



## Estándares de nomenclatura para Databricks

### CONFIDENCIALIDAD

Este documento es CONFIDENCIAL. Se prohíbe su reproducción, distribución o publicación total o parcial, incluyendo su contenido: diseño gráfico, funcionalidades o recomendaciones cuya propiedad intelectual pertenece a HandytecMobi S.A. El Cliente se compromete a mantener la información confidencial en estricta reserva y no revelar ningún dato de la información a ninguna otra parte, relacionada o no, sin el consentimiento previo escrito de HandytecMobi S.A.

<b>Cliente</b>	Banco Solidario S. A.
<b>Proyecto</b>	Estrategia de datos y plataforma de analítica
<b>Autor</b>	Irina Ávalos
<b>Versión</b>	1.0
<b>Fecha</b>	2022-04-11
<b>Estado</b>	Final

Documentos relacionados:

Documento	Formato	Ubicación
No aplica	No aplica	No aplica

Control de cambios:

Fecha	Autor	Cambios
2022-04-11	Irina Ávalos	Creación de versión inicial del documento



## Contenido

1. Introducción.....	4
2. Workspace.....	4
2.1. Nombrado de directorios .....	4
2.2. Nombrado de notebooks.....	4
2.3. Nombrado de objetos dentro de un notebook.....	5
3. Delta Lake.....	6
3.1. Nombrado de base de datos .....	6
3.2. Bases de datos por defecto .....	6
3.3. Nombrado de Tablas.....	7
3.4. Nombrado de columnas .....	7
3.5. Ejemplo de aplicación de estándares en creación de tabla .....	8
4. Mejores prácticas de desarrollo de notebooks .....	8



## 1. Introducción

El siguiente documento tiene como objetivo presentar los estándares sugeridos de nomenclatura para Databricks.

## 2. Workspace

### 2.1. Nombrado de directorios

El nombre debe cumplir las siguientes características:

- Aplicar estilo snake\_case (separación con guiones bajos y texto en minúsculas).
- No exceder los 32 caracteres.
- No usar caracteres especiales.
- No usar acentos como tildes, diéresis u otros similares. Reemplazar acentos por su equivalente sin acento. Ejemplo: línea con linea, año con anio, ñandu con nandu.
- Cuando se requiera almacenar directorios ordenados se recomienda usar dígitos al principio del nombre de los archivos para establecer una secuencia de pasos o etapas.
- La estructura recomendada de directorios es la siguiente:

scala (nombre del proyecto)

- 01\_ingestion (contiene notebooks sobre la implementación del proceso de ingesta de datos)
- 02\_engineering (contiene notebooks sobre procesos de ingeniería de datos organizados por dominio)
  - domain\_1
  - domain\_2
- 03\_data\_quality (Contiene notebooks de evaluación sobre la calidad de los datos)
- initial (contiene notebooks para la creación inicial de base de datos / tablas / configuraciones / etc.)
- utils (contiene notebooks utilitarios para las etapas mencionadas en los puntos anteriores. Pueden estar agrupados por funcionalidad)
  - ingestion
  - quality
  - engineering

### 2.2. Nombrado de notebooks

El nombre de los notebooks debe cumplir las siguientes características:

- Aplicar estilo snake\_case (separación con guiones bajos y texto en minúsculas).
- No exceder los 45 caracteres.



- No usar guiones, puntos ni caracteres especiales.
- No usar acentos como tildes, diéresis u otros similares. Reemplazar acentos por su equivalente sin acento. Ejemplo: línea con linea, año con anio, ñandu con nandu.
- Simplificar los nombres evitando el uso de preposiciones como: de, a, con, en, para, por, sin, y.
- Reemplazar los caracteres especiales por una abreviación de hasta 4 letras que haga referencia a su significado. Ejemplo: %=porc.
- Nombrado en inglés.
- Opcionalmente, se puede utilizar un prefijo de dos dígitos para determinar el orden de ejecución de los notebooks.
- Se recomienda utilizar un sufijo que especifique lenguaje predominante del notebook. Se establece la siguiente tabla de acuerdo al uso.

Lenguaje	Siglas	Ejemplo
python	py	datalake_functions_py
sql	sql	01_ingest_bronze_tables_sql 02_process_business_logic_sql 03_create_gold_tables_sql
scala	sca	load_tables_sca
R	r	clustering_model_r

### 2.3. Nombrado de objetos dentro de un notebook

Se debe seguir la Guía de estilos para código de Python ([pep-008](#)).

Objeto	Nombrado	Ejemplo
Variables	Utilizar nombres de variables significativos y pronunciables (Guía de Clean Code)	current_date
Constantes	Utilizar nombres de variables significativos y pronunciables (Guía de Clean Code)	PATH_LOG
Funciones	Usar el mismo vocabulario para el mismo tipo de variable (Guía de Clean Code)	get_user_info() get_user_data() get_user_record()
Funciones UDF	Usar prefijo udf_	udf_convert_to_usd()



### 3. Delta Lake

#### 3.1. Nombrado de base de datos

El nombre debe cumplir las siguientes características:

- Aplicar estilo snake\_case (separación con guiones bajos y texto en minúsculas).
- Delta Lake no admite mayúsculas en nombrado.
- No exceder los 32 caracteres.
- No usar caracteres especiales.
- No usar acentos como tildes, diéresis u otros similares. Reemplazar acentos por su equivalente sin acento. Ejemplo: línea con linea, año con anio, ñandu con nandu.
- Los nombres usados deben ser descriptivos al uso, no son válidos nombres como: demo, prueba, test, nombres propios (juan\_perez) y cualquier combinación como demo1\_juan ; en el caso de pruebas de concepto use el prefijo poc; Ejemplo: poc\_<cliente o aplicación>
- De preferencia y por optimización de costos, se debe especificar la ubicación de guardado de los datos en un Datalake. Siga el siguiente esquema de organización: /mnt/aci\_deltalake/<nombre de base de datos>.db

#### 3.2. Bases de datos por defecto

- db\_parameters: contiene las tablas creadas para las parametrizaciones de ingesta, calidad entre otros.
- db\_bronze: contiene las tablas crudas ingestadas desde las fuentes de datos (zona de staging).
- db\_silver: contiene las tablas depuradas, limpiadas y procesadas.
- db\_gold: contiene tablas sumarizadas con lógica del negocio.
- db\_analytics (opcional): contiene las tablas que serán consumidas por los modelos de analítica.
- db\_tmp (opcional): contiene tablas temporales. Se debe evitar crear tablas dentro de la base de datos default, ya que esto conlleva a un costo adicional.

Ejemplo:

```
CREATE DATABASE db_parameters location
```

```
 '/mnt/aci_deltalake/db_parameters.db';
```

```
CREATE DATABASE db_bronze location '/mnt/aci_deltalake/db_bronze.db';
```

```
CREATE DATABASE db_silver location '/mnt/aci_deltalake/db_silver.db';
```

```
CREATE DATABASE db_gold location '/mnt/aci_deltalake/db_gold.db';
```



```
CREATE DATABASE db_analytics location
```

```
'/mnt/aci_deltalake/db_analytics.db';
```

```
CREATE DATABASE db_tmp location '/mnt/aci_deltalake/db_tmp.db';
```

### 3.3. Nombrado de Tablas

El nombre debe cumplir las siguientes características:

- Aplicar estilo snake\_case (separación con guiones bajos y texto en minúsculas). Delta Lake no admite mayúsculas en nombrado.
- No exceder los 45 caracteres.
- No usar caracteres especiales.
- No usar acentos como tildes, diéresis u otros similares. Reemplazar acentos por su equivalente sin acento. Ejemplo: línea con linea, año con anio, ñandu con nandu.
- Los nombres usados deben ser descriptivos al uso, no son válidos nombres como: demo, prueba, test, nombres propios (juan\_perez) y cualquier combinación como demo1\_juan.
- Excepción: El nombrado de tablas de esta sección no aplica para las tablas de la base de datos db\_bronze. Esto debido a que el nivel bronze extrae las fuentes crudas.

Ejemplo:

- db\_silver.cliente
- db\_silver.pago
- db\_gold.ventas\_anuales
- db\_gold.ventas\_mensuales

### 3.4. Nombrado de columnas

El nombre debe cumplir las siguientes características:

- Usar estándar lowerCamelCase (primera letra con minúscula, separación con mayúscula).
- No exceder los 45 caracteres.
- No usar guiones, puntos ni caracteres especiales.
- No usar acentos como tildes, diéresis u otros similares. Reemplazar acentos por su equivalente sin acento. Ejemplo: línea con linea, año con anio, ñandu con nandu.
- Simplificar los nombres evitando el uso de preposiciones como: de, a, con, en, para, por, sin, y.
- Reemplazar los caracteres especiales por una abreviación de hasta 4 letras que haga referencia a su significado. Ejemplo: %=porc.



- Excepción: El nombrado de columnas de esta sección no aplica para las tablas de la base de datos db\_bronze o db\_silver. Esto debido a que el nivel bronce extrae las fuentes crudas y se mantiene la estructura hasta db\_silver.

Ejemplo de nombres de columnas:

Nombre de campo	Nombre de columna
Número de cargas familiares	numeroCargasFamiliares
Año de consumo	anioConsumo
% de endeudamiento	porcEndeudamiento

### 3.5. Ejemplo de aplicación de estándares en creación de tabla

**CREATE TABLE** db\_silver.pago (

pagold int,  
mtcn double,  
paisPago string,  
paisEnvio string,  
tipo string,  
montoUsd decimal,  
fecha timestamp,  
operador int,  
dbOrigen string,  
anio int,  
mes int

**USING DELTA PARTITIONED BY** (dbOrigen, anio, mes)

## 4. Mejores prácticas de desarrollo de notebooks

- Usar markdown para identificar secciones/subsecciones relevantes, explicar y documentar procesos que lo requieran.
- Cuando se desarrolle clases o funciones utilitarias, se recomienda utilizar Docstring tipo NumPy.

