

AUTOMATIZACIÓN DEL PROCESO PARA LA GESTIÓN DE LA INFORMACIÓN TÉCNICA DE LOS PROYECTOS EN PWO DE MÉXICO, MEDIANTE EL USO DE POWERSHELL DE TECNOLOGÍA .NET

AUTOMATION OF THE PROCESS FOR THE MANAGEMENT OF THE TECHNICAL INFORMATION OF THE PROJECTS IN PWO OF MEXICO, THROUGH THE USE OF POWERSHELL OF .NET TECHNOLOGY

¹M.C. Violeta Martínez Ramírez, ²M.I. Alejandro Gil Vázquez,
³M.C. Juan Gonzalo Alarcón Xicotencatl, ⁴Ing. Ángel de Jesús Servin Chávez

^{1,4}Instituto Tecnológico de Puebla
Tecnológico Nacional de México
Departamento de Sistemas y Computación
Av. Instituto Tecnológico #420, colonia Maravillas. Puebla, Puebla
¹violeta.martinez@itpuebla.edu.mx, ⁴i15221977.19@puebla.tecnm.mx

²Instituto Tecnológico de Tláhuac
Tecnológico Nacional de México
Departamento de Sistemas y Computación
Av. Estanislao Ramírez #301 Col Selena, Alcaldía Tláhuac, Ciudad de México
²ing.gil@ittlahuac.edu.mx

³Instituto Tecnológico de Puebla
Tecnológico Nacional de México
Departamento de Eléctrica y Electrónica
Av. Instituto Tecnológico #420, colonia Maravillas. Puebla, Puebla
³gonzalo.alarcon.x@gmail.com

Resumen – El crecimiento constante de las empresas, así como el aumento de la demanda en los productos que estas mismas ofrecen, las ha orillado a incrementar sus procesos y de esta forma necesitar un control más exacto de los mismos, con lo cual han requerido automatizar ciertos procesos que no necesitan ser controlados por humanos, ya que estos pueden ahorrar tiempos y disminuir la carga de trabajo de algunos empleados.

Mediante el uso de la metodología ágil Scrum, se llevó a cabo el desarrollo de la automatización a un proceso productivo de la empresa PWO de México S.A. de C.V. El cual se realizó mediante iteraciones (Sprints) en tiempos cortos (de 1 a 4 semanas) durante los cuales se realizaron entregas a los clientes, de esta manera se aseguró un desarrollo confiable y acorde a los gustos de estos. Los resultados obtenidos con este proyecto se obtuvo un ahorro de casi 98% en el tiempo que un empleado invierte en la realización de este proceso.

Palabras Clave: Automatización, proceso, powershell, directorios, permisos de acceso.

Abstract -- The constant growth of companies, as well as the increase in demand for the products they offer, has led them to increase their processes and thus need a more exact control of them, with which

they have required to automate certain processes. They do not need to be controlled by humans, as these can save time and reduce the workload of some employees.

Using the agile Scrum methodology, the development of automation was carried out in a production process of the company PWO de México S.A. de C.V. Which was carried out through iterations (Sprints) in short times (from 1 to 4 weeks) during which deliveries were made to customers, in this way a reliable development was ensured according to their tastes. The results obtained with this project were obtained a saving of almost 98% in the time that an employee invests in carrying out this process.

Key words – Automation, process, powershell, directories, access permissions.

INTRODUCCIÓN

Debido a la creciente demanda de trabajo, la optimización de los procesos productivos en las empresas busca una optimización de éstos, en ocasiones, la automatización de es la respuesta a sus necesidades, debido a considerables ahorros de tiempo en las tareas y minimización de la interacción del personal en ellos, evitando la

generación de errores por captura de datos erróneos.

El presente trabajo de investigación da a conocer de manera detallada el proceso de desarrollo llevado a cabo en la empresa PWO de México S.A. de C.V. para automatizar la generación de la estructura de directorios de números de parte fabricados por la planta de México, así como, la asignación de permisos de acceso a diferentes grupos de trabajo en los directorios obtenidos.

PLANTEAMIENTO DEL PROBLEMA

La creación y control de la estructura de almacenamiento, así como, la asignación de los permisos de acceso en los directorios de nuevos números de parte creados en PWO de México S.A. de C.V., son actividades que se han realizado de manera manual, si bien, existen scripts que crean directorios y asignan permisos de acceso a los grupos ya establecidos por la empresa, estas actividades deben repetirse cada que un nuevo número de parte es establecido, que resulta como una tarea tediosa y repetitiva para la o las personas encargadas.

Otro parámetro a considerar es el tiempo de uso. Al ser un proceso reactivo, el tiempo de las actividades será directamente proporcional al tiempo requerido para obtener actualizada la estructura de los números de parte, esta tarea no se realiza de manera secuencial, sino cada intervalo transcurrido. atrasando los tiempos de visualización y, por consecuencia, el proceso de producción.

Toda la problemática anteriormente expuesta, trae como resultado que la creación de directorios y el control de la estructura de almacenamiento de datos sea lenta y, por ende, que la estructura no pueda ser visualizada de manera rápida afectando sustancialmente a las áreas involucradas en el proceso de producción quienes hacen uso de éstos

al no contar con una ubicación estándar para el almacenamiento de datos requeridos para el trabajo de diseño y/o medición de piezas.

OBJETIVO GENERAL

Implementar un sistema de información que controle la estructura de almacenamiento digital de proyectos generados en el proceso productivo de la

empresa PWO de México S.A. de C.V., mediante el uso de la interfaz de línea de comandos PowerShell de Tecnología .NET.

OBJETIVOS ESPECÍFICOS

- Analizar la información y los requerimientos.
- Crear una estructura de carpetas recursiva basada en el formato de almacenamiento preestablecido.
- Establecer permisos mediante Microsoft's Access Control Lists (ACL) en carpetas y archivos.
- Descargar archivos de una unidad de red específica que alberga datos requeridos por la empresa ubicada en la central en Alemania (unidad N:) a través de FTP y ajustar los permisos de lectura.
- Prueba de software en periodos programados.

JUSTIFICACIÓN

Debido a una demanda alta en la creación de nuevos números de parte producidos por diversas áreas involucradas dentro del proceso productivo de la empresa, se debe contar con una estructura de almacenamiento para datos CAD y programación de medición de piezas. Esta estructura debe estar actualizada, simulando una sincronización entre los datos existentes en el ERP y los datos almacenados para fines de diseño o medición. Al tratarse de datos no directamente relacionados a un sistema de ERP, la empresa ha creado una estructura específica que garantice la seguridad de los datos mediante el uso de tecnología de asignación de permisos basado en Access Control List (ACL) y Active Directory (AD).

El excesivo tiempo invertido en la creación de directorios, control de la estructura de almacenamiento de datos y la asignación de controles de acceso en la creación de nuevos

números de parte en la empresa, exige la automatización de estas actividades con la finalidad de optimizar la visualización de documentos por los departamentos involucrados en el proceso de producción, Asimismo, favorecer en las personas involucradas enfocarse en nuevos proyectos y/o tareas. En este sentido, el presente trabajo, muestra la implementación de un sistema para el control de la estructura de almacenamiento

digital mediante el uso de la tecnología .NET, haciendo uso de la interfaz de consola PowerShell. La aportación de este trabajo se verá manifestada en los beneficios que resultan de las acciones de mejora: la optimización del proceso de creación de directorios y las actividades que esto conlleva. Asimismo, se reducirán las demoras que impactan directamente en los tiempos de consulta de los documentos, las actividades se llevarán a cabo de manera uniforme, en menor tiempo y de forma más práctica.

MARCO TEÓRICO

PWO (Progress-Werk Oberkirch AG) es proveedor en la industria automotriz mundial para el desarrollo y la producción de componentes y subsistemas metálicos avanzados que utilizan una construcción ligera. El Grupo ha desarrollado un conocimiento único en la formación y unión de metales en el transcurso de sus 100 años de historia desde su fundación en 1919. Con su experiencia en construcción ligera y rentable, contribuye a una conducción respetuosa con el medio ambiente y a mayores distancias [1].

Windows PowerShell es una interfaz de consola (CLI) con posibilidad de escritura y conjunción de comandos por medio de guiones (scripts en inglés). Es mucho más rica e interactiva que sus predecesores, desde DOS hasta Windows 7. Esta interfaz de consola está diseñada para su uso por parte de administradores de sistemas, con el propósito de automatizar tareas o realizarlas de forma más controlada. Originalmente denominada como MONAD en 2003, su nombre oficial cambió al actual cuando fue lanzada al público el 25 de abril del 2006.

El estándar utilizado por PowerShell son los cmdlets de Microsoft Office 365 para Windows PowerShell para realizar diversas tareas administrativas desde la línea de comandos. Se enumeran y describen todos los cmdlets de Office 365 por tarea administrativa común, como el

control de usuarios y la administración de dominios. [2].

Windows PowerShell presenta el concepto de cmdlet ("command-let"). Un cmdlet es una herramienta simple de línea de comandos de una sola función incluida en el shell, cuya finalidad es manipular objetos. Puede reconocer los cmdlets por el formato de su nombre: un verbo y un nombre

separados mediante un guion (-), por ejemplo, Get-Help y Get-State. Los verbos expresan acciones específicas de Windows PowerShell, mientras que los nombres describen tipos específicos de objetos.

Es posible que los cmdlets de PowerShell tengan parámetros obligatorios y opcionales. Si falta un parámetro obligatorio, el sistema le solicitará que lo introduzca. Si falta un parámetro opcional, PowerShell utilizará el valor predeterminado [3].

La plataforma .NET es un ambiente utilizado para desarrollar aplicaciones y ejecutarlas en sistemas operativos WIN32. El desarrollo de aplicaciones en esta plataforma no depende de ningún lenguaje, es decir, se puede desarrollar de forma multilinguaje.

El framework .NET es la parte medular de la plataforma .NET, es una infraestructura que se utiliza para integrar y ejecutar aplicaciones basadas en Microsoft Windows. Se compone principalmente de un conjunto de compiladores de diversos lenguajes, una biblioteca de datos y una máquina virtual [4].

Las listas de control de acceso (ACL) realizan el filtrado de paquetes para controlar el movimiento de paquetes a través de una red. El filtrado de paquetes proporciona seguridad al limitar el acceso del tráfico a una red, restringir el acceso de usuarios y dispositivos a una red y evitar que el tráfico salga de la red. Las listas de acceso IP reducen la posibilidad de falsificación y ataques de denegación de servicio, y permiten el acceso de usuario dinámico y temporal a través de un firewall [5].

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Se podría decir que SCRUM se basa en cierto caos controlado, pero establece ciertos mecanismos para controlar esta indeterminación, manipular lo impredecible y controlar la flexibilidad.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en

entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales [6].

ESTADO DEL ARTE

El trabajo de Olmeda de la Casa implementa un sistema de clonación de imágenes base con el sistema operativo (Windows 10 Enterprise de 64 bits) junto con el software y utilidades aplicadas en las aulas informáticas de un departamento universitario para la realización de las prácticas docentes. De esta manera, se dispone de una configuración idéntica en las aulas, sin menoscabo de equipos informáticos con diferente hardware facilitando la impartición de prácticas independientes del aula asignada. Mediante la solución propuesta, es posible realizar mantenimiento eficiente y sencillo de los equipos informáticos frente a futuras actualizaciones del software, probables ampliaciones en el número de equipos que conforman las aulas o fallos de hardware o software.

Se ha desarrollado una serie de scripts empleando la herramienta de Powershell para la automatización de tareas rutinarias periódicamente ejecutadas en la administración de equipos que conforman las aulas informáticas. Esto, facilita la gestión de los sistemas operativos de forma sencilla y segura, minimizando los posibles errores frente a la tediosa repetición de tareas de elegir el mantenimiento equipo por equipo. Además, se aprovecha que está en equipos con directorio activo -AD- configurados con parámetros de grupo en directivas (GPO) para complementar y facilitar la disposición del sistema de los equipos informáticos [7].

Huera Paltán presenta una investigación sobre scripts generados con Microsoft PowerShell y la gestión gráfica para administrar Windows Server

2008, permite seleccionar la gestión óptima en la academia Microsoft de la Escuela Superior Politécnica de Chimborazo. El análisis comparativo se sustenta en método Científico, con técnicas de observación directa, y experimentación, mediante la implementación de herramientas Software: Powershell 2.0, PowerGUI y monitor del Sistema.

La metodología de administración motivo de investigación, fueron integrados en Powershell y modo gráfico, para automatizar sistemas servidores, permitiendo determinar cuál posee mejor rendimiento, usabilidad y portabilidad, dando como resultado cuantitativo una diferencia de indicadores un 59% para la administración gráfica vs un 66% para la administración con PowerShell [8].

DESARROLLO DE LA INVESTIGACIÓN

El desarrollo del proyecto se llevó a cabo mediante el uso de la metodología ágil Scrum, por lo que se realizaron distintas iteraciones ejecutadas en bloques temporales cortos (de 1 a 4 semanas) y al término de cada bloque era entregado un avance del proyecto al cliente quien proponía añadir, eliminar o modificar ciertas funcionalidades al proyecto.

En la figura 1 se muestra un esquema que representa la metodología Scrum aplicada al proyecto, sin embargo, en párrafos posteriores se describe cada etapa desarrollada.

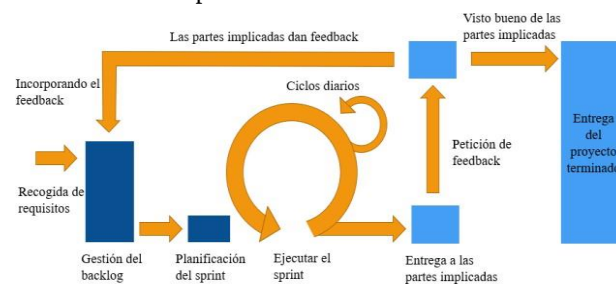


Figura 1. Diagrama de Metodología ágil Scrum
Fuente: Elaboración propia.

Recopilación de información y requisitos (Reunión con el equipo de trabajo)

El cliente presenta a los miembros del equipo de trabajo del proyecto, registrándose los requerimientos priorizados. Se identifica el objetivo a seguir para el desarrollo sistema de información que automatice la creación y control

de directorios de números de parte de cada componente dentro de los departamentos involucrados en el proceso de producción de la empresa, de igual manera se establecieron los productos entregables al término del primer Sprint.

Ejecución de las iteraciones (Sprints).

Considerando la metodología scrum, el diagrama de la figura 2, permite conocer los objetivos logrados por cada sprint ejecutado en el desarrollo del proyecto.



Figura 2. Desarrollo de la metodología ágil Scrum con sus Sprints y resultados
Fuente: Elaboración propia.

Reestructuración de archivos y extracción de contenido

Para el primer sprint del desarrollo del proyecto el objetivo alcanzado fue la reestructuración de archivos complementarios en el proceso, los cuales son esenciales para ejecutar funcionalidades en el proyecto, un ejemplo de estos es un archivo con nombre "Acl_config.ini" (figura 3), el cual será leído por el software para obtener datos precisos sobre grupos y permisos de acceso.

```

Acl_config.ini
1 \Qualitaet;mxpwo\dm_MX_Qualitaet_R:(OI)(CI)(RX)
2 \Qualitaet;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(RX)
3 \Qualitaet\Messdaten\Bilder;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(M)
4 \Qualitaet\Messdaten\Doku;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(M)
5 \Qualitaet\Messdaten\Faehigkeit;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(M)
6 \Qualitaet\Messdaten\HyperReport;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(M)
7 \Qualitaet\Messdaten\Programme;mxpwo\dm_MX_Qualitaet_RWD:(OI)(CI)(M)
8 \Stadienplan;mxpwo\dm_MX_Stadienplan_R:(OI)(CI)(RX)
9 \Stadienplan;mxpwo\dm_MX_Stadienplan_RWD:(OI)(CI)(M)
10 \Werkzeuge;mxpwo\dm_MX_Werkzeuge_R:(OI)(CI)(RX)
11 \Werkzeuge;mxpwo\dm_MX_Werkzeuge_RWD:(OI)(CI)(M)
    
```

Figura 3. Estructura del documento "Acl_config.ini"
Fuente: Elaboración propia.

También se trabajó con la extracción de información de los documentos sobre los cuales se realiza la ejecución del software, estos documentos contienen información importante, como el número de parte del componente a crear, la planta sobre la cual se crea e información adicional. Un ejemplo de este documento es el que se muestra en la figura 4.

1	PUE01358	P04141	000521	5000PUE15016	PUE08014	1019	210	E11449	101	I-B
2	PUE01358	P04141		5000PUE15016	PUE08014	1019	110	E11449	101	I-B

Figura 4. Estructura de archivo .erl
Fuente: Documentación PWO.

Creación de directorios y asignación de permisos

En el segundo Sprint, se realizaron las correcciones y nuevos requerimientos proporcionados en la revisión anterior, a su vez se desarrolló el proceso de creación de directorios, estructura interna y asignación de permisos de acceso a los diferentes grupos que existen en el proceso de producción de la empresa. En la tabla 1, se muestra la estructura interna que cada número de parte debe contener al ser creado.

Raíz	Nivel 1	Nivel 2	Nivel 3
P04141	Qualitaet	Messdaten	Bilder
			Doku
			Faehigkeit
			HyperReport
	Stadienplan		Programme
	Werkzeuge		

Tabla 1. Estructura interna de un directorio de número de parte

Fuente: Elaboración propia.

Una vez creada la estructura de los directorios, se procedió a asignar los permisos de acceso a los diferentes grupos involucrados en el proceso de producción. La figura 5, presenta un fragmento de código escrito para la asignación de permisos ACL; en ella, se observan cuatro sentencias.

La primera, se encarga de obtener la información de ACL que incluye por defecto, el directorio.

En la segunda sentencia, se crea un nuevo objeto de ACL donde se especifica el grupo de usuarios que tendrán acceso, el tipo de acceso que posee y el comando "Allow" para aprobar la asignación. En la tercera sentencia, se agrega a la información ACL obtenida de la primera sentencia del permiso creado anteriormente.

Finalmente, en la última línea se guarda información generada de las sentencias anteriores, especificando la ruta del directorio sobre el cual se aplicarán los permisos y el objeto que incluye la información.

```
#Getting default Acl from directory
$Acl = Get-Acl "$routepn\partnumber"
#Set default group in directory and access type
$Ar = New-Object System.Security.AccessControl.FileSystemAccessRule($groupAD, "ReadAndExecute", "Inheritance", "Allow")
$Acl.SetAccessRule($Ar)
#Applying access to directory
Set-Acl -Path "$routepn\partnumber" -AclObject $Acl
```

Figura 5. Fragmento de código para asignación de permisos ACL.

Fuente: Elaboración propia.

Ejecución de validaciones

En la tercera iteración del desarrollo del proyecto, se dirigió a la creación de las validaciones, quienes son parte fundamental del óptimo funcionamiento del software por su reducción de riesgos al momento en el que los datos ingresan al procesamiento de la información por medio del código y asegurarán la precisión y confiabilidad del dato que ingresa. Algunas de las validaciones de mayor consideración son:

- Verificar que el número de parte sea de tipo 'P'
- Verificar que la planta sobre la cual fue creado el número de parte pertenezca a Puebla (planta 5000)
- Verificar longitud de los números de parte.

La nomenclatura interna de los números de parte indica que, si tienen una longitud mayor a 6 dígitos, significa que en el documento del que son extraídos contienen un espacio, siendo reemplazado por un guion bajo '_' al momento de crear el directorio.

Optimización de código y validaciones

El cuarto Sprint en el desarrollo del proyecto, está dirigido a la optimización del código, identificando todos los casos posibles en los que surgieran errores y/o excepciones, logrando así controlarlos y evitar que el proceso pare. Es de suma importancia identificar estos casos a tiempo,

manejarlos y evitar la creación de archivos "basura" que no cumplen con las normas de calidad especificadas inicialmente.

En la tabla 2, se exhiben dos ejemplos de manejo de excepciones en dos etapas importantes del proceso: la creación de directorios y la asignación de permisos ACL a los mismos.

Proceso	Excepción	Mensaje
---------	-----------	---------

Creación de directorio	System.IO.IOException	"Director y already exists"
	Others	"an error occurred"
Asignación de permisos	System.Security	"you may not have enough permissions"
	Others	"an error occurred"

Tabla 2. Estructura interna de un directorio de número de parte

Fuente: Elaboración propia.

Proyecto terminado

El desarrollo de la quinta iteración está orientado en la realización de pruebas localmente que darán paso a las pruebas en servidores del cliente y corrección de errores en tiempo de ejecución.

Inician las pruebas de software de forma local, creando grupos ficticios en el equipo de prueba, modificando algunos de los archivos esenciales y ejecutando el software para probar su efectividad.

Uno de los archivos editados para llevar a cabo la fase de pruebas de manera local, fue el documento "Acl_config.ini" (figura 6), en él se sustituyen los grupos pertenecientes a la empresa, por los grupos ficticios creados en el equipo.

```
Acl_config.ini
1 \Qualitaet;DESKTOP-G94N9I1\grupo_prueba_rx:(OI)(CI)(RX)
2 \Qualitaet;DESKTOP-G94N9I1\grupo_prueba_rx:(OI)(CI)(RX)
3 \Qualitaet\Messdaten\Bilder;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
4 \Qualitaet\Messdaten\Doku;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
5 \Qualitaet\Messdaten\Faehigkeit;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
6 \Qualitaet\Messdaten\HyperReport;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
7 \Qualitaet\Messdaten\Programme;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
8 \Stadienplan;DESKTOP-G94N9I1\grupo_prueba_rx:(OI)(CI)(RX)
9 \Stadienplan;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
10 \Werkzeuge;DESKTOP-G94N9I1\grupo_prueba_rx:(OI)(CI)(RX)
11 \Werkzeuge;DESKTOP-G94N9I1\grupo_prueba_m:(OI)(CI)(M)
```

Figura 6. Estructura del archivo "Acl_config.ini" para pruebas

Fuente: Elaboración propia.

Una vez que los documentos fueron editados para las pruebas correspondientes, los resultados satisfactorios derivados de las simulaciones locales fueron aprobados para continuar con la fase de pruebas sobre servidores del cliente.

La figura 7, presenta dos ejemplos de números de parte generados por el software, cada uno con formato diferente de los dos posibles para números de parte generados.

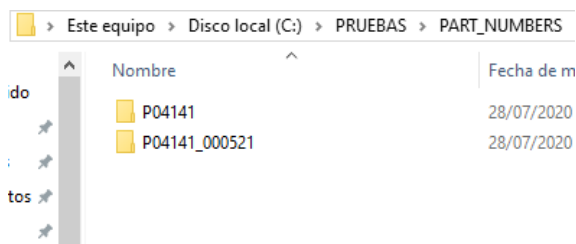


Figura 7. Números de parte creados en fase de pruebas
Fuente: Elaboración propia.

En la figura 8 se aprecian los permisos de acceso al directorio principal del número de parte que está seleccionado del grupo “prueba_grupo_1” asignado por software.

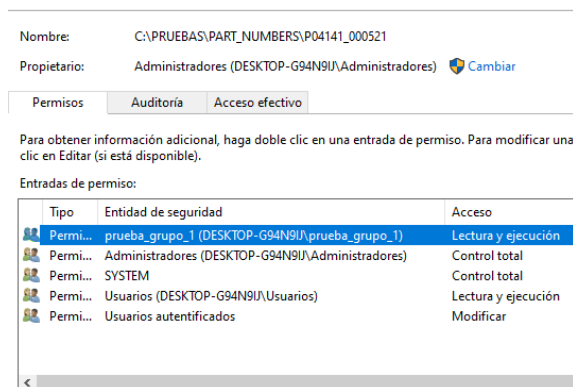


Figura 8. Grupo ficticio asignado en la carpeta raíz de un número de parte
Fuente: Elaboración propia.

En la figura 9, se observa el directorio “Qualitaet”, establecido como interno dentro de la estructura del número de parte, el grupo “grupo_prueba_rx” está asignado a él con sus permisos de acceso.

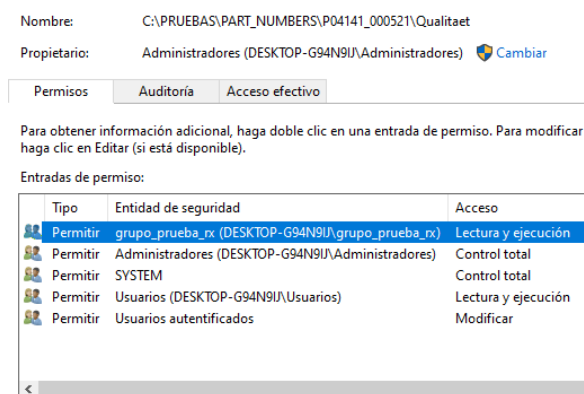
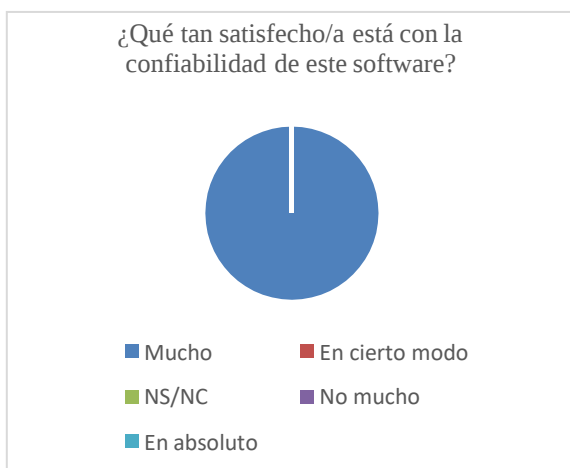


Figura 9. Grupo ficticio asignado en la carpeta Qualitaet de un número de parte
Fuente: Elaboración propia.

DISCUSIÓN Y ANÁLISIS DE RESULTADOS

Una vez ejecutado el software en los servidores de la empresa, se alcanzó los objetivos inicialmente esperados y se ejecutó en tiempo real en producción, cumpliendo con todos y cada uno de los requisitos solicitados por la empresa. Finalmente, se aplicó una encuesta de satisfacción al equipo del departamento de Organización y tecnologías de la Información (OI) para evaluar la funcionalidad del software. Cada ítem sigue el modelo de escala Likert, que contempla 5 posibles respuestas que representa el grado de satisfacción en cada indicador (estas respuestas están representadas por colores en las gráficas que se muestran en las gráficas 1, 2, 3, 4, 5, 6 y 7), en el departamento laboran 3 personas, conformando así el 100% de la recolección de datos en cada pregunta, cada integrante representa el 33% de cada respuesta en el instrumento de investigación. Análisis de los resultados:

Pregunta 1: Esta pregunta está enfocada a la dimensión de Confiabilidad que se tiene con respecto al funcionamiento del software, por ejemplo, la corrección de errores que surgieron durante el desarrollo y ahora están resueltos.

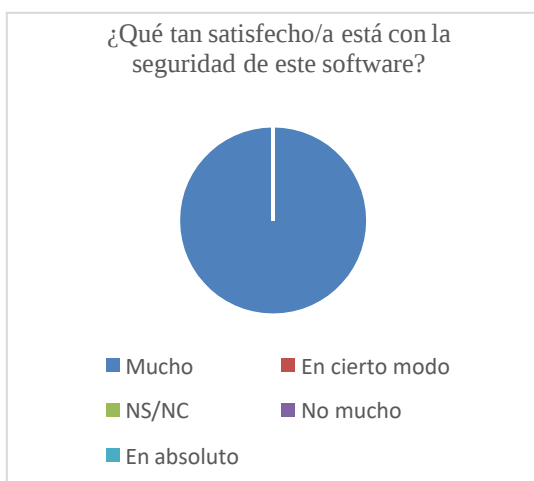


Gráfica 1. Resultado de la pregunta 1, con respecto a la confiabilidad del software.

Fuente: Elaboración propia.

El 100% de los encuestados está satisfecho con la confiabilidad del software, debido a las pruebas realizadas con anterioridad, y a las correcciones realizadas durante estas mismas.

Pregunta 2: La pregunta número 2 de la encuesta, dimensión Seguridad del cliente con el manejo de excepciones que pueden surgir durante la ejecución del software en producción.



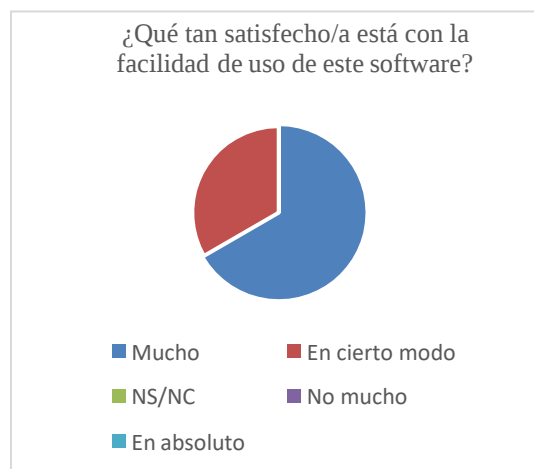
Gráfica 2. Resultado de la pregunta 2, referente a la seguridad del software

Fuente: Elaboración propia.

La seguridad del software está respaldada por la programación de éste, incluye todos los casos en los que pudiera ser inseguro, de esta forma se evitan problemas que pudiera presentar durante su

funcionamiento. Así es como todos los encuestados concuerdan en que el software es seguro.

Pregunta 3: El uso del software es automatizado, a través de un script que se ejecuta cada cierto tiempo, sin embargo, también se puede ejecutar manualmente, para esto requiere un previo conocimiento del lenguaje de programación utilizado.

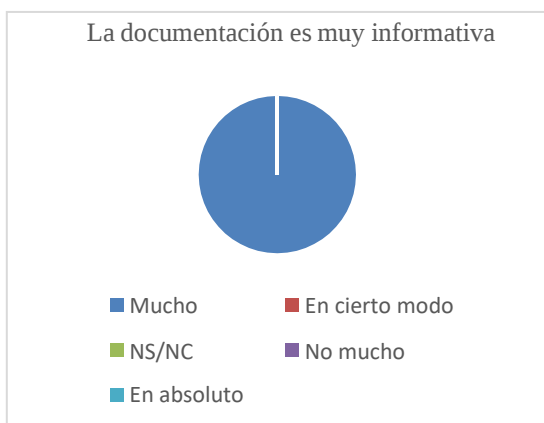


Gráfica 3. Resultado de la pregunta 3, satisfacción de los encuestados con respecto a la facilidad de uso del software

Fuente: Elaboración propia.

En este ítem, el 67% de los encuestados coincide en una interacción con el software sencilla. Dado a que es una automatización, involucrarse con el código es innecesario. La razón por la que esté satisfecho el 33% es porque no es necesario tener conocimientos del lenguaje, al menos, que se requiera cambios o efectuar mantenimiento profundo será necesario un conocimiento previo en el lenguaje de programación.

Pregunta 4: La documentación es tan importante como el software mismo, de ésta dependen las correcciones a errores y mantenimiento a futuro, entregado como manual de usuario.



Gráfica 4. Resultado de la pregunta 4, sobre la documentación entregada a la empresa
Fuente: Elaboración propia.

Debido a la amplia documentación entregada al equipo de OI en el manual de usuario, sobre el funcionamiento del software y el uso de este, la encuesta arrojó un resultado de 100% de satisfacción por parte de los encuestados.

Pregunta 5: Las excepciones son errores en tiempo de ejecución que se presentan cuando un proceso no se realiza correctamente, lo cual puede frenar la ejecución del software, cuando estos aparecen, se deben manejar y notificar al usuario que esto ha ocurrido.

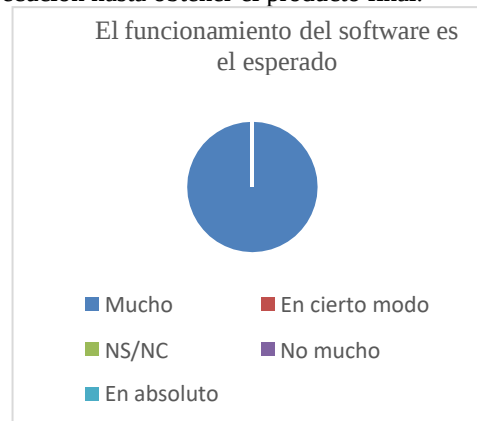


Gráfica 5. Resultado de la pregunta 5, mensajes de error con manejo de excepciones
Fuente: Elaboración propia.

El 100% de los encuestados están satisfechos con la representación de los mensajes de error cuando aparecen.

Pregunta 6: Al ejecutarse el software, se realizan diferentes tareas, todas éstas contienen

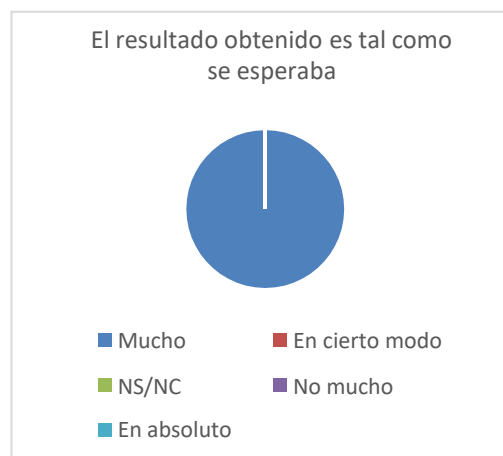
validaciones necesarias para poder continuar con la ejecución hasta obtener el producto final.



Gráfica 6. Resultado de la pregunta 6, respecto al funcionamiento del software.
Fuente: Elaboración propia.

El 100% del equipo de OI está de acuerdo en que el funcionamiento del software funciona tal como inicialmente fue trazado.

Pregunta 7: El resultado del software es una estructura de directorios con permisos de acceso a diferentes grupos de la empresa, involucrados en el proceso de producción.



Gráfica 7. Resultado de la pregunta 7, resultados obtenidos con la ejecución del software
Fuente: Elaboración propia.

Según el resultado obtenido en las primeras ejecuciones del software en la fase productiva, los encuestados convergen en que se obtuvo lo proyectado.

Los datos de la encuesta indican que el funcionamiento del software cumple con los requerimientos solicitados por los clientes a través de los resultados obtenidos en cada ítem del instrumento de investigación aplicado.

CONCLUSIONES

De los resultados obtenidos en la ejecución del software, se concluye que el funcionamiento cumple con los requerimientos planteados inicialmente por el cliente, su funcionamiento y ejecución han marcado un notable y considerable ahorro de tiempos al personal de OI, gracias al proceso de creación de directorios y estructura de números de parte demandaba un tiempo aproximado de 40 minutos, con la implementación del software, el tiempo de creación de la estructura es por debajo de 60 segundos, logrando así, un ahorro del 97.5%, se deberá considerar que los tiempos varían dependiendo del número de documentos nuevos a procesar.

Es menester mencionar que, los tiempos de ahorro pertenecen al tiempo invertido por el personal de OI, por lo consiguiente, otros departamentos de la empresa son indirectamente beneficiados del ahorro. El proceso de creación se detonaba a solicitud de OI que significaba un estado de urgencia contar con la estructura e iniciar un proceso, originado desde el área de producción u alguna otra bajo una simbiosis corporativa vital. Con la implementación del software, se ha logrado evitar que continúe repitiéndose pues se ejecuta en lapsos preestablecidos, buscando así, que cuando se tenga que crear una nueva estructura, el proceso sea ejecutado.

En síntesis, la implementación del proyecto en la empresa PWO S.A. de C.V. representa una disminución en los tiempos de creación y espera, directamente a los integrantes del departamento de OI y como consecuencia, al resto de los departamentos involucrados en el proceso de producción, mejorando la eficiencia de los procesos productivos.

REFERENCIAS

[1] Progress-Werk Oberkirch AG. (2020). *WHO WE ARE & WHERE WE ARE LOCATED*. Obtenido de <https://www.progress-werk.de/en/investors-press/about-pwo/>

[2] Wilson, E. (2008). *Windows Powershell Scripting Guide*. Washington, Estados Unidos de Norteamérica: Microsoft Press

[3] CA ARCserve. (2012). *Guía de comandos de PowerShell*. Islandia, Nueva York: CA Technologies. Obtenido de

<https://documentation.arcserve.com/Arcserve-RHA/Available/R16/ESP/Bookshelf Files/PDF/XO POWER W ESN.pdf>

[4] Ball, D. (2005). *Migración de aplicaciones de software del sistema de información de deuda de PEMEX a tecnología .NET*. Ciudad de México: Universidad Nacional Autónoma de México. Obtenido de

<http://132.248.9.195/ptd2005/00327/0348907/Ind ex.html>

[5] Cisco Systems, Inc. (2015). *Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S*. San Jose: Cisco Systems, Inc. Obtenido de

https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/xs-3s/sec-data-acl-xe-3s-book.pdf

[6] Maida, E. G., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Buenos Aires: Universidad Católica de Argentina. Obtenido de

<http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>

[7] Olmeda de la Casa, F. M. (2019). *Despliegue de imágenes del sistema operativo en aulas departamentales con OpenGnsys y posterior automatización de procesos mediante Powershell*. Escuela superior de ingeniería informática. Valencia, España: Universidad Politécnica de Valencia. Obtenido de

<https://riunet.upv.es/bitstream/handle/10251/127678/Olmeda%20-%20Despliegue%20de%20im%20c3%a1genes%20del%20sistema%20operativo%20en%20aulas%20departamentales%20con%20OpenGnsys%20y%20posterior%20automatizaci%20c3%b3n%20de%20procesos%20mediante%20Powershell.pdf>

[8] Huera Paltán, B. P. (2014). *Generación de scripts con Microsoft Powershell para la administración de servidor Microsoft de la academia Microsoft epoch*. Facultad de informática y electrónica. Riobamba, Ecuador: Escuela superior politécnica de chimborazo. Obtenido de

<http://dspace.epoch.edu.ec/bitstream/123456789/3316/1/18T00543.pdf>

