



**UNIVERSIDAD JOSE ANTONIO PAEZ**

**DESARROLLO DE APLICACIÓN  
INTERACTIVA CON INTELIGENCIA  
ARTIFICIAL PARA LA VALIDACIÓN  
DE FORMULARIOS WEB**

**Autores:** Ana Gabriela Guzmán Padilla.

C.I: 21.216.201.

José Vidal Miranda Yaguarin

C.I: 20.383.873.

Urb. Yuma II, Calle No 3. Municipio San Diego  
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE COMPUTACIÓN**

**DESARROLLO DE UNA APLICACIÓN INTERACTIVA  
CON INTELIGENCIA ARTIFICIAL PARA LA VALIDACIÓN DE  
FORMULARIOS WEB**

**Trabajo de grado presentado como requisito para optar al título de  
INGENIERO DE COMPUTACIÓN**

**Autores:** Ana Gabriela Guzmán Padilla.

C.I: 21.216.201.

José Vidal Miranda Yaguarin

C.I: 20.383.873.

**Tutor:** Ing. Belkys Araujo

San Diego, Julio 2018



Universidad José Antonio Páez  
Facultad de Ingeniería

FI -C-015-2018-IICR

Valencia, 06 de Noviembre de 2018

Ciudadano:  
Ana Guzmán  
C.I: 21.216.201  
Jose Miranda  
C.I:20.383.873  
Presente.-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 01-2018 de fecha 06-11-2018 aprobó el proyecto de trabajo de grado titulado **DESARROLLO DE APLICACIÓN INTERACTIVA CON INTELIGENCIA ARTIFICIAL PARA LA VALIDACIÓN DE FORMULARIOS WEB** presentado por usted(es) como requisito para optar al título de Ingeniero en Computación.

Se ratifica la designación de la Ing. Belkys Araujo, C.I: 6.906.234 y la Ing. Alicia Yáñez, C.I.: 4.598.880 como Tutores Académicos que lo asesorarán en el desarrollo de este proyecto.

Atentamente,



**Prof. Zulay Salcedo**  
Decana de la Facultad de Ingeniería

e. c. Coordinación de Pasantías y Trabajo de Grado (1).

ZS/fr



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE COMPUTACIÓN

ACEPTACIÓN DEL TUTOR

Quién suscribe, Ing. Belkys Araujo, portadora de la cédula de identidad N° V-6.906.234 hace constar que ha leído el Proyecto del Trabajo de Grado, presentado por el ciudadano José Miranda, portador de la cédula de identidad N° V-20.383.873 y la ciudadana Ana Guzmán, portador de la cédula de identidad N° V-21.216.201, titulado **DESARROLLO DE APLICACIÓN INTERACTIVA CON INTELIGENCIA ARTIFICIAL PARA LA VALIDACIÓN DE FORMULARIOS WEB**, presentado como requisito parcial para optar al título de Ingeniero de computación, y acepta la tutoría del mencionado Proyecto durante su etapa de desarrollo hasta su elaboración y evaluación; según las condiciones de la Coordinación de Pasantías y Trabajo de Grado de la Facultad de Ingeniería de la Universidad José Antonio Páez y sus correspondientes Reglamentos.

En San Diego, a los 20 días del mes de mayo del año dos mil dieciocho.

Firma  
Ing. Belkys Araujo  
V- 6.906.234



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE COMPUTACIÓN

San Diego, Julio de 2018

**ACTA DE REVISIÓN METODOLÓGICA DEL TRABAJO DE GRADO**

Quienes suscriben esta Acta, dejan constancia que el Proyecto de Trabajo de Grado: **DESARROLLO DE APLICACIÓN INTERACTIVA CON INTELIGENCIA ARTIFICIAL PARA LA VALIDACIÓN DE FORMULARIOS WEB**, Ha sido revisado y, cumplido con los requisitos exigidos para su aprobación, recomiendan su tramitación ante el organismo académico correspondiente.

Ing. Belkys Araujo  
Tutor Académico

Firma

30/07/18

Fecha

Ing. Alicia de Pizzella  
Tutor Metodológico

Firma

30-7-18

Fecha

## **AGRADECIMIENTOS**

En primer lugar, le agradeceré a mi madre y a mi abuela por haber estado ahí y haberme brindado el apoyo que siempre me han brindado por creer en mí y por los valores que me han inculcado.

Agradezco a José Miranda por haber sido mi compañero de tesis y por brindarme ese apoyo y compartido sus conocimientos conmigo en la realización de este trabajo de grado.

Doy gracias Andrés Lacruz por haberme apoyado y por darme la confianza para continuar adelante con mis proyectos.

A mis profesores que durante el tiempo que he estado en la universidad aprendí mucho de ellos y me enseñaron las herramientas para desenvolverme en el mundo laboral con tan importante carrera como lo es la Ing. De computación.

Y a todos mis compañeros que en algún momento de la carrera me brindaron su apoyo y compartieron conmigo durante el trayecto recorrido.

Ana Gabriela Guzmán Padilla.

## **DEDICATORIA**

Dedico en primer lugar a mi madre María Isabel Padilla y a mi abuela María del Rosario Padilla ya que sin ellas este logro no hubiese sido posible mis triunfos son parte gracias a todo el apoyo que me dieron durante todo este tiempo.

Ana Gabriela Guzmán Padilla.

# ÍNDICE GENERAL

## CONTENIDO

Pág.

<b>RESUMEN.....</b>	<b>x</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>

### CAPÍTULO

#### I EL PROBLEMA

1.1 Planteamiento del problema.....	3
1.2 Formulación del problema.....	5
1.3 Objetivos de la investigación.....	5
1.3.1 Objetivo General.....	5
1.3.2 Objetivos Específicos.....	6
1.4 Justificación de la investigación.....	6
1.5 Alcance.....	6

#### II MARCO TEÓRICO

2.1 Antecedentes.....	8
2.2 Bases teóricas.....	11
2.2.1 Sistema automatizado.....	11
2.2.2 Redes de neuronas artificiales.....	11
2.2.3 Algoritmos genéticos.....	11
2.2.4 Programación genética.....	12
2.2.5 C#.....	12
2.2.6 Selenium.....	13
2.2.7 Seleniumweb Driver.....	13
2.2.8 Metodología XP.....	14
2.2.9 Formularios Web.....	14
2.2.10 Pruebas.....	15

2.2.11 Pruebas unitarias.....	15
2.2.12 Pruebas de integración .....	15
2.2.13 Pruebas del sistema.....	16
2.2.14 Pruebas de regresión.....	16
2.2.15 Pruebas de humo.....	17
2.2.16 Pruebas de validación.....	17
2.2.17 Pruebas de recuperación.....	17
2.2.18 Pruebas de rendimiento.....	17
2.2.19 Pruebas de seguridad.....	18
2.2.20 Pruebas de interfaces gráficas de usuario.....	18
2.2.21 Pruebas automatizadas .....	18
2.3 Bases legales.....	19
2.4 Definición de términos.....	20

### **III MARCO METODOLÓGICO**

3.1 Tipo de investigación.....	22
3.2 Diseño de la investigación.....	22
3.3 Nivel de la investigación.....	22
3.4 Población y muestra.....	23
3.5 Técnicas e instrumentación de recolección de datos.....	23
3.6 Procesamiento y análisis de información.....	24
3.7 Fases metodológicas.....	24

### **IV RESULTADOS**

4.1 Resultados .....	27
4.1 Fase I: Planificación del proyecto.....	27
4.2 Fase II: Diseño.....	30
4.3 Fase III: Codificación.....	38
4.4 Fase IV: Pruebas.....	42

**CONCLUSIÓN..... 48**

**RECOMENDACIONES..... 50**

**BIBLIOGRAFÍAS..... 51**



**REPÚBLICA BOLIVARIANA DE VENEZUELA**  
**UNIVERSIDAD JOSÉ ANTONIO PÁEZ**  
**CARRERA INGENIERÍA**  
**FACULTAD DE INGENIERÍA EN COMPUTACIÓN**

**DESARROLLO DE APLICACIÓN INTERACTIVA CON INTELIGENCIA  
ARTIFICIAL PARA LA VALIDACIÓN DE FORMULARIOS WEB**

**Autores:** Ana Guzmán  
José Miranda  
**Tutor:** Ing. Belkis Araujo  
**Fecha:** Julio, 2018

**RESUMEN**

Las pruebas son el único instrumento que permite conocer el estado de calidad en el que se encuentra un software, por lo cual son un factor importante para determinar si cumple o no los requisitos funcionales del sistema. En la actualidad, muchas empresas tecnológicas llevan a cabo el desarrollo y mantenimiento de portales web para grandes sistemas corporativos. En función del estado de los proyectos, el producto suele pasar por distintos entornos bien diferenciados: local (L), integración (I), pre-producción (PP) y producción (P). El problema viene dado al pasar el producto de un entorno a otro, es decir, de L - I, de I - PP y de PP - P. Esto es así porque hay que probar que tanto las nuevas funcionalidades desarrolladas como todas las anteriores sigan funcionando como se espera y se deben generar los informes correspondientes. A día de hoy, aún hay proyectos en los que se sigue haciendo a mano. Esto es una tarea repetitiva (se ha de comprobar lo mismo al pasar de un entorno a otro) y presumible de contener errores que posteriormente deberán corregirse. El propósito de este trabajo de grado consiste en crear una herramienta de testing de formularios web para obtener resultados y generar informes automáticamente, reduciendo así el tiempo invertido en la realización de las pruebas y el trabajo repetitivo, aumentando la fiabilidad de las pruebas y posibilitando su reutilización.

**Descriptor:** Aplicación de escritorio, inteligencia artificial, formularios web, programación.

## INTRODUCCIÓN

La presente investigación está orientada al ámbito de formularios web y busca desarrollar una solución que podría ser usada en la automatización de pruebas de sistemas web.

“Los sistemas software son cada vez más grandes y más complejos y los clientes demandan software más confiable que sea desarrollado más rápidamente” [Som’05].

Esta cita induce a pensar que las pruebas de software tienen cada día más importancia porque garantizan la satisfacción de los requisitos. Sin embargo, cuando éstas se automatizan, además de garantizar la satisfacción de los requisitos del software, se garantiza su calidad y se ahorra tiempo y recursos en su desarrollo al localizar el mayor número de deficiencias en el menor tiempo. La comprobación del software consiste en la verificación dinámica del comportamiento de un 1 programa en un conjunto finito (número limitado) de casos de prueba.

Los casos de prueba permiten tanto verificar el comportamiento esperado de una aplicación, como detectar errores. Interesa que con el menor número de casos se encuentren el mayor número de errores. Para ello se necesitan unos buenos casos de prueba, que se caracterizan por:

En el capítulo 1 se presenta la problemática de donde se extrajo el problema que tuvo como resultado la propuesta del presente proyecto de fin de carrera, el objetivo general, objetivos específicos y finalmente se menciona el alcance.

En el capítulo 2 se encuentra sustentado por los antecedentes de la investigación y los fundamentos teóricos que sirven de soporte a la variable de estudio. Asimismo, se muestra la definición de los términos básicos.

En el capítulo 3 constituye el marco metodológico, en donde se establece la modalidad de la investigación, el tipo, la operacionalización de las variables, la población y objeto de estudio, las técnicas utilizadas para la recolección de información, la validez, así como el procedimiento para el desarrollo de la investigación.

En el capítulo 4 se detallan las actividades de cada fase, se mostraron los resultados de las herramientas aplicadas, se anexaron gráficos y diagramas que permiten la fácil visualización y entendimiento de lo planteado.

Cuando un software no es probado puede tener una gran cantidad de errores, al punto que pueda que haya pérdida de información, incoherencia en la misma, cierres inesperados del programa, entre muchas otras consecuencias.

La prueba exhaustiva del producto antes de ser entregado al cliente implica probar el comportamiento del mismo para todas las combinaciones válidas e inválidas de entradas, bajo cada estado posible del sistema. Esto, incluso para un programa pequeño puede llevar cientos de años y es económicamente inviable. Debido a esto, las empresas que desarrollan software intentan entregar sus productos con la menor cantidad de defectos posible. Para esto, definen cuándo entregar el producto a los usuarios en función del costo y el beneficio de realizar las pruebas.

El desafío está en encontrar los defectos que tendrían mayor impacto negativo para el negocio del cliente antes de que el producto se haya entregado. Para lograr este objetivo, existen técnicas y estrategias para definir y ejecutar las pruebas.

# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1 Planteamiento del Problema**

En los últimos años el crecimiento de las empresas dedicadas al desarrollo de software ha sido a gran escala, el mercado se ha diversificado y han surgido nuevas oportunidades, la alta competencia ha provocado la constante investigación y el desarrollo de nuevas técnicas enfocadas a optimizar procesos, que disminuyan costos e incrementen el rendimiento de las aplicaciones a desarrollar. Internet maneja grandes cantidades de información ingresada de manera directa o indirectamente por sus usuarios, la forma más común de ingresar información de manera directa es mediante formularios, esta es registrada con la aprobación del usuario dentro del sistema en el que se esté interactuando.

En las páginas web durante su desarrollo, son utilizados una variedad de formularios con múltiples fines, entre los más utilizados, el formulario de registro, este es susceptible a muchos errores de escritura que son evaluados por tester o los mismos desarrolladores de forma manual, lo que implica un trabajo tedioso y repetitivo que ocupa mucho tiempo, aun así en la mayoría de los casos no son evaluados todas las capacidades de estos, por lo tanto pueden generar errores en el almacenamiento de la información, lo que significa conflictos en el sistema y posibles errores graves que atenten contra la integridad de la web.

La calidad de un producto de software es percibida principalmente cuando supera las expectativas por los usuarios. La empresa dedicada al testing tiene como objetivo detectar todos los fallos existentes en el software esto significa un costo adicional implementado para el desarrollo del proyecto, estos costos son proporcionales a la cantidad de elementos a evaluar.

Toda empresa que desarrolla software a menudo entrega el producto con anomalías de diferente gravedad, unas más visibles que otras. Incluso cuestionan cómo es posible que puedan pasar desapercibidas tales fallas a los probadores de software cuando incluso tienen un entrenamiento en pruebas. Los desarrolladores conocen esa frustración cuando reciben reportes de errores por parte de los usuarios y es aquí cuando

cuestionan ¿Cómo puede ser posible cuándo se invirtieron incontables horas en validaciones, cuidado y seguimiento de variables y sentencias de código?

Las respuestas a esta interrogante pueden ser las siguientes:

El usuario ejecuta código no probado. Por falta de tiempo no es raro que los desarrolladores liberen código sin probar, en el que los usuarios pueden encontrar anomalías.

El orden en el que se ejecutan las declaraciones en el ambiente de uso del que se utilizó durante la prueba. Este orden puede determinar si el software funciona o no.

El usuario aplica una combinación de valores de entrada no probados. Las posibles combinaciones de valores de entrada, que miles de usuarios pueden hacer a través de una interfaz de software, simplemente son demasiado numerosas para que los testers las apliquen todas y, como se debe tomar decisiones acerca de qué valores de entrada probar, a veces se toman las equivocadas.

Los testers de software deben considerar lo siguiente al planificar y ejecutar las pruebas: el software y función del cálculo, las entradas y cómo se pueden combinar y el entorno en el que el software eventualmente funcionará.

Entonces para acercarse a la fase de pruebas se proponen la aplicación de cuatro fases:

Modelar el entorno del software.

Seleccionar escenarios de prueba.

Ejecutar y evaluar los escenarios.

Medir el progreso de la prueba.

Por lo tanto, la implementación de un proceso de pruebas permite evaluar la calidad del producto final, la información está controlada por lo que permite tener un control de calidad más efectivo y el proceso de pruebas permite detectar y dar seguimiento a los errores y así entregar las soluciones a los clientes.

Los procesos de pruebas de software son una forma de elevar la calidad de un producto por lo cual es importante aplicar estrategias que permitan este objetivo.

Finalmente, ésta investigación consiste en la indagación profunda de diferentes tipos de pruebas de software, las metodologías existentes y la selección de las

herramientas más adecuadas para cumplir con la resolución de la naturaleza del problema planteado.

Para lograr el objetivo se realizaron pruebas para la validación de campos de texto así también como de URLs para validar que los usuarios no puedan hacer uso del sistema sin autenticarse.

La solución planteada para la realización de las pruebas a cualquier formulario web se puede realizar por medio de dispositivos como tablet, laptop. Este probador de software ejecutará dos principales pruebas de software:

**Validación de campos:** Esta prueba consistirá en validar cada uno de los campos del sistema, los campos numéricos solo acepten este tipo de caracteres, los alfanuméricos tanto letras como números así mismo como la magnitud máxima de ellos.

**Validación de URLs:** Esta prueba validará cada una de las URLs para evitar que los usuarios puedan hacer uso del sistema sin antes estar registrados y haber iniciado sesión.

Para poder realizar estas pruebas fue necesaria la creación de la interfaz del software ya que por medio de ella se accederá a las funciones del sistema. Una vez realizada la prueba se genera un reporte de cada una de ellas, con las observaciones sobre los posibles errores encontrados en el sistema y soluciones con su respectiva captura para una mejor visualización.

Esta herramienta permite encontrar los errores que pasaron desapercibidos durante el desarrollo del software y que pudieran comprometer la integridad de la información del mismo.

## **1.2 Formulación del problema**

Teniendo en cuenta que los formularios son propensos a gran número de errores y que determinar estos errores de manera manual es un trabajo tedioso y que ocupa mucho tiempo que podría ser invertido en otro fin más productivo ¿Cómo se puede evaluar el desempeño de un formulario web, mediante una prueba automatizada?

## **1.3 Objetivos de la investigación**

### **1.3.1 Objetivo General**

Desarrollar una aplicación para pruebas automatizadas en la verificación de pruebas a formularios web. Utilizando Scripts genéricos en Selenium Web Driver y el

lenguaje de programación C#. Para optimizar procesos y garantizar la fiabilidad del sistema a evaluar.

### **1.3.2 Objetivos Específicos**

- 1) Determinar los requerimientos funcionales y no funcionales mediante técnicas de recolección de datos.
- 2) Diseñar la aplicación de pruebas automáticas mediante la metodología XP ( )
- 3) Desarrollar Scripts genéricos en Selenium Web Driver para la creación de pruebas automáticas de testeo web.
- 4) Aplicar dichas pruebas en diferentes formularios para la evaluación de sus resultados.

### **1.4 Justificación de la investigación**

Esta investigación es realizada con la finalidad de crear una herramienta para solventar y agilizar el desarrollo de software mediante pruebas automatizadas generando automáticamente un informe con los respectivos errores en formularios web que requieran validación y verificación. Debido a que hacer las pruebas de manera manual implica tener que hacer un trabajo repetitivo cuando se quieren llevar a cabo las mismas pruebas en entornos y navegadores distintos. La automatización es una inversión, y como tal, se debe estudiar su viabilidad. En definitiva, se optimiza el proceso de pruebas disminuyendo su tiempo de ejecución y aumentando su fiabilidad.

Esto implica una reducción de tiempo en la revisión del sistema a evaluar además de menos personal para la elaboración de ese fin, lo que también significa una reducción en el costo utilizado. Serán utilizados algoritmos que implementen inteligencia artificial para las detecciones y evaluación de los formularios, lo que significa un aporte universitario significativo, debido a la expansión del conocimiento y el desarrollo de nuevas tecnologías para la universidad José Antonio Páez.

### **1.5 Alcance**

Dicho sistema tiene la capacidad de evaluar y detectar formularios exponiendo los resultados de la evaluación de manera sencilla en un informe realizado de manera automática. Dicho informe contará con los resultados de la ejecución de la prueba así como un resumen de casos correctos e incorrectos, las observaciones referentes al

rendimiento de los elementos evaluados en todos los campos del formulario de registro, solo con tener el link donde se encuentra el formulario éste realizará todo el proceso de una manera rápida e intuitiva, esto gracias a la IA que poseerá para detectar y evaluar todos los campos, por lo tanto el usuario no tendrá que aprender el funcionamiento del sistema por su sencillez.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

Para sustentar este proyecto es necesario consultar diferentes trabajos de investigación relacionados con el mismo para servir como guías de complemento al proceso de alcanzar los objetivos.

#### **2.1 Antecedentes**

Con el propósito de ampliar y profundizar la temática que se aborda en el presente trabajo de investigación como es la inteligencia artificial en las pruebas automatizadas en formularios web, además del funcionamiento de aplicaciones interactivas, se incluyen como apoyo, una serie de estudios previos elaborados por diferentes investigadores, cuyas conclusiones contribuirán al logro de los objetivos planteados.

quienes realizaron su trabajo de grado titulado **Evaluación de accesibilidad de sitios web de las Universidades Públicas Peruanas**, realizado en la Unidad Católica Del Perú. El ámbito educativo no es ajeno al creciente uso de las tecnologías de información y comunicación (TICs), por eso se encuentra que las instituciones de Educación Superior Universitaria publican en sus sitios web la información y los servicios más relevantes que brindan. Por otro lado, el acceso a los sistemas de información por parte de las personas con discapacidad constituye un derecho y para poder alcanzarlo es importante desarrollar características que permitan su accesibilidad.

Conociendo esta problemática se realizó un diagnóstico de un conjunto de universidades públicas peruanas, a fin de conocer los problemas de accesibilidad web que presentan a nivel global y proveer recomendaciones adecuadas. Con este fin se seleccionaron un conjunto de páginas web de las Universidades Públicas Peruanas, considerando las recomendaciones de la Iniciativa de Accesibilidad Web (WAI) del World Wide Web Consortium (W3C) y su Metodología de Evaluación de Conformidad de la Accesibilidad Web en su versión 2.0. En función a los estudios de accesibilidad, se definió la forma de llevar a cabo el método de evaluación el cual se aplicó al dominio de estudio, obteniéndose bajos niveles de accesibilidad en los contenidos de las páginas

Principal y de Oferta Académica de las Universidades Públicas analizadas, así como una lista de problemas comunes, para así plantear recomendaciones generales.

El siguiente trabajo de grado ayudo en la comprensión, del manejo de páginas web y su accesibilidad con y sin las mejoras que ellos aconsejan, este conocimiento nos traerá como beneficio el mejor acceso y detección de formularios en dichas páginas web, punto indispensable en el trabajo de grado presentado.

Quintanilla Paredes, José Guillermo; Rodríguez Mantilla, José Luis (2017) **Sistema experto basado en redes neuronales para mejorar la determinación del impuesto predial en la Municipalidad Distrital de La Esperanza** realizado en la Universidad Nacional de Trujillo. La presente tesis trata acerca del problema de la ineficiente determinación del impuesto predial en la municipalidad distrital de La Esperanza y de la solución propuesta para este problema que consiste en un sistema experto. Este sistema experto será desarrollado utilizando el paradigma de las redes neuronales artificiales.

Este sistema experto será de gran ayuda para determinar los impuestos prediales en menos tiempo y de forma más confiable que utilizando el procedimiento tradicional. El capítulo 1 consiste en la introducción, la cual incluye la formulación del problema, los objetivos de la tesis, y la metodología de la investigación. El capítulo 2 comprende el marco teórico, que incluye los conceptos y definiciones más relevantes para esta tesis. El capítulo 3 consiste en la propuesta o tema central de la tesis. El capítulo 4 comprende los resultados de esta tesis. El capítulo 5 consiste en las consideraciones finales, que incluyen las conclusiones y trabajos futuros.

Este trabajo de grado fue seleccionado por el uso de las redes neuronales artificiales, debido a que su comprensión nos facilita el desarrollo e implementación de esa tecnología, para mejorar el funcionamiento de la aplicación que próximamente será presentada.

Castañeda Rojas, Edson Bryan quienes realizaron su trabajo de grado titulado **Propuesta de patrón de diseño de software orientado a prevenir la extracción automatizada de contenido web** realizado Universidad Catolica Del Peru. 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.

Uno de los trabajos más relevantes con respecto al trabajo de grado presentado, debido a que dificulta la automatización de las pruebas automatizadas generando aleatoriedad en las páginas web, esto representa un escalón más de dificultad lo que mejora el funcionamiento de la futura aplicación, debido a que está al realizar las detecciones a través de inteligencia artificial no importara el orden que se desenvuelve el formulario, automáticamente detectara los inputs a evaluar.

Alvarez Díaz, Jhon Manuel; Torres Aguilar, Miguel Angel (2014) quienes realizaron su trabajo de grado titulado **Desarrollo de un portal web que permita mejorar la gestión y el acceso a la información para la Escuela de Informática de la Universidad Nacional de Trujillo** realizado en la Universidad Nacional de Trujillo. Debido al déficit con el que cuenta la actual página web de Informática de la Universidad Nacional de Trujillo, en cuestión de informar y brindar servicios académicos a los alumnos de Informática de la Universidad Nacional de Trujillo; y aprovechando las tecnologías de información existentes, se desarrolló el portal web para la Escuela de Informática de la Universidad Nacional de Trujillo., el cual permitió mejorar la gestión y el acceso a la información.

A lo largo del desarrollo de la presente tesis, se ha aplicado la metodología RUP; así como todos los artefactos, diagramas UML y modelamientos necesarios para el

posterior desarrollo del Portal web, fue desarrollado el portal web utilizando el CMS Joomla. Además, se realizó el diseño de pruebas, para comprobar la funcionalidad del portal web, las cuales fueron satisfactorias. Para la demostración de la tesis, se aplicaron dos encuestas a los usuarios finales del portal web, la primera encuesta fue aplicada antes de la implementación del portal web propuesto, cuando los usuarios contaban con la antigua página web; la segunda encuesta fue aplicada una vez finalizado el portal web propuesto; todo esto con la finalidad de demostrar la mejora de la gestión y el acceso a la información del portal web.

Los resultados finales obtenidos y posteriormente discutidos, concluyen que, el portal web desarrollado logró mejorar la gestión y el acceso a la información para la escuela de Informática de la Universidad Nacional de Trujillo.

Esta investigación ayudó en la comprensión de los procesos logísticos de las web y el desarrollo de las mismas, nos aporta en el conocimiento general del funcionamiento de las páginas web además de aportar en el proceso de pruebas para el desempeño y la funcionalidad.

Roberto Luis Felipe y Vargas Guevara (2012) quienes realizaron su trabajo de grado titulado **Desarrollo de Aplicación Interactiva Para Enseñar Vocabulario En Inglés a Niños Desde Edad Pre Escolar** realizado en la Universidad De Chile Facultad de Ciencias Físicas Y Matemáticas Departamento De Ciencias De La Computación En esta tesis se presenta el diseño, desarrollo y resultados de validación de una aplicación iPad que permite a niños de los primeros años de enseñanza básica aprender vocabulario en inglés, a través de distintas alternativas de interacción. La enseñanza de idiomas extranjeros es una parte fundamental de la educación. En particular, el aprendizaje de vocabulario es crítico para el progreso, ya que es la base de la puesta en práctica de las demás habilidades. Si se lleva a cabo en los primeros años de enseñanza (entre los cinco y siete años), una de las barreras que deben superar los niños es la motivación.

El estudio de idiomas puede ser estresante y producir rechazo en algunos estudiantes. La tecnología surge como una forma de reducir esta aversión, debido, entre otros factores, a la posibilidad de presentar atractivos recursos multimedia, y formas de interacciones novedosas que invitan a los niños a acceder al contenido. La hipótesis es que, a través de la tecnología, es posible tanto facilitar como motivar el aprendizaje de vocabulario en inglés para los niños de primeros años de enseñanza básica. La solución

propuesta tiene dos objetivos globales. Por un lado, ayudar al aprendizaje de vocabulario, con una interfaz centrada en la asociación entre expresiones y significados, a través de gestos sobre la pantalla. Por otro lado, mantener el interés en el estudio y motivar a los alumnos que pueden sentirse excluidos, ofreciendo distintas formas de interacción.

Este trabajo de grado aportó contenido teórico del correcto funcionamiento en la interacción que tiene el usuario con la aplicación, conocimiento que será utilizado en la creación de la aplicación de escritorio que será presentada junto a este proyecto.

Almudena Jiménez Sierra, Fco. Javier Muñoz Rodríguez, Gonzalo Torres Porta (2006) quienes realizaron su trabajo de grado titulado **Aplicación de métodos de Inteligencia Artificial para la toma de decisiones en simulación de móviles**. Realizado en Universidad Complutense de Madrid, Facultad de Informática. Este proyecto, realizado en la Universidad Complutense de Madrid para la asignatura de Sistemas Informáticos a petición de la empresa Eads-Casa, consiste en la simulación mediante técnicas de inteligencia artificial del comportamiento de un UAV (Vehículo Aéreo no Tripulado), capaz de tomar decisiones sobre su trayectoria y encontrar un camino óptimo entre dos puntos frente a un conjunto de obstáculos y teniendo en cuenta diversos factores limitantes. El algoritmo principal está implementado en java con una interfaz gráfica en 3-D en Matlab.

Este trabajo de grado aportó aún más conocimiento teórico para el entendimiento y desarrollo de una inteligencia artificial además de la interacción del usuario con la aplicación.

## **2.2 Bases teóricas**

Para la realización del presente trabajo de investigación se toman en cuenta diferentes estudios, definiciones relacionadas con el tema, el cual corresponde a la base de sustentación, basándose en la teoría existente de acuerdo a las necesidades y elementos necesarios para llevar a cabo el sistema automatizado.

### **2.2.1 Sistema automatizado**

Es un conjunto ordenado de normas y procedimientos que regulan el funcionamiento de un grupo o colectividad. Son las acciones organizadas y relacionadas que interactúan entre sí para que se lleve a cabo la operación deseada. El propósito principal de un sistema automatizado es ayudar a acelerar un proceso. Las tareas que

consumen tiempo o son inconvenientes, a menudo se incorporan a los sistemas. En la industria informática, hay muchas tareas que no requieren atención humana constante.

### **2.2.2 Redes de neuronas artificiales**

Las “Redes de Neuronas Artificiales” (RNA) también llamadas “Sistemas Conexionistas” son un sistema de procesamiento de información inspirado en el funcionamiento del sistema nervioso biológico (paradigma de sistema inteligente). Representan un intento de emular el comportamiento del cerebro y aumentar el conocimiento acerca de él. Funcionan como un sistema masivamente paralelo que se ha mostrado eficiente para la resolución inteligente de diferentes tipos de problemas.

La RNA consta de elementos básicos de procesos, normalmente organizados en capas (basados en las neuronas biológicas) y con gran cantidad de conexiones entre ellos (basadas en las sinapsis de las neuronas).

### **2.2.3 Algoritmos genéticos**

Los Algoritmos Genéticos (AG) son algoritmos de búsqueda inspirados en procesos de selección natural, basados en la teoría de la evolución de Darwin. Establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural. Se aplican principalmente en problemas de optimización y se comportan de un modo muy eficaz en problemas de superficie compleja, con múltiples mínimos locales y grandes espacios de búsqueda.

Los AG trabajan sobre poblaciones de individuos que representan soluciones al problema. Para ello utilizan las operaciones de selección, cruce y mutación, propias de la genética. De esta manera, mediante individuos se codifican diferentes soluciones al problema, inicialmente aleatorias. Luego estos individuos se agrupan en una población y se van combinando entre sí de forma que a medida que pasan las generaciones se van ajustando más a la solución final.

### **2.2.4 Programación genética**

John Koza, en 1992, adaptó los AG para la creación de programas de computador, permitiendo que las técnicas de Computación Evolutiva se utilizaran para construir grandes poblaciones de programas que evolucionan y finalmente obtienen, de forma automática, un programa que constituye la solución a un problema dado. Él mismo bautizó a esta nueva técnica como “GeneticProgramming” o “Programación Genética”.

Así, la Programación Genética (en adelante PG) surge como una evolución de los AG tradicionales, manteniendo el mismo principio de selección natural. Es una técnica de búsqueda en la que, a partir de un conjunto inicial de programas (población), éstos se van combinando sucesivamente, formando distintas generaciones para dar lugar a programas. Al igual que los AG, la PG también forma parte de las denominadas técnicas de Computación Evolutiva.

### 2.2.5 C#

Es un lenguaje programación orientados a objetos diseñado por Microsoft y sus creadores fueron AndersHejlsberg y Scott Wiltamuth. Es un lenguaje que toma características de diferentes lenguajes de programación y los combina en uno. Según Troelsen (2012) define C# como, es un lenguaje de programación cuya sintaxis central se parece mucho a la sintaxis de Java. Sin embargo, para llamada C# un clon de Java es inexacto. En realidad, tanto C# como Java son miembros de la familia C de lenguajes de programación (por ejemplo, C, Objetivo C, C ++ ) y, por lo tanto, comparten una sintaxis similar.

En C# existen características importantes:

**Sencillez:** elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.

**Orientado a objetos:** como todo lenguaje de programación de propósito general actual. Soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo.

**Orientado a componentes:** C# incluye elementos propios de diseño que otros lenguajes deben simular mediante construcciones complejas.

**Eficiente:** el código de C# incluye restricciones que asegura el no permitir el uso de punteros. Sin embargo y a diferencia de java, es posible utilizar las saltarse las restricciones manipulando los objetos a través de punteros. Lo que resulta muy importante en procesos donde se necesite eficiencia y velocidad al procesar grandes cantidades de información.

**Compatibilidad:** C# tiene una sintaxis similar a C, C++ o JAVA que permite incluir directamente en código escrito en C# partes de código escrito en estos lenguajes.

### 2.2.6 Selenium

Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas (**Selenium IDE**). Incluye también un lenguaje específico de dominio para pruebas (**Selenese**) para escribir pruebas en un amplio número de lenguajes de programación populares incluyendo Java, C#, Ruby, Groovy, Perl, Php y Python. Las pruebas pueden ejecutarse entonces usando la mayoría de los navegadores web modernos en diferentes sistemas operativos como Windows, Linux y OSX.

### **2.2.7 SeleniumWebDriver**

SeleniumWebDriver es el sucesor de Selenium RC. SeleniumWebDriver acepta comandos (enviados en Selenese o vía el API de cliente) y los envía a un navegador. Esto se implementa a través de un controlador del navegador específico para cada navegador que envía los comandos y trae los resultados de regreso. La mayoría de los controladores del navegador lanzan y acceden a la aplicación de navegador (como Mozilla Firefox o Internet Explorer), pero también hay un controlador para HtmlUnit que simula un navegador.

### **2.2.8 Metodología XP**

El autor de la XP (eXtremeProgramming) es Beck (2005) que define "XP es un Metodología ligera para los equipos pequeños o medianos de desarrollo de software frente a requisitos vagos o que cambian rápidamente" (p. 3), que con su larga experiencia como programador eligió las mejores características de las metodologías y profundizó en las relaciones de éstas y como se reforzaban unas a otras. Por tanto, la XP no se basa en principios nuevos, sino que todas, o casi todas, sus características ya se conocen dentro de la ingeniería del software, las cuales se complementan para minimizar los tópicos problemas que pueden surgir en todo desarrollo de proyectos software.

Para esta metodología de desarrollo está dividido en 4 fases:

1. Planificación: se obtiene la información referente a lo que se va a desarrollar.
2. Diseño: se levantan los requerimientos funcionales y no funcionales se diseña la arquitectura del software.
3. Desarrollo: codificación de la aplicación basada en las necesidades de cliente.
4. Pruebas: se le realizan las pruebas de calidad al sistema desarrollado.

### **2.2.9 Formularios Web**

Un formulario web dentro de una página web permite al usuario introducir datos los cuales son enviados a un servidor para ser procesados. Los formularios web se parecen a los formularios de papel porque los internautas llenan dichos formularios usando casillas de selección, botones de opción, o campos de texto. Por ejemplo, los formularios web pueden ser usados para introducir datos de envío con el objetivo de solicitar un producto o bien ser utilizada para solicitar datos (p. ej., al buscar en un motor de búsqueda).

Además de servir como plantillas para nueva información, los formularios web también pueden ser usados para consultar y mostrar información existente en forma similar a los formularios de combinación de correspondencia, incorporando las mismas ventajas. La separación entre la estructura y los datos subyacentes de un mensaje permite a ambos variar independientemente. El uso de formularios webs para este propósito evita los problemas asociados con la creación explícita de páginas web separadas para cada registro en una base de datos.

Los formularios web están definidos en lenguajes de programación como HTML, Perl, Java o .NET. Las implementaciones de estos lenguajes usualmente invocan automáticamente los idiomas de la interfaz de usuario y otras de sus características, tales como el diseño estructural, y tema, minimizando el tiempo, el costo y el tiempo de programación.

### **2.2.10 Pruebas**

Las pruebas tienen objetivos puntuales. Según los objetivos se determinan qué tipo de pruebas aplican para cada escenario en particular, además de contar con una estrategia de prueba. Resulta difícil probar todos los escenarios ya que las entradas y las circunstancias en las que se puede encontrar el componente probado son muchas y en una situación real, se cuenta con los recursos limitados por lo que es necesario priorizar las pruebas necesarias e ir escalando con las prioridades hasta consumir los recursos disponibles.

En resumidas cuentas, es conveniente priorizar los aspectos a revisar en un proceso de pruebas, priorizando lo esencial [Graham, 2008].

#### **2.2.11 Prueba unitaria**

La prueba unitaria es el proceso de probar los componentes individuales (subprogramas o procedimientos) de un programa. El propósito es descubrir discrepancias entre la especificación de la interfaz de los módulos y su comportamiento real. Una unidad es la pieza de software más pequeña que se puede probar, es en general el trabajo de un programador y consiste de cien o menos líneas de código. La prueba unitaria es conocida también como prueba de módulo o prueba de componente.

#### **2.2.12 Prueba de integración**

Es el proceso en el cual los componentes son agregados para crear componentes más grandes. Es el testing realizado para mostrar que aunque los componentes hayan pasado satisfactoriamente las pruebas unitarias, la combinación de componentes es incorrecta o insatisfactoria.

#### **2.2.13 Prueba del sistema**

Refiere al comportamiento del sistema entero. La mayoría de las faltas funcionales deben haber sido identificadas ya durante las pruebas de unidad e integración. La prueba del sistema generalmente se considera apropiada para probar los requerimientos no funcionales del sistema, tales como seguridad, desempeño, exactitud, y confiabilidad. Las interfaces externas, los dispositivos del hardware, o el ambiente de funcionamiento también se evalúan a este nivel.

#### **2.2.14 Pruebas de regresión**

Su objetivo es verificar que no ocurrió una regresión en la calidad del producto luego de un cambio, asegurándose que los cambios no introducen un comportamiento no deseado o errores adicionales. Implican la re-ejecución de alguna o todas las pruebas realizadas anteriormente.

#### **2.2.15 Prueba de humo**

Las pruebas de humos son un conjunto de pruebas aplicadas a cada nueva versión, su objetivo es validar que las funcionalidades básicas de la versión se cumplen

según lo especificado. Estas pruebas buscan grandes inestabilidades o elementos clave faltantes o defectuosos, que hacen imposible realizar la prueba como fue planificado para la versión. Si la versión no pasa las pruebas de humo, no se comienza la ejecución de las pruebas planificadas de la versión.

#### **2.2.16 Pruebas de validación**

Las pruebas de validación comienzan en la culminación de las pruebas de integración, cuando se ejercitaron componentes individuales, el software está completamente ensamblado como un paquete y los errores de interfaz se descubrieron y corrigieron. Las pruebas se enfocan en las acciones visibles para el usuario y las salidas del sistema reconocibles por el usuario.

#### **2.2.17 Pruebas de recuperación**

Muchos sistemas basados en computadoras deben recuperarse de fallas y reanudar el procesamiento con poco o ningún tiempo de inactividad. En algunos casos, un sistema debe ser tolerante a las fallas, es decir, las fallas del procesamiento no deben corregirse dentro de un periodo de tiempo específico u ocurrirán severos daños económicos.

#### **2.2.18 Pruebas de rendimiento**

La prueba de rendimiento se diseña para poner a prueba el rendimiento del software en tiempo de corrida, dentro del contexto de un sistema integrado. La prueba del rendimiento ocurre a lo largo de todos los pasos del proceso de prueba. Incluye en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan las pruebas. Sin embargo, no es sino hasta que todos los elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento de un sistema.

#### **2.2.19 Pruebas de seguridad**

Cualquier sistema basado en computadora que gestione información sensible o cause acciones que puedan dañar (o beneficiar) de manera inadecuada a individuos es un blanco de penetración inadecuada o ilegal. La penetración abarca un amplio rango de actividades: hackers que intentan penetrar en los sistemas por deporte, empleados

resentidos que intentan penetrar por venganza, individuos deshonestos que intentan penetrar para obtener ganancia personal ilícita.

#### **2.2.20 Pruebas de interfaces gráficas de usuario**

Las interfaces gráficas para usuario (GUI, por sus siglas en inglés) presentan interesantes retos de prueba. Puesto que los componentes reutilizables ahora son parte común en los entornos de desarrollo GUI, la creación de la interfaz para el usuario se ha vuelto menos consumidora de tiempo y más precisa. Pero, al mismo tiempo, la complejidad de las GUI ha crecido, lo que conduce a más dificultad en el diseño y ejecución de los casos de prueba. Debido a que muchas GUI modernas tienen la misma apariencia y ambiente, puede derivarse una serie de pruebas estándar. Es posible usar las gráficas de modelado de estado finito para derivar una serie de pruebas que abordan objetos de datos y programa específicos que sean relevantes para la GUI.

#### **2.2.21 Pruebas automatizadas**

Hay herramientas que apoyan diversos aspectos de la prueba. A continuación se presenta una clasificación posible para las herramientas [ISQ05]:

Administración de las pruebas y el proceso de pruebas: Herramientas para la administración de las pruebas, para el seguimiento de incidentes, para la gestión de la configuración y para la administración de requerimientos.

Pruebas estáticas: Herramientas para apoyar el proceso de revisión, herramientas para el análisis estático y herramientas de modelado.

Especificación de las pruebas: Herramientas para el diseño de las pruebas y para la preparación de datos de prueba.

Ejecución de las pruebas: herramientas de ejecución de casos de prueba, herramientas de pruebas unitarias, comparadores, herramientas de medición del cubrimiento, herramientas de seguridad.

Desempeño y la monitorización: Herramientas de análisis dinámico, herramientas de desempeño, de carga y de estrés, herramientas de monitorización.

### **2.3 Bases legales**

Las bases legales constituyen el conjunto de documentos de naturaleza legal que dan soporte a la investigación. Estas son descritas a continuación haciendo referencia a los artículos más resaltantes.

A. Constitución de la República Bolivariana de Venezuela (1999)

“**Artículo 110.** El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico social y político del país, así como para la seguridad y soberanía nacional. “

Se infiere que todas las iniciativas en función de innovar los sistemas de información serán reconocidas como un instrumento para el desarrollo de las instituciones nacionales y por ende para el desarrollo nacional.

B. Ley Orgánica de la Administración Pública (2014)

“**Artículo 152.** Los órganos y entes de la Administración Pública podrán incorporar tecnologías y emplear cualquier medio electrónico, informático, óptico o telemático para el cumplimiento de sus fines. Los documentos reproducidos por los citados medios gozarán de la misma validez y eficacia del documento original, siempre que se cumplan los requisitos exigidos por la ley y se garantice la autenticidad, integridad e inalterabilidad de la información.”

Se entiende que cualquier desarrollo privado o público siempre y cuando siempre que cumpla con las bases legales establecidas podrá ser utilizado por cualquier establecimiento y sus documentos tendrá la misma validez que el documento legal.

1. Ley Orgánica de Ciencia, Tecnología e Innovación (2005)

“**Artículo 3.** Forman parte del sistema nacional de ciencia tecnología e innovación las instituciones públicas o privadas que generen y desarrollen conocimientos científicos y tecnológicos, como procesos de innovación, y las personas que se dediquen a la planificación, administración, ejecución y aplicación de actividades que posibiliten la vinculación efectiva entre la ciencia, la tecnología y la sociedad.”

La ley ampara un desarrollo privado que ayude a automatizar los procesos de la administración.

## 2.4 Definición de términos

**Aplicación:** Es un tipo de programa informático que se presenta como herramienta para la comodidad o facilidad de realizar algún o varios tipos de trabajos.

**Framework:** Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

**Módulo:** Porción de programa de ordenador, cumple una tarea necesaria para atender una función del mismo.

**MVC:** Patrón de arquitectura de las ciencias de la computación, conocido como Modelo-Vista-Controlador que separa la interfaz de los datos y las rutas de comunicación.

**NET:** Es un entorno de ejecución a tiempo real que administra aplicaciones cuyo destino es .NET Framework. Incorpora CLR y una biblioteca de clases completa, permitiendo aprovechar el código sólido y confiable de todas las áreas principales del desarrollo de aplicaciones.

**Protocolo:** Es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen.

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

**SDK:** Conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear aplicaciones para un sistema en específico.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

Este capítulo a continuación, se enfocó en los aspectos relativos a la metodología que se utilizó para realizar el presente estudio, tomando en consideración el tipo de investigación, diseño, la población, la muestra y se describen las técnicas e instrumentos de recolección de los datos, los procedimientos que se emplearon para darle validez y confiabilidad a fin de procesar y analizar los resultados y de esta manera obtener una conclusión que permita dar respuestas a los objetivos planteados.

#### **3.1 Tipo de investigación**

La investigación a desarrollar se considera un Proyecto factible según las normas para la elaboración y presentación de trabajo de grado tesis doctorales de la universidad experimental pedagógica de libertador (U.P.E.L.) 2006, explica, El Proyecto Factible consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El Proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades.

#### **3.2 Diseño de la investigación**

Según el autor (Fidias G. Arias (2012)), define: La investigación de campo es aquella que consiste en la recolección de todos directamente de los sujetos investigados, o de la realidad donde ocurren los hechos (datos primarios), sin manipular o controlar variables alguna, es decir, el investigador obtiene la información pero no altera las condiciones existentes. De allí su carácter de investigación no experimental.

Claro está, en una investigación de campo también se emplea datos secundarios, sobre todo los provenientes de fuentes bibliográficas, a partir de los cuales se elabora el marco teórico. No obstante, son los datos primarios obtenidos a través del diseño de campo, lo esenciales para el logro de los objetivos y la solución del problema planteado.

#### **3.3 Nivel de la investigación**

De acuerdo con el tipo de investigación dentro del cual se ubica este proyecto,

el nivel al cual pertenece es descriptivo debido a que busca los orígenes de la problemática, sus efectos y los compara con otros hechos ocurridos para determinar los factores causantes de la situación y sobre estos hechos presentar una interpretación correcta.

Según Tamayo y Tamayo (2003), en su libro *Proceso de Investigación Científica*, la investigación descriptiva “comprende la descripción, registro, análisis e interpretación de la naturaleza actual, y la composición o proceso de los fenómenos. El enfoque se hace sobre conclusiones dominantes o sobre grupo de personas, grupo o cosas, se conduce o funciona en presente” (p.47).

### **3.4 Población y Muestra**

Según Morales, citado por Arias (1998), “la población o universo se refiere al conjunto por el cual serán válidas las conclusiones que se obtengan: a los elementos o unidades (personas, instituciones o cosas) involucradas en la investigación” (p.49).

Para Ander – Egg (citado por Tamayo y Tamayo 1998 Pág. 115). La muestra es el conjunto de operaciones que se realizan para estudiar la distribución de determinados caracteres en la totalidad de una población universo o colectivo partiendo de la observación de una fracción de la población considerada.

Las muestras pueden ser probabilísticas o no probabilísticas. Una muestra probabilística se elige mediante reglas matemáticas, por lo que la probabilidad de selección de cada unidad es conocida de antemano. Por el contrario, una muestra no probabilística no se rige por las reglas matemáticas de la probabilidad. De ahí que, mientras en las muestras probabilísticas es posible calcular el tamaño del error muestral, no es factible hacerlo en el caso de las muestras no probabilísticas. Ejemplos de éstas últimas son la muestra accesible (que está conformada por personas de fácil acceso para el investigador como ser colegas o alumnos de su clase.) y la muestra voluntaria (donde los sujetos de la muestra no han sido seleccionados matemáticamente).

### **3.5 Técnica e instrumentación de recolección de datos**

Para la obtención de información en el desarrollo de una investigación, es imprescindible la aplicación de técnicas que permitan recopilar los datos necesarios y el uso de instrumentos que faciliten el registro o almacenamiento de dichos datos para su posterior interpretación. Con el propósito de alcanzar los objetivos planteados, en este proyecto se emplearon distintas técnicas descritas a continuación:

**a) Observación directa:** Esta técnica es utilizada con el propósito de conocer y analizar la manera cómo son realizadas las actividades dentro de una organización. Según Tamayo y Tamayo (2003), “es aquella en la que el investigador puede observar y recoger datos mediante su propia observación.” (p.183).

**b) Entrevista no estructurada:** Este tipo de entrevistas se basa en la recopilación de información mediante la realización de preguntas que no presentan un orden preestablecido, sino que surgen de las respuestas del entrevistado. Al respecto Sabino (2002) señala que: “La entrevista no estructurada es una forma específica de interacción social que tiene por objeto recolectar datos para una investigación”.

### **3.6 Procesamiento y análisis de información.**

Una vez llevado a cabo la recopilación de los datos, es necesario organizarlos, describirlos y analizarlos. Tomando en consideración las técnicas e instrumentos utilizados, se empleó el análisis de contenido para el procesamiento de los datos recolectados, el cual es una técnica que permite reducir y sistematizar cualquier tipo de información cualitativa acumulada (documentos escritos, filmes, grabaciones, otros) en datos, respuestas o valores correspondientes a variables que investigan en función de un problema.

Al respecto Hurtado (2007) destaca que, “El análisis de contenido puede ser utilizado en investigaciones descriptivas, cuando se pretende hacer un diagnóstico y agrupar contenidos significativos de una serie de entrevistas, conversaciones y observaciones.” (p.57).

### **3.7 Fases metodológicas**

Proyecto: desarrollo de aplicación interactiva con inteligencia artificial para la validación de formularios web a partir de los parámetros establecidos por la metodología de desarrollo XP eXtremeProgramming y basado en los datos recolectados a varios testers de software, para darle mayor veracidad a la información obtenida mediante técnicas acerca de varias variables como sería el uso y la aceptación de un sistema automatizado debido a que los tests funcionales comprueban la ejecución, revisión y retroalimentación de las funcionalidades diseñadas para el software. Prueban y validan que el software hace lo que debe y, sobre todo, lo que se ha especificado. El

objetivo es comprobar que el sistema cumple con los requisitos funcionales: navegación, entrada de datos, procesamiento y obtención de resultados.

La metodología que se utilizara para el desarrollo de este proyecto es la XP (eXtremeProgramming). Esta metodología a su vez está conformada por un conjunto de reglas y prácticas. De una manera genérica las podemos agrupar por fases, como:

1. Planificación.
2. Diseño.
3. Desarrollo.
4. Pruebas.

### **Fase I: Planificación**

Durante esta fase se realizará una investigación exhaustiva de la problemática planteada para identificar la situación actual de la misma, que permitirá crear historias de usuario de forma que se pueda establecer la mejor manera para solventar la situación.

Actividad1: Aplicar entrevista

Actividad 2: Observar de forma directa el proceso de verificación de formularios web.

Actividad 3: Determinar los requerimientos funcionales y no funcionales.

### **Fase II: Diseño**

Se determinan los requerimientos del sistema como su orden de importancia, pudiendo planificar el tiempo de las iteraciones y las fechas de los planes de entrega de las versiones del sistema. La metodología XP propone realizar diseños simples y sencillos para evitar complicaciones, y de esta manera lograr un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.

En esta fase se crea la interfaz web y se diseñará el mapa de navegación del sistema y se establecerán elementos del software como colores, logo y fuentes.

Este proceso se realizará bajo las siguientes actividades:

Actividad 1: Se determinarán los requerimientos para el Diseño del software.

Descripción de los actores.

Descripción de casos de uso.

Modelo de clases.

Actividad 2: Diseño del sistema:

Diseño conceptual.

Diagrama de actividades.

Mapa de navegación.

Diagrama de clases.

Actividad 3: Diseño de la presentación.

Bosquejo de la interfaz.

### **Fase III: Desarrollo**

En esta fase se incluirá toda la parte de codificación o programación para el desarrollo del sistema, apoyada bajo los estándares de programación de tal manera que el código sea consistente, legible y escalable.

El proceso de desarrollo de este sistema se realizará bajo las siguientes actividades:

Actividad 1: Se describirán las herramientas de desarrollo.

Actividad 2: Se desarrollará e interconectará el módulo de la aplicación con la interfaz.

### **FASE IV: Pruebas**

Como etapa final, se someterá el sistema a pruebas de caja blanca y caja negra a nivel de unidad, de funcionalidad y de integración para comprobar su respectiva funcionalidad, de esta manera se logrará cumplir los requisitos planteados realizando las siguientes actividades:

Actividad 1: Se aplicarán las pruebas de caja blanca y caja negra en las cuales se verifican los fallos que tengan los sistemas, con la finalidad de corregirlos.

## **CAPÍTULO IV**

### **RESULTADOS**

En este capítulo se detallarán las actividades de cada fase, se mostrarán los resultados de las herramientas aplicadas, se anexará gráficos y diagramas que permitirán la fácil visualización y entendimiento de lo planteado.

#### **4.1 Fase I: Planificación del proyecto.**

##### **4.1.1 Actividad 1: Aplicar entrevista.**

Se aplicaron entrevistas a 4 testers y desarrolladores de software de la empresa de testing MakeItWork, esta técnica hace parte del proceso de recopilación de información del presente trabajo de grado. Con las siguientes preguntas:

1. ¿Cómo funciona en forma general el proceso de soporte y pruebas a formularios de registro?

“Primero se realiza un plan de prueba, determinando los requerimientos necesarios para la verificación y validación, en este plan de prueba se determinan cuales son los posibles valores de entradas y todas la posibles combinaciones a probar”.

2. ¿Cuáles son los principales problemas que se pueden observar en la verificación y validación de un formulario?
  - a) No se prueban todas las posibles combinaciones de entrada.
  - b) Se invierte más tiempo del que se estima.
  - c) El proceso es manual.

3. Describa las personas que intervienen en los procesos de soporte y pruebas de formularios.

“Los desarrolladores y testers”

4. ¿Cuánto tiempo invierte en la realización de las pruebas a los formularios web?

“Debido que el proceso es manual puede ocupar un tiempo al tester de dos a tres días, ingresando todas las posibles combinaciones de entrada”.

5. ¿Cómo se podría automatizar los procesos de soporte y pruebas a los formularios?

“Desarrollando un sistema que por medio de un algoritmo genere todas las posibles combinaciones de entrada de manera aleatoria y mucho mas rápida”

De esta manera se determinó que el proceso que se debe realizar para llevar a cabo las tareas es totalmente manual, lo que implica que se invierte mucho tiempo en la creación de combinaciones de entrada para la verificación y validación de formularios de registros.

Esta técnica de recolección de datos por medio de una entrevista, aporta de manera significativa información necesaria e indispensable para la realización de esta investigación.

#### **4.1.2 Actividad 2: Observar de forma directa el proceso de verificación de formularios web.**

En esta etapa se procedió a revisar diferentes formularios web con la finalidad de determinar cuáles serian los modelos a evaluar por los testers y desarrolladores, luego se entregan los documentos de casos de uso al ingeniero constructor de pruebas, es decir la etapa de construcción del software.

En la etapa de construcción de software se realiza la transformación de las especificaciones de diseño en un software ejecutable capaz de operar para satisfacer los requerimientos de los usuarios.

Como primera actividad del proceso de ejecución de las pruebas, se definirá la visión general de las mismas, las políticas generales, el marco general de operación y la estructura de recursos humanos que se necesitan.

La segunda actividad será construir el documento completo de pruebas y el cronograma general de la etapa de pruebas.

La tercera actividad será la ejecución de las pruebas: esta se inicia con la aplicación de los datos registrados en el documento de pruebas para obtener unos resultados y entrar en la etapa de evaluación y comparación con respecto a los resultados esperados. Si se presentan inconformidades con respecto a los resultados esperados se realiza el proceso de reporte de hallazgo.

La cuarta actividad es la elaboración de reportes de hallazgo: se elaborarán reportes de hallazgo, donde se especifica y soporta el error encontrado durante la prueba ejecutada.

#### **4.1.2.1 Planificación de las pruebas.**

Realizar el plan de pruebas brinda la posibilidad de recolectar las ideas y cristalizarlas en tareas concretas, puede ser visto como un medio de comunicación entre el equipo de pruebas y el cliente.

En el plan de pruebas se incluye:

Alcance del proyecto.

Calendario.

Requerimientos necesarios de hardware y software para ejecutar las pruebas.

Técnicas usadas para el diseño de las pruebas.

Procedimientos a seguir para el reporte y seguimiento de incidentes.

Especificación de qué reportes se entregarán al cliente al finalizar la evaluación del mismo.

#### **4.1.3 Actividad 3: Determinar los requerimientos funcionales y no funcionales.**

##### **4.1.4.1 Requerimientos Funcionales**

Evaluar los formularios mediante el ingreso del link.

Permitir el registro al usuario.

Los usuarios deben ingresar al sistema con un usuario y contraseña.

Detectar los identificadores (Link, Id, Type, Name), para hacer la evaluación correspondiente a cada campo de texto ingresando valores aleatorios para la verificación del mismo.

Generar un reporte en pdf con la evaluación del formulario y sus respectivas observaciones y recomendaciones.

Almacenar las capturas que realiza el sistema por cada campo que ingresa para la verificación y validación del mismo.

Almacenar los reportes de pruebas realizadas en el sistema.

Registrar y consultar el estado de las solicitudes que se hagan sobre las bases de datos autorizadas en las actividades de soporte y pruebas.

La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional además de un navegador web.

La aplicación debe poder utilizarse con los navegadores web Chrome, Firefox e Internet Explorer.

#### **4.1.4.2 Requerimientos no funcionales**

Correcto funcionamiento de la página principal.

El sistema debe proporcionar mensajes de error que sean informativos y orientados al usuario final.

La aplicación web debe poseer un diseño responsive a fin de garantizar la adecuada visualización en múltiples computadores personales y dispositivos tabletas.

Interfaz de usuario sencilla y agradable: Se muestra una interfaz intuitiva en la cual se integra las funcionalidades de la aplicación y una combinación de colores adecuados con tonalidades que van en contraste para una mayor experiencia con el usuario.

Portabilidad: La aplicación puede ser utilizada en diferentes navegadores Web.

#### **4.2 Fase II: Diseño**

Las pruebas de software se dividen en dos grandes grupos, funcionales y no funcionales. En esta investigación se tendrá en cuenta las funcionales, las cuales se aplican al producto final, luego de la fase de desarrollo, éstas permiten detectar cuáles son los puntos que no cumplen los requerimientos dados por el usuario además que puedan fallar en cuanto a la funcionalidad desarrollada.

Para generar los casos de pruebas funcionales se debe tener en cuenta la información de entrada que se necesita para obtener los casos de prueba a ejecutar. La información de entrada es:

1. Casos de uso del sistema.
2. Lista de verificación de aspectos a chequear en pruebas funcionales.
3. Plantilla caso de prueba.
4. Aplicación lista para probar. Siendo la salida: Casos de pruebas listos para la ejecución por el tester.

Luego que se definen los procedimientos de prueba, a partir de los ciclos funcionales que se seleccionan para ejecutar cada prueba, se especifican los scripts que las ejecutarán. Para cada script se debe definir la tarea que debe realizar y las verificaciones y/o validaciones que debe contener. Se codifican las pruebas obteniendo como resultado los script integrantes de las suites. Se verifica el correcto funcionamiento de cada script.

## 4.2.1 Actividad 1: Determinar los requerimientos para el diseño del software.

### 4.2.1.1 Descripción de los actores

Usuario: Probador del software.

Administrador de BD: El usuario y contraseña de conexión a una base de datos seleccionada, será administrado por este actor.

### 4.2.1.2 Modelo de casos de uso

El caso de uso Realizar Pruebas se puede apreciar en la figura 1, el cual muestra la interacción que tienen los usuarios con el sistema al momento de realizar las pruebas de software, en donde se tendrá que generar un reporte sobre la situación del formulario web.

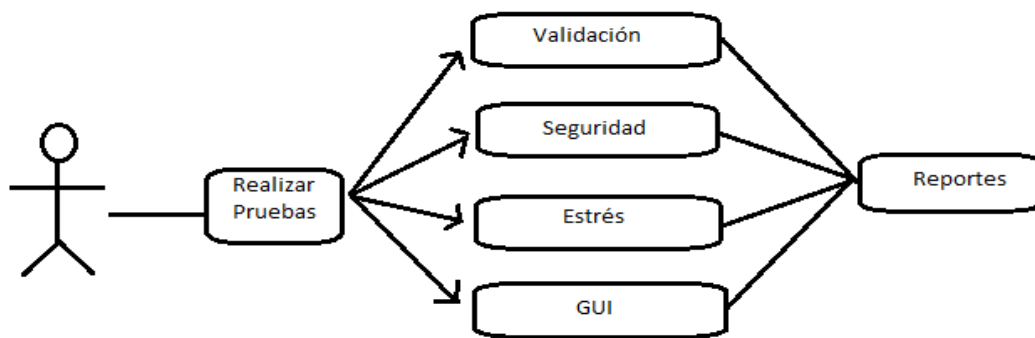
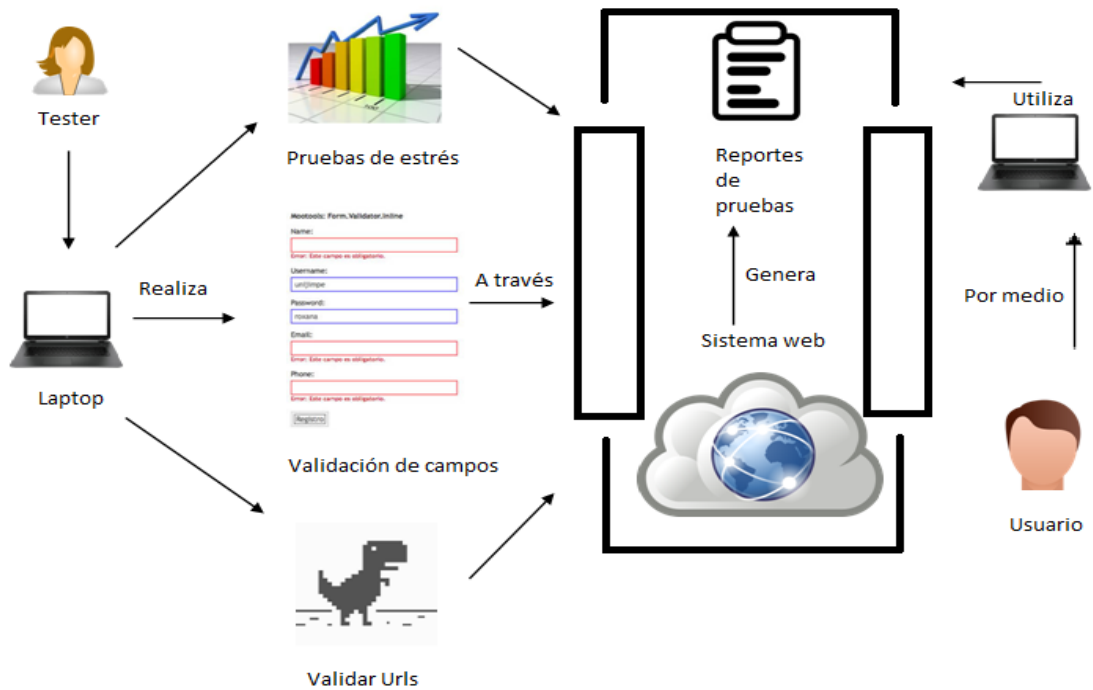


Figura 1. Caso de uso realizar pruebas

## 4.2.3 Actividad 2: Diseño del sistema

### 4.2.3.1 Diseño conceptual

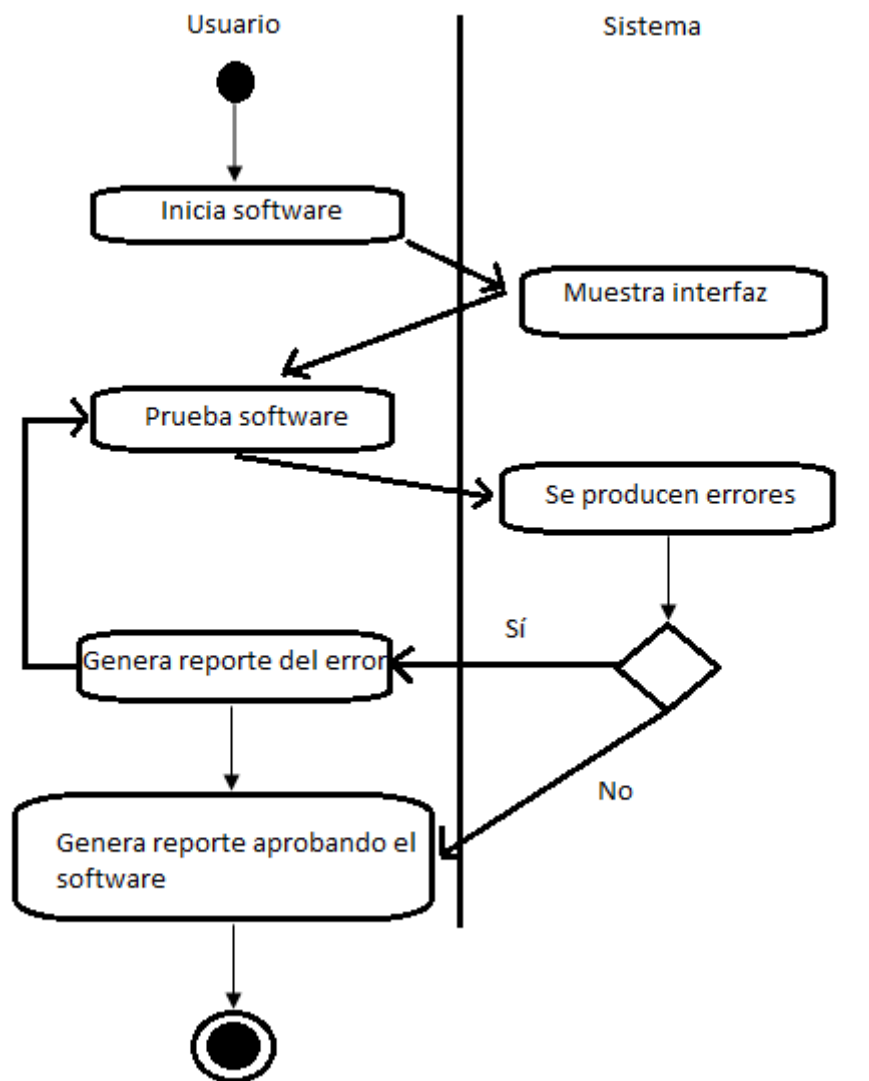
En este modelo conceptual de la figura 2 muestra las pruebas de software que se aplicarán al sistema como el flujo que debe seguir los testers de software para la ejecución de las mismas. En donde con ayuda de la interfaz que proporciona el sistema de QA Testing se tendrán que generar reportes del estado de la calidad del software.



**Figura 2. Modelo Conceptual.**

#### 4.2.3.2 Diagrama de actividades

Se considera el diagrama de actividades tanto una variante del diagrama de estados como de los diagramas de interacción, ya que sirve para describir los estados de una actividad, que es un conjunto de acciones en secuencia y/o concurrentes en el cual intervienen clasificadores.



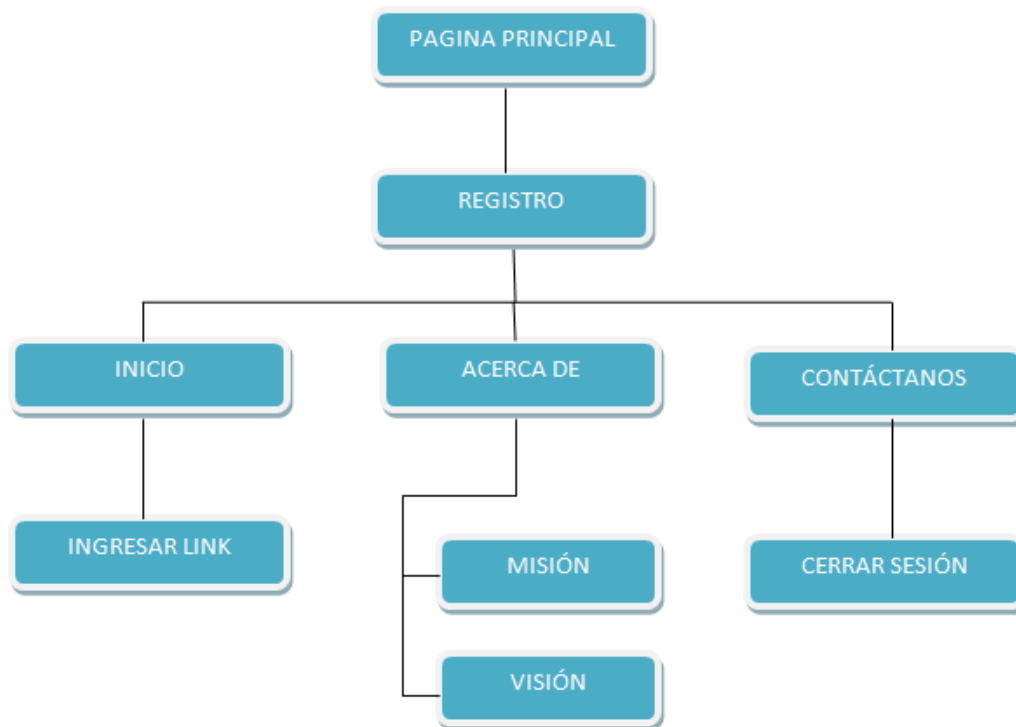
**Figura 3. Diagrama de actividades**

En la figura 3 muestra la interacción entre los probadores de software (usuario) y el sistema a probar, donde el actor que inicia con la actividad son los testers, ejecutando el software y probándolo hasta encontrar los errores posibles que pueda tener el formulario web. Una vez que se encuentran los errores el sistema deberá generar un reporte que contenga las características en las cuales ocurrió la falla y sus posibles soluciones.

#### **4.2.3.3 Mapa de navegación**

Los mapas de navegación proporcionan una representación esquemática de la estructura de hipertexto, indicando los principales conceptos incluidos en el espacio de

la información y las interrelaciones que existen entre ellos. Es una representación completa o resumida del sitio web para orientar al usuario durante el recorrido y facilitarle un acceso directo al lugar que le interese.

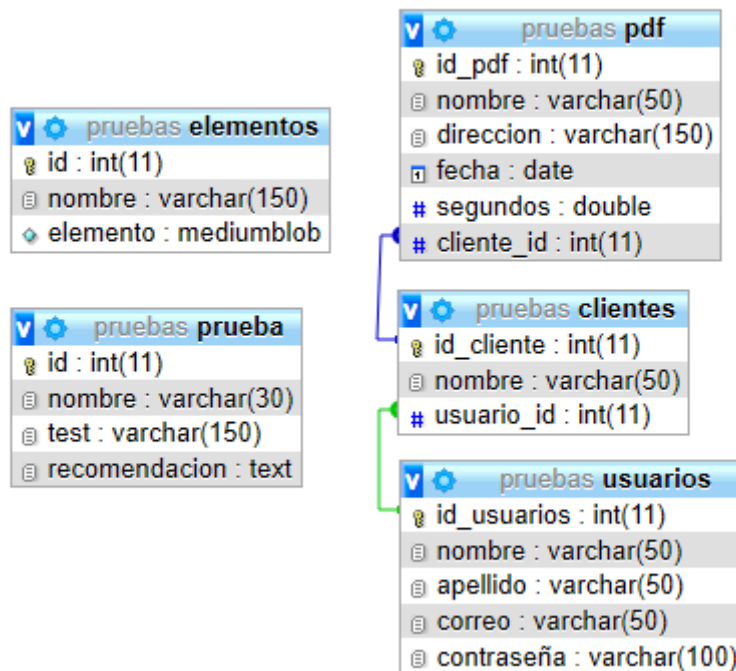


**Figura 4. Mapa de navegación**

En la figura 4 muestra el mapa de navegación del sistema, para facilitar la ruta del usuario o probador de software.

#### **4.2.3.4 Diagrama de clases**

El diagrama de entidad-relación es una herramienta de modelado de datos que describen las asociaciones que existen entre las diferentes categorías de datos dentro de un sistema de una empresa o de información. La figura 5 muestra las entidades existentes, así como sus relaciones en el sistema de QA Testing propuesto para la verificación y validación de formularios web.

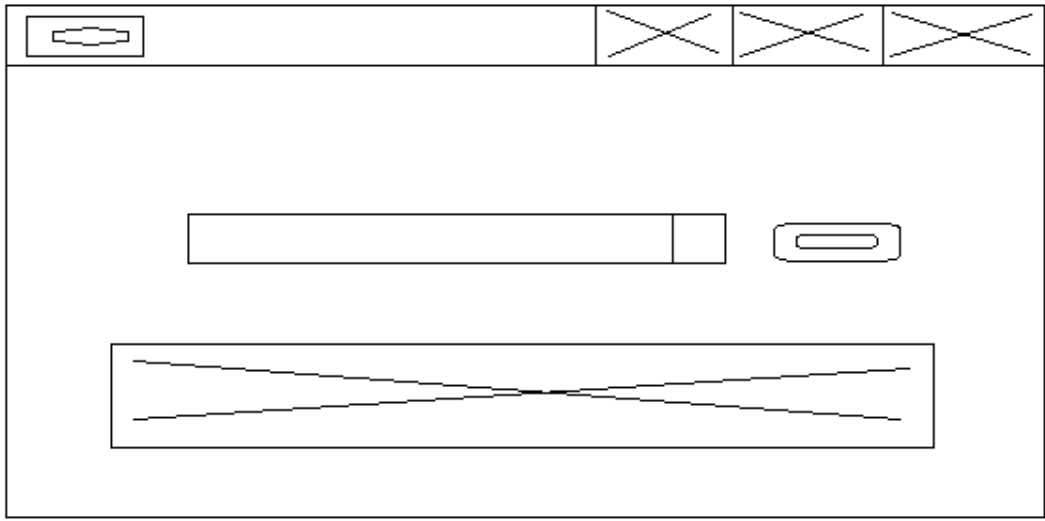


**Figura 5. Diagrama de clases**

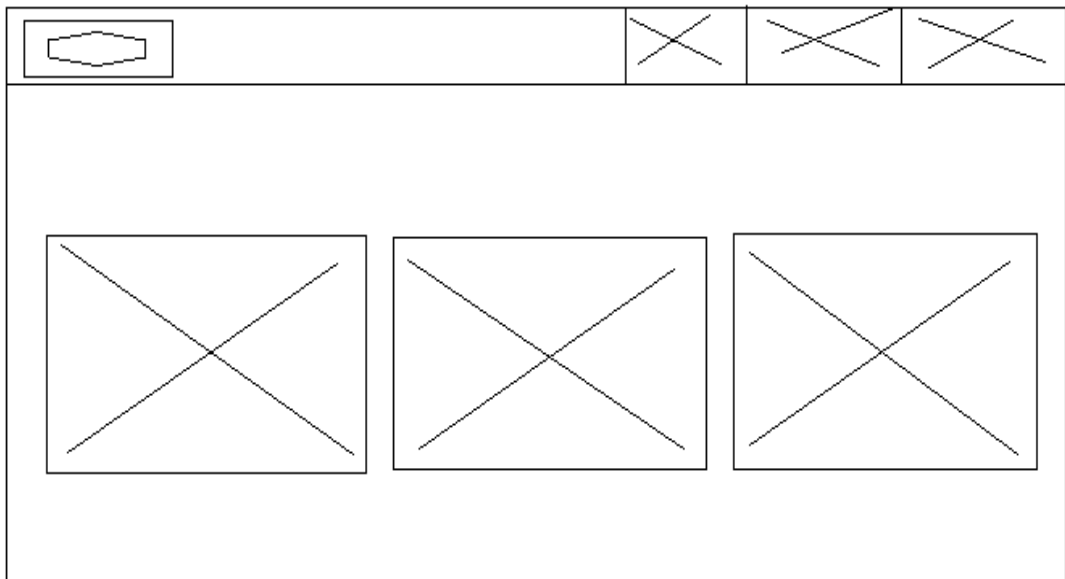
### 4.2.3 Actividad 3: Diseño de la interfaz

El diseño de la interfaz representa un concepto básico y temprano de las interfaces gráficas que conforman al sistema, por lo general suelen ser trazos simples sin tanto formalismo, con el fin de cubrir los elementos y la navegación que conformarán al software. En la industria del desarrollo móvil y web suelen llamarse mockups y aunque también existen wireframes el primero tiende a ser más simple, mientras que el segundo representa una copia más real del producto final. Las figuras 6, 7 y 8 representan los menús de navegación y el formulario de registro así también como el cuadro de ingreso del link que se utilizará para evaluar el enlace de la página la cual contiene el formulario.

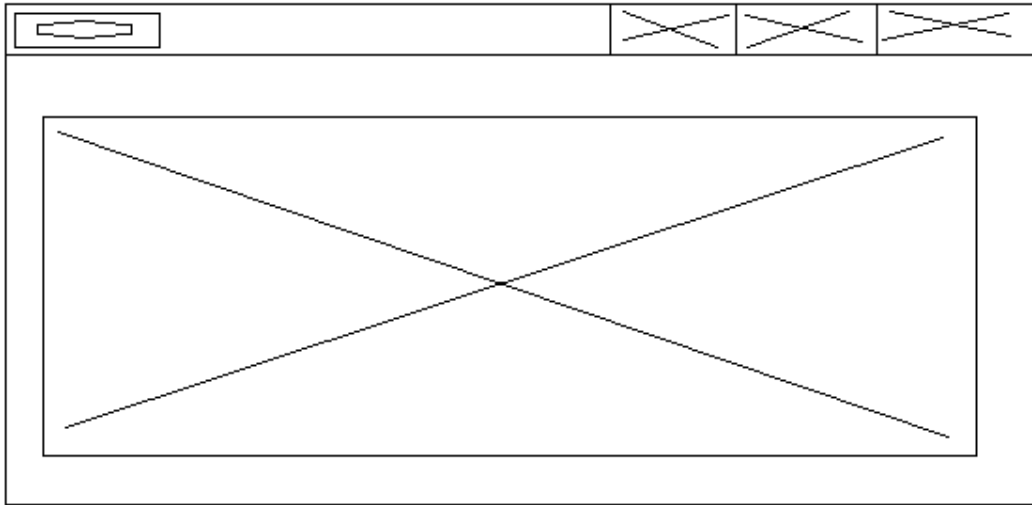
A continuación se muestra un bosquejo de lo que será el sistema:



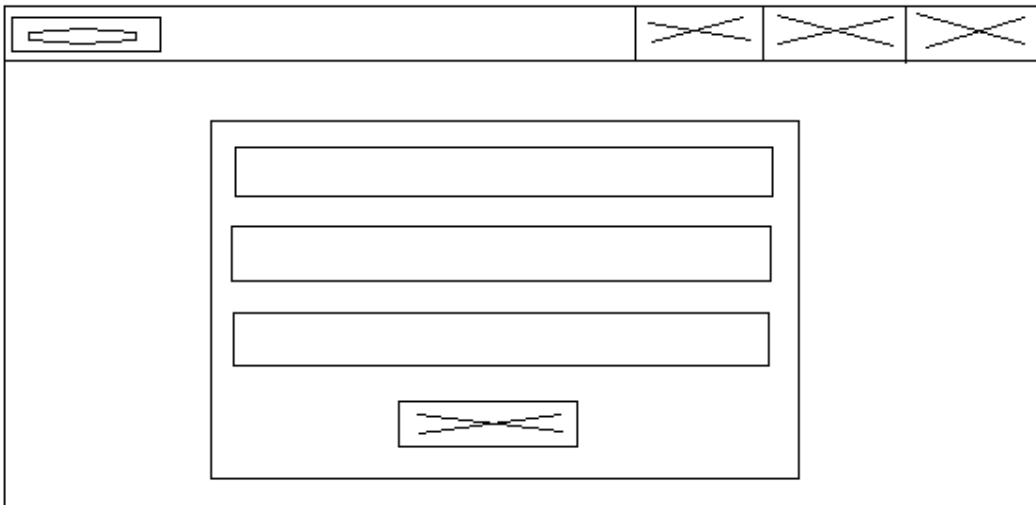
**Figura 6.** Página Principal



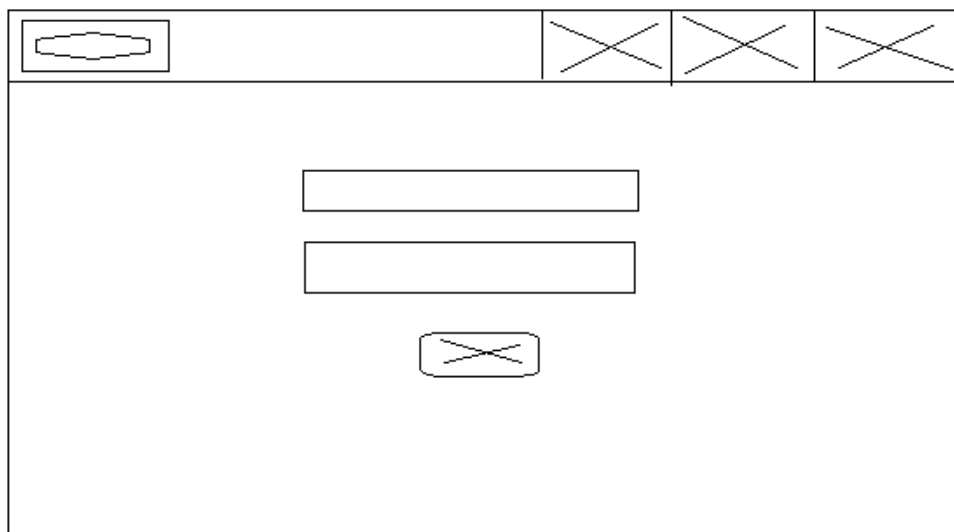
**Figura 7.** Inicio



**Figura 8.** Acerca de



**Figura 9.** Contáctanos



**Figura 10. Registrarse**

### **4.3 FASE III: Desarrollo**

#### **4.3.1 Actividad 1: Se describirán las herramientas de desarrollo**

El entorno de desarrollo que se emplea para la realización de este trabajo de grado es Visual Studio 2017, y para el desarrollo de la interfaz se usó el editor de texto Atom.

El grupo de herramientas Selenium que se emplearon para la ejecución de pruebas, está constituido por SeleniumCore, Selenium IDE y SeleniumRemote Control. Selenium es un framework que nos permite crear y ejecutar pruebas automatizadas sobre aplicaciones web, en este caso se empleó en formularios Web.

Incluye características como grabación, playback, selección de campos, autocompletar formularios, pruebas de recorrido, debug y otros formatos.

La herramienta principal de Selenium es SeleniumCore. Es una herramienta de automatización desarrollada inicialmente por programadores y testers de ThoughtWorks y actualmente por OpenQA. Es una aplicación Web desarrollada con HTML y Javascript que soporta una gran variedad de plataformas y navegadores. Permite ejecutar una prueba o conjunto de pruebas automatizadas.

Selenium IDE permite crear scripts de pruebas de manera rápida ya que trabaja como una “grabadora” que captura todas las acciones que se hacen sobre la interfaz de usuario, generando automáticamente los comandos del Scripts de prueba. Esta

herramienta trabaja como un complemento de Firefox, el cual permite grabar y reproducir de manera simple las interacciones con el browser.

Otras características:

1. Manejo inteligente de la identificación de campos o elementos en la página Web, usando IDs, Xpath, CssSelector entre otros.
2. Autocompleta todos los comandos con Selenium.
3. Permite grabar los Scripts en formato HTML, JUnit, entre otras.
4. Soporte para el archivo “user-extensions.js”, el cual permite agregar funciones javascript personalizadas.
5. Opciones para realizar validaciones automáticas del título para cada página en la que se esté navegando.

#### **4.3.2 Actividad 2: Se desarrollará e interconectara el módulo de la aplicación con la interfaz.**

A continuación se muestran capturas del sistema:



**Figura 1. Pantalla de inicio del sistema.**

En esta figura se muestra el inicio del sistema, el cual contiene un campo de texto donde se ingresa el link de la página del formulario que se va a evaluar junto al botón de realizar prueba.



**Figura 2. Continuación de inicio.**

En esta figura mostramos algunos ejemplos de las capturas que el sistema puede realizar al momento del análisis del sistema.



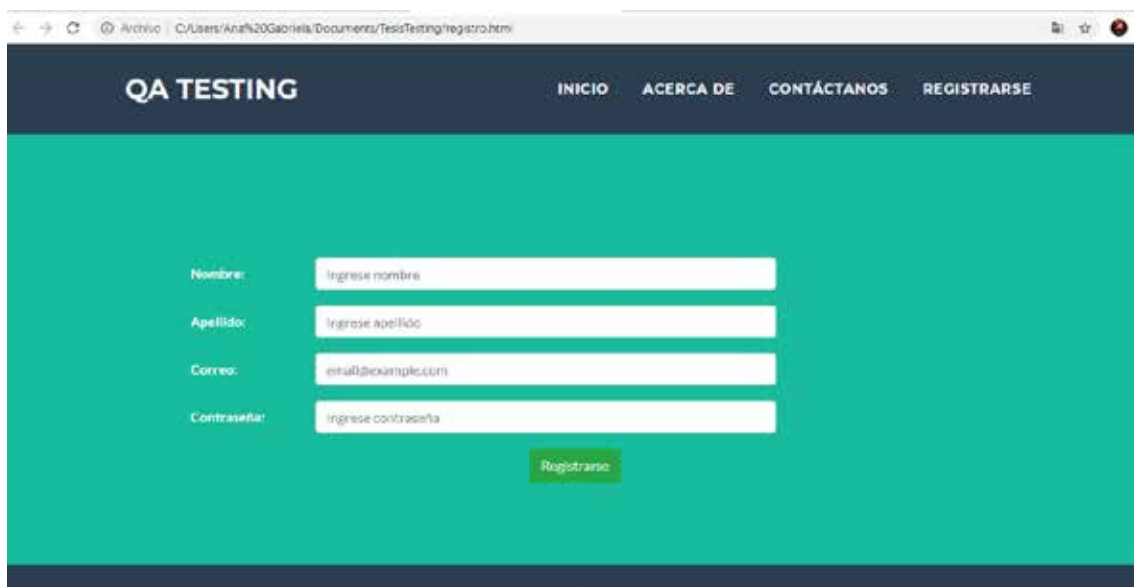
**Figura 3. Acerca de**

En la figura 3 se da una breve descripción de la misión y visión en la que se basa el sistema QA Testing.



**Figura 4. Contáctanos**

En la figura 4 se muestra un formulario donde el cliente puede comunicarse con nosotros a través de un mensaje por correo electrónico y enviarnos sus comentarios o sugerencias.



**Figura 5. Registrarse**

En la figura 5 se muestra un formulario donde el cliente puede registrarse para poder hacer uso del sistema QA Testing.



**Figura 6. Iniciar sesión**

En la figura 6 se muestra un login donde el cliente puede iniciar sesión para poder hacer uso del sistema QA Testing.

#### **4.4 FASE IV: Pruebas**

**4.4.1 Actividad 1:** Se aplicarán las pruebas de caja blanca y caja negra en las cuales se verifican los fallos que tengan los sistemas, con la finalidad de corregirlos.

##### **4.4.1.2 Pruebas de Caja Blanca**

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: 1) garantice que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez. (Pressman, 2015).

Las siguientes tablas muestran el formato a utilizar para las capturas de las pruebas de validación donde será importante contar con los resultados esperados para confirmar que coincidan con los obtenidos por el sistema.

**Tabla 1.** Página principal

<b>Tipos de prueba:</b>	Caja Blanca
<b>Caso de prueba:</b>	Página Principal
<b>Prueba Número 1</b>	Verificar opciones del menú
<b>Descripción</b>	Recorrer el menú principal verificando su correcto funcionamiento y distribución dentro de la vista principal.
<b>Entrada</b>	-
<b>Resultado esperado</b>	Distribución y correcto funcionamiento.
<b>Resultado</b>	Distribución y correcto funcionamiento.
<b>Observaciones</b>	-
<b>Solución</b>	-

**Fuente:** Guzmán, Miranda(2019)

**Tabla 2.** Validación del link

<b>Tipo de prueba:</b>	Caja Blanca
<b>Caso de prueba:</b>	Ingreso del link de la página a evaluar
<b>Prueba Número 2</b>	Prueba de campo vacío
<b>Descripción</b>	Se pulsa en el botón de realizar prueba sin ingresar datos en el campo del link.
<b>Entrada</b>	-
<b>Resultado esperado</b>	Mostrar mensaje de campo vacío
<b>Resultado</b>	Por favor ingresar link a evaluar

<b>Observaciones</b>	No se ingresó ningún valor
<b>Solución</b>	Ingresar link valido

**Fuente:** Guzmán, Miranda (2019)

**Tabla 3.** Validación del link

<b>Tipo de prueba:</b>	Caja Blanca
<b>Caso de prueba:</b>	Ingreso del link de la página a evaluar
<b>Prueba Número 3</b>	Probar datos inválidos
<b>Descripción</b>	Se ingresan números en lugar de un link
<b>Entrada</b>	Dato no válido: 0123456789
<b>Resultado esperado</b>	Mensaje de alerta
<b>Resultado</b>	Mensaje en alerta: “Ingrese un link valido”
<b>Observaciones</b>	Se ingresó números en lugar de un link
<b>Solución</b>	Ingresar un link valido

**Fuente:** Guzmán, Miranda(2019)

**Tabla 4.** Validación del link

<b>Tipo de prueba:</b>	Caja Blanca
<b>Caso de prueba:</b>	Ingreso del link de la página a evaluar
<b>Prueba Número 4</b>	Realizar prueba
<b>Descripción</b>	Se ingresa un link
<b>Entradas</b>	<a href="https://id.spreedly.com/singup?submissionGuid=9fa589e1">https://id.spreedly.com/singup?submissionGuid=9fa589e1</a>
<b>Resultado esperado</b>	Redireccionamiento a la página principal donde se encuentra el formulario que va a ser evaluado.

<b>Resultado</b>	Redireccionamiento a la página principal donde se encuentra el formulario que va a ser evaluado.
<b>Observaciones</b>	-
<b>Solución</b>	-

**Fuente:** Guzmán, Miranda(2019)

**Tabla 5.** Formulario

<b>Tipo de Prueba:</b>	Caja Blanca
<b>Caso de Prueba:</b>	Validar Formulario
<b>Prueba Número 5</b>	Ingreso de valores
<b>Descripción</b>	Ingresar letras mayúsculas
<b>Entrada</b>	ASDFSADF45645654
<b>Resultado esperado</b>	Mensaje de error
<b>Resultado</b>	Mensaje de error
<b>Observaciones</b>	Los campos tipo nombre y apellido, deben aceptar solo letras y no números o correos.
<b>Solución</b>	Escribir solo letras

**Fuente:** Guzmán, Miranda(2019)

**Tabla 6.** Formulario

<b>Tipo de Prueba:</b>	Caja Blanca
<b>Caso de Prueba:</b>	Validar Formulario
<b>Prueba Número 6</b>	Ingreso de valores
<b>Descripción</b>	Ingresar números
<b>Entradas</b>	0123456789

<b>Resultado esperado</b>	Mensaje de error
<b>Resultado</b>	Mensaje de error
<b>Observaciones</b>	Los campos de teléfonos solo deben aceptar números
<b>Solución</b>	Se deben ingresar números solo en el campo contactos

**Fuente:** Guzmán, Miranda(2019)

#### 4.4.1.3 Pruebas de Caja Negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca.

Las pruebas de caja negra intentan encontrar errores en las categorías siguientes: 1) funciones incorrectas o faltantes, 2) errores de interfaz, 3) errores en las estructuras de datos o en el acceso a bases de datos externas, 4) errores de comportamiento o rendimiento y 5) errores de inicialización y terminación.

#### **Prueba 1 Caja Negra. Sistema QA Testing**

<b>Nro Secuencia</b>	<b>Módulo</b>	<b>Acción</b>	<b>Resultado</b>	<b>Observaciones</b>
1	Página inicial	Menú	Correcto	-
2	Página inicial	Contáctanos	Correcto	-
3	Página inicial	Ingreso link	Correcto	-

**Fuente:** Guzmán, Miranda(2019)

#### **Prueba 2 Caja Negra. Formulario**

<b>Nro Secuencia</b>	<b>Elemento</b>	<b>Acción</b>	<b>Resultado</b>	<b>Observaciones</b>
1	Formulario	Nombre	Correcto	-
2	Formulario	Apellido	Correcto	-
3	Formulario	Teléfono	Correcto	-
4	Formulario	Dirección	Correcto	-
5	Formulario	Correo	Correcto	-

**Fuente:** Guzmán, Miranda(2019)

## CONCLUSIONES

La calidad en un producto de software no se puede mirar como una opción en los tiempos actuales donde los usuarios son más exigentes en cuanto a los servicios y productos que consumen de las empresas, y donde una vulnerabilidad puede ser aprovechada para afectar a millones de usuarios con su información o incluso puedan verse afectadas sus finanzas o las de la misma empresa.

En este trabajo de grado se ha conseguido implementar un sistema que cumple los objetivos establecidos y durante su desarrollo se concluyó.

Para poder contar con aplicaciones de calidad es necesario realizar un diagnóstico para determinar los requerimientos necesarios para su correcto desarrollo y realizar un control de seguimiento sobre las pruebas.

La permanente comunicación con los clientes dio como resultado una buena retroalimentación que se refleja en el éxito de este sistema de evaluación de formularios y en el cumplimiento de los objetivos.

Se mantuvo un especial cuidado en la elección de las herramientas de desarrollo y arquitectura del proyecto desde la recopilación de información para poder llevar a cabo la investigación bajo niveles de calidad; como se ha demostrado a lo largo de este documento, las herramientas utilizadas para el desarrollo del QA Testing, son herramientas de última tecnología lo bastante robustas y completas que permiten tener seguridad y confianza sobre el producto final que se está entregando con este documento.

El sitio web generado para la evaluación de las pruebas resulta agradable a la vista, permitiendo al usuario una navegación fácil y rápida. Durante la definición del sistema QA Testing, se han ingresado más de 20 links de formularios a evaluar y en diferentes navegadores y ha demostrado trabajar bien en cualquiera de éstos.

Se puede decir que para lograr la calidad de un producto de software es necesario integrar una metodología del proceso de pruebas al ciclo de vida del software, el cual permita realizar las actividades de lo que concierne a calidad durante todo el proceso que lleva el desarrollo de un software. Esto implica involucrar las pruebas desde las fases de análisis para ir semblanteando el desarrollo de las mismas, pasando por la codificación donde pueden estar involucradas las pruebas de caja negra o caja blanca dependiendo del tipo de prueba que se vaya a realizar, y así hasta la fase de

actualización que permita al software adaptarse a las nuevas necesidades del mercado o errores que no fueron descubiertos durante las fases previas para así garantizar la calidad y la mejora continua del software.

## **RECOMENDACIONES**

Por medio de una metodología que garantice la retroalimentación de la información obtenida durante el proceso de pruebas del software se puede asegurar la correcta aplicación de las pruebas de software y del plan de pruebas en el cual estén contempladas. Modelos para la aplicación de pruebas hay muchos, con sus respectivas ventajas y desventajas, pero lo importante al utilizar cualquier modelo, consiste en usar el que mejor se adecue al equipo de trabajo y el que permita alcanzar los objetivos deseados. Además, nunca se debe olvidar el objetivo principal de lograr una calidad óptima que complazca al usuario final, ya que este será el que interactúa con el software y de quien dependerá que el software se utilice o quede en el abandono.

Se observa que el proceso de pruebas impacta en los riesgos del producto software; por ende, para las empresas de software es vital formular y adecuar un buen proceso de pruebas de calidad de software.

La mayoría de las universidades en Latinoamérica han estado aisladas en cuanto a la implementación de pruebas de calidad de software y la generación de espacios de laboratorio para el desarrollo de técnicas, indagaciones e investigación en las áreas de informática, sistemas o afines, que permitan afianzar el aseguramiento de la calidad de los productos software.

Con la claridad de la importancia del proceso de pruebas de calidad en los productos software se eleva la calidad y fiabilidad dentro del ciclo de vida de un proyecto. Asimismo, al implementar un proceso de pruebas de calidad de software se genera un control y seguimiento a los defectos o faltas y a los fallos, de manera que las soluciones que se generen tengan un mayor nivel de calidad.

En un sentido general, la implementación de un proceso de pruebas de calidad de software en las empresas o universidades instituye un gran avance en el intento por garantizar márgenes de calidad en los productos de desarrollo software.

## BIBLIOGRAFÍAS

Balestrini, Alberto (2009) Como se Elabora un Proyecto de Investigación. Caracas-Venezuela.

trabajo de grado titulado **Evaluación de accesibilidad de sitios web de las Universidades Públicas Peruanas**, realizado en la Unidad Católica Del Perú.

Quintanilla Paredes, José Guillermo; Rodríguez Mantilla, José Luis (2017) **Sistema experto basado en redes neuronales para mejorar la determinación del impuesto predial en la Municipalidad Distrital de La Esperanza** realizado en la Universidad Nacional de Trujillo.

Castañeda Rojas, Edson Bryan trabajo de grado titulado **Propuesta de patrón de diseño de software orientado a prevenir la extracción automatizada de contenido web** realizado PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ.

Alvarez Díaz, Jhon Manuel; Torres Aguilar, Miguel Angel (2014) trabajo de grado titulado **Desarrollo de un portal web que permita mejorar la gestión y el acceso a la información para la Escuela de Informática de la Universidad Nacional de Trujillo** realizado en la Universidad Nacional de Trujillo.

Roberto Luis Felipe y Vargas Guevara (2012) trabajo de grado titulado **Desarrollo de aplicación interactiva para enseñar vocabulario en inglés a niños desde edad de preescolar** realizado en la UNIVERSIDAD DE CHILE FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN.

Almudena Jiménez Sierra, Fco. Javier Muñoz Rodríguez, Gonzalo Torres Porta (2006) trabajo de grado titulado **Aplicación de métodos de Inteligencia Artificial para la toma de decisiones en simulación de móviles**. Realizado en Universidad Complutense de Madrid, Facultad de Informática.

Rojas-Montes, M. L., Pino, F. J., & Martínez, J. M. (2015). **Proceso de pruebas para pequeñas organizaciones desarrolladoras de software**. Facultad de Ingeniería, 24(39), 55-70.

Jústiz-Núñez, Dalila, Gómez-Suárez, Darlene, & Delgado-Dapena, Marta Dunia. (2014). **Proceso de pruebas para productos de software en un laboratorio de calidad**. Ingeniería Industrial. 35(2), 131-145.

Luis González Varela (2012). **Introducción al Software Testing.** – eseida.wikispaces.<https://eseida.wikispaces.com/file/view/Introducci%C3%B3n+al+Software+Testing.pdf> [Accessed 19 Sep 2016] .

Ammann, Paul y Offutt, Jeff.**Introduction to Software Testing.**Cambridge University Press, 2008.

Anuradha K. (2016). **Important Software Test Metrics and Measurements – Explained with Examples and Graphs.** – Software Testing Help. [online] Available at:<http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/> [Accessed 26 Dec 2016]

Estrada, A. F., García, T. C., Perdomo, Y. L., Cintra, A. V., Martínez, R. D., & Díaz, R. C. (2011). **Una experiencia novedosa para el testing desarrollada por un departamento de pruebas de software.** Revista Cubana de Ciencias Informáticas, 5(2).

Gómez Reynoso, Juan Manuel; Andrade, Estela Lizbeth Muñoz; and MacíasDíaz, Jorge Eduardo. (2010). "**Utilizando el Modelo de Calidad de McCall y el Estándar ISO-9126 para la Evaluación de la Calidad de Sistemas de Información por los Usuarios**". AMCIS 2010 Proceedings. Paper 89.