



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



INTEGRACIÓN DE AUTOMATIZACIÓN Y DISTRIBUCIÓN DE PRUEBAS
UTILIZANDO MICROSOFT AZURE Y CONTENEDORES DOCKER EN LA
EMPRESA DIGITAL WARE

AUTOR: DIEGO JAVIER LEAL DIAZ

PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, ELÉCTRICA, SISTEMAS Y
TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURAS



UNIVERSIDAD DE PAMPLONA

PAMPLONA, 16 DE MAYO DEL 2021



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



**INTEGRACIÓN DE AUTOMATIZACIÓN Y DISTRIBUCIÓN DE PRUEBAS
UTILIZANDO MICROSOFT AZURE Y CONTENEDORES DOCKER EN LA
EMPRESA DIGITAL WARE**

DIEGO JAVIER LEAL DIAZ

**Trabajo de grado presentado como requisito para optar al título de
INGENIERO EN TELECOMUNICACIONES**

**Director: HERNANDO JOSÉ VELANDIA VILLAMIZAR
ING. ELECTRÓNICO
MAG. INGENIERÍA BIOMEDICA
Codirector: GUSTAVO ADOLFO QUIJADA MACUART
ING. EN TELECOMUNICACIONES**

**PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA, ELÉCTRICA, SISTEMAS Y
TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURAS
UNIVERSIDAD DE PAMPLONA
PAMPLONA, 16 DE MAYO DEL 2021**

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES**



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



**PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES
TRABAJO PRESENTADO PARA OPTAR POR EL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

TEMA:

**INTEGRACIÓN DE AUTOMATIZACIÓN Y DISTRIBUCIÓN DE PRUEBAS,
UTILIZANDO MICROSOFT AZURE Y CONTENEDORES DOCKER EN LA
EMPRESA DIGITAL WARE**

FECHA DE INICIO DEL TRABAJO:

FECHA DE TERMINACIÓN DEL TRABAJO:

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:

Diego Leal Diaz

**DIEGO JAVIER LEAL DIAZ
AUTOR**

**HERNANDO JOSÉ VELANDIA
DIRECTOR**

**HERNANDO JOSÉ VELANDIA
DIRECTOR DE PROGRAMA**

JURADO CALIFICADOR:

**GERMÁN ARLEY PORTILLA
ING. ELECTRÓNICO**

**JOHRMAN DE JESUS VIDES
ING. EN TELECOMUNICACIONES**



SC-CER96940

"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



AGRADECIMIENTOS

En primera instancia doy gracias a Dios por ser mi apoyo, mi guía, mi sustento y darme la fuerza ante las distintas adversidades de la vida, infinitamente agradezco a mis padres por todo el apoyo incondicional, por creer en mis capacidades y por darme la oportunidad de formarme como profesional y poder lograr esa meta que tanto anhelábamos desde el inicio de la carrera y a mis hermanos que han sido fuente y apoyo para solventar las distintas situaciones y adversidades que se presentaron durante el proceso de formación y durante toda mi vida. Agradezco a mi novia por haberme acompañado en todo momento y siempre motivarme, amarme y apoyarme ante las distintas dificultades y problemas presentados.

Doy gracias a cada uno de mis docentes que impartieron parte de su conocimiento a lo largo de toda mi formación profesional y me ayudaron a ser lo que soy hoy en día, agradezco a cada uno de mis compañeros que compartieron parte de su vida conmigo, en especial a Fernando y Darwin que fueron los guerreros de batallas y trasnochos para poder llegar a la meta sin rendirnos en ningún momento.

Agradezco al ingeniero Gustavo Quijada por todos los consejos brindados, por ser una gran ayuda durante el proceso de pasantías y principalmente por ser un gran amigo, agradecer de la misma manera al ingeniero Edwin Barreto por abrirme las puertas en el área de automatización de la empresa Digital Ware como pasante universitario y poner ese voto de confianza en mi crecimiento personal y profesional. A Digital Ware por permitirme conocer verdaderamente la industria y formar en mí el verdadero significado de la responsabilidad.

Gracias a todas las personas que hicieron parte de mi proceso de formación profesional y aportaron un granito de arena para lograr el sueño anhelado.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



ÍNDICE

RESUMEN.....	10
ABSTRACT	11
CAPÍTULO I: INTRODUCCIÓN.....	12
1.1. JUSTIFICACIÓN.....	13
1.2. DIAGRAMA CAUSA Y EFECTO.....	14
1.3. DELIMITACIÓN	15
1.3.1. OBJETIVOS.....	15
CAPÍTULO II: MARCO INSTITUCIONAL	16
2.1. DIGITAL WARE	16
2.1.1. MISIÓN	16
2.1.2. VISIÓN.....	16
2.2. CARGO DESEMPEÑADO	18
CAPÍTULO III: MARCO REFERENCIAL.....	21
3.1. ESTADO DEL ARTE.....	21
CAPÍTULO IV: MARCO TEÓRICO.....	23
4.1. AUTOMATIZACIÓN DE LAS PRUEBAS	23
4.1.1. Niveles de Pruebas	24
4.1.2. Pruebas de caja negra	25
4.2.3. Pruebas funcionales.....	25
4.2.4. Pruebas no funcionales	26
4.2.5. Pruebas de caja blanca.....	26
4.2. EMULACIÓN VS VIRTUALIZACIÓN	26
4.3. VIRTUALIZACIÓN COMO PLATAFORMA	27
4.4. MÁQUINAS VIRTUALES	27
4.4.1. QEMU	28
4.5. CONTENEDORES.....	28



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



4.6. MÁQUINAS VIRTUALES VS CONTENEDORES	28
4.7. DOCKER	30
4.7.1. Dockerfile	30
4.8. BASH.....	31
4.9. WINAPPDRIVER	31
4.10. DEVOPS.....	31
4.10.1. Control de Versiones	32
4.10.2. Integración continúa	32
4.10.3. Despliegue continuo	32
4.11. AZURE DEVOPS	33
4.11.1. Pipeline.....	33
4.11.2. Agente	34
CAPÍTULO V: METODOLOGÍA.....	35
5.1. DESCRIPCIÓN Y CARACTERÍSTICA DEL LUGAR.....	36
5.2. TIPO DE METODOLOGÍA	36
5.3. DESARROLLO DE METODOLOGÍA POR FASES	36
5.3.1. FASE 1: Selección de todos los recursos necesarios para el proyecto ..	36
5.3.2. FASE 2: Proceso de virtualización y configuración	39
5.3.3. FASE 3: Implementación sistema de integración en Azure	58
5.3.2. FASE 4: Validación del sistema.....	80
CAPÍTULO VI: RESULTADOS	85
CAPÍTULO VII: CONCLUSIONES	98
REFERENCIAS.....	100



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



FIGURAS

Figura 1. Diagrama causa y efecto.	14
Figura 2. Representación máquinas virtuales. Fuente [16].	27
Figura 3. Diferencias entre máquina virtual y contenedor Docker. Fuente [18].	29
Figura 4. Metodología empleada en el desarrollo de este proyecto.	35
Figura 5. Diagrama funcionamiento del proyecto. [Elaboración Propia]	40
Figura 6. Verificación soporte KVM. [Elaboración Propia]	40
Figura 7. Instalación del gestor de máquinas virtuales. [Elaboración Propia]	41
Figura 8. Imágenes ISO. [Elaboración Propia]	42
Figura 9. Creación imagen del disco. [Elaboración propia]	42
Figura 10. Comando para ejecución del proceso de instalación de la máquina virtual. [Elaboración propia]	43
Figura 11. Primera script de bash creación máquina virtual. [Elaboración propia]	44
Figura 12. Continuación primer script de bash creación máquina virtual. [Elaboración propia]	45
Figura 13. Parte final primer script de bash creación máquina virtual. [Elaboración propia]	45
Figura 14. Primera parte ejecución Script 1. [Elaboración Propia]	46
Figura 15. Segunda parte ejecución Script 1. [Elaboración Propia]	47
Figura 16. Cargar controlador Virtio en la instalación del S.O. [Elaboración Propia]	48
Figura 17. Seleccionar controlador Virtio en la instalación del S.O. [Elaboración Propia]	48
Figura 18. Detección controlador SCSI Red Hat VirtIO. [Elaboración propia]	49
Figura 19. Inicio máquina virtual creada. [Elaboración propia]	49
Figura 20. Actualización del controlador Ethernet. [Elaboración propia]	50
Figura 21. Cargar driver para Ethernet desde el CDROM Virtio. [Elaboración propia]	51
Figura 22. Instalar controlador Ethernet. [Elaboración propia]	51
Figura 23. Conexión a internet máquina virtual. [Elaboración propia]	52
Figura 24. Activando modo desarrollador. [Elaboración Propia]	53
Figura 25. Desactivando Firewall en la máquina virtual. [Elaboración Propia]	53
Figura 26. Agregando a sitios de confianza en Internet Explorer URL de la página a automatizar. [Elaboración Propia]	54
Figura 27. Herramientas de compilación para .NET. [Elaboración Propia]	55
Figura 28. Creación snapshot de la máquina virtual. [Elaboración Propia]	55
Figura 29. Confirmación creación snapshot de la máquina virtual. [Elaboración Propia]	56
Figura 30. Ejecutando máquina virtual desde el Snapshot. [Elaboración Propia]	56
Figura 31. Segundo Script de bash creación snapshot. [Elaboración Propia]	56
Figura 32. Ejecución segundo Script de bash creación snapshot. [Elaboración Propia]	57
Figura 33. Estructura implementación sistema de integración. [Elaboración Propia]	58
Figura 34. Creando organización Azure. [Elaboración Propia]	59
Figura 35. Configuración organización. [Elaboración Propia]	59





ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Figura 36. Creación de nuevo proyecto dentro de Azure. [Elaboración Propia]	60
Figura 37. Pool de agente. [Elaboración Propia]	60
Figura 38. Creación grupo de agente. [Elaboración Propia]	61
Figura 39. Ruta de descarga del agente. Elaboración Propia]	61
Figura 40. Generación Personal Access Token. [Elaboración Propia]	62
Figura 41. PAT generado. [Elaboración Propia]	62
Figura 42. Agente de Azure descargado. [Elaboración Propia]	63
Figura 43. Configuración agente de Azure en máquina virtual. [Elaboración Propia]	63
Figura 44. Verificación agente instalado. [Elaboración Propia]	64
Figura 45. Sistema de control de versiones. [Elaboración Propia].	65
Figura 46. Clonar repositorio. [Elaboración Propia]	65
Figura 47. Administrar conexiones Visual Studio. [Elaboración Propia]	66
Figura 48. Ingreso del proyecto y la solución al repositorio local. [Elaboración Propia]	67
Figura 49. Visualizando solución en el Visual Studio. [Elaboración Propia]	67
Figura 50. Subir cambios al servidor de Azure. [Elaboración Propia]	68
Figura 51. Validando cambios en el repositorio de Azure. [Elaboración Propia]	68
Figura 52. Selección ubicación del código. [Elaboración Propia]	69
Figura 53. Selección del repositorio en Pipeline. [Elaboración Propia]	70
Figura 54. Generando plantilla .NET Desktop. [Elaboración Propia]	70
Figura 55. Selección del pool de agentes para el Pipeline. [Elaboración Propia]	71
Figura 56. Nombre de la tubería. [Elaboración Propia]	71
Figura 57. Configuración del Test Assemblies. [Elaboración Propia]	72
Figura 58. Guardando estado de la máquina virtual. [Elaboración Propia]	73
Figura 59. Archivo Dockerfile. [Elaboración Propia].	73
Figura 60. Script bash activación máquina virtual dentro de contenedor. [Elaboración propia]	74
Figura 61. Ingresando Dockerfile y entrypoint.sh para ejecución del contenedor. [Elaboración Propia]	75
Figura 62. Obteniendo imagen de Ubuntu. [Elaboración Propia]	75
Figura 63. Construyendo imagen del contenedor. [Elaboración Propia]	76
Figura 64. Ejecución del contenedor. [Elaboración propia]	76
Figura 65. Tercer Script automatizado ejecutar snapshot. [Elaboración Propia]	77
Figura 66. Continuación Tercer Script automatizado ejecutar snapshot. [Elaboración Propia]	78
Figura 67. Ejecución tercer script de bash para el proceso de virtualización. [Elaboración Propia]	78
Figura 68. Ejecución script container. [Elaboración Propia]	79
Figura 69. Ejecución cuarto script del proceso de containerización. [Elaboración Propia].	79
Figura 70. Método para el proceso de login. [Elaboración Propia]	82
Figura 71. Método de prueba AbrirConsultoría. [Elaboración Propia]	81
Figura 72. Continuación método para el proceso de login. [Elaboración Propia]	82
Figura 73. Traer caso desde base de datos. [Elaboración Propia]	83
Figura 74. Descargar caso desde el maestro. [Elaboración Propia]	84



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Figura 75. Agente Online. [Elaboración Propia]	87
Figura 76. Imagen de los contenedores. [Elaboración Propia]	88
Figura 77. Lanzamiento dos contenedores. [Elaboración Propia]	89
Figura 78. Ejecución Pipeline 1. [Elaboración Propia]	90
Figura 79. Ejecución Pipeline 2. [Elaboración Propia]	91
Figura 80. Visualización de capturas del proceso. [Elaboración Propia]	92
Figura 81. Lanzamiento 4 contenedores al tiempo. [Elaboración Propia]	93
Figura 82. Agentes encendidos prueba contenedores. [Elaboración Propia]	94
Figura 83. Ejecución Pipeline en paralelo. [Elaboración Propia]	95
Figura 84. Resultado final Pipelines en ejecución. [Elaboración Propia]	95
Figura 85. Porcentaje de consumo CPU. [Elaboración Propia]	96
Figura 86. Consumo de CPU y memoria modo gráfico con contenedores en ejecución. [Elaboración Propia]	96
Figura 87. Consumo de CPU y memoria modo gráfico con contenedores detenidos. [Elaboración Propia] 97	

TABLAS

Tabla 1. Trabajos consultados como base de la investigación.....	21
Tabla 2. Herramientas necesarias para las máquinas.	49



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



RESUMEN

En el presente trabajo se expone el desarrollo de un sistema de integración de automatización y distribución de pruebas que permite ejecutar pruebas utilizando las tecnologías Microsoft Azure y Docker, dando solución a dos de las principales problemáticas existentes actualmente en el área de automatización de Digital Ware y estas son el tiempo de ejecución de las pruebas y el requerimiento de una pantalla de escritorio activa para ejecutarlas. Lo que se plantea es el despliegue de varios contenedores Docker que posibiliten la ejecución de pruebas aplicando el paralelismo dentro de un mismo dispositivo físico, cada uno de los contenedores Docker contiene una máquina virtual que será configurada con su respectivo agente de prueba y herramientas necesarias para la ejecución de los test de manera remota, liberando las máquinas de desarrollo del área de automatización de la compañía y permitiendo que corran todas las pruebas dentro de un solo equipo servidor en los diferentes contenedores, siendo el medio de integración la plataforma Azure. Al finalizar el proyecto se logró la ejecución de pruebas desplegadas desde Azure dentro de los contenedores Docker, llevando a cabo una ejecución sin la renderización de la capa de aplicación y directamente dentro de un mismo equipo de cómputo.

Palabras Claves: *Automatización de pruebas, Docker, Máquinas virtuales, Microsoft Azure, Virtualización.*



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



ABSTRACT

In this work, the development of a test automation and distribution integration system is exposed that allows tests to be executed using Microsoft Azure and Docker technologies, solving two of the main problems currently existing in the area of Digital Ware automation and these are the execution time of the tests and the requirement of an active desktop screen to run them. What is proposed is the deployment of several Docker containers that allow the execution of tests applying parallelism within the same physical device, each of the Docker containers contains a virtual machine that will be configured with its respective test agent and tools necessary for the remote execution of the tests, freeing the development machines from the company's automation area and allowing all the tests to be executed within a single server in the different containers, the Azure platform being the means of integration. At the end of the project, the execution of tests deployed from Azure within the Docker containers was achieved, carrying out an execution without the rendering of the application layer and directly within the same computer equipment.

Keywords: *Test automation, Docker, Virtual machines, Microsoft Azure, Virtualization.*



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



CAPÍTULO I: INTRODUCCIÓN

En un campo tan amplio como lo es el desarrollo de software profesional y las pruebas de software existen tres grandes problemas que van a estar presentes en cualquier tipo de tecnología o en cualquier aplicación, estos se pueden resumir en: construir, distribuir y ejecutar, básicamente es cómo armamos el software o las pruebas que se van a aplicar a ese software, cómo lo llevamos, dónde tiene que estar para que le sirva a los usuarios, dónde se tiene que llevar para que las pruebas se puedan ejecutar y finalmente de qué manera hacemos para que ese software o prueba se ejecute/corra de la manera en que esperamos teniendo en cuenta el lugar específico donde lo tiene que hacer.

Actualmente la solución a estos problemas la industria de software utiliza la virtualización y containerización, en primer lugar, la virtualización se conoce como las máquinas virtuales que consisten en crear una versión virtual ya sea de un sistema operativo o máquina y son ampliamente utilizadas, en segundo lugar, los contenedores permiten construir y desplegar software de una forma rápida y sencilla.

Para el desarrollo del presente trabajo se usa máquinas virtuales, contenedores Docker y la plataforma Microsoft Azure para el proceso de integración y despliegue de las pruebas de software aprovechando los beneficios que brindan estas herramientas y dando la posibilidad de una mayor productividad en el equipo de trabajo del área de automatización ya que permite liberar las máquinas al instante de la ejecución y el despliegue de dichas pruebas.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



PROBLEMA

1.1. JUSTIFICACIÓN

Actualmente en un mercado tan competitivo solamente los productos de calidad son los que sobreviven; la calidad depende directamente de la filosofía de la empresa la cual busca ofrecer productos y servicios que logren satisfacer las necesidades de sus clientes, por esta razón es de suma importancia llevar a cabo el testing de las aplicaciones por medio de herramientas avanzadas y personalizadas para la automatización y la realización de pruebas de software permitiendo que los errores no se acumulen en la fase final de la entrega del producto.

Digital Ware es una empresa que brinda diferentes soluciones integrales en diversos sectores tales como financiero, salud, educación, gobierno, etc., por medio de diferentes productos y servicios como Kactus-HCM, Seven-ERP, Ophelia-BPM, entre otros, permitiendo cubrir distintas problemáticas existentes dentro del mercado. Uno de los productos principales más vendidos dentro de la compañía es Kactus-HCM que permite la administración de todas las áreas de los recursos humanos [1], este producto cuenta con más de 2000 programas empresariales no webs renderizados en Internet Explorer.

La identificación de errores existentes dentro de cada uno de estos programas es una prioridad, esto se lleva a cabo en el equipo de automatización. Actualmente en el área de automatización las pruebas desarrolladas son programadas para después ser enviadas a un servidor en grandes cantidades las cuales se van ejecutando una a una, este modo de funcionamiento cumple a cabalidad con el objetivo pero existen diversos problemas que influyen directamente, uno de los principales es el tiempo de ejecución sumado con la no productividad al momento de ejecutarlas ya que estas requieren de una pantalla de escritorio activa, por tanto, al momento en que corren las pruebas no es posible seguir trabajando en otras cosas importantes de forma paralela, esto conlleva a un retraso significativo en la entrega de resultados y por ende la aplicación nueva o actualizada demora un poco más en llegar a producción.

Por esta razón se implementa un sistema de integración de automatización y distribución de pruebas basado en Microsoft Azure que permite incrementar la eficiencia al momento de ejecución de pruebas automatizadas a nivel de servidor, por medio de



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750

la implementación de un nuevo servidor que se basa en contenedores con una interfaz gráfica virtualizada, cada uno de estos contenedores se va a ejecutar de forma paralela dentro de esta máquina y dentro de cada uno de los contenedores se estará ejecutando pruebas automatizadas permitiendo la optimización de tiempo, mayor productividad y eficiencia dentro del equipo de trabajo de automatización de Digital Ware.

1.2. DIAGRAMA CAUSA Y EFECTO

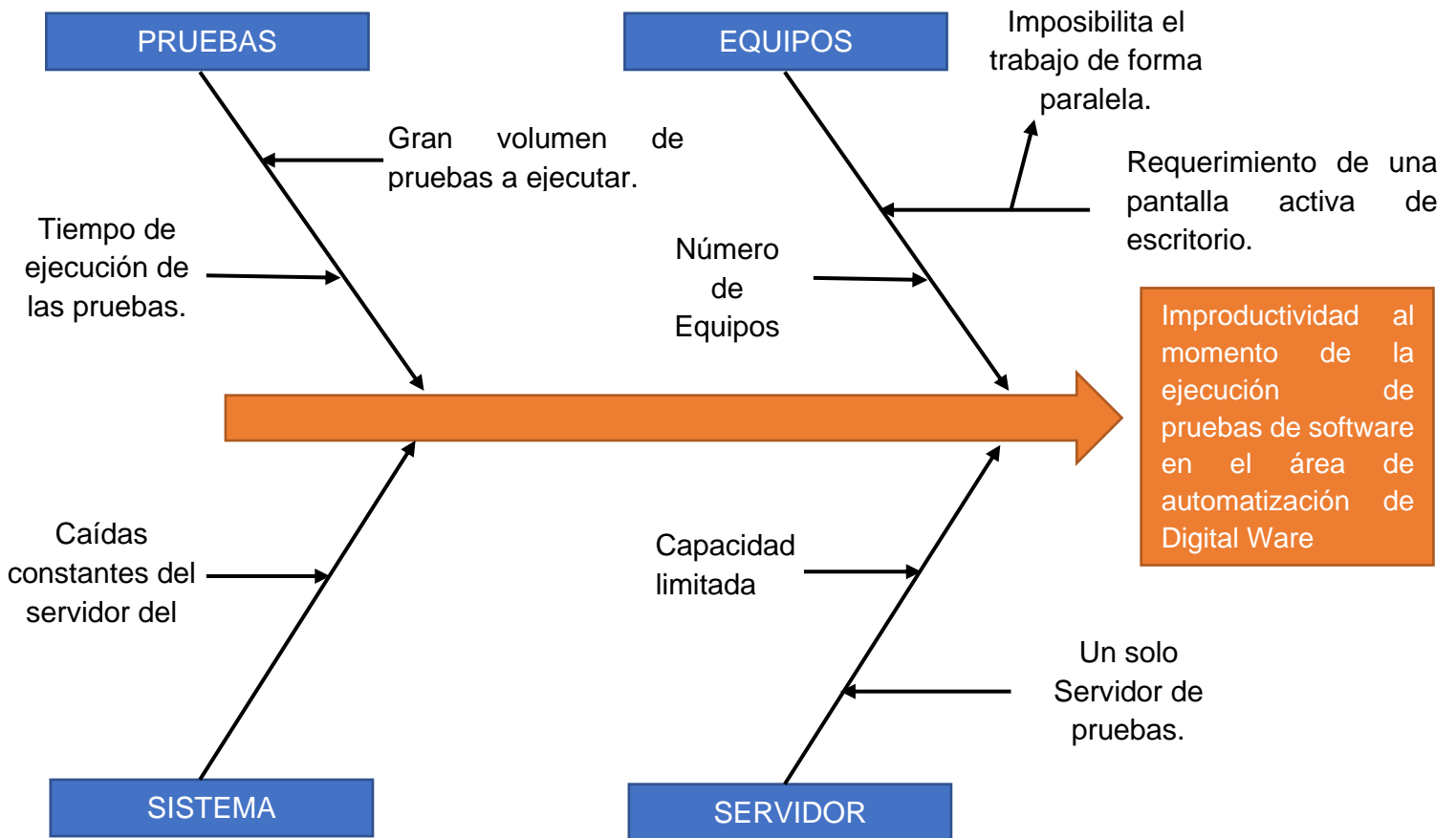


Figura 1. Diagrama causa y efecto.



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



1.3. DELIMITACIÓN

1.3.1. OBJETIVOS

1.3.1.1. OBJETIVO GENERAL

- Implementar un sistema de integración de automatización y distribución de pruebas, utilizando Microsoft Azure y contenedores Docker.

1.3.1.2. OBJETIVOS ESPECIFICOS

- Llevar a cabo la definición de los distintos recursos tecnológicos necesarios basado en un apropiamiento teórico inherente a la temática propuesta en el presente proyecto.
- Realizar la etapa de virtualización y configuración de los agentes de prueba y dependencias necesarias para la automatización.
- Implementar el sistema de integración y la distribución de pruebas.
- Validar el sistema de integración con un experto en el tema por medio de la ejecución de las pruebas automatizadas mediante paralelismo.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



CAPÍTULO II: MARCO INSTITUCIONAL

La práctica empresarial fue desarrollada en la empresa DIGITAL WARE de forma remota desde el Municipio de Chinácota con una duración de 6 meses, tiempo ocupado entre el 04 de enero de 2021 hasta el 4 de julio de 2021

2.1. DIGITAL WARE

Según la página oficial Digital Ware es una empresa del sector tecnológico que se especializa en Software ERP, Software de Nómina y Gestión Humana, y software para IPS y Clínicas, cuenta con más de 25 años en el mercado y es líder en el diseño e implantación de diferentes soluciones empresariales en áreas como finanzas, logística, manufactura, seguridad, cajas de compensación, gobierno, educación, salud, entre otras [2].

Las soluciones tecnológicas de Digital Ware están presentes en más de 10 países de Latinoamérica, con más de 1000 organizaciones de las más reconocidas e importantes de Colombia y Latinoamérica, cuenta con más de 11 premios y reconocimientos nacionales e internacionales por su innovación y aporte a la economía del país, actualmente ha impactado de forma positiva a más de 20 millones de personas con la tecnología brindada [3].

2.1.1. MISIÓN

Mejorar la vida de las personas y ayudar a las empresas a cumplir sus metas, utilizando el conocimiento, el talento y la tecnología, soportados en visión, pasión, conciencia, disciplina e intensidad [3].

2.1.2. VISIÓN

Nos convertiremos en la más innovadora, motivadora y servicial empresa de la tecnología en américa latina en el desarrollo de plataformas informáticas con foco en:



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



ERP: Enterprise Resource Plannig, planificación de recursos empresariales. **HR:** Human Resources - Recursos Humanos. **HIS:** Hospital Information System - Sistemas de Información Hospitalario. **BPM:** Business Process Management- Gestión por Procesos Inteligentes [3].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



2.2. CARGO DESEMPEÑADO

El cargo desempeñado durante el proceso de pasantía en la empresa Digital Ware fue de desarrollador de pruebas automatizadas, el objetivo principal a llevar a cabo dentro del equipo de trabajo era el de automatizar pruebas funcionales del producto KACTUS-HCM, inicialmente en el proceso de aprendizaje y adaptación al equipo de trabajo se inició un proceso de realización de pruebas funcionales en una aplicación híbrida la cual contiene tecnología web y de escritorio llamada Ophelia, estas pruebas buscaban testear funciones principales de esa aplicación como la realización de CRUDs que consiste en poder agregar, editar, cancelar y eliminar datos de un programa de forma correcta, la generación de reportes PDF y Excel, entre otras funcionalidades.

Estas pruebas se desarrollaron en el lenguaje C# junto con el framework de automatización Appium en el entorno de desarrollo Visual Studio Community 2019. Después de un tiempo y una adaptación clara en el proceso de automatización de maestros (programas) en Ophelia, se desarrolló la primera tarea a gran escala que llevaba por nombre RolAuditor la cual consistía en crear una sola Función global que permitiera generar todos los reportes preliminares diferentes que existían en cada uno de los maestros, aplicando el concepto de recursividad, esta función tuvo aproximadamente 500 líneas de código y sirvió para la ejecución de pruebas de aproximadamente 2000 programas y se sigue extendiendo. Teniendo un gran bagaje en el concepto de automatización fue asignada la tarea de creación de dos funcionalidades generales y principales que se deben probar en las diferentes ejecuciones, estas son la captura del Código del programa y Nombre del programa, funcionalidades que no habían podido captura en algún tiempo pero que con un gran análisis y lógica se pudo llevar a cabo de una forma sencilla.

En base a lo anterior sucedieron muchos cambios dentro del equipo de trabajo con respecto a los compañeros de equipo y debido a que la automatización en la parte web había quedado un poco atrás comenzó el proceso de desarrollo de pruebas de automatización en web en los sitios Smart People y Reclutamiento, estas pruebas se desarrollaron en el lenguaje de programación C# con el framework de automatización Selenium en el entorno de desarrollo Visual Studio 2017.

En este punto de la pasantía quedamos junto con un compañero (el cual se encuentra realizando prácticas) como los ingenieros con más experiencias y como coordinadores



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



en el equipo de trabajo a pesar de la salida de algunos compañeros, ingresó nuevo personal y se desarrolló el proceso de capacitar a los nuevos compañeros que iban a ser parte del equipo de trabajo, de la misma manera se dio la oportunidad de llevar a cabo pruebas de ingreso a la compañía para una vacante en el equipo de trabajo, debido a la nueva responsabilidad fueron asignadas nuevas tareas más grandes, con mayor peso como el desarrollo de pruebas de automatización para procesos de liquidación de vacaciones, al mismo tiempo se logró coordinar diversos lanzamientos de pruebas de abrir programa, SmokeTest, Checklist que se ejecutan aproximadamente 2000 pruebas en cada lanzamiento como la creación de diferentes planes de prueba, se implementó una nueva idea para el desarrollo de procesos que buscaba la disminución de líneas de código y el tiempo de ejecución en los procesos de validación de los datos, logrando un gran desempeño en dichos procesos dando como la reducción de hasta 2 semanas de trabajo por parte de los ingenieros; se obtuvo la responsabilidad de dirigir un equipo de trabajo y buscar las maneras más adecuadas para que todos los ingenieros cumplan con sus respectivas tareas.

Todos los planes de trabajo para las diferentes pruebas de automatización son establecidos según los requerimientos del área de calidad los cuales son los clientes directos del equipo de automatización, ellos son los encargados de dar las pautas, así mismo los pasos de las funcionalidades a probar y como debe funcionar la aplicación, en base a eso se desarrolla el código para la ejecución de pruebas en los distintos programas.

Todos los errores encontrados durante un proceso de lanzamiento de pruebas son reportados a calidad o desarrollo según sea el caso para que puedan ser corregidos antes de que la aplicación salga a producción, como finalidad el área de automatización es el que tiene la última palabra y la decisión de liberar a producción la aplicación que se está probando.

Para un seguimiento claro de las distintas actividades desarrolladas por cada trabajador de la compañía se aplica la metodología SCRUM dentro de cada área de la compañía la cual consiste en realizar una pequeña reunión no muy larga aproximadamente de 15 a 20 minutos, en donde todas las personas comentan las tareas que realizaron el día de ayer, que problemas tuvo y cuales tareas va a llevar a cabo el día de hoy esto para tener una visual general de lo que se realizó para poder solucionar los inconvenientes que pueden afectar el flujo de trabajo.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



A nivel de resumen se llevaron a cabo dichas actividades nombradas anteriormente, de forma paralela se ejecutó la investigación, la implementación de la integración, distribución de pruebas con Azure y contenedores Docker que permitiera ejecutar las pruebas de automatización dentro de máquinas virtuales que se encontraban en contenedores Docker, proyecto que al implementarlo cuando se obtengan los recursos necesarios permitirá liberar las máquinas de desarrollo de los ingeniero logrando una mayor productividad.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



CAPÍTULO III: MARCO REFERENCIAL

3.1. ESTADO DEL ARTE

El siguiente estado del arte muestra algunos de los estudios realizados por estudiantes en la Universidad de Pamplona en la empresa Digital Ware, los cuales han sido de gran ayuda al desarrollo del presente proyecto y han puesto las bases para que se pueda llevar a cabo.

AÑO	AUTOR	TRABAJO
2019	Jesús Leonardo García Hoyos	Migración del software KACTUS-HCM de una arquitectura cliente-servidor a una arquitectura cliente-contenedor
2020	Juan Felipe Gómez Castillo	Desarrollo de demo portable y automatizado para las aplicaciones KACTUS-HCM fundamentado en contenedores sobre Docker para la empresa Digital Ware
2020	Wilmar Daniel Bustos Morales	Implementación de servidor de pruebas a partir de contenedores Docker y virtualización anidada en la empresa Digital Ware

Tabla 1. Trabajos consultados como base de la investigación.

En primera instancia García Jesús propone una mejora en un producto de la compañía Digital Ware la cual consiste en cambiar el servidor físico o virtualizado que se está manejando en una arquitectura cliente-servidor por un contenedor de software el cual contenga todas las herramientas necesarias para realizar un despliegue sencillo de la aplicación, en pocas palabras como su nombre lo indica se lleva a cabo un proceso de migración del software Kactus HCM de una arquitectura cliente-servidor a una



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



arquitectura cliente-contenedor permitiendo así una disminución de costos a nivel de infraestructura y una disminución de los tiempos de despliegue de la aplicación [4].

Así mismo el autor Gómez Felipe En el año 2020 con el fin de tener una nueva alternativa de la elaboración de los Demos para las aplicaciones Kactus-HCM que ofrece la empresa Digital Ware, realiza la implementación de contenedores Docker para poder aprovechar de una mejor manera los distintos recursos de la maquina anfitrión que lo contiene. Este trabajo desarrollado buscaba que todos los servicios implementados en el Demo sean portables para permitir una actualización de ellos de forma más sencilla y al mismo tiempo que al momento que el cliente requiera el producto no sea necesario que este le entregue la máquina de trabajo a la persona encargada de instalar los Demos [5].

Para ese mismo año Bustos Daniel dentro del área de automatización de la empresa Digital Ware desarrolló el proyecto que consistía en la creación de contenedores Docker que permitieran la ejecución en paralelo de pruebas de automatización, cada contenedor llevaba a cabo en su interior un proceso de virtualización anidada, el contenedor ejecutaba un snapshot de una máquina virtual del sistema operativo Windows 10 y permitía dar acceso al driver de Internet Explorer posibilitando la ejecución en memoria de pruebas lanzadas a través del mismo dispositivo que estaban corriendo los contenedores [6].

A nivel externo En el año 2018 Gonzales Sergio en México llevó a cabo un proyecto de automatización de pruebas para validar la funcionalidad de una web de ejemplo, estas pruebas fueron ejecutadas de forma paralela por diferentes sistemas operativos y al mismo tiempo diferentes navegadores, embebidos dentro de un mismo equipo con sistema operativo Windows, para poder ejecutar de forma paralela sobre un mismo equipo de escritorio fue necesario apoyarse en containerización y virtualización las cuales fueron utilizadas las tecnologías Docker e Hyper-V; para el proceso de automatización web fue utilizada la librería Selenium, esta solución permitió disponer de un entorno multidispositivo con un bajo coste y con el potencial de poder ejecutar diversas pruebas en un mayor número de máquinas de forma paralela [7].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



CAPÍTULO IV: MARCO TEÓRICO

Como se pudo observar anteriormente el presente proyecto consiste en realizar el despliegue y distribución de pruebas de automatización dentro de contenedores Docker usando Microsoft Azure. Contextualizando, se hace necesario comenzar a definir los elementos más importantes a trabajar en el presente proyecto de una forma sencilla y directa.

Actualmente todos los sistemas de software son consumidos por millones de clientes los cuales han tenido diversas experiencias con dichos productos, en muchos casos esa experiencia de la persona con algún producto de software no es muy buena ya que no funciona como lo esperaba y esto puede causar muchos problemas como lo es la pérdida de dinero, tiempo o reputación; Es por esta razón que se hace necesario probar el software y esto es una forma de evaluar esa calidad que ofrece dicho software, al mismo tiempo permite reducir en gran cantidad el riesgo de fallos tanto en operaciones como en producción.

4.1. AUTOMATIZACIÓN DE LAS PRUEBAS

Para poder comenzar a hablar de una forma más detallada sobre la automatización de las pruebas es de suma importancia comprender que representa una prueba de software; estas pruebas se definen como “el proceso de ejecución de un programa con la intención de encontrar errores” [8] el cual busca llevar a cabo un proceso de evaluación de los diferentes productos, al mismo tiempo verificar que se cumplan todos los requisitos especificados, encontrar diferentes fallos y defectos del producto permitiendo así una mejora directa en la calidad.

En la mayoría de productos en los cuales se realiza un proceso de aplicación de pruebas se hace extenso llevarlas a cabo de forma manual, en consecuencia, conllevaría el gasto de un sinnúmero de recursos y lo más importante que es el tiempo debido al volumen de pruebas a realizar.



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Es por esta razón que en el ámbito de las pruebas de software se implementó lo que se conoce como la automatización lo cual se encarga de buscar pruebas que puedan ser ejecutadas de forma desatenida, es decir, sin intervención humana a través de alguna herramienta que permita realizar el proceso automáticamente, esto aportara más velocidad, eficiencia, precisión y resistencia en el proceso de ejecución de las pruebas logrando un incremento en la velocidad de ejecución de las pruebas [9].

4.1.1. Niveles de Pruebas

Existen diferentes tipos de niveles de pruebas, donde cada uno de ellos busca un objetivo en específico, actuando de acuerdo al software en un nivel característico desde unidades sencillas o componentes individuales hasta sistemas completos [10].

- Prueba de componente
- Prueba de integración
- Prueba de sistema
- Prueba de aceptación

4.1.1.1. Pruebas de componente

Son también conocidas como pruebas unitarias o de módulo las cuales estas se basan en todos los componentes que se pueden probar por separado, por lo general se realizan de forma aislada al resto del sistema, estas buscan probar que una unidad de código como una función o un método de una clase este ejecutándose de forma correcta y son realizadas generalmente por desarrolladores [9].

4.1.1.2. Pruebas de integración

Las pruebas de integración trabajan un poco más a profundidad, estas se basan en todas las interacciones entre componentes o sistemas, es decir, puede que cada uno de los componentes de un producto funcionen correctamente, pero al momento de juntar los diferentes componentes pueden generar o causar un error que no se tenía presente, este tipo de pruebas se centran en bases de datos, infraestructura, interfaces, API, microservicios, entre otros [9].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



4.1.2.3. Pruebas de sistema

Las pruebas de sistemas se fundamentan directamente en el comportamiento y las capacidades de todo un sistema o producto, esta prueba en su mayoría de casos es la prueba final en nombre del desarrollo para verificar que el sistema que se va a entregar cumpla con todas las especificaciones, estas pruebas deben estudiar todos los requerimientos funcionales y no funcionales de todo el sistema [9].

4.1.2.4. Pruebas de aceptación

Después de haber realizado todos los procedimientos de pruebas anteriores y se hayan corregido todos o la gran mayoría de defectos, el sistema es entregado al cliente final para que se lleve a cabo el proceso de prueba de aceptación y este permite la preparación del sistema para el despliegue final por parte del cliente, en pocas palabras el software se entrega al cliente para que este realice sus pruebas las cuales son las mismas que se realizaron en el sistema, teniendo en cuenta de que se realizan en el entorno del cliente [9].

Actualmente existen diferentes tipos de pruebas de automatización, donde cada una de ellas se basa en probar específicamente cierta característica de un sistema de software o una parte del sistema, algunas de las más importantes son:

4.1.2. Pruebas de caja negra

Para la ejecución de este tipo de pruebas no se requiere un conocimiento en profundidad de la manera en cómo trabaja a nivel interno una aplicación de software, sino que la persona encargada de realizar estos test únicamente conoce las entradas que este debe introducir y cuáles son sus respectivas salidas que debe esperar. Este tipo de pruebas van de la mano con las pruebas funcionales de un programa debido a que no es necesario tener un conocimiento en profundidad de la aplicación para que se puedan ejecutar [11].

4.2.3. Pruebas funcionales

Al aplicar pruebas funcionales sobre un sistema estas van a permitir evaluar cada una de las funciones que el sistema debe realizar, por lo tanto, estas funciones van a permitir describir “que” debe hacer el producto, en pocas palabras se puede decir que las



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



pruebas funcionales permiten verificar que la aplicación de software lleve a cabo todos los objetivos para la cual fue desarrollada a nivel funcional.

Este tipo de pruebas también se pueden denominar pruebas de caja negra ya que cualquier persona que se encargue de aplicar este tipo de pruebas en un producto no se enfoca en la manera en que se generan las respuestas del sistema, sino que se basa en analizar los datos de entrada, salida y que el producto o software haga lo que tenga que hacer [11].

4.2.4. Pruebas no funcionales

Al aplicar pruebas no funcionales sobre un sistema van a permitir evaluar las características de sistemas de software como lo es la usabilidad, cual es la eficiencia del desempeño del software, la seguridad, rendimiento, exactitud, velocidad y confiabilidad, estas pruebas van a permitir probar que tan bien se va a comportar el sistema [12].

4.2.5. Pruebas de caja blanca

La aplicación de este tipo de pruebas se basan directamente en la estructura interna del sistema o la implementación internamente del software, es decir, que se fundamente directamente en la lógica interna del producto y al mismo tiempo de la estructura del código de la aplicación, para llevar a cabo este tipo de pruebas se hace necesario tener un conocimiento claro sobre el funcionamiento interno del producto como el código, arquitectura, flujo de trabajo o datos, entre otros [13].

4.2. EMULACIÓN VS VIRTUALIZACIÓN

Es necesario tener muy claro dos conceptos que suenan parecido pero que realmente son conceptos totalmente diferentes, estos son la emulación y la virtualización, la primera de ellas va a permitir como su nombre lo dice emular una arquitectura de hardware dentro de la arquitectura en la cual se está ejecutando, a diferencia de la virtualización que permite emplear distintos dispositivos de hardware en ambientes de ejecución que son independientes; estos son controlados por VMM [14].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



4.3. VIRTUALIZACIÓN COMO PLATAFORMA

Actualmente el proceso de virtualización de recursos es uno de los más importantes y es uno de los conceptos claves en el campo empresarial, este hace referencia a la creación de una versión virtual de algo en vez de una versión real, sin necesidad de limitar a una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento o diversos recursos de una red informática.

La virtualización utiliza una capa intermedia existente dentro del software para así proporcionar abstracción de múltiples recursos virtuales. Actualmente existen 2 grandes grupos de modelos de virtualización: Las máquinas virtuales y los contenedores.

4.4. MÁQUINAS VIRTUALES

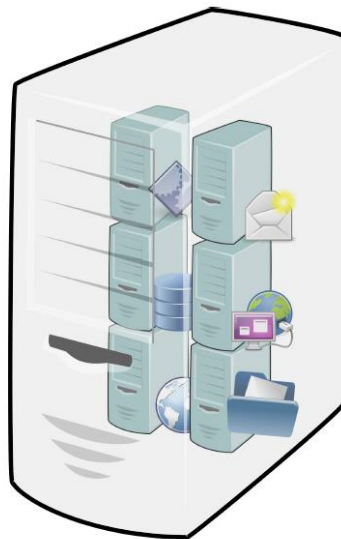


Figura 2. Representación máquinas virtuales. Fuente [16]

Las Máquinas virtuales conocidas por sus siglas VM son componentes de software de recursos virtuales, los cuales pueden ser descritos como contextos de ejecución aislados. Dentro del mercado existen diferentes técnicas de virtualización, una de las más importantes y más populares es la virtualización basada en Hypervisor, el cual requiere de un Virtual Machine Monitor (VMM) ejecutándose sobre un sistema operativo anfitrión para así proporcionar una abstracción completa que permite ejecutar diversos sistemas operativos en un solo anfitrión real. Algunos de los ejemplos de virtualización



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



mediante Hypervisor incluyen herramientas como XEN, KVM, QEMU, Virtual Box y VM-Ware [15].

4.4.1. QEMU

QEMU es un emulador de máquinas virtuales libre, multiplataforma el cual es capaz de soportar la emulación y la virtualización de hardware, este emulador utiliza un traductor binario dinámico el cual permite convertir en tiempo de ejecución las instrucciones de la arquitectura emulada en instrucciones de la arquitectura sobre la cual ejecuta el emulador, QEMU hace el uso del módulo KVM (Kernel-based Virtual Machine) del kernel de Linux actuando como Hypervisor para el control de las diferentes máquinas virtuales. Actualmente QEMU soporta cualquier tipo de arquitectura de hardware a diferencia de otros emuladores que soportan específicamente ciertas arquitecturas [14].

4.5. CONTENEDORES

Cuando se habla de contenedores se puede relacionar directamente con los containers que se utilizan para el transporte de mercancía por medio de los barcos de un lugar a otro, si se observa de una forma más detallada la estructura general de cada uno de esos contenedores se puede decir que se caracterizan por ser estandarizados ya que poseen siempre el mismo tamaño y la misma forma, solamente cambia lo que lleva dentro, algo muy parecido sucede con los contenedores de software ya que dentro de ellos podemos introducir o mantener todas las herramientas o dependencias que una aplicación necesite para ser ejecutado, como código, librerías, dependencias, entre otros.

Estos siempre se mantienen aislados para que puedan ser ejecutados en cualquier lugar, en pocas palabras un contenedor representa una forma o manera de empaquetamiento lógico donde contiene todo lo que necesita una aplicación para que se pueda ejecutar y se lleve a cabo de forma independiente del sistema operativo permitiendo maximizar su portabilidad [17].

4.6. MÁQUINAS VIRTUALES VS CONTENEDORES

En la mayoría de casos se tiende a confundir un poco los términos de virtualización y contenedores; en primer lugar, la virtualización permite tener distintas máquinas virtuales con su propio sistema operativo invitado, ejecutándose sobre un mismo



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



sistema operativo anfitrión y es una práctica ampliamente utilizada para el alojamiento de distintas aplicaciones, por otro lado, los contenedores se ejecutan directamente sobre el sistema operativo anfitrión de una forma aislada, pero no se hace necesario un propio sistema operativo ya que estos se ejecutan directamente sobre el Kernel de la máquina, es por esta razón que son mucho más ligeros.

Un contenedor se caracteriza por que es mucho más liviano que una máquina virtual, ya que la máquina virtual tiene que emular internamente todo un sistema operativo y todo lo que se requiere para su respectivo arranque y funcionamiento, a diferencia de los contenedores que permiten aislar diferentes procesos dentro del mismo sistema operativo anfitrión y ejecutarlo cuando sea necesario o se requiera, permitiendo una mayor escalabilidad y rendimiento [19].

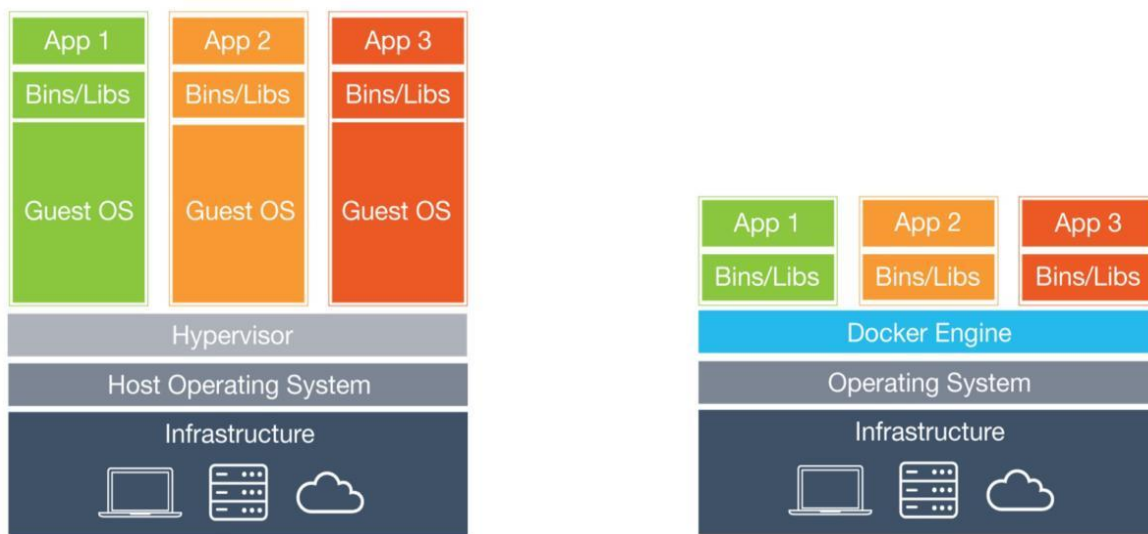


Figura 3. Diferencias entre máquina virtual y contenedor Docker. Fuente [18].

Teniendo en cuenta la imagen anterior podemos observar que tanto en las máquinas virtuales como en los contenedores existe una capa de infraestructura de hardware el cual es la parte física de toda máquina o computadora, seguida de ella continua la capa del sistema operativo anfitrión, esta se encarga de administrar los diferentes componentes con el hardware basándose directamente en las máquinas virtuales, se puede observar que existe un capa de Hypervisor; esto es una herramienta que permite



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



ejecutar esas máquinas virtuales en un equipo donde cada una de ellas contiene su respectivo sistema operativo con sus respectivas librerías o runtimes que se necesitan para que puedan funcionar, por último, se encuentran las aplicaciones que serán las que podrá observar el usuario final, a diferencia de la arquitectura de los contenedores está después de la capa de sistema operativo, este contiene el ejecutor de contenedores permitiendo el despliegue de varios contenedores al mismo tiempo que se caracterizan porque no poseen un sistema operativo virtualizado propio sino que estos cuentan directamente con sus propias bibliotecas o librerías para que la aplicación se pueda ejecutar o desplegar y puedan ser observadas por el usuario final [19].

4.7. DOCKER

Docker es una plataforma utilizada para la virtualización de nivel de sistema operativo, este permite empaquetar una aplicación junto con todas sus librerías y dependencias dentro de un contenedor y posteriormente sea posible ejecutarla en cualquier otra máquina. Docker permite un modelo de implementación el cual se basa en imágenes permitiendo compartir diferentes aplicaciones o servicios en varios entornos [20].

Docker permite desplegar; desarrollar; ejecutar aplicaciones empaquetadas las cuales se basan en imágenes y contenedores.

Una imagen Docker es un binario el cual contiene todos los componentes requeridos para ejecutar un contenedor Docker, mientras que un contenedor es una instancia de una imagen en tiempo de ejecución, por tanto, tiene un estado. Un contenedor al momento que se crea basado en una imagen se puede encender o apagar en cualquier momento [20].

4.7.1. Dockerfile

Dockerfile es un fichero o documento de texto el cual contiene las diferentes reglas o comandos que se necesitan para crear o definir una imagen Docker, estas imágenes de construyen de forma automática leyendo todas las instrucciones dadas en el Dockerfile [20].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



4.8. BASH

Bash representa la versión libre desarrollada por el proyecto GNU, es el intérprete de comandos Shell de los sistemas operativos que se basan en Kernel Linux, su función es trabajar directamente como intermediario entre el usuario y el núcleo de la máquina para que se pueda ejecutar lo que el usuario desea, por tanto, se asemeja a una interfaz donde le usuario ingresa comandos que son interpretados por la Shell y son enviados al kernel de la máquina para que realice las respectivas operaciones [21].

4.9. WINAPPDRIVER

WinAppDriver es la abreviatura de Windows Application Driver, es una herramienta de estándar abierto utilizado para el proceso de automatización de pruebas, esta aplicación permite utilizar Appium para probar diferentes aplicaciones de Windows permitiendo probar tanto en una máquina local como en cualquier máquina remota, se requiere de esta herramienta abierta para poder ejecutar una prueba de automatización con Appium [22].

4.10. DEVOPS

DevOps posee una serie de elementos principales que se basan directamente en las personas las cuales son los usuarios finales o desarrolladores o personas involucradas dentro de una institución, grosso modo los procesos son aquellos procesos de desarrollo de software y no solamente a nivel programático sino también de despliegue, monitorización, configuración, entre otros.

Los productos, valga la redundancia, son todos los productos finales que se construyen dentro de una compañía DevOps, es la unión existente entre personas, procesos y productos, su nombre proviene de Developers y Operations, el cual es el conjunto de ambas prácticas que permite construir software de una manera excelente, más sencilla y con mayor calidad desde su planeación hasta su despliegue [23].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



4.10.1. Control de Versiones

El control de versiones es un sistema que permite administrar cambios en los diferentes archivos en cualquier tipo de binario a través del tiempo como archivos de código fuente, documentos, entre otros.

Con el sistema de control de versiones es posible determinar quién realizó un cambio y en qué momento ya que estos cambios son guardados en una base de datos, al mismo tiempo es posible comparar las versiones de un archivo a través del tiempo, es decir, comparar el estado actual de un archivo con una versión del pasado para observar que había antes de la versión actual en el equipo, al mismo tiempo brinda el beneficio de restaurar archivos es decir traerlos del pasado al presente rescatando lo que se realizó en su momento y permite sobrellevar una colaboración directa entre todos los miembros de un equipo de trabajo [24].

4.10.2. Integración continúa

CI por sus siglas en inglés Continuous Integration es el proceso que permite integrar nuevo código de una nueva rama principal que es compartida por todos los desarrolladores, este proceso de integración permitirá automatizar el proceso de compilación y pruebas continuas o automáticas de código para asegurar que todo funcione correctamente. La intención de la integración continua es que cada vez que un desarrollador suba cambios al repositorio el proceso de integración continua se dispare para obtener todo el repositorio con los últimos cambios cargados, compilados y ejecutados en pruebas de forma automática permitiendo detectar de forma temprana los errores del código, reduciendo errores de integración permitiendo entregar software de mejor calidad y mucho más rápido [25].

4.10.3. Despliegue continuo

Construir software que no llegue a las manos de los usuarios o que llegue de manera tardía no sirve de mucho de ahí que el despliegue continuo es uno de los conceptos críticos en la cultura DevOps; se puede definir como el proceso de compilación, prueba, configuración y despliegue en un ambiente de producción, en otras palabras, poder dar a los usuarios el software para que lo puedan utilizar de forma automatizada [26].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



4.11. AZURE DEVOPS

Azure DevOps es un servicio de la empresa Microsoft hospedado en la nube el cual ofrece tanto a empresas como desarrolladores independientes herramientas y servicios para desarrollar, probar y desplegar software de forma continua, permitiendo administrar proyectos en todas sus fases como la de desarrollo, compilación, pruebas, despliegue, y monitorización.

Esta plataforma soporta cualquier lenguaje de programación y cualquier plataforma de desarrollo, tiene la capacidad de integrarse con otras plataformas como GitHub para descargar o sincronizar repositorios, Docker para la creación o descargas de imágenes de contenedores, entre otros [27]. Azure DevOps posee diferentes servicios como lo son:

- Azure Repos; es un servicio para el almacenamiento de repositorio de código fuente y soporta dos tipos de sistemas de control de versiones los cuales son Git y Team Foundation Version Control.
- Azure Pipelines es posible crear definiciones de pipelines de integración continua y despliegue continuo para poder automatizar las fases de desarrollo, compilación, pruebas y despliegue de software.
- Azure Boards es posible administrar los esfuerzos del desarrollo de software por medio de listas de características, reportes, entre otras permitiendo que los desarrolladores puedan observar su lista de tareas y al mismo tiempo asignar más tareas a los diferentes usuarios del equipo de trabajo.
- Azure Artifacts; es un almacenamiento y directorio compartido de componentes.
- Azure Test Plans todo producto requiere de una etapa de testing y pruebas, esta herramienta hace posible hacer definiciones de pruebas como pruebas de carga o de aceptación, entre otras.

4.11.1. Pipeline

Pipeline conocido como canalización permite llevar a cabo el proceso para crear y probar de forma automática diferentes proyectos de código para sea posible que estén disponibles para otros. Pipeline realiza una combinación entre la integración continua (CI) y entrega continua para probar y compilar todo el código de forma automática, a su vez, enviarlo a cualquier destino.

Un pipeline es construido por una serie de pasos que van realizando pequeñas acciones o tareas dentro del equipo, este finalmente es compilado y ejecutado de forma





ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



automática cada vez que existe un cambio en el repositorio protegido en el servidor [28].

4.11.2. Agente

Azure Pipeline para poder crear e implementar el código requiere de un agente; se instala en una máquina o dispositivo que permite ejecutar el trabajo internamente dentro de él, de acuerdo a lo que se está llevando a cabo en la aplicación, bien sea un proceso de ejecución de pruebas o el despliegue de un sitio web, entre otros, cada vez que se ejecuta una canalización esta se conecta con el agente configurado y realizar el trabajo que tiene programado, actualmente existen dos tipos de agentes:

4.11.2.1. Agentes alojados en Microsoft

Los agentes alojados en Microsoft son máquinas virtuales prestadas por la compañía de Microsoft las cuales permiten desplegar la acción que se quiere llevar a cabo con el pipeline, este agente es encendido durante el proceso de ejecución de la canalización, se paga por el tiempo de uso y consumo de recursos. [29].

4.11.2.2. Agentes autohospedados

Los agentes autohospedados, valga la redundancia, son agente que se descargan y se pueden instalar en cualquier máquina bien sea física o virtual, este agente se caracteriza porque es controlado por el mismo usuario y dentro de él puede tener todos los programas o herramientas que se necesitan para ejecutar el pipeline, los recursos de este agente se adaptan a los recursos propios de la máquina en la que fue instalada [29].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



CAPÍTULO V: METODOLOGÍA

A continuación, se expone la metodología aplicada para llevar a cabo el presente proyecto.

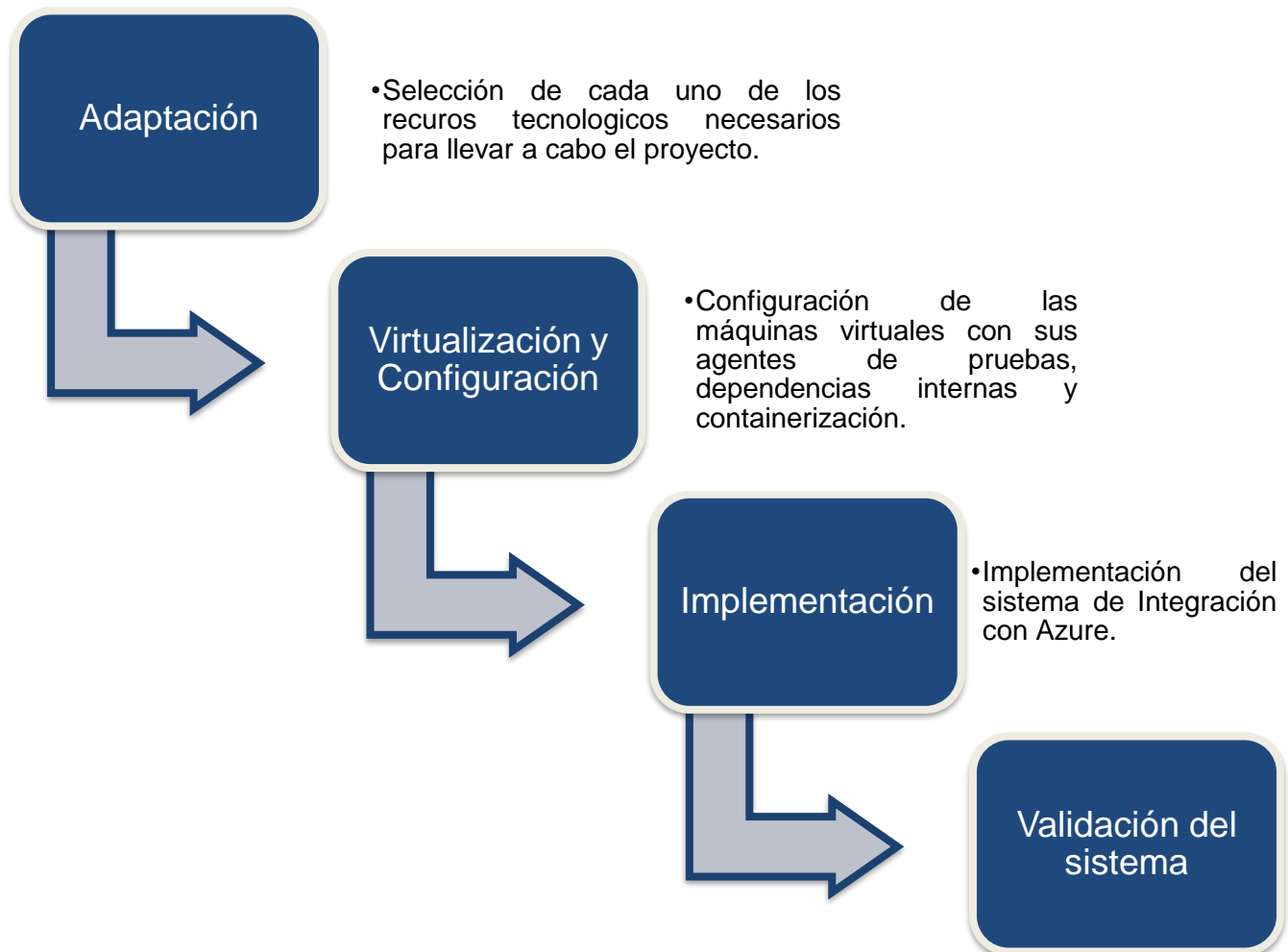


Figura 4. Metodología empleada en el desarrollo de este proyecto.



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



5.1. DESCRIPCIÓN Y CARACTERÍSTICA DEL LUGAR

Digital Ware S.A. es una compañía desarrolladora de software que ofrece diferentes productos como soluciones en diferentes áreas del mercado, su oficina principal se encuentra ubicada en la Avenida Calle 72 en la ciudad de Bogotá D.C, esta empresa cuenta con sedes en otras ciudades de Colombia, pero es allí en donde se encuentra ubicado el equipo de automatización de pruebas liderado por el Ingeniero Edwin Barreto, aquí es donde se lleva a cabo todo el proceso de pruebas de caja negra en el producto Kactus-HCM, permitiendo encontrar distintos errores funcionales en los nuevos productos que se van liberando o en los distintos productos que ingresan para sus respectivos ajustes o correcciones.

5.2. TIPO DE METODOLOGÍA

Para el desarrollo del presente proyecto se propone la implementación de una metodología tipo mixta la cual implica un proceso de recolección, análisis y vinculación de datos cuantitativos y cualitativos en un mismo estudio [30]. La metodología presente se basa en 4 etapas fundamentales las cuales son la etapa de adaptación, etapa de virtualización y configuración, etapa de implementación y finalmente la etapa de validación.

Dentro del proceso de desarrollo de todo el proyecto se va a realizar un proceso de exploración de datos cualitativos para la identificación de los diferentes conceptos y herramientas necesarias para su ejecución, de la misma manera es necesario la definición de características cuantitativas para un proceso de selección adecuada del tipo de servidor que se va a utilizar y definir el número máximo de contenedores que se pueden ejecutar de forma paralela en un mismo dispositivo de acuerdo a la capacidad que contiene, de esta manera a nivel general es posible integrar los dos tipos de datos en un único proyecto que va a beneficiar de forma directa al equipo de Automatización de la empresa Digital Ware.

5.3. DESARROLLO DE METODOLOGÍA POR FASES

5.3.1. FASE 1: Selección de todos los recursos necesarios para el proyecto

Inicialmente se llevó a cabo un proceso de selección de las diferentes herramientas y recursos tecnológicos para llevar a cabo el proyecto, debido a la idea de ejecución de



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



pruebas de automatización de forma paralela en un mismo equipo de cómputo se seleccionó el uso de contenedores Docker para el proceso de despliegue,

Primeramente se aplicó un concepto puramente de Docker en Windows, el cual buscaba poder desplegar el navegador de Internet Explorer dentro de los contenedores y al mismo tiempo que la capa de aplicación fuera renderizada en memoria caso que no se pudo llevar a cabo debido al poco soporte que actualmente se le brinda al navegador, este no permite una renderización en memoria, a diferencia de otros navegadores como Google Chrome y Firefox que son constantemente actualizados y estos permiten la ejecución de ciertos procesos en segundo plano, es decir, que no sean renderizados en pantalla, pero para este proyecto es necesario el navegador Internet Explorer ya que es el medio que utiliza la compañía para presentar el producto como una arquitectura híbrida la cual funciona con parte de la aplicación web y al mismo tiempo con parte de aplicación de escritorio.

Debido a las problemáticas existentes se comenzó a visualizar otras herramientas que permitieran llevar a cabo este procedimiento y al mismo tiempo que lo realice de forma más sencilla, se logró evidenciar la existencia de una herramienta llamada xvfb (Virtual Frame Buffer) la cual permite la ejecución de procesos renderizados en memoria pero solamente era posible aplicarse en sistemas Linux, es por esta razón que se seleccionó un sistema operativo Linux como base para el equipo que funcionara como servidor central.

Actualmente existen diferentes distribuciones GNU/Linux, de acuerdo a muchos sitios y comunidades de este mundo Linux se pudo observar que la distribución Ubuntu es una de las más completas y una de las más populares en el mercado y esto se debe a su gran facilidad de uso [30], por esta razón, se seleccionó como sistema operativo base la distribución GNU/Linux Ubuntu 20.04 brindando una versión estable y al mismo tiempo con excelentes funcionalidades como la facilidad de uso de la herramienta Docker ya que este utiliza características nativas del kernel de Linux [31].

Una vez seleccionado el sistema operativo en base a GNU/Linux se comenzó a analizar las herramientas que se necesitaban para poder ejecutar una prueba de automatización de acuerdo a como se lleva a cabo en el área de la empresa, actualmente esas pruebas son desarrolladas con el lenguaje C# y el framework de automatización Appium en el entorno de desarrollo Visual Studio Community 2019, para poder llevar a cabo el proceso de ejecución se necesita una herramienta llamada WinAppDriver la cual



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



permite el despliegue de pruebas en aplicaciones de escritorio, estas pruebas para que funcionen y se ejecuten de manera correcta se tienen que realizar en el sistema operativo Windows 10 debido a características que este brinda para que funcione correctamente el framework utilizado.

A causa de lo anterior, surge la necesidad de crear máquinas virtuales con el sistema operativo Windows 10 Pro que es el que actualmente usan las máquinas en la compañía con su debida configuración interna para que las pruebas se ejecuten correctamente, inicialmente se utilizó como programa de virtualización Hyper-V de forma gráfica.

Se realizó el proceso de instalación natural del sistema operativo y herramientas, pero se presentó la problemática de la incompatibilidad para la utilización de la herramienta xvfb con este virtualizador para la renderización en memoria, indagando un poco más se logró observar un enfoque que permitía combinar las tecnologías anteriormente dichas, esto permitía la ejecución de navegadores Windows en Linux de manera virtualizada y no renderizada en pantalla por medio del gestor de máquinas virtuales QEMU.

Simultáneamente se pudo observar que la creación de máquinas virtuales con este gestor se podía llevar a cabo por comandos, lo cual posibilitaba crear archivos de bash para crear una máquina virtual mucho más fácil y sencilla, es por esta razón que se tomó la decisión de utilizar ese gestor como base para la creación de las máquinas virtuales.

Se selecciono Docker como herramienta para la creación y el despliegue de contenedores mucho más sencillo, permitiendo al mismo tiempo introducir un sistema operativo dentro, facilitando la ejecución de las máquinas virtuales de forma paralela.

Se tomó la decisión de generar un ScreenShot de la máquina virtual la cual va a permitir guardar un estado de esa máquina y al momento de ejecutarla dentro del contenedor inmediatamente encienda en el estado que tomo el ScreenShot y no tenga que iniciar la máquina desde cero logrando así reducir tiempo en el arranque del contenedor.

Finalmente se selecciona como sistema de integración de las pruebas la plataforma Microsoft Azure que facilita el despliegue continuo de pruebas de automatización junto con la capacidad de integrar de una forma más sencilla el código con las máquinas de ejecución.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



5.3.2. FASE 2: Proceso de virtualización y configuración

Para comprender de una mejor manera el funcionamiento del desarrollo del presente proyecto se elaboró el esquema de la figura 5, la cual resume a nivel general lo que se quiere llevar a cabo, en primer lugar, se tiene la máquina de desarrollo las cuales se identifican por las máquinas de desarrollo del equipo de automatización de Digital Ware, estas máquinas contienen el entorno de desarrollo para codificar las diferentes pruebas de automatización aplicando el concepto de integración continua permitiendo registrar de manera continua los diferentes cambios del proyecto en un repositorio en Microsoft Azure, seguido de ellos se tienen máquinas de Test que como bien se sabe se van a utilizar para probar los diferentes scripts de automatización de pruebas a través de Azure.

Actualmente dentro de la compañía las máquinas de desarrollo son las misma que se utilizan para pruebas, como causa de esto, para el desarrollo del presente proyecto se buscan librerías de dichas máquinas, por eso se plantea un proceso de virtualización, donde cada una de las máquinas virtuales creadas representará una máquina de test y todo este proceso se podrá integrar bajo la plataforma nombrada que es Microsoft Azure.

Teniendo como base las herramientas para poder llevar a cabo el proyecto y después de realizar un proceso de instalación del sistema operativo Ubuntu 20.04 en una máquina Lenovo Ideapad 320, se inicia con el proceso de virtualización para la generación de las máquinas virtuales y el proceso de containerización.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Developer Machine

Esta máquina contiene Visual Studio para escribir y mantener scripts de automatización de pruebas.



Run Build

Ver informe de ejecución de prueba



Test Machine

Esta máquina se utilizará para probar los scripts de automatización de pruebas a través de Azure DevOps.

Figura 5. Diagrama funcionamiento del proyecto. [Elaboración Propia]

El primer paso para poder comenzar a realizar el procedimiento de virtualización es comprobar si en el sistema operativo se encuentra habilitado el proceso de virtualización KVM la cual es una tecnología de Open Source que se encuentra integrada en Linux y esto permite que un equipo físico de Linux ejecute entornos virtuales múltiples y aislados, para ello se ingresa el comando que se visualiza en la figura 6 y si arroja el resultado que se observa allí identifica que posee habilitado esta opción.

```
diego@diego-Lenovo-ideapad-320-15ABR: ~  
diego@diego-Lenovo-ideapad-320-15ABR:~$ ls -l /dev/kvm  
crw-rw----+ 1 root kvm 10, 232 may 21 13:30 /dev/kvm  
diego@diego-Lenovo-ideapad-320-15ABR:~$
```

Figura 6. Verificación soporte KVM. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



Teniendo claro que es posible llevar a cabo un proceso de virtualización se procede a instalar el gestor de máquinas virtuales, para la realización de este proyecto se utilizó QEMU, el cual permite emular y virtualizar a nivel de hardware, para el proceso de instalación se utiliza el comando que se observa en la figura 7.

```
diego@diego-Lenovo-ideapad-320-15ABR: ~  
diego@diego-Lenovo-ideapad-320-15ABR:~$ sudo apt install qemu-system-x86  
[sudo] contraseña para diego:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
qemu-system-x86 ya está en su versión más reciente (1:4.2-3ubuntu6.16).  
Los paquetes indicados a continuación se instalaron de forma automática y ya no  
son necesarios.  
  linux-headers-5.8.0-44-generic linux-hwe-5.8-headers-5.8.0-44  
  linux-image-5.8.0-44-generic linux-modules-5.8.0-44-generic  
  linux-modules-extra-5.8.0-44-generic  
Utilice «sudo apt autoremove» para eliminarlos.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 48 no actualizados.  
diego@diego-Lenovo-ideapad-320-15ABR:~$
```

Figura 7. Instalación del gestor de máquinas virtuales. [Elaboración Propia]

Anteriormente se puede observar que el paquete del gestor de máquinas virtuales ya se encontraba instalado, el siguiente paso es iniciar con el proceso de instalación de la máquina virtual como tal; como se había nombrado anteriormente se necesita instalar el sistema operativo Windows 10 Pro el cual es el que utilizan todas las máquinas de la empresa Digital Ware, en primer lugar, es necesario descargar la imagen ISO de Windows y al mismo tiempo estas máquinas requieren ciertos driver para su correcto funcionamiento como el red el cual permite que la maquina tenga su respectivo acceso a la internet.

Actualmente la infraestructura Fedora aloja controladores Virtio y agentes de software adicionales para máquinas virtuales Windows que se ejecutan en máquinas virtuales basadas en kernel (KVM), esto facilita mucho el proceso sobre todo de conexión a red ya que Virtio es un estándar utilizado para la virtualización para controladores de dispositivos de disco y de red, por tanto se hace necesario la descarga de esta imagen Virtio.

Estas imágenes pueden ser bajadas de las siguientes URLs.



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



<https://www.microsoft.com/en-us/software-download/windows10ISO>

<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/virtio-win-0.1.141-1/virtio-win-0.1.141.iso>

Teniendo descargadas las dos imágenes nombradas en el proceso anterior se hace necesario crear una nueva carpeta en cualquier ubicación del equipo e introducir estas imágenes obtenidas dentro de ese directorio.

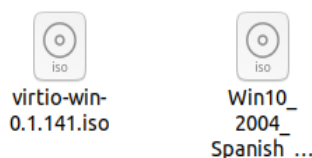


Figura 8. Imágenes ISO. [Elaboración Propia]

Continuando con el procedimiento se requiere de la creación de la imagen del disco que va a permitir guardar e instalar la máquina virtual de Windows, esto se lleva a cabo con el comando que se observa en la siguiente figura, cabe resaltar que en este instante se le da la cantidad de espacio a la partición de la máquina virtual, para este caso se le dieron 40 GB para que las máquinas tengan espacio suficiente para sus dependencias y herramientas.

```
diego@diego-Lenovo-ideapad-320-15ABR: ~/MaquinaVMDocker
diego@diego-Lenovo-ideapad-320-15ABR:~/MaquinaVMDocker$ qemu-img create -f qcow2 hdd.img 40G
Formatting 'hdd.img', fmt=qcow2 size=42949672960 cluster_size=65536 lazy_refcounts=off refcount_bits=16
diego@diego-Lenovo-ideapad-320-15ABR:~/MaquinaVMDocker$
```

Figura 9. Creación imagen del disco. [Elaboración propia]

Al momento de ejecutar el comando anterior inmediatamente se crea la imagen del disco en la carpeta que contenía las imágenes ISO descargadas.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



5.3.2.1. Virtualización de Windows sobre Ubuntu

Con una imagen del disco de almacenamiento disponible para la máquina virtual, se procede a ejecutar el comando de QEMU que va a permitir crearla como se observa en la siguiente figura.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~/MaquinaVMDocker
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaVMDocker$ sudo qemu-system-x86_64 -enable-kvm \
-m machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
-usb -device usb-kbd -device usb-tablet -rtc base=localtime \
-net nic,model=virtio -net user,hostfwd=tcp::4723-:4723 \
-drive file=hdd.img,media=disk,if=virtio \
-drive file=Win10_2004_Spanish_x64.iso,media=cdrom \
-drive file=virtio-win-0.1.141.iso,media=cdrom
[sudo] contraseña para diego:
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
qemu-system-x86_64: warning: guest updated active QH
```

Figura 10. Comando para ejecución del proceso de instalación de la máquina virtual. [Elaboración propia]

La creación de VM con el gestor de máquinas virtuales QEMU es un poco diferente a como se lleva a cabo el proceso de instalación con otros gestores ya que esos se llevan a cabo por medio de interfaz virtual, para este caso se realiza por código, pero en realidad el funcionamiento y estructura es la misma. Para la ejecución inicial se necesita correr el comando en modo super usuario, se debe identificar que el proceso se va a llevar a cabo con el emulador **qemu-system-x86_64**, seguido de ello en la segunda fila se determina la configuración interna de la CPU virtual.

-machine q35 representa un emulador chipset ICH9 que ofrece un controlador EHCI (USB 2.0) que soporta hasta 12 dispositivos, al mismo tiempo se le indica los **sockets**, **cores** y **threads** y al **final -m** representa la cantidad de memoria que se le va a dar a la máquina como tal, para esta máquina se le esta asignando 2048 MiB que hace referencia a aproximadamente 2 GB. La fila representada por **-usb** que es la tercera se relaciona directamente con todas las configuraciones del mouse y el teclado, teniendo en cuenta los movimientos y también la sincronización con la hora del anfitrión.

La instalación con los controladores de red Virtio se hacen más sencillo instalarlos con el argumento **-net**, aquí se le está indicando a la máquina virtual que se va a utilizar los controladores Virtio, a su vez, se habilita la comunicación TCP entre el host principal y la máquina virtual por el puerto que se indica, finalmente con **-drive** se carga la imagen



del disco con controladores Virtio en un dispositivo cdrom junto con el instalador de Windows y de los controladores.

Al momento de aplicar el comando anterior inmediatamente inicia el proceso de instalación del sistema operativo Windows como tal. Observando que se hace un poco incomodo introducir todos esos comandos mostrados anteriormente, se realizó un proceso de creación de script de bash el cual permite crear cada máquina virtual de una forma más sencilla y automatizada, el script de bash es el que se muestra a continuación.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~  
└─$ /bin/bash  
#Script para la instalación de una máquina virtual con el gestor de máquinas virtuales Qemu  
#Autor: Diego Leal  
  
echo "Bienvenidos al script para la creación de una máquina virtual con Qemu" | boxes -d ian_jones  
  
#Buscar en el dispositivo si se encuentra habilitada la opción de virtualización anidada  
echo -e "\n"  
echo "Virtualización Anidada" | boxes -d stone  
echo "En este apartado se verifica si la opción de virtualización anidada se encuentra habilitada"  
ls -l /dev/kvm  
  
#Se procede a instalar el gestor de máquinas virtuales  
  
echo -e "\n"  
echo "Instalación gestor de máquinas virtuales Qemu" | boxes -d stone  
echo -e "\n Verificar instalación de qemu-system-x86..."  
verifyInstall=$(whereis qemu)  
if [ $? -eq 0 ]; then  
    echo -e "\n qemu-system-x86 Ya se encuentra instalado en el equipo"  
else  
    echo -e "\n El programa no se encuentra instalado inicio instalación..."  
    read -e -s -p "\n Por favor ingresar la contraseña de sudo: " password  
    echo "password" | sudo apt update  
    echo "password" | sudo apt -y install qemu-system-x86  
fi  
  
#Se procedo a crear la la carpeta y introducir la imagen ISO de Windows y el driver de Virtio  
  
echo -e "\n"  
echo "Creando directorio y copiando archivos" | boxes -d stone  
echo -e "\n Creando directorio..."  
read -p "Por favor ingrese el nombre de la carpeta a crear: " directorio  
  
#Creando directorio  
mkdir -m 755 $directorio  
ls -la $directorio
```

Figura 11. Primera script de bash creación máquina virtual. [Elaboración propia]

En esta primera parte del script de bash creado se puede observar que realiza el mismo procedimiento anterior, pero de forma automatizada dado que ya no es necesario introducir todos los comandos, sino que se ejecuta el bash e inmediatamente se pide algunos datos importantes y arranca la maquina en cuestión de segundos, este script crea la carpeta que necesita, se copia las imágenes ISOS ya descargadas, crea la imagen del disco con la capacidad que se le ingresa y finalmente ejecuta el inicio de la máquina.



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



```
diego@diego-Lenovo-Ideapad-320-15ABR: ~  
read -p "Por favor ingrese el nombre de la carpeta a crear: " directorio  
  
#Creando directorio  
mkdir -m 755 $directorio  
ls -la $directorio  
echo -e "\n Directorio creado correctamente"  
  
#Copiando los archivos de un directorio a otro  
echo -e "\n"  
echo "Copiando archivos..."  
cd /home/diego/ImagenesISOS  
cp /home/diego/ImagenesISOS/virtio-win-0.1.141.iso /home/diego/$directorio  
cp /home/diego/ImagenesISOS/Wln10_2004_Spanish_x64.iso /home/diego/$directorio  
  
#Se procede a crear la imagen del disco donde se va a crear la máquina virtual  
echo -e "\n"  
echo "Creando imagen disco de la VM" | boxes -d stone  
  
cd /home/diego/$directorio  
  
chmod 755 virtio-win-0.1.141.iso  
chmod 755 Wln10_2004_Spanish_x64.iso  
  
nombrediscov='^[a-z]{3}$'  
capacidadvmv='^[0-9]{2}G$'  
  
echo -e "\n"  
read -p "Por favor ingrese la capacidad de la máquina en G (Se recomienda 50G): " capacidadvm  
  
echo -e "\n"  
echo "Creando la imagen del disco..."  
  
if [[ $capacidadvm == $capacidadvmv ]]; then  
qemu-img create -f qcow2 hdd.img $capacidadvm  
else  
echo -e "\n"  
echo "Posiblemente el valor que ingreso es invalido verifique que la capacidad sean 2 numeros seguido de la letra G mayuscula..."  
fi
```

Figura 12. Continuación primer script de bash creación máquina virtual. [Elaboración propia]

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~  
nombrediscov='^[a-z]{3}$'  
capacidadvmv='^[0-9]{2}G$'  
  
echo -e "\n"  
read -p "Por favor ingrese la capacidad de la máquina en G (Se recomienda 50G): " capacidadvm  
  
echo -e "\n"  
echo "Creando la imagen del disco..."  
  
if [[ $capacidadvm == $capacidadvmv ]]; then  
qemu-img create -f qcow2 hdd.img $capacidadvm  
else  
echo -e "\n"  
echo "Posiblemente el valor que ingreso es invalido verifique que la capacidad sean 2 numeros seguido de la letra G mayuscula..."  
fi  
  
# Se procede a crear la máquina virtual del gestor de máquinas virtuales  
  
echo -e "\n"  
echo "Creando la máquina virtual" | boxes -d stone  
  
echo -e "\n"  
read -p "Por favor introduzca el puerto por el cual quiere que se ejecute la máquina virtual: " puerto  
  
echo -e "\n"  
echo "Creando máquina virtual..."  
cd /home/diego/$directorio  
sudo qemu-system-x86_64 -enable-kvm \  
-machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \  
-usb -device usb-kbd -device usb-tablet -rtc base=localtime \  
-net nic,model=virtio -net user,hostfwd=tcp::$puerto::$puerto \  
-drive file=hdd.img,media=disk,if=virtio \  
-drive file=Wln10_2004_Spanish_x64.iso,media=cdrom \  
-drive file=virtio-win-0.1.141.iso,media=cdrom
```

Figura 13. Parte final primer script de bash creación máquina virtual. [Elaboración propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



En las tres últimas figuras visualizadas anteriormente se puede observar el desarrollo del script de bash el cual permite de una manera más automatizada ejecutar los primeros pasos para el proceso de virtualización cuando se ejecuta el script se obtiene como resultado lo siguiente.

NOTA: Para la ejecución del script de bash se escribe el comando **bash** seguido del nombre del script.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~
diego@diego-Lenovo-Ideapad-320-15ABR:~$ bash createvirtualmachine2.sh

      //\\//
     ( ( ) ( ) )
    .000o--( )--o000.
-----
| Bienvenidos al script para la creación de una máquina virtual con Qemu |
-----
      oooo      oooo.
     { }      { }
-----
+-----+
| Virtualización Anidada |
+-----+
En este apartado se verifica si la opción de virtualización anidada se encuentra habilitada
crw-rw----+ 1 root kvm 10, 232 may 21 21:07 /dev/kvm

+-----+
| Instalación gestor de máquinas virtuales Qemu |
+-----+
Verificar instalación de qemu-system-x86...
qemu-system-x86 Ya se encuentra instalado en el equipo

+-----+
| Creando directorio y copiando archivos |
+-----+
Creando directorio...
Por favor ingrese el nombre de la carpeta a crear: MáquinaTestis
total 8
drwxr-xr-x  2 diego diego 4096 may 21 21:12 .
drwxr-xr-x 29 diego diego 4096 may 21 21:12 ..
Directorio creado correctamente

Copiando archivos...
```

Figura 14. Primera parte ejecución Script 1. [Elaboración Propia]

Si se observa en la figura anterior primeramente realiza el procedimiento de verificación del KVM dentro del host principal para poder confirmar que es posible llevar a cabo el procedimiento, partiendo de esto, se instala el paquete del gestor de máquinas virtuales, si la máquina no lo tiene instalado y finalmente le pide por teclado un nombre de carpeta la cual se crea de manera automática e inmediatamente se pasan las imágenes ISO que se necesitan como Windows 10 y Virtio.



A continuación, se digita la cantidad en memoria para la creación del disco, se crea la imagen del disco y finalmente se ejecuta el comando con las configuraciones de la máquina virtual para llevar a cabo el proceso de instalación.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~
+-----+
Verificar instalación de qemu-system-x86...
qemu-system-x86 Ya se encuentra instalado en el equipo
+-----+
| Creando directorio y copiando archivos |
+-----+
Creando directorio...
Por favor ingrese el nombre de la carpeta a crear: MaquinaTesis
total 8
drwxr-xr-x  2 diego diego 4096 may 21 21:12 .
drwxr-xr-x 29 diego diego 4096 may 21 21:12 ..
Directorio creado correctamente

Copiando archivos...
+-----+
| Creando imagen disco de la VM |
+-----+
Por favor ingrese la capacidad de la máquina en G (Se recomienda 50G): 50G

Creando la imagen del disco...
Formatting 'hdd.img', fmt=qcow2 size=53687091200 cluster_size=65536 lazy_refcounts=off refcount_bits=16

+-----+
| Creando la máquina virtual |
+-----+
Por favor introduzca el puerto por el cual quiere que se ejecute la máquina virtual: 4444

Creando máquina virtual...
[sudo] contraseña para diego:
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
qemu-system-x86_64: warning: guest updated active QH
|
```

Figura 15. Segunda parte ejecución Script 1. [Elaboración Propia]

Durante el proceso de instalación se hace necesario cargar los driver de virtio junto con el disco para darle esta capacidad asignada a la máquina virtual, cabe resaltar que el sistema operativo interno de las máquinas virtuales es Windows 10 Pro el cual es el que se utiliza en todas las máquinas de la compañía Digital Ware y se realiza una instalación personalizada para poder cargar los drivers mencionados anteriormente.

Al momento de asignar las respectivas particiones Windows no podrá encontrar el dispositivo de almacenamiento de la VM de forma predeterminada; debido a que el controlador de almacenamiento virtio no está instalado en Windows, es por esta razón que se hace necesario cargar el controlador.

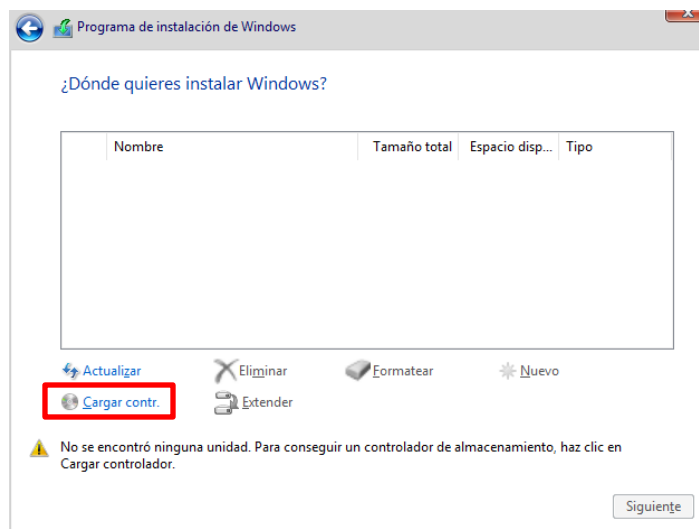


Figura 16. Cargar controlador Virtio en la instalación del S.O. [Elaboración Propia]

Se procede a examinar y desde la unidad de CD virtio se selecciona la carpeta **Viostor** > **w10** (Para Windows 10) > **amd64** y se da en la opción de aceptar.

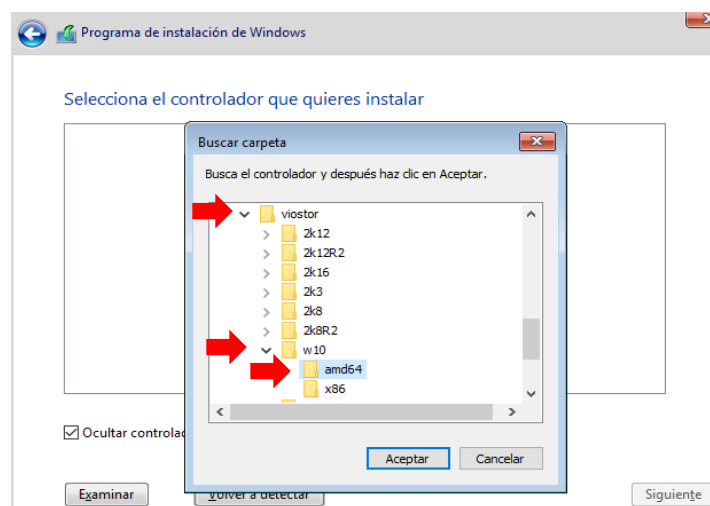


Figura 17. Seleccionar controlador Virtio en la instalación del S.O. [Elaboración Propia]

Inmediatamente el instalador del controlador debe detectar el controlador del controlador SCSI Red Hat VirtIO del CD, se da en la opción siguiente.

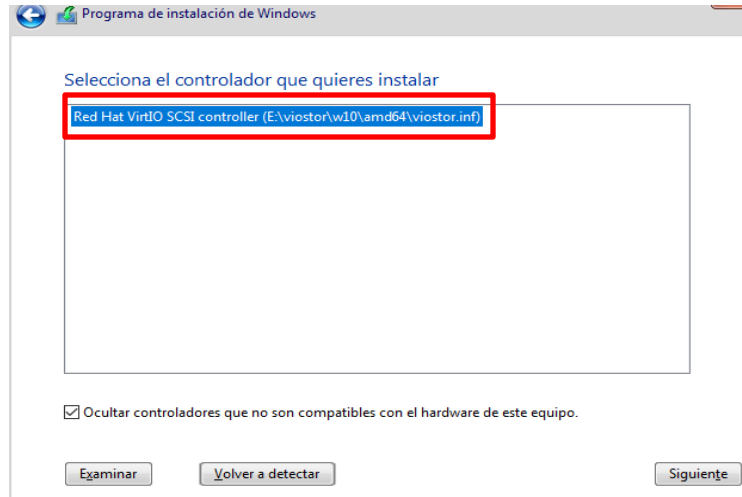


Figura 18. Detección controlador SCSI Red Hat VirtIO. [Elaboración propia]

Ahora, debería ver el dispositivo de almacenamiento en el instalador de Windows con la capacidad que le fue asignada en su momento y se procede a continuar con el proceso de instalación normal del sistema operativo hasta que arranque el sistema.

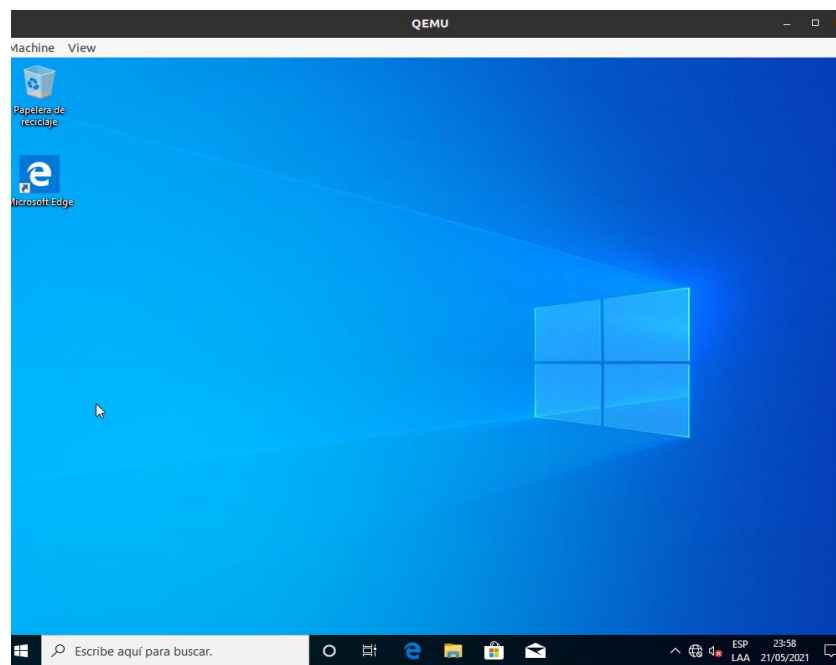


Figura 19. Inicio máquina virtual creada. [Elaboración propia]



Para realizar el proceso de ejecución de pruebas de automatización es necesario tener conexión a internet por el hecho de que el producto al que se le aplican las respectivas pruebas se ejecuta en el navegador Internet Explorer.

Cuando la máquina inicia no se tiene aún acceso directo a conexión, esto se hace por medio de los drivers de virtio, pero si se selecciona la opción de Administrador de dispositivo es posible observar que el dispositivo de red virtio no se reconoce en Windows, por tanto, se requiere instalarlo, para ello se le da clic derecho en el adaptador Ethernet y se selecciona actualizar controlador.

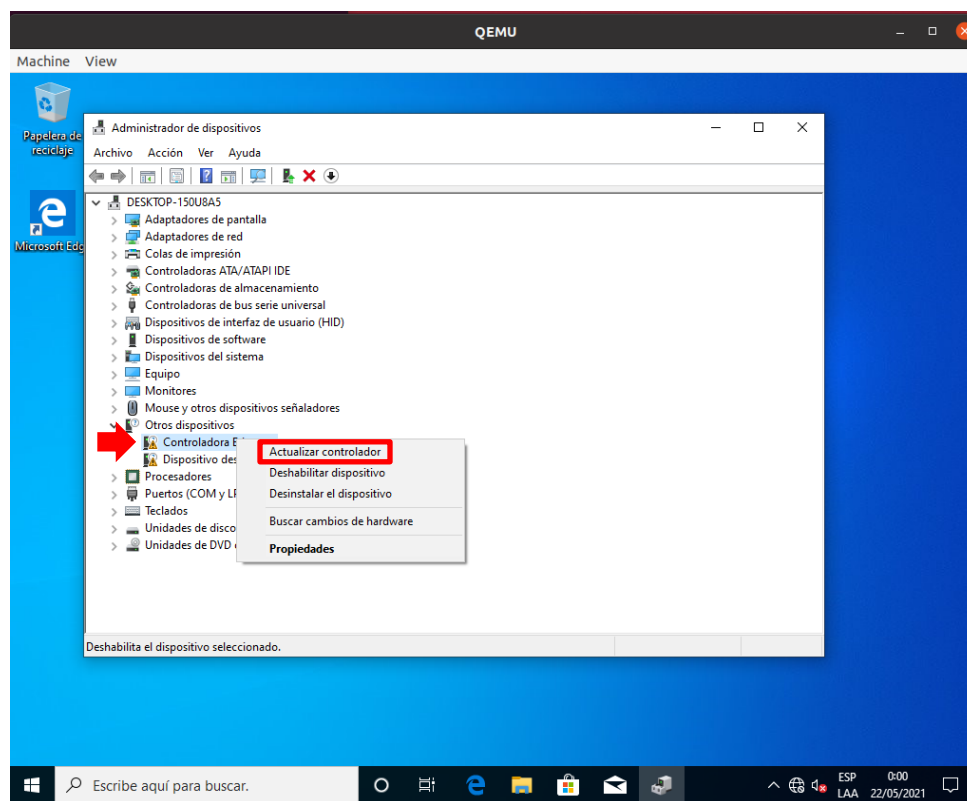


Figura 20. Actualización del controlador Ethernet. [Elaboración propia]

Se busca el controlador de software dentro de la máquina como tal, se da clic en la opción de examinar y finalmente se selecciona el CDROM del controlador virtio y se le da clic en Aceptar.

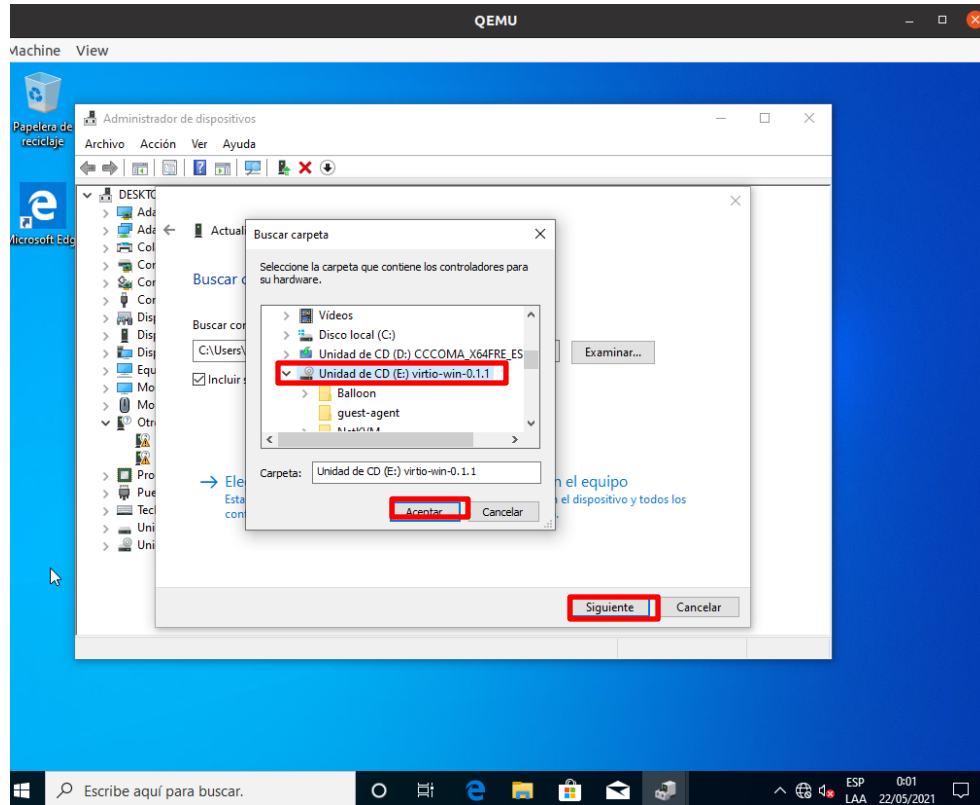


Figura 21. Cargar driver para Ethernet desde el CDROM Virtio. [Elaboración propia]

Finalmente se lleva cabo el proceso de instalación del controlador e inmediatamente la máquina reconoce como adaptador Ethernet Red Hat Virtio.

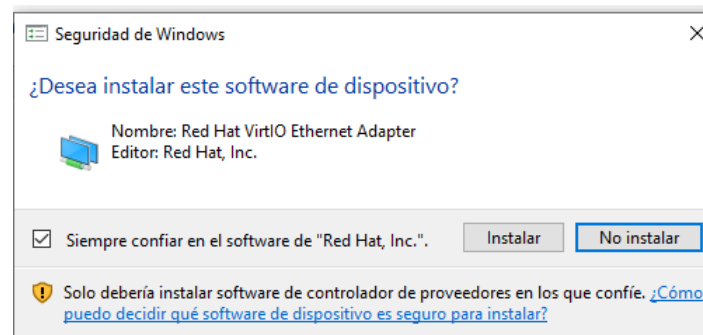


Figura 22. Instalar controlador Ethernet. [Elaboración propia]



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Como se observa en la siguiente imagen la máquina virtual ya tiene acceso a internet.

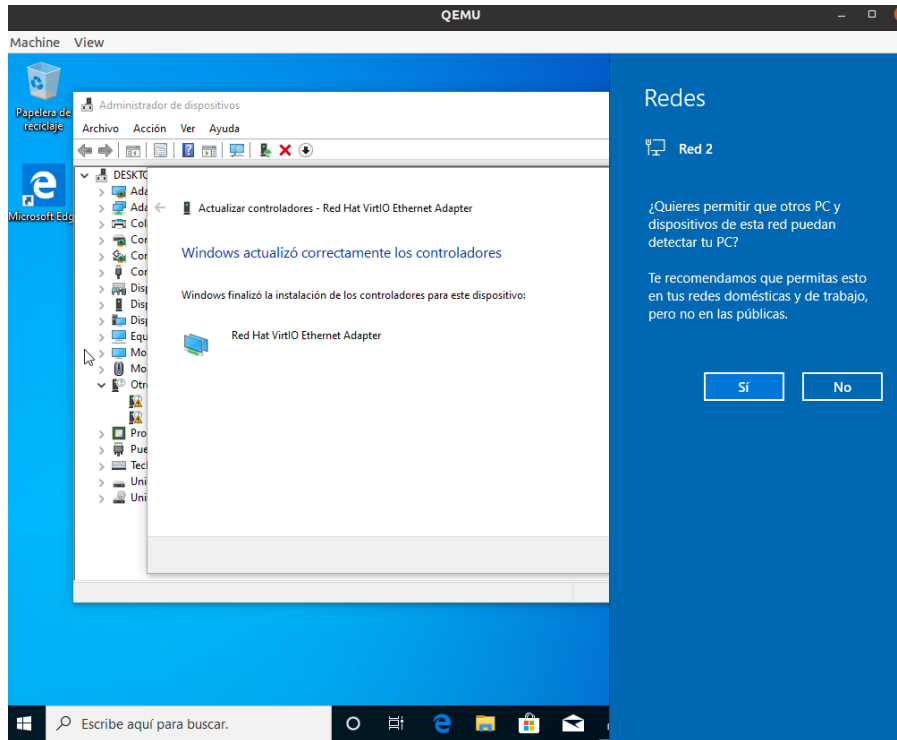


Figura 23. Conexión a internet máquina virtual. [Elaboración propia]

El paso a seguir es el de configurar todas las herramientas necesarias dentro de la máquina para que sea posible la ejecución de las pruebas de automatización, para ello tendremos en cuenta la siguiente tabla.

Developer Machine	Test Machine
<ul style="list-style-type: none"> - Visual Studio Community 2019. - WinAppDriver. - .Net Desktop Development. - Modo desarrollador encendido. 	<ul style="list-style-type: none"> - Visual Studio Build Tools 2019. - WinAppDriver. - Azure DevOps Agent. - Modo desarrollador encendido - Sincronizador del servidor de hora y fecha. - Sincronizador Google Drive.

Tabla 2. Herramientas necesarias para las máquinas. [Elaboración Propia]

Observando la tabla anterior y continuando con el proceso de virtualización se inicia con el procedimiento de instalación de todas las herramientas.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
 Pamplona - Norte de Santander - Colombia
 Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



En primera instancia se activa el modo desarrollador en la máquina.

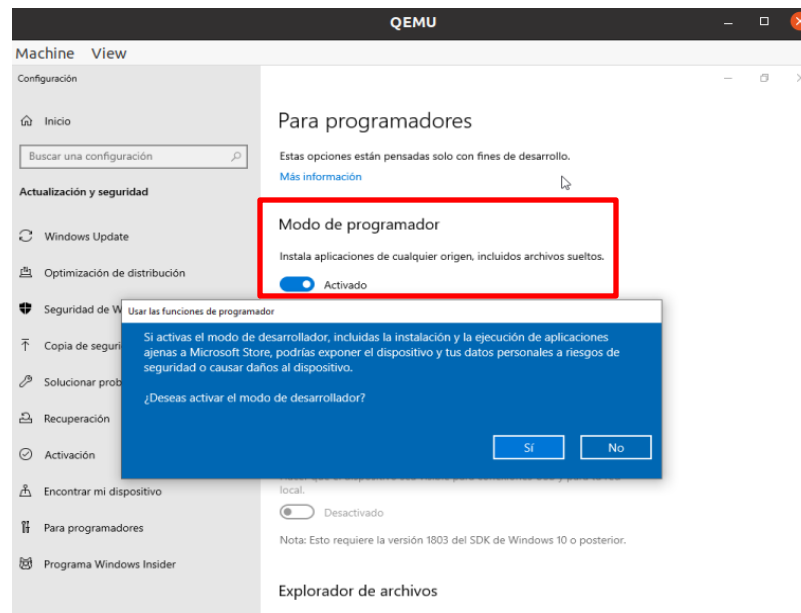


Figura 24. Activando modo desarrollador. [Elaboración Propia]

Se desactiva el firewall para que no exista ningún inconveniente con la comunicación con el agente de Azure o por la escucha del puerto por donde ejecuta el WinAppDriver.

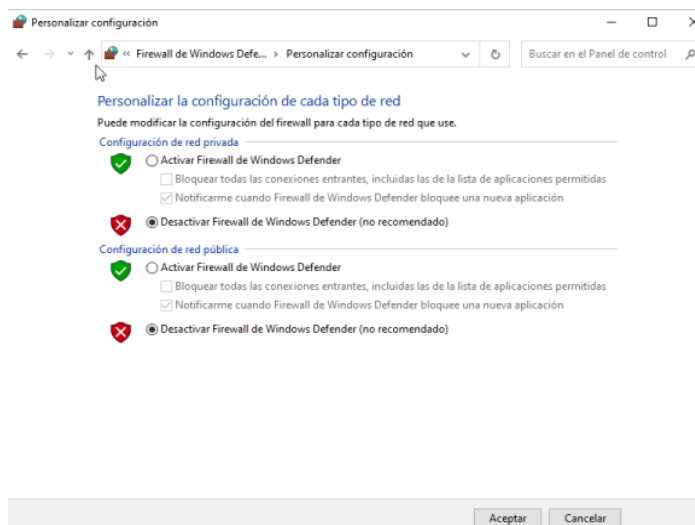


Figura 25. Desactivando Firewall en la máquina virtual. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Debido a que se va a ejecutar una pequeña prueba de muestra dentro de las máquinas que simulara la ejecución de las pruebas reales de automatización, se requiere el uso del navegador Internet Explorer agregando a sitios de confianza la página a la cual se va a ingresar para realizar el proceso de automatización.

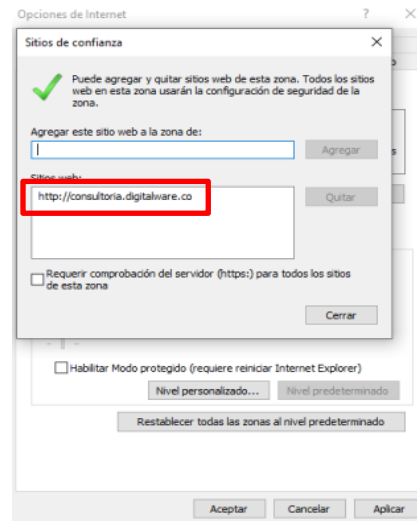


Figura 26. Agregando a sitios de confianza en Internet Explorer URL de la página a automatizar. [Elaboración Propia]

En base a la tabla 2 se puede verificar se solicita la instalación del Windows Application Driver (WinAppDriver) que va a permitir ejecutar las pruebas de automatización utilizando el framework de Appium, a su vez, se instala Visual Studio Build Tools, esto va a permitir que cada una de las máquinas tenga las herramientas suficientes como la compatibilidad con las librerías .NET framework para que se posible la ejecución de las pruebas, estas herramientas se pueden descargar de los siguientes URLs.

<https://github.com/microsoft/WinAppDriver>

<https://visualstudio.microsoft.com/es/downloads/>



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750

Para el proceso de instalación de Visual Studio Build Tools se utilizan las herramientas relacionadas con .NET, por eso se instalan solamente las **Herramientas de compilación para escritorio de .NET** con sus respectivas herramientas de desarrollo .NET Core y .NET Framework en todas sus versiones

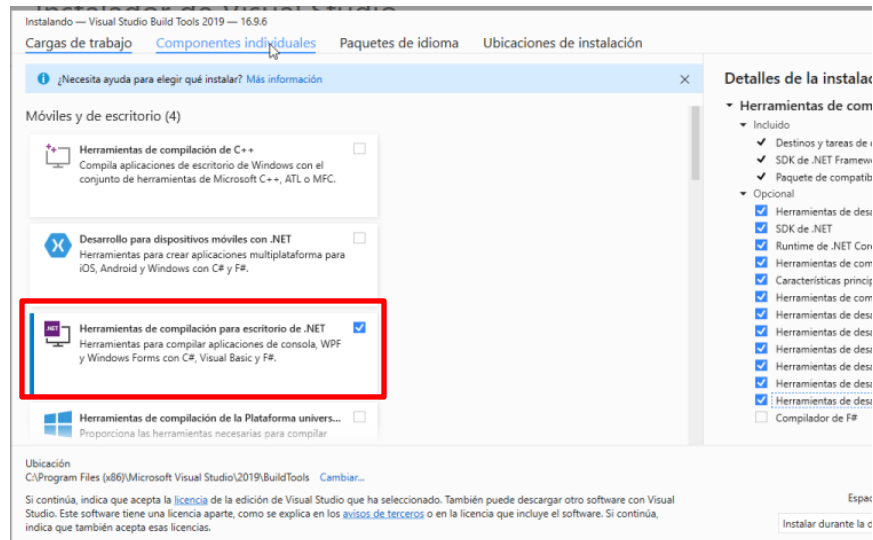


Figura 27. Herramientas de compilación para .NET. [Elaboración Propia]

Hasta el momento se tienen instaladas y configuradas algunas de las herramientas fundamentales para poder llevar a cabo el procedimiento de automatización, debido a que se planteó la toma de un snapshot de la máquina virtual para permitir que esta no tenga que hacer el proceso de inicio, sino que inmediatamente arranque desde el estado en el que fue guardado, entonces se procede a la creación y para ello se utiliza el comando que se observa en la figura 28.

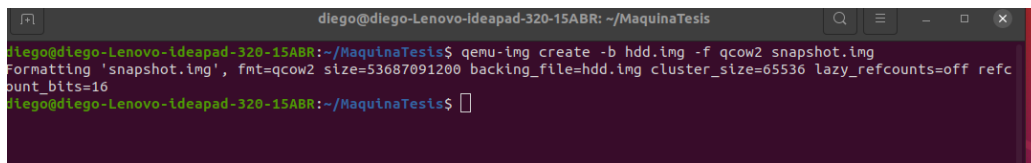


Figura 28. Creación snapshot de la máquina virtual. [Elaboración Propia]

Se puede observar que a partir de la imagen hdd.img que es la que contiene la VM se está creando el snapshot con el formato de imagen de disco qcow2 el cual es una representación de un dispositivo de bloque de tamaño fijo en un archivo.

Se puede confirmar que se creó la imagen respectiva del snapshot.

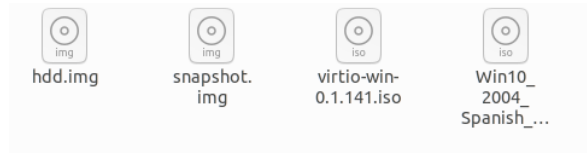


Figura 29. Confirmación creación snapshot de la máquina virtual. [Elaboración Propia]

El siguiente paso es ejecutar la máquina virtual teniendo en cuenta que se ejecuta desde la imagen que se creó del snapshot y eso se realiza con el siguiente comando.

```
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaTesis$ sudo qemu-system-x86_64 -enable-kvm \
> -machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
> -usb -device usb-kbd -device usb-tablet -rtc bases=localtime \
> -net nic,model=virtio -net user,hostfwd=tcp::4444-:4444 \
> -drive file=snapshot.img,media=disk,if=virtio \
> -monitor stdio
[sudo] contraseña para diego:
QEMU 4.2.1 monitor - type 'help' for more information
(qemu) qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
qemu-system-x86_64: warning: guest updated active QH
█
```

Figura 30. Ejecutando máquina virtual desde el Snapshot. [Elaboración Propia]

Para realizar este procedimiento de una forma más automatizada también se creó otro script de bash como se muestra en la siguiente figura.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~
❏ !/bin/bash
#Script que permitira crear el snapshot de la máquina virtual
#Autor: Diego Leal

echo "$Port, $Name"
echo "Bienvenidos al Script para crear el Snapshot de la VM" | boxes -d ian_jones

echo -e "\n"
echo "Creando Snapshot" | boxes -d stone

read -p "Por favor ingrese el mismo nombre de la carpeta que creo anteriormente para crear la máquina virtual: " nombrecarpeta
echo -e "\n"
read -p "Por favor ingrese el puerto por el cual creo la máquina virtual anteriormente: " puertovm

cd /home/diego/$nombrecarpeta

qemu-img create -b hdd.img -f qcow2 snapshot.img

echo -e "\n"
echo "Ejecutando el Snapshot" | boxes -d stone

sudo qemu-system-x86_64 -enable-kvm \
-m machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
-usb -device usb-kbd -device usb-tablet -rtc base=localtime \
-net nic,model=virtio -net user,hostfwd=tcp::$puertovm-::$puertovm \
-drive file=snapshot.img,media=disk,if=virtio \
-monitor stdio
```

Figura 31. Segundo Script de bash creación snapshot. [Elaboración Propia]

Para el proceso de ejecución de ese script se requiere el ingreso del nombre de la carpeta donde se aloja la máquina virtual y el puerto en el que se ejecutó para que pueda crear la imagen del snapshot y ejecute la VM como tal.

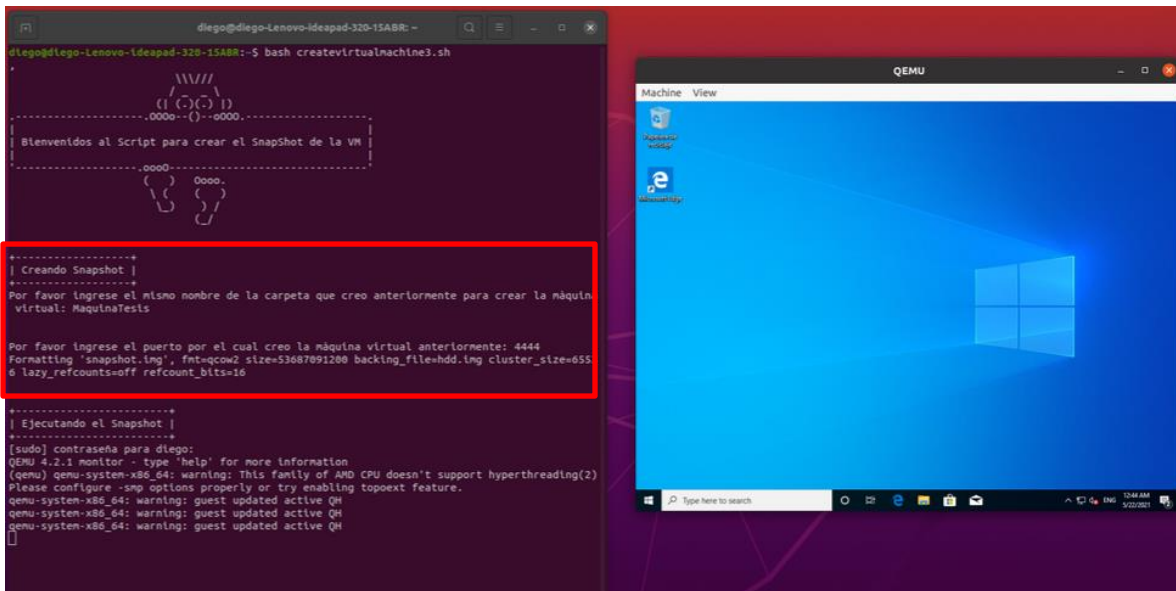


Figura 32. Ejecución segundo Script de bash creación snapshot. [Elaboración Propia]

NOTA: Con todas las máquinas que se vayan a crear se debe llevar a cabo exactamente el mismo procedimiento, teniendo en cuenta de que le debe dar otro puerto a la máquina y otro nombre a la carpeta donde se alojara.

Para poder continuar con el proceso se recurre a la instalación del Agente de pruebas asociado con la plataforma Microsoft Azure, Azure DevOps proporciona una solución completa para almacenar scripts de prueba y al mismo tiempo ejecutarlos, para llevar a cabo el proceso de ejecución utiliza los Pipelines, estos al momento de ejecutarse requieren de un agente que permita llevar a cabo el proceso que se quiere realizar, para este proyecto se planteó la utilización de agentes autohospedados, es decir, agentes que se instalan directamente en las máquinas y son controladas por el mismo usuario y no por Microsoft.

Para la instalación del agente es necesario crear un grupo de agentes (Pool Agent) el cual va a permitir identificar cada máquina dentro de un pipeline para su debida ejecución, para este procedimiento se hace necesario saltar a la Fase 3 que es la de



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



integración con Microsoft Azure y después se continua con el proceso de final de la virtualización.

5.3.3. FASE 3: Implementación sistema de integración en Azure

Para realizar el proceso de la implementación de la integración con Azure es de vital importancia observar la siguiente estructura y de acuerdo esto se ira configurando cada una de las opciones en Microsoft Azure.

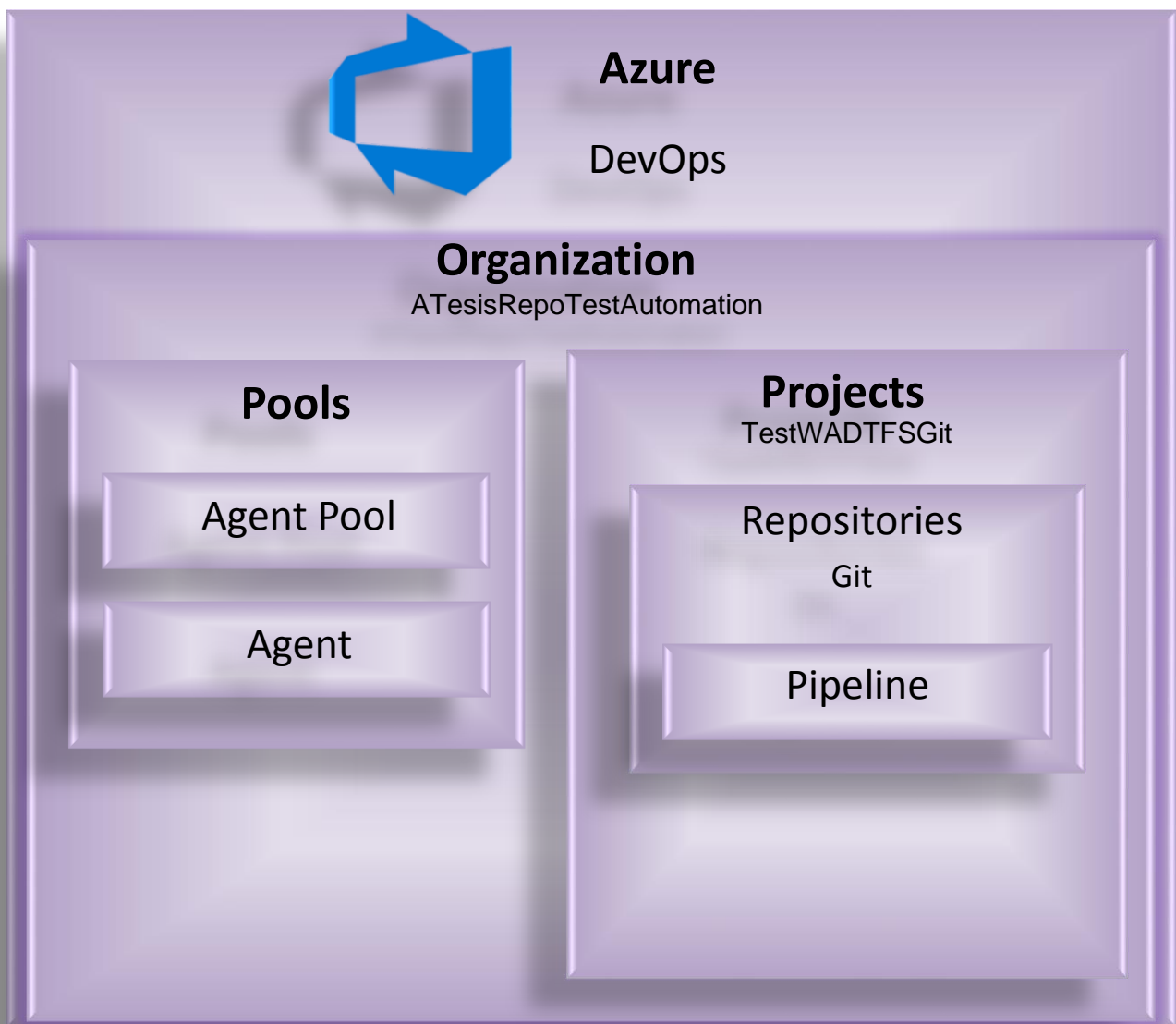


Figura 33. Estructura implementación sistema de integración. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Como se aprecia en la figura anterior la plataforma Microsoft Azure internamente siempre se basa en una organización general la cual va a permitir crear diversos grupos de agentes y al mismo tiempo distintos proyectos y dentro de cada uno de los proyectos creados es posible crear un repositorio interno para el proceso de integración continua y el despliegue de diferentes Pipelines.

Para crear una nueva organización que funcionará como un servidor en la nube simplemente se debe hacer lo siguiente.

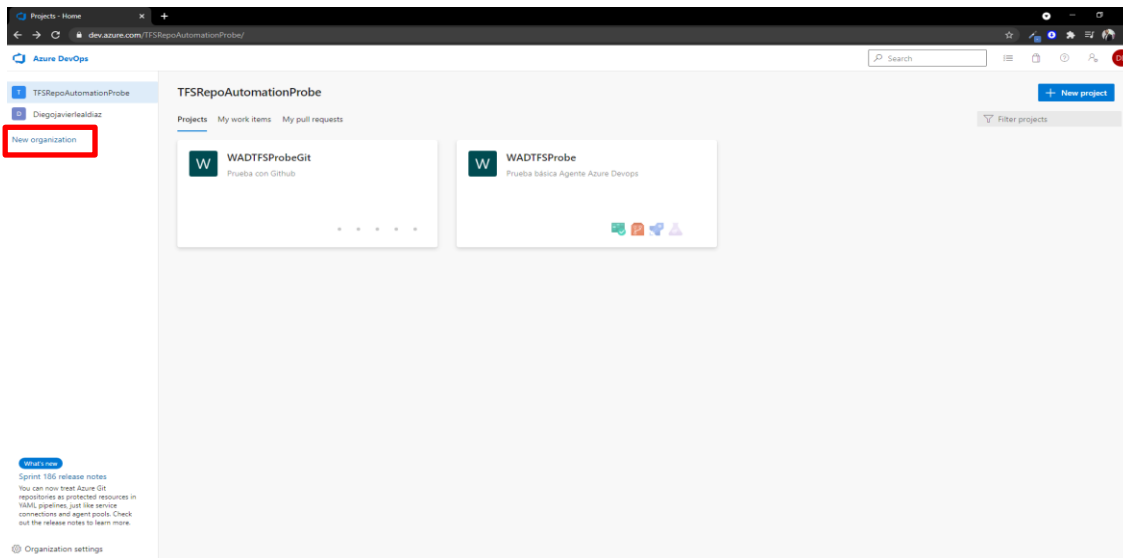


Figura 34. Creando organización Azure. [Elaboración Propia]

En la siguiente ventana que sale se debe definir el nombre que le va a dar a esa nueva organización y al mismo tiempo el servidor de Microsoft Azure más cercano al lugar en el que cada persona se encuentre en este caso Brasil.

Name your Azure DevOps organization

We'll host your projects in

Figura 35. Configuración organización. [Elaboración Propia]



Al crear la organización se procede a crear un nuevo proyecto dentro de ella, el proyecto que se va a crear utilizará como sistema de control de versiones Git, se le da un nombre específico, una descripción si lo quiere y se selecciona el tipo de proyecto si será privado o público que para el desarrollo de este proyecto es privado.

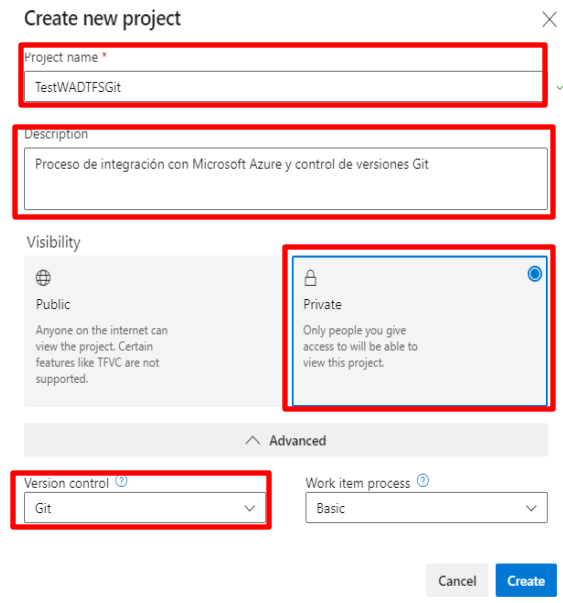


Figura 36. Creación de nuevo proyecto dentro de Azure. [Elaboración Propia]

El siguiente procedimiento es crear el Pool de Agentes o el grupo de agente para ello simplemente se debe seleccionar la configuración de la organización y se selecciona Agent Pools.

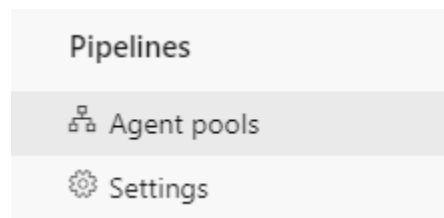


Figura 37. Pool de agente. [Elaboración Propia]



Dentro de Agent Pools se debe crear un nuevo grupo de agente dando en Add Pool y se ingresa la información del agente primero asignando que tipo de agente es, si es autohospedado o prestado por Microsoft, se le da un nombre al grupo y una descripción.

Add agent pool ×

Agent pools are shared across an organization.

Pool type:
Self-hosted ▼

A pool of agents that you set up and manage on your own to run jobs. [Learn more.](#)

Name:
GAgente01

Description (optional):
Agregar grupo de agentes 01

[Markdown supported.](#)

Pipeline permissions:

- Grant access permission to all pipelines
- Auto-provision this agent pool in all projects

Figura 38. Creación grupo de agente.
[Elaboración Propia]

Dentro del pool de agentes se procede a copiar una ruta la cual va a permitir descargar el agente autohospedado en la máquina virtual, para ello simplemente copiamos la ruta especificada que genere el Azure y se pega en cualquier navegador de la VM e inmediatamente se descarga.

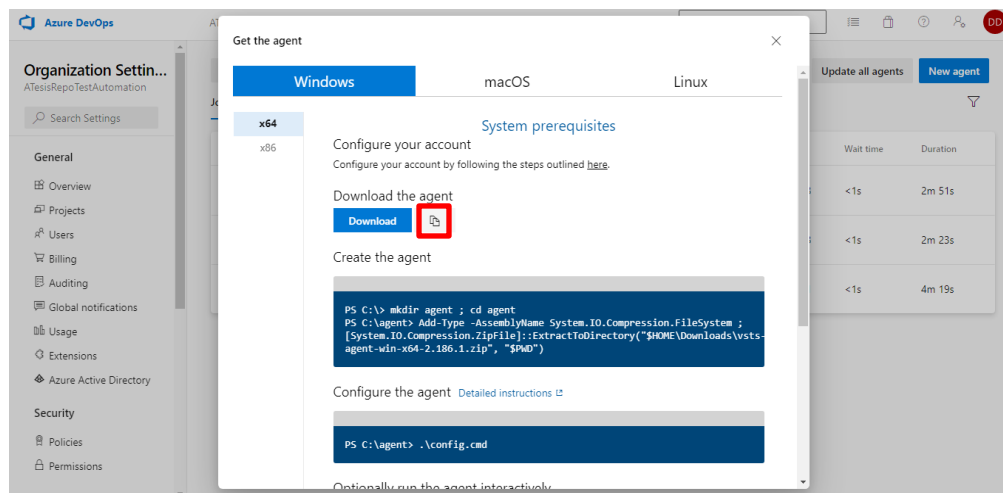


Figura 39. Ruta de descarga del agente. [Elaboración Propia]



Para poder hacer que el Agente conecte directamente con la organización que se acabó de crear es necesario generar un Token que va a permitir llevar a cabo esa conexión y al mismo tiempo indicar que pertenece a la organización creada anteriormente.

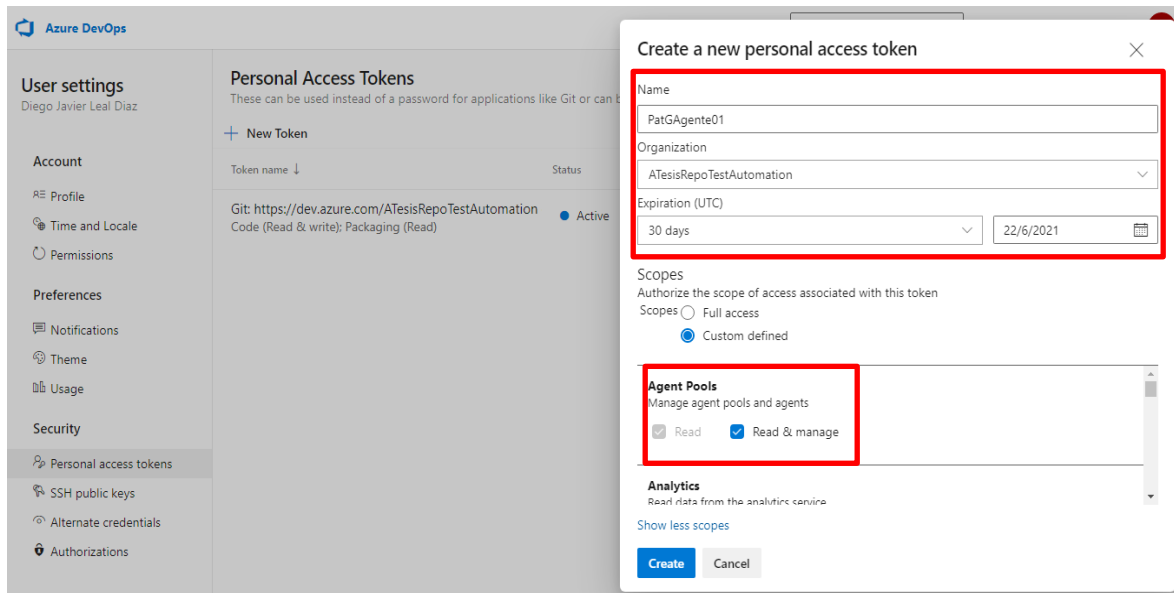


Figura 40. Generación Personal Access Token. [Elaboración Propia]

Finalmente, Azure va a generar un PAT que se debe copiar y guardar para tenerlo en cuenta en el proceso de instalación del agente como tal.

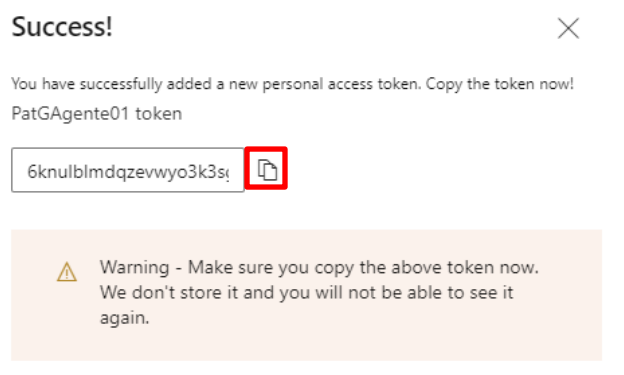


Figura 41. PAT generado. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Se procede a instalar el agente dentro de la máquina virtual, para ello se descarga primero en la ruta que se copió en la figura 39, se descomprime, se le cambia el nombre y finalmente se pega en el disco C de la máquina tal y como se observa en la figura 42.

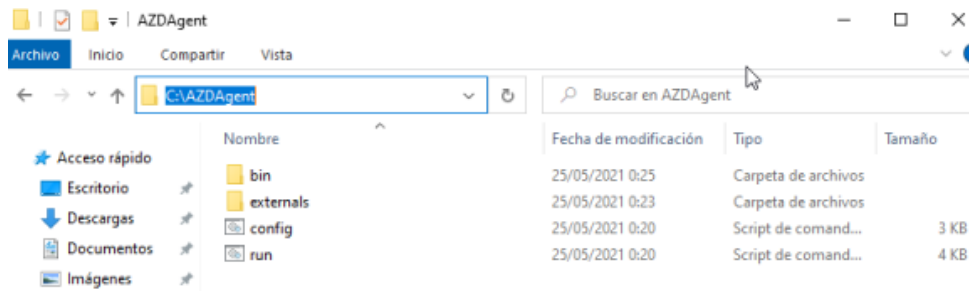


Figura 42. Agente de Azure descargado. [Elaboración Propia]

El siguiente paso es abrir un PowerShell dando todos los permisos de administrador.

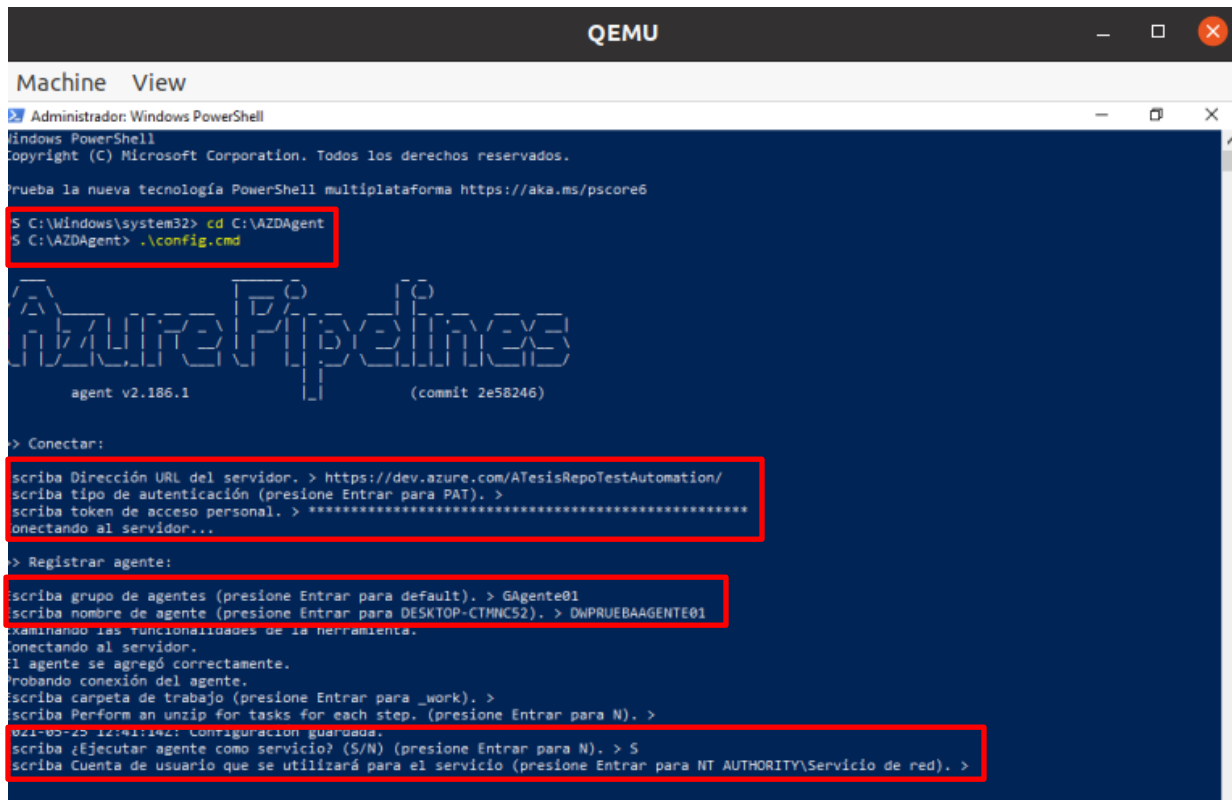


Figura 43. Configuración agente de Azure en máquina virtual. [Elaboración Propia]



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Para la configuración del agente dentro del PowerShell como se observó en la figura 43 se accede a la ruta de la carpeta del agente que se descargó y se colocó en el disco local C, se procede a seleccionar el archivo de configuración con `.\config.cmd`, dentro del archivo de configuración primero se ingresa la ruta de la organización que se acabó de crear que para este caso fue <https://dev.azure.com/ATesisRepoTestAutomation/>, ya teniendo configurado hacia dónde va a apuntar el agente se debe ingresar el PAT generado en pasos anterior para permitir una conexión directa, después se ingresa el nombre del grupo de agente o Agent Pool que se creó en la figura 38 (**GAgente01**), se le da un nombre cualquiera respectivo a ese agente y finalmente se le da que **Si** en la opción de ejecutar el agente como un servicio de Windows.

Esto va a permitir que no se tenga que ejecutar el agente de forma manual cada vez que se abra el snapshot de la máquina virtual sino que funcione como un servicio que se enciende cada vez que la máquina se encuentra activa.

Al instante que termina se puede validar que el agente se haya creado de forma correcta dentro del grupo de agentes GAgente01 y se encuentre en estado Online.

Name	Last run	Current status	Agent version	Enabled
DWPRUEBAAGENTE01 ● Online		Idle	2.186.1	On

Figura 44. Verificación agente instalado. [Elaboración Propia].



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



El siguiente procedimiento que se debe llevar a cabo es el de la integración del código con el sistema de control de versiones que brinda Azure que se llama Azure Repos, primero vamos a Azure Repos y se comprueba que esté utilizando Git que es uno de los sistemas de control de versiones más utilizados últimamente.

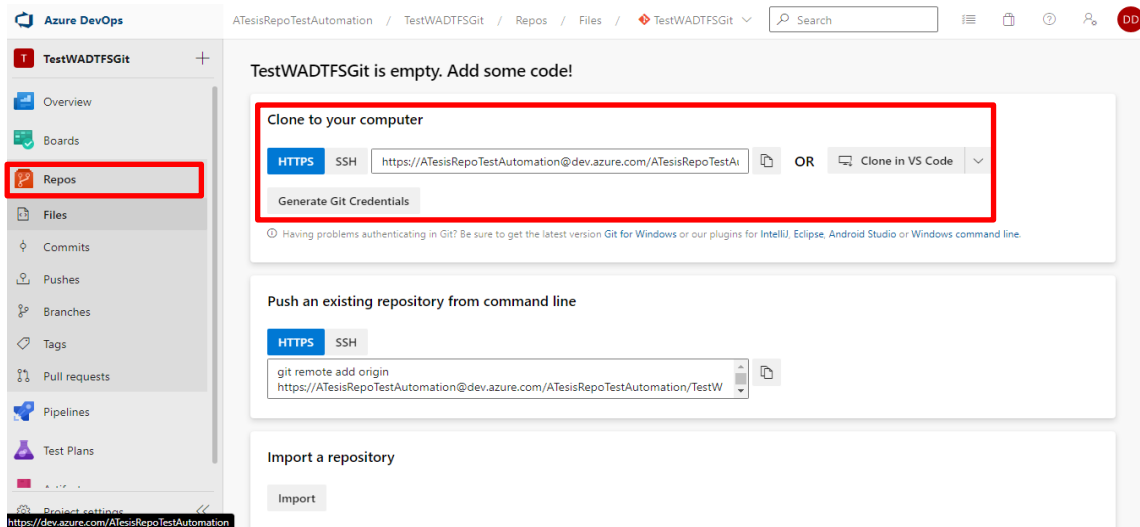


Figura 45. Sistema de control de versiones. [Elaboración Propia].

El siguiente paso es desde la máquina de desarrollo dentro del Visual Studio Community 2019 conectarse de forma directa a la organización que se creó anteriormente al proyecto que es el TestWADTFSGit, para ellos se abre el IDE y se selecciona en primera instancia continuar sin código.

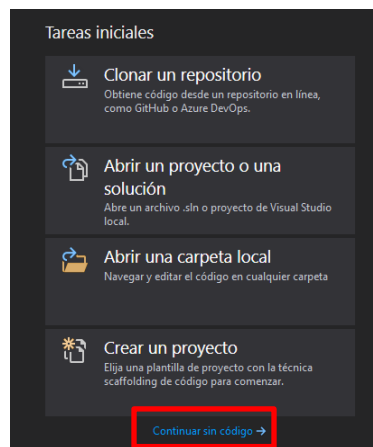


Figura 46. Clonar repositorio. [Elaboración Propia]



Al abrir las distintas opciones del Visual Studio se selecciona la ventana que se llama Team Explorer, se selecciona la opción que dice administrar conexiones, inmediatamente sale la opción de ingresar una cuenta, se asocia la cuenta del Microsoft Azure, por consiguiente se generan las organizaciones y proyectos creados, se selecciona el proyecto luego se da la ruta donde se va a clonar el proyecto y se da clic en la opción de clonar.

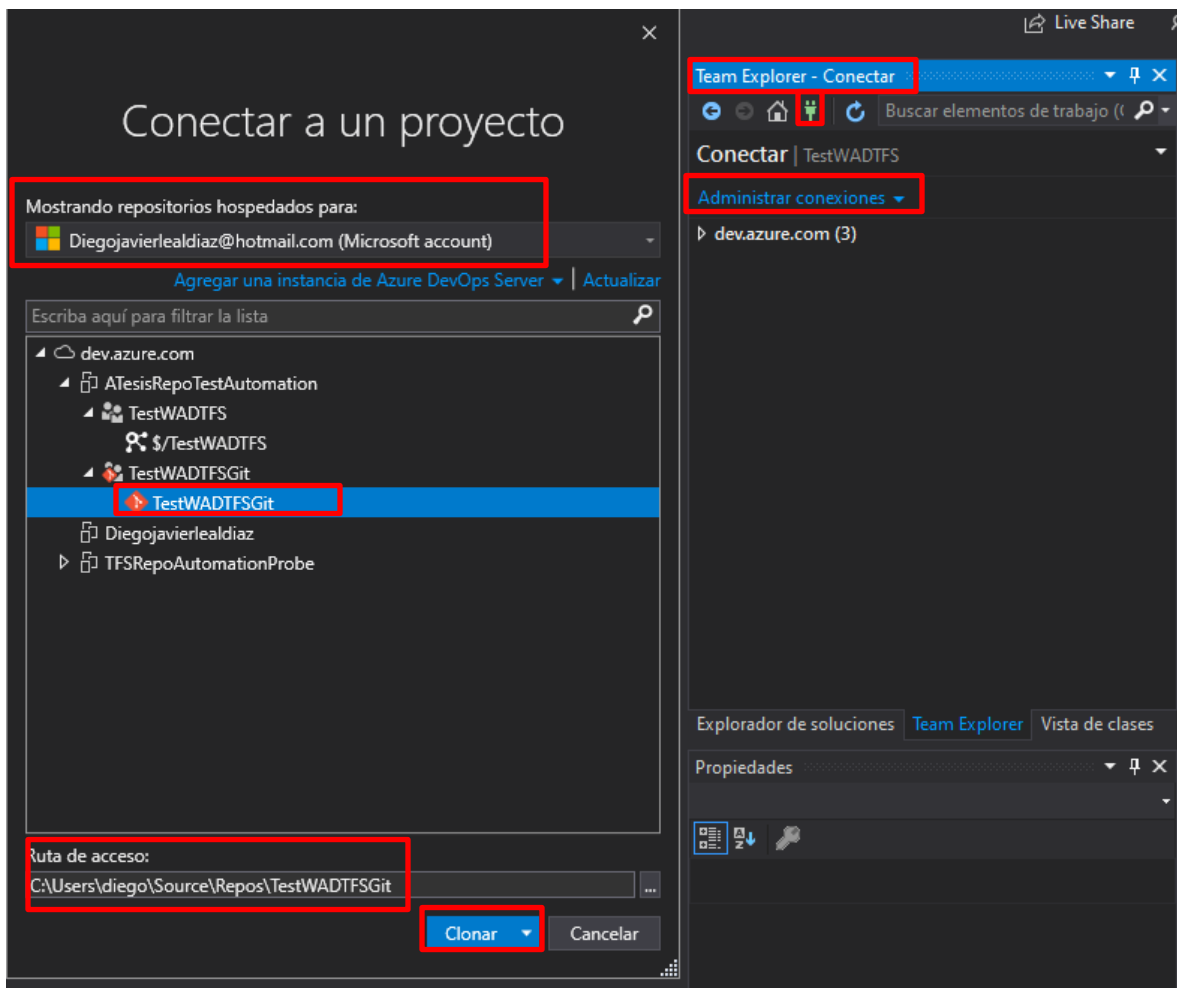


Figura 47. Administrar conexiones Visual Studio. [Elaboración Propia]

El proyecto inmediatamente se clono en la ruta especificada, el siguiente paso es agregar dentro de esa ruta clonada el proyecto con su solución, sus distintos archivos



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



y .dll pero se debe tener en cuenta de que no se le debe pasar la carpeta de paquetes ya que estos paquetes se instalan de manera automática al ejecutarse el Pipeline.

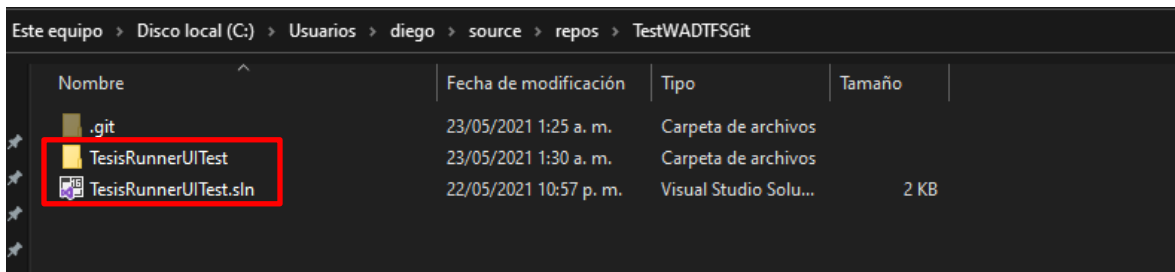


Figura 48. Ingreso del proyecto y la solución al repositorio local. [Elaboración Propia]

Si se retorna al Visual Studio se puede observar que ya se tiene cargada la solución y se procede a compilar, para ello se debe seleccionar el .sln y en el menú de herramientas en la parte superior existe una opción llamada compilar, por tanto, se da en esa opción y queda listo el proyecto para subirlo al repositorio.

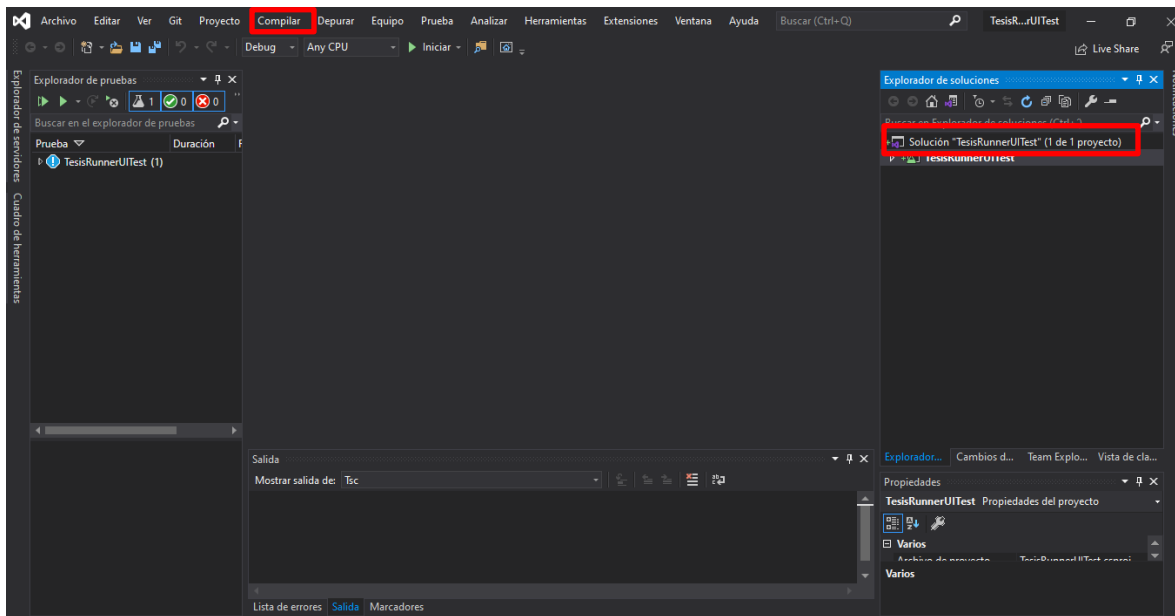


Figura 49. Visualizando solución en el Visual Studio. [Elaboración Propia]

Para subir el código al repositorio se debe seleccionar en el explorador de soluciones la opción que dice Cambios de Git, cuando se selecciona esa carpeta inmediatamente deben salir los archivos del proyecto que se agregaron al repositorio local, se verifica de que lo que vaya a subir al repositorio sea correcto, se selecciona la opción de



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



confirmar todo y subir, se agrega un comentario respectivo y finalmente se da clic en la opción de la parte superior derecha en la flecha en dirección hacia arriba.

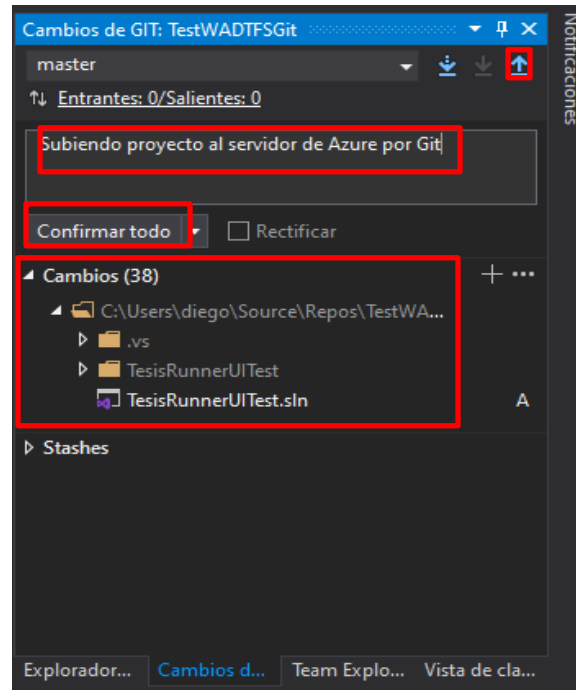


Figura 50. Subir cambios al servidor de Azure.
[Elaboración Propia]

Por último, confirmamos que los cambios se hallan subido de forma correcta al repositorio en el servidor Azure.

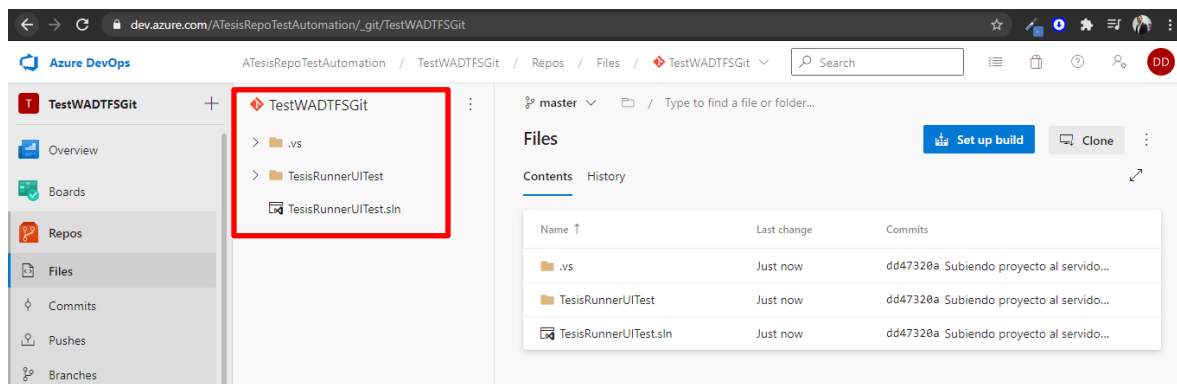


Figura 51. Validando cambios en el repositorio de Azure. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



El paso a seguir es la creación del Pipeline dentro de Microsoft Azure para poder ejecutar las pruebas a automatización de grosso modo de una forma automatizada, primero se selecciona la opción del Pipeline y se da en new Pipeline y se procede a seleccionar la ubicación del código como se observa en la siguiente figura.

New pipeline

Where is your code?



Azure Repos Git YAML

Free private Git repositories, pull requests, and code search



Bitbucket Cloud YAML

Hosted by Atlassian



GitHub YAML

Home to the world's largest community of developers



GitHub Enterprise Server YAML

The self-hosted version of GitHub Enterprise



Other Git

Any generic Git repository



Subversion

Centralized version control by Apache

[Use the classic editor to create a pipeline without YAML.](#)

Figura 52. Selección ubicación del código. [Elaboración Propia]

Como se puede observar en la figura anterior existen diferentes opciones para seleccionar la ubicación del código o repositorio, actualmente ese repositorio se encuentra ubicado en Azure Repos Git pero si se ingresa en esa opción se debe configurar el pipeline en un archivo YAML, para facilitar un poco más la creación de este se selecciona el uso del editor clásico, el cual va a permitir generar los distintos pasos de una manera más gráfica e interactiva para llevar a cabo el proceso de compilación.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Se selecciona la fuente del repositorio para este caso utilizados el sistema de control de versiones Git.

Select a source

Azure Repos Git	TFVC	GitHub	GitHub Enterprise Server	Subversion	Bitbucket Cloud	Other Git
-----------------	------	--------	--------------------------	------------	-----------------	-----------

Team project

TestWADTFSGit

Repository

TestWADTFSGit

Default branch for manual and scheduled builds

master

Continue

Figura 53. Selección del repositorio en Pipeline. [Elaboración Propia]

Debido a que el proyecto de automatización que se trabaja se basa en C# bajo la tecnología .NET se procede a crear una plantilla .NET Desktop como se muestra a continuación.

Featured

.NET Desktop Build and test a .NET or Windows classic desktop solution.	Apply
Android Build, test, sign, and align an Android APK.	
ASP.NET Build and test an ASP.NET web application.	

Figura 54. Generando plantilla .NET Desktop. [Elaboración Propia]

Se selecciona el trabajo del agente 1 y se establece directamente el Pool de agentes que se configuro en el procedimiento anterior para que se conecte directamente ese Pipeline al agente que se encuentra dentro de ese grupo.

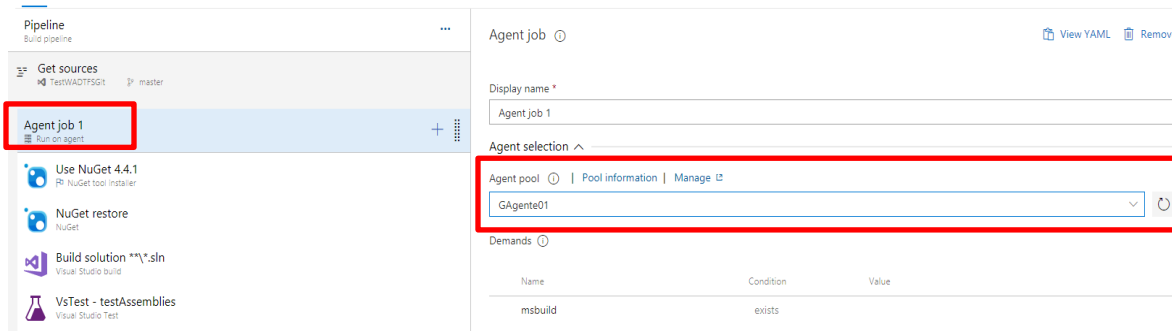


Figura 55. Selección del pool de agentes para el Pipeline. [Elaboración Propia]

Se debe tener muy claro que un Pipeline es un conjunto de tareas que se enumeran de arriba hacia abajo, las cuales se realizan de forma secuencial se configura el nombre de la tubería y nuevamente se debe seleccionar el pool de agentes.

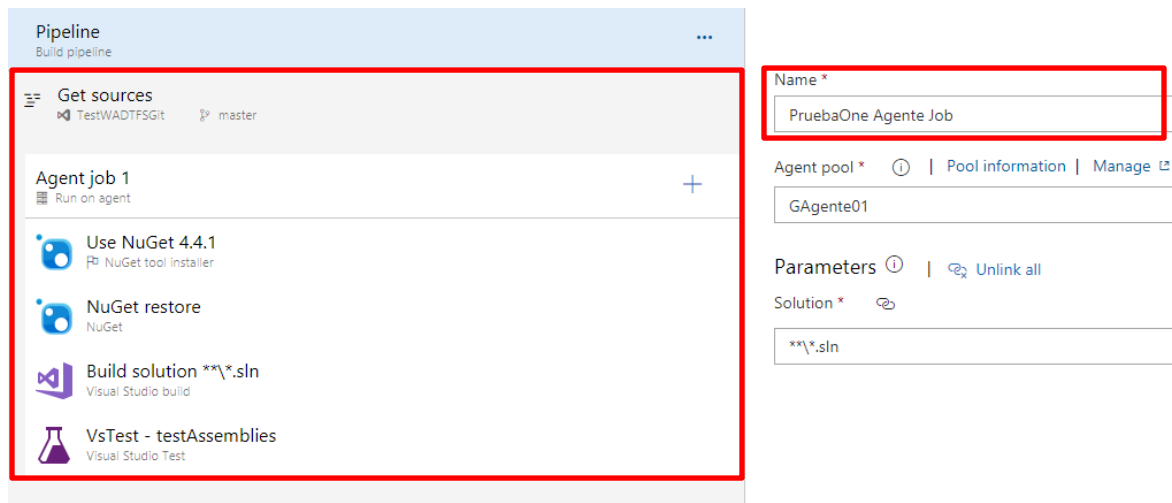


Figura 56. Nombre de la tubería. [Elaboración Propia]

El pool de agentes se puede definir como un conjunto de máquinas que se van a utilizar para ejecutar sus pipelines de construcción.



La primera tarea de todo este proceso es **obtener fuentes** y lo hace en **Get Source** y permite que la máquina virtual donde se encuentra instalado el agente clone el repositorio dentro de ella.

Después de eso, existen dos tareas Nuget para instalar el Nuget y para instalar todos los paquetes relacionados, es por esta razón que el código no se clono al repositorio ya que con esa sección que ejecuta el pipeline se restauran los paquetes.

El siguiente paso que realiza es construir la solución, esto permitirá construir cualquier solución que esté disponible en el código fuente que recupera del servidor y finalmente viene el proceso de ejecución de pruebas como tal, de acuerdo a la configuración de la figura 57 va a permitir ejecutar las pruebas de todos los archivos .dll que contenga la palabra Test en su nombre.

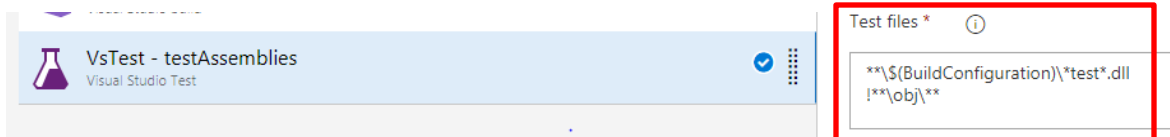


Figura 57. Configuración del Test Assemblies. [Elaboración Propia]

Una vez guardado ese Pipeline, este estará integrado a la máquina virtual con la plataforma Microsoft Azure, logrando el cumplimiento del tercer objetivo y a pocos pasos de completar el segundo objetivo.

Se debe tener en cuenta que para poder ejecutar las pruebas en cada máquina virtual esta requiere de un Pipeline de ejecución solo para ella, esto va posibilitar dar cumplimiento a una de las grandes reglas utilizadas en el mundo de las pruebas de software y esto indica que entre menor sea el número de pruebas que se ejecuten en un mismo plan o en una misma máquina es mucho mejor ya que primero posibilita encontrar ciertos errores mucho más sencillo, si las pruebas se llegan a estallar por algún factor externo y si toca llevar a cabo un proceso de recompilación no se tenga que ejecutar todas las pruebas nuevamente.

Para finalizar el proceso de virtualización se hace necesario primero guardar el snapshot de la máquina, teniendo en cuenta de que dentro de la máquina se configuro todo lo que se necesitaba se procede a dejar abierto el WinAppDriver y ejecutar los comandos que muestra la siguiente figura para el proceso de guardar el estado de esa máquina.

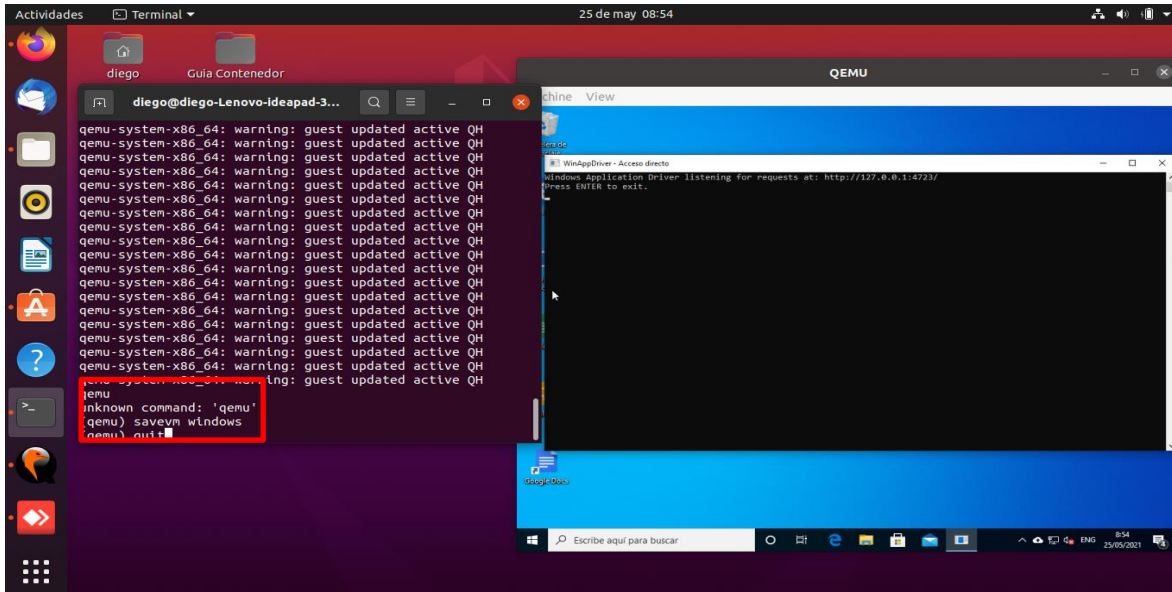


Figura 58. Guardando estado de la máquina virtual. [Elaboración Propia]

El siguiente paso es realizar el proceso de ingresar el snapshot creado anteriormente dentro de un contenedor, esto permite que se puedan ejecutar las máquinas de manera paralela para la compilación de las pruebas.

Para la creación de los contenedores se creó un archivo Dockerfile que permite enviar algunas instrucciones para su configuración y al mismo tiempo instalar las dependencias necesarias.

```

1 FROM ubuntu:20.04
2 RUN apt update && \
3     apt -y upgrade && \
4     apt -y install xvfb x11-utils x11vnc qemu && \
5     apt -y install qemu-system-x86
6
7 COPY hdd.img /
8 COPY snapshot.img /
9 COPY entrypoint.sh /
10 ENTRYPOINT ["/entrypoint.sh"]

```

Figura 59. Archivo Dockerfile. [Elaboración Propia].



Antes de ejecutar el archivo Dockerfile, se debe descargar la imagen de Ubuntu 20.04, como observamos en ese archivo primero llama a la imagen en la cual se basa para crear el contenedor, después de ellos instala las dependencias que ese contenedor va a necesitar para poder ejecutar la máquina virtual dentro de él, aquí se instaló la herramienta Xvfb (Virtual Frame Buffer) la cual permite ejecutar cualquier programa de software en una capa oculta o virtualizada.

Esto va a permitir que no se necesite una pantalla o display directamente, posibilitando al mismo tiempo la ejecución de las máquinas virtuales dentro de contenedores Docker.

Se puede observar que dentro de ese contenedor también se copia la imagen hdd.img, el snapshot capturado y finalmente se le introduce un script de bash llamado entrypoint.sh este archivo se genera para que se ejecute de una manera automática el snapshot de la máquina virtual y al mismo tiempo se le aplique el proceso de ejecución en memoria.

```
1 #!/bin/bash
2 SCREEN_RESOLUTION=${SCREEN_RESOLUTION:-"1024x768x24"}
3 DISPLAY_NUM=99
4 export DISPLAY=":$DISPLAY_NUM"
5
6 clean() {
7     if [ -n "$XVFB_PID" ]; then
8         kill -TERM "$XVFB_PID"
9     fi
10    if [ -n "$X11VNC_PID" ]; then
11        kill -TERM "$X11VNC_PID"
12    fi
13 }
14
15 trap clean INT TERM
16
17 xvfb-run -l -n $DISPLAY_NUM -s "-ac -screen 0 $SCREEN_RESOLUTION -noreset -listen tcp" \
18 qemu-system-x86_64 -enable-kvm \
19     -machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
20     -usb -device usb-kbd -device usb-tablet -rtc base=localtime \
21     -net nic,model=virtio -net user,hostfwd=tcp::4444-:4444 \
22     -drive file=snapshot.img,media=disk,if=virtio \
23     -loadvm windows &
24
25 XVFB_PID=$!
26
27 retcode=1
28 until [ $retcode -eq 0 ]; do
29     xdpinfo -display $DISPLAY >/dev/null 2>&1
30     retcode=$?
31     if [ $retcode -ne 0 ]; then
32         echo Waiting xvfb...
33         sleep 1
34     fi
35 done
36
37 x11vnc -display $DISPLAY -passwd selenoid -shared -forever -loop500 -rfbport 5900 -rfbportv6 5900 -logfile /dev/null &
```

Figura 60. Script bash activación máquina virtual dentro de contenedor. [Elaboración propia]

En el bash anterior se observó la realización de un proceso de limpieza del servidor que ejecuta todas las operaciones gráficas en memoria, después se procede a ejecutar el



ultimo snapshot de la máquina virtual aplicándole el proceso de ejecución con XVFB, cabe resaltar que se debe configurar con los parámetros correctos para la ejecución del pipeline por cada máquina virtual manda su proceso a segundo plano y ejecuta sencillamente la máquina.

Se hace necesario agregar el archivo Dockerfile y el entrypoint.sh dentro de la misma carpeta en la cual se encuentra el hdd.img y el snapshot.img.

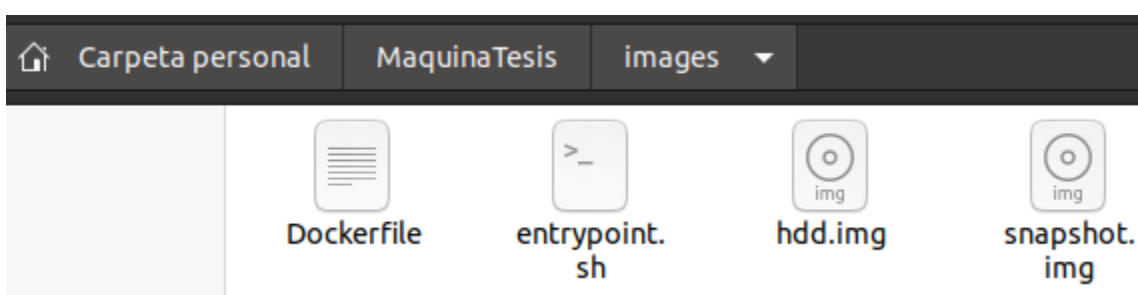


Figura 61. Ingresando Dockerfile y entrypoint.sh para ejecución del contenedor. [Elaboración Propia]

Se procede a instalar la imagen de Ubuntu 20.04 que permitirá construir el contenedor con el archivo Dockerfile.

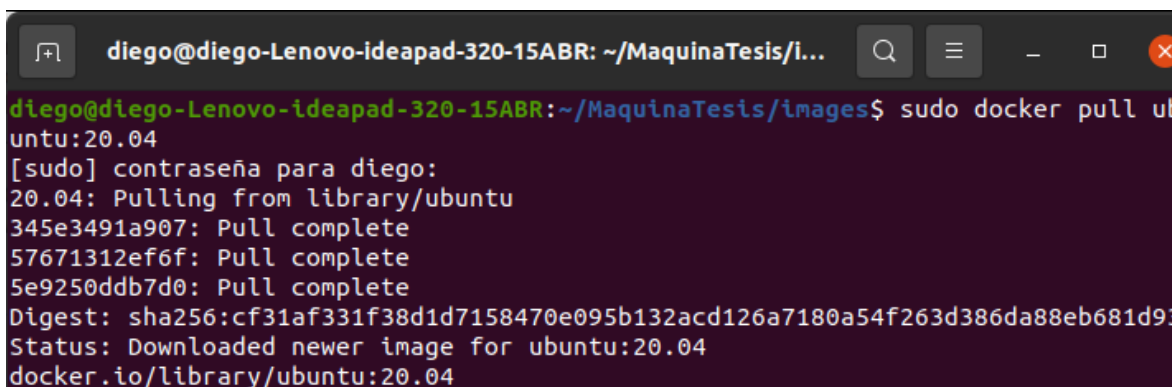


Figura 62. Obteniendo imagen de Ubuntu. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Se construye la nueva imagen del contenedor, partiendo del archivo Dockerfile

```
diego@diego-Lenovo-ideapad-320-15ABR: ~/MaquinaTesis/i...
diego@diego-Lenovo-ideapad-320-15ABR:~/MaquinaTesis/images$ sudo docker build -t windowscontainer .
Sending build context to Docker daemon 1.778GB
```

Figura 63. Construyendo imagen del contenedor. [Elaboración Propia]

Finalmente se ejecuta el contenedor Docker dándole todos los privilegios existentes, teniendo en cuenta los puertos asignados a la máquina virtual y al mismo tiempo se le agregan los puertos por donde uno quiere que sea publicado ese contenedor.

```
diego@diego-Lenovo-ideapad-320-15ABR: ~/MaquinaTesis/i...
diego@diego-Lenovo-ideapad-320-15ABR:~/MaquinaTesis/images$ sudo docker run -it --rm --privileged -p 4444:4444 -p 5902:5902 windowscontainer
Waiting xvfb...

--- x11vnc loop: 1 ---

--- x11vnc loop: waiting for: 36

PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
```

Figura 64. Ejecución del contenedor. [Elaboración propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Inmediatamente se iniciará la ejecución del contenedor con el snapshot pero no podrá ser visualizada ya que se está ejecutando en memoria.

Para automatizar de una manera más sencilla se crearon 2 archivos más de script de bash que permiten crear una nueva carpeta tal y como se apreció en la figura 61 e introduce el archivo Dockerfile y el entrypoint, esto pasa los archivos .img que se pueden observar allí y lanza el snapshot de la máquina virtual.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~  
❯ /bin/bash  
#Script para ejecutar el estado del Snapshot que se creo para la máquina virtual  
  
echo "Bienvenidos al Script para ejecutar el estado del Snapshot de la VM" | boxes -d tan_jones  
  
#Ejecutando el comando que permite correr el estado del snapshot  
  
echo -e "\n"  
echo "Ejecutando Snapshot" | boxes -d stone  
  
echo -e "\n"  
read -p "Por favor ingrese el nombre de la carpeta donde creo la Máquina Virtual y el SnapShot anteriormente: " Carpeta  
echo -e "\n"  
read -p "Por favor digite el mismo puerto con el que fue creado la máquina virtual y el snapshot en los Script anteriores: " puerto  
  
echo -e "\n"  
echo "Creando directorio y moviendo archivos" | boxes -d stone  
echo -e "\n"  
echo "Creando directorio..."
```

Figura 65. Tercer Script automatizado ejecutar snapshot. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750


```

diego@diego-Lenovo-Ideapad-320-15ABR: ~
echo -e "\n"
echo "Creando directorio y moviendo archivos" | boxes -d stone
echo -e "\n"
echo "Creando directorio..."

#Creando directorio
cd $Carpeta/
mkdir -m 755 images

#Copiando archivos
echo -e "\n"
echo "Moviendo archivos..."
mv /home/diego/$Carpeta/hdd.img /home/diego/$Carpeta/images
mv /home/diego/$Carpeta/snapshot.img /home/diego/$Carpeta/images
#Copiando archivos
echo -e "\n"
echo "Copiando archivos..."
cd /home/diego/ImagenesISOS
cp /home/diego/ImagenesISOS/Dockerfile /home/diego/$Carpeta/images
cp /home/diego/ImagenesISOS/entrypoint.sh /home/diego/$Carpeta/images

cd /home/diego/$Carpeta/images
chmod 755 entrypoint.sh
chmod 755 hdd.img
chmod 755 snapshot.img
chmod 755 Dockerfile

#Ejecutando máquina
sudo qemu-system-x86_64 -enable-kvm \
-m machine q35 -smp sockets=1,cores=1,threads=2 -m 2048 \
-usb -device usb-kbd -device usb-tablet -rtc base=localtime \
-net nic,model=virtio -net user,hostfwd=tcp::$puerto:$puerto \
-drive file=snapshot.img,media=disk,if=virtio \
-loadvm windows

-- INSERTAR --

```

Figura 66. Continuación Tercer Script automatizado ejecutar snapshot. [Elaboración Propia]

El script requiere que le ingrese obligatoriamente el nombre de la carpeta principal donde creo por primera vez la máquina virtual y el puerto, de resto el script inmediatamente le carga una nueva carpeta y le ingresa la información que necesita para ejecutar el contenedor.

```

diego@diego-Lenovo-Ideapad-320-15ABR:~$ bash createvirtualmachine5.sh
      \\\|\\|
      (| (.) (.) |)
      .000o--()--o000.
-----
| Bienvenidos al Script para ejecutar el estado del Snapshot de la VM |
-----
      .000o.  000o.
      (| (.) (.) |)
      .000o--()--o000.

+-----+
| Ejecutando Snapshot |
+-----+

Por favor ingrese el nombre de la carpeta donde creo la Máquina Virtual y el Snapshot anteriormente: MaquinaTestis
Por favor digite el mismo puerto con el que fue creado la máquina virtual y el snapshot en los Script anteriores: 444

+-----+
| Creando directorio y moviendo archivos |
+-----+
:creando directorio...

```

Figura 67. Ejecución tercer script de bash para el proceso de virtualización. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Finalmente se procede a ejecutar el script de crear contenedores de forma automatizado.

```
#!/bin/bash
#Script para la creación del contenedor de la máquina virtual y ejecución del mismo
#Autor: Diego Leal

echo "Bienvenido al Script para la creación y el despliegue del contenedor Docker" | boxes -d ian_jones

echo -e "\n"
read -p "Por favor ingrese el nombre de la carpeta en la que creo la máquina virtual: " Carpeta
echo -e "\n"
read -p "Por favor ingrese el puerto con el cual creo la máquina virtual: " puertovm
echo -e "\n"
read -p "Por favor ingrese el puerto por el cual va a ser publicado el contenedor Docker: " puertopublic
echo -e "\n"
read -p "Por favor ingrese el nombre que le quiere dar al contenedor Docker: " nombrecontenedor

cd /home/diego/$Carpeta/images

echo -e "\n"
echo "Creando el contenedor..."
#Descargamos la imagen del contenedor de Ubuntu
sudo docker pull ubuntu:20.04

#Construimos el contenedor
sudo docker build -t $nombrecontenedor .

#Ejecutando el contenedor
sudo docker run -it --rm --privileged -p $puertovm:$puertovm -p $puertopublic:$puertopublic $nombrecontenedor
```

Figura 68. Ejecución script container. [Elaboración Propia]

Tal y como se observa, este script pide por teclado el nombre de la carpeta donde se encuentra la imagen de la máquina virtual, el puerto donde se ejecutó esa máquina, el puerto por donde desea publicar el contenedor, el nombre del contenedor y el inmediatamente descarga la imagen de Ubuntu, crea el contenedor y lo lanza.

```
diego@diego-Lenovo-Ideapad-320-15ABR:~$ bash createcontainer.sh
      \\\|\\|
      ( | ( ) ( ) | )
-----o000o--( )--o000o-----
|
| Bienvenido al Script para la creación y el despliegue del contenedor Docker |
|
|-----o000o-----
|
| ( ) ( )
| \ ) \ )
|  \ )  \ )

Por favor ingrese el nombre de la carpeta en la que creo la máquina virtual: MaquinaTesis
Por favor ingrese el puerto con el cual creo la máquina virtual: 4444
Por favor ingrese el puerto por el cual va a ser publicado el contenedor Docker: 5902
Por favor ingrese el nombre que le quiere dar al contenedor Docker: containerwindows
```

Figura 69. Ejecución cuarto script del proceso de containerización. [Elaboración Propia].



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Es posible afirmar que una vez desarrolladas las dos fases anteriores se le da cumplimiento al segundo y tercer objetivo el cual es realizar el proceso de virtualización e integración con la plataforma Microsoft Azure.

5.3.2. FASE 4: Validación del sistema

La ejecución de pruebas en esta nueva estructura desarrollada en los pasos anteriores representa una ventaja gigantesca con respecto al modo de ejecución de las pruebas actualmente en Digital Ware, esto debido a que permitirá liberar las máquinas de desarrollo y facilitará un trabajo de forma paralela logrando así una reducción de tiempo al momento de ejecución.

Para la validación de este sistema se realizó una prueba directamente con un programa de Digital Ware, la estructura de este programa es idéntica a cada uno de los programas en los cuales se aplican pruebas dentro del área, dentro de esta pequeña validación se realizó un login respectivo el cual traerá algún registro de la base de datos que se encuentra dentro de una opción llamada Qbe para descargarlo, esto para validar funcionalidades anteriores, en caso de que no se logre realizar el login está como variante abrir la aplicación de la calculadora de Windows y realice una prueba pequeña de una operación.

Para llevar a cabo esta prueba de validación se desarrolló el siguiente Test, permitiendo revisar de que el sistema se encuentra funcionando de manera correcta y haga lo que tenga que hacer.

En primera instancia se tiene una clase de prueba unitaria llamada ConsultoriaTest, esta clase de prueba unitaria va a permitir realizar distintos métodos de prueba en los escenarios de la aplicación que se quiere llevar cabo el test y a modo de ejemplo del funcionamiento de una prueba se creó el método de prueba AbrirConsultoria, el cual permite abrir el producto de consultoría de Digital Ware que es utilizado para descargar distintos casos de errores de script de bases de datos como SQL Server y ORACLE para poderlos impactar en cada motor.

Dentro del producto se hace necesario realizar un proceso de login con credenciales que da internamente la compañía lo cual se observa a continuación que lo realice de forma correcta, se realiza un proceso de traer el caso desde una base de datos llamado QBE y finalmente se realiza la respectiva prueba de que descargue el caso en la máquina.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



```
[TestClass]
- referencias
public class ConsultoriaTest
{
    [TestMethod]
    - referencias
    public void AbrirConsultoria()
    {
        WindowsDriver<WindowsElement> session = null;
        AppiumOptions appOptions = new AppiumOptions();
        appOptions.AddAdditionalCapability("app", @"C:\Program Files (x86)\Internet Explorer\iexplore.exe");
        appOptions.AddAdditionalCapability("appArguments", @"http://consultoria.digitalware.co/");
        try
        {
            session = new WindowsDriver<WindowsElement>(new Uri("http://localhost:4723/"), appOptions, TimeSpan.FromSeconds(60));
        }
        catch
        {
            session = new WindowsDriver<WindowsElement>(new Uri("http://localhost:4723/"), appOptions, TimeSpan.FromSeconds(60));
        }
        session.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(15);
        Thread.Sleep(2000);
        session.Manage().Window.Maximize();
        Thread.Sleep(2000);
        var ApplicationWindow = session.FindElementByClassName("IEFrame");
        var ApplicationSessionHandle = ApplicationWindow.GetAttribute("NativeWindowHandle");
        ApplicationSessionHandle = (int.Parse(ApplicationSessionHandle)).ToString("x");
        Thread.Sleep(7000);
        LogIn(session, "kauto1", ██████████);
        Thread.Sleep(2000);
    }
}
```

Figura 70. Método de prueba *AbrirConsultoria*. [Elaboración Propia]

Se crea el método para que realice el login y se ingresa la segunda opción de abrir la calculadora en caso de que el producto falle al abrir ya que es un producto bastante pesado y debido a que no se encuentra la máquina internamente dentro de la red de la compañía genera ciertos problemas que afectan directamente a la prueba de automatización.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



```
1 referencia | 1/1 pasando
public static void LogIn(WindowsDriver<WindowsElement> desktopSession, string user, string password)
{
    var wait = new DefaultWait<WindowsDriver<WindowsElement>>(desktopSession)
    {
        Timeout = TimeSpan.FromSeconds(120),
        PollingInterval = TimeSpan.FromSeconds(1)
    };
    wait.IgnoreExceptionTypes(typeof(InvalidOperationException));
    //Busca la ventana del login
    WindowsDriver<WindowsElement> rootSession = null;
    rootSession = RootSession();
    rootSession = ReloadSession(rootSession, "TSAcAccesFrm");
    if (rootSession != null)
    {
        //Realiza el login
        var editFields = rootSession.FindElementsByClassName("TEdit");
        editFields[0].Clear();
        editFields[0].SendKeys(user);
        editFields[1].Clear();
        editFields[1].SendKeys(password);
        Screenshot image = ((ITakesScreenshot)rootSession).GetScreenshot();
        image.SaveAsFile(string.Format("Login" + Hora() + ".png", ScreenshotImageFormat.Png));
        Thread.Sleep(2000);
        var Aceptar = rootSession.FindElementsByName("Aceptar");
        Aceptar[0].Click();
        Thread.Sleep(10000);
        Screenshot images = ((ITakesScreenshot)desktopSession).GetScreenshot();
        images.SaveAsFile(string.Format("Carga" + Hora() + ".png", ScreenshotImageFormat.Png));
        Thread.Sleep(2000);
        //Llama al Qbe
        QbeQry(desktopSession);
    }
}
```

Figura 71. Método para el proceso de login. [Elaboración Propia]

En caso de que no pueda hacer el login se realiza el proceso de abrir la calculadora de Windows.

```
}
else
{
    desktopSession.Close();
    //Abre la calculadora de Windows
    WindowsDriver<WindowsElement> sessionCalc = null;
    AppiumOptions appCapabilities = new AppiumOptions();
    appCapabilities.AddAdditionalCapability("app", "Microsoft.WindowsCalculator_8wekyb3d8bbwe!App");

    sessionCalc = new WindowsDriver<WindowsElement>(new Uri("http://127.0.0.1:4723"), appCapabilities, TimeSpan.FromSeconds(60));

    sessionCalc.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(2);

    System.Threading.Thread.Sleep(TimeSpan.FromSeconds(3));
    Screenshot Calculadora = ((ITakesScreenshot)sessionCalc).GetScreenshot();
    Calculadora.SaveAsFile(string.Format("Calculadora" + Hora() + ".png", ScreenshotImageFormat.Png));
    Thread.Sleep(1000);
    TestTwoXTwoIsFourMouseTest(sessionCalc);
}
}
```

Figura 70. Continuación método para el proceso de login. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Se realiza el método para el proceso del QBE y traer el caso de la Base de datos.

```
1 referencia
public static void QbeQry(WindowsDriver<WindowsElement> desktopSession)
{
    var wait = new DefaultWait<WindowsDriver<WindowsElement>>(desktopSession)
    {
        Timeout = TimeSpan.FromSeconds(120),
        PollingInterval = TimeSpan.FromSeconds(1)
    };
    wait.IgnoreExceptionTypes(typeof(InvalidOperationException));
    //Busca icono QBE
    var Element = desktopSession.FindElementByName("tbrOpciones");
    desktopSession.Mouse.MouseMove(Element.Coordinates, 47, 13);
    desktopSession.Mouse.Click(null); //TQbe
    Thread.Sleep(1000);
    WindowsDriver<WindowsElement> rootSession = null;
    rootSession = RootSession();
    rootSession = ReloadSession(rootSession, "TQbe");
    //Ingresa la data en el Qbe
    rootSession.Keyboard.SendKeys("642542");
    Thread.Sleep(1000);
    Screenshot Qbe = ((ITakesScreenshot)rootSession).GetScreenshot();
    Qbe.SaveAsFile(string.Format("Qbe" + Hora() + ".png", ScreenshotImageFormat.Png));
    Thread.Sleep(1000);
    rootSession.Keyboard.SendKeys(OpenQA.Selenium.Keys.Enter);
    Thread.Sleep(2000);
    GuardandoCaso(desktopSession);
}
```

Figura 71. Traer caso desde base de datos. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



Finalmente se crea el método para descargar el script desde el producto.

```
4. referencia
public static void GuardandoCaso(WindowsDriver<WindowsElement> desktopSession)
{
    var wait = new DefaultWait<WindowsDriver<WindowsElement>>(desktopSession)
    {
        Timeout = TimeSpan.FromSeconds(120),
        PollingInterval = TimeSpan.FromSeconds(1)
    };
    wait.IgnoreExceptionTypes(typeof(InvalidOperationException));
    //Clic en ventana aseguramiento de calidad

    var Element = desktopSession.FindElementByName("Aseg. Calidad");
    Element.Click();
    Thread.Sleep(1000);
    Screenshot ACalida = ((ITakesScreenshot)desktopSession).GetScreenshot();
    ACalida.SaveAsFile(string.Format("Aseguramiento de Calidad" + Hora() + ".png", ScreenshotImageFormat.Png));
    Thread.Sleep(1000);

    //Clic en Descargar programa
    var Element1 = desktopSession.FindElementByName("Descargar progr.");
    Element1.Click();
    Thread.Sleep(1000);

    //Seleccionar carpeta
    WindowsDriver<WindowsElement> rootSession = null;
    rootSession = RootSession();
    Thread.Sleep(1000);
    Screenshot Carpeta = ((ITakesScreenshot)rootSession).GetScreenshot();
    Carpeta.SaveAsFile(string.Format("Seleccionar carpeta" + Hora() + ".png", ScreenshotImageFormat.Png));
    Thread.Sleep(1000);
    rootSession.Keyboard.SendKeys(OpenQA.Selenium.Keys.Enter);
    Thread.Sleep(15000);

    WindowsDriver<WindowsElement> rootSession1 = null;
    rootSession1 = RootSession();
    Thread.Sleep(1000);
    Screenshot Accept = ((ITakesScreenshot)rootSession1).GetScreenshot();
    Accept.SaveAsFile(string.Format("Aceptar Descarga" + Hora() + ".png", ScreenshotImageFormat.Png));
    Thread.Sleep(1000);
    rootSession1.Keyboard.SendKeys(OpenQA.Selenium.Keys.Enter);
}
```

Figura 72. Descargar caso desde el maestro. [Elaboración Propia]

El proceso de prueba que se aplicó anteriormente es a modo de ejemplo, dentro del equipo de trabajo el flujo de pruebas y la forma de realizarlas se llevan a cabo de una manera muy parecida, estos procesos permiten observar los distintos problemas funcionales que puede poseer una aplicación a nivel general.



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



CAPÍTULO VI: RESULTADOS

De acuerdo con las necesidades del equipo de automatización en la compañía Digital Ware al realizar el proceso de ejecución de pruebas automatizadas conociendo que estas se ejecutan de manera secuencial en una computadora que funciona como servidor y no permite realizar ninguna acción en paralelo o de manera simultánea, se crea una nueva alternativa que se basa en contenedores, los cuales van a permitir la ejecución de pruebas en un mismo equipo de cómputo permitiendo aplicar el concepto de paralelismo.

Cabe aclarar que para el desarrollo de este proyecto no se pudo integrar directamente con el proyecto del equipo de automatización debido a algunas limitantes que no se pudieron solventar por los lineamientos de la compañía durante el proceso de la pasantía como la no entrega del servidor dentro de la empresa para la realización de la configuración del proceso de virtualización, de la misma forma sabiendo que la plataforma Microsoft Azure es una plataforma de pago por uso, no se logró gestionar el acceso directo de manera rápida a alguna cuenta y dichos recursos fueron aplicados por mi parte y debido a que la compañía no permite un acceso directo a los servidores internos de forma remota por seguridad no se pudo obtener el acceso al proyecto, por tanto se realiza una prueba que permite observar el mismo modo de funcionamiento, teniendo en cuenta de que el proceso de configuración es exactamente el mismo.

Gracias a algunos trabajos realizados por algunos compañeros que han pasado por un proceso de pasantía en la empresa Digital Ware, se logró dar terminación a este proyecto definiendo las tecnologías más adecuadas para su realización, primordialmente y la base fundamental es el uso de un proceso de virtualización con contenedores que permitirá encapsular distintas máquinas virtuales logrando una ejecución en paralelo dentro de un único equipo de cómputo y así mismo utilizando la plataforma Microsoft Azure para llevar a cabo un proceso de integración entre el proceso de virtualización y el poder ejecutar las pruebas de una forma más sencilla y sin tanto trabajo.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



En primera instancia se realizó una validación de que cada una de las máquinas virtuales tuviera acceso a la página donde se va aplicar el proceso de automatización, esto debido a que en algunos casos la versión del Internet Explorer y su configuración interna no posibilita un acceso directo a este producto.

Se realizó el respectivo procedimiento de configuración y se pudo obtener acceso a la página de consultoría de la compañía, pero en distintas pruebas que se llevaron a cabo para abrir el producto se pudo observar que el tiempo de carga de todos los elementos era distinto; esto va muy de la mano con la calidad del internet que cada uno tenga, este tiempo de demora en cargar afecta directamente sobre la prueba de automatización a realizar ya que se encuentra programada para que se ejecuten los distintos métodos cada cierto tiempo determinado.

Por parte de la empresa se recomienda para una correcta compilación que el producto sea abierto directamente desde las máquinas de desarrollo ya que estas se encuentran en la misma red interna, debido a que esto no se pudo contemplar se buscó una alternativa que lograra solventar de cierta forma este problema, es por esto que se codifico una prueba automatizada de tal manera de que si no alcanza a abrir el producto en ese tiempo determinado se debe cerrar la página y se ejecuta una prueba en la calculadora de Windows, esto para confirma de que si se estén ejecutando las pruebas dentro de los contenedores.

Cabe resaltar que si la máquina se encuentra internamente en la compañía ya no presentara este problema porque los tiempos de carga son muy estables debido a la corta trazabilidad que tienen que recorrer los paquetes hasta llegar al servidor.

Después de tener un acceso al producto donde se va a llevar a cabo el proceso de automatización se convirtió en un reto grande la configuración del agente autohospedado de Microsoft Azure en cada máquina, esto se debe a que la configuración que se estaba llevando a cabo necesitaba de la ejecución de un archivo .exe de forma manual para que el agente encendiera y se conectara con la plataforma Azure, esto de una forma directa no iba a funcionar ya que la capa de aplicación de las máquinas será renderizada en memoria y no se tendría un acceso para ejecutar el agente manualmente.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Primero se intentó que el archivo .exe se ejecutara con una tarea programada de Windows cada vez que la máquina iniciara, funciono pero surgió otro problema y este se debe a que se iba a captura un estado de la máquina virtual (snapshot) ese estado cuando se captura guarda el instante en el que fue captado con fecha y hora, por tanto, para que el agente lograra conectarse con el servidor de Azure necesitaba que la fecha y hora se encontrara sincronizada con la del servidor, de no ser así, el agente y el servidor no tendrían una comunicación HTTP entre ellas, por tanto el archivo .exe arrancaba pero el agente no se encendía.

Para dar solución a estos dos grandes problemas anteriores se buscó la manera en primer lugar de que el agente configurado en la máquina se ejecutara como un servicio de Windows y ya no se tendría que programar ninguna tarea para que ejecute el archivo de arranque del agente, lo dicho anteriormente le dio solución al primer problema, pero se seguía presentando el inconveniente de la sincronización.

Inicialmente se buscó sincronizar también con una tarea programada la cual hacía que ejecutara un archivo .bat con una serie de comando para la sincronización, las primeras 12 horas funciono correctamente pero después se generaba un problema por el tiempo ya que la diferencia entre el tiempo que quedo guardado el snapshot y el tiempo actual era muy grande y generaba un error.

Se comenzó a buscar una alternativa por medio de un programa para mantener la hora exacta del sistema y para ello se encontró el programa NetTime el cual es un cliente del protocolo de internet para sincronizar el reloj, este programa se ejecuta como servicio de Windows y se programó para que cada 15 minutos sincronizara el reloj, dando solución a la problemática presente.

<https://www.timesynctool.com/>



Figura 73. Agente Online. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Por medio del proceso de containerización con Docker se busca que la prueba se ejecutara en un entorno totalmente aislado, permitiendo mayor flexibilidad en cuanto a la ejecución de las pruebas y que las máquinas serán dedicadas solamente para ese proceso.

Debido a que todo el proceso se encuentra gestionado por Docker es posible la creación de varios contenedores generados con mayor facilidad por el archivo Dockerfile y de esta manera es posible ejecutar pruebas paralelas, el entorno creado requiere de un Net Time tiempo para la inicialización de los contenedores antes de que pueda ser utilizado, ese proceso es de aproximadamente entre 15 a 20 minutos para que el contenedores arranque y al mismo tiempo que agente se conecte con Azure y pueda estar activo, sin embargo esto es proceso que se realiza solamente una vez ya que dentro de la compañía todas las máquinas están encendidas las 24 horas y después de iniciar por primera vez puede usarse para la ejecución de pruebas consecutivas.

```
diego@diego-Lenovo-ideapad-320-15ABR: ~  
diego@diego-Lenovo-ideapad-320-15ABR:~$ docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
windowspruebacondos latest      3c385c7ec3fa     29 hours ago    22.4GB  
windowscontainersone latest      d6d51a3fb03b     2 days ago      26.8GB  
twowindowscontainers latest      7393819cdb71     3 days ago      25.3GB  
windowscontainer    latest      f36d745ee88a     4 days ago      24.6GB  
ubuntu              20.04      7e0aa2d69a15     5 weeks ago     72.7MB
```

Figura 74. Imagen de los contenedores. [Elaboración Propia]

En la figura anterior es posible visualizar las imágenes de contenedores que se han creado, para este caso en el procedimiento anterior se creó el contenedor para la imagen windowscontainer y al mismo tiempo para ejecución se crea al contenedor con la imagen llamada twowindowscontainers para la creación de este nuevo contenedor se realiza exactamente el mismo procedimiento que en todos los pasos anteriores.

Para el proceso de ejecución de contenedores se necesita que el entorno Host donde se ejecute sea una distribución Linux, ya que brinda con mayor facilidad herramientas para el proceso de virtualización y containerización, especialmente para la implementación de la herramienta Xvfb utilizada para la renderización de procesos en segundo plano.



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



A continuación se visualiza el resultado del lanzamiento de dos contenedores.

```
diego@diego-Lenovo-Ideapad-320-15ABR: ~/MaquinaTesis/1...
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaTesis/images$ sudo docker run -it
--rm --privileged -p 4444:4444 -p 5902:5902 windowscontainer
[sudo] contraseña para diego:
Waiting xvfb...

--- x11vnc loop: 1 ---

--- x11vnc loop: waiting for: 36

PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.

diego@diego-Lenovo-Ideapad-320-15ABR: ~/MaquinaTesis02...
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaTesis02/images$ sudo docker run -i
t --rm --privileged -p 4445:4445 -p 5903:5903 twowindowscontainers
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...

--- x11vnc loop: 1 ---

--- x11vnc loop: waiting for: 40

PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.
█
```

Figura 75. Lanzamiento dos contenedores. [Elaboración Propia]

Actualmente al ejecutar pruebas de automatización en el servidor, este tomaba el control completo sobre el equipo permitiendo que estas se ejecutarán de manera secuencial irrumpiendo en el trabajo diario por parte de los ingenieros, sin embargo, con la implementación de este proyecto cambiaría todo ya que es posible la ejecución de grupos de pruebas dentro de los contenedores Docker y no en la máquina de trabajo como tal.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Cuando se lanza el contenedor y se verifica en Azure que el agente se encuentra activo se ejecuta el Pipeline que se asocia al pool agent donde se creó el respectivo agente.

The screenshot displays the Azure DevOps interface for a pipeline named 'PruebaOne Agente Job'. The left sidebar shows navigation options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Project settings. The main area shows 'Jobs in run #20210...' with a list of steps: Initialize job (23s), Checkout T... (1m 27s), Use NuGet 4.4.1 (26s), NuGet restore (2m 0s), Build solutio... (2m 5s), VsTest - te... (7m 13s), Post-job: Check... (8s), Finalize Job (<1s), and Report build st... (<1s). The 'Agent job 1' step is expanded, showing a terminal window with the following output:

```
1 Pool: GAgente01
2 Queued: Today at 6:43 p. m. [manage_parallel_jobs]
3 Agent: DWPUEBAAGENTE01
4 Started: Today at 6:43 p. m.
5 Duration: 13m 54s
6
7 Waiting on the agent to receive and acknowledge the request...
8 Job preparation parameters
9 3 queue time variables used
10 100% tests passed
11 Job live console data:
12
diego@diego-Lenovo-Ideapad-320-15ABR: ~/MaquinaTesis/ln...
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaTesis/lnages$ sudo docker run -it
--rm --privileged -p 4444:4444 -p 5902:5902 windowscontainer
[sudo] contraseña para diego:
Waiting xvfb...
--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 36
PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
```

Figura 76. Ejecución Pipeline 1. [Elaboración Propia]

Es posible observar que la ejecución de la canalización para el contenedor 1 se ejecutó correctamente.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Es necesario resaltar que durante el proceso de ejecución de las pruebas no se podrá observar nada de la capa de aplicación ya que esta se encontrará renderizada en memoria.

The screenshot displays a CI/CD pipeline execution interface. On the left, a list of jobs is shown for 'PruebaTwo Agente 2'. The main part of the image shows the details for 'Agent job 2', which is completed. The terminal output shows the execution of a Docker container with the following steps:

```
1 Pool: GAgente02
2 Queued: Today at 9:44 p. m. [manage_parallel_jobs]
3 Agent: DWPRUEBA02
4 Started: Today at 9:45 p. m.
5 Duration: 10m 17s
6
7 Waiting on the agent to receive and acknowledge the request...
8 Job preparation parameters
9 6. 3 queue time variables used

diego@diego-Lenovo-Ideapad-320-15ABR: ~/MaquinaTesis02...
diego@diego-Lenovo-Ideapad-320-15ABR:~/MaquinaTesis02/images$ sudo docker run -i
t --rm --privileged -p 4445:4445 -p 5903:5903 twowindowscontainers
[sudo] contraseña para diego:
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...

--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 44

PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreading(2)
Please configure -smp options properly or try enabling topoext feature.
```

Figura 77. Ejecución Pipeline 2. [Elaboración Propia]

Para asegurar de que las pruebas se están ejecutando de forma correcta se hace necesario la toma de capturas de pantalla durante el proceso de automatización, el reto era como poder sacar estas capturas de la máquina puesto que las pruebas se ejecutaban en memoria y no la podíamos visualizar, esto con llevó a sincronizar una carpeta de la máquina virtual con un drive de algún correo específico y por el correo podíamos visualizar las respectivas capturas de pantalla del proceso de ejecución de las pruebas.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750

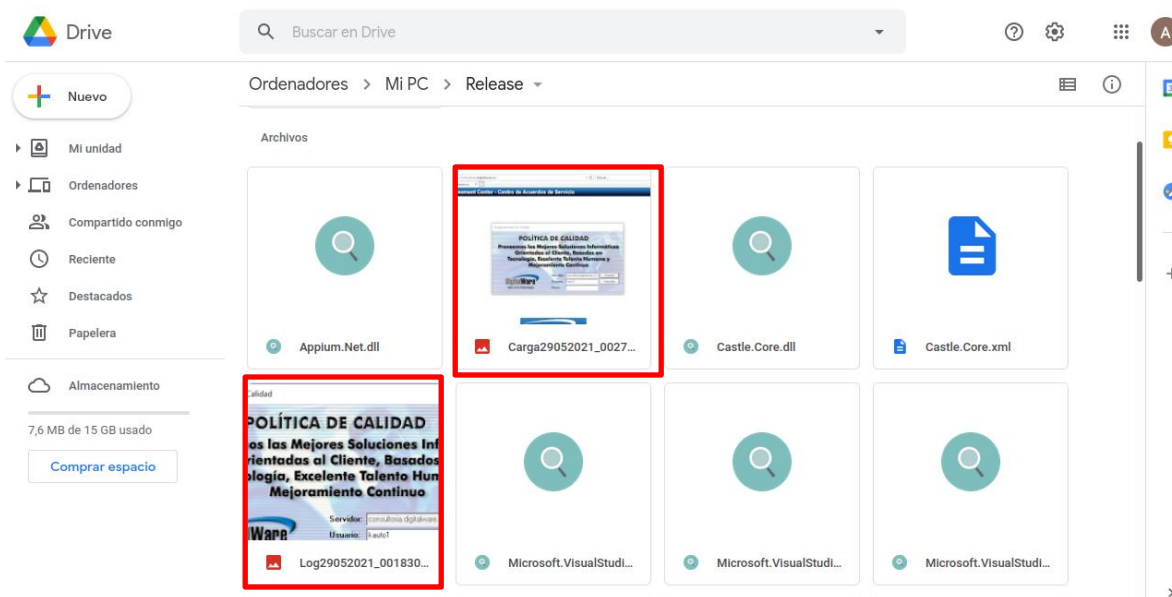


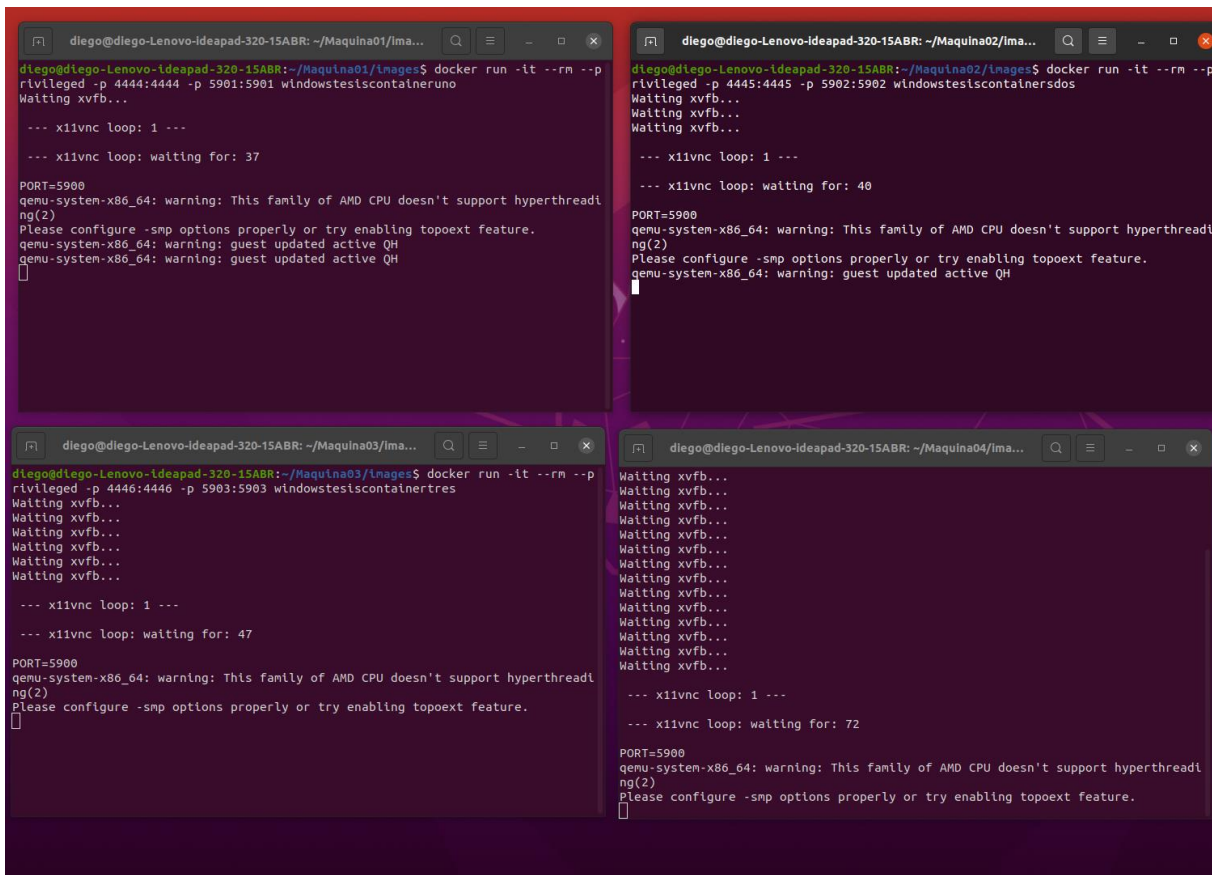
Figura 78. Visualización de capturas del proceso. [Elaboración Propia]

Se busco la manera de ejecutar el mayor número de contenedores en paralelo posibles dentro de un equipo de cómputo Lenovo con memoria RAM DDR4 de 16 GBytes y especificaciones AMD A12-9720P, RADEON R7, COMPUTE CORES 4C+8G, durante este procedimiento existieron algunos factores que afectaron directamente sobre el proceso de ejecución y el principal es el no contar con una suscripción de Azure paga y esto no permitía ejecutar más de una canalización en paralelo al tiempo, por tanto, se recurre a crear una organización con un proyecto de tipo público, para este modo de funcionamiento fue posible ejecutar más de una pipeline de forma paralela pero el modo de ejecución es mucho más lento ya que la plataforma se adapta a un nivel gratuito libre de uso para cualquier usuario a nivel de recursos y esto afecto directamente en el procedimiento.

De acuerdo al desarrollo que se realizó para esta actividad fue posible la ejecución de 4 contenedores Docker en paralelo que permitían la ejecución de pruebas en cada uno de ellos, este número se limita solamente a 4 debido a que la ejecución de las canalizaciones ya comenzaba a colocar problemas principalmente al iniciar el proceso de reconocimiento del Agente instalado en cada máquina y no era posible que las pruebas arrancaran.



De la misma manera el tiempo que tarda en encender el agente; inicialmente cuando se realizó la prueba con 2 contenedores aproximadamente duraban en encender los dos agentes completamente entre media hora a una hora, pero cuando se realizó el procedimiento de lanzar más de 2 contenedores el tiempo de encendido de los agente aumento, para que encendieran los 4 se tardó un tiempo aproximado de 7 a 8 horas esto debido a la alta carga de procesamiento que requieren arrancar los contenedores pero una vez los agentes encendieron no hubo ningún inconveniente para que las pruebas corrieran y a nivel de procesamiento era posible lanzar más contenedores.



```
diego@diego-Lenovo-Ideapad-320-15ABR: ~/Maquina01/lma...
diego@diego-Lenovo-Ideapad-320-15ABR:~/Maquina01/Ima...$ docker run -it --rm --p
rivilged -p 4444:4444 -p 5901:5901 windowstestiscontaineruno
Waiting xvfb...
--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 37
PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.
qemu-system-x86_64: warning: guest updated active QH
qemu-system-x86_64: warning: guest updated active QH
[]

diego@diego-Lenovo-Ideapad-320-15ABR: ~/Maquina02/lma...
diego@diego-Lenovo-Ideapad-320-15ABR:~/Maquina02/Ima...$ docker run -it --rm --p
rivilged -p 4445:4445 -p 5902:5902 windowstestiscontainersdos
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 40
PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.
qemu-system-x86_64: warning: guest updated active QH
qemu-system-x86_64: warning: guest updated active QH
[]

diego@diego-Lenovo-Ideapad-320-15ABR: ~/Maquina03/lma...
diego@diego-Lenovo-Ideapad-320-15ABR:~/Maquina03/Ima...$ docker run -it --rm --p
rivilged -p 4446:4446 -p 5903:5903 windowstestiscontaintres
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 47
PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.
[]

diego@diego-Lenovo-Ideapad-320-15ABR: ~/Maquina04/lma...
diego@diego-Lenovo-Ideapad-320-15ABR:~/Maquina04/Ima...$ docker run -it --rm --p
rivilged -p 4447:4447 -p 5904:5904 windowstestiscontaintres
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
Waiting xvfb...
--- x11vnc loop: 1 ---
--- x11vnc loop: waiting for: 72
PORT=5900
qemu-system-x86_64: warning: This family of AMD CPU doesn't support hyperthreadi
ng(2)
Please configure -smp options properly or try enabling topoext feature.
[]
```

Figura 79. Lanzamiento 4 contenedores al tiempo. [Elaboración Propia]

Se confirma que cada uno de los agentes instalados en las máquinas que se encuentran dentro de los contenedores haya encendido de manera correcta, estos agentes se



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



encuentran ubicados en los distintos grupos de agentes en Azure que se crearon para llevar una mejor organización en el proceso de lanzamiento de las pruebas y poder identificar cuales pruebas se lanzan en cada contenedor.

Name	Last run	Current status	Agent version	Enabled
Default				
DWAGENTESIS03	Now	Running build 20210613.3	2.187.2	On
GAgente01				
DWAGENTESIS02	Now	Running build 20210613.3	2.187.2	On
GAgente02				
DWAGENTESIS01	Now	Running build 20210613.2	2.187.2	On
GAgente03				
DWAGENTESIS04	Now	Running build 20210613.3	2.187.2	On

Figura 80. Agentes encendidos prueba contenedores. [Elaboración Propia]

Se realiza el proceso de ejecución de las diferentes canalizaciones creadas para el conjunto de pruebas, para este proceso y a modo de obtener una mejor organización y distribución de las pruebas automatizadas, a nivel de código se crearon 4 clases de tipo prueba unitaria una por cada contenedor, cada clase contiene un puerto diferente asociado a la conexión del WinAppDriver representando el puerto asociado en la creación de cada máquina. Actualmente dentro de Digital Ware se trabaja de la misma manera para el proceso de creación de módulos de automatización para los distintos maestros a probar.

Debido a un inconveniente presentado últimamente en la compañía Digital Ware no se pudo realizar la prueba de automatización sobre la página de consultoría de la misma, por tanto, se acude a seguir probando la automatización de la calculadora de Windows, esto a modo de ejemplo, pero nuevamente se indica que a nivel de funcionamiento y configuración es exactamente lo mismo.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



Jobs	Duration
Agent job 2	7m 5s
Initialize job	1s
Checkout TestWADTFSGitPublic...	59s
Use NuGet 4.4.1	24s
NuGet restore	1m 9s
Build solution "***.sln	1m 11s
VsTest - testAssemblies 2	2m 52s
Post-job: Checkout TestWADTFSG...	25s
Finalize Job	<1s
Report build status	<1s

Jobs	Duration
Agent job 1	5m 9s
Initialize job	<1s
Checkout TestWADTFSGitPublic...	20s
Use NuGet 4.4.1	1s
NuGet restore	17s
Build solution "***.sln	1m 13s
PowerShell Script	5s
VsTest - testAssemblies	2m 48s
Post-job: Checkout TestWADTFSG...	18s
Finalize Job	1s
Report build status	<1s

Jobs	Duration
Agent job 3	34m 24s
Initialize job	1s
Checkout TestWADTFSGitPu...	3m 40s
Use NuGet 4.4.1	2m 23s
NuGet restore	11m 20s
Build solution "***.sln	4m 48s
VsTest - testAssemblies	12m 5s
Post-job: Checkout TestWADTFSG...	1s
Finalize Job	1s
Report build status	<1s

Jobs	Duration
Agent job 4	5m 47s
Initialize job	1s
Checkout TestWADTFSGitPublic...	23s
Use NuGet 4.4.1	9s
NuGet restore	1m 4s
Build solution "***.sln	23s
VsTest - testAssemblies	3m 28s
Post-job: Checkout TestWADTFSG...	3s
Finalize Job	<1s
Report build status	<1s

Figura 81. Ejecución Pipeline en paralelo. [Elaboración Propia]

Finalmente se obtiene un resultado positivo al obtener de que las pruebas automatizadas en los distintos agentes se ejecutaron de forma correcta.

Pipeline	Last run
✓ Pipeline Agente Tesis 04	#20210615.8 • Cambio server Manually triggered for master
✓ Pipeline Agente Tesis 03	#20210615.2 • Cambio server Manually triggered for master
✓ Pipeline Agente Tesis 02	#20210615.3 • Cambio server Manually triggered for master
✓ Pipeline Agente Tesis 01	#20210615.3 • Cambio server Manually triggered for master

Figura 82. Resultado final Pipelines en ejecución. [Elaboración Propia]



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



A nivel de porcentaje de consumo de CPU y memoria se obtiene un resultado que posibilita el lanzamiento de otros contenedores más, solamente que no se pudo llevar a cabo por la problemática propuesta anteriormente y al mismo tiempo la mala conectividad de internet que afecta directamente sobre el proceso de ejecución.

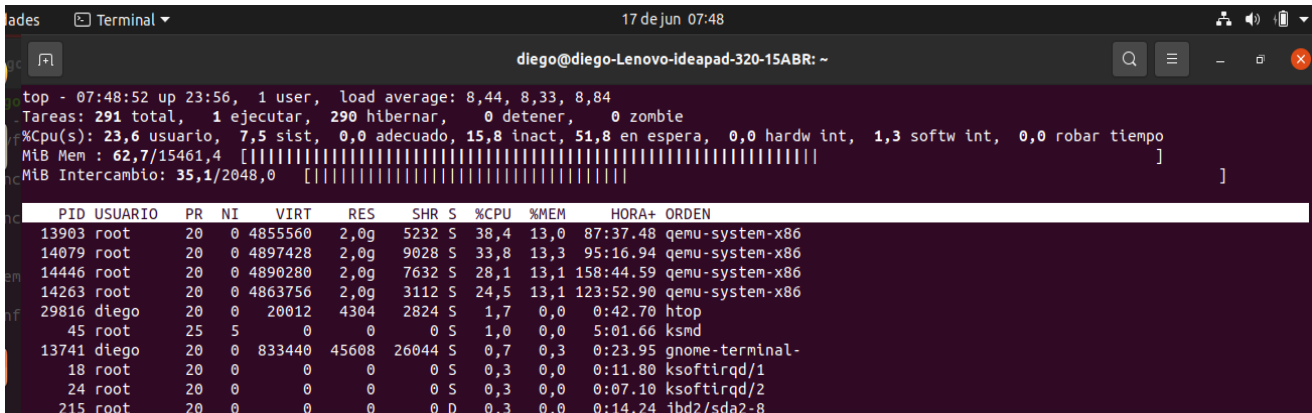


Figura 83. Porcentaje de consumo CPU. [Elaboración Propia]

Observando de una forma gráfica el consumo de CPU y de memoria se obtiene lo siguiente.

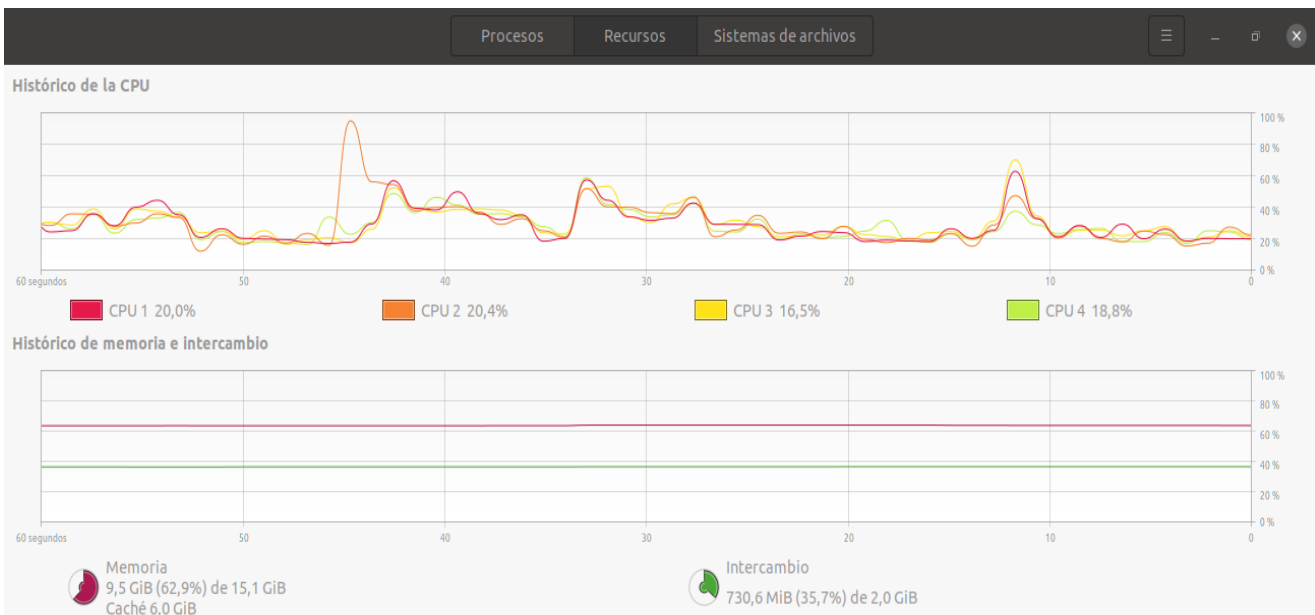


Figura 84. Consumo de CPU y memoria modo gráfico con contenedores en ejecución. [Elaboración Propia]



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



Es posible validar que el consumo de CPU varía entre un 50% a un 20% de acuerdo a la existencia de más procesos en paralelo, posiblemente pueda afectar un poco más a nivel de memoria ya que al ejecutar los contenedores de forma paralela tiene un porcentaje de ocupación de aproximadamente el 63% pero esto posibilita aun así la ejecución de aproximadamente 2 o 3 contenedores más en esta máquina, siendo un buen número de máquinas disponibles para la ejecución de pruebas.

Si se detienen los contenedores el porcentaje tanto de CPU como de memoria disminuyen directamente como se observa en la siguiente imagen.

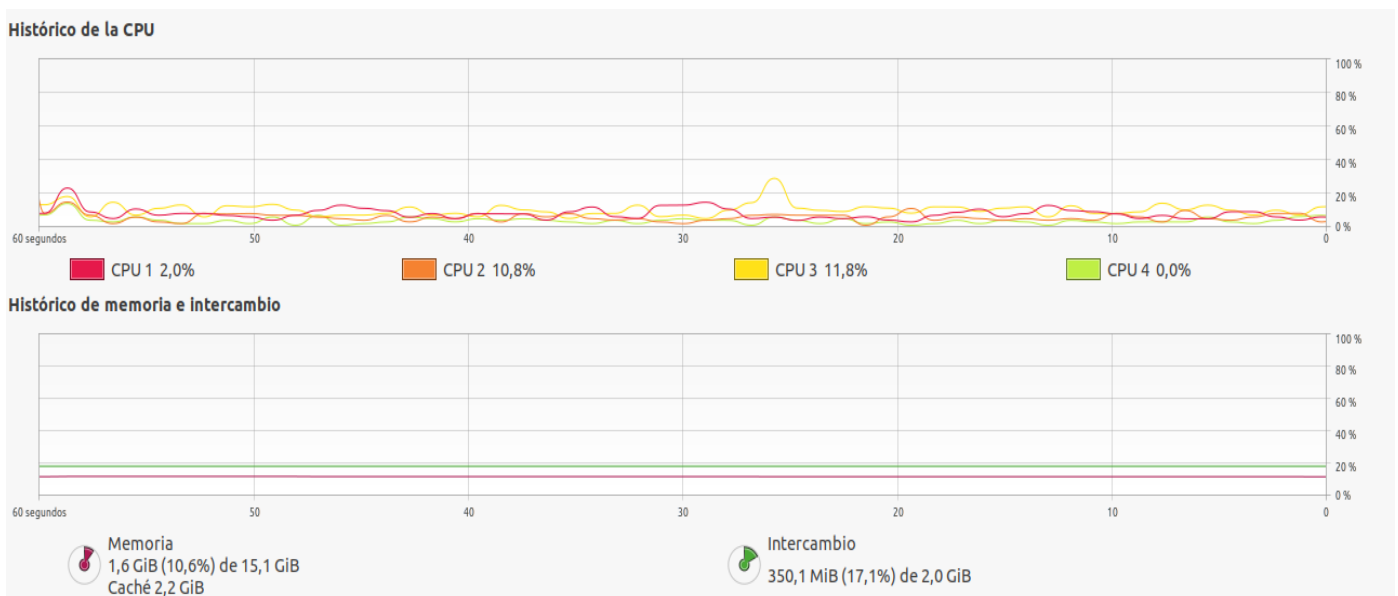


Figura 85. Consumo de CPU y memoria modo gráfico con contenedores detenidos. [Elaboración Propia]

Es posible visualizar una gran disminución del porcentaje a nivel de proceso, manteniendo una estabilidad en el porcentaje de uso de CPU menor al 20% y en memoria de aproximadamente solo el 11%.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



CAPÍTULO VII: CONCLUSIONES

Por medio del desarrollo de este trabajo usando herramientas de virtualización, contenedores basados en sistema Linux y la plataforma Microsoft Azure es posible realizar principalmente un proceso de integración de las pruebas de automatización, permitiendo la ejecución de pruebas desarrolladas dentro de contenedores Docker de una manera muy sencilla y rápida liberando las máquinas de desarrollo utilizadas en el área de automatización de la empresa Digital Ware.

Después de implementar los contenedores Docker internamente con el snapshot de la máquina virtual creada se consiguió que el agente autohospedado instalado en cada uno de ellos se ejecutara como servicio de Windows para su arranque automático y se realice un proceso de sincronización de fecha y hora con el programa NetTime para que sea posible la conexión entre el agente y el servidor Azure.

La aplicación de este método para la ejecución de las pruebas realizadas en el área de automatización de los programas Ophelia va a permitir reducir la velocidad de ejecución de las listas de test, esto debido a la ejecución de pruebas dentro de contenedores que se lanzan en un mismo equipo físico de forma paralela, en base a esto, se puede deducir que el costo por incluir capacidades de mejora con respecto a la velocidad y escalamiento de las pruebas se reduce a la inversión de Digital Ware en un solo Host con suficiente potencia de procesamiento.

Directamente no fue posible implementar el sistema de integración con los proyectos internos de la compañía esto debido a la no entrega del equipo que funcionara como servidor por parte de Digital Ware y al mismo tiempo el no acceso de una cuenta de Microsoft Azure, pero esto no fue impedimento para llevar a cabo el proyecto, de la manera en que se realizaron todas las configuraciones se debe hacer en el propio servidor solamente cambia el proyecto que será subido al repositorio.

En cuanto a la eficiencia del método de ejecución de pruebas implementado en este proyecto con respecto al modo en que se ejecutan en la empresa Digital Ware depende



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL

Avanzamos... ¡Es nuestro objetivo!



directamente de la cantidad de contenedores que ejecute en el servidor físico y esto se relaciona directamente con la capacidad de los contenedores de ejecutarse en paralelo, al mismo tiempo solventa muchos problemas que se presentan diariamente como el daño de alguna máquina física, en este caso si un contenedor ya no funciona por algún motivo simplemente es generar otra máquina virtual, introducirla dentro del contenedor, configurar y ejecutar nuevamente por tanto no perjudicaría directamente en el tiempo de entrega de las prueba automatizadas.

El número máximo de contenedores que se lograron ejecutar de forma paralela fueron 4, este número no se limitó directamente por las características de la máquina, sino que fueron factores externos como el no contar con una suscripción paga en la plataforma de Azure y la conexión a internet los que influyeron sobre la ejecución de estos. El tiempo de encendido de los agentes dentro de los contenedores aumento a media que se ejecutaban un mayor número de contenedores en paralelo, para este caso tomo aproximadamente entre 7 a 8 horas en encender la totalidad de estos pero una vez encendidos fue posible la ejecución de pruebas de automatización obteniendo un porcentaje de consumo de CPU del equipo servidor entre el 20% al 50% y una ocupación de memoria de aproximadamente el 63%, esto posibilita ejecutar aproximadamente entre 2 a 3 contenedores más logrando obtener un buen número de máquinas disponibles para el proceso de ejecución de las pruebas.

A nivel personal el desarrollo de esta pasantía genero un impacto gigantesco en mi vida, principalmente me enseñó el verdadero significado de la responsabilidad ya que es necesario el cumplimiento de actividades que son importantes para que un producto de la compañía salga a producción de manera correcta en un tiempo determinado, al mismo tiempo me enseñó a desarrollar habilidades comunicativas ya que constantemente se presentan reuniones y se hace necesario hacer partícipe de ellas, me enseñó la importancia del trabajo en equipo para poder solventar todas las actividades regidas por el plan de trabajo establecido, me permitió aprender a manejar personal a cargo, tomar decisiones en instantes de tiempo corto, motivar al equipo de trabajo y los más importante me enseñó a creer más en mí, en mis capacidades y darme cuenta que lo más importante sobre todas las cosas es siempre ser humilde en todas las situaciones.



SC-CER96940



"Formando líderes para la construcción de un nuevo país en paz"

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



REFERENCIAS

- [1] Digital Ware, «Portafolio de productos,» [En línea]. Available: <https://www.digitalware.com.co/>. [Último acceso: 18 05 2021].
- [2] Digital Ware, «ACERCA DE DIGITAL WARE,» [En línea]. Available: <http://www.kactus.com.co/index.php/nosotros-2>. [Último acceso: 18 05 2021].
- [3] Digital Ware, «SOBRE NOSOTROS,» [En línea]. Available: <https://www.digitalware.com.co/sobre-nosotros/>. [Último acceso: 18 05 2021].
- [4] J. L. Garcia Hoyos, *Migración del software KACTUS-HCM de una arquitectura cliente-servidor a una arquitectura cliente-contenedor*, Pamplona, 2019.
- [5] J. F. Gomez Castillo, *Desarrollo de demo portable y automatizado para las aplicaciones KACTUS-HCM fundamentado en contenedores sobre Docker para la empresa Digital Ware*, Pamplona, 2020.
- [6] W. D. Bustos Morales, *Implementación de servidor de pruebas a partir de contenedores Docker y virtualización anidada en la empresa Digital Ware*, Pamplona, 2020.
- [7] S. Gonzales Casco, *Pruebas automáticas con Selenium sobre Docker Windows Containers*, México, 2018.
- [8] G. J. Myers, T. M. Thomas y J. Wiley, *The Art of Software Testing*, Second Edition, 2004.
- [9] J. M. Sanchez, *Pruebas de Software. Fundamentos y Técnicas*, Madrid, 2015.
- [10] R. CUBOS MONTENEGRO, *Testing y calidad de software. automatización de pruebas con selenium webdriver*, 2017.
- [11] J. C. Franco Ochoa, *Metodología para testing de software basado en componentes*, Medellin, 2010.



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



- [12] I. C. Society, Swebok - Guía al cuerpo de conocimiento de la Ingeniería de Software, California, 2004.
- [13] J. Vela Garcia, *DISEÑO Y DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN AUTOMÁTICA DE PRÁCTICAS*, Madrid, 2009.
- [14] A. Fajardo Moya, *QEMU, una alternativa libre para la emulación de arquitecturas de hardware*, Cuba, 2016.
- [15] N. S. S. Singh, «Containers & Docker: Emerging roles & future of Cloud technology,» *Computer Science*, pp. 804-807, 2016.
- [16] O. S. e. l. nube, «Máquinas virtuales ¿Cuál es su importancia en el sector empresarial 2019?,» 05 05 2019. [En línea]. Available: <https://on-cloud.blog/2019/03/05/maquinas-virtuales-cual-es-su-importancia-en-el-sector-empresarial-2019/>. [Último acceso: 18 05 2021].
- [17] J. M. Ortega, *DOCKER. Seguridad y monitorización en contenedores e imágenes*, 2019.
- [18] sitiobigdata.com, «¿Qué es Docker? Contenedores Docker explicados,» 10 09 2018. [En línea]. Available: <https://sitiobigdata.com/2018/09/10/que-es-docker-contenedores-explicados/#>. [Último acceso: 18 05 2021].
- [19] J. M. Alarcon, «¿Qué diferencia hay entre Docker (Contenedores) y Máquinas virtuales (VMWare, VirtualBox...)?,» 14 06 2018. [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-diferencia-hay-entre-docker-contenedores-y-maquinas-virtuales.aspx>. [Último acceso: 18 05 2021].
- [20] D. Maggi, *DISEÑO DE UNA ARQUITECTURA BASADA EN CONTENEDORES PARA LA INTEGRACIÓN Y EL DESPLIEGUE CONTINUO (CI/CD)*, Málaga , 2020.
- [21] V. Zuñiga, *INTRODUCCIÓN A LA PROGRAMACIÓN DE SCRIPT EN BASH*, 2006.
- [22] J. Maret, *Getting Started with WinAppDriver*, 2018.
- [23] J. A. Carvalho, L. Moreira, L. Rosso y C. Ibirí, *Introducción a DevOps y DevSecOps*, Argentina.



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750



ACREDITACIÓN INSTITUCIONAL
Avanzamos... ¡Es nuestro objetivo!



- [24] E. Leal, C. Sosa y D. Leal, «REVISIÓN DE LOS SISTEMAS DE CONTROL DE VERSIONES UTILIZADOS EN EL DESARROLLO DE SOFTWARE,» *Ingenierías USBMed*, vol. 3, nº 1, pp. 74-81, 2012.
- [25] R. J. García Pinel, *Análisis de mecanismos de integración de herramientas para suministro continuo (Continuous Delivery)*, Madrid, 2015.
- [26] C. Gómez Sanchez, *Estudio de herramientas de despliegue continuo de aplicaciones, y sus ventajas competitivas en un mundo marcado por la agilidad*, Madrid, 2019.
- [27] Microsoft, «Azure DevOps,» [En línea]. Available: <https://azure.microsoft.com/es-es/services/devops/>. [Último acceso: 18 Mayo 2021].
- [28] Microsoft, «What is Azure Pipelines?,» [En línea]. Available: <https://docs.microsoft.com/en-us/azure/devops/pipelines/get-started/what-is-azure-pipelines?view=azure-devops>. [Último acceso: 18 Mayo 2021].
- [29] Microsoft, «Azure Pipelines agents,» [En línea]. Available: <https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=azure-devops&tabs=browser>. [Último acceso: 18 Mayo 2021].
- [30] C. Collado Fernández, P. Lucio Baptista y S. H. Rober, *Metodología de investigación*, McGraw Hill Education, 2017.
- [31] M. Parada, «Qué distribuciones Linux usar en 2021,» 12 Octubre 2020. [En línea]. Available: <https://openwebinars.net/blog/que-distribuciones-linux-usar-en-2021/>. [Último acceso: 19 Mayo 2021].
- [32] D. Guide, «Virtualización con contenedores Docker: alternativas,» 09 Julio 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/alternativas-a-los-contenedores-en-docker/#:~:text=Docker%20utiliza%20caracter%C3%ADsticas%20nativas%20del,virtualizaci%C3%B3n%20total%20con%20m%C3%A1quinas%20virtuales.&text=%E2%9C%94%20Docker%20sopor>. [Último acceso: 19 Mayo 2021].



SC-CER96940



“Formando líderes para la construcción de un nuevo país en paz”

Universidad de Pamplona
Pamplona - Norte de Santander - Colombia
Tels: (7) 5685303 - 5685304 - 5685305 - Fax: 5682750