

Exposición sobre Kubernetes, Hadoop y Spark

Autores: Ericksson Estévez¹, Jorge Gualpa¹

¹Facultad de Ciencias de la Ingeniería – Universidad Técnica Estatal de Quevedo (UTEQ) -
Quevedo – Los Ríos – Ecuador
[ericksson.estevez2016, jorge.gualpa2015]@uteq.edu.ec

Resumen. Kubernetes es una herramienta de orquestación de contenedores que permite la gestión automatizada de aplicaciones en contenedores, mientras que Hadoop es un marco de procesamiento distribuido que se utiliza para el almacenamiento y análisis de grandes conjuntos de datos. Spark, por otro lado, es un motor de procesamiento de datos en memoria que puede trabajar con conjuntos de datos mucho más grandes y de manera más eficiente que Hadoop. Al utilizar Kubernetes para orquestar contenedores de Spark y Hadoop, las organizaciones pueden crear soluciones de big data que sean altamente escalables y tolerantes a fallos. Además, Kubernetes proporciona herramientas para la gestión de recursos y la programación de tareas, lo que permite a los desarrolladores implementar soluciones de big data de manera más rápida y eficiente.

Palabras claves: Kubernetes, Hadoop, Spark, Contenedores

1 Introducción

En la actualidad, la industria de la tecnología se encuentra en constante evolución y con ella, surgen nuevas herramientas y tecnologías que buscan mejorar la eficiencia y escalabilidad de las aplicaciones y el procesamiento de datos. Entre las tecnologías más destacadas en este campo se encuentran Kubernetes, Hadoop y Spark[1].

Kubernetes es una plataforma de orquestación de contenedores que permite gestionar y escalar aplicaciones en un entorno distribuido y escalable. Hadoop, por su parte, es un framework de procesamiento distribuido de grandes volúmenes de datos que permite el procesamiento de datos a gran escala en múltiples nodos. Y Spark es un framework de procesamiento de datos de código abierto que se ejecuta en clústeres de servidores y ofrece una forma más rápida y eficiente de procesar grandes volúmenes de datos que Hadoop.

Cada una de estas tecnologías ofrece soluciones únicas para diferentes necesidades de computación, y su popularidad y adopción continúan creciendo en la industria tecnológica. En este sentido, es importante conocer las características y beneficios de estas herramientas para poder aprovechar al máximo su potencial en el desarrollo y procesamiento de aplicaciones y datos.

2 Contenedores

Los contenedores son una tecnología de virtualización de sistemas operativos que permite empaquetar y distribuir aplicaciones junto con sus dependencias en un entorno aislado y portátil. Los contenedores son una forma de encapsular una aplicación y todas sus dependencias, como bibliotecas y otros componentes, en un paquete autónomo que se puede ejecutar en cualquier sistema operativo que tenga un motor de contenedores compatible[2].

A diferencia de la virtualización tradicional, donde se crea una máquina virtual que contiene un sistema operativo completo, los contenedores comparten el kernel del sistema operativo anfitrión, lo que los hace más ligeros, más rápidos y eficientes. Además, los contenedores se pueden ejecutar en diferentes plataformas, desde ordenadores de sobremesa hasta servidores en la nube [3].

Los contenedores se han vuelto cada vez más populares en el desarrollo de aplicaciones debido a su capacidad para proporcionar un entorno de ejecución consistente y portátil que se puede mover fácilmente de un entorno a otro, lo que ayuda a reducir los problemas de compatibilidad y acelera el proceso de desarrollo y despliegue de aplicaciones [3].

Algunos ejemplos de motores de contenedores populares son Docker, Podman y rkt. Estos motores permiten a los usuarios crear, desplegar y administrar contenedores de manera eficiente y consistente.

2.1 Ventajas de Contenedores

Los contenedores ofrecen varias ventajas sobre otras formas de virtualización y desarrollo de aplicaciones. Algunas de las principales ventajas son:

- **Portabilidad:** Los contenedores son portátiles y pueden ser ejecutados en diferentes plataformas y entornos, incluyendo nubes públicas y privadas, servidores bare-metal y dispositivos IoT [2].
- **Aislamiento:** Los contenedores ofrecen un alto nivel de aislamiento, lo que significa que las aplicaciones en contenedores pueden ser ejecutadas sin interferir con otras aplicaciones o componentes del sistema. Esto reduce el riesgo de conflictos y errores en el sistema [1].
- **Escalabilidad:** Los contenedores son fácilmente escalables, lo que significa que pueden ser duplicados rápidamente para satisfacer las demandas de una carga de trabajo en particular. Esto permite a las aplicaciones escalar horizontalmente y afrontar de manera efectiva picos de demanda [2].
- **Eficiencia:** Los contenedores son más eficientes que las máquinas virtuales tradicionales, ya que comparten el kernel del sistema operativo anfitrión, lo que reduce la sobrecarga de recursos y mejora el rendimiento [4].
- **Facilidad de gestión:** Los contenedores son fáciles de gestionar y mantener, ya que pueden ser gestionados y orquestados a través de herramientas como Kubernetes, Docker Swarm y OpenShift [4].

3 Kubernetes

Kubernetes es un software de código abierto que ofrece una solución eficiente para implementar y administrar contenedores a gran escala. De hecho, la palabra "Kubernetes" proviene del griego y significa "timonel de un buque o piloto", lo que refleja su capacidad para navegar y controlar aplicaciones en un entorno complejo. [5].

Con Kubernetes, es posible crear, entregar y escalar aplicaciones en contenedores de manera más rápida y sencilla. Agrupa los contenedores que conforman una aplicación en unidades lógicas, lo que permite una gestión y descubrimiento eficiente. Además, Kubernetes cuenta con la experiencia de Google de más de 15 años en la ejecución de cargas de trabajo de producción, combinada con las mejores prácticas de la comunidad.[6].

De hecho, Kubernetes se ha convertido en un estándar de facto para la orquestación de contenedores y se ha convertido en un tipo de sistema operativo de clúster para cargas de trabajo nativas de la nube. Es utilizado por más de cien proveedores y tiene el potencial de proteger a los clientes del bloqueo del proveedor [7].

3.1 Componentes de kubernete

A continuación, se describen algunos de los componentes clave de Kubernetes:

- **Pod:** Un pod es la unidad más pequeña que puede ser desplegada y gestionada en Kubernetes. Un pod contiene uno o más contenedores que comparten el mismo espacio de red y recursos [1].
- **ReplicaSet:** Un ReplicaSet es un controlador que garantiza que un número específico de réplicas de un pod se estén ejecutando en el clúster en todo momento [8].
- **Deployment:** Un Deployment es un objeto de Kubernetes que define cómo se deben desplegar y actualizar los pods y los ReplicaSets [1].
- **Servicio:** Un Servicio es un objeto de Kubernetes que expone los pods como servicios de red estables a través de una dirección IP y un nombre de DNS [8].
- **Namespace:** Un Namespace es un objeto de Kubernetes que se utiliza para crear una división lógica en un clúster y separar los recursos en diferentes entornos [8].

3.2 Historia y evolución de Kubernetes

Kubernetes fue iniciado en el año 2014 por Google, como un proyecto de código abierto basado en su experiencia interna en la gestión de contenedores a gran escala. Desde entonces, ha evolucionado y crecido gracias a la contribución de una gran comunidad de desarrolladores y empresas que han adoptado y contribuido al proyecto. Actualmente, Kubernetes es uno de los proyectos de código abierto más populares y de mayor crecimiento, con una amplia adopción en la industria de la tecnología[9].

3.3 Como funciona Kubernetes

Kubernetes funciona mediante la creación de un clúster de servidores, en el cual cada servidor puede ser un nodo maestro o un nodo de trabajo. El nodo maestro es responsable de la gestión y el control del clúster, mientras que los nodos de trabajo son los encargados de ejecutar las aplicaciones y servicios en contenedores [9].

La arquitectura de Kubernetes se basa en el concepto de "pods", que son la unidad básica de despliegue en Kubernetes. Cada pod puede contener uno o varios contenedores, y se encarga de mantenerlos juntos y proporcionarles un espacio de red compartido. Los pods son escalables horizontalmente, lo que significa que se pueden crear múltiples copias de un mismo pod para manejar cargas de trabajo más grandes[10].

Kubernetes también utiliza un sistema de programación de recursos para garantizar que los contenedores tengan acceso a los recursos necesarios, como CPU y memoria. Los usuarios pueden definir límites y solicitudes de recursos para cada contenedor, lo que permite a Kubernetes asignar recursos de manera eficiente y garantizar un rendimiento óptimo [10].

3.4 Arquitectura de Kubernetes

La arquitectura de Kubernetes se divide en dos partes principales: el plano de control (control plane) y los nodos de trabajo (worker nodes).

Plano de control (Control Plane):

El plano de control de Kubernetes es el cerebro de la plataforma y consta de los siguientes componentes:

- **API Server:** Es el punto de entrada para la gestión de Kubernetes y proporciona una API RESTful para interactuar con los diferentes componentes de Kubernetes [10].
- **Etcd:** Es una base de datos distribuida y consistente que almacena el estado del clúster, como la configuración y los metadatos de los objetos de Kubernetes [10].
- **Scheduler:** Es responsable de programar los pods en los nodos disponibles [10].
- **Controller Manager:** Es un conjunto de controladores que supervisan y controlan el estado del clúster [9].

Nodos de trabajo (Worker Nodes):

Los nodos de trabajo son los componentes que ejecutan los contenedores. Cada nodo de trabajo tiene los siguientes componentes:

- **Kubelet:** Es el agente que se ejecuta en cada nodo de trabajo y se encarga de la gestión de los pods y sus contenedores [9].
- **Kube-proxy:** Es responsable de la comunicación de red entre los diferentes servicios y pods del clúster [10].

- **Container Runtime:** Es el motor que ejecuta los contenedores en los nodos de trabajo [9].

3.5 Ventajas de Kubernetes

- **Escalabilidad:** Kubernetes permite escalar de forma automática y rápida las aplicaciones según las necesidades de tráfico y carga de trabajo. Además, facilita la gestión de múltiples réplicas de una aplicación [8].
- **Alta disponibilidad:** Kubernetes garantiza la alta disponibilidad de las aplicaciones a través de la monitorización constante de los nodos y la capacidad de reemplazar automáticamente los contenedores en caso de fallos [9].
- **Portabilidad:** Kubernetes es compatible con múltiples proveedores de servicios en la nube, lo que facilita la portabilidad de las aplicaciones entre diferentes proveedores y entornos [10].
- **Actualizaciones sin interrupciones:** Las actualizaciones y los parches pueden realizarse sin interrupciones en la disponibilidad de las aplicaciones, lo que garantiza que los usuarios finales no experimenten ningún tiempo de inactividad [11].
- **Automatización:** Kubernetes automatiza muchas tareas relacionadas con la gestión de aplicaciones y la infraestructura, lo que reduce la carga de trabajo de los administradores de sistemas y desarrolladores [8].
- **Control y gestión centralizados:** Kubernetes proporciona una interfaz centralizada para la gestión y el control de aplicaciones, lo que facilita la implementación y el mantenimiento de la infraestructura [10].
- **Seguridad:** Kubernetes ofrece múltiples opciones de seguridad para proteger las aplicaciones y los datos, como la autenticación, autorización y la gestión de identidades [8].

3.6 Desventajas de Kubernetes

Aunque Kubernetes es una plataforma ampliamente utilizada para la gestión de aplicaciones en contenedores, también presenta algunas desventajas potenciales, que incluyen:

- **Complejidad:** Kubernetes puede tener una curva de aprendizaje pronunciada y puede resultar complejo de configurar, administrar y operar, especialmente para aquellos que son nuevos en la tecnología de contenedores y la orquestación de aplicaciones [8].
- **Requisitos de recursos:** Kubernetes requiere una infraestructura de hardware y recursos computacionales adecuados para funcionar correctamente. Esto puede implicar un aumento en los costos de infraestructura y recursos, lo que podría ser una desventaja para organizaciones con recursos limitados [11].
- **Configuración y administración:** La configuración y administración de clústeres de Kubernetes puede ser compleja, lo que puede requerir un esfuerzo significativo

para asegurarse de que esté correctamente configurado y asegurado, lo que podría suponer un desafío para algunos equipos de TI [8].

- **Curva de aprendizaje:** La comprensión completa de los conceptos y términos específicos de Kubernetes puede requerir tiempo y esfuerzo, lo que puede representar una barrera para aquellos que no están familiarizados con la tecnología de contenedores y la orquestación de aplicaciones [1].
- **Posibles problemas de compatibilidad:** A medida que Kubernetes sigue evolucionando, puede haber cambios y actualizaciones que pueden tener implicaciones en la compatibilidad con versiones anteriores o en la integración con otras herramientas o servicios [1].
- **Complejidad en la gestión de almacenamiento y redes:** La gestión de almacenamiento y redes en Kubernetes puede resultar compleja y requiere una configuración y administración cuidadosas para garantizar un rendimiento y seguridad adecuados [8].
- **Dependencia de la infraestructura:** Kubernetes depende de una infraestructura de contenedores subyacente, como Docker, lo que significa que cualquier cambio o problema en esa infraestructura puede afectar el funcionamiento de Kubernetes [4].

3.7 Como Realizar la instalación de Kubernetes en Windows

Para instalar Kubernetes primero instalamos kubectl ubicando en el cmd el siguiente comando:

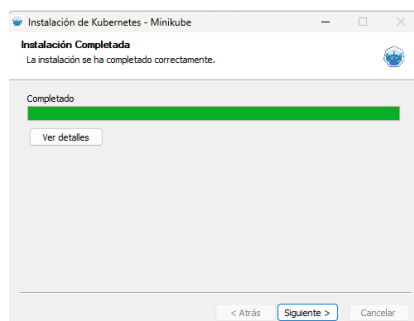
```
curl -LO https://dl.k8s.io/release/v1.27.0/bin/windows/amd64/kubectl.exe
```

para verificar la versión utilizamos el siguiente comando:

```
kubectl version --client=true
```

Para instalar el custle de Kubernetes descargamos y instalamos minikube en el siguiente enlace:

<https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe>

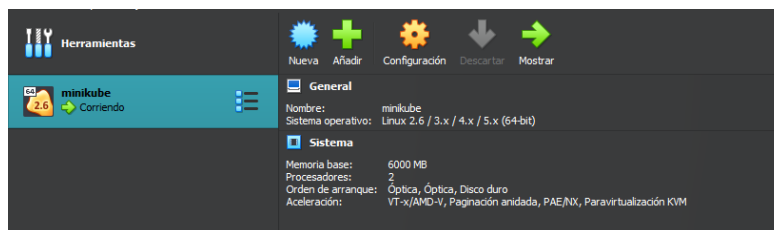


Para arrancar minikube ubicamos:

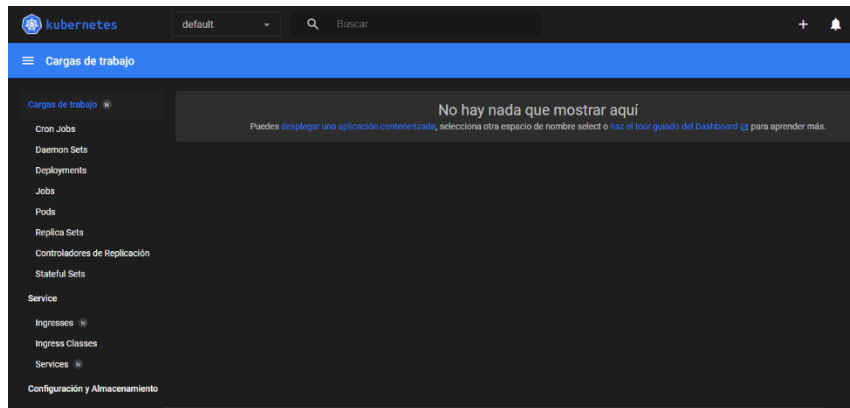
minikube start

```
C:\Windows\System32>minikube start
* minikube v1.30.1 en Microsoft Windows 11 Pro 10.0.22623.1325 Build 22623.1325
* Controlador virtualbox seleccionado automáticamente
* Descargando la imagen de arranque de la VM
  > minikube-v1.30.1-amd64.iso...: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > minikube-v1.30.1-amd64.iso: 19.03 MiB / 282.84 MiB 6.73% 5.74 MiB p/s E_
```

Esto nos crea una máquina virtual de minikube:

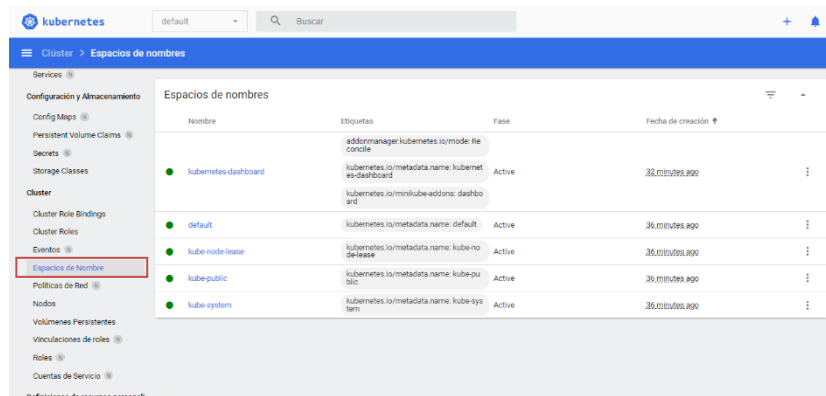


Para entrar a la herramienta grafica utilizamos el comando: minikube dashboard



Para pararlo utilizamos el comando: minikube stop El cual tambien parara la máquina virtual.

Como podemos ver namespaces en Kubernetes donde podemos ver la información como despliegues, etc.



Y para saberlo por medio de la línea de comando es la siguiente:
 kubectl get namespaces

```
C:\Windows\System32>kubectl get namespaces
NAME                STATUS    AGE
default             Active    42m
kube-node-lease     Active    42m
kube-public         Active    42m
kube-system         Active    42m
kubernetes-dashboard Active    37m

C:\Windows\System32>
```

Como crear un namespaces: Lo tenemos que realizar por medio de comando

```
C:\Windows\System32>kubectl create namespace eduardo-estevez
namespace/eduardo-estevez created
```

Y como vemos el namespaces fue creado con éxito

```
C:\Windows\System32>kubectl get namespaces
NAME                STATUS    AGE
default             Active    45m
eduardo-estevez     Active    2s
kube-node-lease     Active    45m
kube-public         Active    45m
kube-system         Active    45m
kubernetes-dashboard Active    40m
```


Espacios de nombres				
Nombre	Etiquetas	Fase	Fecha de creación ↑	
● eduardo-estevez	kubernetes.io/metadata.name: eduardo-estevez	Active	a minute ago	⋮
● kubernetes-dashboard	addonmanager.kubernetes.io/mode: Reconcile	Active	42 minutes ago	⋮
	kubernetes.io/metadata.name: kubernetes-dashboard			
	kubernetes.io/minikube-addons: dashboard			
● default	kubernetes.io/metadata.name: default	Active	46 minutes ago	⋮
● kube-node-lease	kubernetes.io/metadata.name: kube-node-lease	Active	46 minutes ago	⋮
● kube-public	kubernetes.io/metadata.name: kube-public	Active	46 minutes ago	⋮
● kube-system	kubernetes.io/metadata.name: kube-system	Active	46 minutes ago	⋮

Para eliminar el namespaces ubicamos el siguiente comando:

```
kubectl delete namespace eduardo-estevez
```

```
C:\Windows\System32>kubectl delete namespace eduardo-estevez
namespace "eduardo-estevez" deleted

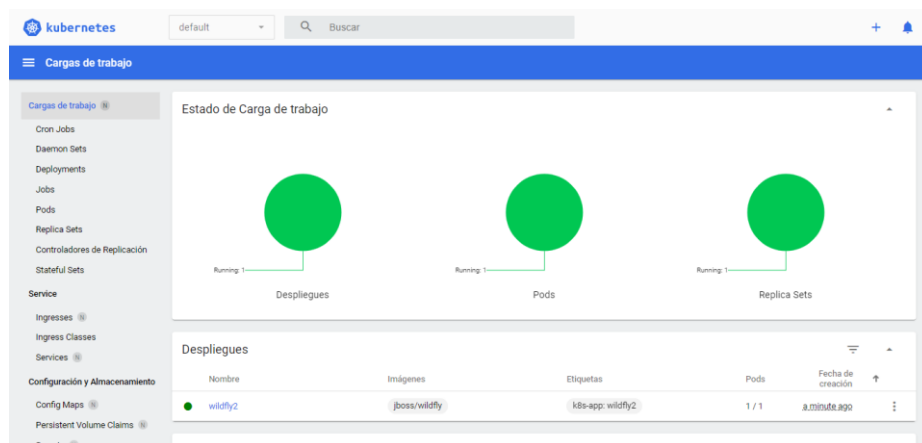
C:\Windows\System32>kubectl get namespaces
NAME                STATUS    AGE
default             Active    49m
kube-node-lease     Active    49m
kube-public         Active    49m
kube-system         Active    49m
kubernetes-dashboard Active    44m
```

Crear pod le damos en el mas que esta en la esquina parte superior

Etiquetas	Fase	Fecha de creación ↑	
addonmanager.kubernetes.io/mode: Reconcile	Active	55 minutes ago	⋮
kubernetes.io/metadata.name: kubernetes-dashboard			
kubernetes.io/minikube-addons: dashboard			
kubernetes.io/metadata.name: default	Active	59 minutes ago	⋮
kubernetes.io/metadata.name: kube-node-lease	Active	59 minutes ago	⋮

Llenamos los campos de los formularios

Y vemos que ya esta cargado el pod



Para ver por la línea de comando es:

```
kubectl get pods
```

```
C:\Windows\System32>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
wildfly2-76ff9769bd-dhvrh          1/1     Running   0           2m47s
C:\Windows\System32>
```

Y con el siguiente comando podemos ver la descripción del pod que hemos creado:

kubectl describe pod wldfly2

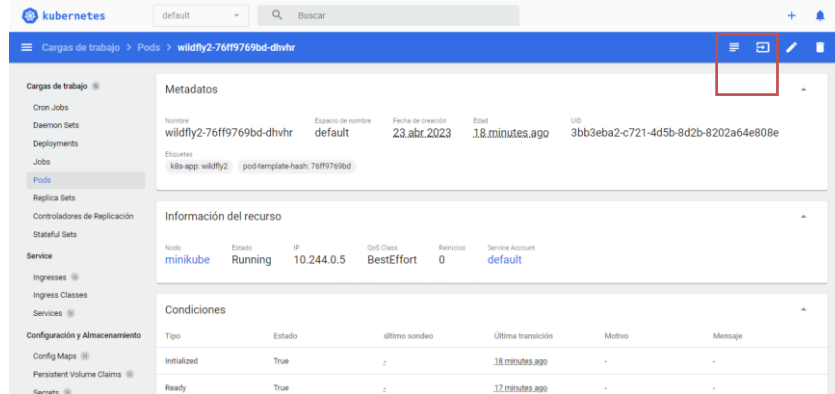
```

C:\Windows\System32\kubectl describe pod wldfly2
Error from server (NotFound): pods "wldfly2" not found

C:\Windows\System32\kubectl describe pod wldfly2
Name:         wldfly2-76ff9769bd-dhvr
Namespace:    default
Priority:      0
Service Account: default
Node:         minikube/192.168.59.100
Start Time:   Sun, 23 Apr 2023 18:48:33 -0500
Labels:       k8s-app=wldfly2
              pod-template-hash=76ff9769bd
Annotations:  <none>
Status:       Running
IP:           10.244.0.5
IPs:          10.244.0.5
Controlled By: ReplicaSet/wldfly2-76ff9769bd
Containers:
  wldfly2:
    Container ID:  docker://8513d71221ff5c1c4f1bfeec4d42e412c68be53fb99f45b079bcb5562d40eb
    Image:          jboss/wildfly
    Image ID:       docker-pullable://jboss/wildfly@sha256:35320abaFdec6d360559b411aff466514d5741c3c527221445f48246359fdfe5
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sun, 23 Apr 2023 18:49:45 -0500
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-58x5v (ro)
Conditions:
  Type              Status
  Initialized        True
  Ready              True
  ContainersReady    True
  PodScheduled       True
Volumes:
  kube-api-access-58x5v:

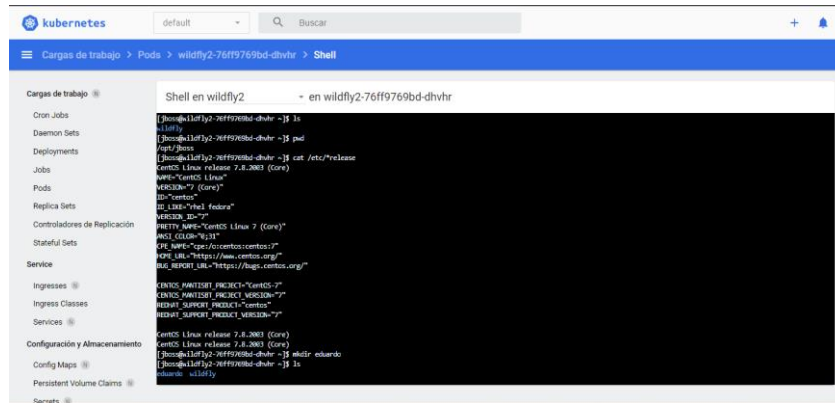
```

Para acceder a un pod podemos ir a la siguiente parte:



The screenshot shows the Kubernetes dashboard interface. The top navigation bar includes the 'kubernetes' logo, a dropdown menu set to 'default', and a search bar. The left sidebar lists various Kubernetes resources: 'Cargas de trabajo' (Jobs), 'Cron Jobs', 'Daemon Sets', 'Deployments', 'Jobs', 'Pods' (selected), 'Replica Sets', 'Controladores de Replicación', 'Statful Sets', 'Service', 'Ingresses', 'Ingress Classes', 'Services', 'Configuración y Almacenamiento', 'Config Maps', 'Persistent Volume Claims', and 'Secrets'. The main content area displays the details for the pod 'wldfly2-76ff9769bd-dhvr'. It includes a 'Metadatos' section with fields for Name, Namespace, Creation Time, Age, and UID. The 'Información del recurso' section shows the pod's status as 'Running' on the 'minikube' node with IP '10.244.0.5'. The 'Condiciones' section shows a table with columns for Type, Status, Last Probe, Last Transition, Reason, and Message, with rows for 'Initialized' and 'Ready' both showing 'True' status.

Nos presentará lo siguiente donde podemos ver que es una terminación igual a Linux



Otra forma seria por el siguiente comando:

```
exec -it wildfly2-76ff9769bd-dhvhhr -- /bin/bash
```

```

C:\Windows\System32>kubectl exec -it wildfly2-76ff9769bd-dhvhhr -- /bin/bash
[jboss@wildfly2-76ff9769bd-dhvhhr ~]$ ls
jboss  wildfly
[jboss@wildfly2-76ff9769bd-dhvhhr ~]$ pwd
/opt/jboss
[jboss@wildfly2-76ff9769bd-dhvhhr ~]$

```

4 Hadoop

4.1 Que es Apache Hadoop

Debemos entender que Apache Hadoop es un framework de software que aporta la capacidad de ejecutar aplicaciones distribuidas y escalables, generalmente para el sector del Big Data. Así, permite a las aplicaciones hacer uso de miles de nodos de procesamiento y almacenamiento y petabytes de datos [12].

Hadoop es una de las tecnologías más populares en el ámbito de aplicaciones Big Data. Es usado en multitud de empresas como plataforma central en sus Data Lakes (Lagos de datos), sobre la que se construyen los casos de uso alrededor de la explotación y el almacenamiento de los datos [12].

Hadoop es un sistema de procesamiento distribuido que utiliza dos componentes principales para procesar grandes cantidades de datos en clústeres de servidores. El sistema de archivos distribuidos de Hadoop (HDFS) es utilizado para el almacenamiento de datos, mientras que el tiempo de ejecución de MapReduce es utilizado para el procesamiento de datos. MapReduce se basa en la idea de dividir los datos en pequeñas tareas (map) y luego combinarlos para producir un único resultado

(reduce), lo que permite procesar grandes cantidades de datos de manera eficiente. Hadoop es ampliamente utilizado por empresas grandes y pequeñas para procesar grandes cantidades de datos, como registros del servidor, análisis de datos de redes sociales, análisis de datos publicitarios, y otros fines similares [13].

4.2 Historia

En el año 2006, los dos componentes que formaban parte de Hadoop: MapReduce y HDFS se cedieron a la Apache Software Foundation como proyecto open source. Esto impulsó su adopción como herramienta Big Data en proyectos en muchas industrias. El proyecto fue desarrollado en el lenguaje de programación Java [14].

La versión 1.0 de Hadoop fue publicada en el año 2012. La versión 2.0 se publicó en el año 2013 añadiendo Yarn como gestor de recursos y desacoplando HDFS de MapReduce. En el año 2017 se publicó Hadoop 3.0 añadiendo mejoras [12].

4.3 Arquitectura de Hadoop

Anteriormente, en los sistemas tradicionales, las tecnologías se han enfocado en traer los datos a los sistemas de almacenamiento. Sin embargo, en los procesos Hadoop, se trata de acercar el procesamiento al lugar en donde se encuentran almacenados los datos y así aprovechar técnicas de paralelización, aumentando de manera importante la escalabilidad y el rendimiento de los sistemas que trabajan con grandes cantidades de datos [14].

La arquitectura de Hadoop y su diseño está basado en la idea de que mover el procesamiento es mucho más rápido, fácil y eficiente que mover grandes cantidades de datos, que pueden producir altas latencias y congestión en la red. El sistema de ficheros distribuido de Hadoop (HDFS) proporciona a las aplicaciones la capacidad de acceder a los datos en el lugar en el que se encuentren almacenados [15].

4.4 Componentes Principales

Para comprender completamente cómo funciona Hadoop debemos entender sus tres componentes principales. Hemos dedicado una entrada a cada uno de ellos:

- **MapReduce:** Hadoop MapReduce es un paradigma de procesamiento de datos caracterizado por dividirse en dos fases o pasos diferenciados: **Map y Reduce**. Estos subprocesos asociados a la tarea se ejecutan de manera distribuida, en diferentes nodos de procesamiento o esclavos. Para controlar y gestionar su ejecución, existe un proceso Master o *Job Tracker*. También es el encargado de aceptar los nuevos trabajos enviados al sistema por los clientes [13].
- **Yarn (Yet Another Resource Negotiator):** Yarn (Yet Another Resource Negotiator) es una pieza fundamental en el ecosistema Hadoop. Es el framework que permite a Hadoop soportar varios motores de ejecución incluyendo MapReduce, y proporciona un planificador agnóstico a los trabajos que se

encuentran en ejecución en el clúster. Esta mejora de Hadoop también es conocida como Hadoop 2. Yarn separa las dos funcionalidades principales: la gestión de recursos y la planificación y monitorización de trabajos. Con esta idea, es posible tener un gestor global (Resource Manager) y un Application Master por cada aplicación [12].

- **HDFS (Hadoop Distributed File System):** HDFS (Hadoop Distributed File System) es el componente principal del ecosistema Hadoop. Esta pieza hace posible almacenar data sets masivos con tipos de datos estructurados, semi-estructurados y no estructurados como imágenes, vídeo, datos de sensores, etc. Está optimizado para almacenar grandes cantidades de datos y mantener varias copias para garantizar una alta disponibilidad y la tolerancia a fallos. Con todo esto, HDFS es una tecnología fundamental para Big Data, o, dicho de otra forma, es el *Big Data File System* o almacenamiento Big Data por excelencia [15].

4.5 Tecnologías relacionadas

Los proyectos open source más populares del ecosistema formado alrededor de Apache Hadoop seguramente te sonarán. Se trata de los siguientes:

- **Spark:** Motor de procesamiento en memoria compatible con HDFS. Aumenta la velocidad de MapReduce en 100 veces. Soporta aplicaciones ETL, Machine learning y Streaming de datos así como consultas SQL [15].
- **Ambari:** Herramienta para gestionar y provisionar clústers de Apache Hadoop y tecnologías relacionadas. Proporciona una interfaz web sencilla y amigable para visualizar y monitorizar el estado del sistema y de todos sus componentes, así como establecer alertas y visualizar estadísticas [15].
- **Oozie:** Permite ejecutar y planificar en el tiempo trabajos y tareas en Hadoop mediante configuraciones XML [15].
- **Pig:** Proporciona el lenguaje de programación Pig Latin, con sintaxis parecida a SQL. Transforma los programas en sentencias MapReduce que ejecutan en un clúster Hadoop [13].
- **Storm:** Componente encargado de procesar flujos de datos en tiempo real. Su uso suele ir acompañado de Apache Kafka [13].
- **Tez:** Framework de programación de flujos de datos. Es la evolución de **MapReduce** que ejecuta sobre Yarn optimizando el código para alcanzar mejoras de hasta 10 veces en el rendimiento. Muchas tecnologías están adoptando Tez como motor de ejecución principal [12].
- **Zookeeper:** Servicio de coordinación para aplicaciones distribuidas, tolerante a fallos. Generalmente, se despliega en 3 nodos [12].

4.6 La función de Hadoop en la IoT (Internet de las cosas)

Una solución que ofrece Hadoop es la capacidad de almacenar y analizar cantidades masivas de datos. Los big data continúan creciendo cada vez más. Cinco años atrás, generábamos un poco más de la mitad de los datos que generamos en la actualidad. Hoy en día, creamos más datos en tres minutos que los que generábamos en un día hace quince años [15].

El motivo principal que dio lugar a este aumento masivo en la generación de datos es la ola tecnológica actual llamada la “Internet de las cosas” (o IoT, por su nombre en inglés). Esto es cuando los objetos físicos comunes se conectan a Internet y se controlan a través de dicha red. El primer paso fueron los smartphones, los televisores inteligentes y los sistemas de alarma. Ahora, se ve en electrodomésticos inteligentes, como refrigeradores, lavavajillas, termostatos, bombillas, cafeteras, cámaras de seguridad, monitores para bebés y mascotas, cerraduras, aspiradoras robot y demás dispositivos con conexión a Internet. Si bien esos electrodomésticos nos simplifican la vida, registran y almacenan datos sobre sus acciones diarias [16].

La IoT también se extiende a entornos profesionales, empresariales y gubernamentales. Las unidades de aire acondicionado inteligentes mantienen la eficiencia en los edificios y las cámaras corporales protegen tanto a agentes de policía como a civiles. Asimismo, los sensores ambientales ayudan a los gobiernos a responder más rápido a los desastres naturales, como terremotos e incendios forestales [16].

En conclusión, todos estos dispositivos registran una asombrosa cantidad de datos, por lo que requieren funcionalidades de supervisión flexibles y una capacidad de adaptación asequible. Por ende, los sistemas como Hadoop suelen ser la solución adecuada para almacenar datos de la Internet de las cosas. Hadoop no es la única opción. Pero sin duda es la más popular, dada la creciente demanda de dispositivos de la Internet de las cosas [14].

4.7 Ventajas de Hadoop

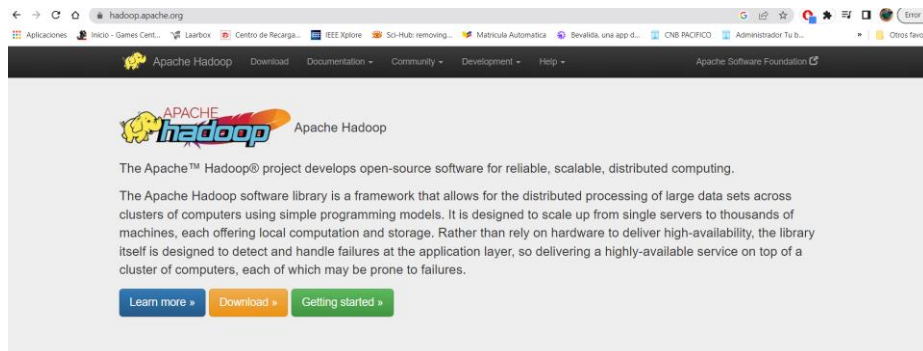
- **Escalabilidad:** esta herramienta permite almacenar y distribuir conjuntos de datos inmensos en sus cientos de servidores que operan en paralelo, permitiendo olvidarse de los límites que otras alternativas imponen [17].
- **Velocidad:** garantiza una eficiencia de procesamiento que nadie puede igualar. ¿de qué otra forma se pueden procesar terabytes de información en pocos minutos? [17]
- **Efectividad en costes:** el almacenamiento de datos se convierte en una realidad para las empresas ya que la inversión necesaria pasa de ser decenas de miles de Euros por terabyte a quedarse reducida a cientos de Euros por terabyte [18].
- **Flexibilidad:** ¿nuevas fuentes de datos? no hay problema, ¿nuevos tipos de datos? por supuesto... Apache Hadoop se adapta a las necesidades del negocio y le acompaña en su expansión, aportando soluciones reales para cualquier iniciativa que surja [18].

- **Resistencia al fracaso:** su tolerancia a errores es uno de sus atributos mejor valorados por los usuarios ya que toda la información contenida en cada nodo tiene su réplica en otros nodos del cluster. En caso de producirse un fallo siempre existirá una copia lista para ser usada [16].

4.8 Hadoop - Configuración Entorno

4.8.1 Descargar los binarios de Hadoop 3.3.5

Para descargar binarios, visite Apache.org y busque los binarios de Hadoop 3.3.5. Debería obtener el archivo `hadoop-3.3.5.tar.gz`.



Por muchas razones obvias, es posible que desee organizar su instalación adecuadamente. Por lo tanto, cree una carpeta separada donde descomprima los archivos binarios. Crearemos la carpeta "C:\hadoop-3.3.5" y la referenciaremos más, pero puede elegir cualquier opción que más le convenga.

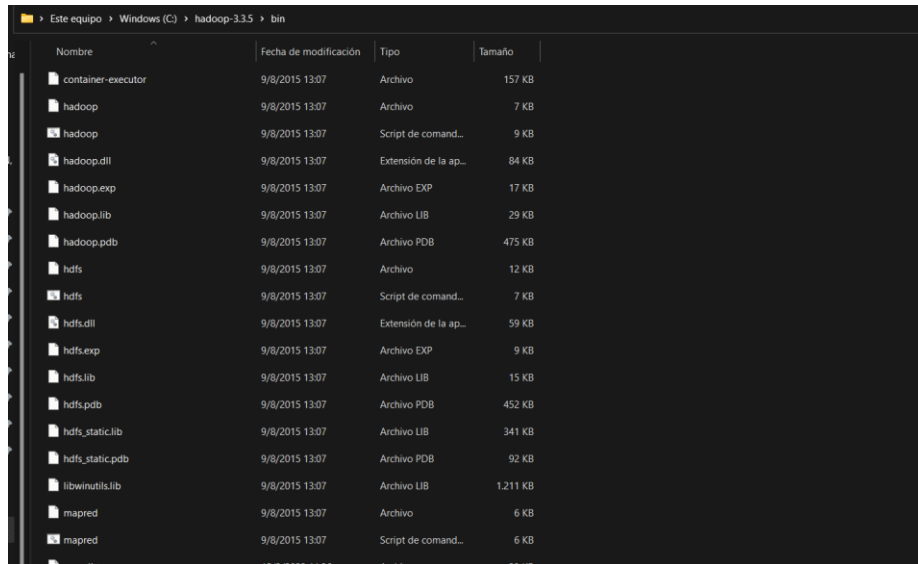
Nombre	Fecha de modificación	Tipo	Tamaño
Archivos de programa	26/4/2023 21:50	Carpeta de archivos	
Archivos de programa (x86)	16/4/2023 21:18	Carpeta de archivos	
BigDataLocal	26/4/2023 16:36	Carpeta de archivos	
ESD	11/4/2023 17:55	Carpeta de archivos	
hadoop-3.3.5	26/4/2023 22:58	Carpeta de archivos	
Intel	26/4/2023 22:45	Carpeta de archivos	
Java	26/4/2023 21:52	Carpeta de archivos	
PerfLogs	7/5/2022 0:24	Carpeta de archivos	
SWSetup	16/4/2023 1:26	Carpeta de archivos	
tmp	26/4/2023 21:38	Carpeta de archivos	
Usuarios	10/4/2023 21:04	Carpeta de archivos	
Windows	16/4/2023 1:21	Carpeta de archivos	

Descomprima la carpeta tar.gzHadoop-3.3.5 en " C:\hadoop-3.3.5"

4.8.2 Descargar binarios compatibles con Windows

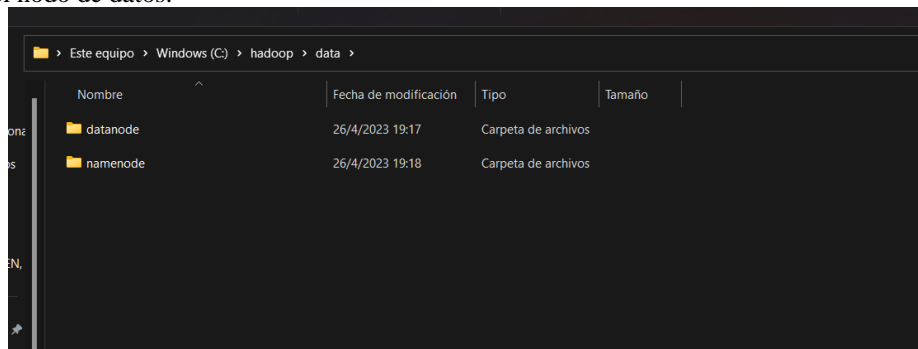
Para descargar los archivos necesarios, se descargara del siguiente repositorio “<https://drive.google.com/drive/folders/1Q8bOFv1jVNreTuE34ISGhN7wzJ5b5w1P>”

Descomprima el zip y copie todos los archivos presentes en la carpeta bin en C:\hadoop-3.3.5\bin También reemplace los archivos existentes.



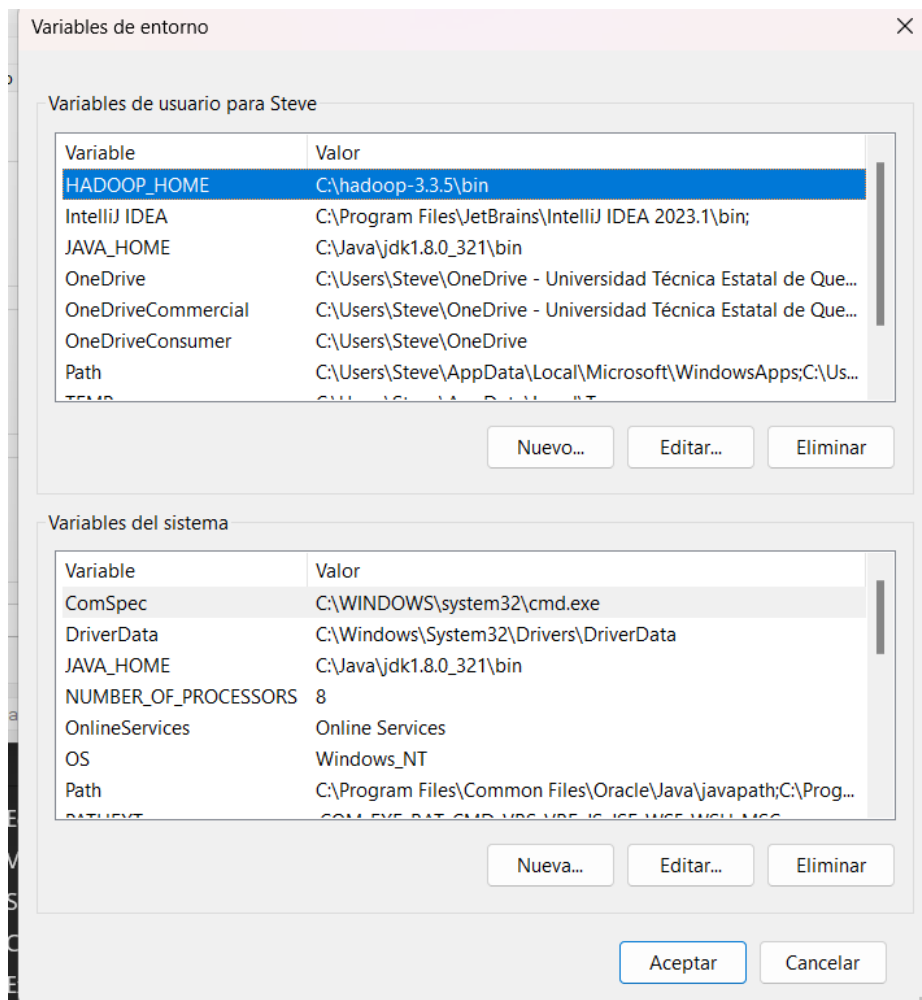
4.8.3 Crear carpetas para datanode y namenode

Vaya a C:\hadoop-3.3.5y cree una carpeta "data". En la carpeta "data", cree dos carpetas "datanode" y "namenode". Los archivos en HDFS se ubicarán en la carpeta del nodo de datos.



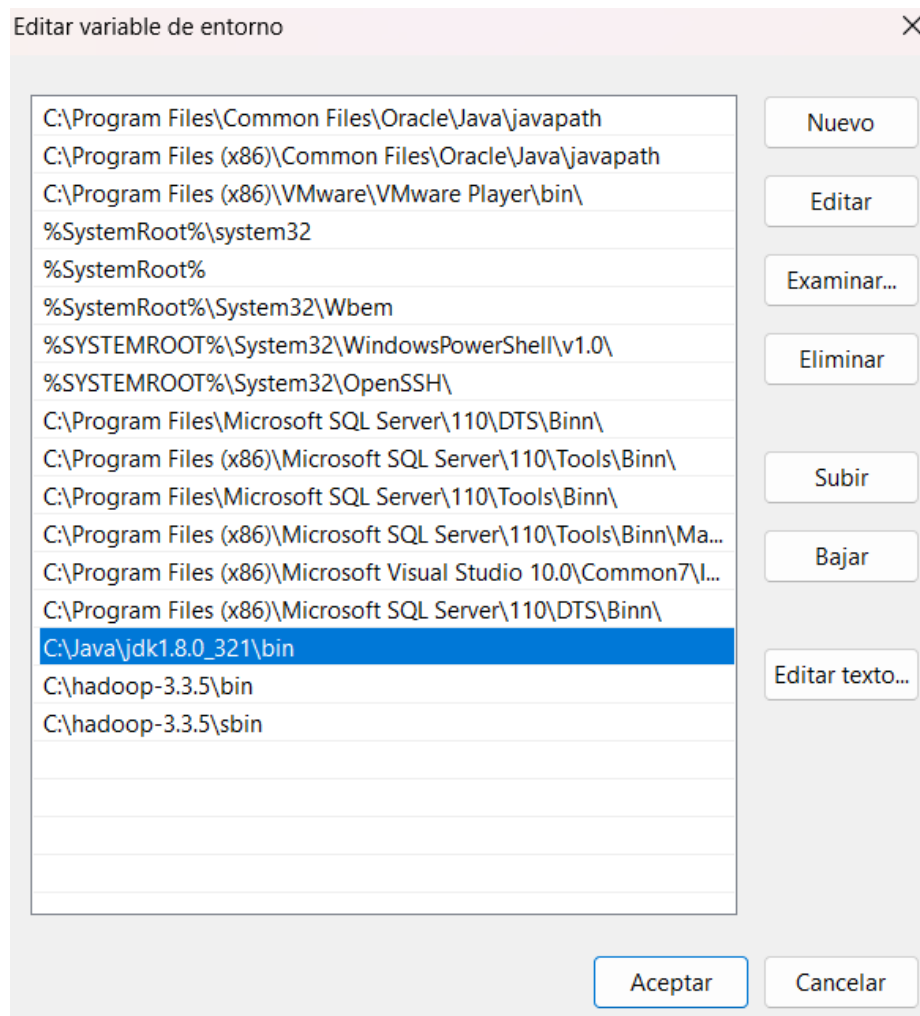
4.8.4 Establecer Variables de entorno de Hadoop

Para configurar estas variables, navegue hasta "Sistema". Haga clic en Configuración avanzada del sistema-> Variables de entorno. Inserte una descripción de la imagen aquí y haga clic en Nuevo para crear una nueva variable de entorno.



Ahora que hemos establecido las variables de entorno, debemos verificarlas. Abra un nuevo símbolo del sistema de Windows y ejecute el comando echo en cada variable para confirmar que se les han asignado los valores requeridos.

4.8.5 Establecer Variables de entorno de PATH



Ahora que hemos establecido las variables de entorno, debemos verificarlas. Abra un nuevo símbolo del sistema de Windows y ejecute el comando `echo` en cada variable para confirmar que se les han asignado los valores requeridos.

4.8.6 Configurar Hadoop

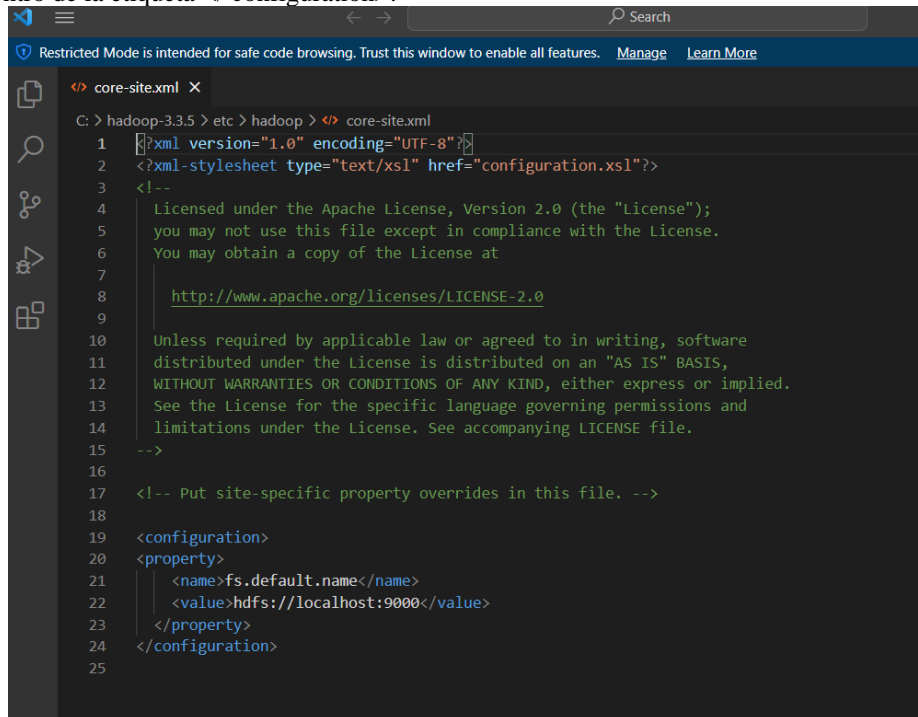
Después de configurar las variables de entorno, necesitamos configurar Hadoop editando el siguiente archivo de configuración.

```
hadoop-env.cmd
core-site.xml
hdfs-site.xml
mapred-site.xml
yarn-site
```

4.8.7 Configurar Hadoop “core-site.xml”

Ahora, configure los ajustes de Hadoop Core.

Abra `C:\hadoop-3.3.5\etc\hadoop\core-site.xml` y ábralo debajo del contenido dentro de la etiqueta `</configuration>`.

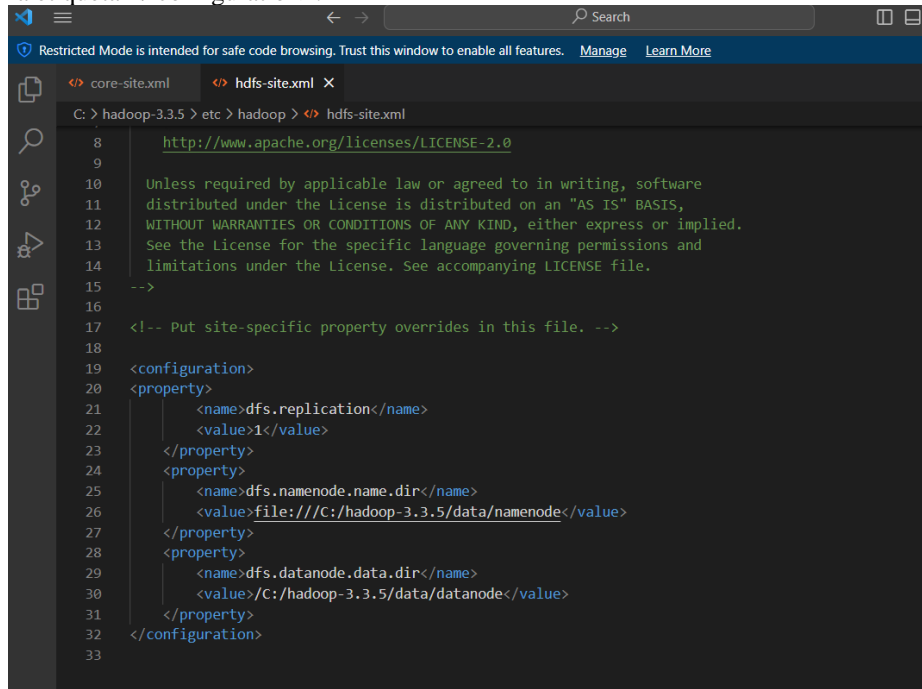


```
C: > hadoop-3.3.5 > etc > hadoop > core-site.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3  <!--
4  Licensed under the Apache License, Version 2.0 (the "License");
5  you may not use this file except in compliance with the License.
6  You may obtain a copy of the License at
7
8  http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20 <property>
21   <name>fs.default.name</name>
22   <value>hdfs://localhost:9000</value>
23 </property>
24 </configuration>
25
```

4.8.8 Configurar Hadoop “hdfs-site.xml”

Después de editar core-site.xml, debe establecer el factor de replicación y la ubicación del namenode y datanode.

Abra C:\hadoop-3.3.5\etc\hadoop\hdfs-site.xml y ábralo bajo el contenido dentro de la etiqueta </ configuration>.



```

8      http://www.apache.org/licenses/LICENSE-2.0
9
10     Unless required by applicable law or agreed to in writing, software
11     distributed under the license is distributed on an "AS IS" BASIS,
12     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13     See the license for the specific language governing permissions and
14     limitations under the license. See accompanying LICENSE file.
15     -->
16
17     <!-- Put site-specific property overrides in this file. -->
18
19     <configuration>
20     <property>
21       <name>dfs.replication</name>
22       <value>1</value>
23     </property>
24     <property>
25       <name>dfs.namenode.name.dir</name>
26       <value>file:///C:/hadoop-3.3.5/data/namenode</value>
27     </property>
28     <property>
29       <name>dfs.datanode.data.dir</name>
30       <value>C:/hadoop-3.3.5/data/datanode</value>
31     </property>
32   </configuration>
33
  
```

4.8.9 Configurar Hadoop “mapred-site.xml”

Configuremos las propiedades para el marco Map-Reduce.

Abra C:\hadoop-3.3.5\etc\hadoop\mapred-site.xml y abra debajo del contenido dentro de la etiqueta </ configuration>. Si no ve mapred-site.xml, abra el archivo mapred-site.xml.template y cámbiele el nombre a mapred-site.xml

```

C:\> hadoop-3.3.5 > etc > hadoop > mapred-site.xml

1  <?xml version="1.0"?>
2  <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3  <!--
4  Licensed under the Apache License, Version 2.0 (the "License");
5  you may not use this file except in compliance with the License.
6  You may obtain a copy of the License at
7
8  http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20 <property>
21   <name>mapreduce.framework.name</name>
22   <value>yarn</value>
23 </property>
24 </configuration>
25

```

4.8.10 Configurar Hadoop “yarn-site.xml”

Ahora, configure los ajustes de Hadoop yarn-site

Abra C:\hadoop-3.3.5\etc\hadoop\ yarn-site.xml y ábralo debajo del contenido dentro de la etiqueta </ configuration

```

C: > hadoop-3.3.5 > etc > hadoop > <> yarn-site.xml
1  <?xml version="1.0"?>
2  <!--
3  Licensed under the Apache License, Version 2.0 (the "License");
4  you may not use this file except in compliance with the License.
5  You may obtain a copy of the License at
6
7  http://www.apache.org/licenses/LICENSE-2.0
8
9  Unless required by applicable law or agreed to in writing, software
10 distributed under the License is distributed on an "AS IS" BASIS,
11 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 See the License for the specific language governing permissions and
13 limitations under the License. See accompanying LICENSE file.
14 -->
15 <configuration>
16
17 <!-- Site specific YARN configuration properties -->
18 <property>
19   <name>yarn.nodemanager.aux-services</name>
20   <value>mapreduce_shuffle</value>
21 </property>
22 <property>
23   <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
24   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
25 </property>
26 </configuration>
27

```

4.8.11 Configurar Hadoop “hadoop-env”

Agregamos la ruta del Jdk que tenemos instalado

```

Apuntes Carlos - install  hadoop-env
File Edit View
@rem The java implementation to use. Required.
set JAVA_HOME=C:\Java\jdk1.8.0_321

```

4.8.12 Inicia Hadoop

Primero abrimos un nuevo símbolo del sistema de Windows como administrador y comprobamos la versión de Hadoop version


```

Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.1555]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>hadoop version
Hadoop 3.3.5
Source code repository https://github.com/apache/hadoop.git -r 706d88266abcee09ed78fbaa0ad5f74d818ab0e9
Compiled by stevel on 2023-03-15T15:56Z
Compiled with protoc 3.7.1
From source with checksum 6bbd9afc4838a0eb12a5f189e9bd7
This command was run using /C:/hadoop-3.3.5/share/hadoop/common/hadoop-common-3.3.5.jar

C:\Windows\System32>

```

Solo por primera vez para instalar usaremos el siguiente comando
`hdfs namenode -format`

```

Administrador: Símbolo del sistema
This command was run using /C:/hadoop-3.3.5/share/hadoop/common/hadoop-common-3.3.5.jar

C:\Windows\System32>hdfs namenode -format

```

Una vez ejecutado esperamos a que termine el proceso.

```

Administrador: Símbolo del sistema
image.ckpt_000000000000000000 using no compression
2023-04-26 22:59:11,989 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop-3.3.5\data\namenode\current\fsimage.ck
pt_000000000000000000 of size 400 bytes saved in 0 seconds .
2023-04-26 22:59:12,009 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-04-26 22:59:12,024 INFO namenode.FSNamesystem: Stopping services started for active state
2023-04-26 22:59:12,024 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-04-26 22:59:12,036 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-04-26 22:59:12,036 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at BRYAN/192.168.154.1
*****/

```

Luego iniciamos Hadoop

```

Administrador: Símbolo del sistema
*****/

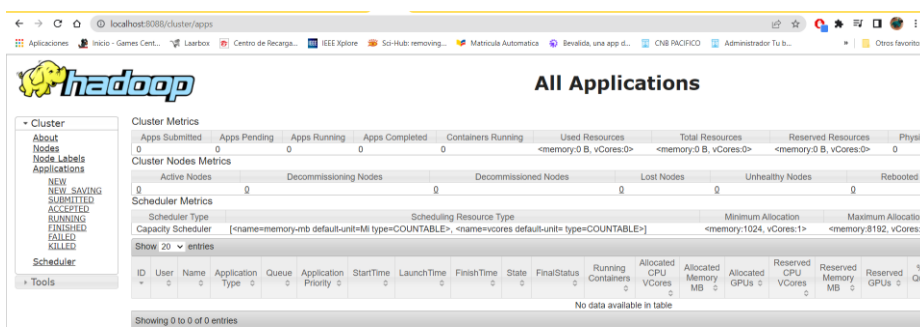
C:\Windows\System32>cd C:\hadoop-3.3.5\sbin

C:\hadoop-3.3.5\sbin>start-all
This script is deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

```

Y se abrirán 4 nuevos terminales cmd de Windows para 4 demonios, a saber, namenode, datanode, nodemanager y resourcemanager. No cierre estas ventanas, minimícelas. Cerrar la ventana terminará el demonio.

Finalmente, controlemos el funcionamiento del demonio Hadoop. Sin mencionar que puede usar la interfaz de usuario web para diversas actividades de administración y monitoreo. Abre tu navegador y comienza a usarlo.



Abrir <http://localhost:8088/cluster/apps> para abrir el administrador de recursos

Overview 'localhost:9000' (✓active)

Started:	Wed Apr 26 23:02:55 -0500 2023
Version:	3.3.5, r706d826abce0bed78baa0aed5f74d818ab0e9
Compiled:	Wed Mar 15 10:56:00 -0500 2023 by stevel from branch-3.3.5
Cluster ID:	CID-37e2a861-1c68-4606-9c02-bb741010be60
Block Pool ID:	BP-393951660-192.168.154.1-1662567951736

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 45.91 MB of 199.5 MB Heap Memory. Max Heap Memory is 889 MB.

Abra <http://localhost:9870/dfshealth.html#tab-overview> para verificar el estado de ejecución del nodo de nombre

4.9 Probando Hadoop

Creación de una carpeta para entrada

```

C:\Windows\System32>cd C:\hadoop-3.3.5\sbin

C:\hadoop-3.3.5\sbin>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.5\sbin>hadoop fs -mkdir /input
"hadoop" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

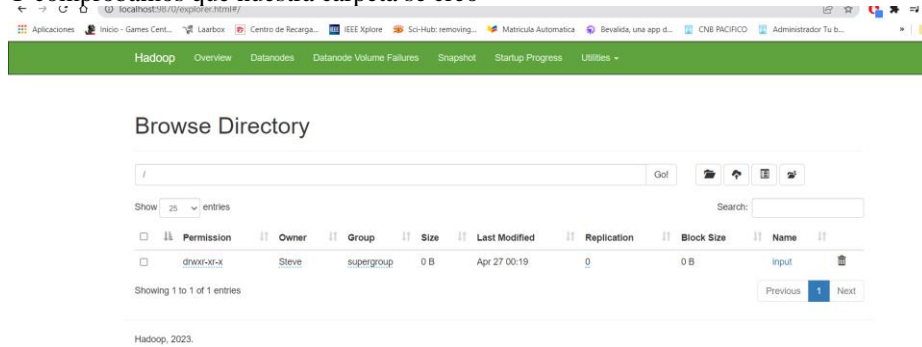
C:\hadoop-3.3.5\sbin>hadoop fs -mkdir /input

C:\hadoop-3.3.5\sbin>

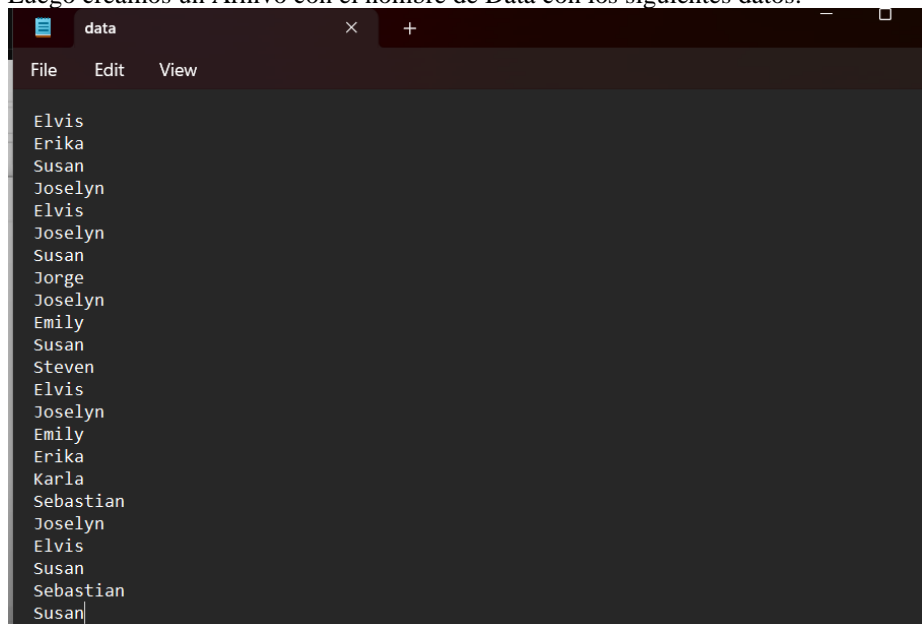
```

Nos dirigimos a Utilidades

Y comprobamos que nuestra carpeta se creo



Luego creamos un Archivo con el nombre de Data con los siguientes datos.



Y lo colocamos en el Disco local C

Este equipo > Windows (C:)

Nombre	Fecha de modificación	Tipo	Tamaño
Archivos de programa	26/4/2023 21:50	Carpeta de archivos	
Archivos de programa (x86)	16/4/2023 21:18	Carpeta de archivos	
BigDataLocal	26/4/2023 16:36	Carpeta de archivos	
ESD	11/4/2023 17:55	Carpeta de archivos	
hadoop-3.3.5	26/4/2023 22:58	Carpeta de archivos	
Intel	26/4/2023 22:45	Carpeta de archivos	
Java	26/4/2023 21:52	Carpeta de archivos	
PerfLogs	7/5/2022 0:24	Carpeta de archivos	
SWSetup	16/4/2023 1:26	Carpeta de archivos	
tmp	26/4/2023 21:38	Carpeta de archivos	
Usuarios	10/4/2023 21:04	Carpeta de archivos	
Windows	16/4/2023 1:21	Carpeta de archivos	
data	27/4/2023 0:23	Text Document	1 KB

```

C:\> Administrador: Símbolo del sistema
***** /

C:\Windows\System32>cd C:\hadoop-3.3.5\sbin

C:\hadoop-3.3.5\sbin>start-all
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.5\sbin>hadooppp fs -mkdir /input
"hadooppp" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\hadoop-3.3.5\sbin>hadoop fs -mkdir /input

C:\hadoop-3.3.5\sbin>hadoop fs -put C:\data.txt /input_

```

5 Spark

Es un sistema de procesamiento de datos distribuido de código abierto que permite procesar grandes cantidades de datos de manera rápida y eficiente. Fue desarrollado por Apache Software Foundation y se basa en el lenguaje de programación Scala [19].

Spark ofrece un conjunto de herramientas para el procesamiento de datos en diferentes formatos, incluyendo CSV, JSON, Parquet y Avro, y es compatible con varios sistemas de almacenamiento de datos, como Hadoop Distributed File System (HDFS), Apache Cassandra, Apache HBase y Amazon S3 [19].

Una de las principales características de Spark es su capacidad para trabajar con datos en memoria, lo que significa que puede procesar grandes volúmenes de datos de manera más rápida que otros sistemas de procesamiento de datos distribuidos que utilizan el disco como almacenamiento principal [20].

Spark también ofrece una amplia gama de APIs y bibliotecas, incluyendo APIs para Python, Java y Scala, y bibliotecas para aprendizaje automático, procesamiento de gráficos y procesamiento de flujo de datos en tiempo real [20].

5.1 Historia de Spark

Fue creado en el año 2009 en la Universidad de California, Berkeley por Matei Zaharia como un proyecto de investigación en el Laboratorio de Análisis de Datos de Berkeley (AMPLab). El objetivo principal del proyecto era crear un motor de procesamiento de datos distribuido que fuera más rápido que el sistema de procesamiento de datos de código abierto existente en ese momento, llamado Apache Hadoop [21].

En 2010, Spark se convirtió en un proyecto de código abierto en Apache Software Foundation, y desde entonces ha sido adoptado y utilizado por muchas empresas y organizaciones en todo el mundo debido a su velocidad, escalabilidad y facilidad de uso. Además, Spark ha evolucionado para incluir módulos como Spark SQL, Spark Streaming, y MLlib, lo que lo hace aún más útil y versátil en una variedad de aplicaciones y casos de uso [20].

5.2 Componentes de Spark

Los componentes centrales son los siguientes:

- **Spark Core:** Es el componente central de Spark que proporciona las funcionalidades básicas de procesamiento distribuido de datos, incluyendo la API para la creación y manipulación de RDDs (Resilient Distributed Datasets), que son la estructura de datos fundamental en Spark [22].
- **Spark SQL:** Es un módulo de Spark que permite trabajar con datos estructurados utilizando el lenguaje SQL. Spark SQL permite ejecutar consultas SQL en RDDs, DataFrames y tablas de bases de datos externas [22].

- **Spark Streaming:** Es un módulo de Spark que permite procesar datos de streaming en tiempo real. Spark Streaming permite la integración con fuentes de datos de streaming como Apache Kafka y Flume [23].
- **MLlib:** Es una biblioteca de aprendizaje automático integrada en Spark que proporciona una variedad de algoritmos de aprendizaje automático, como regresión lineal, regresión logística, clasificación Naive Bayes, clustering K-means y más [23].
- **GraphX:** Es una biblioteca de Spark para el procesamiento de grafos que permite realizar operaciones de análisis de redes sociales, recomendación de productos, y más [23].

5.3 Como funciona Spark

Spark funciona como un motor de procesamiento de datos distribuido que permite procesar grandes conjuntos de datos en clústeres de computadoras. Spark se basa en el modelo de datos RDD (Resilient Distributed Datasets) que permite a Spark procesar datos de manera distribuida de forma tolerante a fallos y en memoria [22].

El funcionamiento de Spark se puede resumir en los siguientes pasos:

- **Spark distribuye los datos:** Spark distribuye los datos en un clúster de computadoras, dividiéndolos en particiones, que se distribuyen en los nodos del clúster [21].
- **Spark procesa los datos:** Spark procesa los datos de forma distribuida mediante la ejecución de operaciones en cada partición de datos en paralelo en diferentes nodos del clúster. Las operaciones se ejecutan en memoria, lo que permite un procesamiento de datos más rápido que en los sistemas de procesamiento de datos tradicionales que acceden a los datos desde el disco [21].
- **Spark almacena los resultados:** Los resultados del procesamiento se almacenan en memoria y en disco, según sea necesario [19].
- **Spark optimiza el procesamiento:** Spark utiliza técnicas de optimización como la partición de datos, el particionamiento de operaciones y la gestión de memoria para maximizar el rendimiento del procesamiento de datos [19].
- **Spark gestiona la tolerancia a fallos:** Spark es tolerante a fallos y puede recuperarse automáticamente de fallos en los nodos del clúster [20].

5.4 Ventajas de Spark

Spark tiene varias ventajas que lo convierten en una de las herramientas más populares para el procesamiento de datos distribuidos, entre ellas:

- **Rendimiento superior:** Spark utiliza una arquitectura de procesamiento en memoria que lo hace mucho más rápido que los sistemas de procesamiento de datos tradicionales, que acceden a los datos desde el disco. Además, Spark tiene la capacidad de procesar grandes conjuntos de datos en paralelo, lo que permite un procesamiento más rápido y escalable [24].

- **Flexibilidad:** Spark es compatible con una variedad de lenguajes de programación, como Java, Scala, Python y R, lo que lo hace accesible a una amplia comunidad de desarrolladores [24].
- **Tolerancia a fallos:** Spark es tolerante a fallos y puede recuperarse automáticamente de fallos en los nodos del clúster, lo que permite una mayor disponibilidad de datos y un menor tiempo de inactividad [22].
- **Integración con otras tecnologías de big data:** Spark se integra bien con otras tecnologías de big data, como Hadoop, Hive, Kafka y más, lo que permite a las organizaciones construir soluciones de big data completas y escalables [23].
- **Bibliotecas de machine learning y análisis de datos:** Spark proporciona una variedad de bibliotecas para el aprendizaje automático y el análisis de datos, lo que permite a las organizaciones realizar análisis complejos de datos y construir modelos de aprendizaje automático [23].
- **Comunidad activa:** Spark cuenta con una comunidad activa de desarrolladores y usuarios que contribuyen al desarrollo de la plataforma y brindan soporte a los nuevos usuarios [25].

5.5 Desventajas de Spark

Aunque Spark es una herramienta muy popular y tiene varias ventajas, también hay algunas desventajas que se deben tener en cuenta, entre ellas:

- **Requiere una curva de aprendizaje:** Spark es una herramienta compleja y requiere una curva de aprendizaje para entender su funcionamiento y utilizarlo de manera efectiva. Además, la necesidad de programar en Scala, Java, Python o R también puede ser un obstáculo para los nuevos usuarios [26].
- **Consumo de memoria:** Aunque Spark utiliza una arquitectura de procesamiento en memoria, esto también significa que puede consumir mucha memoria, lo que puede ser un problema en clústeres de computadoras con recursos limitados [26].
- **Dificultades de depuración:** El procesamiento distribuido en Spark puede dificultar la depuración de errores en el código, lo que puede llevar más tiempo que la depuración en sistemas de procesamiento de datos tradicionales [27].
- **Configuración del clúster:** La configuración y el mantenimiento de un clúster de Spark pueden ser complicados y requerir habilidades de administración de sistemas, lo que puede aumentar el costo y la complejidad de implementar y mantener una solución de big data basada en Spark [27].
- **Limitaciones en operaciones complejas:** Aunque Spark es capaz de procesar grandes conjuntos de datos, algunas operaciones complejas, como aquellas que requieren una gran cantidad de transferencia de datos entre nodos, pueden ser menos eficientes en Spark que en otros sistemas de procesamiento de datos distribuidos [22].

5.6 Como Realizar la instalación de Spark en Windows

Descargamos e instalamos primero el JDK8

Mac OS X x64	249.15 MB	jdk-8u202-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.09 MB	jdk-8u202-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.1 MB	jdk-8u202-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	124.37 MB	jdk-8u202-solaris-x64.tar.Z
Solaris x64	85.38 MB	jdk-8u202-solaris-x64.tar.gz
Windows x86	201.64 MB	jdk-8u202-windows-i586.exe
Windows x64	211.58 MB	jdk-8u202-windows-x64.exe

Luego nos dirigimos a la página oficial de Apache Spark

Download Apache Spark™

- Choose a Spark release: **3.4.0 (Apr 13 2023)**
- Choose a package type: **Pre-built for Apache Hadoop 3.3 and later**
- Download Spark: [spark-3.4.0-bin-hadoop3.tgz](#)
- Verify this release using the 3.4.0 [signatures](#), [checksums](#) and [project release KEYS](#) by following these [procedures](#).

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

Link with Spark

Spark artifacts are [hosted in Maven Central](#). You can add a Maven dependency with the following coordinates:

```
groupId: org.apache.spark
artifactId: spark-core_2.12
version: 3.4.0
```

Installing with PyPi

PySpark is now available in pypi. To install just run `pip install pyspark`.

Convenience Docker Container Images

Spark Docker Container images are available from [DockerHub](#), these images contain non-ASF software and may be subject to

Latest News

- Spark 3.4.0 released (Apr 13, 2023)
- Spark 3.2.4 released (Apr 13, 2023)
- Spark 3.3.2 released (Feb 17, 2023)
- Spark 3.2.3 released (Nov 28, 2022)

[Archive](#)

APACHE EVENTS [LEARN MORE](#)

DOWNLOAD SPARK

Built-in Libraries:

- SQL and DataFrames
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)
- Third-Party Projects

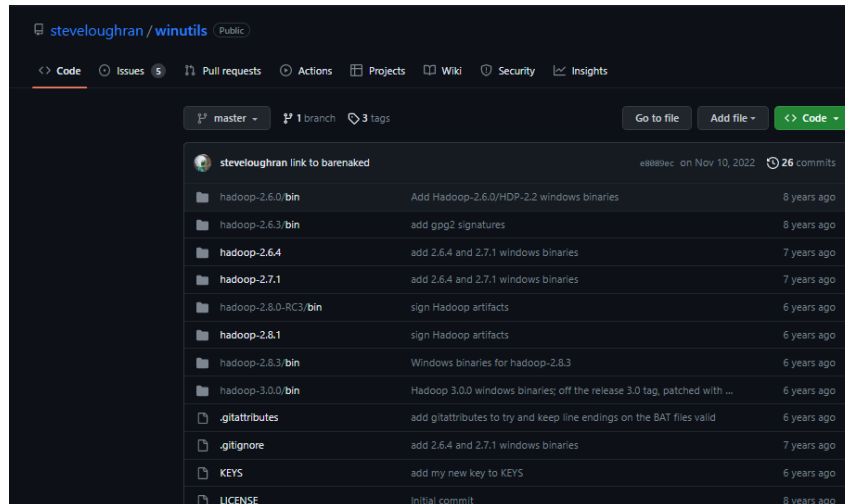
Descargamos Spark

Download Apache Spark™

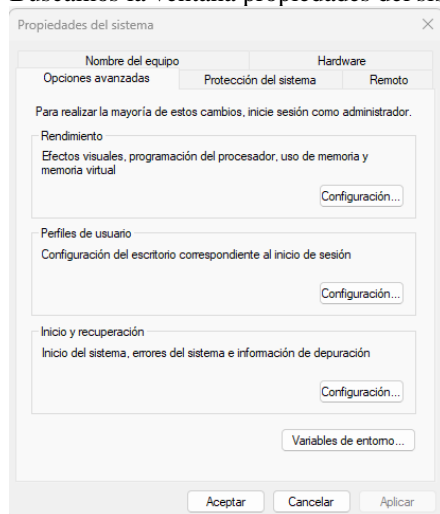
- Choose a Spark release: **3.4.0 (Apr 13 2023)**
- Choose a package type: **Pre-built for Apache Hadoop 3.3 and later**
- Download Spark: [spark-3.4.0-bin-hadoop3.tgz](#)
- Verify this release using the 3.4.0 [signatures](#), [checksums](#) and [project release KEYS](#) by following these [procedures](#).

Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.

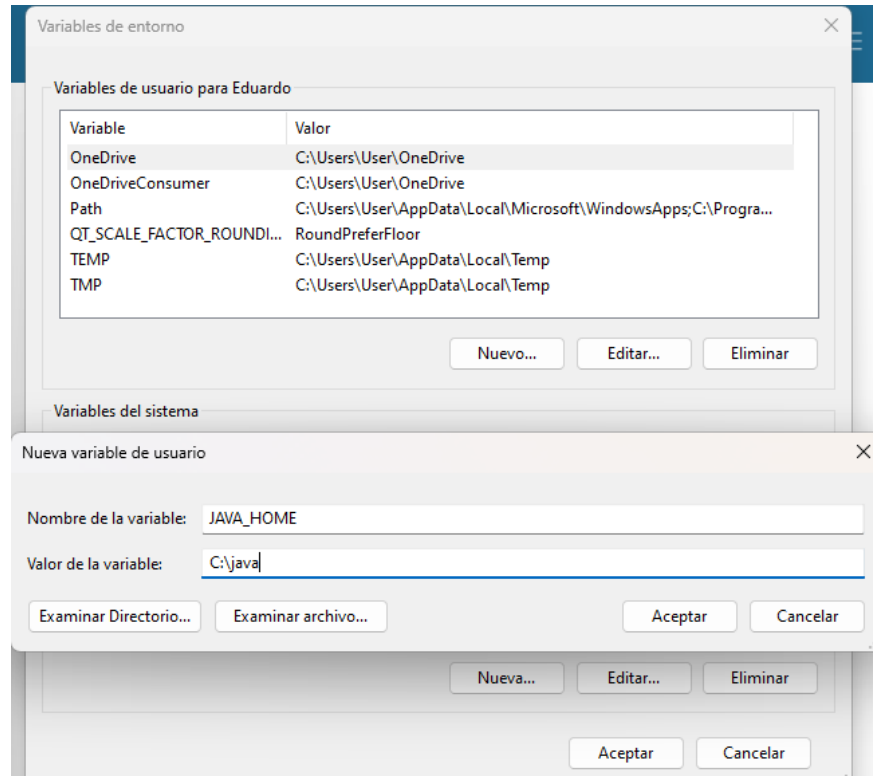
Por último descargamos el repositorio winutils



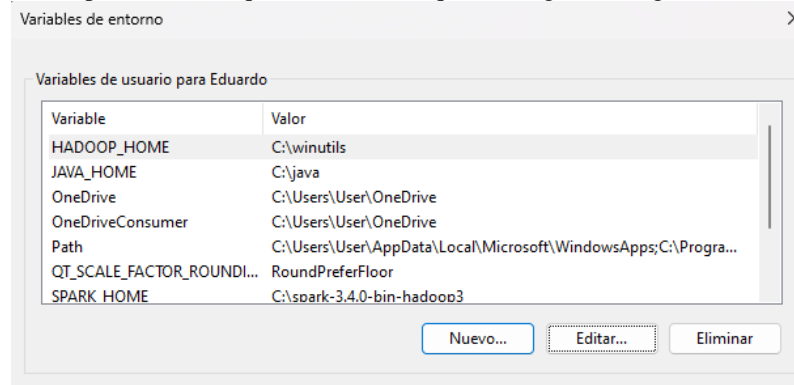
Buscamos la ventana propiedades del sistema



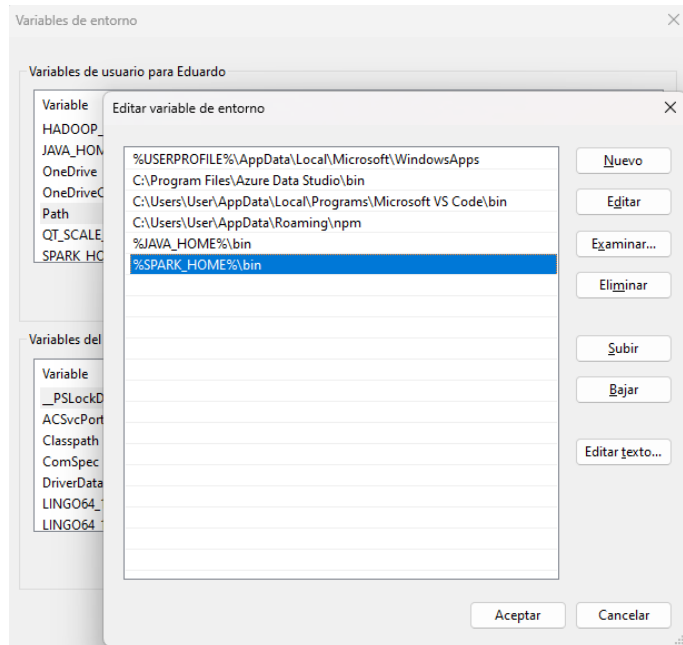
Creamos una variable de entorno con la dirección del java



Y lo repetimos hasta que las tres cosas que descargamos tengan su variable de entorno



Editamos el Path y agregamos esas dos variables de entorno
 %JAVA_HOME%\bin %SPARK_HOME%\bin



Verificamos que tengamos la versión de java

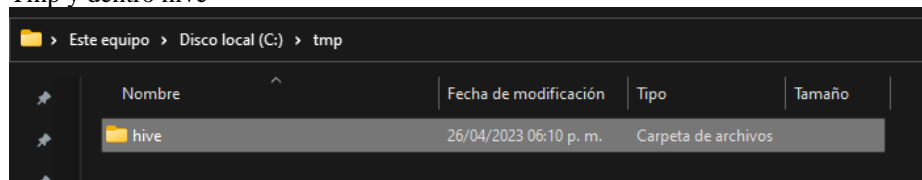
```

C:\Windows\System32>java -version
java version "19.0.2" 2023-01-17
Java(TM) SE Runtime Environment (build 19.0.2+7-44)
Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)

C:\Windows\System32>D_

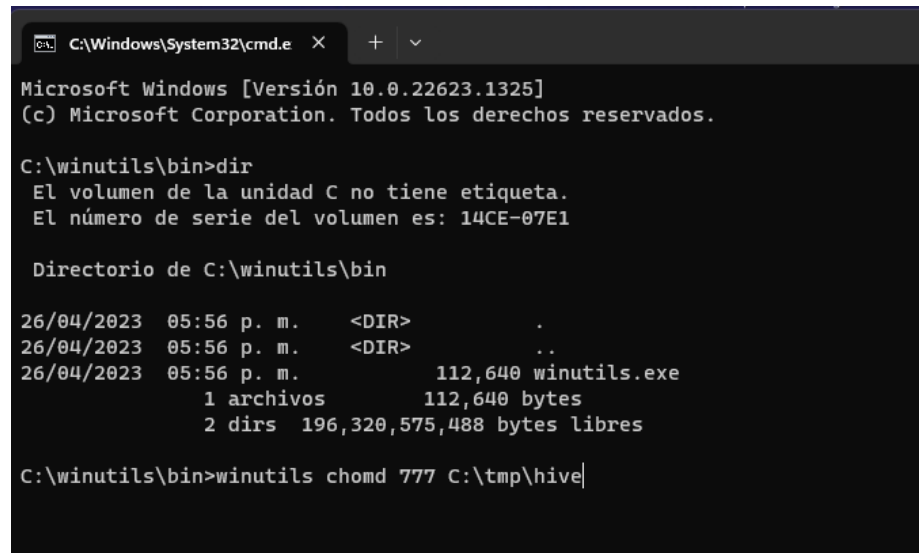
```

Creamos dos carpetas en el disco local C
Tmp y dentro hive



Para terminar de instalar en un CMD nos ubicamos en la dirección del winutils y procedemos a poner el siguiente comando

winutils chomd 777 C:\tmp\hive



```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Versión 10.0.22623.1325]
(c) Microsoft Corporation. Todos los derechos reservados.

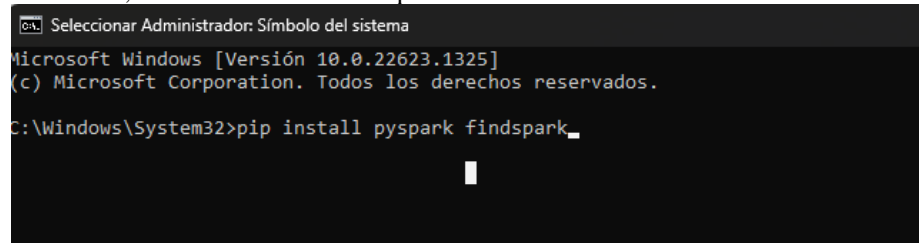
C:\winutils\bin>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 14CE-07E1

Directorio de C:\winutils\bin

26/04/2023  05:56 p. m.  <DIR>          .
26/04/2023  05:56 p. m.  <DIR>          ..
26/04/2023  05:56 p. m.                112,640 winutils.exe
                1 archivos            112,640 bytes
                2 dirs  196,320,575,488 bytes libres

C:\winutils\bin>winutils chomd 777 C:\tmp\hive|
```

Por último, instalamos las librerías que vamos a utilizar



```
Selecciónar Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.22623.1325]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>pip install pyspark findspark_
█
```

```

Administrador: Símbolo del sistema - pip install pyspark findspark
Microsoft Windows [Versión 10.0.22623.1325]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>pip install pyspark findspark
Collecting pyspark
  Downloading pyspark-3.4.0.tar.gz (310.8 MB)
    ----- 310.8/310.8 MB 3.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting findspark
  Downloading findspark-2.0.1-py2.py3-none-any.whl (4.4 kB)
Collecting py4j==0.10.9.7
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
    ----- 200.5/200.5 kB 12.7 MB/s eta 0:00:00
Installing collected packages: py4j, findspark, pyspark
  DEPRECATION: pyspark is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproj
  ect.toml' and the 'wheel' package is not installed. pip 23.1 will enforce this behaviour change. A possible replacement
  is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip/issues/8559
  Running setup.py install for pyspark ... \_

```

Abrimos un jupyter notebook para hacer una prueba

```

C:\Windows\System32>
C:\Windows\System32>
C:\Windows\System32>jupyter notebook_

```

Dentro del jupyter realizamos una prueba para verificar si está bien instalado creamos una nueva SparkSession por medio de los siguientes comandos.

```
conda install -c conda-forge findspark
```

```
import findspark
findspark.init()
```

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.getOrCreate()
```

y para mirar si fue creada escribimos: `spark`

```

In [5]: import findspark
        findspark.init()

In [6]: from pyspark.sql import SparkSession

In [7]: spark = SparkSession.builder.getOrCreate()

In [8]: spark
Out[8]: SparkSession - in-memory
SparkContext

Spark UI
Version
v3.4.0
Master
local[*]
AppName
pyspark-shell

```

Ejemplo de Spark.

- Cómo cargar y transformar datos utilizando PySpark en un entorno de Spark configurado con findspark

➤ Importar la librería findspark e inicializarla

```
import findspark
findspark.init()
```

```
In [1]: import findspark
        findspark.init()
```

➤ Importar la clase SparkSession de PySpark

```
from pyspark.sql import SparkSession
```

```
In [2]: from pyspark.sql import SparkSession
```

➤ Crear una instancia de SparkSession

```
spark = SparkSession.builder \
    .appName("MiApp") \
    .getOrCreate()
```

```
In [3]: spark = SparkSession.builder \
        .appName("MiApp") \
        .getOrCreate()
```

- Cargar un archivo CSV como un DataFrame

```
df = spark.read.csv("datos.csv", header=True, inferSchema=True)
```

	A	B	C	
1	nombre	edad	ciudad	
2	Juan	25	Madrid	
3	Maria	35	Barcelona	
4	Pedro	42	Madrid	
5	Ana	18	Valencia	
6	Luis	30	Barcelona	
7				

```
In [15]: df = spark.read.csv("datos.csv", header=True, inferSchema=True)
```

- Mostrar el esquema del DataFrame

```
df.printSchema()
```

```
In [16]: df.printSchema()
```

```
root
 |-- nombre: string (nullable = true)
 |-- edad: integer (nullable = true)
 |-- ciudad: string (nullable = true)
```

- Realizar una operación de transformación en el DataFrame

```
df2 = df.filter(df["edad"] >= 18).groupBy("ciudad").count()
```

```
In [17]: df2 = df.filter(df["edad"] >= 18).groupBy("ciudad").count()
```

- Mostrar los resultados en la consola

```
df2.show()
```

```
In [18]: df2.show()
```

```
+-----+-----+
| ciudad|count|
+-----+-----+
| Madrid|    2|
|Barcelona|    2|
| Valencia|    1|
+-----+-----+
```


6 Conclusiones

En un mundo cada vez más centrado en la tecnología, las empresas necesitan soluciones innovadoras para manejar grandes cantidades de datos de manera efectiva. Kubernetes, Hadoop y Spark son tecnologías que pueden ayudar en este sentido, ofreciendo herramientas para el procesamiento, almacenamiento y gestión de grandes volúmenes de datos.

7 Referencias

1. Sayfan, G. *Mastering Kubernetes : Automating Container Deployment and Management*; ISBN 9781786461001.
2. Turnbull, J. *The Docker Book*; 2014;
3. Renzo, A. *Containerization with Ansible 2 Implement Container Management, Deployment, and Orchestration within the Ansible Ecosystem*;
4. Clingan, J.; Finnigan, K. *Kubernetes Native Microservices with Quarkus and MicroProfile*;
5. Azure Microsoft ¿Qué Es Kubernetes? | Microsoft Azure Available online: <https://azure.microsoft.com/es-es/topic/what-is-kubernetes/#overview> (accessed on 8 August 2022).
6. Kubernetes Available online: <https://kubernetes.io/es/> (accessed on 8 August 2022).
7. Truyen, E.; Kratzke, N.; Landuyt, D. van; Lagaisse, B.; Joosen, W. Managing Feature Compatibility in Kubernetes: Vendor Comparison and Analysis., doi:10.1109/ACCESS.2020.3045768.
8. Arundel, J.; Domingus, J. *Praise for Cloud Native DevOps with Kubernetes*;
9. Poulton, N. *The Kubernetes Book*; 2017;
10. Lukša, M. *Kubernetes in Action*;
11. Brendan Burns, J.B.& K.H. *Kubernetes_book*.
12. Owens, J.R.; EBSCO Publishing (Firm) *Hadoop Real World Solutions Cookbook*; Packt Publishing, 2013; ISBN 9781849519120.
13. Turkington, G.; Modena, G. *Learning Hadoop 2 : Design and Implement Data Processing, Lifecycle Management, and Analytic Workflows with the Cutting-Edge Toolbox of Hadoop 2*; ISBN 9781783285518.
14. Perera, Srinath.; Gunarathne, Thilina. *Hadoop MapReduce Cookbook : Recipes for Analyzing Large and Complex Datasets with Hadoop MapReduce*; Packt Pub, 2013; ISBN 9781849517287.
15. Grover, M.; Malaska, T.; Seidman, J.; Shapira, G. *Hadoop Application Architectures*;
16. Tom, W.; Hadoop, W. *PROGR AMMING LANGUAGES/HADOOP Hadoop: The Definitive Guide " Nowyouhavethe Hadoop: The Definitive Guide FOURTH EDITION The Definitive Guide STORAGE AND ANALYSIS AT INTERNET SCALE*; ISBN 978-1-491-90163-2.
17. Holmes, A. *Hadoop in Practice*;

18. Boris-Lublinsky_-Kevin-T.-Smith_-Alexey-Yakubovich-Professional-Hadoop-Solutions-Wrox-_2013_.
19. Aven, J. *Sams Teach Yourself Apache Spark in 24 Hours*; ISBN 9780672338519.
20. Ryza, S.; Laserson, U.; Owen, S.; Wills, J. *Advanced Analytics with Spark*;
21. Kienzler, Romeo. *Mastering Apache Spark 2.x - Second Edition.*; Packt Publishing, 2017; ISBN 9781786462749.
22. Chambers, B. (William A.; Zaharia, M. *Spark: The Definitive Guide: Big Data Processing Made Simple*; ISBN 9781491912218.
23. Yadav, R. *Spark Cookbook: Over 60 Recipes on Spark, Covering Spark Core, Spark SQL, Spark Streaming, MLib, and GraphX Libraries*; ISBN 9781783987061.
24. Introducción_a_Apache_Spark.
25. *The Data Scientist's Guide To*;
26. www.it-ebooks.info *High Performance Spark*;
27. Karau, H. *Learning Spark: Lightening Fast Data Analysis*; ISBN 9781449358624.