

SNOWPARK: NUEVAS CAPACIDADES DE PROGRAMACIÓN DE DATOS EN EL DATA CLOUD



ÍNDICE

- 2** Resumen ejecutivo
- 3** Creación de flujos y modelos de datos optimizados con Snowpark
- 5** Casos de uso de Snowpark para la ingeniería de datos
- 8** Casos de uso de Snowpark para data science
- 11** Casos de uso de Snowpark para la seguridad y la gobernanza de datos
- 12** Casos de uso de otros partners de Snowflake
- 13** Comience su experiencia con Snowpark
- 14** Acerca de Snowflake

RESUMEN EJECUTIVO

Snowflake comenzó su experiencia en el Data Cloud rediseñando por completo el entorno de los datos y replanteando cómo debe ser la planificación de un sistema de procesamiento de datos fiable, seguro, de alto rendimiento y escalable pensado para la nube.

El resultado es Snowpark, un nuevo marco de desarrollo para Snowflake. Snowpark permite a ingenieros, científicos y desarrolladores de datos codificar de la forma que están acostumbrados a hacerlo y con el lenguaje de su elección, con el fin de ejecutar aplicaciones de datos, flujos de trabajo de aprendizaje automático (machine learning; ML) y flujos de forma más rápida y segura en una misma plataforma.

La API de Snowpark aporta una programación de estilo DataFrame profundamente integrada a los lenguajes preferidos por los desarrolladores, incluidos Scala y Java. Las funciones definidas por el usuario (user-defined functions, UDF) de Snowpark ayudan a ampliar fácilmente los casos de uso de datos y se ejecutan dentro de Snowflake. Estas UDF incluyen las UDF de Java y JavaScript, las funciones externas y, próximamente, también incluirán las UDF de Python. Snowpark se ha diseñado para facilitar la creación de flujos de datos complejos y permitir a los desarrolladores interactuar directamente con Snowflake sin necesidad de mover los datos.

Estos son algunos de los casos de uso posibles con Snowpark:

- uso de ML para aumentar los datos alojando modelos entrenados en Java;
- análisis de los datos para identificar anomalías;
- desarrollo de rutinas para identificar información de identificación personal (personally identifiable information; PII);
- implementación de bibliotecas Java compartidas con el fin de estandarizar la lógica empresarial.

Snowpark representa un gran avance en la programación de datos y permite obtener mucho más de la plataforma de Snowflake con facilidad. Desde su lanzamiento, Snowpark ha acumulado casos de uso convincentes que incluyen a muchos de nuestros partners del programa Snowpark Accelerated.

¿Qué puede crear con Snowpark? En este eBook se describen varios casos de uso a modo de ejemplo, pero puede crear muchos más. Comencemos.

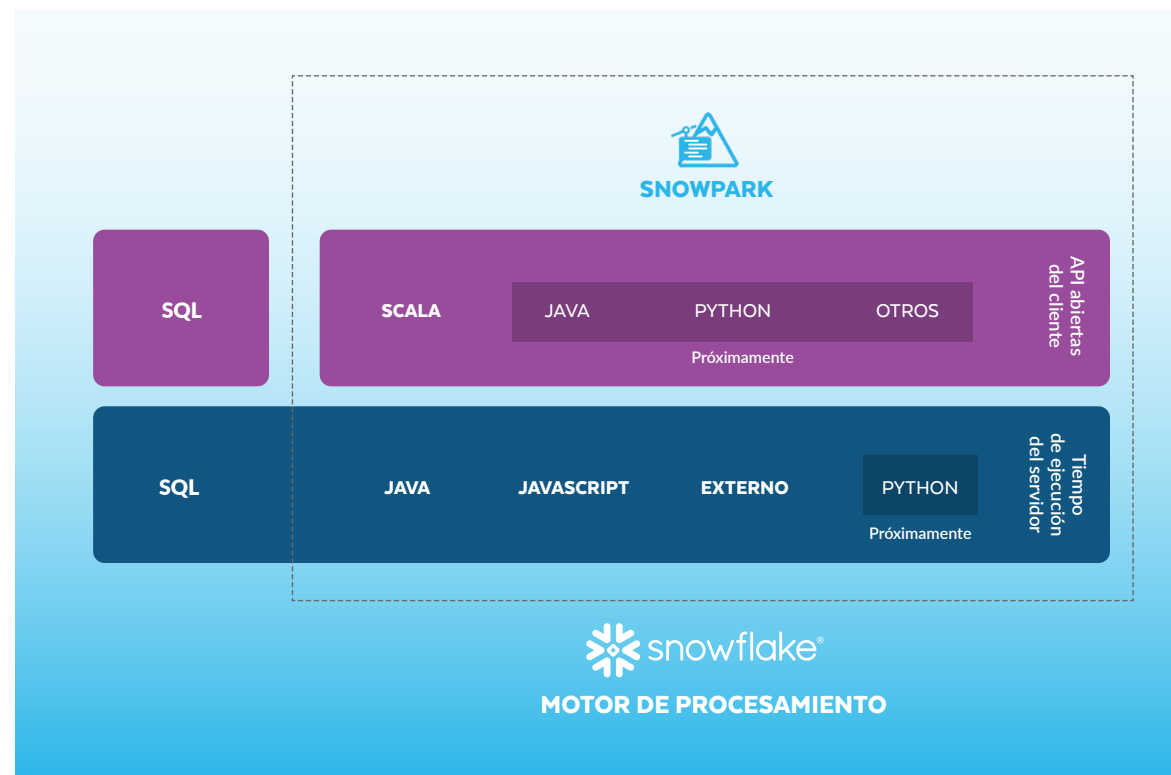


Figura 1: Snowpark permite a desarrolladores e ingenieros interactuar directamente con Snowflake sin necesidad de mover datos.

CREACIÓN DE FLUJOS Y MODELOS DE DATOS OPTIMIZADOS CON SNOWPARK

Para entender cómo funciona Snowpark, echemos un vistazo a un ejemplo genérico con PII.

CONSULTAS OPTIMIZADAS, CONVERSIONES FLUIDAS

Con la API de Snowpark, los desarrolladores pueden crear consultas utilizando DataFrames en su código sin necesidad de crear ni transferir cadenas SQL, por ejemplo:

```
val sess = // conectarse a Snowflake

val sales:DataFrame = sess.table("sales")
val line_items:DataFrame = sess.table("sales_details")

val query = sales.join(line_items, sales("id") ===
line_items("sid"))
    .groupBy(line_items("product_id"))
    .count()
```

Dado que la API de Snowpark utiliza estructuras lingüísticas de primera clase, también proporciona soporte de primera clase en el entorno de desarrollo. Esto incluye la comprobación de tipos, IntelliSense e informes de errores. Además, Snowpark convierte sin problemas estas operaciones en un lenguaje SQL que se ejecuta dentro del motor escalable de alto rendimiento de Snowflake.

Pero la API de Snowpark es mucho más que un método sencillo de escribir consultas. También puede utilizar su propia lógica personalizada. A continuación, se muestra un ejemplo que utiliza código personalizado para enmascarar la PII:

```
val maskPii = (s:String) => {
    // Lógica de detección de PII personalizada.
}
```

Con la API de Snowpark, puede clasificar fácilmente este código como una **UDF**. Después, puede utilizarlo en las operaciones de DataFrame:

```
val maskPiiUdf = udf(maskPii)
sess.table("emails")
    .withColumn("body", maskPiiUdf(col("body")))
    .show()
```

La API de Snowpark traslada su lógica a Snowflake, por lo que se ejecuta junto a sus datos. El código se aloja en una máquina virtual de Java (Java Virtual Machine, JVM) de pruebas, segura y protegida dentro de los almacenes de Snowflake.

LA AUTOMATIZACIÓN ELIMINA LA CODIFICACIÓN MANUAL

Para saber cómo Snowpark simplifica las operaciones habituales, veamos cómo permite aplicar la lógica de detección de PII a todas las columnas de cadena de una tabla. Con SQL, tendría que codificar manualmente una consulta para cada tabla o escribir código para generar la consulta. Sin embargo, con Snowpark puede escribir fácilmente una rutina genérica:

```
val maskTable = (df:DataFrame) => {
    df.select(df.schema.map(field =>
        if (field.dataType == StringType) maskPiiUdf(col(field.name))
        else col(field.name))
    )
}
```

A continuación, puede utilizar esa rutina genérica para enmascarar fácilmente toda la PII de cualquier tabla:

```
val maskedEmails = maskTable(sess.table("emails"))
```

Solo necesita estas pocas líneas de código para que Snowpark genere una consulta sólida y basada en esquemas de forma dinámica.

CREACIÓN DE LÓGICAS COMPLEJAS CON UDF DE JAVA

El uso de SQL sigue siendo habitual en Snowflake y los usuarios de SQL pueden beneficiarse plenamente de las nuevas capacidades de la plataforma a través de las UDF de Java.

Con las UDF de Java, puede crear lógicas complejas que expongan una interfaz de función simple:

```
public class Sentiment
{
    public float score(String text)
    {
        // Su lógica de análisis de sentimiento.
    }
}
```

Para crear estas funciones, puede utilizar el conjunto de herramientas habitual, incluido el control de código fuente, los entornos de desarrollo y las herramientas de depuración, así como cualquier biblioteca que necesite. Snowpark ofrece la posibilidad de tomar código útil de GitHub u otras fuentes y utilizarlo en Snowflake.

El proceso para trasladar el código a SQL es sencillo. Cree un JAR (o varios), cárguelo en Snowflake y registre una función:

```
create function sentiment(txt string) returns float
language java
imports = ('@jars/Sentiment.jar')
handler = 'Sentiment.score';
```

Ahora, cualquier usuario de SQL puede utilizar la lógica que ha creado como cualquier otra función:

```
select id, sentiment(body)
from emails;
```

Las UDF de Java le permiten utilizar las herramientas con las que ya cuenta para casos complejos. Sin embargo, a veces el caso de uso es más básico, por lo que esta nueva funcionalidad también incluye definiciones simples e insertadas:

```
create or replace function reverse(s string) returns string
language java
handler = 'Reverse.reverse'
target_path = '@jars/Reverse.jar'
as
$$
public class Reverse
{
    public String reverse(String s)
    {
        return new StringBuilder(s).reverse().toString();
    }
}
$$;
```

CASOS DE USO DE SNOWPARK PARA LA INGENIERÍA DE DATOS

La ingeniería de datos es compleja por dos razones principales. En primer lugar, los ingenieros de datos a menudo necesitan utilizar varios sistemas y soluciones, lo que hace que la arquitectura del flujo de datos sea rígida y demasiado compleja. Una de las principales razones de esto es que la ingeniería de datos es un esfuerzo de colaboración entre equipos. Los analistas de datos podrían preferir herramientas basadas en SQL y GUI. Por el contrario, los científicos de datos suelen preferir cuadernos y Python para preparar los datos, mientras que los desarrolladores e ingenieros de datos necesitan herramientas adicionales para abordar complejas estructuras de código y programación. A menudo, los datos deben recorrer estos diferentes sistemas para que los flujos funcionen, lo que genera arquitecturas complejas que pueden poner en peligro la seguridad y afectar a la gobernanza de los datos.

En segundo lugar, gestionar y operar infraestructuras de procesamiento de datos suele conllevar un esfuerzo manual considerable y una sobrecarga de mantenimiento. Como resultado, es habitual que los ingenieros de datos estén muy dispersos y dediquen la mayor parte de su tiempo al mantenimiento y reparación de flujos.

Snowpark se ha diseñado para resolver ambos problemas. Lea los siguientes casos de uso de nuestros partners para obtener más información.

CASO DE USO: ENTREGA DE DATOS FIABLES CON TALEND

Su capacidad de análisis es tan buena como la calidad de sus datos. ¿Cómo se mide? Ahora, puede realizar una comprobación del estado de los datos dentro de Snowflake utilizando Talend Trust Score™, una funcionalidad esencial de la solución Talend Data Inventory. Trust Score™ le ayuda a identificar y diagnosticar problemas importantes relacionados con los datos, además de impulsar los esfuerzos de todas las partes interesadas clave por fomentar la confianza en la toma de decisiones.

Normalmente, para perfilar una gran cantidad de conjuntos de datos mediante una aplicación externa, se extrae una muestra de los datos



con el fin de ejecutar el análisis en un sistema distinto al lugar en el que se almacenan. Trasladar los datos fuera de Snowflake plantea ciertos desafíos: riesgos para la seguridad y privacidad de los datos, cargos por entrada y salida de datos, procesamientos fallidos debido a la imposibilidad de escalar los recursos y resultados imprecisos al estar basados en una muestra de los datos.

Los cálculos de Trust Score™ para los conjuntos de datos de Snowflake alivian muchas de estas dificultades, pues el procesamiento para calcular la puntuación se produce de forma nativa dentro de Snowflake utilizando UDF de Java. Es un procedimiento muy preciso, ya que los resultados se basan en todo el conjunto de datos, no solo en una muestra. Las UDF de Java se pueden invocar a través

de [Snowpark](#) o [SQL](#). Talend utilizó ambos enfoques: uno durante la fase de creación de prototipos y el otro al implementar la función en el producto Talend Data Inventory. Comience hoy mismo su viaje hacia una cultura de datos óptimos probando Talend en [Snowflake Partner Connect](#).

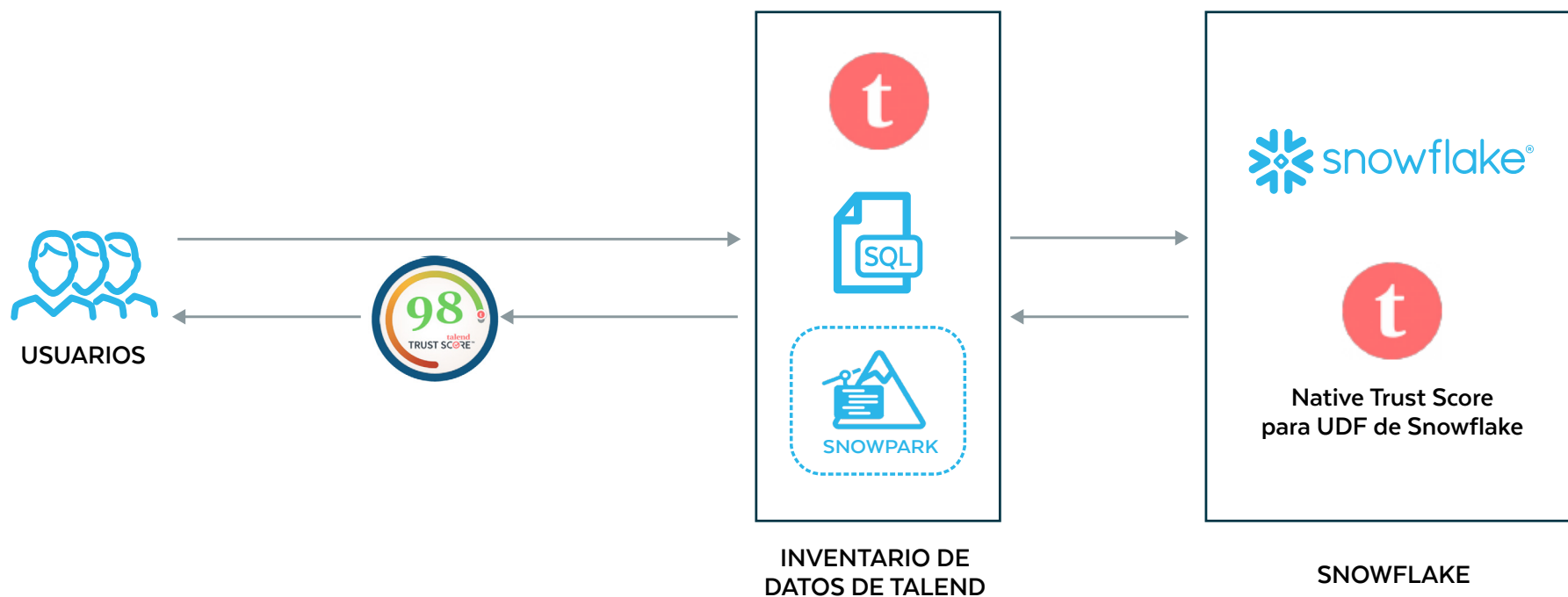


Figura 2: Entrega de datos fiables con Snowflake y Talend.¹

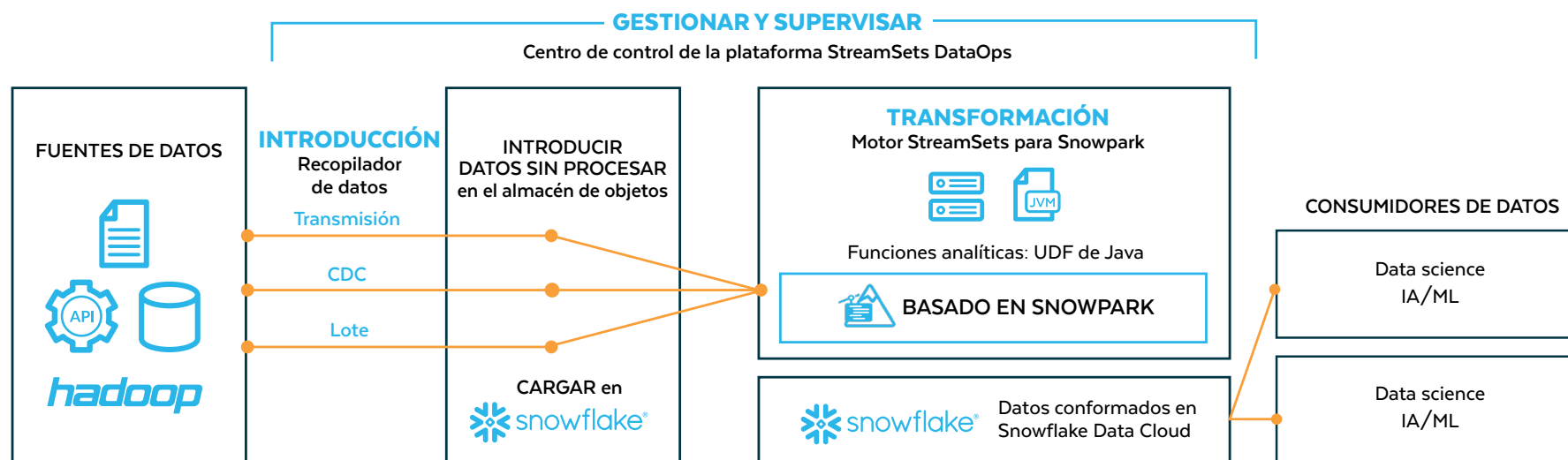


Figura 3: Optimización de DataOps con el motor de StreamSets para Snowpark.³

CASO DE USO: OPTIMIZACIÓN DE DATAOPS CON EL MOTOR STREAMSETS PARA SNOWPARK

El motor StreamSets para Snowpark se basa en Snowpark para así aprovechar la expresividad y flexibilidad de la compatibilidad con varios lenguajes que ofrece Snowpark, además de por la simplicidad de las operaciones del Data Cloud. Los ingenieros de datos pueden ir más allá de SQL para expresar una potente lógica de flujo de datos con StreamSets DataOps Platform.

Usando Scala o Java con una interfaz gráfica intuitiva, tiene la opción de no usar código en absoluto o introducirlo cuando lo desee. El motor StreamSets para Snowpark incluye las ventajas de StreamSets DataOps Platform, como la supervisión y coordinación integradas de complejos flujos de datos a escala, todo en la nube y sin necesidad de hardware adicional.

CASO DE USO: GESTIÓN DEL CICLO DE VIDA DEL DESARROLLO CON DATAOPS.LIVE

Una amplia colaboración entre DataOps.live y Snowflake permite gestionar completamente los workloads de Snowpark junto con el resto de objetos de Snowflake durante todo el ciclo de vida de desarrollo, pruebas y producción. Además de toda la «fuente de información fiable de los datos» que se almacena en el repositorio Git de DataOps.live, el repositorio también proporciona todos los requisitos habituales de desarrollo de software. Con DataOps.live, puede ramificar, crear versiones, compilar, probar e implementar software y producir artefactos igual que si se tratase de cualquier otro proyecto de software.

Ejecutar la API de Snowpark significa ejecutar una aplicación de Snowpark en la que se utilizan las bibliotecas de Snowpark. Por ejemplo, si se

utiliza Scala, se necesita un entorno con todas las herramientas de tiempo de ejecución para ese lenguaje específico y las bibliotecas apropiadas (además de las propias bibliotecas de Snowpark).

En muchos casos, se utilizará una aplicación de Snowpark para realizar una manipulación avanzada de los datos, en particular más allá de lo que se puede lograr con SQL, pero los resultados se seguirán almacenando en Snowflake. En estos casos, la prueba de datos automatizada en el motor de modelado y transformación de DataOps.live se puede utilizar para validar los resultados de la aplicación de Snowpark.

CASOS DE USO DE SNOWPARK PARA DATA SCIENCE

Los avances en programación de datos de Snowpark proporcionan una mayor flexibilidad y extensibilidad, lo que permite a los científicos de datos utilizar sus lenguajes de programación favoritos, como Python, para acceder a los datos, visualizarlos y procesarlos como parte de sus flujos de trabajo de ML. Esto ayuda a las organizaciones a maximizar el valor de los datos, incluidos los no estructurados y los de terceros.

Los partners de data science de Snowflake han tomado la iniciativa aprovechando estos avances para acelerar los flujos de trabajo de ML con una experiencia más integrada. Snowpark permite a los científicos de datos trabajar con el cuaderno que

preferan, al tiempo que trasladan el procesamiento al lugar donde residen los datos. Estos son algunos de los casos de uso de Snowpark para data science:

- Ingeniería de funciones
- Inferencia de modelos de ML
- ML integral en Snowflake con SQL

CASO DE USO: PREPARACIÓN DE UN DATAFRAME CON SNOWPARK PARA INVOCAR LA INFERENCIA Y EL ENTRENAMIENTO DE IA

Snowpark hace que los datos de Snowflake estén disponibles como DataFrame utilizando código Java y Scala que se ejecuta en el entorno de Snowflake.

Snowpark se ha diseñado para facilitar la creación de flujos de datos complejos y capacitar a los desarrolladores, ingenieros y científicos de datos que utilizan código como parte de la programación basada en cuadernos utilizando Jupyter o Dataiku, H2O.ai y Zepl (ahora parte de la plataforma de DataRobot). Con Snowpark, pueden utilizar su lenguaje preferido para acelerar los esfuerzos de ingeniería de funciones aplicando conceptos de programación conocidos como DataFrames para después ejecutar estos workloads directamente en Snowflake. Por ejemplo, los científicos de datos pueden definir nuevas funciones en cuadernos de H2O.ai y ejecutar el procesamiento en Snowflake para beneficiarse de su escalabilidad y rendimiento.

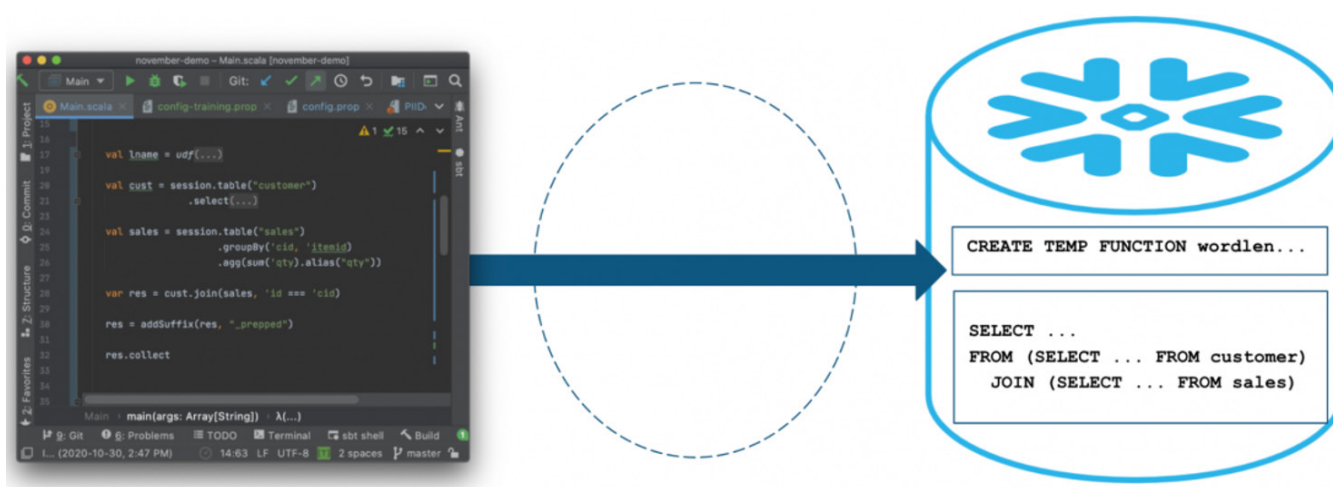


Figura 4: Preparación de un DataFrame con Snowpark para invocar el entrenamiento de Driverless AI en un cuaderno de H2O.ai.⁷

CASO DE USO: AMPLIACIÓN DE LAS FUNCIONES DE PREPARACIÓN DE DATOS Y PUNTUACIÓN DE MODELOS PREDICTIVOS CON DATAIKU

Antes de Snowpark, varias funciones de preparación de datos y puntuación de modelos predictivos se ejecutaban en Dataiku y otros motores al no poder expresarse en SQL. Esto requería mover los datos dentro y fuera de Snowflake, lo que afectaba al rendimiento. Ahora, los clientes pueden aprovechar al máximo las ventajas de rendimiento y gobernanza de Snowflake al poder ejecutar completamente sus flujos de Dataiku en Snowflake.

Los usuarios sin código y con poco código que deseen crear flujos de preparación de datos en Dataiku se benefician del motor de Snowflake con UDF de Java. Las UDF de Java permiten que los workloads expresados en Java se ejecuten en una JVM dentro de Snowflake.

Dado que Java es el motor principal de Dataiku, es conveniente que Dataiku vuelva a empaquetar las funciones de preparación de datos y traslade su ejecución a Snowflake. Esto permite escalar sin esfuerzo con el fin de gestionar cualquier número de usuarios, trabajos o datos, a la vez que se simplifica la arquitectura.

CASO DE USO: SIMPLIFICACIÓN DEL CAMINO HACIA LA PRODUCCIÓN CON LA INFERENCIA DE MODELOS EN SNOWFLAKE

Para ejecutar la inferencia de modelos a escala, los equipos de operaciones suelen necesitar pasar los datos de producción de una fuente de información fiable gobernada a un entorno diferente para implementar los modelos. El traslado de grandes volúmenes de datos tiene consecuencias, tanto para los costes como para la seguridad, que, con frecuencia, impiden que los modelos pasen a producción.

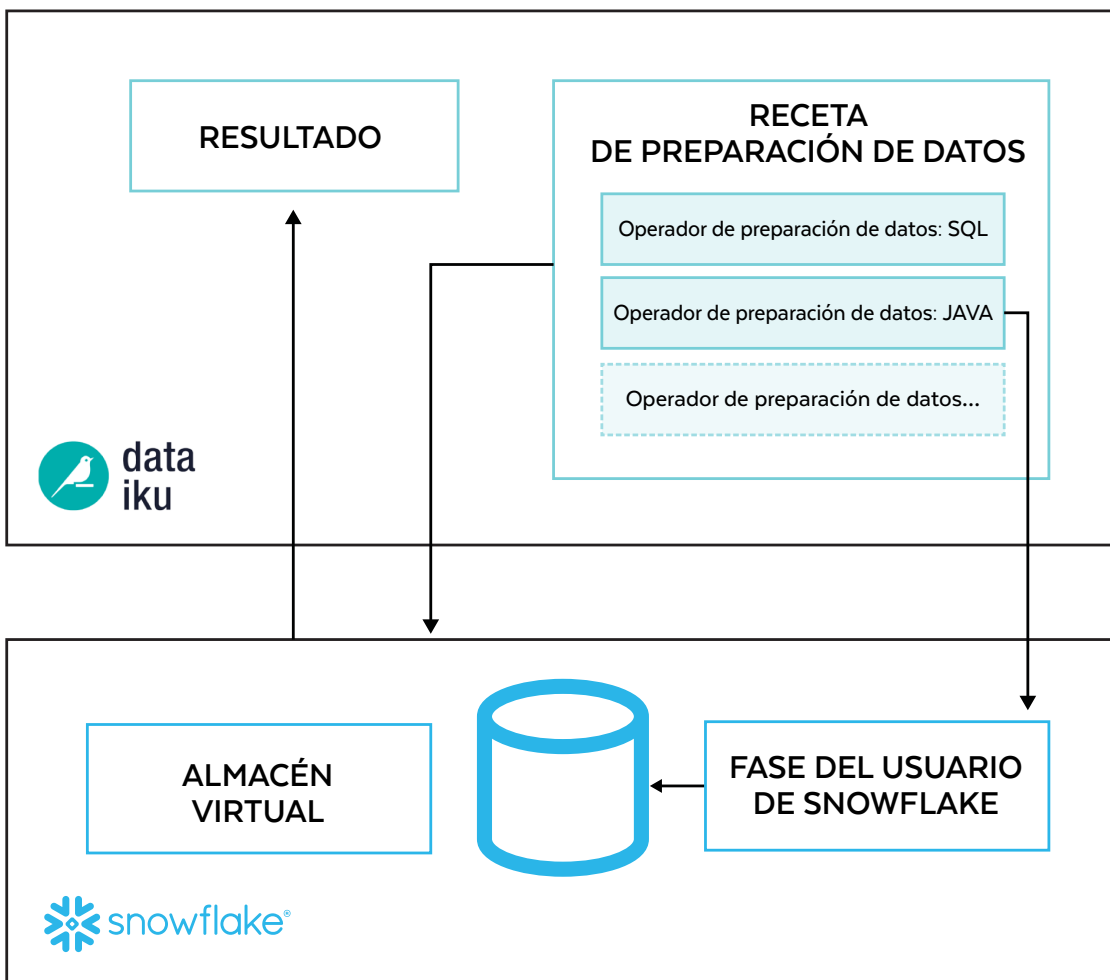


Figura 5: Ampliación de las funciones de preparación de datos y puntuación de modelos predictivos con Dataiku.⁸

Para abordar la complejidad que supone el traslado de los datos desde el lugar donde se encuentran, los equipos de desarrollo y operaciones de ML pueden adoptar un enfoque distinto en el que el modelo llegue hasta los datos utilizando UDF de Java. Las UDF de Java pueden utilizar modelos entrenados, expresados en Java, para ejecutar inferencias dentro de Snowflake.

Los partners de data science de Snowflake, DataRobot y H2O.ai, ya cuentan con la funcionalidad de exportar modelos e implementarlos en Snowflake para la inferencia de modelos ampliable basada en lotes y de invocar la función externa para una inferencia en tiempo real.

La plataforma DataRobot AI Cloud te ofrece una gestión y un desarrollo continuos del ciclo de vida de la IA para el Data Cloud. Con DataRobot puede conectar y aprovechar sin problemas los datos de Snowflake, transformar y preparar esos datos para la IA, realizar la detección automatizada de funciones dentro de Snowflake, crear modelos de IA y exportarlos como archivos JAR para producción e implementarlos en Snowflake para una inferencia de modelos escalable.

DataRobot AI Cloud dispone de opciones de implementación flexibles, incluida la producción de código de puntuación. Con DataRobot, todos los modelos, en los que Snowflake tiene una biblioteca de soporte, ahora se pueden implementar directamente dentro de una UDF de Java en Snowflake. Esta opción de implementación proporciona una velocidad de puntuación optimizada con respecto a los métodos de puntuación de modelos tradicionales que anteriormente se ejecutaban fuera de Snowflake.

Además, los usuarios de DataRobot pueden introducir datos de servicio y predicción en DataRobot AI Cloud para analizar la desviación del modelo a lo largo del tiempo. DataRobot también le permite supervisar cualquier modelo que no sea de DataRobot y que se haya implementado en Snowflake para así mantener una única fuente de información fiable de IA en todas sus iniciativas de IA.

En un caso de uso en particular, H2O.ai combinó su conjunto de datos para predecir los impagos de préstamos con un conjunto de datos disponible públicamente de LendingClub y un conjunto de datos demográficos de Snowflake Data Marketplace. Eso mejoró la precisión del modelo y redujo la puntuación (tiempo de inferencia), ya que podía escalarse dentro del entorno de Snowflake. Una vez que H2O.ai entrenó su modelo basándose en sus datos y los datos de terceros de Snowflake Data Marketplace, pudo importar el modelo al lugar donde se alojan los datos para una implementación más sencilla y una inferencia más eficaz, ya que el modelo se traslada al lugar donde están los datos.

Por último, al ejecutar un modelo en Snowflake, puede seguir utilizando la plataforma de ML que prefiera para supervisar los modelos. Por ejemplo, los usuarios de DataRobot pueden introducir datos de servicio y predicción en DataRobot MLOps para analizar la desviación del modelo a lo largo del tiempo.

Además, Dataiku utiliza la misma tecnología para empaquetar e implementar modelos de ML en Snowflake, lo que permite producir una puntuación predictiva sin mover los datos fuera de Snowflake.

Portable Predictions

Portable predictions allow DataRobot models to be deployed on a variety of external infrastructures. These models can report monitoring statistics back to this deployment using the Monitoring Agent.

1. Download Scoring Code

Portable predictions allow DataRobot to be deployed directly into Snowflake as User-Defined Functions.

To get started, first download the model's Scoring Code and follow the instructions to configure the model in Snowflake.

[Download Scoring Code \(.jar\)](#)

2. Configure and execute installation script

This script uploads the JAR to a Snowflake stage and creates a User-Defined Function for making predictions using Scoring Code. Update the script template with the desired parameters for the Snowflake warehouse, database, schema and JAR file location: [Copy to clipboard](#)

```

1 -- Replace with the warehouse to use
2 USE WAREHOUSE my_warehouse;
3 -- Replace with the database to use
4 USE DATABASE my_database;
5
6 -- Replace with the schema you want to use
7 CREATE SCHEMA IF NOT EXISTS scoring_code_udf_schema;
8 USE SCHEMA scoring_code_udf_schema;
9
10 -- Update this path to match the Scoring Code jar's location
11 PUT 'file:///path/to/6081ee9727a78f2a2247cafa-6076150513693237b74e6997.jar' '@~/jars/' AUTO_COMPRESS=FALSE;
12
13 -- Create the UDF
14 CREATE OR REPLACE FUNCTION datarobot_deployment_6081ee9727a78f2a2247cafa(RowValue Object)
15 RETURNS OBJECT
16 LANGUAGE JAVA
17 IMPORTS=(' @~/jars/6081ee9727a78f2a2247cafa-6076150513693237b74e6997.jar')
18 HANDLER='com.datarobot.prediction.simple.ClassificationPredictor.score';

```

Figura 6: Script generado por DataRobot para cargar un JAR y crear una UDF asociada con el fin de realizar la inferencia directamente en Snowflake.⁹

CASOS DE USO DE SNOWPARK PARA LA SEGURIDAD Y LA GOBERNANZA DE DATOS

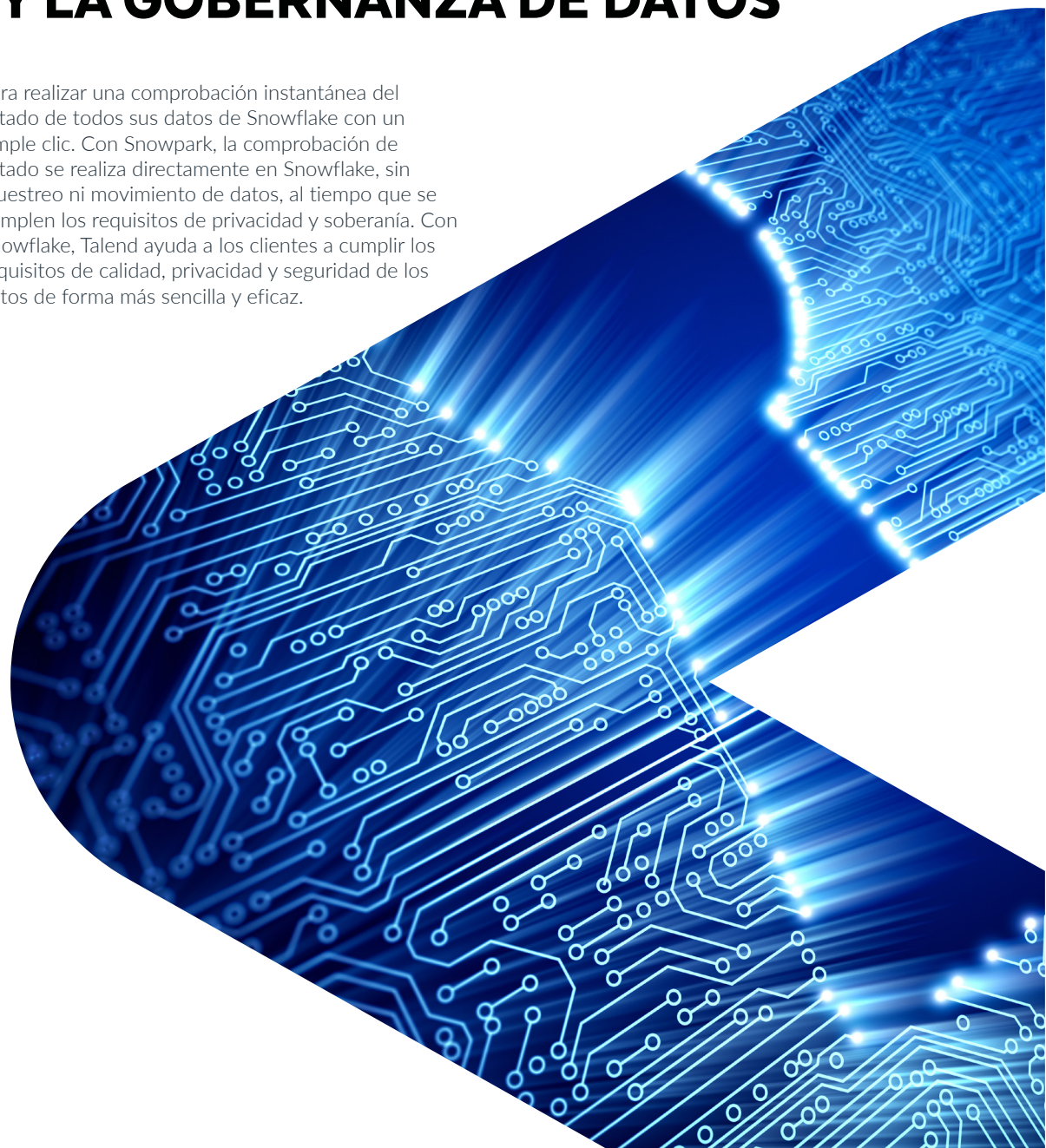
Snowpark se puede utilizar para crear soluciones que respalden los esfuerzos de gobernanza de datos. A continuación, se muestran algunos ejemplos de lo que se puede crear:

- **Perfiladores de calidad de los datos:** Snowpark se puede utilizar para ayudar a las organizaciones a conocer y mejorar la calidad de sus datos. Una función puede analizar la integridad de los datos y detectar anomalías para señalar posibles problemas de calidad.
- **Clasificadores de datos:** aunque la capacidad de clasificación de Snowflake identifica automáticamente PII, Snowpark puede utilizarse para crear la lógica que permita clasificar otros tipos de datos que sean esenciales para una organización.
- **Funciones de enmascaramiento que conservan el formato:** Snowpark se puede utilizar para cubrir necesidades más personalizadas para la función de enmascaramiento de datos dinámico de Snowflake. Por ejemplo, puede crear funciones personalizadas para enmascarar o reemplazar números de tarjetas de crédito.

CASO DE USO: ENTREGA DE DATOS FIABLES CON TALEND

Algunos de los casos de uso de la ingeniería de datos mencionados anteriormente están muy cerca de la gobernanza de datos. Con Snowpark, los partners pueden ejecutar su software directamente en Snowflake, lo que simplifica la postura en materia de gobernanza de los clientes al reducir la necesidad de mover los datos. Por ejemplo, los clientes de Snowflake y Talend ahora pueden utilizar Talend Trust Assessor

para realizar una comprobación instantánea del estado de todos sus datos de Snowflake con un simple clic. Con Snowpark, la comprobación de estado se realiza directamente en Snowflake, sin muestreo ni movimiento de datos, al tiempo que se cumplen los requisitos de privacidad y soberanía. Con Snowflake, Talend ayuda a los clientes a cumplir los requisitos de calidad, privacidad y seguridad de los datos de forma más sencilla y eficaz.



CASOS DE USO DE OTROS PARTNERS DE SNOWFLAKE

LTI

Gracias a Snowpark, LTI Mosaic⁵ puede simplificar y ampliar las siguientes capacidades de Snowflake:

- **DataOps (la aplicación de los principios de DevOps a los datos)**
- **MLOps (operaciones de ML)**
- **ModelOps (un método integral para mover modelos de forma rápida e iterativa a lo largo del ciclo de vida de los análisis)**

Gracias a Snowpark, los clientes de Mosaic pueden implementar modelos de ML, insertar inferencias de modelos y escribir código que se puede pasar a Snowflake para su procesamiento.

phData

Los clientes de phData quieren realizar tareas propias de ML, como el procesamiento del lenguaje natural (PLN) o el reconocimiento de imágenes, con sus datos de Snowflake. Tradicionalmente, este ha sido un proceso complejo que no podía hacerse solo con SQL. Con Snowpark, los clientes de phData ahora pueden gestionar este complejo flujo dentro de Snowflake, lo que reduce significativamente los costes y la complejidad.⁶



COMIENZE SU EXPERIENCIA CON SNOWPARK

Para empezar a utilizar Snowpark, consulte nuestra [documentación](#) y la [guía de inicio rápido paso a paso](#). Esperamos deseando ver todo lo que será capaz de crear.

Colaboradores:

Carlos Bouloy

Frank Pacione

Julian Forero

Miles Adkins

Paul Gantz

Shiyi Gu





ACERCA DE SNOWFLAKE

Snowflake permite a cualquier organización movilizar sus datos con Snowflake Data Cloud. Los clientes utilizan Data Cloud para unificar, descubrir y compartir datos de forma segura, y ejecutar diversos workloads analíticos. Independientemente de la ubicación de los datos o de los usuarios, Snowflake ofrece una experiencia de datos única que abarca varias nubes y regiones geográficas. Miles de clientes de numerosos sectores, incluidas 543 de las empresas que figuran en Forbes Global 2000 (G2K) (2022), a fecha del 31 de octubre de 2022, utilizan Snowflake Data Cloud para impulsar sus negocios.

Más información en [Snowflake.com](https://www.snowflake.com).



©2021 Snowflake Inc. Todos los derechos reservados. Snowflake, el logotipo de Snowflake y el resto de nombres de productos, funciones y servicios de Snowflake mencionados en este documento son marcas registradas o marcas comerciales de Snowflake Inc. en Estados Unidos y otros países. El resto de logotipos o nombres de marcas mencionados o utilizados en este documento se usan únicamente con fines identificativos y pueden ser las marcas comerciales de sus respectivos titulares. Snowflake puede no estar asociado con, patrocinado o apoyado por cualquiera de dichos titulares.

CITAS

¹ bit.ly/2XAxZGU

² bit.ly/3nZCICA

³ bit.ly/3tWMAbw

⁴ bit.ly/3AvdrOV

⁵ bit.ly/3nO8VY1

⁶ bit.ly/3zyQpEZ

⁷ bit.ly/3tLY1mh

⁸ bit.ly/39bbue0

⁹ bit.ly/3tNTKZh

¹⁰ bit.ly/3zcZzqH

¹¹ bit.ly/3tLY1mh

¹² bit.ly/3kgmugA