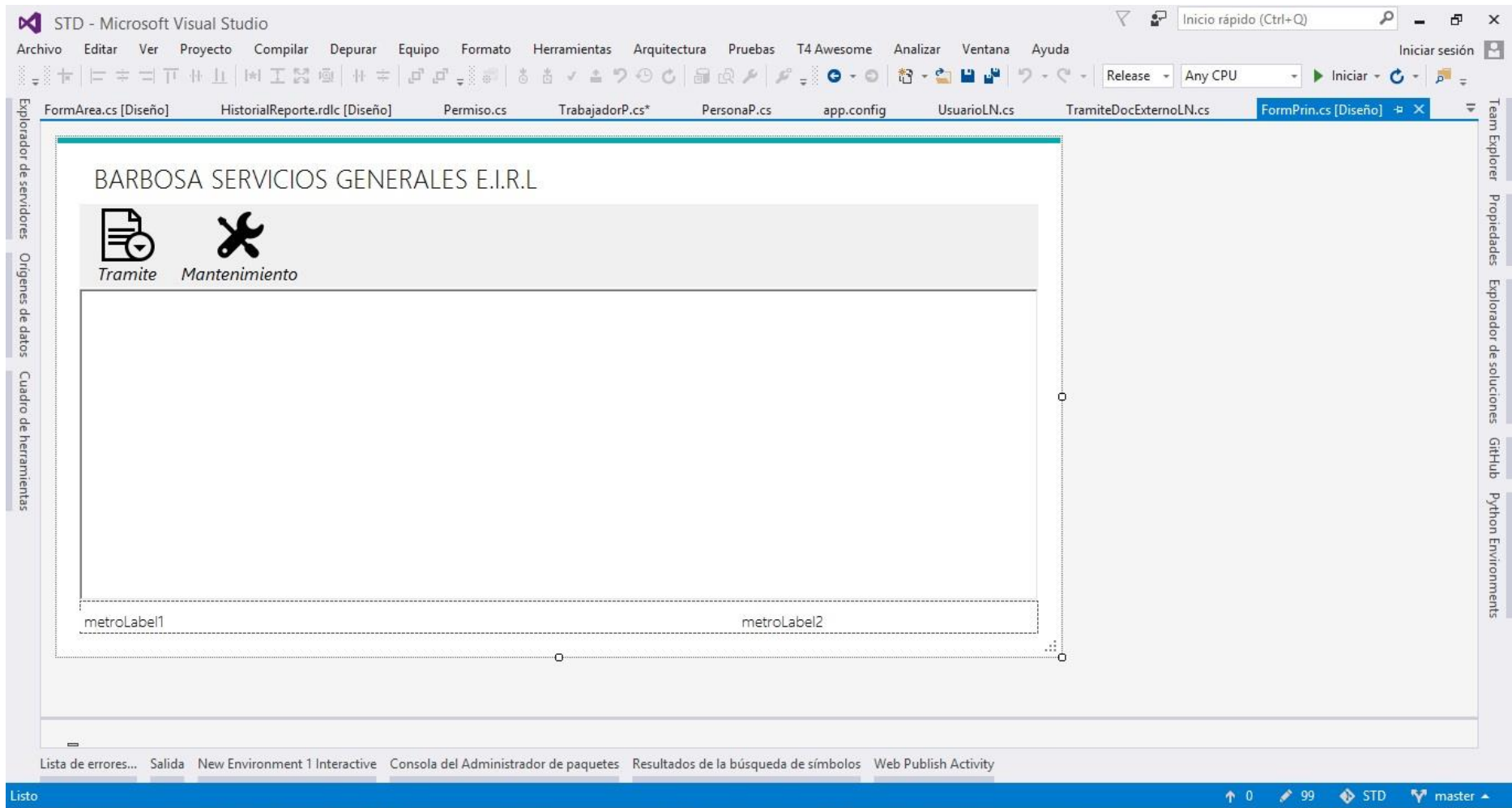
Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo Lin 18 Col 9 Car 9 INS 0 99 STD master










STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Formato Report Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión

HistorialReporte.rdlc [Diseño] Permiso.cs TrabajadorP.cs* PersonaP.cs app.config UsuarioLN.cs TramiteDocExternoLN.cs PersonaLN.cs ProfesionLN.cs

Explorador de servidores Datos del informe Orígenes de datos Cuadro de herramientas Team Explorer Propiedades Explorador de soluciones GitHub Python Environments

SERVICIOS GENERALES BARBOSA E.I.R.L 

HISTORIAL DE TRAMITE [n°cargo]

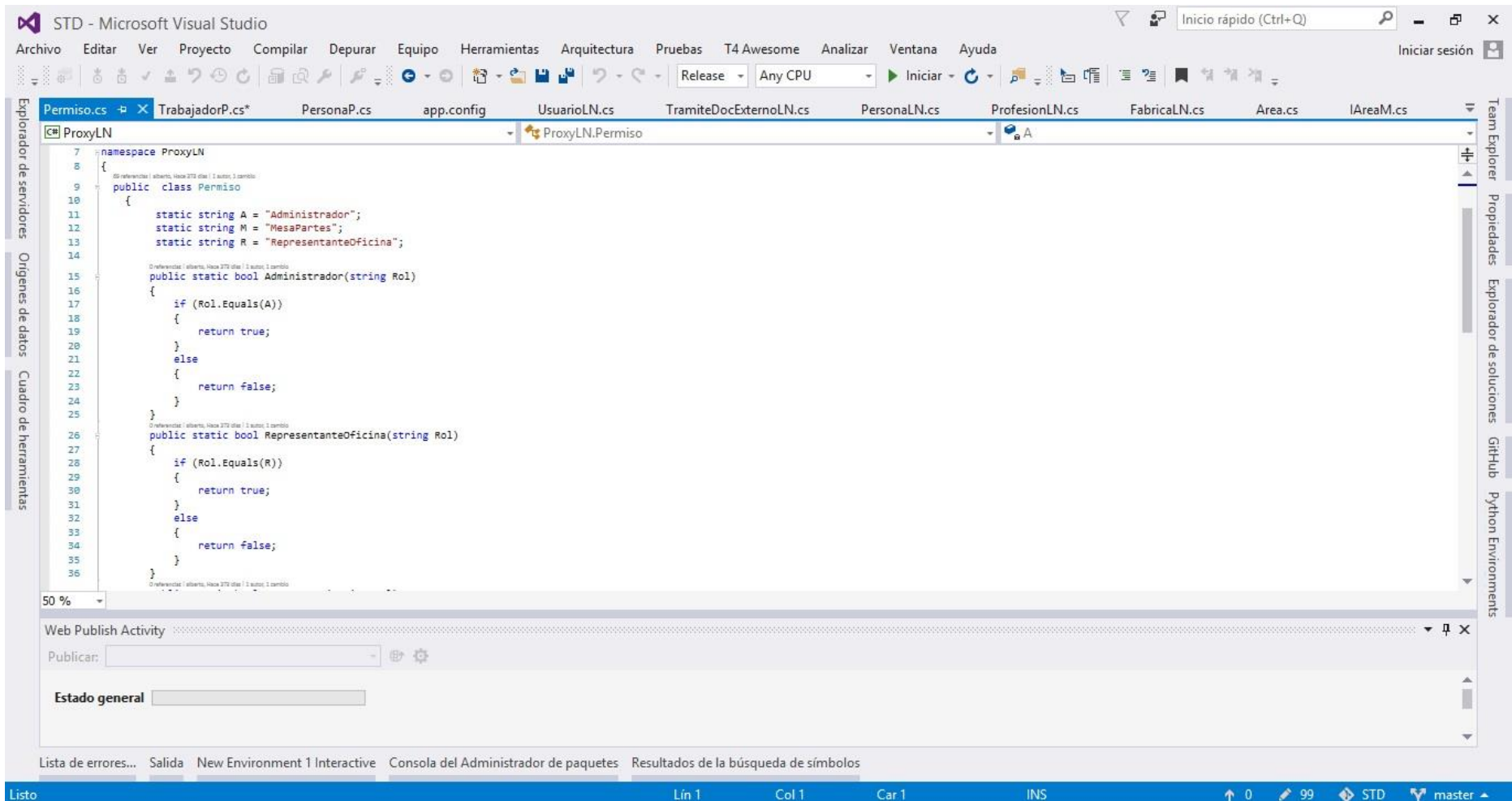
Fecha :	«Expr»
Tramite Realizado Por :	«Expr»
Razon Social:	«Expr»
Tipo Documento :	«Expr»
N° Cargo:	«Expr»
N° Folios :	«Expr»
Representante Documento :	«Expr»
Estado :	«Expr»
Asunto :	«Expr»

Derivado: [Area]	Fecha Derivacion : [Fecha]
Respuesta : [Respuesta]	Fecha : [FechaRespuesta]

Grupos de filas Grupos de columnas

Listado de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos Web Publish Activity

Listo 0 99 STD master



STD - Microsoft Visual Studio

Inicio rápido (Ctrl+Q)

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda Iniciar sesión



TrabajadorP.cs* PersonaP.cs app.config UsuarioLN.cs TramiteDocExternoLN.cs PersonaLN.cs ProfesionLN.cs FabricaLN.cs Area.cs IAreaM.cs App.Config

ProxyLN ProxyLN.Proxy.TrabajadorP TrabajadorP()

```
13 public class TrabajadorP : ITrabajadorM
14
15
16 {
17     UsuarioUniversal usu;
18     string rol;
19     public TrabajadorP(UsuarioUniversal p)
20     {
21         this.usu = p;
22         rol = usu.Rol;
23     }
24     public TrabajadorP()
25     {
26         rol = UsuarioUniversal.GetInstance().Rol;
27     }
28     FabricaLN tra = FabricaLN_instancia();
29
30     public bool Actualizar(Trabajador p)
31     {
32         if (Permiso.Todo(rol))
33         {
34             return tra.Trabajador().Actualizar(p);
35         }
36         else
37         {
38             Mensajes.denegacion(UsuarioUniversal.GetInstance().NombreCompleto);
39         }
40         return false;
41     }
42 }
```

50 %

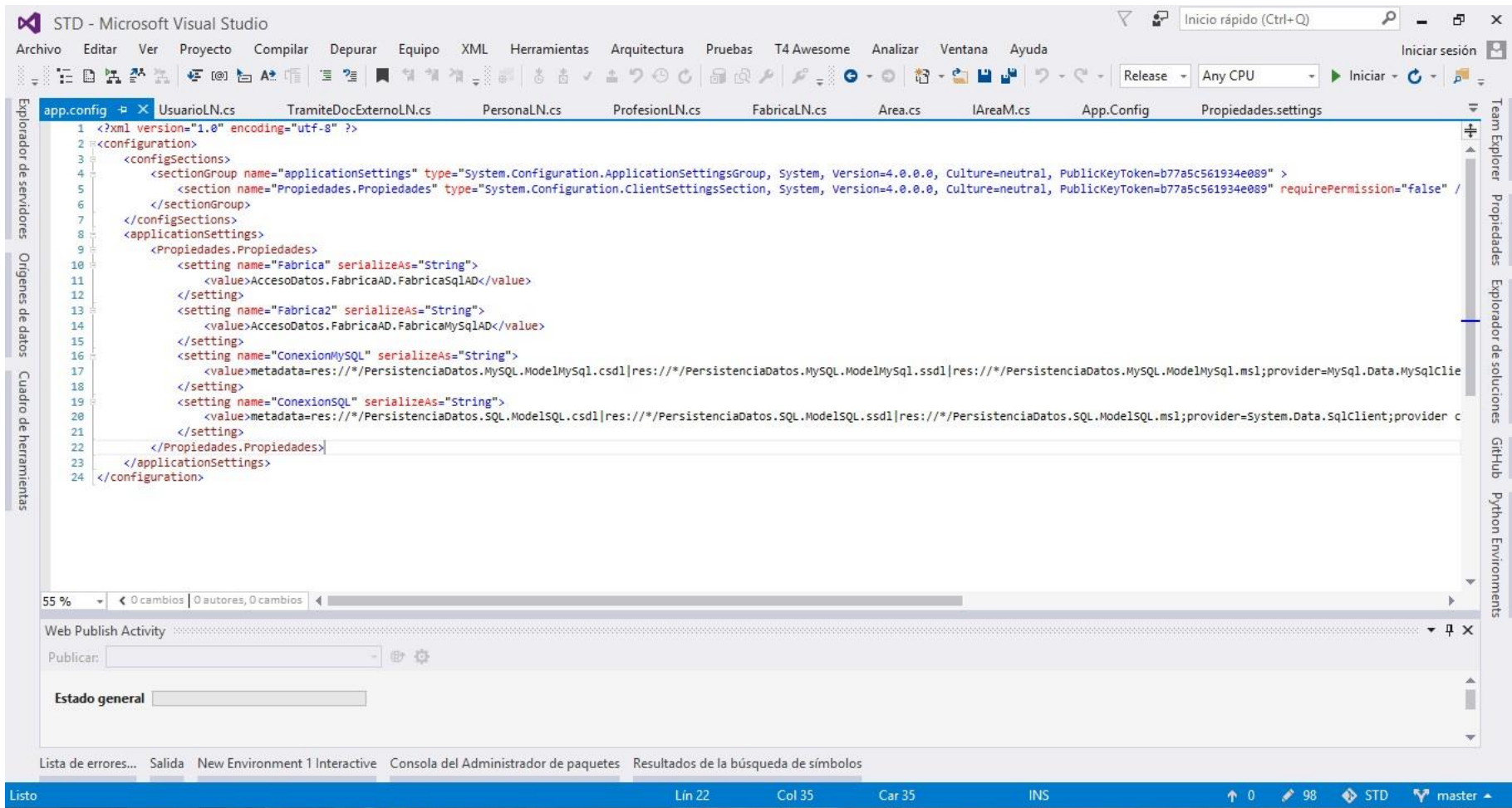
Web Publish Activity

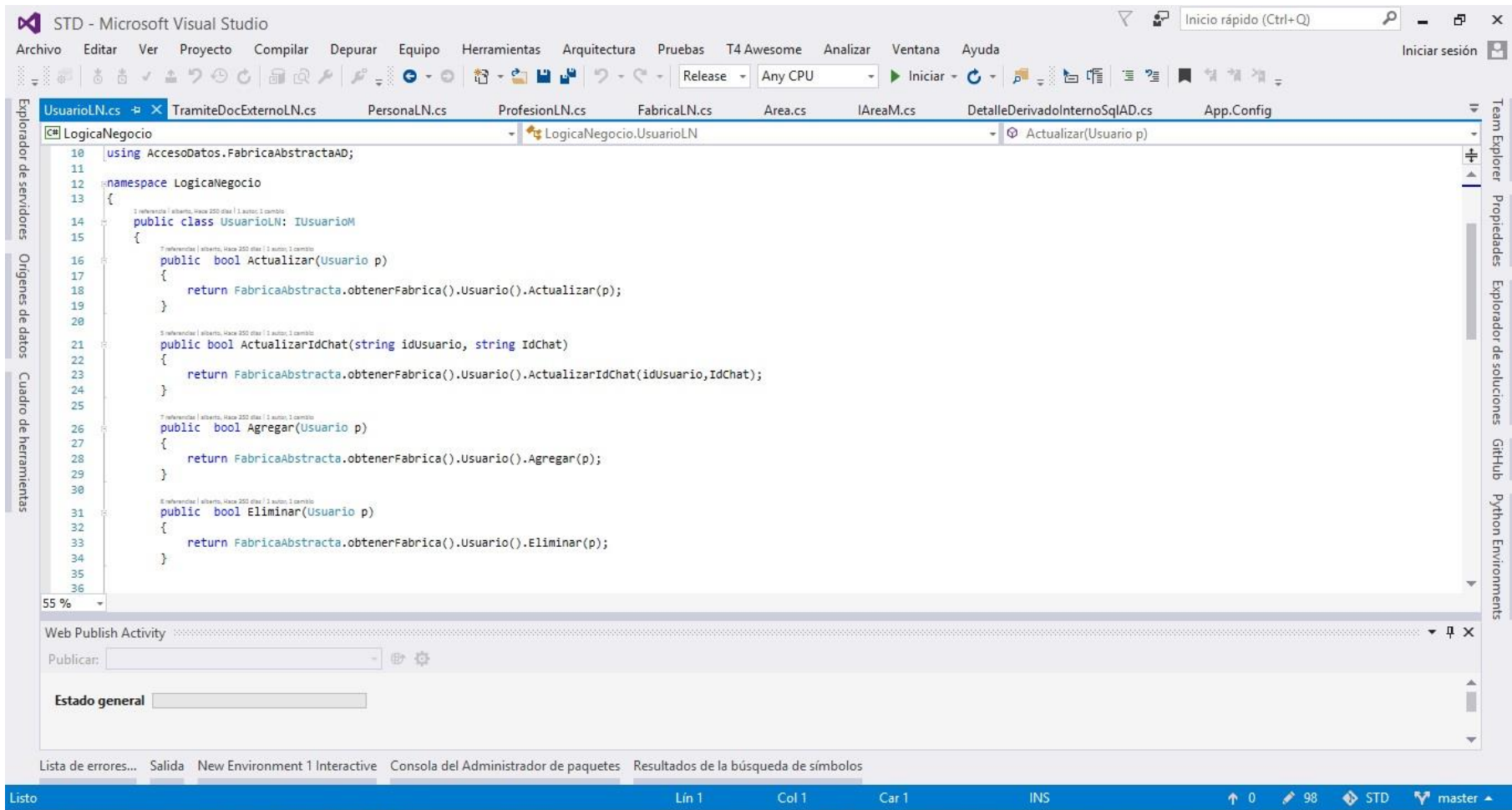
Publicar:  

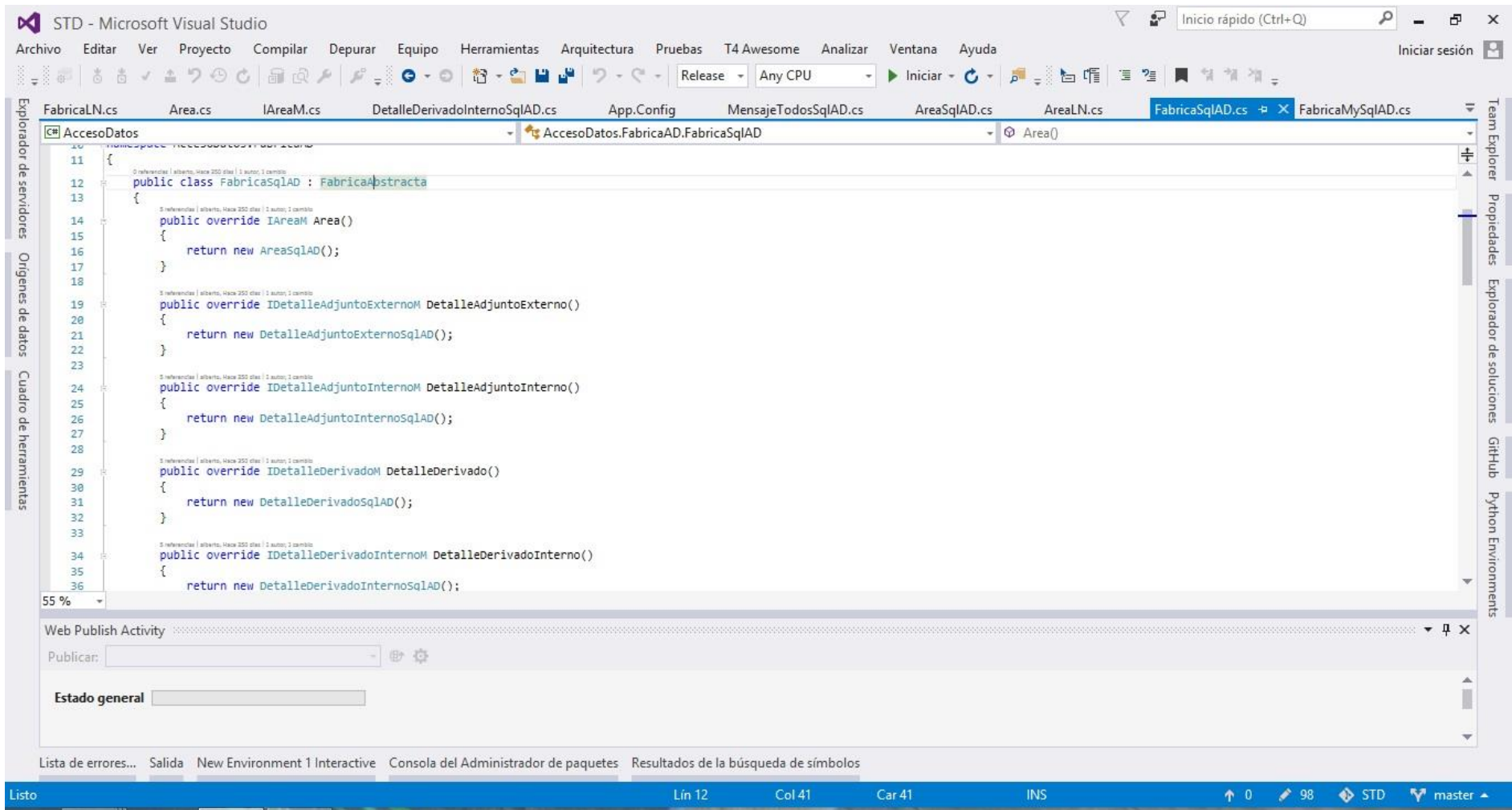
Estado general

Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos

Listo Lin 27 Col 10 Car 10 INS ↑ 0 99 STD master







STD - Microsoft Visual Studio

Archivo Editar Ver Proyecto Compilar Depurar Equipo Herramientas Arquitectura Pruebas T4 Awesome Analizar Ventana Ayuda

Inicio rápido (Ctrl+Q)

Release Any CPU Iniciar

Entidades

```
public partial class Area
{
    16
    17     1 referencia | alberto, Hace 350 días | 1 autor, 4 cambios
    public Area()
    18     {
    19         this.DetalleDerivado = new HashSet<DetalleDerivado>();
    20         this.DetalleDerivadoInterno = new HashSet<DetalleDerivadoInterno>();
    21     }
    22
    23
    24     10 referencias | alberto, Hace 402 días | 1 autor, 1 cambio
    public int IdArea { get; set; }
    25     8 referencias | alberto, Hace 402 días | 1 autor, 1 cambio
    public string Area1 { get; set; }
    26     5 referencias | alberto, Hace 402 días | 1 autor, 1 cambio
    public int IdTrabajador { get; set; }
    27
    28     1 referencia | alberto, Hace 402 días | 1 autor, 1 cambio
    public virtual Trabajador Trabajador { get; set; }
    29     1 referencia | alberto, Hace 402 días | 1 autor, 1 cambio
    public virtual ICollection<DetalleDerivado> DetalleDerivado { get; set; }
    30     1 referencia | alberto, Hace 350 días | 1 autor, 5 cambios
    public virtual ICollection<DetalleDerivadoInterno> DetalleDerivadoInterno { get; set; }
    31
    32 }
    33
    34
```

67%

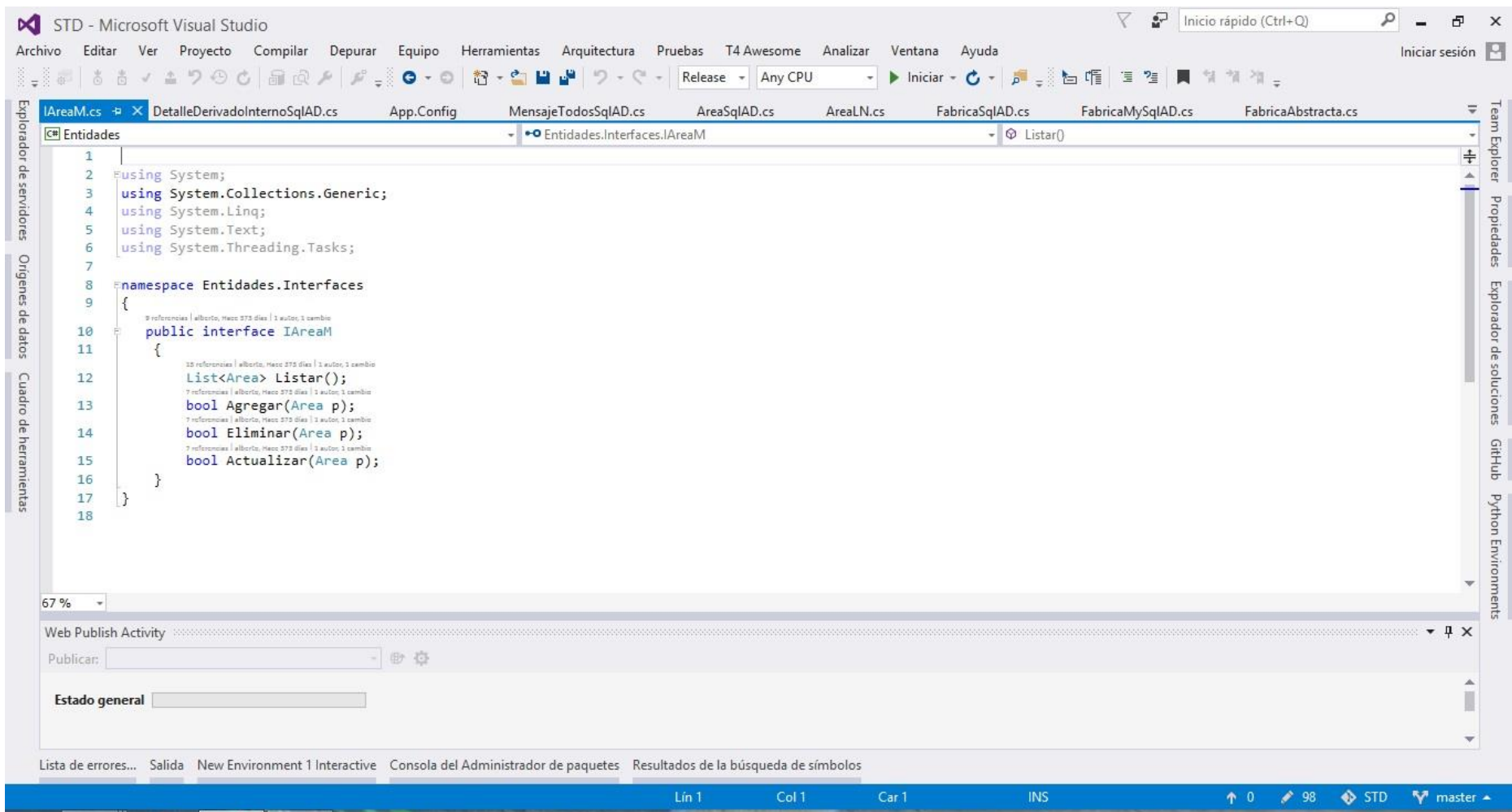
Web Publish Activity

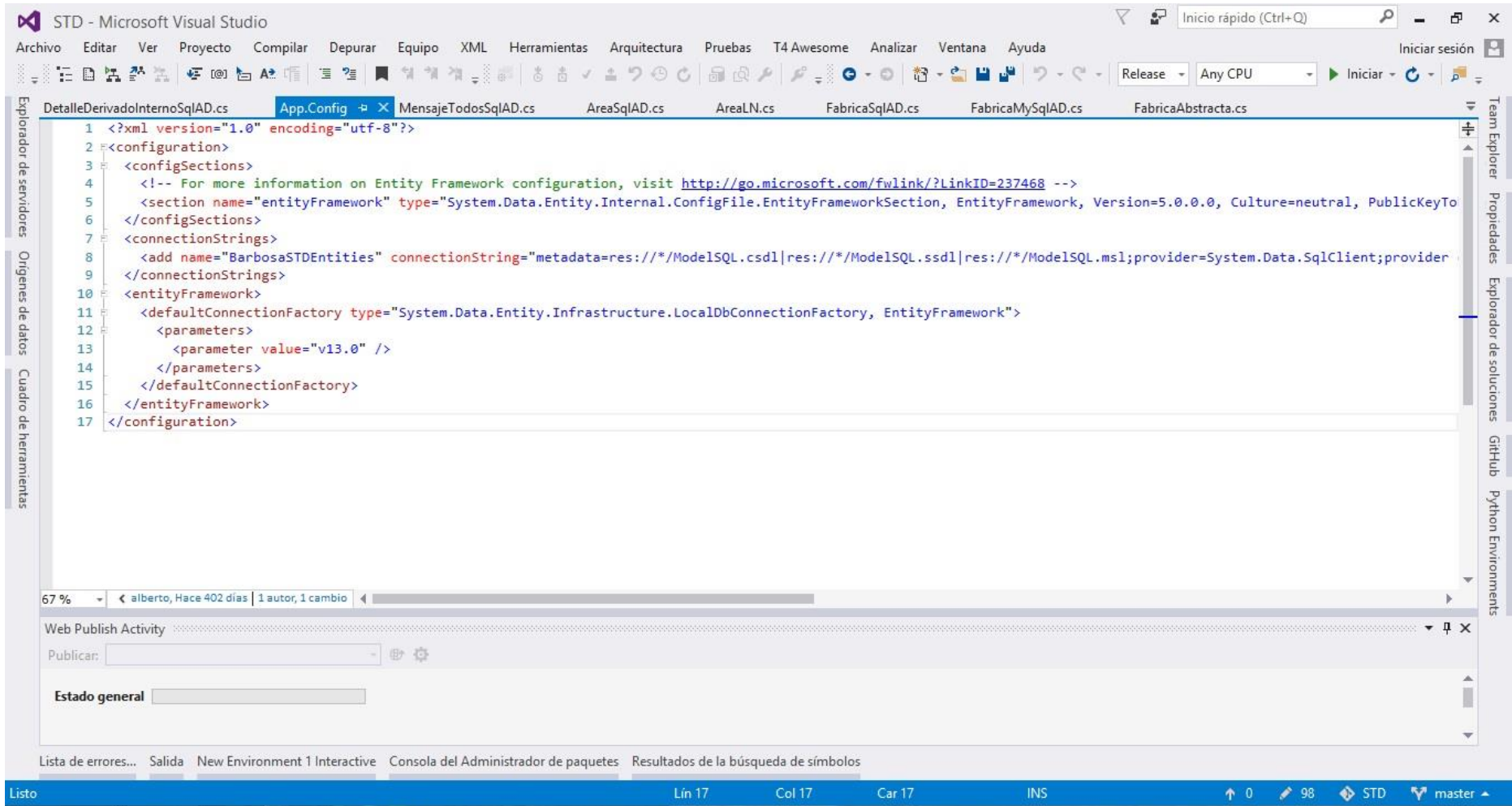
Publicar:

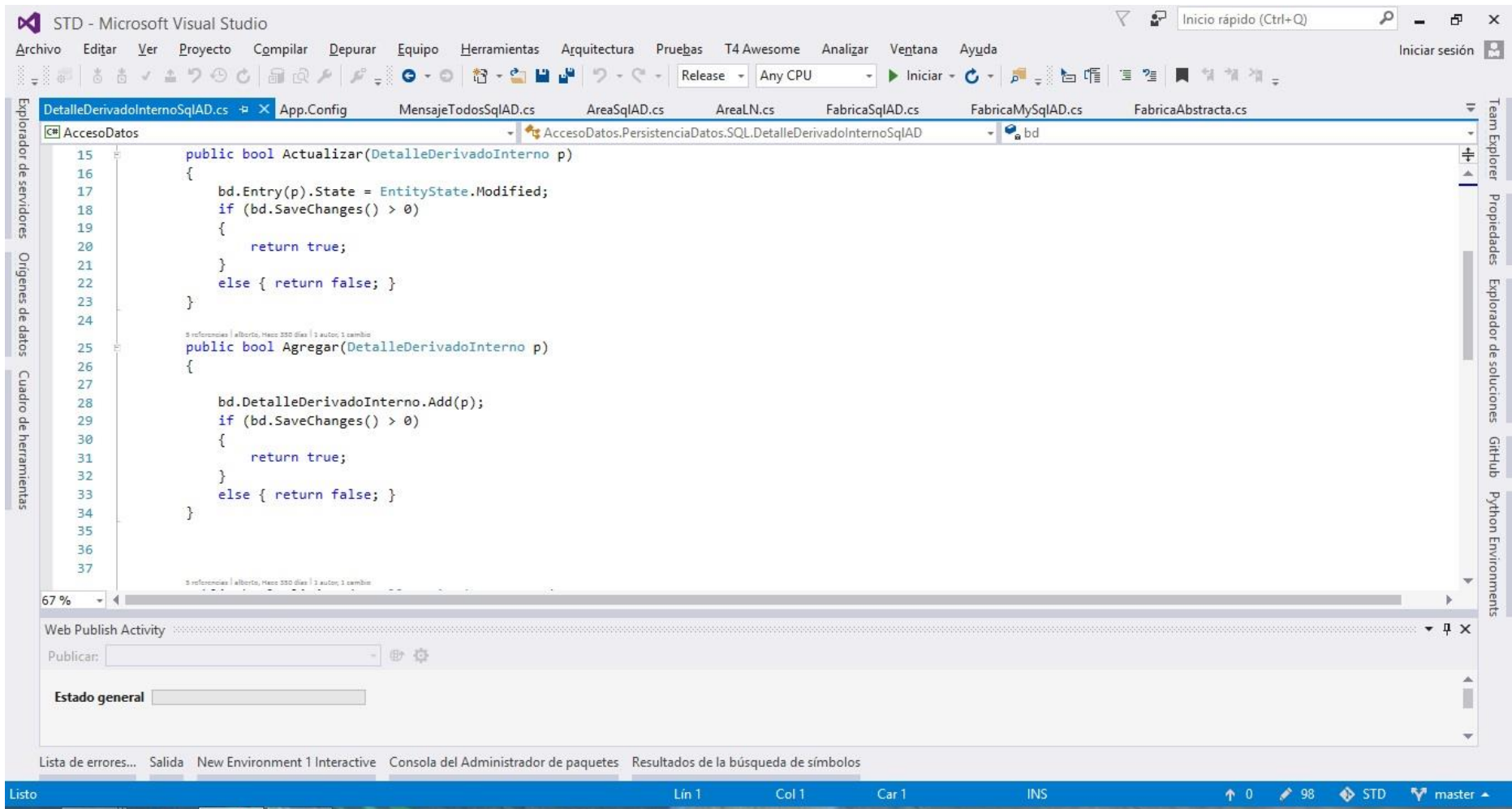
Estado general

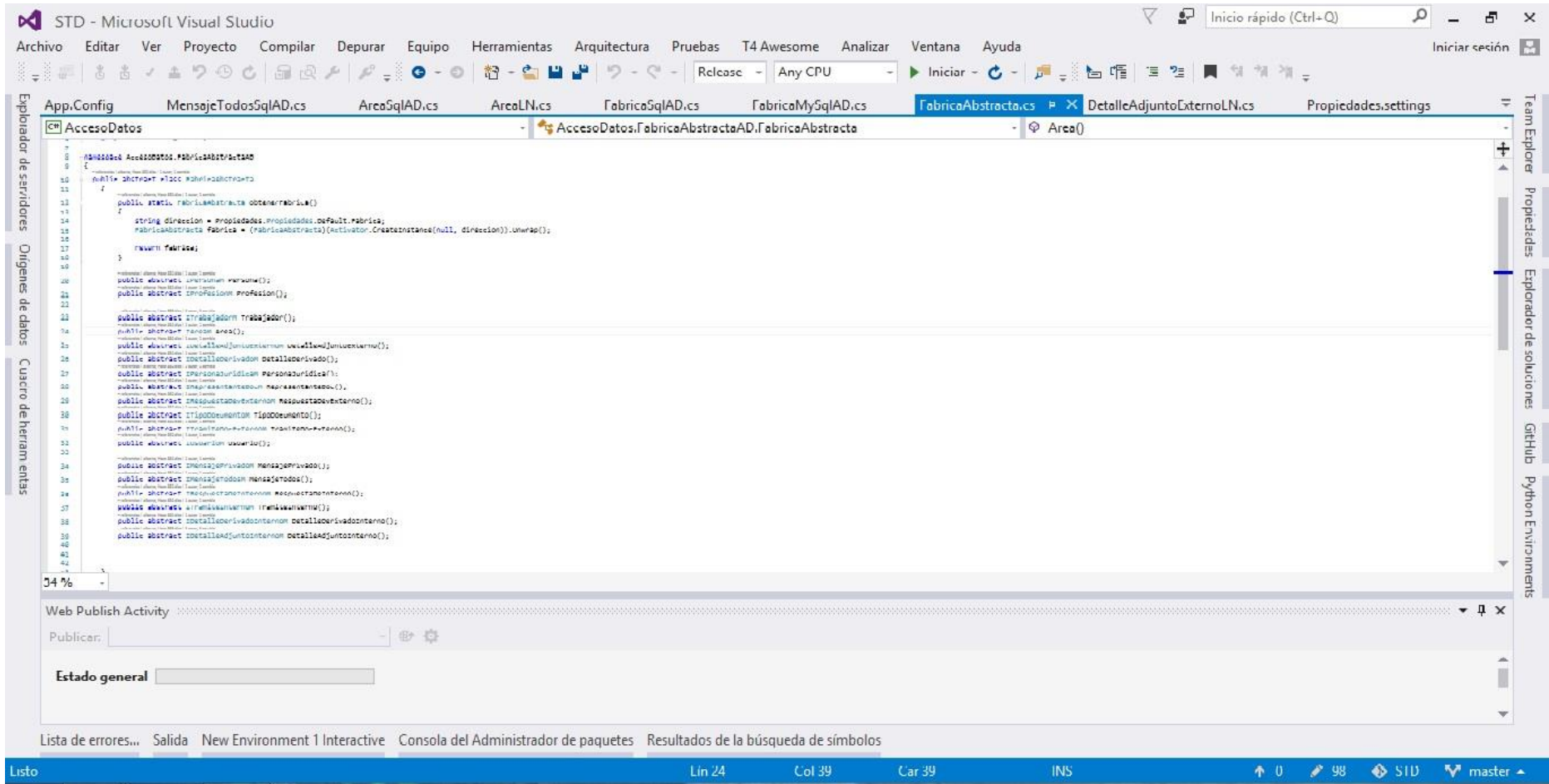
Lista de errores... Salida New Environment 1 Interactive Consola del Administrador de paquetes Resultados de la búsqueda de símbolos

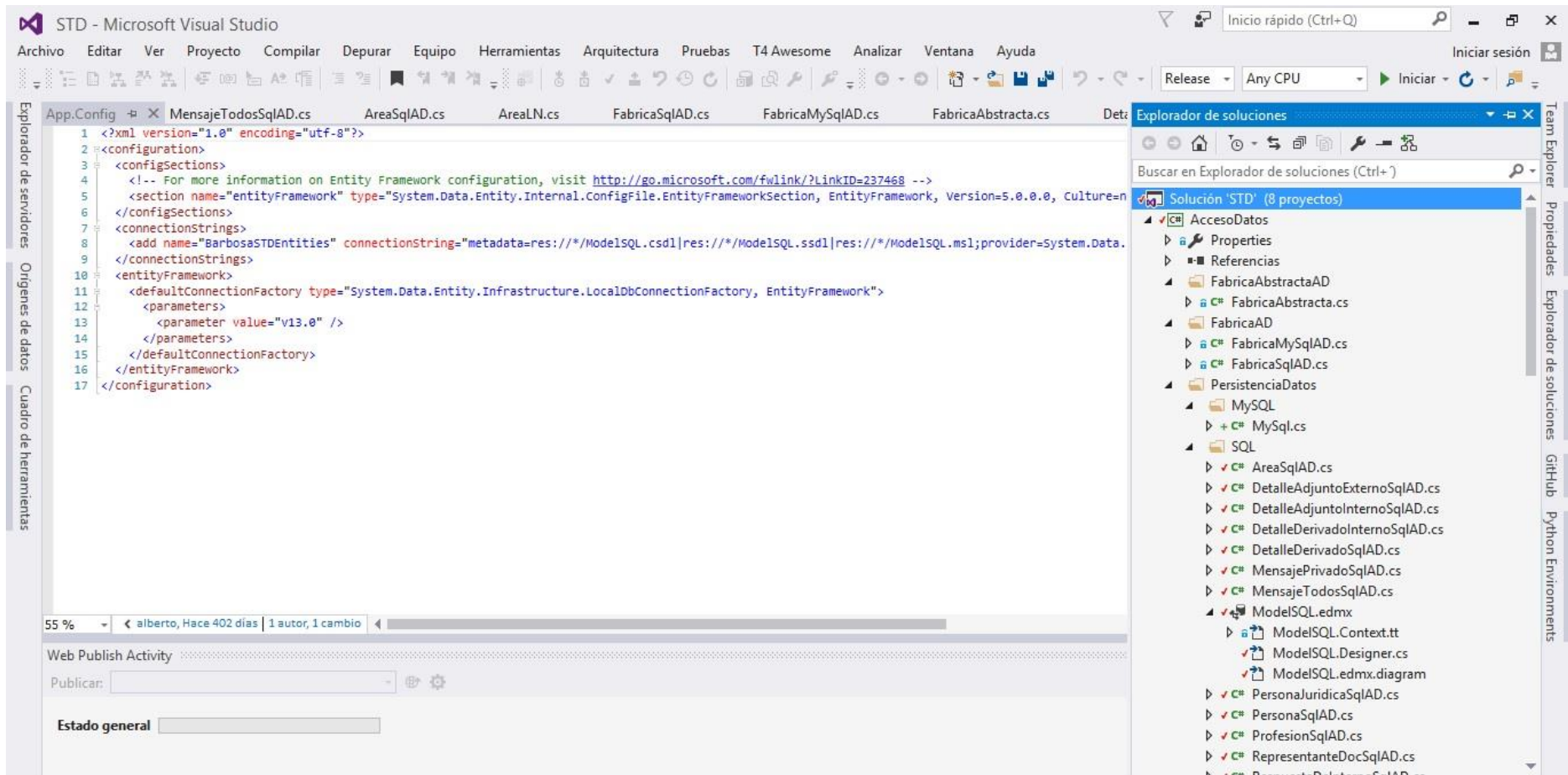
Listo Lin 1 Col 1 Car 1 INS 0 98 STD master











ANEXO 5: MANUAL DE USUARIO

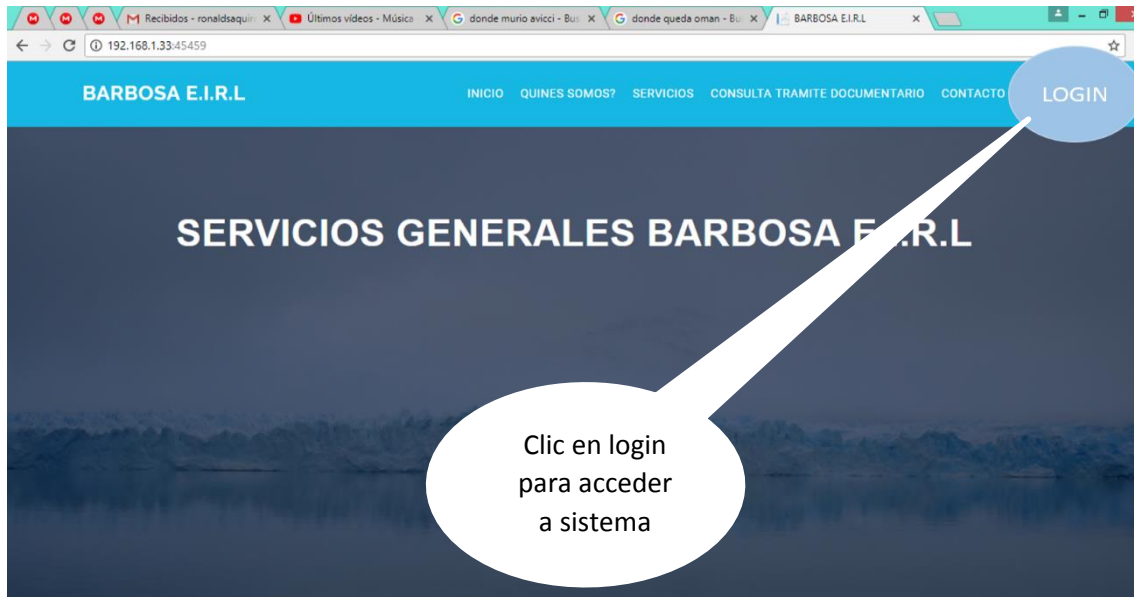
MANUAL DE USUARIO

V.WEB



SISTEMA DE TRAMITES BARBOSA –PLATAFORMA WEB

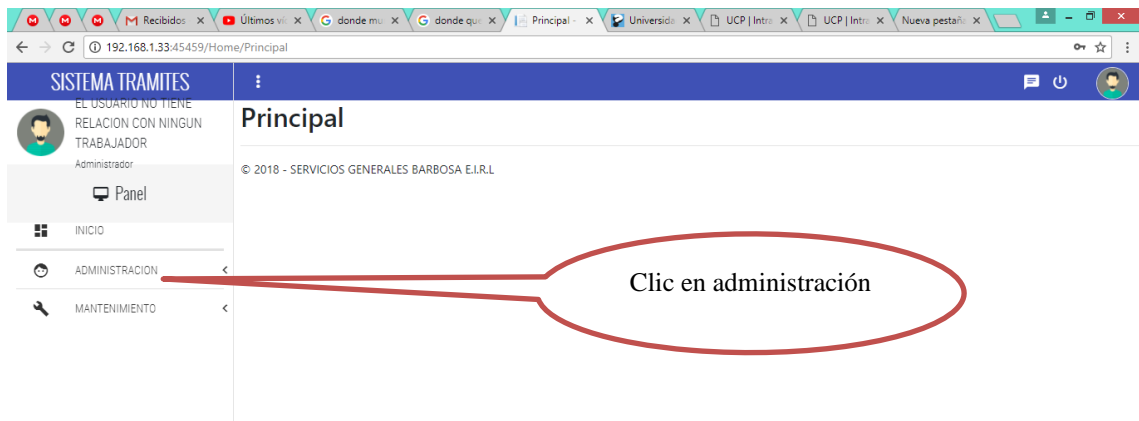
1. vista de la página principal del portal web



2. Vista del formulario de login
3. Vista principal de sistema web de tramite

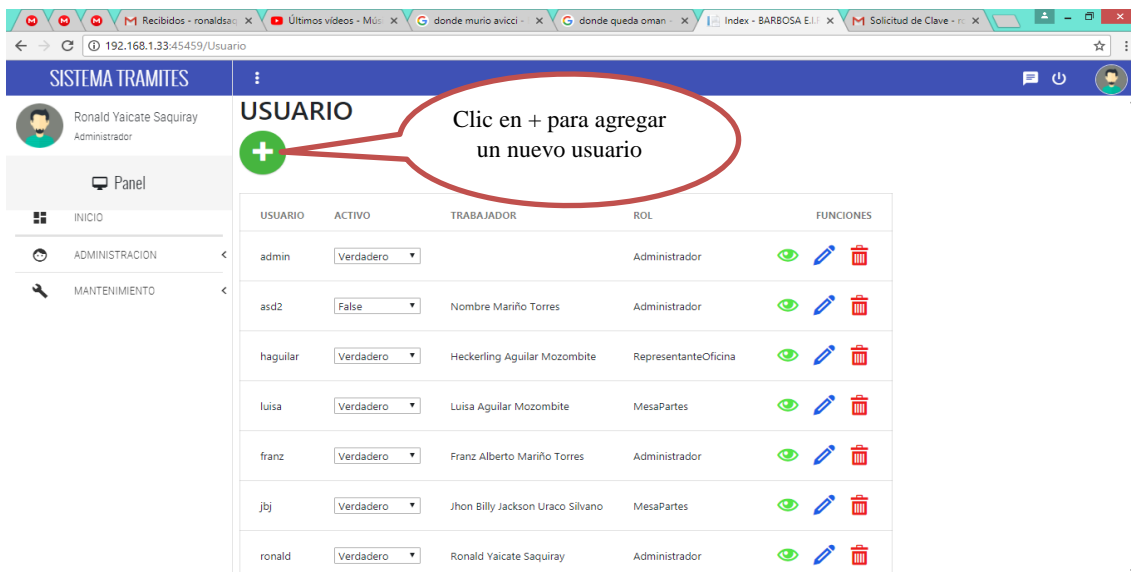


Mantenimiento de usuario



Se abrirá el formulario del usuario donde se puede editar, eliminar y agregar un nuevo usuario.

Vista del formulario usuario



Formulario agregar

SISTEMA TRAMITES

Ronald Yaicte Saquiray
Administrador

Panel

INICIO

ADMINISTRACION

MANTENIMIENTO

NUEVO USUARIO

USUARIO

Contraseña

ACTIVO

Sin establecer

SELECCIONE TRABAJADOR

SELECCIONE ROL

Administrador

CREAR

Una vez que llenaste todo los campos clic en crear y se agregara un nuevo usuario

ronald | Verdadero | Ronald Yaicte Saquiray

04:27 p.m. 24/04/2018

Editar usuario

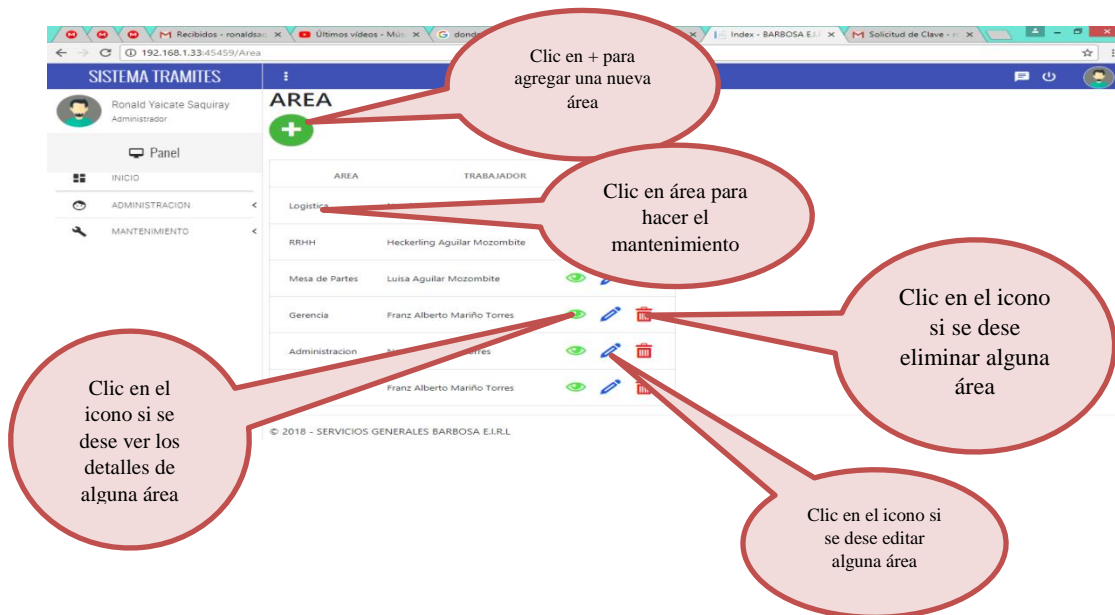
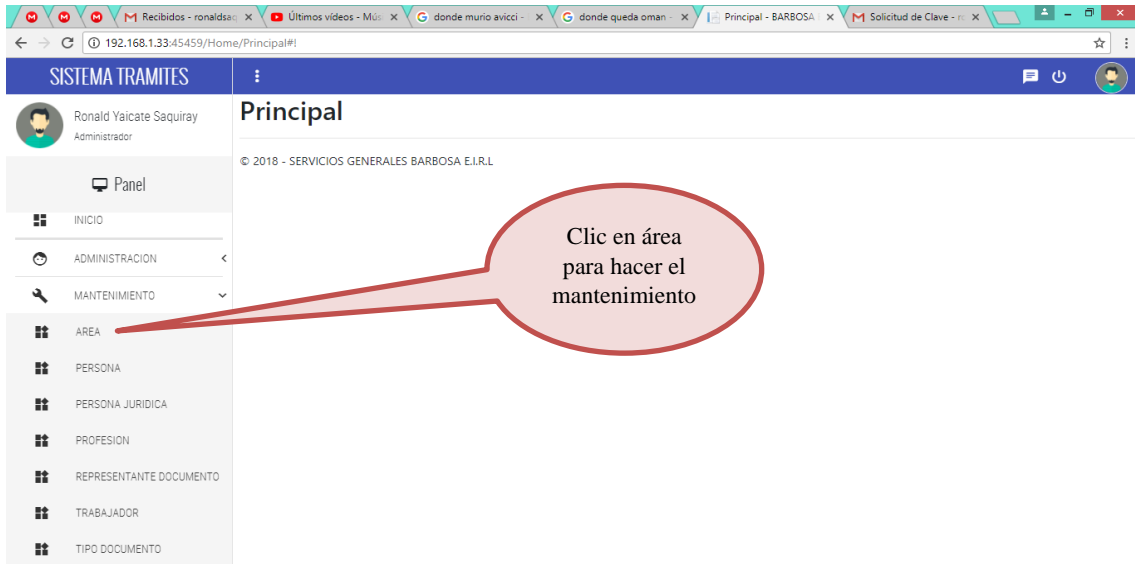
Mantenimiento de los actores del sistema





Nos parecen las opciones para el mantenimiento









Formulario de Área



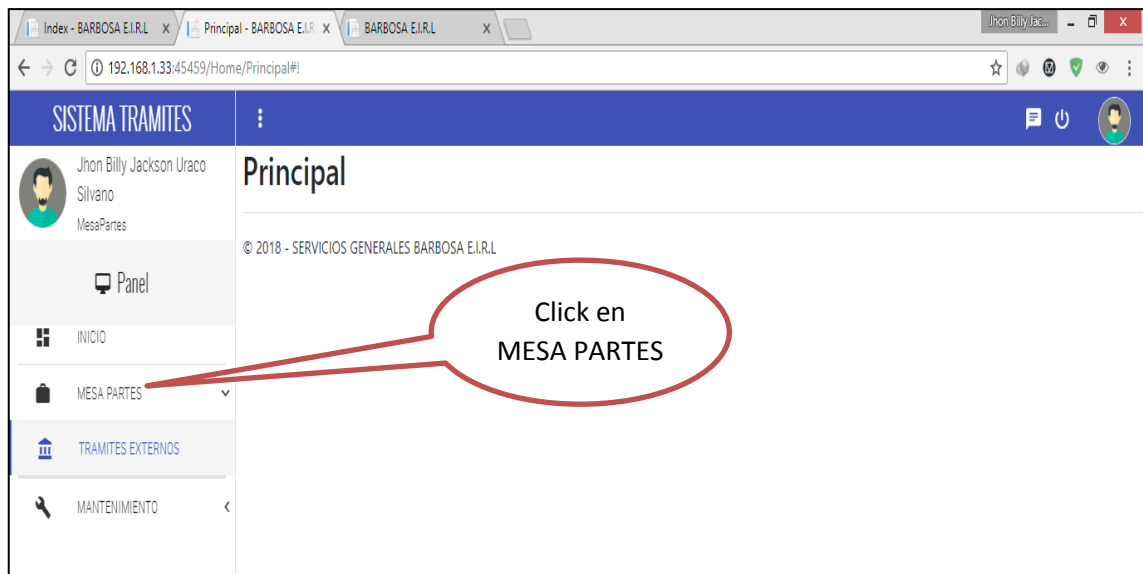
Mantenimiento Persona

The screenshot shows a web application interface for 'PERSONA' management. The interface includes a sidebar with navigation options like 'INICIO', 'ADMINISTRACION', and 'MANTENIMIENTO'. The main content area displays a table of people with columns for 'IdPersona', 'DNI', 'NombreCompleto', 'Email', and 'FUNCIONES'. Three callout boxes provide instructions on how to interact with the table:

- Clic en + para agregar una nueva persona**: Points to a green plus icon in the top left corner of the table area.
- Clic en el icono si se dese ver los detalles de la persona**: Points to a magnifying glass icon in the 'FUNCIONES' column of the first row.
- Clic en el icono si se dese eliminar alguna Persona**: Points to a red trash can icon in the 'FUNCIONES' column of the first row.
- Clic en el icono si se dese editar alguna persona**: Points to a blue pencil icon in the 'FUNCIONES' column of the first row.

IdPersona	DNI	NombreCompleto	Email	FUNCIONES
1	76028998	Nombre Mariño Torres	franz.1996.2015@gmail.com	  
	76028998	Franz Alberto Mariño Torres	franz.1996.2015@gmail.com	  
		ening Aguilar Mozombite	heckerling@gmail.com	  
		Luisa Aguilar Mozombite	dff	  

1. Mesa de Partes.



- Al hacer click en **TRAMITES EXTERNOS**, nos mostrara el formulario donde se puede ver los todos los trámites realizados.

2. Vista LISTA DE TRÁMITES.

- **Formulario NUEVO TRAMITE DOCUMENTO EXTERNO**

SISTEMA TRAMITES

Jhon Billy Jackson Uraco
Silvano
MesaPartes

Panel

INICIO

MESA PARTES

MANTENIMIENTO

NUEVO TRAMITE DOCUMENTO EXTERNO

RAZON SOCIAL
SELECCIONE RAZON SOCIAL

REPRESENTANTE DOCUMENTO
SELECCIONE REPRESENTANTE DOCUMENTO

TIPO DE DOCUMENTO
SELECCIONE TIPO DOCUMENTO

Asunto

NumFolios

AREA
SELECCIONE AREA DESTINADA

GUARDAR

Una vez que llenaste todos los campos clic en GUARDAR

SISTEMA TRAMITES

Jhon Billy Jackson Uraco
Silvano
MesaPartes

Panel

INICIO

MESA PARTES

MANTENIMIENTO

LISTA DE TRAMITES

Clic en + para agregar un

N° CARGO	RAZON SOCIAL	REPRESENTANTE DOCUMENTO	TIPO DE DOCUMENTO
2018-000	Luisa Aguilar Mozombite	Nombre Mariño Torres	
E0-2018-008	Luisa Aguilar Mozombite	Nombre Mariño Torres	
E0-2018-009	Luisa Aguilar Mozombite	Nombre Mariño Torres	
E0-2018-0010	Luisa Aguilar Mozombite	Nombre Mariño Torres	
E0-2018-0011	Luisa Aguilar Mozombite	Julio Mariño Saquiray	
E0-2018-0012	Luisa Aguilar Mozombite	Julio Mariño Saquiray	

3. Consultar Tramite Externo.

Vista CONSULTA TRAMITE DOCUMENTA

