

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

PAQUETE JAVA PARA LA INTEGRACIÓN DE HERRAMIENTAS DE PRUEBAS DE SOFTWARE BASADO EN PATRONES DE SOFTWARE

YHON EDINSON ESTEVEZ MENDOZA
Código: 20191099028

Universidad Distrital Francisco Jose de Caldas
Facultad de Ingenieria
Especializacion en Ingenieria de Software
Bogota, Colombia
2019

PAQUETE JAVA PARA LA INTEGRACIÓN DE HERRAMIENTAS DE PRUEBAS DE SOFTWARE BASADO EN PATRONES DE SOFTWARE

Yhon Edinson Estevez Mendoza

Tesis presentada como requisito para optar por el titulo de:
Especialista en Ingeniera de Software

Director:

Sandro Javier Bolaños

Revisor:

Jorge Mario Calvo

Universidad Distrital Francisco Josse de Caldas
Facultad de Ingenieria
Especializacion en Ingenieria de Software
Bogota, Colombia
2019

Resumen
El presente proyecto tiene como objetivo Diseñar y construir un adaptador que permita integrar tecnologías y herramientas de pruebas basado en patrones de software que facilite el desarrollo de aplicaciones de automatización.Se presenta: la contextualizacion de la investigación, teniendo en cuenta la problemática, los objetivos y la justifican de la investigacion; el desarrollo de la investigacion en cada una de las fases de la construcción del prototipo; la arquitectura empresarial modelada con Archimate y el cierre de la investigación, en el que se analiza la información estadística recolectada y se presenta la conclusión de la investigación.

Abstract
This project aims to design and build an adapter that allows integrating technologies and testing tools based on software patterns that facilitate the development of automation applications.It is presented: the contextualization of the research taking into account the problem, objectives and the justification of the research; the development of investigation in each one of the prototype's building phases; the architecture modeled with Archimate and closure of the investigation, in which it is analyzed the collected statistics information and it is presented the final conclusion of investigation.

Índice general

I PARTE I. CONTEXTUALIZACION DE LA INVESTIGACIÓN 11

1. CAPÍTULO 1. DESCRIPCIÓN DE INVESTIGACIÓN 13
1.1. Planteamiento/identificación del problema 13
1.2. Objetivos 14
1.3. Justificación del trabajo/investigación 15
1.4. Hipótesis 15
1.5. Marco referencial 16
1.6. Patrones de diseño 19
1.7. Metodología de la investigación 23
1.8. Organización del trabajo de grado 24
1.9. Estudio de sistemas previos 24

2. CAPÍTULO 2. DISEÑO 27
2.1. Fase de análisis 27
2.2. Fase de diseño 28
2.3. Fase de pruebas 37

3. CAPITULO 3. ADM-ARCHIMATE 41
3.1. Organización 42
3.2. Punto de vista cooperación de actor 43
3.3. Punto de vista de función del negocio 44
3.4. Punto de vista proceso del negocio 46
3.5. Punto de vista cooperación proceso del negocio 47
3.6. Punto de vista de producto 49
3.7. Punto de vista de la aplicación 50
3.8. Punto de vista de comportamiento de aplicación 50
3.9. Punto de vista de cooperación de aplicación 52
3.10. Punto de vista de estructura de aplicación 53

3.11. Punto de vista de uso de aplicación 54
3.12. Punto de vista tecnológico 55
3.12.1. Punto de vista de infraestructura 55
3.13. Punto de vista motivacional 56
3.13.1. Punto de vista de los skateholder 56
3.13.2. Punto de vista de realización de objetivos 57
3.13.3. Punto de vista de principios 58
3.13.4. Punto de vista de realización de requerimientos 59
3.13.5. Punto de vista de motivación 60
3.14. Punto de vista implementación y migración 61
3.14.1. Modelo punto de vista de proyecto. 61
3.14.2. Modelo punto de vista de proyecto. 62
3.14.3. Punto de vista de migración e implementación. 63

4. CAPITULO 4. RESULTADOS Y DISCUSIÓN 67
4.1. Recolección, análisis y presentación de información 67

5. CAPITULO 5. CONCLUSIONES 71
5.1. Verificación, contraste y evaluación de los objetivos 71
5.1.1. Síntesis del modelo propuesto 72
5.1.2. Aportes originales 72
5.1.3. Trabajos o Publicaciones derivadas 72

6. CAPITULO 6.PROSPECTIVA DEL TRABAJO DE GRADO 73
6.1. Lineas de investigación futuras 73
6.2. Trabajos de investigación futuros 73

Índice de figuras

2.1. Caso de uso para la automatización de una tarea 28

3.1. Modelo punto de vista Organizacional 42
3.2. Caso de estudio punto de vista Organizacional. Fuente los autores 43
3.3. Modelo punto de vista cooperación de actor. 43
3.4. Caso estudio punto de vista cooperación de actor. Fuente: los autores 44
3.5. Modelo punto de vista de función del negocio. 44
3.6. Caso estudio punto de vista de función del negocio. Fuente: los autores 45
3.7. Modelo punto de vista proceso del negocio. 46
3.8. Caso estudio punto de vista proceso del negocio. Fuente: los autores 46
3.9. Modelo punto de vista cooperación proceso del negocio. . . . 47
3.10. Caso estudio punto de vista cooperación proceso del negocio. Fuente: los autores 48
3.11. Modelo punto de vista de producto. 49
3.12. Caso estudio punto de vista de producto. 50
3.13. Modelo punto de vista de comportamiento de aplicación. . . . 51
3.14. Caso estudio punto de vista de comportamiento de aplicación. Fuente: los autores 51
3.15. Modelo punto de vista de cooperación de aplicación. 52
3.16. Caso estudio punto de vista de cooperación de aplicación. Fuente: los autores 52
3.17. Modelo punto de vista de estructura de aplicación. 53
3.18. Caso estudio punto de vista de estructura de aplicación. Fuente: los autores 53
3.19. Modelo punto de vista de uso de aplicación. 54

3.20. Caso estudio punto de vista de uso de aplicación. Fuente: los autores 54
3.21. Modelo punto de vista de infraestructura. 55
3.22. Modelo punto de vista de infraestructura.Fuente: los autores . 55
3.23. Modelo punto de vista de los skateholder 56
3.24. Caso estudio punto de vista de los skateholder. Fuente: los autores 56
3.25. Modelo punto de vista de realización de objetivos 57
3.26. Caso estudio punto de vista de realización de objetivos. Fuente: los autores 57
3.27. Modelo punto de vista de principios. 58
3.28. Caso estudio punto de vista de principios. Fuente: los autores 58
3.29. Modelo punto de vista de realización de requerimientos. . . . 59
3.30. Caso estudio punto de vista de realización de requerimientos. Fuente: los autores 59
3.31. Modelo punto de vista de motivación 60
3.32. Caso estudio Modelo punto de vista de motivación. Fuente: los autores 60
3.33. Modelo punto de vista de proyecto. 61
3.34. Caso estudio Modelo punto de vista de proyecto. Fuente: los autores 61
3.35. Modelo punto de vista de migración. 62
3.36. Modelo punto de vista de migración. Fuente: los autores . . . 62
3.37. Modelo punto de vista de migración e implementación. 63
3.38. Modelo punto de vista de migración. Fuente: los autores . . . 63

4.1. Resultados de encuesta de implementación 2015-2017. % de encuestados Fuente: [3] 68

INTRODUCCIÓN

La automatización de procesos tiene el potencial de beneficiar a un gran número de organizaciones en un futuro no muy lejano, lo que hace que las empresas empiecen a buscar que procesos puede ser automatizado dentro de la empresa para aumentar la calidad y producir más con menos inversión.
El auge de la automatización hace que las empresas tengan que irse adaptando y realizar cambios en sus estrategias para ser más competitivos.
Las tareas repetitivas que no producen grandes beneficios a la empresa pero que no pueden dejar de hacer y que no requieren de personal muy capacitado son uno de los objetivos a automatizar, tareas como llenar formularios, tipificación de procesos, consulta de información en aplicativos de terceros para luego procesarla, esta información por lo general esta guardada en base de datos o en tablas de Excel y deben de ser ingresadas en una página web o un aplicativo de terceros que son del cliente para el cual desarrollan la actividad.
Lo primero que conviene advertir es que, aunque empleemos la palabra robots, no debemos pensar en androides al estilo de las películas de ciencia ficción, ni siquiera en los brazos articulados programables que vemos en tantas y tantas fábricas fuertemente robotizadas. [7]
De lo que hablamos en este caso es de robots software, es decir, pequeños programas que automatizan tareas. Además, el ámbito de aplicación de estos robots es normalmente en tareas de oficina, más que industriales. En lo que sí se parecen los robots objeto de la RPA a aquellos que se aplican en los entornos industriales, es en su capacidad de automatización, en su versatilidad y en que pueden realizar con notable rapidez, acierto y eficiencia, algunas de las tareas más masivas, repetitivas y tediosas de los entornos de trabajo [4, 8]
La automatización de procesos a través de robots (RPA) es el uso de un software que imita la forma en que las personas llevan a cabo tareas repetitivas, pero de manera más eficiente [12].
La automatización de procesos robóticos (RPA) es una nueva ola de tecno-

logías futuras. Proceso robótico La automatización es una de las tecnologías más avanzadas en el área de informática, Electrónica y comunicaciones, ingeniería mecánica, y tecnología de la información. Es una combinación de tanto hardware como software, redes y automatización para realizar cosas muy simples [3, 11].
El término .Automatización de procesos robóticosçonnota visiones de robots físicos errantes alrededor de las oficinas que realizan tareas humanas, el término realmente significa la automatización de las tareas de servicio que fueron previamente realizados por los humanos. Para procesos de negocios, el término RPA se refiere comúnmente a configurar el software para realizar el trabajo realizado previamente por personas [1].

Parte I

PARTE I. CONTEXTUALIZACION DE LA INVESTIGACIÓN

Capítulo 1

CAPÍTULO 1. DESCRIPCIÓN DE INVESTIGACIÓN

1.1. Planteamiento/identificación del problema

Planteamiento del Problema
Las tareas repetitivas como llenar formularios, consultar información son trabajos que muchas veces no requieren de mucho razonamiento ni una preparación especializada para realizarlos, lo que hace que las personas se vuelvan mecánicas y se cansen de hacer lo mismo, causan problemas físicos debido al exceso de uso de un cierto tipo de movimiento en las manos o en los brazos. ¡Cualquier movimiento repetitivo puede causar una lesión inclusive enviar mensajes de texto a través del teléfono!
El auge de la automatización hace que las empresas tengan que irse adaptando y realizar cambios en sus estrategias para ser más competitivos. Las tecnologías de automatización se están volviendo obligatorias como parte de las operaciones comerciales en las organizaciones alrededor del mundo. Cada vez empresas buscan producir más con menos inversión, es más eficiente que sus empleados se dediquen a realizar tareas de mayor valor como toma decisiones y buscar automatizar las tareas repetitivas como la consulta de información, llenado de formularios , descargar y subir archivos, enviar correos etc.
Hoy en día las operaciones comerciales están avanzando hacia el nuevo terreno tecnológico, el llamado "Proceso Robótico.Automatización. "Por lo tanto, el año"2018 "se conoce como el año de la.Automatización de procesos robóticos". La industria está entrando en el nuevo mundo de la tecnología Robotic Process Automation". RPA". La adopción oportuna de RPA se ha

convertido en una necesidad en las actividades comerciales diarias. Si las empresas no configuran esta tecnología en sus operaciones, es posible que no se sostengan en un negocio.

Formulación del problema

¿Cómo construir un paquete en java que sirva como base para crear aplicaciones que permitan automatizar tareas repetitivas en un computador, que permitirá aumentar su productividad disminuyendo los costos y el tiempo en terminar los procesos sin perder calidad, evitando que sus empleados sufran problemas de salud debido a los movimientos repetitivos que ejecutan al realizar estas tareas, teniendo en cuenta que para automatizar una tarea se debe diseñar una aplicación específicamente para cada tarea?

Sistematización del problema

- ¿Cómo integrar las tecnologías de pruebas y automatización de software para crear un paquete que facilite la creación de aplicaciones de automatización?
- ¿Qué tipo de patrones son los más adecuados para realizar la integración?
- ¿Qué tipo de arquitectura se debe manejar para realizar el paquete?

1.2. Objetivos

Objetivo general

Diseñar y construir un adaptador que permita integrar tecnologías y herramientas de pruebas basado en patrones de software que facilite el desarrollo de aplicaciones de automatización

Objetivos Específicos

- Estudiar el funcionamiento de las tecnologías de automatización y pruebas de software, y el cómo integrarlas para realizar un paquete java.

- Definir el tipo el tipo de patrón que se usara para integrar las herramientas en un paquete java y que facilite la utilización a la hora de desarrollar aplicaciones de automatización.
- Realizar una aplicación que nos permita mostrar en funcionamiento del paquete java bien sea en una aplicación de escritorio o página web.

1.3. Justificación del trabajo/investigación

La adopción oportuna de RPA nos ayudara estar al día en estos procesos de automatización procesos. Si las empresas no configuran esta tecnología en sus operaciones, es posible que no se sostengan en un negocio. La competencia en un futuro próximo, la automatización de procesos robóticos (RPA) es una forma emergente de negocio. Tecnología de automatización de procesos basada en la noción de robots de software o inteligencia artificial (AI). Trabajar en RPA se ha convertido en el nuevo lenguaje de los negocios. Esta

tecnología actualizada es más potente entre las tecnologías del siglo XXI. La forma en que las empresas hacen su negocio, la forma la gente que realiza el trabajo oficial, así como la vida cotidiana pública, será asistida drásticamente por nuevos hardware, tecnologías de software junto con dispositivos inteligentes. Los estilos de vida del ser humano están cambiando debido a colaboraciones globales, empresas multinacionales, nuevas TI / TI avances en la tecnología RPA. Junto con las nuevas tecnologías existentes funcionales en la vida humana. Internet de las cosas, análisis de big data, aprendizaje profundo, inteligencia artificial, el aprendizaje automático y otras tecnologías relacionadas, la automatización de procesos robóticos (RPA) se está convirtiendo una de las tecnologías perturbadoras notables. Por esta razón

se decide trabajar en el desarrollo o la integración de herramientas que permitan la automatización de procesos repetitivos en un computador, debemos empezar a prepararnos para estar al día con las nuevas tecnologías que nos permitan ser más competitivos y no desaparecer del mercado.

1.4. Hipótesis

Las aplicaciones que automatizan tareas repetitivas en un computador manipulando los aplicativos como si lo hiciera una persona son los que hacen

que los procesos en las empresas se realicen de manera más eficiente y sin perder calidad logrando una mayor productividad.

1.5. Marco referencial

Marco teórico

Automatización de pruebas para aplicaciones web

Muchas, tal vez la mayoría, de las aplicaciones de software de hoy en día están escritas como aplicaciones basadas en la web que se ejecutan en un navegador de Internet. La efectividad de probar estas aplicaciones varía ampliamente entre compañías y organizaciones. En una era de procesos de software altamente interactivos y receptivos en los que muchas organizaciones utilizan alguna forma de metodología ágil, la automatización de pruebas se está convirtiendo en un requisito para los proyectos de software. La automatización de pruebas es a menudo la respuesta. La automatización de prueba significa usar una herramienta de software para ejecutar pruebas repetibles contra la aplicación que se va a probar. Para las pruebas de regresión esto proporciona esa capacidad de respuesta. Hay muchas ventajas para probar

la automatización. La mayoría está relacionada con la repetibilidad de las pruebas y la velocidad a la que se puede ejecutar las pruebas. Hay una serie de herramientas comerciales y de código abierto disponibles para ayudar con el desarrollo de la automatización de pruebas. El selenio es posiblemente la solución de código abierto más utilizada. Esta guía del usuario ayudará a los usuarios nuevos y experimentados de Selenium a aprender técnicas efectivas en la creación de automatización de pruebas para aplicaciones web.

Esta guía del usuario presenta Selenium, enseña sus características y presenta las mejores prácticas de uso común acumuladas de la comunidad de Selenium. Se proporcionan muchos ejemplos. Además, se proporciona información técnica sobre la estructura interna de Selenium y los usos recomendados de Selenium.

Presentando Selenium

Selenium es un conjunto de herramientas de software diferentes, cada una con un enfoque diferente para admitir la automatización de pruebas. La mayoría de los ingenieros de Selenium QA se centran en una o dos herramientas

que satisfacen las necesidades de su proyecto, sin embargo, aprender todas las herramientas le dará muchas opciones diferentes para abordar problemas de automatización de pruebas. El grupo completo de herramientas da como resultado un amplio conjunto de funciones de prueba específicamente orientadas a las necesidades de prueba de aplicaciones web de todo tipo. Estas operaciones son muy flexibles, ya que permiten muchas opciones para ubicar elementos de la IU y comparar los resultados esperados de las pruebas con el comportamiento real de la aplicación. Una de las características clave de Selenium es el soporte para ejecutar las pruebas en múltiples plataformas de navegador. Muchas, tal vez la mayoría, de las aplicaciones de software de hoy

en día están escritas como aplicaciones basadas en la web que se ejecutan en un navegador de Internet. La efectividad de probar estas aplicaciones varía ampliamente entre compañías y organizaciones. En una era de procesos de software altamente interactivos y receptivos en los que muchas organizaciones utilizan alguna forma de metodología ágil, la automatización de pruebas se está convirtiendo en un requisito para los proyectos de software. La automatización de pruebas es a menudo la respuesta. La automatización de prueba significa usar una herramienta de software para ejecutar pruebas repetibles contra la aplicación que se va a probar. Para las pruebas de regresión esto proporciona esa capacidad de respuesta. Hay muchas ventajas para probar

la automatización. La mayoría está relacionada con la repetibilidad de las pruebas y la velocidad a la que se puede ejecutar las pruebas. Hay una serie de herramientas comerciales y de código abierto disponibles para ayudar con el desarrollo de la automatización de pruebas. El selenio es posiblemente la solución de código abierto más utilizada. Esta guía del usuario ayudará a los usuarios nuevos y experimentados de Selenium a aprender técnicas efectivas en la creación de automatización de pruebas para aplicaciones web. Esta

guía del usuario presenta Selenium, enseña sus características y presenta las mejores prácticas de uso común acumuladas de la comunidad de Selenium. Se proporcionan muchos ejemplos. Además, se proporciona información técnica sobre la estructura interna de Selenium y los usos recomendados de Selenium [10].

Herramienta de automatización Sikuli

Sikuli es una herramienta para automatizar interfaces gráficas de usuario (GUI) utilizando el método de "Visual Image Match". En Sikuli, todos los elementos web deben tomarse como una imagen y almacenarse dentro del

proyecto. Sikuli activará las interacciones GUI basadas en la coincidencia visual de la imagen, la imagen que hemos pasado como parámetro junto con todos los métodos. Sikuli puede ser muy útil para automatizar objetos flash (que no tienen ID o nombre). Puede ser útil en la situación, donde tenemos una GUI estable (es decir, los componentes de la GUI no cambian). Incluso

las aplicaciones basadas en Windows también se pueden automatizar utilizando Sikuli. Sikuli proporciona Sikuli-script.jar muy amigable, que puede usarse fácilmente junto con Selenium WebDriver. Incluso podemos automatizar Adobe Video / Audio Player, Flash Games en el sitio web utilizando Sikuli. Con API simple, facilita la codificación.

Usos prácticos.

1. Sikuli se puede utilizar para automatizar objetos Flash / sitios web Flash
2. Puede ser útil automatizar la aplicación basada en Windows. Podemos automatizar, lo que estamos viendo en la pantalla.
3. Proporciona, API simple. Es decir, se puede acceder a todos los métodos utilizando el objeto de clase de pantalla.
4. Se puede integrar fácilmente con Selenium y todas las demás herramientas.
5. Usando Sikuli podemos automatizar aplicaciones de escritorio
6. La mayoría de las herramientas de prueba de automatización no son compatibles con la automatización de objetos flash (por ejemplo, Selenium). Sikuli proporciona un amplio soporte para automatizar objetos flash.
7. Utiliza un poderoso mecanismo de çoincidencia visual"para automatizar los objetos de escritorio y flash.

Beneficios

- Herramienta de código abierto.
- Una de las mayores ventajas de Sikuli es que puede automatizar fácilmente los objetos Flash.
- Facilita la automatización de la aplicación de Windows.

- Cuando está probando una aplicación en desarrollo y no sabe el ID / nombre de los elementos, entonces puede usar Sikuli. Verificará el aspecto de la imagen y, si se encuentra una coincidencia, interactuará con la imagen en consecuencia [9]

1.6. Patrones de diseño

Patrones Creacionales : Son patrones de diseño relacionados con la creación o construcción de objetos. Estos patrones intentan controlar la forma en que los objetos son creados implementando mecanismos que eviten la creación directa de objetos.

Patrones Estructurales: Son patrones que tiene que ver con la forma en que las clases se relacionan con otras clases. Estos patrones ayudan a dar un mayor orden a nuestras clases ayudando a crear componentes más flexibles y extensibles.

Patrones de Comportamiento: Son patrones que están relacionados con procedimientos y con la asignación de responsabilidad a los objetos. Los patrones de comportamiento engloban también patrones de comunicación entre ellos.

Como mencionamos anteriormente, los patrones creacionales nos sirven para controlar la forma en que creamos los objetos, de entrada nos puede parecer un poco extraño, ya que estamos acostumbrados a crear libremente nuestros objetos, sin embargo, existen situaciones donde por conveniencia es necesario establecer un mecanismo que nos permita crear instancias de una forma controlada. Esta necesidad puede nacer debido a que queremos que sólo exista una instancia de una clase o no se sabe exactamente qué objeto debemos instanciar sino hasta en tiempo de ejecución o porque queremos que nuestras clases sean creadas de una forma más simple mediante una clase de utilidad. Pueden existir cientos de motivos por los cuales deseamos que nuestras clases sean creadas de forma controlada, sin embargo, lo importante es tener la visión de identificarlas y utilizar patrones creacionales que se adapten a nuestro problema.

Los patrones que abordaremos en este capítulo son:

Patrón Factory Method: Patrón que se centra en la creación de una clase fábrica la cual tiene métodos que nos permitan crear objetos de un subtipo determinado.

Patrón Abstract Factory: Patrón muy similar al Factory Method, sin embargo, este patrón nos permite crear objetos de una determinada familia de clases.

Patrón Singleton: Patrón utilizado para controlar la creación de una clase determinada, de esta forma sólo se puede crear una única instancia en toda la aplicación.

Patrón Builder: Patrón que permitir la creación de objetos complejos desde un objeto Builder. El objeto Builder se compone de una variedad de partes que contribuyen individualmente a la creación del objeto.

Patrón Prototype: Este patrón se centra en la creación de objetos a partir de la clonación de otros objetos existentes. Es mucho más rápido clonar un objeto que crear uno nuevo.

Patrón Object Pool: Patrón que se utiliza para mantener un conjunto de objetos creados listos para ser utilizados, evitando crearlos bajo demanda cada vez que se requieran. Los objetos desocupados son devueltos al pool en vez de destruirse (muy utilizado para Pool de Conexiones) [6].

Patrón Adapter: Una clase Adapter implementa un interfaz que conoce a sus clientes y proporciona acceso a una instancia de una clase que no conoce a sus clientes, es decir convierte la interfaz de una clase en una interfaz que el cliente espera. Un objeto Adapter proporciona la funcionalidad prometida por un interfaz sin tener que conocer que clase es utilizada para implementar ese interfaz. Permite trabajar juntas a dos clases con interfaces incompatibles.

Patrón Iterator: Define un interface que declara métodos para acceder secuencialmente a los objetos de una colección. Una clase que accede a una colección solamente a través de un interface independiente de la clase que implementa el interface.

Patrón Bridge: Es útil cuando hay una jerarquía de abstracciones y la correspondiente jerarquía de implementaciones. Más que combinar las abstracciones e implementaciones en muchas clases distintas, el patrón Bridge implementa las abstracciones e implementaciones como clases independientes que se pueden combinar dinámicamente. Es decir, desacopla una abstracción de su implementación y les permite variar independientemente.

Patrón Facade: Simplifica los accesos a un conjunto de objetos relacionados proporcionando un objeto que todos los objetos de fuera del conjunto utilizan

para comunicarse con el conjunto. Define una interface de más alto nivel que permite usar el sistema más fácil.

Patrón Flyweight : Si instancias de una clase contienen la misma información pueden ser utilizadas intercambiablemente, el patrón Flyweight permite a un programa evitar el coste de múltiples instancias que contienen la misma información compartiendo una única instancia. Soporta la representación de un gran número de objetos pequeños de una manera eficiente.

Patrón Dynamic Linkage : Permite a un programa solicitar la carga y utilización de clases arbitrarias que implementan un interface conocido.

Patrón Virtual Proxy : Si un objeto es caro de instanciar y puede que no sea necesario, podría tener ventajas posponer la instanciación hasta que este claro que el objeto se necesita. El patrón Virtual Proxy oculta el hecho de que el objeto todavía no existe a sus clientes, teniendo éstos acceso al objeto indirectamente a través de un objeto proxy que implementa el mismo interface que el objeto que puede no existir. La técnica de retrasar la instanciación de un objeto hasta que este es necesario se llama instanciación perezosa.

Patrón Decorator : Extiende la funcionalidad de un objeto dinámicamente de tal modo que es transparente a sus clientes, utilizando una instancia de una subclase de la clase original que delega las operaciones al objeto original. Provee una alternativa muy flexible para agregar funcionalidad a una clase.

Patrón Cache Management : Permite el acceso rápido a objetos que de lo contrario podrían llevar un gran tiempo de acceso. Esto implica tener una copia de los objetos que son costosos de construir. El objeto podría ser costoso de construir por un número de razones tales como necesitar un gran cálculo o ser sacado de una base de datos.

PATRONES DE COMPORTAMIENTO

Los patrones de este tipo son utilizados para organizar, manejar y combinar comportamientos.

Patrón Chain of Responsability : Permite a un objeto enviar un comando sin conocer que objeto o objetos lo recibirán, evitando el acoplamiento entre el que envía y el que recibe una petición. Esto permite el paso del comando a un objeto de la cadena que es parte de una gran estructura. Cada objeto de

la cadena podría manejar el comando, pasar el comando al siguiente objeto de la cadena, o las dos cosas.

Patrón Command : Encapsula una operación en un objeto, permitiendo parametrizar operaciones, de tal forma que se pueda controlar su selección y secuencia, ponerlas en la cola, deshacerlas y manipularlas.

Patrón Little Language : Supón que necesitas resolver muchos problemas similares y te das cuenta de que las soluciones a estos problemas pueden ser expresadas como diferentes combinaciones de un pequeño número de elementos o operaciones. El modo más simple de expresar las soluciones a estos problemas podría ser definir un pequeño lenguaje. Los tipos comunes de problemas que se pueden solucionar con un pequeño lenguaje son búsquedas de estructuras de datos comunes, creación de complejas estructuras de datos, y asignación de formato a los datos.

Patrón Mediator : Define un objeto que encapsula la forma en la cual interactúan otros. Utiliza un objeto para coordinar cambios de estado entre otros objetos. Colocando la lógica en un objeto para manejar cambios de estado de otros objetos, en lugar de distribuir la lógica sobre los otros objetos, dando como resultado una implementación mas cohesiva de la lógica y decrementando las relaciones entre otros objetos. Promueve el acoplamiento débil.

Patrón Snapshot : Captura una instantánea del estado de un objeto pudiendo ser restaurado más tarde. El objeto que inicializa la captura o restauración del estado no necesita conocer ninguna información del estado. Sólo se necesita conocer cual es el objeto cuyo estado se va a restaurar o capturar implementando un interface particular.

Patrón Observer : Permite a los objetos captar dinámicamente las dependencias entre objetos, de tal forma que un objeto notificará a los objetos dependientes de él cuando cambia su estado, siendo actualizados automáticamente.

Patrón State : Encapsula los estados de un objeto como objetos separados, cada pertenencia en una subclase separada de una clase abstracta estado. Permite a un objeto cambiar su comportamiento cuando su estado interno cambia. El objeto parecerá haber cambiado de clase.

Patrón Null Object : Facilita una alternativa para utilizar null para indicar la ausencia de un objeto al que delegue una operación. Utilizando null para

indicar la ausencia de cada uno de los objetos requeridos se realiza una pregunta para ver si es null antes de cada llamada a los métodos de otros objetos. En lugar de utilizar null, el patrón Null Object utiliza una referencia a un objeto que no hace nada.

Patrón Patrón Strategy : Encapsula algoritmos relacionados en clases que son subclases de una superclase común. Esto permite la selección de un algoritmo que varia según el objeto y también le permite la variación en el tiempo.

Patrón Template Method : Escribe una clase abstracta que contiene parte de la lógica necesaria para realizar su finalidad. Organiza la clase de tal forma que sus métodos concretos llaman a un método abstracto donde la lógica buscada tendría que aparecer. Facilita la lógica buscada en métodos de subclases que sobrescriben a los métodos abstractos. Define un esqueleto de un algoritmo, delegando algunos pasos a las subclases. Permite redefinir ciertos pedazos del algoritmo sin cambiar su estructura.

Patrón Visitor : Un modo de implementar una operación que involucra a los objetos en una estructura compleja es facilitar la lógica en cada una de sus clases para soportar la operación. El patrón Visitor facilita un modo alternativo para implementar cada operación que evita complicar la estructura de las clases de los objetos colocando toda la lógica necesaria en una clase Visitor separada. El patrón Visitor también permite variar la lógica utilizando diferentes clases Visitor, es decir, permite definir nuevas operaciones sin cambiar las clases de los objetos sobre las cuales operan.

1.7. Metodología de la investigación

Para lograr el análisis del problema planteado, se utilizaran las siguientes metodologías de la investigación:

- Método de observación como procedimiento de diagnóstico hacia la problemática de la investigación, logrando obtener información como base que lleven a la contextualización del problema.
- Método analítico que obtiene las variables y datos que logran identificar la problemática de la investigación, las cuales llevan a un análisis exhaustivo que permitirá obtener una solución al problema

Fuentes y Técnicas para la recolección de la información

Para la recolección de la información del desarrollo de la investigación se apoyará en la revisión de registros para obtener la información de documentos que orienten a la identificación de la investigación.

1.8. Organización del trabajo de grado

Para completar los objetivos de la investigación se inicio realizando una búsqueda sobre las herramientas que se usan para automatización de pruebas y automatización, de las cuales se encontraron algunas como jUnit, selenium siendo esta ultima la herramienta que mas se adapta a nuestras necesidades ya que nos permite controlar remotamente el navegador y así automatizar las paginas web y se puede programar usando código java el cual es el lenguaje en que se desarrollara la herramienta, para la automatización de aplicaciones de escritorio se analizaron aplicaciones como autohotkey, autoit y sikuli siendo la ultima una de las que se acopla a nuestras necesidades y tiene un paquete de que se puede usar en java.

1.9. Estudio de sistemas previos

. SikuliX automatiza todo lo que ve en la pantalla de su computadora de escritorio con Windows, Mac o algunos Linux / Unix. Utiliza el reconocimiento de imágenes con tecnología OpenCV para identificar y controlar los componentes de la GUI. Esto es útil en los casos en que no hay un acceso fácil a los elementos internos de una GUI o al código fuente de la aplicación o página web en la que desea actuar. SikuliX admite como lenguajes de script

lenguaje Python nivel 2.7 (compatible con Jython) se admite la ejecución de scripts de texto RobotFramework nivel de lenguaje Ruby 1.9 y 2.0 (compatible con JRuby) JavaScript (compatible con Java Scripting Engine) y puede usarlo en programación Java y programación / scripting con cualquier lenguaje de programación / scripting compatible con Java (Jython, JRuby, Scala, Clojure,).[sikuli]. Aunque SikuliX actualmente no está disponible en

ningún dispositivo móvil, se puede usar con los emuladores respectivos en una computadora de escritorio o en soluciones VNC. Una solución para dispositivos Android basada en ADB (Android Debugging Bridge) se encuentra en una etapa experimental temprana.

PARTE II. DESARROLLO DE LA INVESTIGACIÓN

Capítulo 2

CAPÍTULO 2. DISEÑO

2.1. Fase de análisis

A continuación se describen roles y se define los requerimientos del sistema

Roles

Usuario: Hace referencia a el programado o desarrollador quien hará uso de las diferentes herramientas que estarán dentro del paquete.

Tarea: Parte de un proceso que se quiere automatizar

Requerimientos del sistema

- Como desarrollado deseo hacer: Una aplicación que me automatice una determinada tarea en una pagina web
- Como desarrollado deseo hacer: Una aplicación para automatizar una tarea en una aplicativo que se tiene que instalar en el equipo local.
- Como desarrollado deseo realizar: Incorporar este paquete de java a un proyecto mas grande para usarlo con mas herramientas y realizar una aplicación mas robusta

Casos de uso

La siguiente figura muestra a nivel general los casos de uso para la elaboración de una aplicación para automatizar una tarea especifica dentro de un proceso, una tarea que sea repetitiva y no requiere de mucho análisis bien sea una pagina web o un aplicativo de escritorio.

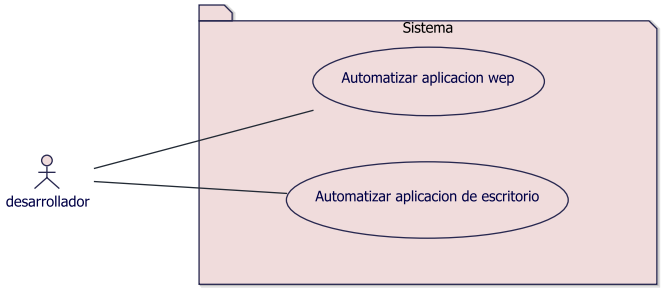


Figura 2.1: Caso de uso para la automatización de una tarea

2.2. Fase de diseño

Para el desarrollo de la herramienta se decidió usar el patrón adapter, después de analizar los patrones se concluyo que es uno de los que nos permitirá ir adaptando las herramientas de automatización y pruebas para obtener una compatibilidad entre ellas, lo que nos facilitara las cosas en un futuro cuando se deseen agregar nuevas funcionalidades sin tener que cambiar las que ya están desarrolladas.
A continuación se describen los métodos que componen las diferentes clases que conforman nuestra herramienta, iniciaremos con la clase padre que es donde se encuentran los método básicos y que se usan en conjunto con las herramientas de automatización y pruebas, para realizar el desarrollo de algunos métodos nos basamos en los ejemplos mostrados en [14] [13] [5].

```
1000  public void sendTex(String text) {
        screen.type(text);
1002  }
```

codigos/KeyboardCommands.java

El anterior método nos permite digitar el texto enviado como parámetro

```
1000  public void sendComand(int KeyEvent) throws
        InterruptedException {
        robot.keyPress(KeyEvent);
1002    Thread.sleep(100);
        robot.keyRelease(KeyEvent);
1004  }
```

codigos/KeyboardCommands.java

El anterior método nos permite presionar una tecla en especifico del teclado

```
1000  public void sendComand(int KeyEvent, int KeyEvent2) throws
        InterruptedException {
        robot.keyPress(KeyEvent);
1002    robot.keyPress(KeyEvent2);
        Thread.sleep(100);
1004    robot.keyRelease(KeyEvent);
        robot.keyRelease(KeyEvent2);
1006  }
```

codigos/KeyboardCommands.java

El anterior método nos permite enviar un comando presionando dos teclas al tiempo como por ejemplo Control + V;

```
1000  public void sendComand(int KeyEvent, int KeyEvent2, int
        KeyEvent3) throws InterruptedException {
        robot.keyPress(KeyEvent);
1002    robot.keyPress(KeyEvent2);
        robot.keyPress(KeyEvent3);
1004    Thread.sleep(100);
        robot.keyRelease(KeyEvent);
1006    robot.keyRelease(KeyEvent2);
        robot.keyRelease(KeyEvent3);
1008  }
```

codigos/KeyboardCommands.java

El anterior método nos permite presionar 3 teclas al tiempo como por ejemplo Control + alt+ supr.

```
1000  public void mouseMove(int x, int y) {
        robot.mouseMove(x, y);
1002  }
```

codigos/KeyboardCommands.java

El anterior comando nos permite mover el mouse a una posición determinada

```
1000  public void mouseWheel(int degrees) {
        robot.mouseWheel(degrees);
1002  }
```

codigos/KeyboardCommands.java

El anterior método permite mover la rueda del mause, recibe como parámetro los grados que se desea girar la rueda

```
1000  public void mousePress() {

1002    robot.mousePress(InputEvent.BUTTON3_DOWN_MASK);
        robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK);
1004  }
```

codigos/KeyboardCommands.java

El anterior método permite presionar la rueda del mause

```
1000  public void paste(String text) {
        screen.paste(text);
1002  }
```

codigos/KeyboardCommands.java

El anterior método permite pegar un texto enviado como para metro

A continuación se muestran los métodos para automatizar una aplicación web .

```
1000    * Retorna el driver de firefox con el que se manejara la
         pagina
        * @param paht direccion donde se encuentra el driver
1002    * Ejemplo C: \\ geckodriver.exe
        * @return
1004    */
      public WebDriver getDriverFirefox(String paht) {
1006    System.setProperty("webdriver.firefox.marionette",paht);
        return  new FirefoxDriver();
1008  }
```

codigos/ToolWep.java

Este método devuelve la instancia del driver con el que se podrá manipular una pagina en fireox, también se encuentran otro métodos que son getDiverChrome() y getDriverExplorer() para automatizar paginas wep en chrome e internetexplorer

```
1000   public WebDriver getDriverFirefox(String paht) {
         System.setProperty("webdriver.firefox.marionette",paht);
1002     return  new FirefoxDriver();
       }
1004
```

codigos/ToolWep.java

Este método se usa para abrir y ingresar la url en el navegador de la pagina que se va automatizar

```
1000   public void openBrowser(String url, WebDriver driver) {
         driver.get(url);
1002   }
```

codigos/ToolWep.java

Este método devuelve el texto que contiene un campo enviando el xpath de campo, se usa para extraer el texto de los formulario o de los botones

```
1000   public String getText(WebDriver driver, String xpathExpression
         ) {

1002     return driver.findElement(By.xpath(xpathExpression)).getText
         ();

1004   }
```

codigos/ToolWep.java

Este método envía texto a un campo usando el xpath, se usa para llenar los formularios

```
1000   public void sendText(WebDriver driver, String xpathExpression,
         String text) {

1002       driver.findElement(By.xpath(xpathExpression)).sendKeys(text
         );;

1004   }
```

codigos/ToolWep.java

Este método se usa para enviar comando de teclado sobre los botones de la pagina

```
1000  public void sendComand(WebDriver driver, String
        xpathExpression,String Keys ) {

1002    driver.findElement(By.xpath(xpathExpression)).sendKeys(Keys
      );

1004  }
```

codigos/ToolWep.java

Este método envía un click al elemento relacionado por un xpath

```
1000  public void sendClick(WebDriver driver, String xpathExpression
        ) {

1002    driver.findElement(By.xpath(xpathExpression)).click();

1004  }
```

codigos/ToolWep.java

Este método se usa para ingresar la dirección de un archivo que se desea subir a la pagina en el campo destinado para tal fin

```
1000  public void sendFile (WebDriver driver, String xpathExpression
        , String pathFile) {
        WebElement addFile=  driver.findElement(By.xpath(
      xpathExpression));
1002    addFile.clear();
        addFile.sendKeys(pathFile);
1004  }
```

codigos/ToolWep.java

Este método selecciona un dato de la lista desplegable por ejemplo una lista de departamentos.

```
1000  public void selectOption(WebDriver driver, String
        xpathExpression, String option) {

1002    Select dropdown = new Select(driver.findElement(By.xpath(
        xpathExpression)));
        dropdown.selectByVisibleText(option);
1004  }
```

codigos/ToolWep.java

Este método se usa para colocar tiempos espera hasta que aparezca el elemento y si no aparece en el tiempo indicado arroja una excepción

```
1000  public WebDriver waitElement(WebDriver driver, String
        xpathExpression, int time ) {
        driver.manage().timeouts().implicitlyWait(time, TimeUnit.
        MILLISECONDS);
1002    return driver;

1004  }
```

codigos/ToolWep.java

A continuación se muestran los métodos para automatizar una aplicación de escritorio

El siguiente método busca la imagen en una región de la pantalla indicado mediante las coordenadas requeridas, que encierren el lugar donde se quiere buscar la imagen en pantalla.

```
1000  public boolean clickSearchMagenRegionScreen(String fileImage,
        int clicks, int x, int y, int x1, int y1,
          int waitTime) throws FindFailed, InterruptedException {
1002
        Match target;
1004    region = new Region(x, y, x1 - x, y1 - y);

1006    for (int i = 0; i < waitTime * 10; i++) {
          target = region.exists(fileImage);
1008      if (target != null) {
            if (clicks == 1) {
1010          target.click();
            }
1012        if (clicks == 2) {
              target.doubleClick();
1014        }
            return true;
1016      }
          Thread.sleep(100);
1018    }
        return false;
1020
      }
```

codigos/ScreenSearch.java

El siguiente método busca la imagen en una región y le da click derecho

```
1000  public boolean clickrightSearchMagenRegionScreen(String
        fileImage, int x, int y, int x1, int y1, int waitTime)
          throws FindFailed, InterruptedException {
1002
        Match target;
1004    region = new Region(x, y, x1 - x, y1 - y);

1006    for (int i = 0; i < waitTime * 10; i++) {
          target = region.exists(fileImage);
1008      if (target != null) {

1010        target.rightClick();
            return true;
1012
          }
1014      Thread.sleep(100);
        }
1016    return false;
      }
```

codigos/ScreenSearch.java

El siguiente método busca la imagen en una región y si la encuentra devuelve el Match que contiene los método para interarticular sobre la imagen, como poner ejemplo click

```
1000  public Match SearchMagenRegionScreen(String fileImage, int x,
        int y, int x1, int y1, int waitTime)
          throws FindFailed {
1002
        Match target;
1004    region = new Region(x, y, x1 - x, y1 - y);

1006    for (int i = 0; i < waitTime + 10; i++) {
          target = region.exists(fileImage);
1008      if (target != null) {
            return target;
1010      }
        }
1012
        return null;
1014  }
```

codigos/ScreenSearch.java

El siguiente método busca la imagen en toda la pantalla y le da click si la encuentra

```
1000  public boolean clickSearchMagenScreen(String fileImage, int
        clicks, int waitTime)
          throws FindFailed, InterruptedException {
1002    Match target;

1004    for (int i = 0; i < waitTime * 10; i++) {
          target = screen.exists(fileImage);
1006      if (target != null) {
            if (clicks == 1) {
1008          target.click();
            }
1010        if (clicks == 2) {
              target.doubleClick();
1012        }
            return true;
1014      }
          Thread.sleep(100);
1016    }
        return false;
1018  }
```

codigos/ScreenSearch.java

El siguiente método busca la imagen en la pantalla y le da click derecho

```
1000
  public boolean clickrightSearchMagenScreen(String fileImage,
    int waitTime) throws FindFailed, InterruptedException {
1002
    Match target;
1004
    for (int i = 0; i < waitTime * 10; i++) {
1006  target = screen.exists(fileImage);
      if (target != null) {
1008
        target.rightClick();
1010    return true;

1012    }
        Thread.sleep(100);
1014  }
    return false;
1016  }
```

codigos/ScreenSearch.java

El siguiente método busca la imagen en la pantalla y si la encuentra devuelve el Match que contiene los métodos para interarticular sobre la imagen, como por ejemplo dar un click o sacar las coordenadas

```
1000  public Match SearchMagenScreen(String fileImage, int waitTime)
        throws FindFailed {

1002    Match target;

1004    for (int i = 0; i < waitTime + 10; i++) {
          target = screen.exists(fileImage);
1006      if (target != null) {
            return target;
1008      }
        }
1010
    return null;
1012  }
```

codigos/ScreenSearch.java

El siguiente método realiza un recorte de la pantalla definido por los parámetros requerido, que en este caso son los puntos de inicio y el ancho, el alto dado pixeles

```
1000   public BufferedImage getCapture(int x, int y, int x1, int y1)
         {
         BufferedImage g;
1002     region = new Region(x, y, x1 - x, y1 - y);
         ScreenImage image = screen.capture(region);
1004     return image.getImage();

1006   }
```

codigos/ScreenSearch.java

El siguiente método da click en las coordenada especificadas

```
1000   public void Click(int x, int y) throws FindFailed {
         screen.click(new Location(x, y));
1002   }
```

codigos/ScreenSearch.java

2.3. Fase de pruebas

Para la fase de pruebas se exploraron varias alternativas de paginas web ya que el objetivo era probar la herramienta automatizando una tarea que se pudieran realizar de dos maneras usando los métodos que son exclusivos para el manejo de paginas web y los que usan imágenes,coordenadas y comando de teclado. Se definió que la tarea automatizar es el envió de un correo usando gmail así no tendremos que usar paginas privadas o pedir algún tipo de permisos a la hora de mostrar los resultados

Para automatizar la tarea primero se hizo con los métodos exclusivos para paginas web donde se inspecciono la pagina para sacar los xpath de los elementos con los que la aplicación va interactuar

```
1000
       toolWep.openBrowser("https://mail.google.com/gmail/",
     driver);
1002     toolWep.sendText(driver, "//*[@id=\"identifierId\"]",
     EMAIL);
       toolWep.sendClick(driver, "//*[@id=\"identifierNext\"]/
     span/span");
```

codigos/App.java

Las anteriores lineas de código se usan para traer la instancia de la clase ToolsWep la cual contiene lo métodos que permiten automatizar una pagina

web, la segunda linea se usa para cargar el driver que nos permitirá abrir la pagina en el navegador chrome

```
1000        Thread.sleep(2000);
            toolWep.sendText(driver, "//*[@id=\'password\']/div[1]/div
        /div[1]/input", PASSWORD);
1002        toolWep.sendClick(driver, "//*[@id=\'passwordNext\']/span/
        span");
            Thread.sleep(3000);
1004        toolWep.sendClick(driver, "/html/body/div[7]/div[3]/div/
        div[2]/div[1]/div[1]/div[1]/div/div/div/div[1]/div/div");
            Thread.sleep(2000);
1006        toolWep.sendText(driver, "/html/body/div[25]/div/div/div/
        div[1]/div[3]/div[1]/div[1]/div/div/div/div[3]/div/div/div
        [4]/table/tbody/tr/td[2]/form/div[1]/table/tbody/tr[1]/td[2]/
        div/div/textarea", EMAIL);
            toolWep.sendText(driver, "/html/body/div[25]/div/div/div/
        div[1]/div[3]/div[1]/div[1]/div/div/div/div[3]/div/div/div
        [4]/table/tbody/tr/td[2]/form/div[3]/div/input", "prueba");
1008        toolWep.sendText(driver, "/html/body/div[25]/div/div/div/
        div[1]/div[3]/div[1]/div[1]/div/div/div/div[3]/div/div/div
        [4]/table/tbody/tr/td[2]/table/tbody/tr[1]/td/div/div[1]/div
        [2]/div[1]/div/table/tbody/tr/td[2]/div[2]/div", "prueba");
            toolWep.sendClick(driver, "/html/body/div[25]/div/div/div/
        div[1]/div[3]/div[1]/div[1]/div/div/div/div[3]/div/div/div
        [4]/table/tbody/tr/td[2]/table/tbody/tr[2]/td/div/div/div[4]/
        table/tbody/tr/td[1]/div/div[2]/div[1]");
1010
```

codigos/App.java

Las anteriores lineas son la automatizan la tarea, abre el navegador se loguea en el correo de gmail redacta un correo y lo envía. Para automatizar la tarea con los métodos que usan imágenes,coordenadas y comando de teclado se realizo el siguiente código.

```
1000    org.sikuli.script.App chrome = org.sikuli.script.App.open("
        \"C:\\Program Files (x86)\\Google\\Chrome\\Application\\
        chrome.exe\"");
        Thread.sleep(3000);
1002    screen.paste("https://mail.google.com/gmail/");
        screen.sendComand(KeyEvent.VK_ENTER);
1004    Thread.sleep(4000);
        if (screen.SearchMagenRegionScreen(RUTA_IMAGE + "nuevo.PNG"
        ,0,100,120,270,0)==null) {
1006        System.out.println("Buscando imagen usarOtraCuenta");
```

codigos/App.java

En la primera linea se trae la instancia de la clase ScreenSearch la cual contiene los métodos con que facilita la búsqueda de imágenes en la pantalla, las segunda linea se utiliza para ejecutar el navegador, en la tercera linea se instancia la clase KeyboardCommands la cual contiene los métodos que facilitan el envió de comando de teclado y clicks en la pantalla, las siguientes lineas se realiza una espera hasta que habrá el navegador se envía la dirección url y se presiona la tecla enter para ingresar a la pagina de gmail.

```
1000        Match target = screen.SearchMagenScreen(RUTA_IMAGE +"
    siguiente.PNG",5);
            screen.paste(EMAIL);
1002        target.click();
            Thread.sleep(2000);
1004        target = screen.SearchMagenScreen(RUTA_IMAGE +"siguiente.
    PNG",10);
            Thread.sleep(1000);
1006        screen.paste(PASSWORD);
            target.click();
1008    }
        System.out.println("Buscando imagen nuevo");
1010    screen.SearchMagenScreen(RUTA_IMAGE+"nuevo.PNG",5).click();
        screen.SearchMagenScreen(RUTA_IMAGE+"mensajeNuevo.PNG",5);
```

codigos/App.java

En las anteriores lineas de código se realiza e el loqueo en gmail, en la primera linea se pregunta si existe nuevo.PNG si esto es verdadero significa que el gmail ya estaba abierto sino inicia inicio de sesión

```
1000    Thread.sleep(1000);
        screen.sendComand(KeyEvent.VK_ENTER);
1002    Thread.sleep(100);
        screen.sendComand(KeyEvent.VK_ENTER);
1004    Thread.sleep(1000);
        screen.sendComand(KeyEvent.VK_TAB);
1006    Thread.sleep(1000);
        screen.paste("PRUEBA");
1008    Thread.sleep(1000);
        screen.sendComand(KeyEvent.VK_TAB);
1010    Thread.sleep(1000);
        screen.paste("esto es una prueba");
1012    screen.clickSearchMagenScreen(RUTA_IMAGE+"enviar.PNG",1,10);
    }
1014}
```

codigos/App.java

Una vez logueado se procede a redactar el correo el correo y se realiza en envió, como se evidencia en los códigos anteriormente mostrados se puede observar como automatizo una tarea de dos manera diferentes, con este sencillo ejemplo podemos observar que la herramienta funciona que puede realizarse en una pocas lineas de código y que la tarea se puede realizar mas rápido que una persona.

Al inicio de la pruebas con los métodos para el manejo de paginas web se presentaron inconvenientes con los tiempos y el diseño de algunos métodos los cuales se fueron corrigiendo conforme se realizaban las pruebas.

Al probar la aplicación que automatiza la tarea usando herramientas para automatizar aplicaciones de escritorio se observo que es mucho mas complejo desarrollarla, es mas tedioso ya que hay que tomar recortes de pantalla y coordenadas de los elementos para poder interactuar con ellos, también se encontró que se gasta mas tiempo ejecutando la tarea en comparación con la que se creo con la otra tecnología.

Al comparar los tiempos para ejecutar la tarea de manera manual y con las dos aplicaciones se obtuvieron los siguientes resultados: De manera manual se obtuvo un promedio fue de 54.53 segundos, usado métodos para escritorio 39.92 segundos y con lo métodos para paginas web 28.1 segundos. lo que demuestra que se puede realizar la tarea me manera mas rápida usando las aplicaciones desarrolladas con el paquete java.

No ejecución	T.Manual	T.Web	T. Escritorio
1	55	28,3	40,2
2	55,3	28,6	39,5
3	53,4	28,5	39,7
4	56,1	27,8	40,3
5	55,1	27,9	39,5
6	55,9	27,7	41,1
7	54,3	27,4	39,3
8	53,5	28,9	39
9	54,6	27,9	40,1
10	52,1	28	40,5

Cuadro 2.1: Tabla tiempos para cada ejecución.

Capítulo 3

CAPITULO 3. ADM-ARCHIMATE

Archimate 2.1 facilita el modelado de la arquitectura empresarial, permitiendo bosquejar la organización desde diversos puntos de vista donde se reflejan tanto los involucrados en el negocio, la infraestructura y la misma solución como los diferentes componentes, dando una visión general que describe a la perfección la organización y su solución informática [2].

3.1. Organización

Modelo

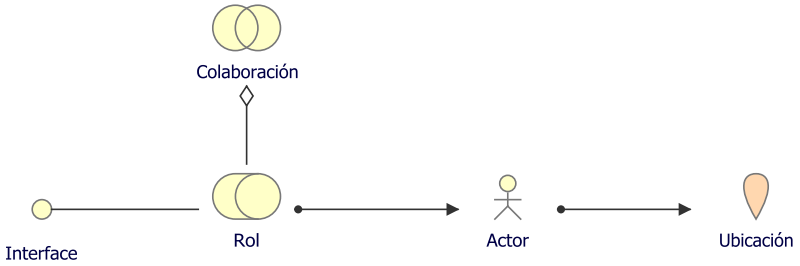


Figura 3.1: Modelo punto de vista Organizacional

El modelo de organización tiene como objeto identificar actores y roles, así como la ubicación de lo mismos en un espacio geográfico o espacio físico organizacional, también se involucran interfaces que nos representan los medios de interacción de la organización con el contexto. Generalmente este punto de vista se nutre del organigrama de la empresa en donde se puede identificar con claridad el nivel jerárquico de actores y roles. Es posible reunir un conjunto de roles y/o actores dentro de colaboración con el objeto de clarificar la estructura cooperativa que tiene la organización.

Modelo

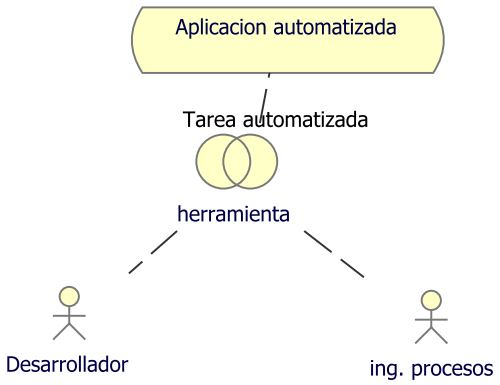


Figura 3.2: Caso de estudio punto de vista Organizacional. Fuente los autores

3.2. Punto de vista cooperación de actor

Modelo

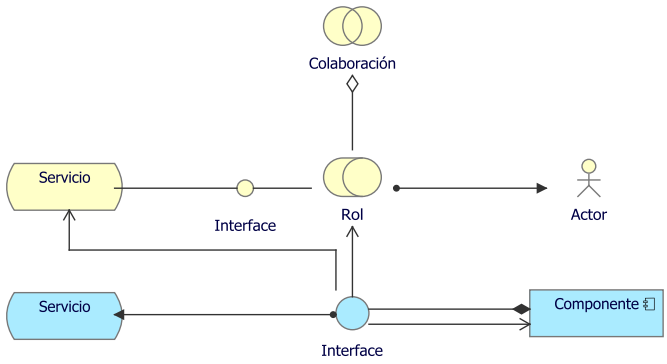


Figura 3.3: Modelo punto de vista cooperación de actor.

Este punto de vista presenta los roles implicados y la forma cooperativa de los mismos junto con los servicios que exponen el producto al cliente, es posible integrar elementos de aplicación a través de interfaces con sus componentes y sus posibles servicios de aplicación.

Caso

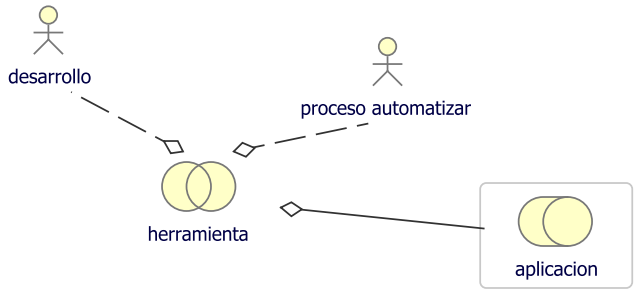


Figura 3.4: Caso estudio punto de vista cooperación de actor. Fuente: los autores

La cooperación del actor desarrollador esta conformada por métodos que proporciona la herramienta en conjunto con el proceso automatizar para tener como producto final la aplicación automatizada

3.3. Punto de vista de función del negocio

Modelo

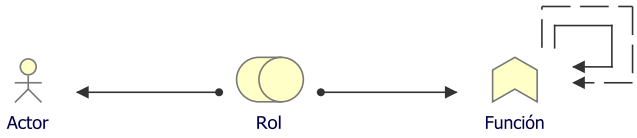


Figura 3.5: Modelo punto de vista de función del negocio.

El punto de vista de la función empresarial muestra las principales funciones empresariales de una organización y sus relaciones en términos de los flujos de información, valor o bienes entre ellos. Negocio Las funciones se utilizan para representar los aspectos mas estables de una empresa en términos de los principales actividades que realiza, independientemente de los cambios organizacionales o desarrollos tecnológicos. Por lo tanto, la arquitectura de la función comercial de las empresas que operan en el mismo mercado a menudo exhibir similitudes cercanas. El punto de vista de la función comercial

proporciona, por lo tanto, una visión de alto nivel en las operaciones generales de la empresa, y puede utilizarse para identificar las competencias necesarias o para estructurar una organización de acuerdo con sus actividades principales.

Caso

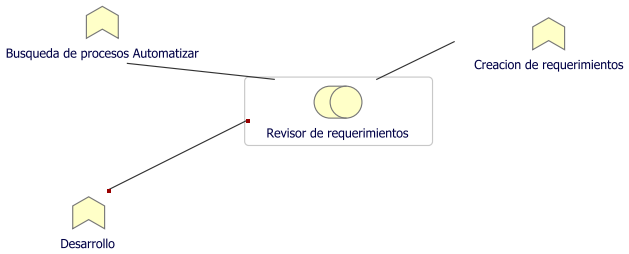


Figura 3.6: Caso estudio punto de vista de función del negocio. Fuente: los autores

la herramienta de automatización esta en cabeza de los desarrolladores quienes deben tener un perfil con conocimientos junior en java y el personal de procesos debe tener los conocimientos adecuados del proceso para saber que tareas se pueden automatizar, las funciones del desarrollador son:

1. Estudiar el manual entregado por el área de procesos
2. Identificar los mejores métodos proporcionados por la herramienta para desarrollar la aplicación que automatizara la tarea

Las funciones de área de procesos son:

1. Conocer el proceso que se va automatizar
2. Redactar los manuales para el desarrollador

3.4. Punto de vista proceso del negocio

Este punto de vista visualiza una estructura de alto nivel y la composición de uno o mas procesos de negocio. Permite mostrar los servicios a través de los procesos que contribuyen al producto final de la organización, así como los roles en los procesos y las responsabilidades de los actores implicados [2].

Modelo

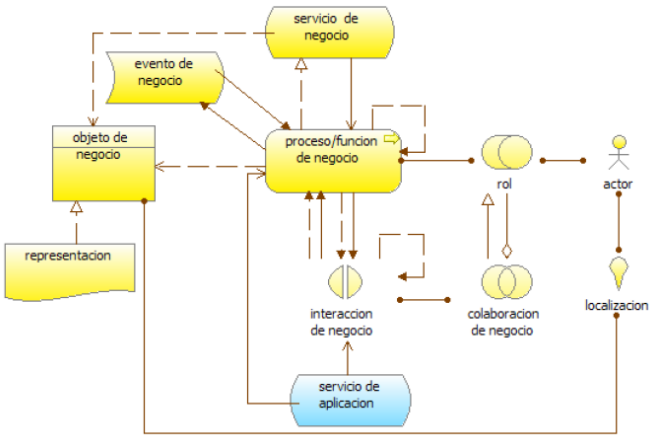


Figura 3.7: Modelo punto de vista proceso del negocio.

Caso

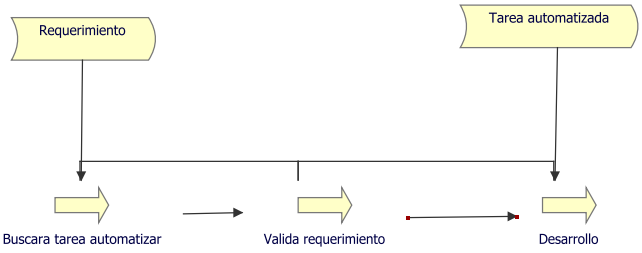


Figura 3.8: Caso estudio punto de vista proceso del negocio. Fuente: los autores

3.5. Punto de vista cooperación proceso del negocio

Este punto de vista visualiza la relación entre los procesos de negocio y su entorno, así como las dependencias entre estos mismos. Sus características son:

1. Presenta relaciones causales entre los procesos de negocio de la organización.
2. Visualiza una correlación entre los procesos de negocio hacia las funciones de negocio.
3. Visualiza la relación de servicios de cada proceso de negocio.
4. Uso de datos compartidos.

Modelo

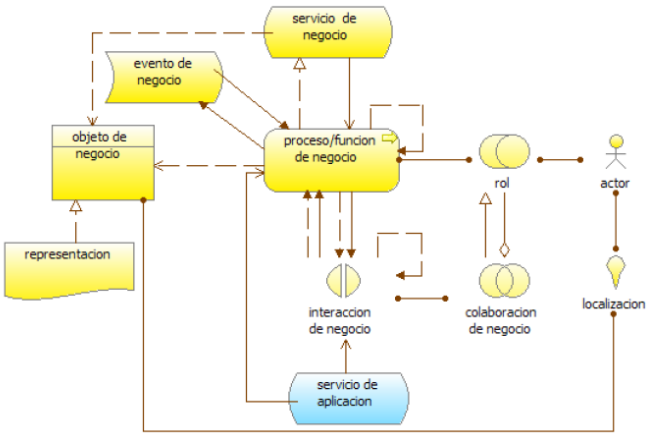


Figura 3.9: Modelo punto de vista cooperación proceso del negocio.

Caso

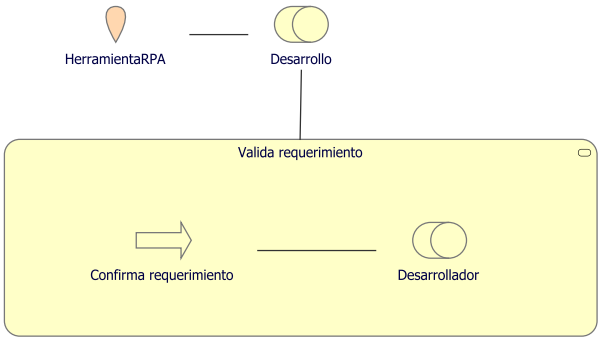


Figura 3.10: Caso estudio punto de vista cooperación proceso del negocio. Fuente: los autores

3.6. Punto de vista de producto

Este punto de vista visualiza el valor de los productos hacia el cliente o las partes externas implicadas, muestra la composición de los productos en función de los servicios, contratos o acuerdos. Visualiza las interfaces en las que el producto es presentado y los eventos asociados con el producto.

Modelo

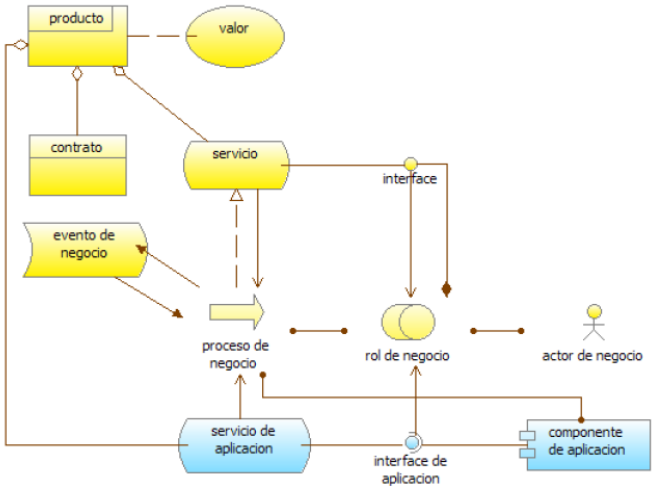


Figura 3.11: Modelo punto de vista de producto.

Caso

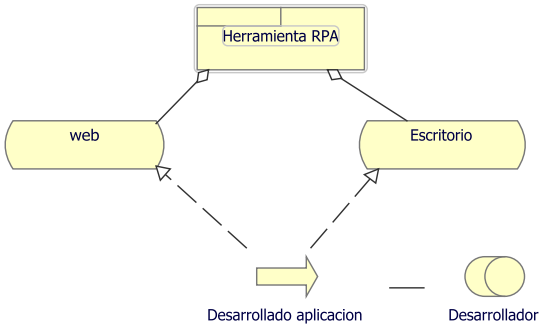


Figura 3.12: Caso estudio punto de vista de producto.

3.7. Punto de vista de la aplicación

Los puntos de vista de la aplicación visualizan y definen las aplicaciones de software que gestionan los componentes del negocio y los servicios, los componentes reusables, así como las interfaces que relacionan y comunican los componentes del negocio.

3.8. Punto de vista de comportamiento de aplicación

Este punto de vista visualiza y define el comportamiento a nivel interno de las aplicaciones, se utiliza para modelar y diseñar el comportamiento de las aplicaciones o componentes y lograr identificar el proceso funcional entre las aplicaciones.

Modelo

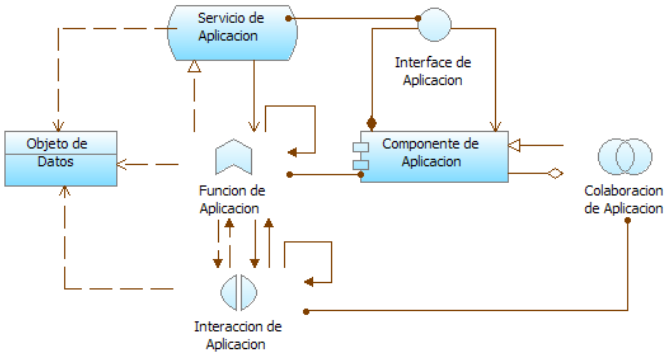


Figura 3.13: Modelo punto de vista de comportamiento de aplicación.

Caso

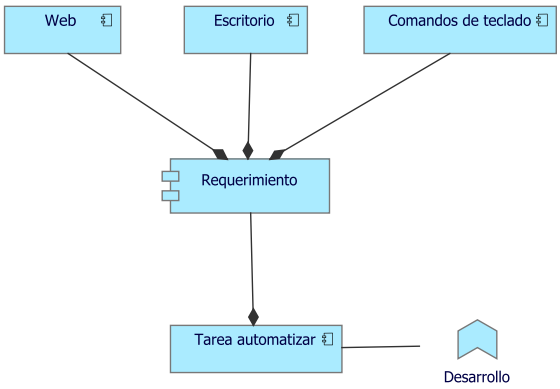


Figura 3.14: Caso estudio punto de vista de comportamiento de aplicación.
Fuente: los autores

3.9. Punto de vista de cooperación de aplicación

Este punto de vista visualiza y define las relaciones de los componentes a través del flujo de información de los servicios ofrecidos o usados por la organización. Permite visualizar la cooperación de los servicios que se unen para la ejecución de los procesos del negocio.

Modelo

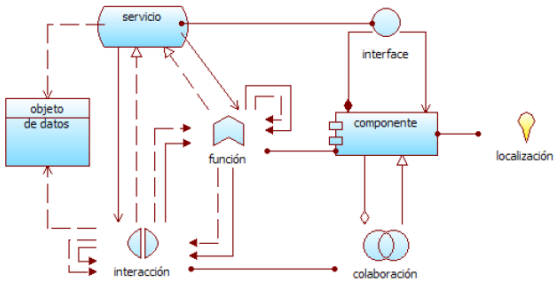


Figura 3.15: Modelo punto de vista de cooperación de aplicación.

Caso

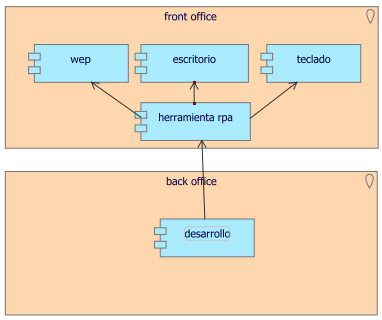


Figura 3.16: Caso estudio punto de vista de cooperación de aplicación. Fuente: los autores

3.10. Punto de vista de estructura de aplicación

Este punto de vista visualiza la estructura de las aplicaciones o de los componentes, permite modelar y diseñar estas estructuras de las aplicaciones o componentes y su información asociada.

Modelo

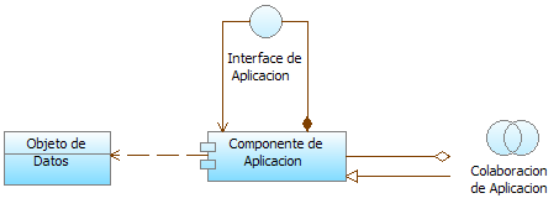


Figura 3.17: Modelo punto de vista de estructura de aplicación.

Caso

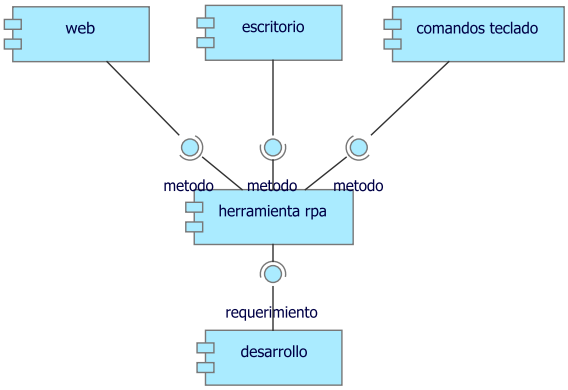


Figura 3.18: Caso estudio punto de vista de estructura de aplicación. Fuente: los autores

3.11. Punto de vista de uso de aplicación

Este punto de vista visualiza y define el uso de las aplicaciones para soportar los procesos del negocio, y como se deben utilizar por otras aplicaciones. Permite modelar y diseñar los servicios de los procesos de negocio y demás aplicaciones de la organización.

Modelo

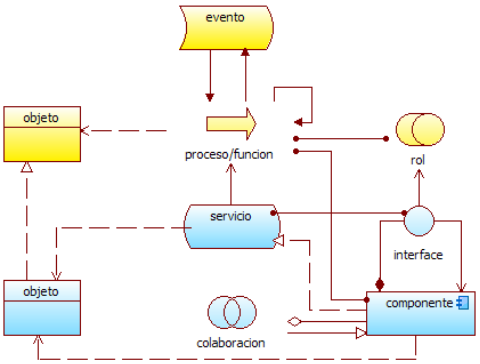


Figura 3.19: Modelo punto de vista de uso de aplicación.

Caso

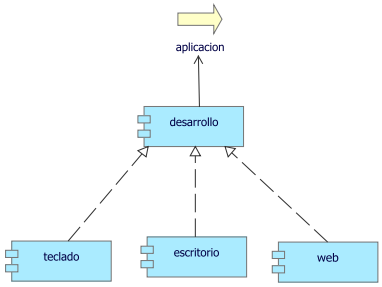


Figura 3.20: Caso estudio punto de vista de uso de aplicación. Fuente: los autores

3.12. Punto de vista tecnológico

Los puntos de vista tecnológicos visualizan y definen la infraestructura que soporta y permite la comunicación de la capa de aplicación, dispone de servicios de infraestructura hardware y software para el despliegue de las aplicaciones en la organización.

3.12.1. Punto de vista de infraestructura

Este punto de vista visualiza y define el software y hardware que soportan las aplicaciones, a través de dispositivos físicos, redes o sistemas de software.

Modelo

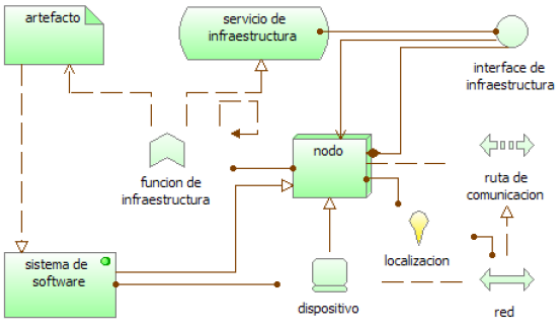


Figura 3.21: Modelo punto de vista de infraestructura.

Caso

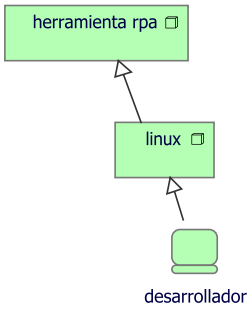


Figura 3.22: Modelo punto de vista de infraestructura.Fuente: los autores

3.13. Punto de vista motivacional

3.13.1. Punto de vista de los skateholder

Este punto de vista visualiza y define los stakeholders, los manejadores internos y externos de cambio, sus fortalezas, debilidades, oportunidades y amenazas (DOFA). Permite el proceso de ingeniería de requerimientos, el refinamiento de objetivos, contribución y análisis de conflictos derivados de los objetivos.

Modelo

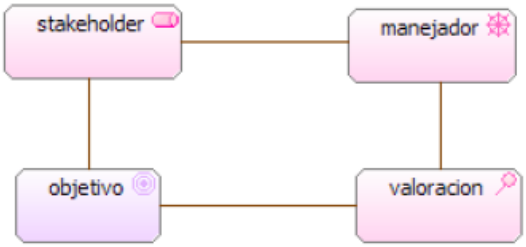


Figura 3.23: Modelo punto de vista de los skateholder

Caso

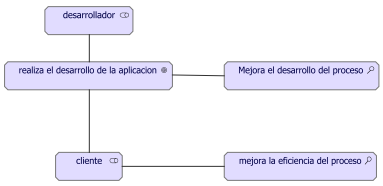


Figura 3.24: Caso estudio punto de vista de los skateholder. Fuente: los autores

3.13.2. Punto de vista de realización de objetivos

Este punto de vista visualiza y define el proceso de mejorar los objetivos en objetivos mas concretos que permitan generar requerimientos o restricciones que ayuden en el cumplimiento de los objetivos planteados. Estas mejoras se modelan a través de la relación de realización.

Modelo

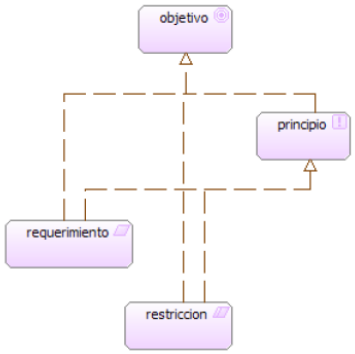


Figura 3.25: Modelo punto de vista de realización de objetivos

Caso

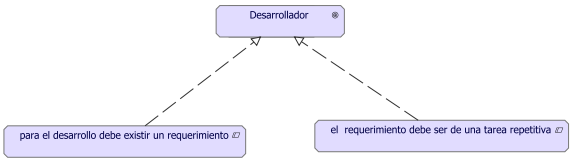


Figura 3.26: Caso estudio punto de vista de realización de objetivos. Fuente: los autores

3.13.3. Punto de vista de principios

Este punto de vista visualiza y define los procesos mas importantes para el diseño a mano y los objetivos que conllevan a estos principios. Además permite modelar las relaciones de los principios con sus objetivos.

Modelo

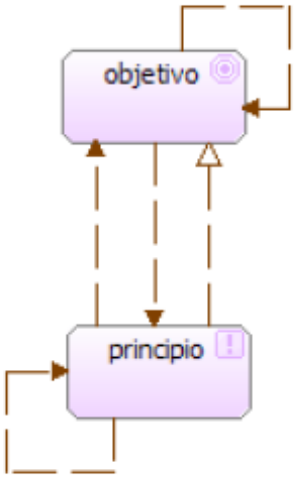


Figura 3.27: Modelo punto de vista de principios.

Caso

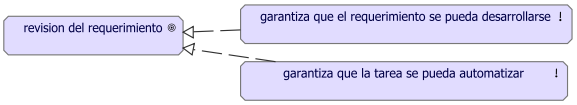


Figura 3.28: Caso estudio punto de vista de principios. Fuente: los autores

3.13.4. Punto de vista de realización de requerimientos

Este punto de vista visualiza y define la relación de requerimientos con los actores de negocio, los servicios, procesos y aplicaciones de negocio.

Modelo

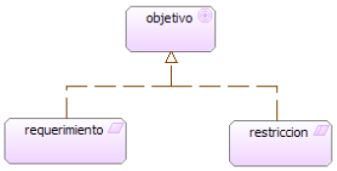


Figura 3.29: Modelo punto de vista de realización de requerimientos.

Caso

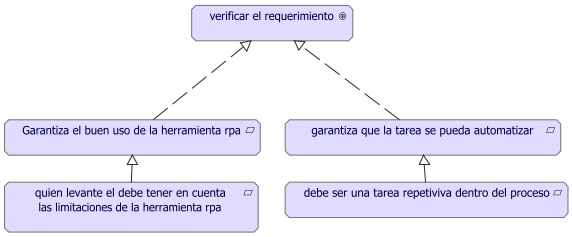


Figura 3.30: Caso estudio punto de vista de realización de requerimientos. Fuente: los autores

3.13.5. Punto de vista de motivación

Este punto de vista visualiza y define la revisión de los aspectos motivacionales relacionados con los stakeholders, sus principios primarios, aplicados y los requerimientos mas relevantes de los servicios, procesos, aplicaciones y objetivos del negocio.

Modelo

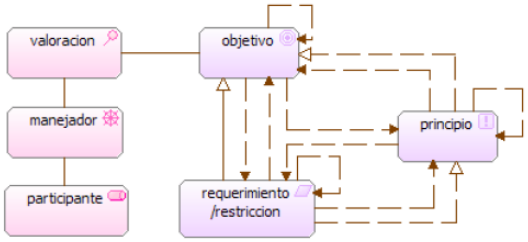


Figura 3.31: Modelo punto de vista de motivación

Caso

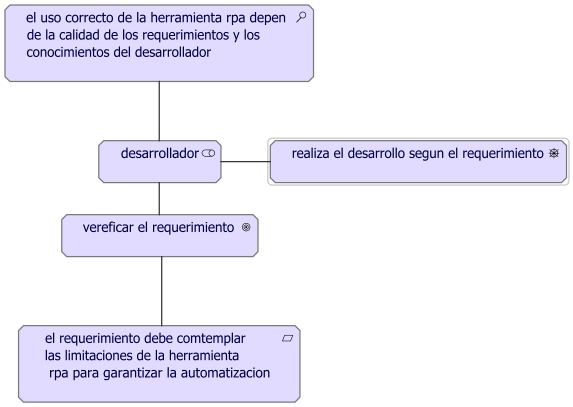


Figura 3.32: Caso estudio Modelo punto de vista de motivación. Fuente: los autores

3.14. Punto de vista implementación y migración

3.14.1. Modelo punto de vista de proyecto.

Este punto de vista visualiza y define los cambios en la arquitectura del proceso de migración desde un estado actual de la arquitectura empresarial a un estado objetivo de la arquitectura empresarial, esta migración aporta de forma significativa en el proceso de crecimiento de la estrategia y las decisiones de los procesos de realización [2].

Modelo

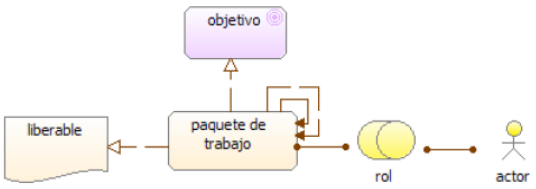


Figura 3.33: Modelo punto de vista de proyecto.

Caso

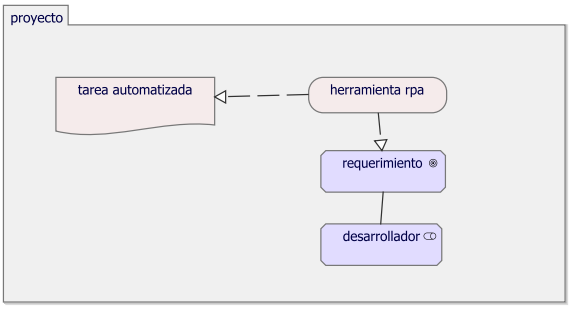


Figura 3.34: Caso estudio Modelo punto de vista de proyecto. Fuente: los autores

3.14.2. Modelo punto de vista de proyecto.

Este punto de vista visualiza y define las características para la transición de una arquitectura existente a una arquitectura deseada. La platea es el estado de la arquitectura que tiene un un tiempo limitado.

Modelo

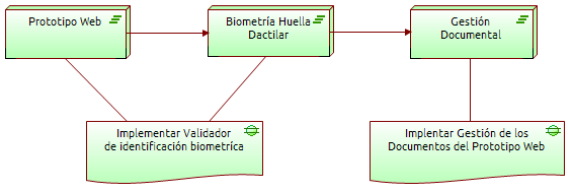


Figura 3.35: Modelo punto de vista de migración.

Caso

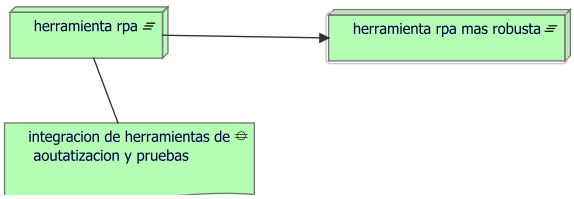


Figura 3.36: Modelo punto de vista de migración. Fuente: los autores

3.14.3. Punto de vista de migración e implementación.

Modelo

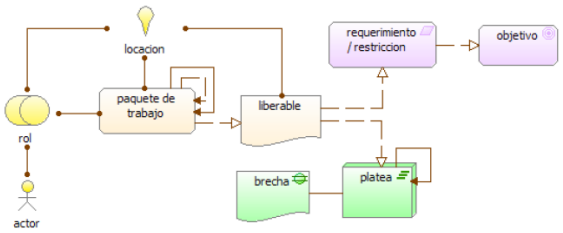


Figura 3.37: Modelo punto de vista de migración e implementación.

Caso

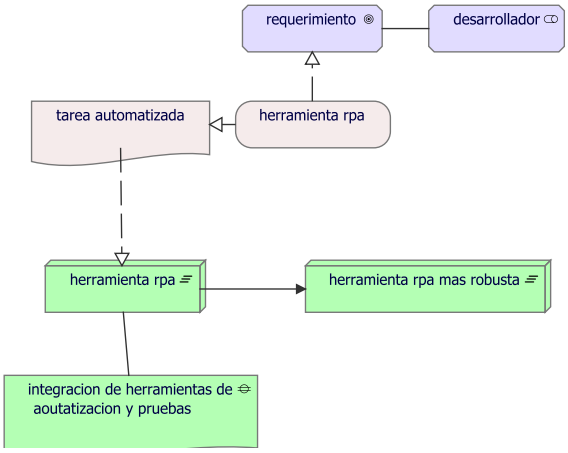


Figura 3.38: Modelo punto de vista de migración. Fuente: los autores

PARTE III. CIERRE DE LA INVESTIGACIÓN

Capítulo 4

CAPITULO 4. RESULTADOS Y DISCUSIÓN

4.1. Recolección, análisis y presentación de información

La automatización de procesos (RPA) ha sido reconocida como el motor transformador para automatizar los procesos de negocio lo que antes parecía inalcanzable debido al costo y los recursos de TI. Hoy en día, los robot de software se diseñan y se despliegan de manera rápida para automatizar tareas repetitivas y estructuradas [1].

Es importante recordar que las aplicaciones de RPA requieren de la intervención humana para su desarrollo, esto debido a que requieren de reglas específicas para poder llevar a cabo sus tareas. Esto habla de una de las limitantes más grandes de RPA, que es que no pueden llevar a cabo actividades que requieran un toma de decisiones. Aun así, su valor no debe ser minimizado, ya que al ejecutar procesos o actividades transaccionales o repetitivas, con alta precisión, permiten que las personas se puedan dedicar a labores mas estratégicas.

A pesar de ser tecnología emergente, los RPAs ya empiezan a encontrar uso en países desarrollados. En el Reino Unido, diversas empresas han empezado a implementar su uso. En una encuesta realizada a diferentes empresarios prominentes, se determinó que hay interés en aplicar RPA a los procesos de negocio más transaccionales como lo son cuentas por pagar y procesamiento de gastos de viaje. Los resultados de la encuesta se muestran a continuación:

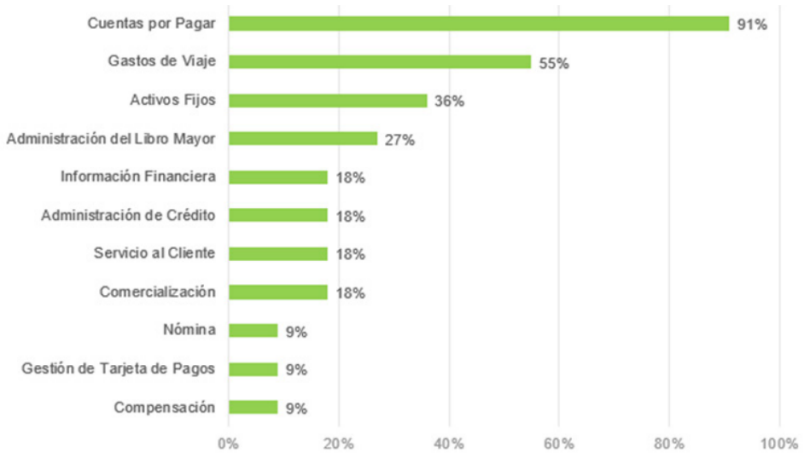


Figura 4.1: Resultados de encuesta de implementación 2015-2017. % de encuestados Fuente: [3]

Para poder implementar un RPA dentro de las funciones de negocio, las organizaciones deberán adaptar tanto su estructura tecnológica como su estructura humana. Deben modificar los roles actuales y adoptar e implementar nuevos. En términos de organización, es necesario contar con 5 actores que permitirán obtener los mejores resultados:

- Desarrolladores de procesos que indiquen las tareas que debe desarrollar el RPA
- Un gestor de robots que asigne y monitoree tareas local.
- El robot que es el software instalado en el ambiente de trabajo e interactúa directamente con las aplicaciones del negocio.
- Los usuarios son aquellos que resuelven las incidencias o situaciones que el robot escala
- La aplicación o plataforma mediante la cual el robot interactúa con el usuario

Beneficio de la RPA

- Pueden ejecutar tareas de manera precisa 24x7, lo cual incrementa la capacidad del procesamiento de los procesos.

- Estandariza y optimiza proceso, mejorando la calidad y el costo de entrega.
- La automatización de procesos ayuda a liberar tiempo para que el talento humano se pueda desarrollar en nuevas competencias que mejoren el negocio.
- El retorno de inversión es bajo y puede ser implementado con bajos costo de integración
- Si un solo RPA no puede procesar todas tareas pendientes puede replicar fácilmente, usando maquinas virtuales por ejemplo

Las ganancias financieras a corto plazo no son los únicos beneficios de RPA.La Automatización Robótica de Procesos ayuda a aumentar la satisfacción del cliente con una mejor calidad de datos, una resolución más rápida de problemas, etc. Es una verdadera palanca para transformar la experiencia del cliente y optimizar una estrategia de Relación Cliente.

La Automatización Robótica de Procesos también tiene beneficios para los empleados. El aumento de la burocracia conlleva una carga de trabajo adicional, no muy gratificante. El robot se encarga de lo que a los humanos no les gusta hacer. Así, la automatización alivia el estrés negativo. Eficiencia, velocidad, calidad, cumplimiento, ahorro, disponibilidad; Esos son los beneficios del ARP.

La RPA debe llevar a las empresas a repensar completamente su modelo de negocio. La tecnología debe empujarlos a construir centros de excelencia que sean la fuente de nuevas técnicas y procesos necesarios para su desarrollo. Hay que pensar a largo plazo: los oficios que habrán desaparecido serán los menos interesantes para el hombre. El RPA significa que los empleados tendrán un trabajo más gratificante

Los escenarios de ARP son variados. Algunas compañías lo utilizan para brindar un mejor servicio al cliente al automatizar procesos como la verificación de la información, la administración de reclamaciones y la respuesta automática por correo electrónico. Otros lo utilizan para automatizar las tareas de contabilidad, optimizar su proceso de facturación o crear pruebas de cumplimiento. Por ejemplo, Deutsch Bank ha automatizado su proceso de validación para la solvencia crediticia de un cliente adherente.

La automatización de procesos por robot se está acercando rápidamente al mundo de los negocios y está alterando los métodos actuales. Estamos en umbral de un cambio en las organizaciones, y cuanto antes se adapten las empresas, más ganarán una ventaja competitiva en su mercado. Sin embargo, el RPA debe estar perfectamente integrado en la estrategia de la empresa, ya que plantea problemas de recursos humanos. Finalmente, al apoyarse en la inteligencia artificial y el machine learning, las herramientas de automatización tendrán la capacidad de adaptarse a un contexto, a un problema particular, activando el proceso adaptado.

Hoy en día las operaciones comerciales están avanzando hacia el nuevo terreno tecnológico, el llamado "Proceso Robótico.Automatización. "Por lo tanto, el año"2018 "se conoce como el año de la.Automatización de procesos robóticos". La industria está entrando en el nuevo mundo de la tecnología Robotic Process Automation". RPA". La adopción oportuna de RPA se ha convertido en una necesidad en las actividades comerciales diarias. Si las empresas no configuran esta tecnología en sus operaciones, es posible que no se sostengan en un negocio.

Capítulo 5

CAPITULO 5. CONCLUSIONES

5.1. Verificación, contraste y evaluación de los objetivos

A continuación se presentan algunas reflexiones que se generaron a lo largo del desarrollo del proyecto y del análisis de sus objetivos.

Hoy en día las personas tienen que ser mas eficientes y los proceso se deben realizar con menos costos y en menor tiempo, la RPA juega un papel muy importante, integrar estas tecnología en el proceso hará que la empresa sea mas competitiva y así mantenerse en el mercado, empresa que no adopte la automatización de procesos muy probablemente estará destinada a desaparecer.

Contar con una herramienta tecnológica que facilite la creación de aplicaciones para automatizar tareas repetitivas que forman parte de un proceso contribuye a que este pueda realizarse de manera mas rápida, sin fallas humanas, se puede operar las 24 horas del día sin descansos o pausas, los que si necesita una persona, disponible para trabajar las 24 horas del día, añadiendo que puede operar en entornos virtuales y no ocupa un espacio físico en la empresa.

Al realizar las respectivas comparaciones para la tarea que se uso como prueba en este proyecto usando los datos que se muestran en la tabla 2.1 donde se tabularon los tiempos que gastaran para realizar la tarea en 10 repeticiones dando como resultado:De manera manual se obtuvo un promedio fue de

54.53 segundos, usado métodos para escritorio 39.92 segundos y con lo métodos para paginas web 28.1 segundos. Lo que demuestra que para este caso las aplicaciones fueron mas rápidas que una persona y esto sin mencionar que las personas se cansan con el tiempo y el proceso se empieza a realizar de manera mas lenta, lo que da ha entender que la RPA tiene mucho futuro.

5.1.1. Síntesis del modelo propuesto

Se desarrolla un paquete java en el que se incluyen herramientas de pruebas y automatización con el fin de facilitar el desarrollo de aplicaciones RPA.

5.1.2. Aportes originales

El paquete java que se desarrollo contiene herramientas de pruebas y automatización como sikuli y selenium, y una arquitectura con un patrón de desarrollo adapter que facilita la integración de nuevas herramientas, lo que hace que en un futuro se pueda seguir desarrollando sin modificar lo que ya esta echo con el objetivo de que se pueda convertir en una herramienta robusta que facilite a un mas el desarrollo de las aplicaciones RPA.

5.1.3. Trabajos o Publicaciones derivadas

Teniendo en cuenta que esta herramienta solo contiene lo esencial para desarrollar una aplicación RPA se le pueden seguir implementando funcionalidades como Reconocimiento óptico de caracteres (OCR) para leer texto de las imágenes, conexión a base de datos, desarrollo de micro servicios.

Capítulo 6

CAPITULO 6.PROSPECTIVA DEL TRABAJO DE GRADO

6.1. Lineas de investigación futuras

Como punto de partida de esta herramienta es posible ver otras alternativas que permitan generar nuevas lineas de investigación como las siguientes.

- Inteligencia artificial en la automatización de procesos.
- AUTOMATIZACIÓN COGNITIVA: Robots que automatizan tareas implicando intuición, razonamiento o resolución de problemas. Imitan la Inteligencia humana [1]

6.2. Trabajos de investigación futuros

Crear una herramienta usando inteligencia artificial y que se integre con con las tecnologías de automatización de pruebas para no se tenga que realizar un desarrollo de las reglas y los métodos a utilizar si no un entrenamiento con datos del desarrollo del proceso.

Bibliografía

[1] Abbyy. La inteligencia de contenido para la automatización robótica de procesos (rpa). Tech. rep.

[2] Amilcar Arnoldo Rojas portilla, J. S. R. Prototipo web de identificacion para los ciudadanos de bogota a traves de huellas dactilares, 2018.

[3] Deloitte. Automatizaciotica de procesos (rpa). 15.

[4] https://www.efficy.com. Todo sobre la arp o automatizacion robotica de procesos, 2019.

[5] https://www.toolsqa.com/. Selenium webdriver switch window commands, 2019.

[6] Iturralde, O. J. B. Introduccióna los patrones de diseño, 2016.

[7] Marrugo-salas, L. Contabilidad y Automatizacion Robotica de Procesos: Una nueva perspectiva de la profesion contable, 2019.

[8] R, I. G. Que trabajen los robots ! la automatización robótica de procesos (rpa)), 2019.

[9] Sikuli. https://www.softwaretestinghelp.com/sikuli-tutorial-part-1/.

[10] Tokyo, S., Documentaci, P. D., Acerca, A., Conjunto, S., Eligiendo, S., and Ide, S. Introduccion Automatizacion de pruebas para aplicaciones web Presentando Selenium Breve historia del proyecto Selenium, 2019.

[11] www.emergys.com.mx. Automatización robótica de procesos (rpa).

[12] www.ey.com. Automatización de procesos a través de robots.

[13] WWW.PROGRAMCREEK.COM. Java code examples for org.openqa.selenium.webdriver, 2019.

[14] WWW.PROGRAMCREEK.COM. Java code examples for org.sikuli.script.screen, 2019.