



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Sistemas

**Implementación del Framework de automatización de
proceso de QA en un proyecto de diseño de software en
una consultora**

INFORME DE TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Edwin CAPCHA CORONADO

ASESOR

Rubén Alexander GIL CALVO

Lima, Perú

2018



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Capcha, E. (2018). *Implementación del Framework de automatización de proceso de QA en un proyecto de diseño de software en una consultora*. [Informe de trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Sistemas]. Repositorio institucional Cybertesis UNMSM.



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Sistemas

**Acta de Sustentación del
Trabajo de Suficiencia Profesional**

Siendo las ^{20:05} horas del día ¹⁷ de diciembre del año 2018, se reunieron los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Ing. Armas Calderón Raul Marcelo (Presidente), Mg. Mac Dowall Reynoso Erwin. (Miembro) y el Lic. Gil Calvo Ruben Alexander (Miembro Asesor) para la sustentación del Trabajo de Suficiencia Profesional Intitulado: **“IMPLEMENTACION DEL FRAMEWORK DE AUTOMATIZACION DE PROCESO DE QA EN UN PROYECTO DE DISEÑO DE SOFTWARE EN UNA CONSULTORA”**, por el Bachiller: **Capcha Coronado, Edwin**; para obtener el Título Profesional de Ingeniero de Sistemas.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecida por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de .16. (En letras) DIECISEIS...

A continuación el presidente de jurados el Ing. , declara al Bachiller Ingeniero de Sistemas.

Siendo las ^{20:30} horas, se levantó la sesión.

Presidente
Ing. Armas calderón Raúl Marcelo

Miembro
Mg. Mac Dowall Reynoso Erwin.

Miembro Asesor
Lic. Gil Calvo Ruben Alexander

FICHA CATALOGRÁFICA

IMPLEMENTACIÓN DEL FRAMEWORK DE AUTOMATIZACIÓN DE PROCESO DE QA EN UN PROYECTO DE DISEÑO DE SOFTWARE EN UNA CONSULTORA.

AUTOR: EDWIN CAPCHA CORONADO

ASESOR: RUBEN ALEXANDER GIL CALVO

LIMA – PERU, 2018

Título Profesional/Grado Académico: Título Profesional de Ingeniero de Sistemas

Área/Programa/Línea de Investigación: Ingenierías / Tecnología de Información y Comunicación / Ingeniería de Software

Pregrado: Universidad Nacional Mayor de San Marcos – Facultad de Ingeniería de Sistemas e Informática – Escuela Profesional de Ingeniería de Sistemas

Formato 28 x 20 cm

Páginas: xii, 83

DEDICATORIA

A mi familia,

especialmente a

mi madre y padre

Quienes han dado todo su amor y su apoyo incondicional

ya que considero que esto se lo debo a ellos debido a que

soy producto de todo su esfuerzo, dedicación y perseverancia

AGRADECIMIENTOS

Gracias a mi asesor Rubén Gil Calvo, por su dedicación, motivación, criterio y aliento. Ha hecho fácil lo difícil. Ha sido un privilegio poder contar con su guía y ayuda.

Gracias a la magnífica Comisión del Programa de Titulación por Trabajo de Suficiencia Profesional de la Escuela Profesional de Ingeniería de Sistemas y de la Escuela Profesional de Ingeniería de Software de la Facultad de Ingeniería de Sistemas e Informática de la UNMSM, por facilitar los mecanismos administrativos y educativos para posibilitar el acceso a los bachilleres en Ingeniería de Sistemas y en Ingeniería de Software para la elaboración del Informe de trabajo de suficiencia profesional; y así preparar a los bachilleres en la obtención del Título Profesional de Ingeniero de Sistemas y de Ingeniero de Software.

Gracias a los profesores que nos enseñaron estos grandiosos cursos en la facultad de Ingeniería de Sistemas y en Ingeniería de Software, por su dedicación, apoyo, colaboración y conocimiento; lo que nos permitió afrontar con éxito los trabajos grupales, presentaciones, exámenes y reuniones que posibilitaron aprobar satisfactoriamente los cursos que formaron parte del Programa de Titulación por Trabajo de Suficiencia Profesional.

Finalmente y no último, gracias a mi familia, sin quienes este esfuerzo nada hubiera sido posible.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

**IMPLEMENTACIÓN DEL FRAMEWORK DE AUTOMATIZACIÓN PARA DE PROCESO DE QA EN
UN PROYECTO DE DISEÑO DE SOFTWARE EN UNA CONSULTORA**

Autor: Capcha Coronado., Edwin

Asesor: Gil Calvo, Rubén Alexander

Título: Informe de Trabajo de Suficiencia Profesional para optar el Título Profesional de Ingeniero de Sistemas

Fecha: Diciembre del 2018

RESUMEN

Actualmente muchas empresas siguen ciertas metodologías de desarrollo de software como el modelo en cascada, prototipado, incremental, scrum entre otros para así poder estructurar, planificar y controlar el proceso de desarrollo de software. La tendencia de la mayoría de estas empresas es utilizar metodologías ágiles, las cuales permiten una entrega de productos de software más rápido en comparación con las que siguen una metodología tradicional. Scrum es una metodología Ágil adaptable a los procesos de desarrollo de las empresas. Uno de los principales procesos es el de QA, este último busca garantizar que las especificaciones de los requerimientos mantengan concordancia antes, durante y después del desarrollo de software. Belatrix es una consultora de software que brinda servicios de software a otras empresas, utiliza la metodología Scrum y siempre redefine sus procesos para dar un mejor servicio. Uno de los procesos que Belatrix ha mejorado es el proceso de QA. Para el desarrollo óptimo de la ejecución de este proceso, se utilizará el framework de automatización para el proceso de QA, el cual busca apoyar al proceso en la reducción del tiempo de la actividad de la regresión, el cual consiste en la ejecución de todos los casos de pruebas.

Palabras claves: QA, actividades de QA, Proceso de QA, Optimización de actividades de QA

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF SYSTEMS AND COMPUTER ENGINEERING

PROFESSIONAL SCHOOL OF SYSTEMS ENGINEERING

**Implementation of automation framework of a QA Process in a Project of Software Design
In a Company**

Author: Capcha Coronado, Edwin

Advisor: Gil Calvo, Ruben Alexander

**Title: Professional Sufficiency Work Report for opt for the Professional Title
of Systems Engineer**

Date: December 2018

ABSTRACT

Currently many companies follow certain software development methodologies such as cascading, prototyping, incremental, scrum and others in order to structure, plan and control the software development process. The tendency of most of these companies is to use agile methodologies, which allow a faster delivery of software products compared to those that follow a traditional methodology. Scrum is an Agile methodology adaptable to the processes of business development. One of the main processes is that of QA, the latter seeks to ensure that the specifications of the requirements maintain consistency before, during and after software development. Belatrix is a software consultancy that provides software services to other companies, uses the Scrum methodology and always redefines its processes to provide a better service. One of the processes that Belatrix has improved is the QA process. For the optimal development of the execution of this process, the automation framework for the QA process will be used, which seeks to support the process in reducing the time of the regression activity, which consists of the execution of all the test cases.

Key words: QA, QA activities, QA process, Optimization of QA activities

INDICE

CARATULA EXTERNA	i
PÁGINA EN BLANCO	ii
CARATULA INTERNA	iii
FICHA CATALOGRÁFICA	iv
DEDICATORIA	v
AGRADECIMIENTOS	vi
RESUMEN	vii
ABSTRACT	viii
INDICE	ix
INDICE DE FIGURAS	x
INDICE DE TABLAS	xi
INTRODUCCION	1
CAPITULO I - TRAYECTORIA PROFESIONAL	2
CAPITULO II - CONTEXTO EN EL QUE SE DESAROLLO LA EXPERIENCIA	3
2.1 EMPRESA - ACTIVIDAD QUE REALIZA	4
2.2 VISION	4
2.3 MISION	4
2.4 ORGANIZACION DE LA EMPRESA	5
2.5 AREA, CARGO Y FUNCIONES DESEMPEÑADAS	7
2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN	7
CAPITULO III - ACTIVIDADES DESARROLLADAS	8
3.1 SITUACION PROBLEMÁTICA	8

3.1.1 DEFINICION DEL PROBLEMA	8
3.2 SOLUCION	9
3.2.1 OBJETIVO GENERAL	9
3.2.2 ALCANCE	9
3.2.3 ETAPAS Y METODOLOGIA	9
CAPITULO IV. REFLEXION CRÍTICA DE LA EXPERIENCIA	20
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	36
5.1 CONCLUSIONES	36
5.2 RECOMENDACIONES	36
5.4 FUENTES DE INFORMACIÓN	36
5.5. GLOSARIO	37
ANEXOS	40
1. INSTALAR TESTLINK	44
2. INSTALACION SELENIUM	45
3. INSTALACION DE VISUAL STUDIO	47

INDICE DE FIGURAS

Figura 1-Organigrama de Consultora Belatrix	18
Figura 2-Metodología Scrum belatrix	19
Figura 3-Actividades de QA	21
Figura 4 - Actividades de QA Incluido Automatización de Casos de pruebas	24
Figura 5-Instalacion del Visual Studio Community	25
Figura 6-Añadir un Nuget al visual Studio	26
Figura 7-Añadir Selenium Nuget al visual Studio	26
Figura 8 -Añadir Nunit Nuget al visual Studio	27
Figura 9 -Conexión con Base de Datos y Url de los Apis	28
Figura 10. Las carpetas del proyecto	29
Figura 11. La clase TestCommon	30
Figura 12. Caso de prueba E2E	30
Figura 13. Caso de prueba Automatizado E2E	31

INDICE DE TABLAS

Tabla 1 – Tabla por sprint en los últimos 6 meses	17
Tabla 2- Tabla 1 del sprint de belatrix	17
Tabla 3- Tabla 2 del sprint de belatrix	18
Tabla 4- Tabla 3 del sprint de belatrix	19
Tabla 5- Tabla 4 del sprint de belatrix	20
Tabla 6- Tabla del reporte de los últimos 6 meses	27
Tabla7- Tabla Caso de prueba Automatizado del TFS de Belatrix	28

INTRODUCCIÓN

Para muchas organizaciones de desarrollo de software es de vital importancia reducir los costos de desarrollo mientras se mantiene alto la calidad del producto. Dado que las pruebas de QA comúnmente constituyen una parte significativa del tiempo de desarrollo, una forma de aumentar la eficiencia es encontrar más fallas tempranamente como se menciona en el artículo de Lars-Ola[1].

Belatrix es una empresa que se encarga de brindar servicios de software a diferentes clientes de Norteamérica y del mundo, por lo que constantemente sus clientes exigen un mejor servicios, por tal motivo Belatrix empezó con la reingeniería de los procesos de desarrollo de software. El proceso de QA pertenece a uno de los procesos de desarrollo de software y fue uno de los procesos que belatrix modificó añadiendo una actividad de Automatización con el que se podrá reducir el tiempo, siendo esto más atractivo para los clientes porque reducción de tiempo significa un costo más bajo y así continuar siendo atractivos para sus clientes a comparación de los clientes.

El presente Informe Profesional desarrolla la implementación del framework de automatización el cual representa la actividad de automatización del proceso de QA en la empresa Belatrix.

Debemos tener en cuenta que las actividades principales del proceso de QA que se realizan para todo desarrollo de software son:

1. Creación y casos de pruebas según los requerimientos.
2. Realizar el framework de automatización para automatizar las pruebas.
3. Validar las funcionalidades en base a los escenarios/casos de pruebas identificados.
4. Automatizar las funcionalidades de las pruebas de UI.
5. Reporte de la ejecución de los test cases automatizados.
6. Registro de las incidencias.

La actividad 1 consiste en identificar todas las casuísticas y flujos que se validará cuando el equipo de desarrollo despliegue la aplicación en el ambiente de pruebas.

La actividad 3 consiste en la ejecución de los casos de pruebas identificados en la actividad 1 y

si se encuentra algún caso de prueba o escenario que no completa el flujo completo, entonces se ejecuta la actividad 7, el cual es identificar y crear una incidencia.

La actividad 2 consiste en diseñar una framework de automatización, en el que sobre este se ejecuta la actividad 4 y 5 las cuales consisten en que se pueda automatizar los casos de pruebas identificados en la actividad 1.

la actividad 6 consiste en obtener un reporte de ejecución de todos los casos de pruebas automatizados en la actividad 4 y 5.

El siguiente informe se enfatizara las actividades siguientes actividades 2, 4, 5, 6 mediante los siguientes capítulos:

En el Capítulo I, desarrollare mi trayectoria profesional, la que refleja la experiencia adquirida y que permitió que pueda liderar de manera adecuada el proyecto de implementación del framework de automatización en el proceso de QA.

En el Capítulo II, desarrollare el contexto en el que se desarrolló la experiencia, describiendo a una Consultora, su visión y misión, organización, área cargo y funciones desempeñadas, así como la experiencia profesional realizada en la organización.

En el Capítulo III, explicaré el problema presentado, los objetivos y alcance, las etapas y metodología empleadas, los fundamentos utilizados e implementación.

En el Capítulo IV, expondré la reflexión crítica de la experiencia.

En el Capítulo V, desarrollaré las conclusiones y recomendaciones del presente Informe Profesional.

CAPITULO I - TRAYECTORIA PROFESIONAL

PRESENTACION PROFESIONAL

Soy un profesional **BACHILLER** en **INGENIERIA DE SISTEMAS**

Tengo amplia experiencia profesional en desarrollo y automatización de casos de prueba en diferentes rubros, mi experiencia es mayor a 5 años. Me gusta trabajar en equipo con proactividad y contribuir al éxito de los objetivos. Me considero un líder que está en busca de un mejora continua a nivel personal como en la organización.

Mi experiencia profesional se da con especial énfasis en los siguientes aspectos:

- Gestión del proceso de QA,
 - Automatización de Casos de pruebas de QA,
 - Liderar al Equipo de QA

Busco el Liderazgo de equipos de trabajo altamente eficientes, dirigiendo, desarrollando y motivando al equipo en la consecución de sus objetivos. Eficiencia, eficacia, habilidad de trabajar de equipo y proactividad permanente en el desempeño profesional.

FORMACION ACADEMICA

EDUCACION SUPERIOR: GRADOS ACADÉMICOS	
Grado Académico de Bachiller en Ingeniería de Sistemas Escuela Académico Profesional de Ingeniería de Sistemas – Facultad de Ingeniería de Sistemas e Informática – Universidad Nacional Mayor de San Marcos	2008 – 2013

EXPERIENCIA EMPRESARIAL

<p>Enero</p> <p>2018 -</p> <p>Actualidad</p>	<p>BELATRIX</p> <ul style="list-style-type: none">• Cargo: QA Leader/Senior QA• Rubros de la empresa: Consultoría en Tecnología de Información,<ul style="list-style-type: none">· El proyecto DentalHub estuvo basado en una arquitectura SOA donde sus operaciones tenían base en Miami, dentro del portal web realizaban todas las transacciones de compra de servicios/productos y se debía validar que las compras realizadas por GiftCard/CreditCard y otros medios de pago se realizarán correctamente.· Mis funciones fueron liderar al equipo de QA y Automatización en cada momento del pase a producción, asimismo me encargaba a la revisión de los casos de pruebas automatizados.• Principales actividades:<ul style="list-style-type: none">○ Mejora del Proceso de QA: Se implementó el servicio de QA el cual fue adoptado por toda la empresa. Se usan las herramientas: Selenium, TFS, Jira, C#.· Mis funciones fueron liderar al equipo de QA y Automatización en cada momento del pase a producción, asimismo me encargaba a la revisión de los casos de pruebas automatizados.· Las tareas diarias fueron Creación de pruebas automatizadas E2E, mantenimiento de casos de pruebas, ejecución de suites de pruebas en FF, IE, Chrome, safari a través de Team city, analizar el reporte y gestionar el Pase a Producción con el equipo de QA todo bajo la metodología ágil Scrum.○ Principales tecnologías: C#, JS, Protractor, Nunit, Appium, Team City, Git, Bitbucket, Selenium, xml, Json, BDD, TFS, Browser Stack, Restful
--	---

noviembre	GLOBANT
2016	<ul style="list-style-type: none"> • Cargo: QA Automation
-	
diciembre	<p style="text-align: center;">Proyecto: Carnival (Nov-2016 – Dec- 2017)</p>
2017	<ul style="list-style-type: none"> · El proyecto estuvo basado en una arquitectura SOA donde sus operaciones tenían base en Miami, dentro del portal web realizaban todas las transacciones de compra de servicios/productos y se debía validar que las compras realizadas por GiftCard/CreditCard y otros medios de pago se realizarán correctamente.
	<ul style="list-style-type: none"> · Mis funciones fueron liderar al equipo de QA y Automatización en cada momento del pase a producción, asimismo me encargaba a la revisión de los casos de pruebas automatizados. <p>Mis tareas diarias fue analizar los requerimientos con personas en Usa, India, China, preparar un documento de ese análisis donde se indicaba los escenarios que se iban a validar en esa iteración, después se realizaba la creación de casos de pruebas, automatización de casos a nivel de FrontEnd(usando selenium) y Backend(usando pruebas unitarias). Asimismo se realizó todas las funciones del proceso de QA como La ejecución de la regresión, pruebas exploratorias, pruebas de humo.</p>
	<ul style="list-style-type: none"> · Las tareas diarias fueron Creación de pruebas automatizadas E2E, mantenimiento de casos de pruebas, ejecución de suites de pruebas en FF, IE, Chrome, safari a través de Team city, analizar el reporte y gestionar el Pase a Producción con el equipo de QA todo bajo la metodología agil Scrum.
	<p style="text-align: center;">Principales tecnologías: C#, JS, Protractor, Nunit, Appium, Team City, Git, Bitbucket, Selenium, xml, Json, BDD , TFS, Browser Stack, Restful.</p>

**Noviembre
2016**
-
**November
2017**

- Cargo: QA Automation

Proyecto: Carnival (Nov-2016 – Dec- 2017)

El Proyecto está basado en las prácticas de ITIL basado en un aplicativo SOA para el manejo de sus sistemas con sede en USA.

Mis tareas diarias fue analizar los requerimientos con personas en Usa, India, China, preparar un documento de ese análisis donde se indicaba los escenarios que se iban a validar en esa iteración, después se realizaba la creación de casos de pruebas, automatización de casos a nivel de FrontEnd(usando selenium) y Backend(usando pruebas unitarias). Asimismo se realizó todas las funciones del proceso de QA como La ejecución de la regresión, pruebas exploratorias, pruebas de humo.

Estuve a cargo del pase a producción asimismo en el proyecto lideraba a 2 QA semisenior en la parte técnica y de gestión proponiendo feedbacks entre integrantes del equipo independiente del proceso de Scrum.

Las funciones realizadas fueron analizar los requerimientos, creación de casos de pruebas, automatizar los casos y Liderar el pase a producción donde se definía los flujos que se ejecutarán en la ventana de mantenimiento así como enviar reportes del progreso del Sprint de todos los equipos.

validación del performance de los servicios RESTful usando Jmeter.

Las tecnologías usadas fueron: java E2E, SPRING, Cucumber, Json, MongoDB, SQL server, Git, Teamcity, Jenkins, Jira, Confluence y se usó la metodologías Devops y Scrum, Soapui, Lisa Virtualization, OWASP ZAP,Json.

Incorpore un equipo que proponía mejoras para el proceso de devops.

<p>noviembre</p>	<p>VERIZON</p>
<p>2015</p>	<ul style="list-style-type: none"> • Cargo: QA Automation III
<p>-</p>	
<p>diciembre</p>	<p>Proyecto: VERIZON HOST(Nov-2016 – Dec- 2017)</p>
<p>2016</p>	<ul style="list-style-type: none"> · El proyecto estuvo basado en una arquitectura SOA donde sus operaciones tenían base en Miami, dentro del portal web realizaban todas las transacciones de compra de servicios/productos y se debía validar que las compras realizadas por GiftCard/CreditCard y otros medios de pago se realizarán correctamente.
	<p>Mis tareas diarias fue analizar los requerimientos con personas en Usa, India, China, preparar un documento de ese análisis donde se indicaba los escenarios que se iban a validar en esa iteración, después se realizaba la creación de casos de pruebas, automatización de casos a nivel de FrontEnd(usando selenium) y Backend(usando pruebas unitarias). Asimismo se realizó todas las funciones del proceso de QA como La ejecución de la regresión, pruebas exploratorias, pruebas de humo.</p>
	<p>Estuve a cargo del pase a producción asimismo en el proyecto lideraba a 2 QA semisenior en la parte técnica y de gestión proponiendo feedbacks entre integrantes del equipo independiente del proceso de Scrum.</p>
	<ul style="list-style-type: none"> · Mis funciones fueron liderar al equipo de QA y Automatización en cada momento del pase a producción, asimismo me encargaba a la revisión de los casos de pruebas automatizados.
	<ul style="list-style-type: none"> · Las tareas diarias fueron Creación de pruebas automatizadas E2E, mantenimiento de casos de pruebas, ejecución de suites de pruebas en FF, IE, Chrome, safari a través de Team city, analizar el reporte y gestionar el Pase a Producción con el equipo de QA todo bajo la metodología agil Scrum.
	<p>Principales tecnologías: C#, JS, Protractor, Nunit, Appium, Team City, Git, Bitbucket, Selenium, xml, Json, BDD , TFS, Browser Stack, Restful.</p>

Marzo

2014

-

Octubre

2015

- Cargo: QA Automation

Proyecto: Ariba Sap (Marzo 2014 – Octubre 2015)

Estuve trabajando en un Producto de la empresa Alemana llamada Ariba network, el cual era una aplicación de tipo "Procurement Process", es decir tenían un catálogo de Productos/Servicios que ofrecían y consumían otras empresas.

El proyecto estaba basado en la fusión de los productos de las empresas que habían comprado SAP.

Mis tareas diarias fue analizar los requerimientos con personas en Usa, India, China, preparar un documento de ese análisis donde se indicaba los escenarios que se iban a validar en esa iteración, después se realizaba la creación de casos de pruebas, automatización de casos a nivel de FrontEnd(usando selenium) y Backend(usando pruebas unitarias). Asimismo se realizó todas las funciones del proceso de QA como La ejecución de la regresión, pruebas exploratorias, pruebas de humo.

Estuve a cargo del pase a producción asimismo en el proyecto lideraba a 2 QA semisenior en la parte técnica y de gestión proponiendo feedbacks entre integrantes del equipo independiente del proceso de Scrum.

Las tecnologías Utilizadas fueron:JUnit, Java 2EE ,Spring Tool Suit, Maven, Spring, TestLink, Jira, Maven, Selenium WebDriver, WebObjects, Jmeter, Servicios RESTFul, Json

- Implementación de los Sistemas de Control de Parque Móvil, de Peticiones y de Red para el Área de Parque Móvil, Gerencia de Sistemas y Gerencia de Red de Telefónica del Perú.
- Desarrollo y Análisis de nuevos requerimientos bajo la arquitectura monolítica. Utilizando la plataforma Genexus.
- Desarrollo de Sistema de Digitalización en Visual .NET 2010.

	<ul style="list-style-type: none"> • Desarrollo de Sistema de envío masivo de SMS de texto. • Desarrollo de Sistema de inventario en lenguaje GENEXUS. • Herramientas:
--	---

<p>Enero 2013</p> <p>-</p> <p>Marzo 2014</p>	<p>Avantica.</p> <ul style="list-style-type: none"> • Cargo: Programador, en el Área de Recursos de Sistemas de AVANTICA • Logros: <ul style="list-style-type: none"> • Implementación de los Sistemas de Control de Parque Móvil, de Peticiones y de Red para el Área de Parque Móvil, Gerencia de Sistemas y Gerencia de Red de Telefónica del Perú. • Desarrollo y Análisis de nuevos requerimientos bajo la arquitectura monolítica. Utilizando la plataforma Genexus. • Implementación de los Sistemas de Control de Parque Móvil, de Peticiones y de Red para el Área de Parque Móvil, Gerencia de Sistemas y Gerencia de Red de Telefónica del Perú. • Desarrollo y Análisis de nuevos requerimientos bajo la arquitectura monolítica. Utilizando la plataforma Genexus. • Desarrollo de Sistema de Digitalización en Visual .NET 2010. • Desarrollo de Sistema de envío masivo de SMS de texto.
--	--

	<ul style="list-style-type: none"> • Desarrollo de Sistema de inventario en lenguaje GENEXUS. • Desarrollo de Sistema de Digitalización en Visual .NET 2010. • Desarrollo de Sistema de envió masivo de SMS de texto. • Desarrollo de Sistema de inventario en lenguaje GENEXUS. • Herramientas: <p>Lenguajes: Java, Genexus, Visual .NET 2010.</p>
<p>Marzo 2012</p> <p>-</p> <p>Enero 2013</p>	<p>Bonus.</p> <ul style="list-style-type: none"> • Cargo: Programador, en el Área de Recursos de Sistemas de SISTEMAS BONUS. • Logros: <ul style="list-style-type: none"> • Implementación de los Sistemas de Control de Parque Móvil, de Peticiones y de Red para el Área de Parque Móvil, Gerencia de Sistemas y Gerencia de Red de Telefónica del Perú. • Desarrollo y Análisis de nuevos requerimientos bajo la arquitectura monolítica. Utilizando la plataforma Genexus. • Implementación de los Sistemas de Control de Parque Móvil, de Peticiones y de Red para el Área de Parque Móvil, Gerencia de Sistemas y Gerencia de Red de Telefónica del Perú. • Desarrollo y Análisis de nuevos requerimientos bajo la arquitectura monolítica. Utilizando la plataforma Genexus. • Desarrollo de Sistema de Digitalización en Visual .NET 2010. • Desarrollo de Sistema de envió masivo de SMS de texto. • Desarrollo de Sistema de inventario en lenguaje GENEXUS. • Desarrollo de Sistema de Digitalización en Visual .NET 2010.

-
- Desarrollo de Sistema de envío masivo de SMS de texto.
 - Desarrollo de Sistema de inventario en lenguaje GENEXUS.
 - Herramientas:

Lenguajes: Java, Genexus, Visual .NET 2010.

CAPITULO II - CONTEXTO EN EL QUE SE DESAROLLO LA EXPERIENCIA

2.1 EMPRESA - ACTIVIDAD QUE REALIZA

Belatrix es una empresa que se dedica a brindar servicios de desarrollo de software. servicios de Gestión de la calidad a distintas empresas.

2.2 VISION

Brindar soluciones innovadoras y soporte especializado, alineados a la necesidad de los clientes ayudándolos en el crecimiento y posicionamiento en el mercado

2.3 MISION

Ofrecer asesoría, soporte y servicios con calidad y mejora continua, elevando la satisfacción de los clientes.

2.4 VALORES

Los valores de Belatrix son:

- **Trabajo en Equipo:** Capacidad de integración y colaboración entre todos los miembros de la organización para potenciar las habilidades hacia un objetivo común Actitud positiva orientada al mejoramiento permanente para alcanzar los objetivos y metas planteados. Alto sentido de colaboración y atención proactiva a clientes internos y externos con cordialidad, eficiencia y rapidez.
- **Compromiso:** Cumplir con las obligaciones en general y lograr los objetivos, metas, procesos y compromisos en los plazos establecidos asimismo brindar respuestas oportunas a nuestros clientes.
- **Pasión por el Servicio:** Servir a nuestros clientes con un servicio superior superando sus expectativas.
- **Innovación:** Crear e implementar ideas para el crecimiento de los negocios y el aumento de la productividad mediante mejoras de productos, servicios y

procesos

2.5 ORGANIZACIÓN DE LA EMPRESA

La empresa belatrix cuenta con un ejecutivo de cuenta o director de cuenta, el cual es el encargado de coordinar directamente con el cliente, también se cuenta con el responsable técnico por el lado de desarrollo y de QA de Belatrix, quienes son los referentes técnicos de buenas prácticas del proyectos. Debajo de ellos se encuentran los desarrolladores y analistas de QA. Como se puede apreciar en la *Figura 1*. Ejecutivo de cuenta, Responsable técnico, y responsable de aseguramiento de calidad.

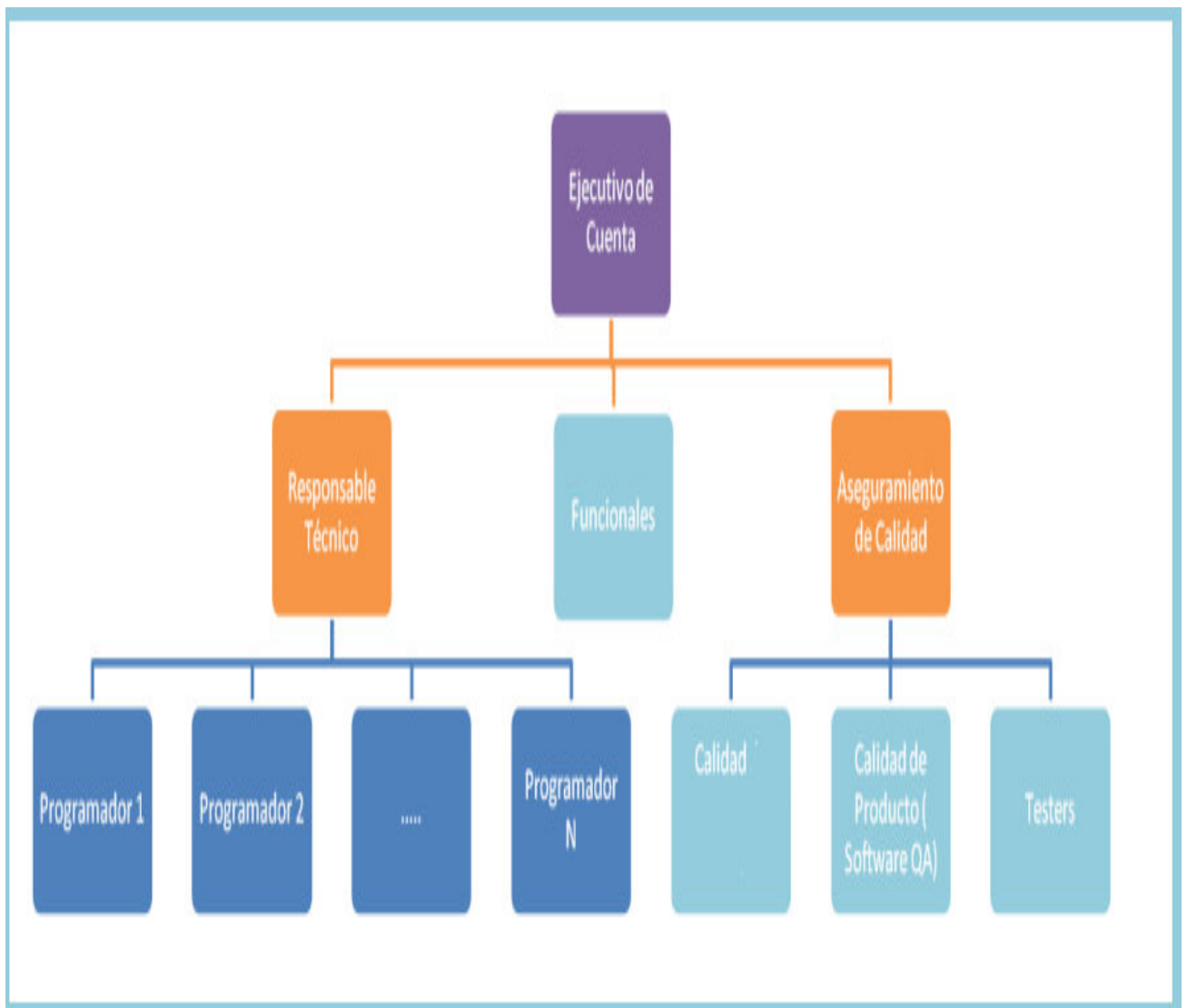


Figura 1-Organigrama de Consultora Belatrix

2.6 ÁREA, CARGO Y FUNCIONES DESEMPEÑADAS

Desempeñe el cargo de **Líder de QA** para el proyecto de Mejora del proceso de QA del Estudio de Quality Engineering de la empresa Belatrix. El estudio de QA Engineering es uno de los estudios que se encarga de la calidad del producto.

Mis funciones son las siguientes:

- a) Redefinir los procesos de QA, analizar e identificar los tiempos que toma las actividades del proceso de QA.
- b) Seguimiento de las actividades y tareas que realizaban los miembros de QA en los proyectos.
- c) Realizar informes semanales de los proyectos certificados y el porcentaje de avance.
- d) Realizar la estrategia de QA Para los proyectos.
- e) Gestionar los planes de pruebas.
- f) Construir el framework de automatización para la automatización de los casos de pruebas.

2.7 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN

Durante mi experiencia profesional en Belatrix, como Líder de QA mis funciones fueron implementar el framework de automatización, el cual permite la automatización de los casos de pruebas tanto para la parte de pruebas de servicio como pruebas E2E.

2.7 Conclusión

Durante mi experiencia profesional en Belatrix, asumi la responsabilidad de poder mejorar el proceso de QA así que necesitaba identificar el problema que se tiene durante la ejecución de las actividades del proceso de QA. El problema identificado fue el Tiempo y la ejecución repetitiva de los casos de pruebas. En la siguiente sección mencionare mas a detalle

CAPÍTULO III - ACTIVIDADES DESARROLLADAS

En el capítulo 3 se desarrollara la definición del problema y el planteamiento de la solución, el cual fue identificado como una actividad dentro del proceso de QA, dicha actividad es la ejecución de los casos de pruebas (Regresión).

3.1 DEFINICIÓN DEL PROBLEMA

La metodología Scrum en Belatrix se consta de varias actividades. Como se puede observar, en la *figura 2* encontramos estas actividades, las cuales se encuentran enumerados en orden de ejecución: primero se crean nuevas historias de usuario (nuevos requerimientos), se realiza la estimación, luego se inicia el desarrollo (programación en código) y se finaliza con la ejecución del proceso de QA, el cual busca garantizar la calidad del desarrollo.

Se debe tener en cuenta que la actividad 1 y 2 se ejecutan en paralelo asimismo como las actividades 3 y 4.

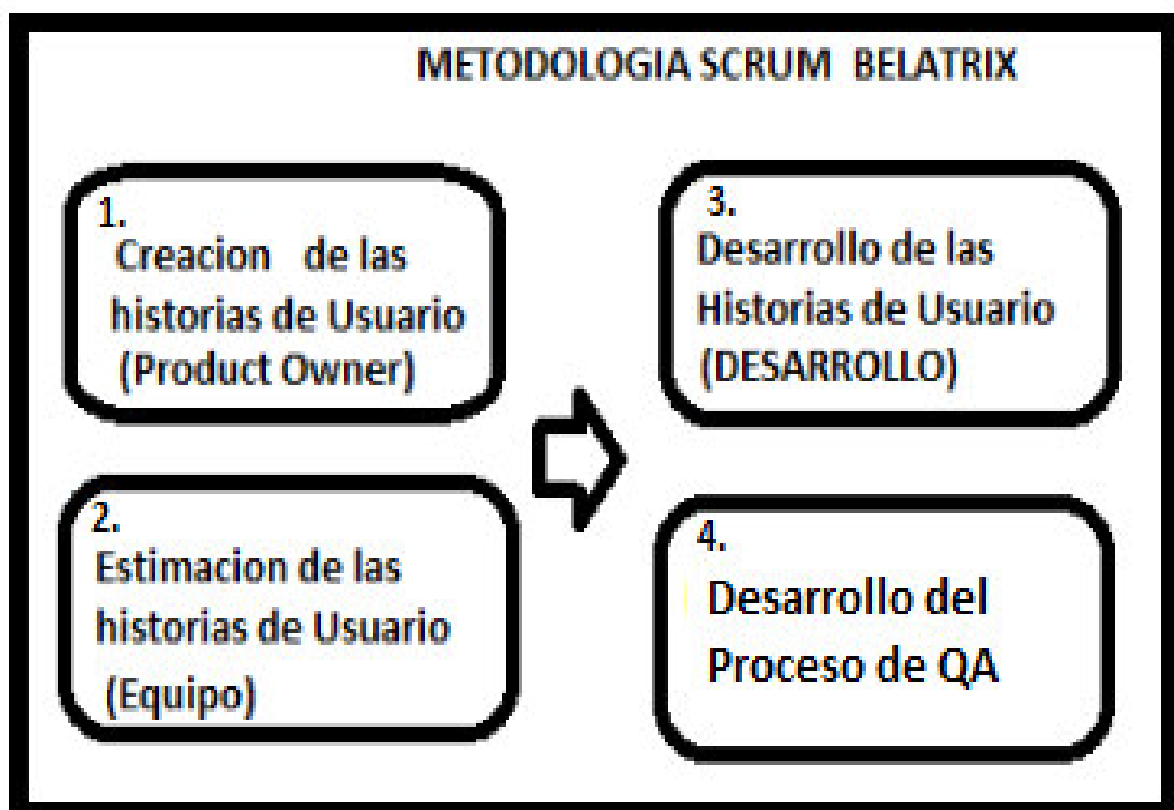


Figura 2-Metodología Scrum belatrix de la consultora Belatrix

A continuación se detalla en qué consiste cada actividad de la metodología scrum:

1. Creación de las historias de Usuario (Product Owner)

Su objetivo es adecuar el modelo a emplear para la solución del Requerimiento, expresado en términos de objetos y relaciones entre ellos.

El desarrollo de las historias de Usuario es una abstracción resumida y precisa de lo que debe de hacer el Requerimiento deseado y no de la forma en que se hará. Los elementos del modelo deben ser conceptos del dominio de aplicación y no conceptos informáticos tales como estructuras de datos.

Los pasos a seguir son:

De acuerdo a la prioridad de la tarea o Proyecto, el Jefe de Proyecto asigna al Product Owner y coordinan el tiempo para el análisis de los requerimientos y presentación de las historias de usuario (documento solución al Requerimiento).

Para una Historia de Usuario: los pasos a seguir del Product Owner son:

- Realiza el análisis según la solicitud de servicio, plantea la solución y completa los documentos: Historia de Usuario.
- Presenta solución al Usuario y la reformula hasta obtener su aprobación.
- Presenta solución al A I Equipo Scrum.

2. Estimación los Historias de usuario (Equipo)

Su objetivo es identificar cuánto esfuerzo tomará desarrollar y completar la historia de Usuario por el equipo (se tiene que evaluar "Definition of Done")

Para ello se tomará los números de Fibonacci para puntuar la historia

Los pasos a seguir son:

Todo el equipo indica cuánto tomará completar la historia de Usuario desde su perspectiva de su Rol (Desarrollador, QA) en el que se justificara la puntuación Dada; los números estén dentro del rango: 1, 2, 3, 5, 8, 13, 21.

3. Desarrollo de las historias de Usuario (DESARROLLO)

Los desarrolladores empiezan con la creación de código para los servicios y la parte de la interfaz de usuario así como la creación de las pruebas unitarias.

4. Pruebas de las de las historias de Usuario (QA)

El equipo de QA se encarga de la creación de los casos de pruebas, ejecución de las pruebas exploratorias, ejecución de los casos de pruebas, registro de incidencias.

Según la *figura 3*, el procedimiento consiste primero en Realizar la creación de los casos de pruebas, mientras el equipo de desarrollo está realizando la programación, cuando el equipo desarrollo termina sus actividades, el equipo de QA empieza con la ejecución de las pruebas exploratorias y la ejecución de los casos de pruebas.

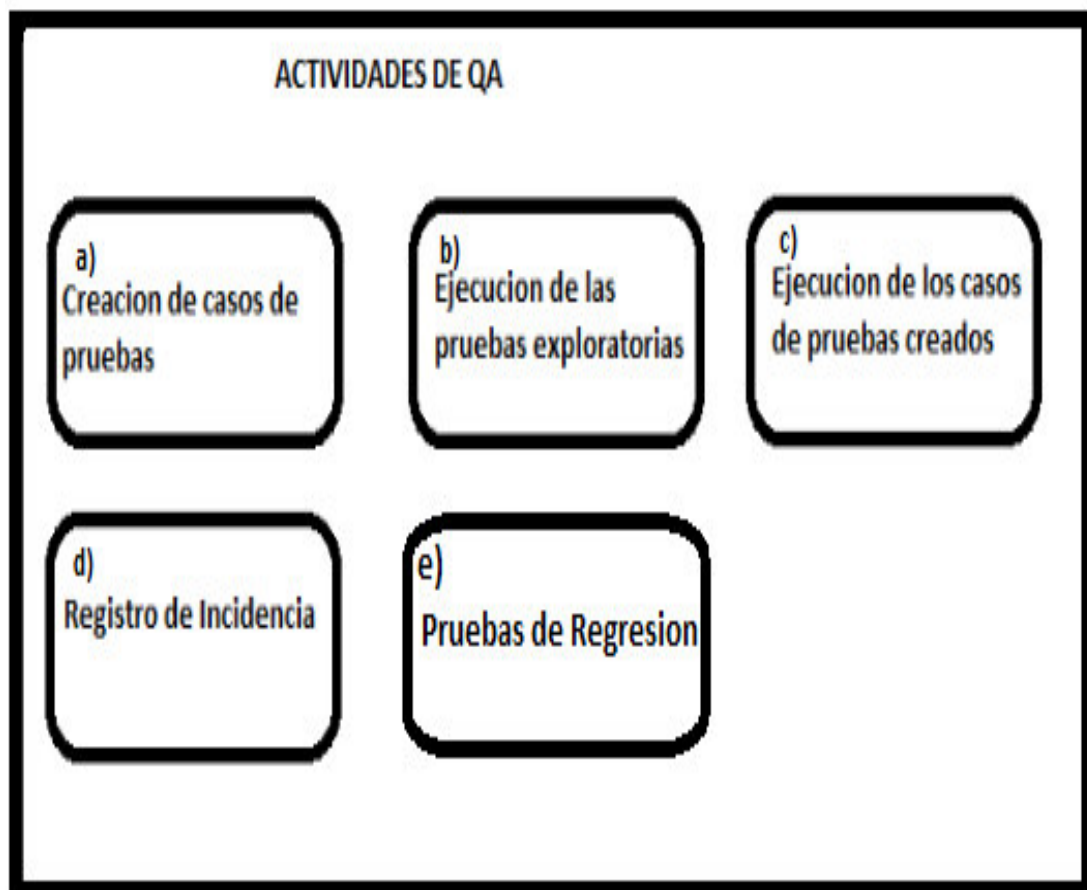


Figura 3-Actividades de QA de la consultora Belatrix

a. Creación de los casos de pruebas consiste en analizar las historias de usuarios y crear nuevos casos de pruebas en Test Link (**Véase en el anexo 1**), los cuales serán ejecutados cuando el desarrollo termine. Se toma en cuenta como caso de prueba a un conjunto de pasos redactados en un documento con una finalidad de validación, en este caso será el test link el repositorio de todos los casos de pruebas creados.

b. Ejecución de las pruebas exploratorias consiste en validar la aplicación de manera ligera y superficial, sin ser meticulosos en las validaciones para identificar rápidamente incidencias, generalmente esta validación demora poco tiempo en comparación de la ejecución de los casos de pruebas, sin embargo la idea de las pruebas exploratorias es detectar rápidamente incidencias.

c. Ejecución de los casos de pruebas consiste en seguir los pasos que indican los casos en pruebas en la aplicación desarrollada, se realiza cuando el desarrollo de las historias de usuario ya está terminado.

d. Registro de la incidencia consiste en que si cuando se están ejecutando los casos de pruebas en la aplicación, si se encuentra alguna incidencia, se registrara en Jira, para que el equipo de desarrollo lo resuelva.

e. Pruebas de Regresión (QA)

Las pruebas de regresión es crucial para la fase de desarrollo, eliminando los errores en producción y el desarrollo del producto , como se indica en la sección de background de AZAD [3].

La regresión de casos de pruebas se realiza después de que todos los casos de pruebas del sprint se hayan ejecutado satisfactoriamente, asimismo las pruebas exploratorias y que todas las incidencias se hayan cerrado, para así dar paso a la regresión el cual consiste en ejecutar todos los casos de pruebas creados anteriormente para garantizar que las funcionalidades anteriores sigan funcionando como se espera.

En las figuras 3.1 ,3.2, 3.3, 3.4 se observa la ejecución de 900 casos de pruebas distribuidas en 5 testers (analistas de QA), también se puede visualizar la cantidad de casos de pruebas asignados.

Pruebas de Regresion Sprint 1 # de Personas	900 casos de pruebas
Tester 1(160 casos)	20 horas
Tester 2(170 casos)	20 horas
Tester 3(170 casos)	20 horas
Tester 4(200 casos)	24 horas
Tester 5(200 casos)	24 horas

Tabla 1 Pruebas de regresion del Sprint 1 de la consultora Belatrix

Pruebas de Regresion Sprint 2 # de Personas	932 casos de pruebas
Tester 1(192 casos)	24 horas
Tester 2(170 casos)	20 horas
Tester 3(170 casos)	20 horas
Tester 4(200 casos)	24 horas
Tester 5(200 casos)	24 horas

Tabla 2 Pruebas de regresión del Sprint 2 de la consultora Belatrix

Pruebas de Regresion Sprint 3 # de Personas	962 casos de pruebas
Tester 1(192 casos)	24 horas
Tester 2(200 casos)	24 horas
Tester 3(170 casos)	20 horas
Tester 4(200 casos)	24 horas
Tester 5(200 casos)	24 horas

Tabla 3 Pruebas de regresión del Sprint 3 de la consultora Belatrix

Pruebas de Regresion Sprint 4 # de Personas	1000 casos de pruebas
Tester 1(200 casos)	24 horas
Tester 2(200 casos)	24 horas
Tester 3(200 casos)	24 horas
Tester 4(200 casos)	24 horas
Tester 5(200 casos)	24 horas

Tabla 4 Pruebas de regresión del Sprint 4 de la consultora Belatrix

Las pruebas de regresión tomaban aproximadamente 3 días(24 horas), ejecutadas por 5 personas.

El cuello de botella que encontramos es la cantidad de días y personas que nos toma la regresión después de finalizar cada sprint, como se puede observar también es que mientras más sprints se desarrollan la cantidad de casos de pruebas aumenta y esto requerirá mayor cantidad de días y más personas.

El principal problema es la cantidad de tiempo que demora actualmente la regresión de los casos de pruebas manuales después de cada desarrollo.

3.2 SOLUCIÓN

3.2.1 OBJETIVO GENERAL

Implementar un framework de automatización para el proceso de QA en un proyecto de desarrollo.

3.2.1 OBJETIVO ESPECIFICOS

- Reducir el tiempo que toma la regresión de los casos de pruebas.
- Evitar la intervención Humana en la ejecución de los casos de pruebas.
- Añadir una actividad nueva denominada Automatización de casos de pruebas dentro del proceso de QA
- Crear el framework de automatización.

Como se puede apreciar en la *figura 4*, se añadió la actividad Automatización de casos de pruebas, el cual se ejecutaría después de que se realice los pasos A , B, C y D y luego de culminar la actividad E (Automatización de casos de pruebas) continúa la regresión la cual es la actividad F.

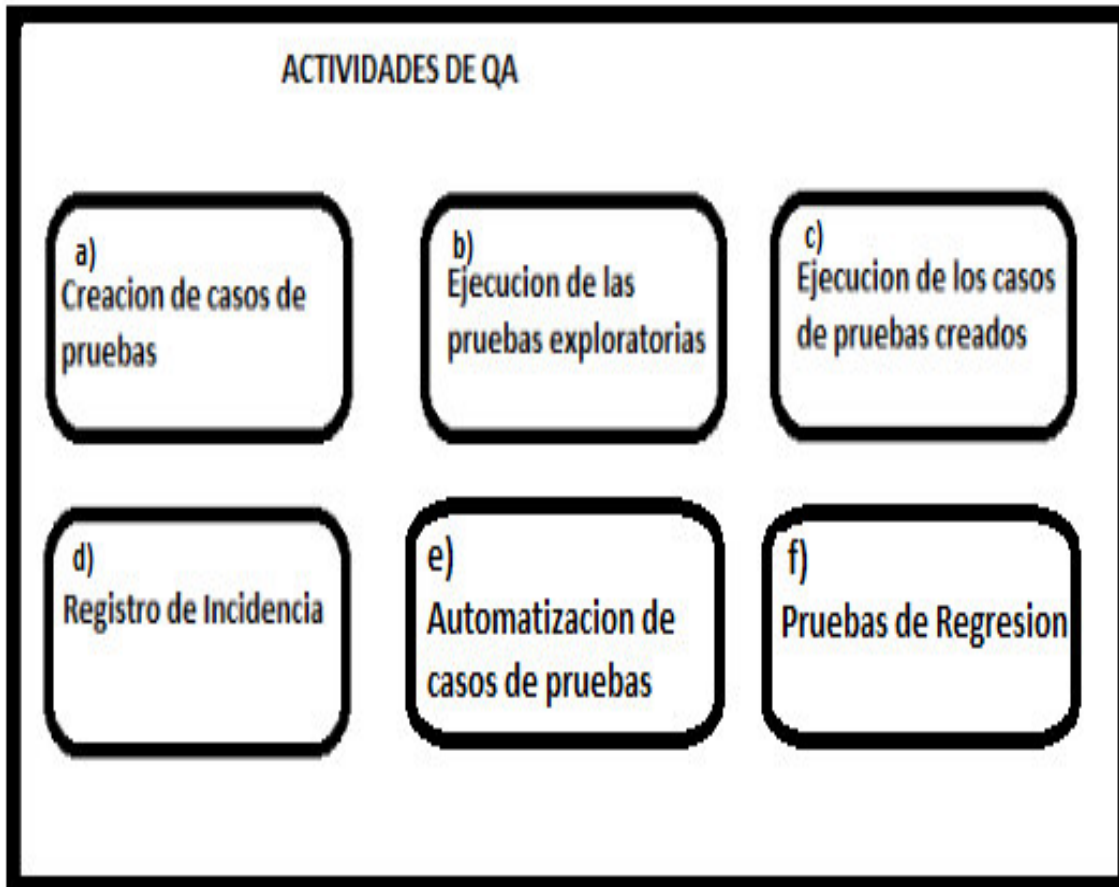


Figura 4 - Actividades de QA Incluido Automatización de Casos de pruebas

3.2.2 ALCANCE

El presente documento brinda información la configuración del framework de automatización, Casos de pruebas Automatizados y un Manual de Usuario de cómo crear un test Automatizado Los casos de pruebas previamente deben estar definidos.

3.2.3 ETAPAS Y METODOLOGÍA

Para empezar a realizar el Framework de automatización se necesita tener en cuenta estas etapas.

1. Diseñar la arquitectura del framework de automatización
2. Elección de la Tecnología y de las librerías.
 - Creación de las carpetas y clases del framework de automatización.

- Creación de la Clase Test el cual aloja a todos los test que se automatizaran

1. Diseño de la arquitectura.

La arquitectura se puede visualizar en la *figura 4.1* , básicamente contiene folders donde estarán contenidos las clases organizados de una manera óptima, la carpeta COMMON contiene la clase Utilitario el cual en su mayoría posee métodos estáticos los cuales serán consumidos en todo el proyecto y clases de configuración, la carpeta DRIVERS, contiene las clases que configuran ciertos navegadores que se aperturan cuando el test se ejecuta, cada clase en la carpeta driver puede ser de tipo Chrome, ie, firefox, safari, opera, etc. la carpeta PÁGINAS contiene las clases que están relacionada a cada página de la aplicación, es decir la pagina Login posee su clase correspondiente, la página de bienvenida similarmente y es ahí donde se aloja los elementos Web. la Carpeta TESTS contiene la clase de los casos de pruebas divididos por funcionalidades, es decir todos los casos de pruebas de login irán en la clase Login Test.

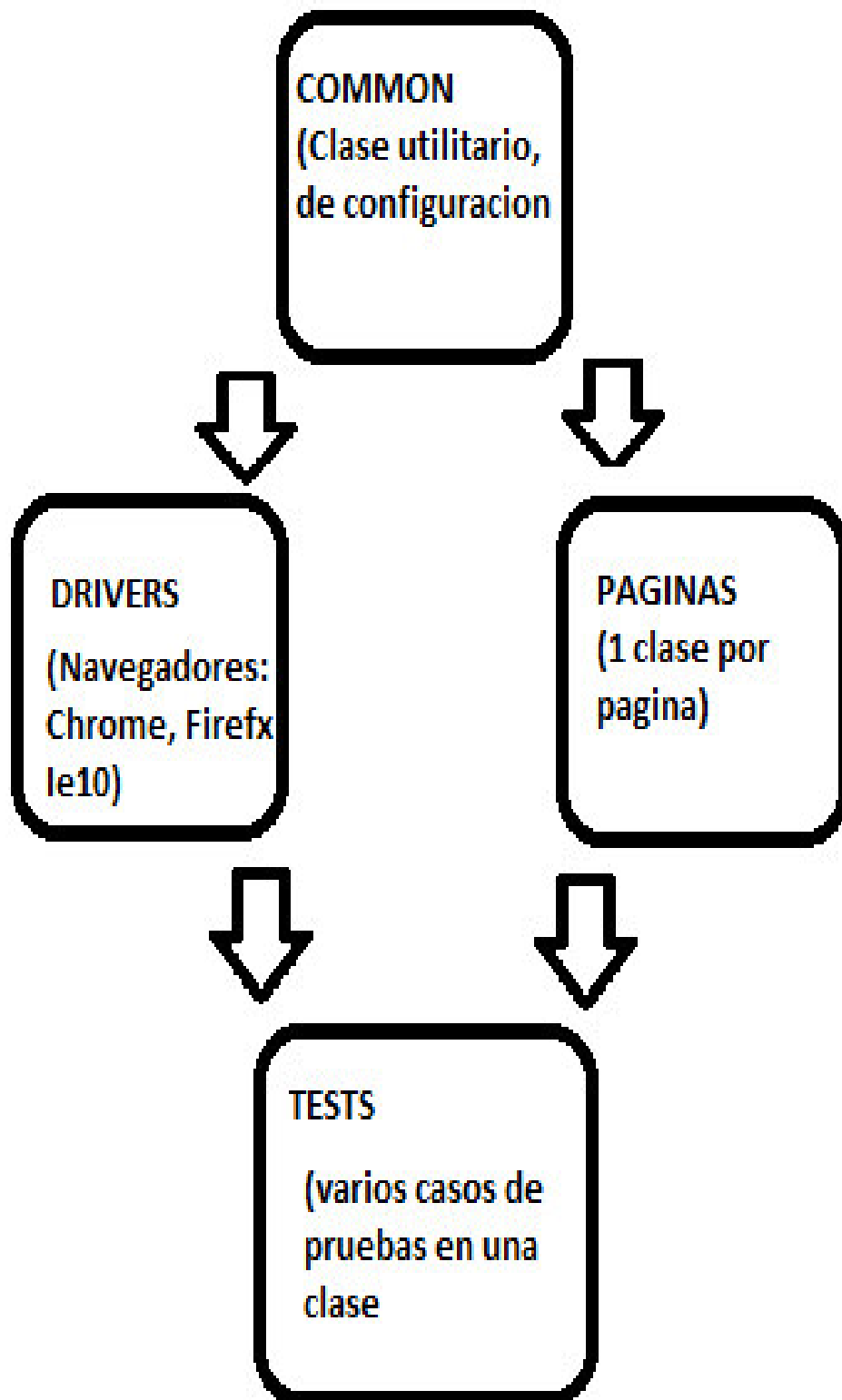


Figura 4.1 - Arquitectura del framework de Automatización de Belatrix

2. Elección de la Tecnología y de las librerías.

-Para la poder construir el framework de automatización desde cero debemos identificar las herramientas tecnológicas, teniendo en cuenta que actualmente muchas de ellas tienen mayor soporte técnico de otras, para esto se identificó las buenas prácticas que usan muchas compañías en el ámbito de la automatización de pruebas. Para esto se escogió la librería Selenium (véase en el anexo 2) debido a que se integra fácilmente con TestNG, el cual me permite generar reportes y porque posee paralelismo en comparación a otras librerías. Selenium permite la conexión con los diferentes browsers como Chrome, Firefox, Internet Explorer entre otros.

Se escogió el lenguaje C# el cual requiere que se instale el Visual Studio Community (véase anexo 3) porque está orientado a POO, porque es Open Source y porque posee librerías como Linq, Bogus y fluent Assertion los cuales son fácilmente añadidos al framework de automatización para realizar validaciones, crear data y realizar conexión a base de datos. Como se puede visualizar en la *Figura 5*, se tiene una libre descarga para el uso personal sin pago alguno.

Realizar la instalación de las librerías

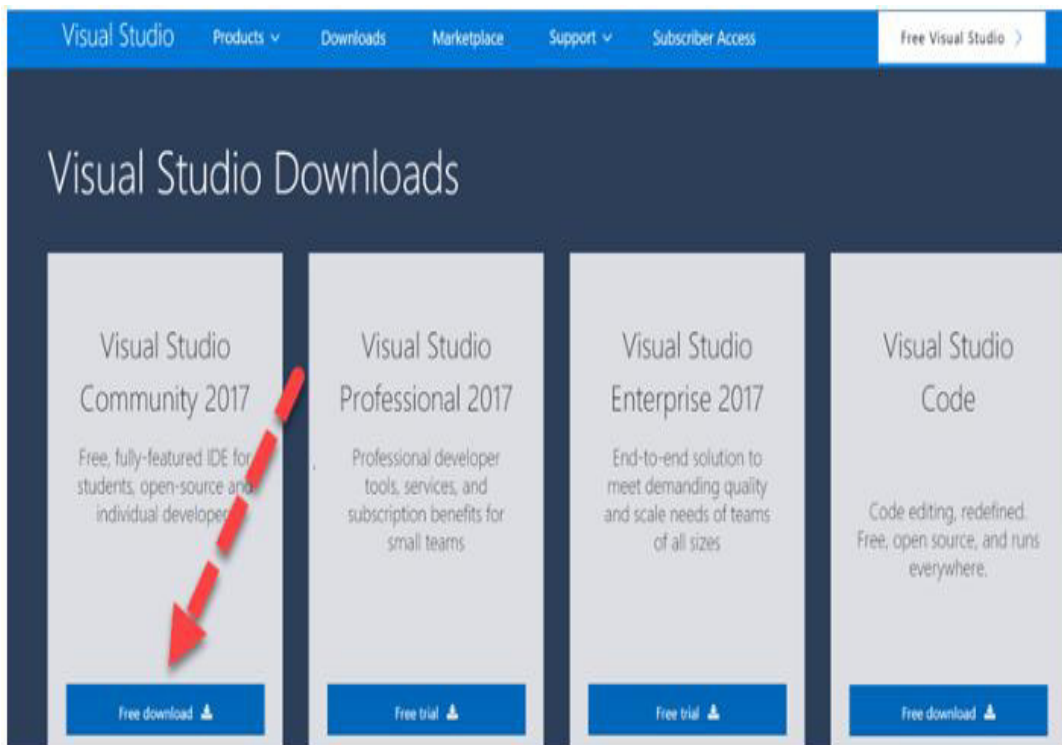


Figura 5 - Instalación del Visual Studio Community del Manual de Instalación de Belatrix

En la *figura 6*. se muestra el visual studio listo para añadir los nugets(las librerías que se utilizarán en la automatización)

Se crea un proyecto nuevo y se instala la herramienta Selenium añadiendo un Nuget

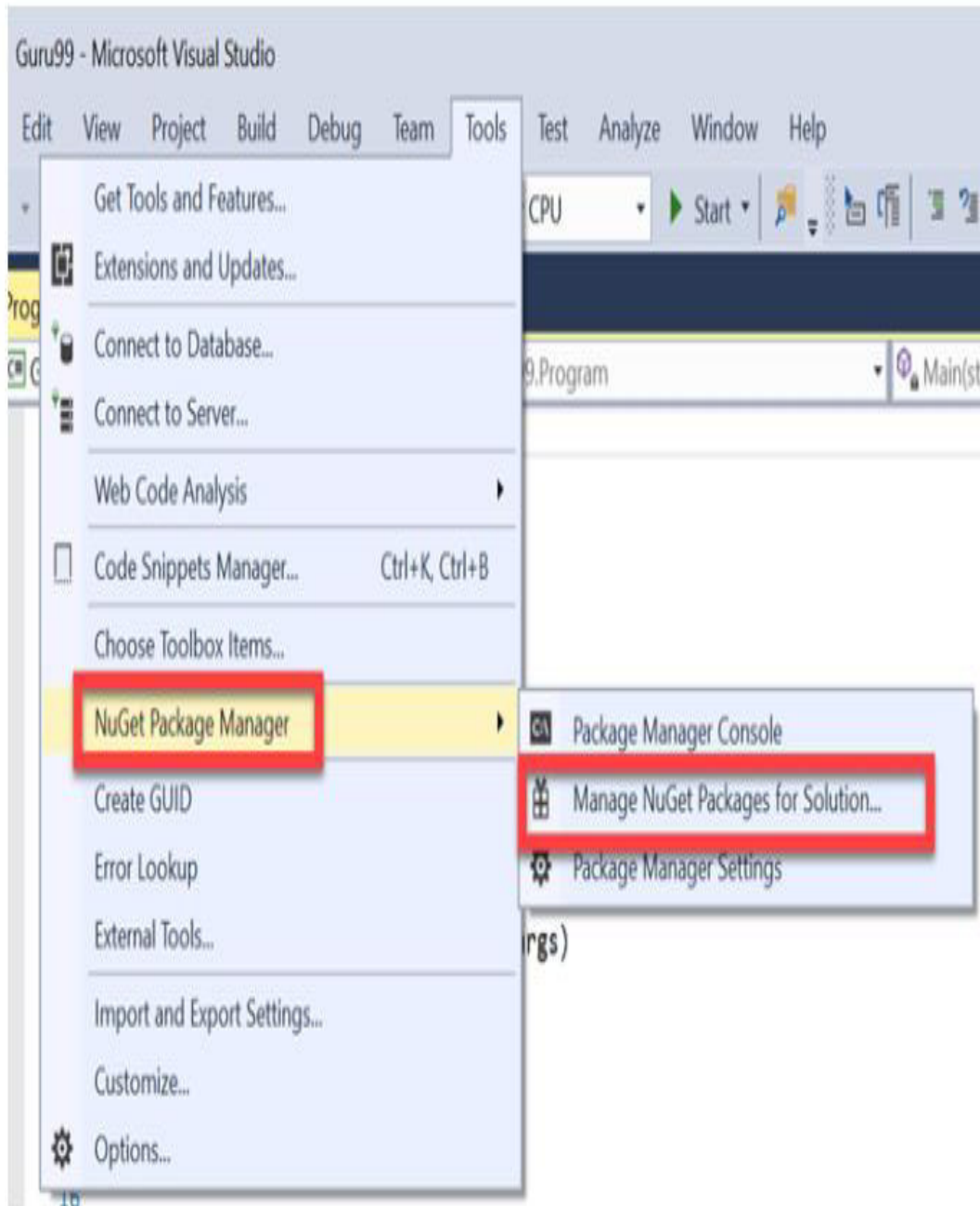


Figura 6-Añadir un Nuget al visual Studio del Manual de Instalación de Belatrix

Se realiza una búsqueda de la librería Selenium Web Driver, el cual nos permite interactuar con Los diferentes browsers (Chrome, Safari, Internet Explorer, Firefox). en la *figura 7* se visualiza la búsqueda realizada con éxito y se encontro la libreria Selenium.

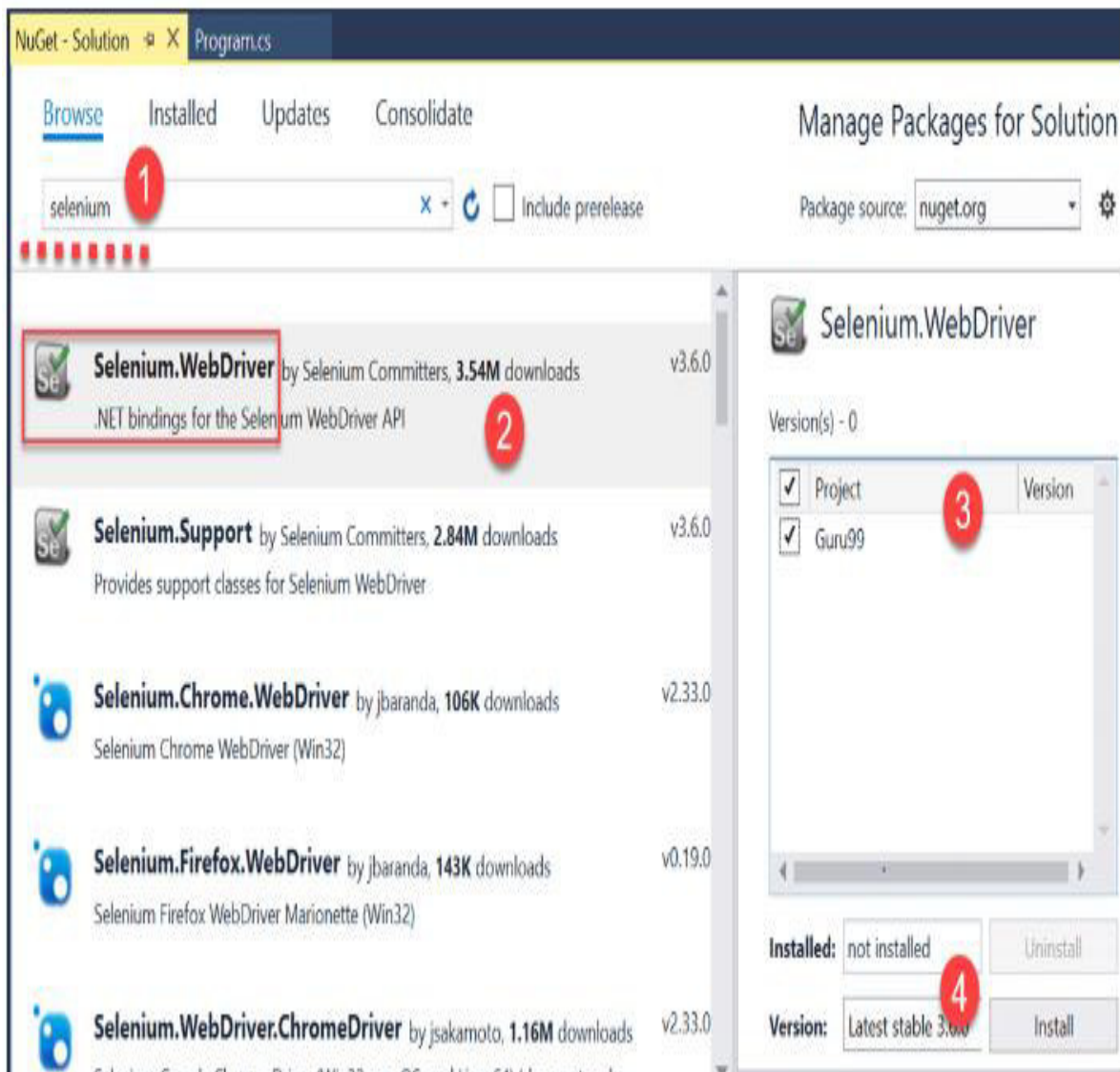


Figura 7-Añadir Selenium Nuget al visual Studio del Manual de Instalacion de Belatrix

Se instala otra librería llamada Nunit el cual nos brindara ciertos métodos para validar los resultados esperados de nuestro test, como se puede visualizar en la *figura 8* obtenemos la búsqueda del Nunit, una libreria util para validar nuestros escenarios

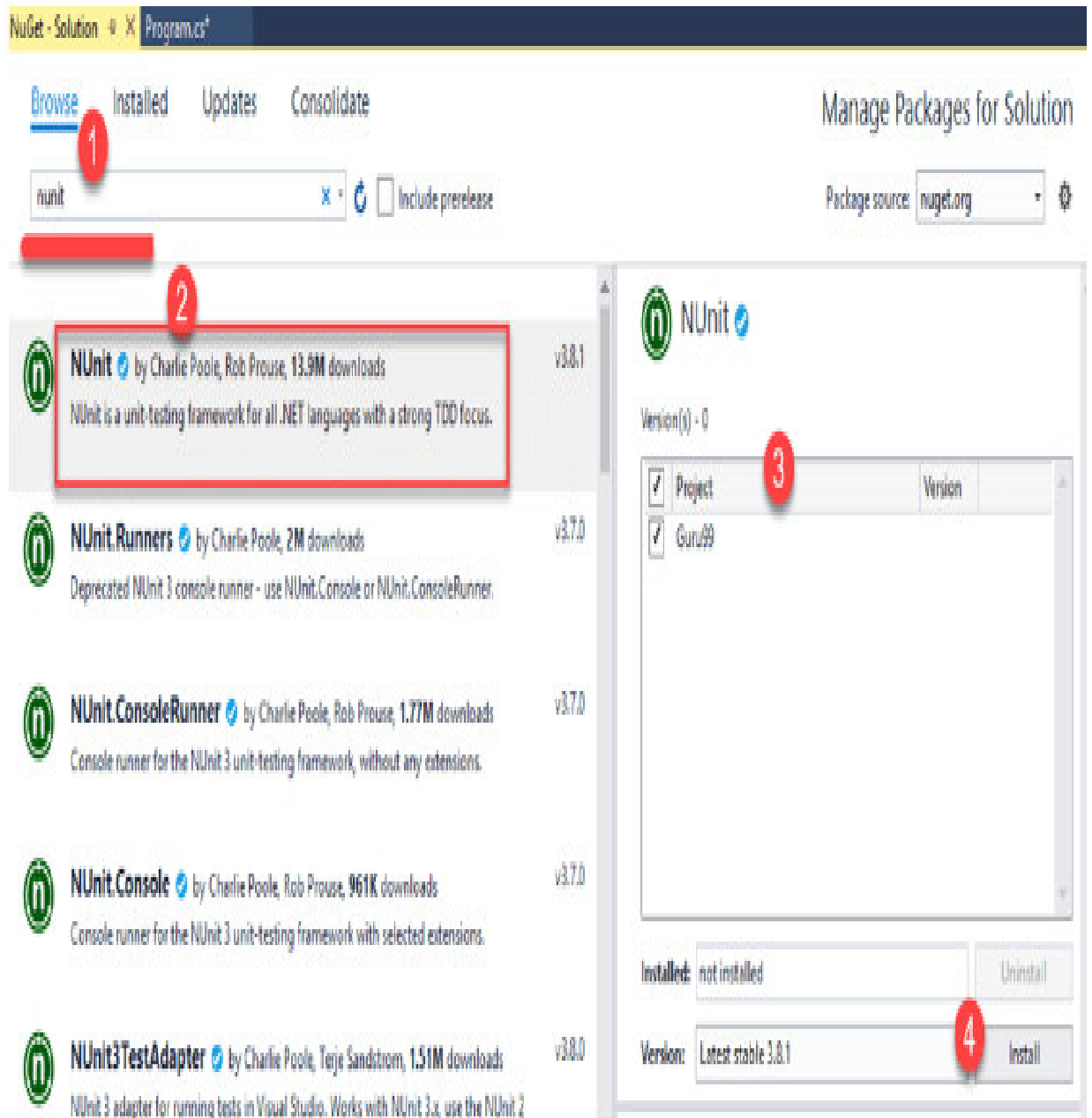


Figura 8 -Añadir Nunit Nuget al visual Studio del Manual de Instalación de Belatrix

En la *figura 9* se puede apreciar las clases divididas dentro de las carpetas anteriormente señaladas. en la carpeta drivers se puede visualizar Drive Chrome, Driver FF, Driver IE las cuales indican que solo se esta ejecutando las pruebas automatizadas para los navegadores, Chrome, Internet explorer y firefox. La idea es separar la arquitectura para tener más código mantenible, si en algún momento queremos añadir otro navegador, solo sería incluir una clase más y no modificar las anteriores clases.

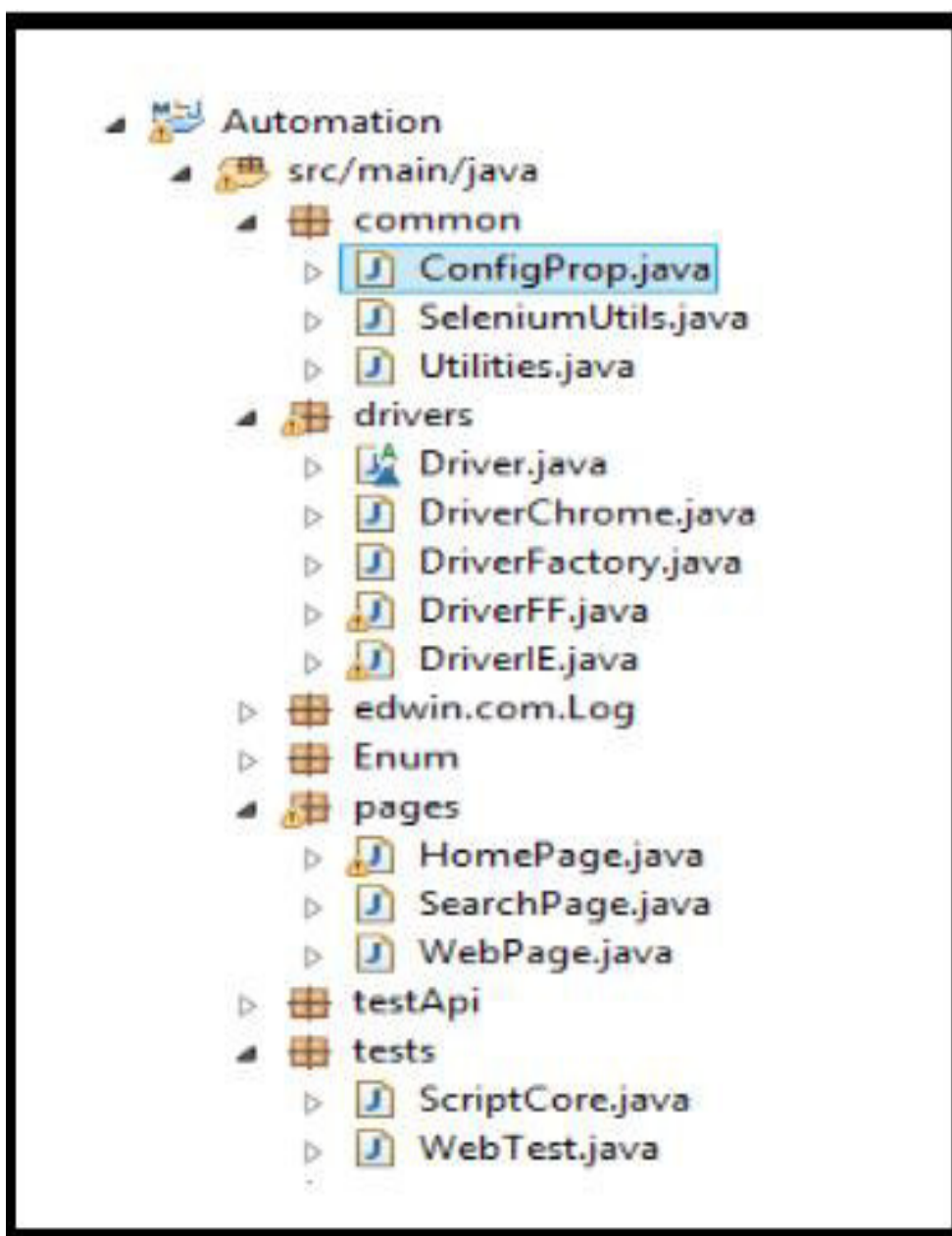


Figura 9 - Arquitectura de las clases del Manual de Instalación de Belatrix

En la figura 10 podemos visualizar un caso de prueba específica los pasos del caso de prueba, en el que se especifica el flujo completo de una validación. Se divide en pasos a seguir y un resultado esperado. al seguir estos pasos en la aplicación se debería comportar como tal , sino es así entonces se crea una incidencia.

TEST CASE 205377

205377 DentalHub - API - Locations Get: a Location that belongs to a different business should return Forbidden (403)

Harold Hernandez 0 comments Add tag Save & Close Follow Refresh ...

State: Ready Area: Main Project\Teams\La Selección Updated by Harold Hernandez 12/14/2018

Reason: Completed Iteration: Main Project\6.6.4\PI 5 - Sprint 4

Steps Summary Associated Automation (1)

Status

Priority: 2

Automation status: Not Automated

Classification

WBX Module: DentalHub

Development

+ Add link

Development hasn't started on this item.

Related Work

+ Add link

Tests

198992 BS(API) - Locations - GET 1 rec... Updated 4 hours ago Active

Steps

Add a comment. Use # to link a work item, I to link a pull request, or @ to mention a person.

Figura 10. Caso de prueba E2E del TFS de Belatrix

En la figura 11, se especifica los pasos del caso automatizado, y la navegación en la aplicación de acuerdo siempre a los pasos de casos de pruebas.

```
package tests;
import org.testng.annotations.Test;

public class ScriptCore extends WebTest{
    public ScriptCore(){
        super();
        // TODO Auto-generated constructor stub
    }

    @Test
    public void ValidateFirst5ResultsInHomePage(){
        home.goHomePage().getHeader().changePrincipalLanguage();
        home.getHeader().insertSearchText(SearchConstants.SHOES.toString()).clickButtonSearch();
        search.selectShoesBrand(BrandConstants.PUMA.toString()).selectShoesSize(SizeConstants.TEN.toString()).validateResultIs
        search.changePriceAscendant().displayFirst5Result().SortFirst5AscendantResultNames();
        search.changePriceDescendant().displayFirst5Result();
    }
}
```

Figura 11. Caso de prueba Automatizado del framework de Automatización de Belatrix

de la figura *figura 12.1* se tiene que en cada sprint el equipo de QA formado por 5 personas toma 120 horas la ejecución de 932 casos de pruebas y *la figura 12.2* muestra un reporte donde la ejecución toma 109 minutos ejecutar los mismos tests, pero ya automatizados por lo que se puede decir que después de automatizar 932 test se pudo evaluar el tiempo de reducción que se ganó con la automatización fue el siguiente:

1 mes = 2 sprints	Duración en Horas/Hombres Sin la automatización	Duración en Horas/Hombres Con la automatización
3 meses de trabajo = 6 sprints	6 sprint x3 días x 5 personas x 8 Horas = 720 horas	6 Sprint x 2 horas= 12 horas
6 meses de trabajo = 12 Sprint	12 sprint x3 días x 5 personas x 8 Horas = 1440 horas	12 sprint x 2 horas= 24 horas

Tabla 5 Reporte de sprints de los últimos 6 meses

Se puede visualizar en el cuadro 1 que sin la automatización en 3 meses de trabajo se requirio 720 horas para ejecutar 932 test manualmente, esto debido a que en 3 meses de trabajo se tienen aproximadamente 6 sprints, y en cada término de 1 sprint se tiene que ejecutar la regresión que toma 3 días utilizando 5 recursos de QA los cuales laboran 8 horas diarias resultan 720 horas.

Y utilizando la automatización esto toma solamente 2 horas ejecutar 932 test por lo que 3 meses de trabajo sólo genera el uso de 12 horas.

La reducción del tiempo es aproximadamente del 98% del tiempo total.

Regresion	# Casos de pruebas	Tiempo	Costo
Personas	930		
QA 1	185	3 días (60 casos de pruebas ejecutados por día)	24 horas-Hombre
QA 2	185	3 días (60 casos de pruebas ejecutados por día)	24 horas-Hombre
QA 3	185	3 días (60 casos de pruebas ejecutados por día)	24 horas-Hombre
QA 4	185	3 días (60 casos de pruebas ejecutados por día)	24 horas-Hombre
QA 5	185	3 días (60 casos de pruebas ejecutados por día)	24 horas-Hombre
			120 horas

Tabla 6 Caso de prueba Automatizado del TFS de Belatrix

En la figura 12.2 se muestra un reporte de la ejecución de los test automatizados, en donde se visualiza la ejecución de los casos de pruebas y el tiempo de ejecución es de 139 minutos en para los tests E2E y de 10 minutos para los de tipo Integration.

Test Group	Test Count	Status	Duration
DentalHubIntegration	1367 tests	200 failed	
DentalHubE2E	790	Failed	139 min
DentalHubE2E.Tests	7	Failed	44 sec
DentalHubE2E.Tests.DentistProfile	224	Failed	45 min
AcIsLicenseTests	18	Failed	4 min
AnesthesiaLicenseTests	21	Failed	3 min
BIsLicenseTests	18	Failed	4 min
CDSLICENSETests	21	Failed	3 min
DEALicenseTests	11	Failed	1 min
DOOAttachmentTests	10	Passed	1 min
GeneralInfoTests	13	Failed	3 min
HospitalAffiliationTests	17	Failed	3 min
MedicaidLicenseTests	18	Failed	3 min
MedicareLicenseTests	15	Failed	2 min
OtherAttachmentTests	8	Passed	58 sec
PracticeInformationTests	8	Failed	1 min
ProfessionalInsuranceTests	8	Failed	2 min
ProviderEducationTests	21	Failed	3 min
StateLicenseTests	12	Failed	1 min
SupplementalQuestionsTests	5	Failed	2 min
DentalHubE2E.Tests.PatientTreatment	559	Failed	93 min
DentalHubIntegration	577	Failed	10 min
DentalHubIntegration.Tests	577	Failed	10 min

Figura 12.2 Caso de prueba Automatizado del framework de Automatización de Belatrix

CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

La inclusión de la actividad de automatizar los casos de pruebas y la construcción del framework de automatización en la consultora belatrix, ha permitido que el tiempo que toma la ejecución de los casos de pruebas se redujera considerablemente. Detallaremos lo más relevante de la experiencia del uso del framework de automatización:

- Se evaluaron escenarios y casos de pruebas principales para el negocio: (i) todos los tests que validan los servicios fueron automatizados, (ii) todos los escenarios relacionado a transacciones fueron automatizados (iii) la ejecución de los casos de pruebas fueron realizados en Browsers de tipo Chrome, Internet Explorer, Firefox y ejecutadas en diferentes servidores de belatrix

- El software base elegido fue Windows 10 seleccionado entre todas las opciones de distribuciones de las versiones de Chrome elegida es la versión 56.0, internet Explorer 10 y Firefox 44.0 y safari se consideraron estas versiones puesto que los usuarios inicialmente utilizan aplicaciones web para poder realizar la transacción.

- Los ambiente que cuenta El equipo de calidad es UAT 1, UAT 2, STAGING, para ello se tiene un archivo properties donde se configura los ambientes donde se desea ejecutar los casos de pruebas automatizados, tener en consideración que en estos ambientes se despliega las nuevas funcionalidades.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se encontrará las conclusiones a lo que se llegó después de la construcción del framework de automatización.

5.1 CONCLUSIONES

El framework de automatización fue adecuadamente implementado, y sirvió para apoyar al área de QA en la mejora del tiempo de ejecución de los casos de pruebas a partir de inicios del 2018.

- Se redujo el tiempo de la ejecución de los casos de pruebas, de 3 días a 2 horas
- Se evitó la Intervención humana en la ejecución
- Se añadió una actividad al proceso de QA.
- Se instaló el ambiente de automatización al equipo del área de QA.

REGRESION ANTES DE LA AUTOMATIZACION				REGRESION DESPUES DE LA AUTOMATIZACION			
Regresion	# Casos de pruebas	Tiempo	Costo	Regresion	# Casos de p	Tiempo	Costo
Personas	930			Personas	930		
QA 1		185 3 dias (60 casos de pruebas ejecutados por dia)	24 horas-Hombre	Ejecucion Automatica	930	1.5 horas	Uso del servidor
QA 2		185 3 dias (60 casos de pruebas ejecutados por dia)	24 horas-Hombre				
QA 3		185 3 dias (60 casos de pruebas ejecutados por dia)	24 horas-Hombre				
QA 4		185 3 dias (60 casos de pruebas ejecutados por dia)	24 horas-Hombre				
QA 5		185 3 dias (60 casos de pruebas ejecutados por dia)	24 horas-Hombre				
			120 horas				

Tabla 6 .Reporte de la ejecución de automatización antes y después del TFS de Belatrix

5.2 RECOMENDACIONES

Replicar el modelo de implementación de framework de automatización para otros proyectos donde aún trabajan con ejecución manual y que como trabajos futuros, se puede identificar como implementar la ejecución de los test automatizados contra servidores en la Nube usando Browserstack.

5.4 FUENTES DE INFORMACIÓN.

La experiencia de implementación del framework de automatización de la empresa Belatrix, del proyecto Carnival - Dental H. fue una experiencia directa, por lo que la fuente de información es primaria.

5.5. GLOSARIO

Alta disponibilidad: Es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

Windows: Es una bifurcación a nivel binario de la distribución de Windows desarrollado por Microsoft, . Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la actual versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.

Cluster: El término clúster (del inglés cluster, que significa grupo o racimo) se aplica a los conjuntos o conglomerados de ordenadores unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora. La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software para aplicaciones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

Selenium: Selenium se compone de múltiples herramientas de automatización de software, como Selenium IDE, Selenium RC (selenium 1.0), y Selenium webdriver (selenium 2.0). Selenium IDE es un entorno de desarrollo integrado para construir el guiones de prueba. Es un complemento de Firefox que le permite grabar, editar y depurar los casos de prueba de selenio. Graba todo las acciones realizadas por el usuario final y generar los scripts de prueba. El control remoto de selenio (RC) fue selenio principal proyecto desde hace mucho tiempo. Selenium RC es más lento que el webdriver de selenio porque usa el programa de script java llamado núcleo de selenio. Selenium RC requiere iniciar el servidor antes de ejecutar los scripts de prueba, como se menciona en el artículo SATISH [2].

TestNG: TestNG se compone de múltiples herramientas de automatización de software, como TestNG Selenium IDE, TestNG Selenium RC (selenium 1.0), y Selenium webdriver (selenium 2.0). Selenium IDE es un entorno de desarrollo integrado para construir el guiones de prueba. Es un complemento de Firefox que le permite grabar, editar y depurar los casos de prueba de selenio. Graba todo las acciones realizadas por el usuario final y generar los scripts de prueba. El control remoto de selenio (RC) fue selenio principal proyecto desde hace mucho tiempo. Selenium RC es más lento que el webdriver de selenio porque usa el

programa de script java llamado núcleo de selenio. Selenium RC requiere iniciar el servidor antes de ejecutar los scripts de prueba, como se menciona en el artículo SATISH [2].

SeleniumIDE: Selenium IDE se compone de múltiples herramientas de automatización de software, como Selenium IDE, Selenium RC (selenium 1.0), y Selenium webdriver (selenium 2.0). Selenium IDE es un entorno de desarrollo integrado para construir el guiones de prueba. Es un complemento de Firefox que le permite grabar, editar y depurar los casos de prueba de selenio. Graba todo las acciones realizadas por el usuario final y generar los scripts de prueba. El control remoto de selenio (RC) fue selenio principal proyecto desde hace mucho tiempo. Selenium RC es más lento que el webdriver de selenium porque usa el programa de script java llamado núcleo de selenio. Selenium RC requiere iniciar el servidor antes de ejecutar los scripts de prueba, como se menciona en el artículo SATISH [2].

Hosting: Es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web. Es una analogía de "hospedaje o alojamiento en hoteles o habitaciones" donde uno ocupa un lugar específico, en este caso la analogía alojamiento web o alojamiento de páginas web, se refiere al lugar que ocupa una página web, sitio web, sistema, correo electrónico, archivos etc. en internet o más específicamente en un servidor que por lo general hospeda varias aplicaciones o páginas web.

IP flotante: En términos generales, una IP flotante es una dirección IP pública enrutable que no se asigna a ninguna instancia de manera automática. En lugar de eso, el propietario de un proyecto la pone a disposición de una o varias instancias de manera temporal. La instancia correspondiente dispone así tanto de una IP estática que le ha sido concedida automáticamente para la comunicación en un ámbito de red privado no enrutable, como también de una IP flotante asignada manualmente. Esto hace accesibles los servicios de una entidad a usuarios fuera de una nube.

En aquellas situaciones configuradas para que tenga lugar la conmutación por error, dicha dirección IP "flota" (inglés: floating = flotar) o se desliza con dinamismo hacia otra unidad activa en la red con el objetivo de que dicha unidad se haga cargo sin demoras de, por ejemplo, la función de una instancia que ya no está activa y de que, en su lugar, responda a las peticiones entrantes.

Máquina virtual: Una máquina virtual es un software que simula un sistema de computación y puede ejecutar programas como si fuese una computadora real. Este software en un principio fue definido como "un duplicado eficiente y aislado de una máquina física". La acepción del término actualmente incluye a máquinas virtuales que no tienen ninguna equivalencia directa con ningún hardware real..

Memcached: Memcached es un sistema distribuido de propósito general para caché basado en memoria, diseñado por Danga Interactive y que es muy usado en la actualidad por múltiples sitios web. Memcached es empleado para el almacenamiento en caché de datos u objetos en la memoria RAM, reduciendo así las necesidades de acceso a un origen de datos externo (como una base de datos o una API). Memcached tiene versiones para Linux, Windows y MacOS y se distribuye bajo licencia de software libre permisiva.

postgresql: postgresql es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web. MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL A.B. fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Nginx: nginx (pronunciado en inglés “engine X”) es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada; también existe una versión comercial distribuida bajo el nombre de nginx plus. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows.

Smtpt: El Simple Mail Transfer Protocol (SMTP) o “protocolo para transferencia simple de correo”, es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA, teléfonos móviles, impresoras, etc). Fue definido inicialmente en agosto de 1982 por el RFC 821 (para la transferencia) y el RFC 822 (para el mensaje). Son estándares oficiales de Internet que fueron reemplazados respectivamente por el RFC 2821 y el RFC 2822, que a su vez lo fueron por el RFC 5321 y el RFC 5322.

El funcionamiento de este protocolo se da en línea, de manera que opera en los servicios de correo electrónico. Sin embargo, este protocolo posee algunas limitaciones en cuanto a la recepción de mensajes en el servidor de destino (cola de mensajes recibidos). Como alternativa a

esta limitación se asocia normalmente a este protocolo con otros, como el POP o IMAP, otorgando a SMTP la tarea específica de enviar correo, y recibirlos empleando los otros protocolos antes mencionados (POP O IMAP).

Wordpress: WordPress es un sistema de gestión de contenidos o CMS (por sus siglas en inglés, Content Management System) enfocado a la creación de cualquier tipo de página web. Originalmente alcanzó una gran popularidad en la creación de blogs, para convertirse con el tiempo en una de las principales herramientas para la creación de páginas web comerciales. Está desarrollado en el lenguaje PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL y es software libre. Sus fundadores son Matt Mullenweg y Mike Little. WordPress fue creado a partir del desaparecido b2/cafelog y se ha convertido en el CMS más popular de la blogosfera y en el más popular con respecto a cualquier otro CMS de uso general. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

5.6. REFERENCIAS BIBLIOGRAFICAS

[1] LARS-OLA D. , LARS L. ; Results from introducing component-level test automation and Test-Driven Development, Ronneby, Sweden

[2] SATISH G. , RAHUL J. ; Analysis and Design of Selenium WebDriver Automation Testing Framework, Pune , India

[3] AZAD M. , SIEVERS J. ; n-Tiered Test Automation Architecture for Agile Software, California , USA

ANEXOS

1. ESPECIFICACIONES TÉCNICAS

INSTALAR TESTLINK

1. Entorno

Este tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Intel Core 2 CPU T7200 @ 2.00GHz x 2
- Sistema Operativo: Ubuntu 13.04 x32
- netBeans IDE 7.3

2. Introducción

Ya sabemos toda la importancia de realizar test de nuestras aplicaciones, pero también es necesario tener los test bien organizados, documentados, descritos y seguros.

Para optimizar la calidad de nuestro software, es importante tener un control de calidad de nuestros test.

Disponemos de muchas herramientas en internet para todo esto, pero voy a hablaros de una en concreto, TestLink.

3. ¿Qué es TestLink?

Es una herramienta Open Source con la que mantendremos una mejor calidad de nuestros test.

Nos permite responder a las siguientes preguntas:

- ¿Para qué requisitos necesitamos escribir o actualizar casos de test?
- ¿Que test queremos ejecutar para esta versión?

- ¿Cuanto hemos progresado testeando esta versión?
- ¿Qué test están fallando? ¿Cuales son los errores?
- ¿Necesitamos algún cambio en esta versión?

Con TestLink tenemos todo esto bajo control.

Podemos descargarla desde Sourceforge: [aquí](#).

Soporta todos los navegadores desde Internet Explorer 6.

Las dependencias de fondo de TestLink son:

- Un servidor web como Apache 1.3 y 2 o superior
- Un gestor de bases de datos como MySQL 4 o superior.
- Php 5.2 o superior.
- Y un sistema de trazas de bug, como colaborador opcional. (Ej: Bugzilla.)

4. Instalación de la aplicación

Para tener esta aplicación instalada en nuestro servidor web, es muy sencillo. Pasos:

- Tener un servidor web como Apache con Php5 y un servicio SQL de base de datos. Por ejemplo XAMPP.
- Tener descargado el paquete de SourceForge.
- Descomprimos el paquete de TestLink en el directorio que queramos dentro de nuestro servidor.
- TestLink posee un script de instalación automática, que te ayudará a instalar todas las directivas de configuración, así como la estructura de la base de deatos. Desde nuestro navegador acceder

a: <http://nuestroServidor/testlink/install/index.php>

TestLink Prague 1.9.5 Installation

You are installing TestLink 1.9.5

Migration from 1.9.3 to 1.9.5 has to be done MANUALLY.

Please read Section on README file or go to www.teamst.org (Forum: TestLink 1.9.5 News,changes, etc)

Open [Installation manual](#) for more information or troubleshooting. You could also look at [README](#) or [Changes Log](#). You are welcome to visit our [forum](#) to browse or discuss.

[New installation](#)

Este nos hará aceptar unas condiciones.

Acceptance of License

- Verification of System and configuration requirements
- Definition of DB access
- Create DB, testlink DB user, structures and default data & create configuration file.
- Verify the procedure result and continue to TestLink login.

TestLink is developed and shared under GPL license. You are welcome to share your changes with community. Please, confirm your understanding below.

The GNU General Public License is a Free Software license.

Like any Free Software license, it grants to you the four following freedoms:

- The freedom to run the program for any purpose.*
- The freedom to study how the program works and adapt it to your needs.*
- The freedom to redistribute copies so you can help your neighbor.*
- The freedom to improve the program and release your improvements to the public, so that the whole community benefits.*

You may exercise the freedoms specified here provided that you comply with the express conditions of this license. The principal conditions are:

- You must conspicuously and appropriately publish on each copy distributed an appropriate copyright notice and disclaimer of warranty and keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of the GNU General Public License along with the Program. Any translation of the GNU General Public License must be accompanied by the GNU General Public License.*

I agree to the terms set out in this license.

Continue

- Completar la información para nuestra base de datos

TestLink TestLink Prague 1.9.5 - New installation

- Acceptance of License
- Verification of System and configuration requirements
- Definition of DB access**
- Create DB, testlink DB user, structures and default data & create configuration file.
- Verify the procedure result and continue to TestLink login.

Database Configuration

Define your database to store TestLink data:

Database Type

Database host

*Note: In the case that you DB connection doesn't use **STANDARD PORT** for , you need to add '**port_number**', at the end Database host parameter. Example: you use MySQL running on port 6606, on server matrix then Database host will be matrix:6606*

Enter the name of the TestLink database . The installer will attempt to create it if not exists.

Database name

*Disallowed characters in Database Name:
The database name can contains any character that is allowed in a directory name, except '/', '\', or '!'.*

Table prefix (optional)

*Note: This parameter should be empty for the most of cases.
Using a Database shared with other applications: Testlink can be installed (using this installer) on a existing database used by another application, using a table prefix.
Warning! PART OF INSTALLATION PROCESS CONSISTS on dropping all TestLink tables present on the database/schema (if any TestLink table exists). Backup your Database Before installing and load after this process.*

Set an existing database user with administrative rights (root):

Database admin login

Database admin password

*This user requires permission to create databases and users on the Database Server.
These values are used only for this installation procedures, and is not saved.*

- Y cumplir ciertos requisitos que nos indicará a continuación:

- ☐ Acceptance of License
- ☐ **Verification of System and configuration requirements**
- ☐ Definition of DB access
- ☐ Create DB, testlink DB user, structures and default data & create configuration file.
- ☐ Verify the procedure result and continue to TestLink login.

TestLink will carry out a number of checks to see if everything's ready to start the setup.

System requirements

Server Operating System (no constrains)	Linux
PHP version	OK (5.2.0 [minimum version] <= 5.4.9-4ubuntu2 [your version])

Web and PHP configuration

Maximum Session Idle Time before Timeout	24 minutes and 0 seconds - (Short. Consider to extend.)
Checking max. execution time (Parameter max_execution_time)	30 seconds - We suggest 120 seconds in order to manage hundred of test cases (edit php.ini)
Checking maximal allowed memory (Parameter memory_limit)	OK (128 MegaBytes)
Checking if Register Globals is disabled	OK
Checking MySQL Database	OK
Checking Postgres Database	Failed! Postgres Database cannot be used.
Checking MSSQL Database	Failed! MSSQL Database cannot be used.
Checking GD Graphic library	Failed! GD Graphic library not enabled. Graph rendering requires it. This feature will be disabled. It's recommended to install it..
Checking LDAP library	Failed! LDAP library not enabled. LDAP authentication cannot be used. (default internal authentication will works).
Checking JSON library	OK

Read/write permissions

For security reason we suggest that directories tagged with [S] on following messages, will be made UNREACHABLE from browser	
Checking if /home/rmacias/projects/testlink-1.9.5/gui/templates_c directory exists	OK
Checking if /home/rmacias/projects/testlink-1.9.5/gui/templates_c directory is writable (by user used to run webserver process)	OK
Checking if /var/testlink/logs/ directory exists [S]	OK
Checking if /var/testlink/logs/ directory is writable (by user used to run webserver process)	OK
Checking if /var/testlink/upload_area/ directory exists [S]	OK
Checking if /var/testlink/upload_area/ directory is writable (by user used to run webserver process)	OK

- Una vez hecho esto la instalación estará completada y podremos continuar:

TestLink setup will now attempt to setup the database:

Creating connection to Database Server:OK!

Database testlink does not exist.

Will attempt to create:

Creating database `testlink` :OK!

Creating Testlink DB user `admin` :OK! (ok - new user)

Processing:sql/mysql/testlink_create_tables.sql

OK!

Writing configuration file:OK!

YOUR ATTENTION PLEASE:

To have a fully functional installation You need to configure mail server settings, following this steps

- copy from config.inc.php, [SMTP] Section into custom_config.inc.php.
- complete correct data regarding email addresses and mail server.

Installation was successful!

You can now log in to Testlink (using login name:admin / password:admin - Please Click Me!).

- El siguiente paso será configurar.
- Si queremos instalar manualmente seguir la documentación en el manual.

5. Configurando la aplicación

- Todos los parámetros de configuración están en el fichero `config.inc.php` y todos los ficheros incluidos en él.
 - **`config.inc.php`**: contiene configuración principal.
 - **`config_db.inc.php`**: contiene parametros de configuración de acceso a la bd.
 - **`custom_config.inc.php`**: sirve para modificar los valores por defecto de los parametros de *config.inc.php*, esto nos facilita la modificacion.
- Antes de cambiar nada es recomendable tener una copia de seguridad de nuestra configuración tras la instalación.
- Se recomienda usar **`custom_config.inc.php`** para nuestros cambios, esto nos permite guardar nuestra configuración en caso de una actualización.
- Testlink nos permite una comunicación directa con nuestro bug tracker. Para habilitarlo tenemos que cambiar el parámetro

Shell



```
1 $g_interface_bugs = 'NO';
```

entre los siguientes valores posibles: 'NO', 'BUGZILLA', 'MANTIS', 'JIRA', 'TRACKPLUS', 'EVENTUM', 'SEAPINE' o 'TRAC'

- Para más opciones de configuración consultar el manual.

6. Terminología y flujo de trabajo

Una vez instalado y configurado a nuestro gusto vamos a profundizar.

Esta aplicación nos proporciona un mecanismo de documentación para nuestros test bastante completo.

Desde organizar un Test Project hasta la misma documentación del resultado de nuestros test.

Tenemos un usuario Administrador creado en la misma instalación, el cual creará un Test Project y mínimo dos usuarios:

☒ un Leader, encargado de definir los requisitos de nuestro software y organizar los casos de prueba vacíos (Test Case) en Suites de prueba (Test Suite).

☒ un usuario "Senior Tester" en cargo de rellenar esos Test Case vacíos con el escenario de prueba. (Steps)

Una vez creados estos Steps, podemos linkear los Test Cases a un Test Plan y a un Build creados anteriormente.

Podemos crear unas palabras clave "Keyword" para tener un filtro de test.

Una vez creado todo esto probaremos los test, y reflejaremos los resultados en esta aplicación.

7. Uso de la aplicación

Nos conectaremos con el usuario admin que hemos creado:



TESTLINK PRAGUE 1.9.5

Please log in ...

Login Name

admin

Password

Login

[New User?](#)

[Lost Password?](#)

[TestLink project Home](#)

TestLink is licensed under the GNU GPL

- There are security warnings for your consideration. See details on file: `/var/testlink/logs/config_check.txt`. To disable any reference to these checkings, set `$tCfg->config_check_warning_mode = 'SILENT';`

– Test project :

Vamos a crear un proyecto para test (necesitamos derechos de administrador).

Se nos presenta la siguiente gui:

TestLink Prague 1.9.5 : admin [admin] [My Settings | Logout]

Test Project ▼

Test Project Management : Create a new project

Name *

Prefix (used for Test case ID) *

Project description

Fuente Tamaño

Enhanced features

- Enable Requirements feature
- Enable Testing Priority
- Enable Test Automation (API keys)
- Enable Inventory

Issue Tracker Integration

- Active
- Issue Tracker ▼


Availability

- Active
- Public

* Mandatory fields




Lo rellenamos y al crearlo llegamos a una tabla con nuestros proyectos:

TestLink Prague 1.9.5 : admin [admin] [My Settings | Logout]

Project | Test Specification | User Management | Events | 01-  Test Project pruebaTestLink ▼

Test Project Management

Create

Name	Description	Prefix	Issue Tracker	Requirement Feature	Active	Public	delete
pruebaTestLink	Proyecto de prueba, y nunca mejor dicho.	01					

Despues crearemos dos usuarios:

- Leader user
- Senior Tester

TestLink Prague 1.9.5 : admin [admin] [My Settings | Logout]

Project | Test Specification | User Management | Events | 01- Test Project pruebaTestLink ▼

User Management

View Users | View roles | Assign Test Project roles | Assign Test Plan roles

Hide inactive users

Login	First Name	Last Name	Email	Role	Locale	Active	disable
admin	Testlink	Administrator		admin	English (wide/UK)		
ddiaz	Daniel	Diaz	ddiaz@autentia.com	senior tester	English (wide/UK)		
rmacias	Rafael	Macías	rmacias@autentia.com	leader	English (wide/UK)		

Create

El leader del test project declara los requisitos del Software y con estos crea unos casos de test que incluiremos en unos suites.

En esta aplicación crearemos primero un Suite de pruebas:

TestLink Prague 1.9.5 : admin [admin] [My Settings | Logout]

Project | Test Specification | User Management | Events | 01- Test Project pruebaTestLink ▼

Navigator - Test Specification

Settings ?

Update tree after every operation

Filters

Test Case ID 01-
Test Case Title prueb
Execution type Man

Apply
Reset Filters

Expand tree
Collapse tree

pruebaTes

Create Test Suite

Create Test Suite Cancel

Test Suite Name

Details

Fuente Tamaño

Keywords

Available Keywords	Assigned Keywords
<input type="text"/>	<input type="text"/>

Create Test Suite Cancel

Y a continuación de este estableceremos los casos de test.



Navigator - Test Specification

Settings

Update tree after every operation

Filters

Test Case ID

Test Case Title

Test Suite

Execution type

Apply Reset Filters

Expand tree Collapse tree

- pruebaTestLink (0)
- pruebaSuite (0)
- suTest (0)

Test Suite : pruebaSuite

Test Suite Operations

Create Edit Move/Copy Delete Sort alphabetically Import Export

Test Case Operations

Create Move/Copy Delete Sort by external ID Import Export

Test Suite : pruebaSuite

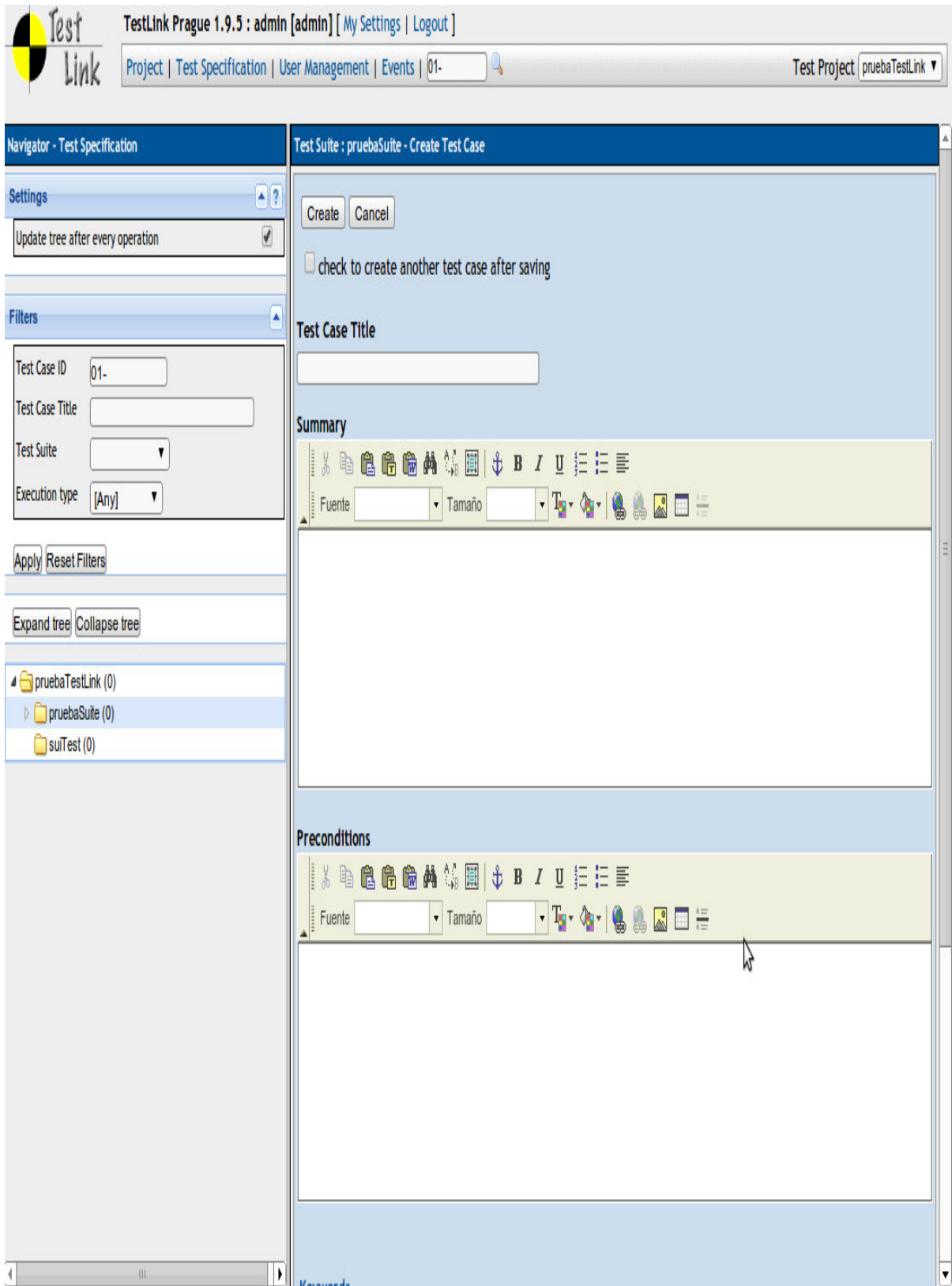
Details

suite de prueba

Keywords : None

Attached files :

Upload new file



En summary indicaremos la especificacion que queremos de nuestro software, por ejemplo:

– “Posibilidad de logear en la aplicacion”

En preconditions especificaremos las precondiciones para esa summary:

– “Tener iniciada la aplicación”

Otro ejemplo más claro seria:

– Summary: “Quiero que se cargue un menú de inicio”

– Preconditions: “El usuario se haya conectado”

De momento dejamos los steps sin definir.

Ahora crearemos un Test Plan, al que ligaremos todos los test cases.

En el panel de “Test Plan” hacemos click en build y nos dice que tenemos que crear uno, sera nuestro control de versión de cada test.

Es decir para el test “Probar un caso de test” creamos una build “Probar un caso de test 0.1” por ejemplo.



The screenshot shows the TestLink interface. At the top, there is a navigation bar with the TestLink logo and the text "TestLink Prague 1.9.3 : admin [admin] [My Settings | Logout]". Below this is a breadcrumb trail: "Project | Test Specification | Test Execution | Test Reports | User Management | Events | 01-". To the right of the breadcrumb trail is a search box and a dropdown menu for "Test Project" with the value "pruebaTestLink".

The main content area is titled "Build management - Test Plan : Prueba de testlink". It contains a table with the following columns: "Title", "Description", "Release date", "Active", "Open", and "delete".

Title	Description	Release date	Active	Open	delete
Probar un caso de test 0.1	Primera prueba	06/05/2013			

Below the table is a "Create" button. Underneath the button, there is explanatory text:

A build is identified by its title. Each build is related to the active Test Plan.
Description should include: list of delivered packages, fixes or features, approvals, status, etc.
A build has two attributes:
Active / Inactive - defines whether the build can be used. Inactive builds are not listed on the execution and reports pages.
Open / Closed - Only for open builds, test results can be modified.

Luego vamos a establecer una serie de Steps en nuestros casos de prueba.

Que van a ser nuestros propios test.

- Step1: Step actions: “Usuario y contraseña correctos”
Expected results: “Login true”
- Step2: Step actions: “Usuario correcto y contraseña incorrecta”
Expected results: “Login false”

The screenshot shows the TestLink Prague 1.9.5 interface. The main window displays the configuration for a test case titled "01-1: Probar un caso de test". The interface includes a left sidebar with a "Navigator - Test Specification" panel showing a tree view of the test suite structure: pruebaTestLink (1) > pruebaSuite (1) > 01-1: Probar un caso de test. The main content area shows the test case details, including its version (Version 1), creation and modification dates, and a summary: "Posibilidad de logear en la aplicacion". Below the summary, there are sections for "Preconditions" (Tener iniciada la aplicacion) and a table of "Step actions" and "Expected Results".

#	Step actions	Expected Results
1	Usuario y contraseña correctos	Login true
2	Usuario correcto y contraseña incorrecta	Login false
3	Usuario incorrecto	Login false
4	[Rich text editor area]	

Si queremos asignar previamente un caso de test a un usuario, en el panel principal buscamos “Assigning Test Case Execution” y lo hacemos, es muy intuitivo:

TestLink Prague 1.9.5 : admin [admin] [My Settings | Logout]

Project | Test Specification | Test Execution | Test Reports | User Management | Events | 01- | Test Project: pruebaTestLink

Execution assignment Assigning Test Case execution tasks for build build 0.1 test leader in test plan Plan del leader

Settings

Test Plan: Plan del leader
 Build to assign: build 0.1 test leader
 Update tree after every operation

Check/uncheck all Test cases: Do
 Bulk user assignment: Do
 Send mail notification to tester

Filters

Test Case ID: 01-
 Test Case Title:
 Test Suite:
 Execution type: [Any]
 Assigned to: [Any] include unassigned Test Cases
 Result: [Any]
 on: specific Build
 Build: build 0.1 test leader

- pruebaTestLink / Plan del leader (1)
 - pruebaLeaderSuite (1)
 - 01-4: case01Leader

<input checked="" type="checkbox"/>	Test Case [Version]	Assigned to	Assign to
<input checked="" type="checkbox"/>	01-4: case01Leader [1]		rmacias

Clicamos en save.

Ahora con todo esto vamos a ver como podemos documentar que nuestros test pasan o no.

Logeamos con el usuario al que se lo asignamos.

Clicamos en Test Execution, y accedemos mediante el panel lateral a nuestros casos.

The screenshot displays the TestLink interface for a test case. The top navigation bar includes the TestLink logo, the user 'rmacias [senior tester]', and links for 'My Settings' and 'Logout'. The breadcrumb trail shows 'Project | Test Specification | Test Execution | Test Reports | 01-'. The main content area is divided into several sections:

- Settings:** Includes dropdowns for 'Test Plan' (Plan del leader) and 'Build to execute' (build 0.1 test leader), a checkbox for 'Update tree after every operation', and an 'Export Test Plan' button.
- Filters:** A sidebar with various filters for 'Test Case ID', 'Test Case Title', 'Test Suite', 'Execution type', 'Assigned to', 'Result', and 'on'. It includes 'Apply', 'Reset Filters', and 'Advanced Filters' buttons.
- Test Case Details:** Shows 'Test Case ID 01-4:: Version : 1', 'case01Leader', and 'Assigned to : rmacias'. It indicates the last execution was 'Not Run'.
- Summary:** A yellow-bordered box containing:
 - Summary:** 'Probando los test'
 - Preconditions:** 'Estar conectado'
 - Execution type:** 'Manual'
 - Table:** A table with columns '#', 'Step actions', and 'Expected Results'. It contains one row: '1 Pruebo el test True'.
- Notes / Description:** A large empty text area for documenting test results.
- Result:** A panel with radio buttons for 'Not Run', 'Passed', 'Failed', and 'Blocked'. It includes 'Save execution' and 'Save and move to next' buttons.

An important notice at the bottom states: 'Important Notice: Once a Result is updated from 'Not Run' to another value, you cannot set it back to 'Not Run'. You can still set the Result to any other value.'

Aquí podremos documentar si pasan o no los test.

Save and move next guardará y nos moverá al siguiente caso.

Podemos clicar en Show completed execution history, y veremos todos los “commits” realizados:

The screenshot displays the TestLink Prague 1.9.5 interface. The top navigation bar includes the TestLink logo, the user 'rmaclias [senior tester]', and links for 'My Settings' and 'Logout'. The breadcrumb trail shows 'Project | Test Specification | Test Execution | Test Reports | 01-'. The main content area is divided into several sections:

- Settings:** Includes dropdowns for 'Test Plan' (Plan del leader) and 'Build to execute' (build 0.1 test leader), a checked 'Update tree after every operation' checkbox, and an 'Export Test Plan' button.
- Filters:** A sidebar with filters for 'Test Case ID' (01-), 'Test Case Title', 'Test Suite', 'Execution type' ([Any]), 'Assigned to' (rmaclias), and 'Result' ([Any]).
- Test Results on Build build 0.1 test leader:** Shows 'Only Test Cases assigned to : rmaclias'. It includes a 'Test Suite : pruebaLeaderSuite/' and a specific test case 'Test Case ID 01-4 :: Version : 1 case01Leader' assigned to 'rmaclias'.
- Execution history - Build : build 0.1 test leader:** A table with columns: Date, Tested by, Status, Test Case Version, attachments, and Run mode.

Date	Tested by	Status	Test Case Version	attachments	Run mode
06/05/2013 13:31:03	rmaclias	Passed	1	[icon]	[icon]
06/05/2013 13:30:37	rmaclias	Failed	1	[icon]	[icon]
06/05/2013 13:29:51	rmaclias	Passed	1	[icon]	[icon]
06/05/2013 13:29:04	rmaclias	Passed	1	[icon]	[icon]
- Summary:** A section with a yellow border containing 'Probando los test', 'Preconditions' (Estar conectado), 'Execution type : Manual', and a table of step actions.

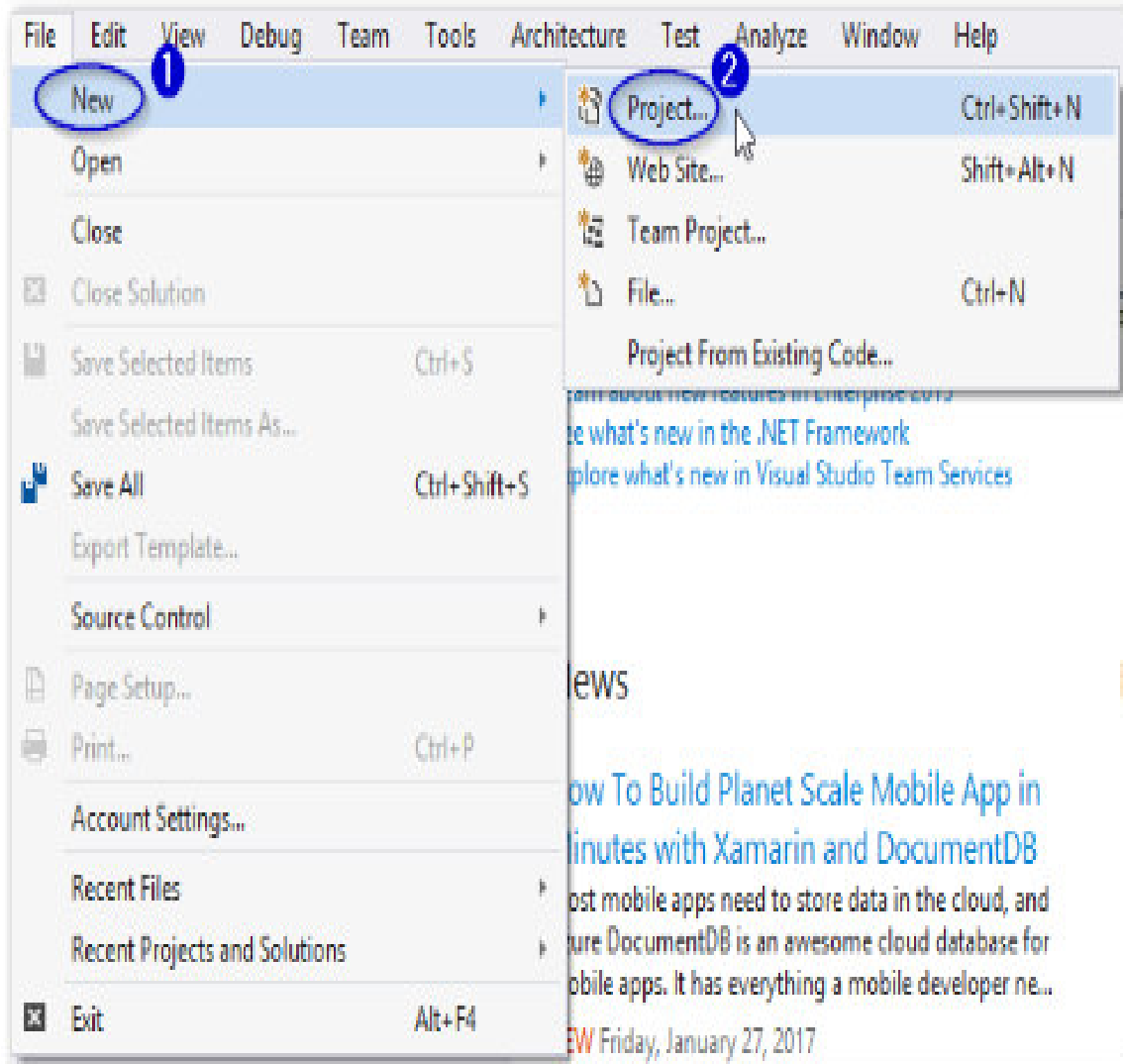
#	Step actions	Expected Results
1	Pruebo el test	True
- Notes / Description:** A large empty text area for notes.
- Result:** A legend for test results: Not Run, Passed, Failed, and Blocked.

2 INSTALAR SELENIUM CON C#

Para mayor detalle puede encontrar el tutorial completo en la siguiente pagina web: <http://learn-automation.com/install-selenium-webdriver-with-c/>

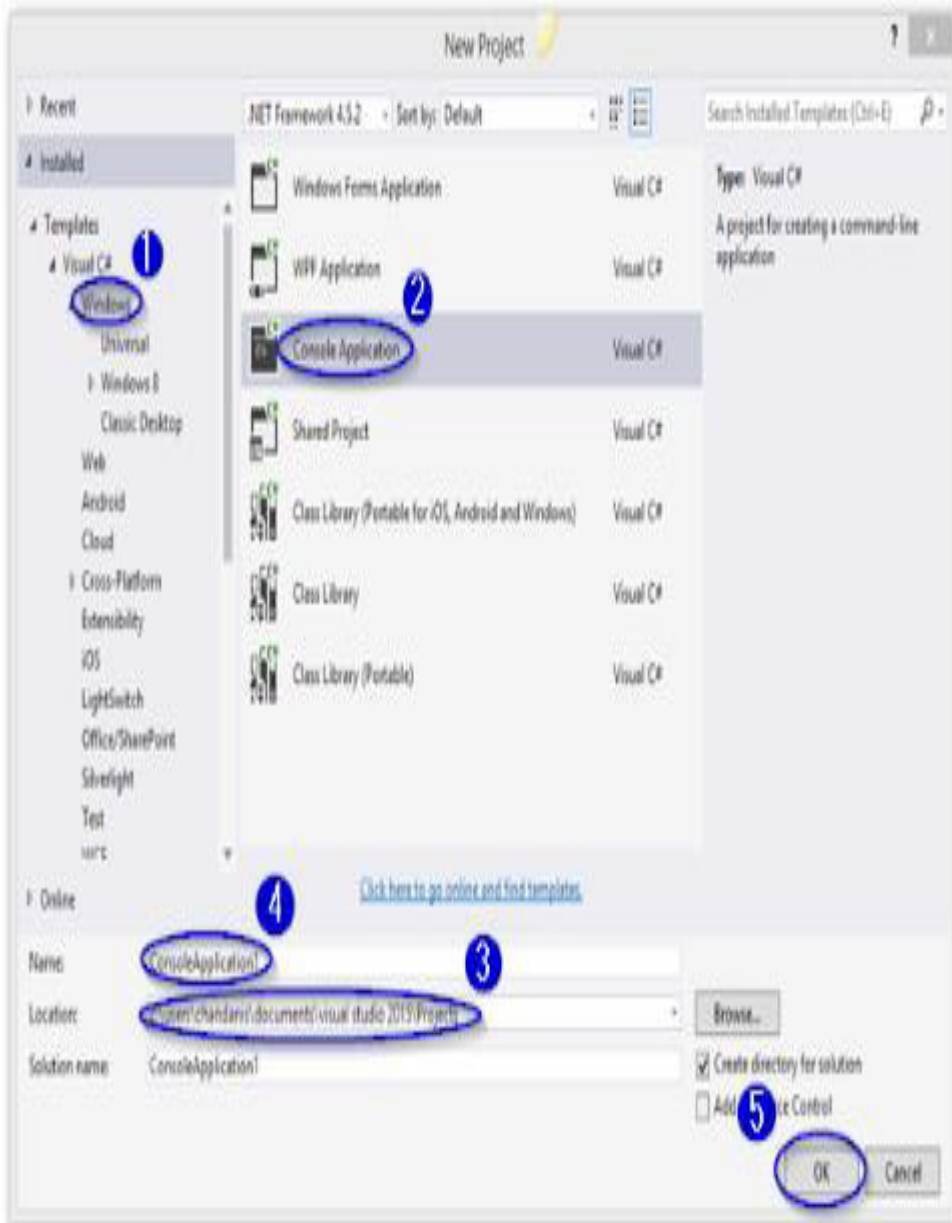
Crear un Nuevo proyecto en Visual Studio:

Paso 1) En el File Menu, Click New > Project

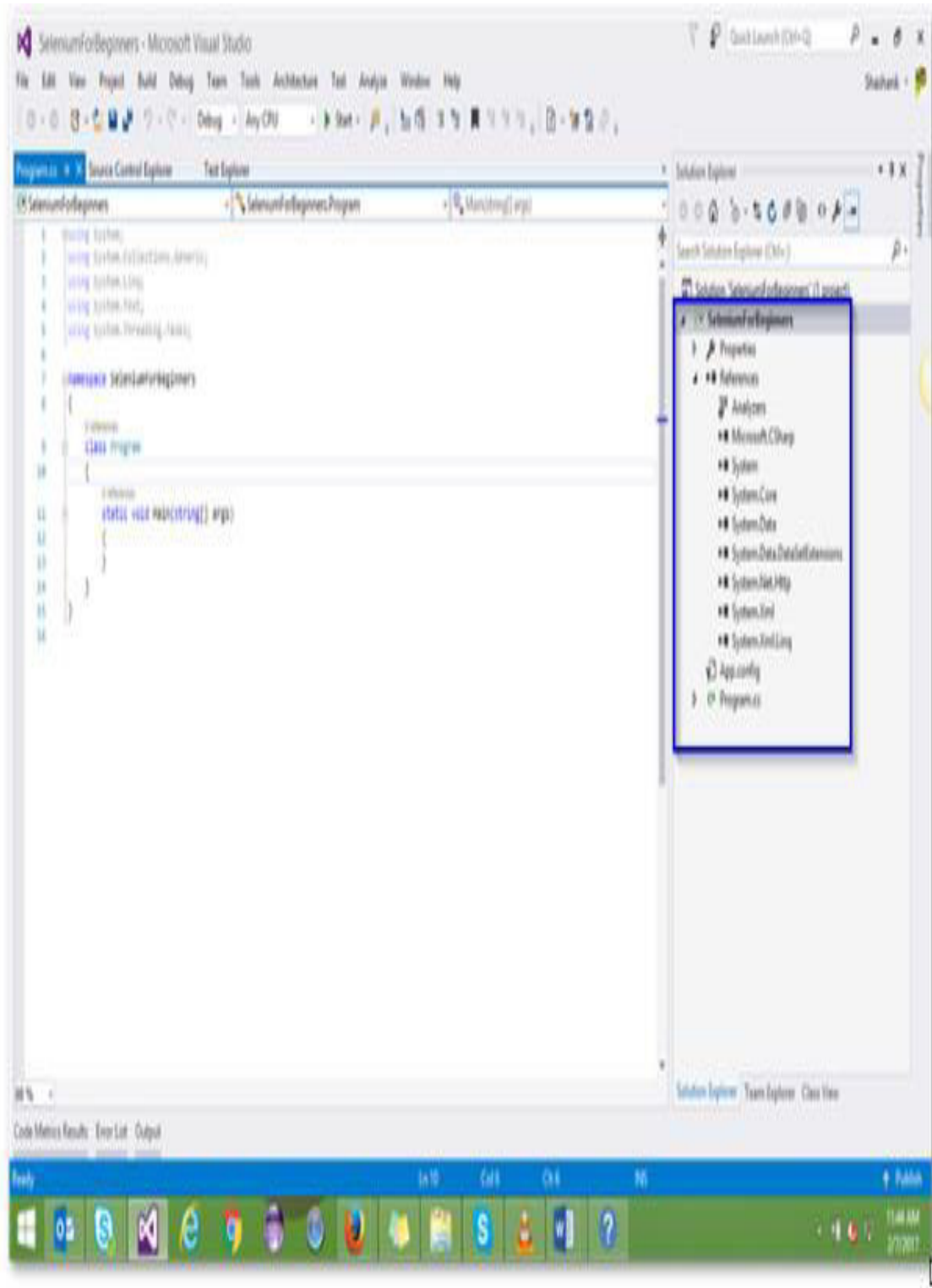


Step 2) En la siguiente pagina ,

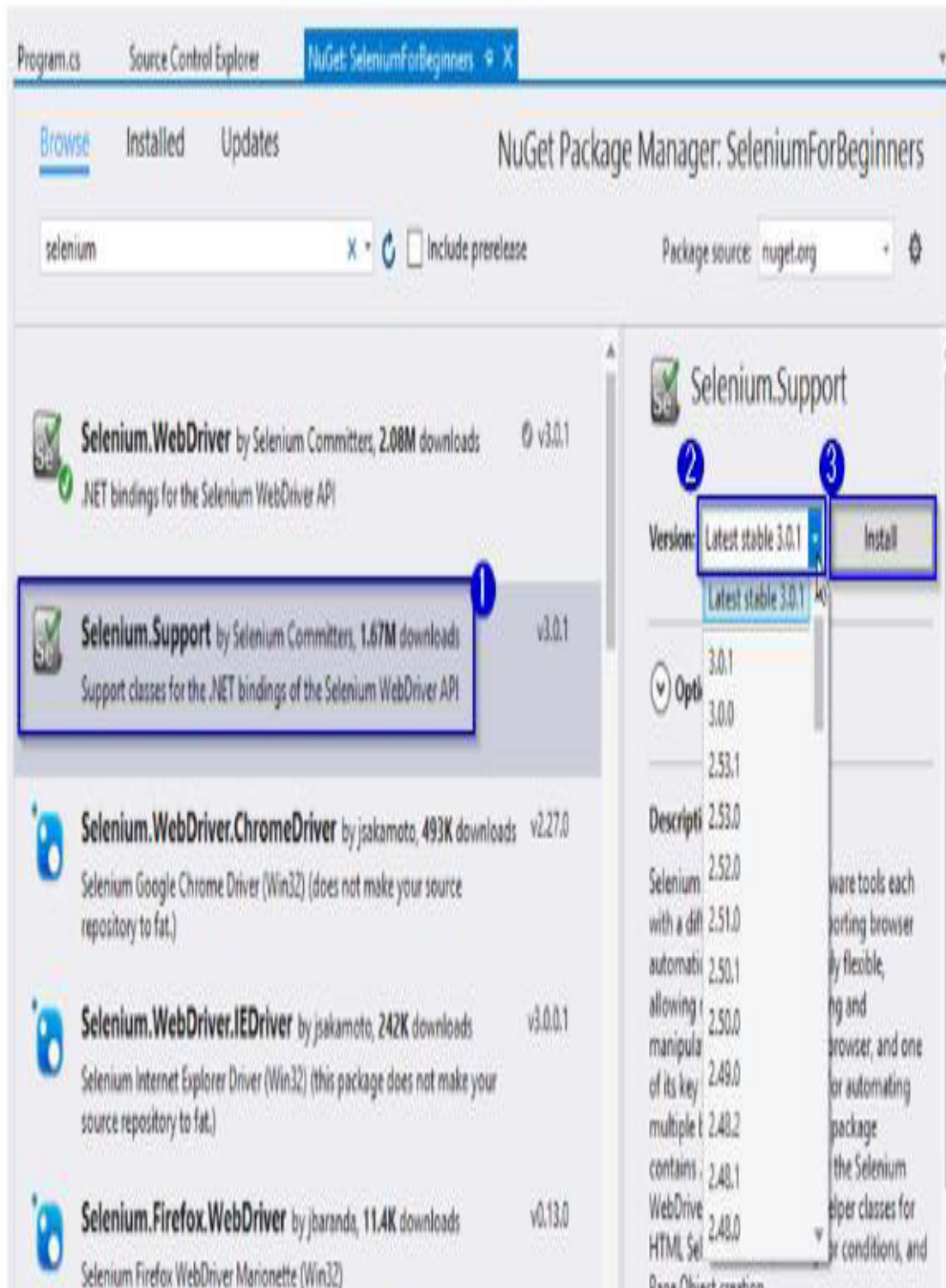
1. Seleccionar the option 'Visual C#'
2. Click on Console App (.Net Framework)
3. Ingresar nombre como "Guru99"
4. Click OK



Step 3) se muestra creado correctamente



Paso 4) seleccionar estas opciones Navigate to Tools -> NuGet Package Manager -> Manage NuGet Packages for Solution



3 INSTALAR VISUAL STUDIO 2017 COMUNIDAD

Visual Studio es el IDE de Microsoft para desarrollar aplicaciones móviles con Xamarin, hoy aprenderemos a instalar la versión **gratuita** para que estés listo a tomar tu Curso de Xamarin disponible ya en Platzi.

Visual Studio trae para nosotros una versión para Windows y ahora también lo tenemos disponible en macOS, el sitio para descargar el entorno correspondiente es <https://www.xamarin.com/download>, en el caso de navegar desde tu computadora Windows, **descarga Visual Studio Community con Xamarin**. El instalador te guiará a través de los pasos necesarios para tener tu entorno Windows listo.

Recuerda que, si deseas probar, diseñar y compilar tus aplicaciones **iOS**, como hemos mencionado anteriormente, necesitarás de una computadora macOS conectada a la misma red.

Asegúrate que, al instalar Visual Studio Community 2017, seleccionas Mobile Development with .NET como una de las funcionalidades que instalarás, adicionalmente puedes seleccionar las que tu puedas necesitar, pero para este curso, ésta será suficiente:

Visual Studio aparte trae para nosotros una versión para Windows y también para MAC y ahora también lo tenemos disponible en macOS, el sitio para descargar el entorno correspondiente es <https://www.xamarin.com/download>, en el caso de navegar desde tu computadora Windows, **descarga Visual Studio Community con Xamarin**. El instalador te guiará a través de los pasos necesarios para tener tu entorno Windows listo.

Recuerda que, si deseas probar, diseñar y compilar tus aplicaciones **iOS**, como hemos mencionado anteriormente, necesitarás de una computadora macOS conectada a la misma red.

Nota también que de lado derecho hay que seleccionar los paquetes que permitirán ciertas características dentro de nuestro entorno, te recomiendo **selecciones todas las opciones** para tener tu entorno tan completo como es posible, las únicas opciones que podrías omitir es la de Windows 10 Mobile, si no vas a desarrollar para esa plataforma,

y el soporte para F# si no vas a usar ese lenguaje. Durante este curso, no usaremos ninguno de los dos, **nos enfocaremos en iOS y Android**, y utilizaremos **C#** como lenguaje de programación.

Visual Studio

Products

Installed

Visual Studio Community 2017

Acquiring Microsoft.VisualStudio.VC.141.Debugger
3%

Applying Microsoft.TeamFoundation.OfficeIntegration
7%

Cancel

Available

<p>Visual Studio Enterprise 2017 Microsoft DevOps solution for productivity and coordination across teams of any size</p> <p>License terms Release notes 15.1 (26403.7)</p> <p>Install</p>	<p>Visual Studio Professional 2017 Professional developer tools and services for small teams</p> <p>License terms Release notes 15.1 (26403.7)</p> <p>Install</p>
---	--

Welcome!

We invite you to go online to hone your skills and find additional tools to support your development workflow.

Learn

Whether you're new to development or an experienced developer, we have you covered with our tutorials, videos, and sample code.

Marketplace

Use Visual Studio extensions to add support for new technologies, integrate with other products and services, and fine-tune your experience.

Need some help?

Check out the [Microsoft Developer Community](#) where developers provide feedback and answers to many common problems.

Get help from Microsoft at [Visual Studio Support](#)

Platzi
1.9.30413.1

NOTA: Es probable que Visual Studio haya sido instalado en inglés, si lo deseas, siempre puedes lanzar el instalador de nuevo para agregar, desde la pestaña de “language packs”, el lenguaje español. Una vez hecho esto, desde Visual Studio, en la pestaña de “tools”, y el menú “options” podrás encontrar dentro de la sección de “Environment” la configuración de “International settings”, donde puedes seleccionar de entre los idiomas instalados.

Tras seleccionar la configuración deseada, tu entorno esta listo para ser utilizado para crear aplicaciones Xamarin.

