

Batiburrillo del programador

Cosas que me sirven y que sirven

- [Creación de diagramas DBML](#)
- [Borrar archivos o directorios que comienzan por letra \(A-Z/a-z\)](#)

Creación de diagramas DBML

Introducción

Me gusta tener un cuadro de lo que tengo en Mysql. A los largo de los años, ha dio cambiando el formato, las utilidades. Lejos queda el insoportable Workbench de mysql.

Ahora puedo usar [TablePlus](#), pero a la postre muchas veces eso se me queda corto para lo que me gusta.

Asi que ahor aprefioer mantener la logica guardada en formato DBML, lo cual me permite una visión más adecuada, y real de lo que hay y lo que no hay en la app, y que uso a menudo con

[DBDiagram](#)

db2dbml

[db2dbml](#) es una puntenteherramienta para generar un fichero DBML contectando a un servidor o motores de basos de datos.

Un ejemplo abajo de como generar en local el diagrama de mi proyecto.

```
> dbdocs db2dbml mysql
'mysql://root:MYPASSWORD@localhost:3306/lowino?socketPath=/tmp/mysql.sock' -o
notas/DBML/lowino.dbml
✓ Connecting to database... done.
✓ Generating DBML... done
✓ Wrote to notas/DBML/lowino.dbml
```

Tras eso tenemos el codigo en el fichero

```
Table "advert_properties" {
  "id" "bigint unsigned" [pk, not null, increment]
  "advert_id" binary(26) [not null]
  "property_id" "bigint unsigned" [not null]
  "value" text
  "created_at" timestamp
  "updated_at" timestamp
```

```

}

Table "adverts" {
  "id" binary(26) [unique, not null]
  "user_id" binary(26) [not null]
  "category_code" "mediumint unsigned" [not null, default: 1, note: 'without classification']
  "title" varchar(80) [not null]
  "slug" varchar(80) [unique, not null, note: 'Integrate -YYMMDDNN']
  "advert" text [not null]
  "price" "bigint unsigned"
  "currency_id" "tinyint unsigned" [not null, default: 1]
  "details" json
  "published_at" timestamp
  "expire_at" timestamp
  "created_at" timestamp
  "updated_at" timestamp
  "deleted_at" timestamp

  Indexes {
    published_at [type: btree, name: "adverts_published_at_index"]
  }
}
...

```

Y con ello ya podemos hacwer y deshacer como queramos, para crear nuestro digrama.

“ Cierto que tambien TablePlus es casi más eficaz en términos visuales, pero el dbml sirve para nuchas más cosas.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).

Borrar archivos o directorios que comienzan por letra (A–Z/a–z)

Objetivo

Eliminar **todos los archivos o carpetas cuyo nombre comience por una letra**, sin distinguir mayúsculas/minúsculas.

Ideal para limpiezas de entorno sin afectar archivos ocultos ni aquellos que comiencen por números o símbolos.

?? En macOS (Zsh)

Zsh no usa `shopt`, pero tiene su propia sintaxis de globbing avanzada.

1. Ver qué se va a borrar (modo seguro)

```
setopt extended_glob  
ls -d (#i)[[:alpha:]]*
```

2. Eliminar los archivos y carpetas

```
setopt extended_glob  
rm -Rf (#i)[[:alpha:]]*
```

Explicación técnica

- `setopt extended_glob` activa los patrones avanzados de Zsh.
- `(#i)` activa insensibilidad a mayúsculas para todo el patrón.
- `[[:alpha:]]*` coincide con cualquier nombre que empiece por letra (A–Z o a–z), de forma portable.

“ Ojo: patrones como `(#i)[a-z]*` no siempre funcionan como se espera, porque `[a-z]` sigue limitado al rango ASCII explícito. Usar `[[:alpha:]]` es más robusto.

? En Linux (Bash)

Bash usa `shopt` para habilitar coincidencias insensibles a mayúsculas.

1. Ver qué se va a borrar

```
shopt -s nocaseglob  
ls -d [a-zA-Z]*  
shopt -u nocaseglob
```

2. Eliminar los archivos y carpetas

```
shopt -s nocaseglob  
rm -Rf [a-zA-Z]*  
shopt -u nocaseglob
```

? Explicación técnica

- *nocaseglob* hace que Bash trate los patrones como insensibles a mayúsculas.
- *[a-zA-Z]** coincide con nombres que comienzan por cualquier letra.

⚠ Advertencias No borra archivos ocultos (*.env*, *.git*, etc.) ni nombres que empiecen con números (2023-img.png).

Usa *ls* antes de *rm -Rf* para confirmar qué se eliminará.

Si necesitas excluir ciertos nombres, puedes añadir filtros con *grep -v*, *find*, o usar listas de exclusión.

? Alternativas útiles

Mover en lugar de borrar:

```
mkdir -p backup_letters  
mv [a-zA-Z]* backup_letters/
```

Ver el tamaño de los elementos que serán eliminados:

```
du -sh [a-zA-Z]*
```

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de [Castris](#)

Si necesitas soporte profesional puedes contratar con Castris [soporte profesional](#).