bitwarden | Artículo del Centro de Ayuda

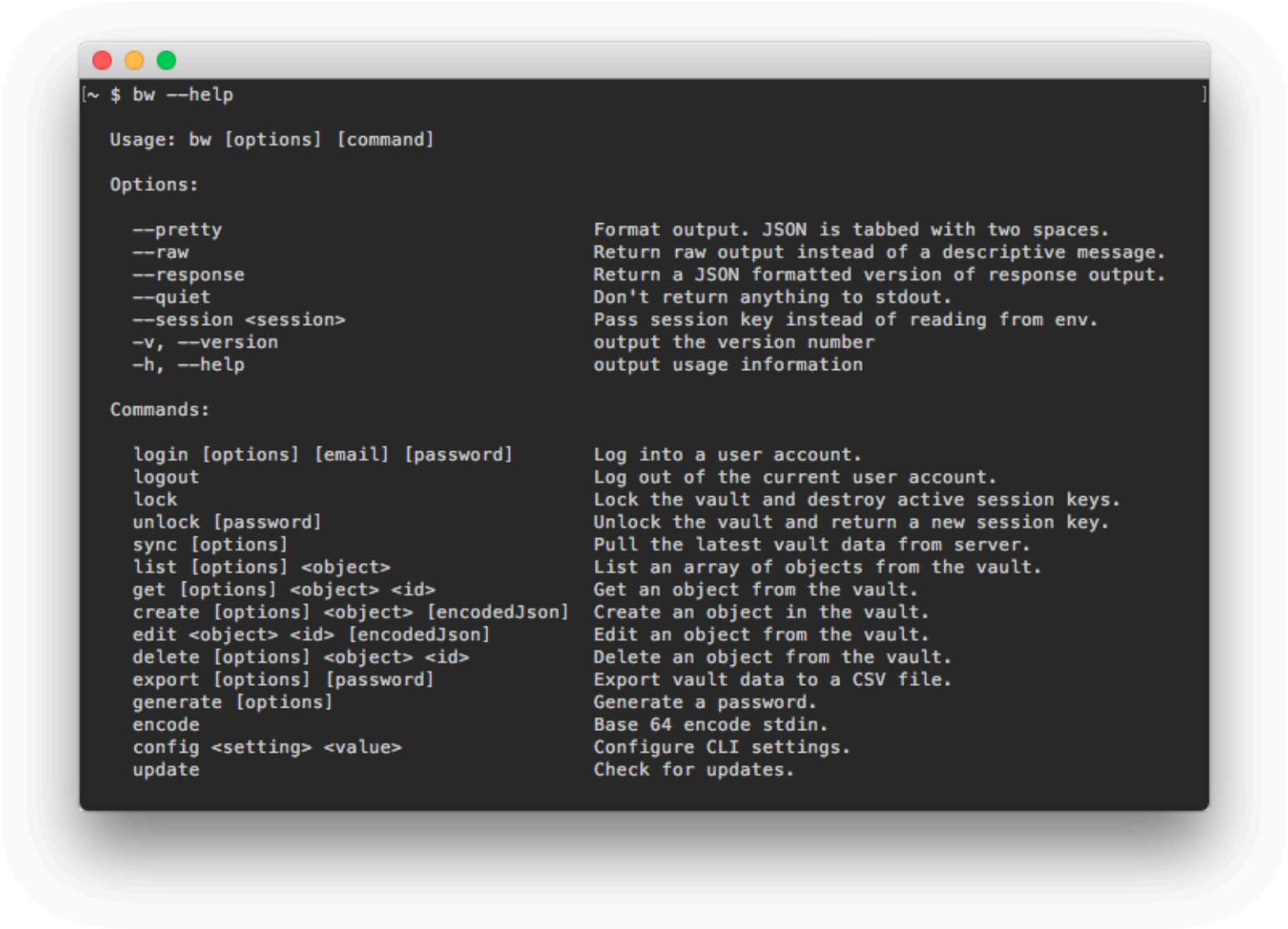
PASSWORD MANAGER > HERRAMIENTAS DE DESARROLLO

Administrador de Contraseñas ILC

Ver en el centro de ayuda:
https://bitwarden.com/help/cli/

Administrador de Contraseñas ILC

La interfaz de línea de comandos de Bitwarden (ILC) es una herramienta poderosa y completamente funcional para acceder y gestionar tu caja fuerte. La mayoría de las funcionalidades que encuentras en otras aplicaciones cliente de Bitwarden (escritorio, extensión de navegador, etc.) están disponibles desde la ILC.



Bitwarden ILC

El Bitwarden ILC está auto-documentado. Desde la línea de comandos, aprenda sobre los comandos disponibles utilizando:

```bash
bw --help
```

O, pasa `--help` como una opción en cualquier comando `bw` para ver las opciones disponibles y ejemplos:

```bash
bw list --help

bw move --help
```

La mayoría de la información que necesitarás se puede acceder utilizando `--help`, sin embargo, este artículo replica toda esa información y profundiza en algunos temas.

Descargar e instalar

El ILC se puede utilizar en varias plataformas en Windows, macOS y distribuciones de Linux. Para descargar e instalar el Bitwarden ILC:

> **Note**
> Para dispositivos arm64, instala el ILC usando `npm`.

⇒Ejecutable Nativo

Versiones empaquetadas de forma nativa de la ILC están disponibles para cada plataforma y no tienen dependencias. Descarga usando uno de estos enlaces:

- Windows x64
- macOS x64
- Linux x64

Ten en cuenta que, al usar el ejecutable nativo descargado, necesitarás agregar el ejecutable a tu PATH o ejecutar comandos desde el directorio donde se descargó el archivo.

> **Tip**
> En los sistemas Linux y UNIX, podrías recibir un mensaje de `Permiso denegado`. Si lo haces, otorga permiso ejecutando:
>
> ```bash
> chmod +x </path/to/executable>
> ```

⇒NPM

Si tienes Node.js instalado en tu sistema, puedes instalar el ILC usando NPM. Instalar con NPM es la forma más sencilla de mantener tu instalación actualizada y debería ser el **método preferido para aquellos que ya están cómodos con NPM**:

```bash
npm install -g @bitwarden/cli
```

Ver el paquete en npmjs.org.

> **Note**
> La instalación de Bitwarden ILC en sistemas Linux utilizando `npm` puede requerir que se instale primero la dependencia build-essential (o el equivalente de la distribución). Por ejemplo:
>
> ```plain text
> apt install build-essential
> ```

⇒Chocolateado

Para instalar con Chocolatey:

```bash
choco install bitwarden-cli
```

Ver el paquete en community.chocolatey.org.

⇒Chasquido

Para instalar con snap:

```bash
sudo snap install bw
```

Ver el paquete en snapcraft.io.

Identificarse

Antes de iniciar sesión, asegúrate de que tu ILC esté conectado al servidor correcto (por ejemplo, nube de la UE o autoalojado) utilizando el comando de configuración (aprende más). Hay tres métodos para iniciar sesión en el ILC de Bitwarden utilizando el comando de `inicio de sesión`, cada uno de los cuales es adecuado para diferentes situaciones. Por favor revise las siguientes opciones para determinar qué método usar:

- Usando correo electrónico y contraseña maestra
- Usando una clave API
- Usando SSO

No importa qué opción uses, siempre asegúrate de usar los comandos `bw bloquear` o `bw cerrar sesión` cuando hayas terminado.

> **Tip**
>
> Iniciar sesión usando correo electrónico y contraseña maestra utiliza tu contraseña maestra y, por lo tanto, puede unir los comandos de `inicio de sesión` y `desbloquear` para autenticar tu identidad y descifrar tu caja fuerte en conjunto. Usar una clave API o SSO requerirá que sigas el comando de `inicio de sesión` con un `bw desbloquear` explícito si vas a trabajar directamente con los datos de la caja fuerte.
>
> Esto es porque tu contraseña maestra es la fuente de la clave necesaria para descifrar los datos de la caja fuerte. Sin embargo, hay algunos comandos que no requieren que su caja fuerte sea descifrada, incluyendo `config`, `encode`, `generar`, `actualizar`, y `status`.

Usando correo electrónico y contraseña

Iniciar sesión con correo electrónico y contraseña es **recomendado para sesiones interactivas**. Para iniciar sesión con correo electrónico y contraseña:

```Bash
bw login
```

Esto iniciará un aviso para su **Dirección de Correo Electrónico**, **Contraseña Maestra**, y (si está habilitado) en **Código de Inicio de Sesión en Dos Pasos**. La ILC actualmente admite el inicio de sesión en dos pasos a través de autenticador, correo electrónico, o YubiKey.

Puede encadenar estos factores en un solo comando como en el siguiente ejemplo; sin embargo, esto no se recomienda por razones de seguridad:

```Bash
bw login [email] [password] --method <method> --code <code>
```

Vea Enums para los valores de inicio de sesión en dos pasos .

> **Tip**
>
> ¿Se le solicita autenticación adicional o recibe un error que dice `Parece que su solicitud de autenticación proviene de un bot.`? Utilice su clave API `client_secret` para responder al desafío de autenticación. Más información.

Usando una clave de API

Se recomienda iniciar sesión con la clave API personal para flujos de trabajo automatizados, para proporcionar acceso a una aplicación externa, o si su cuenta utiliza un método de 2FA no compatible con la ILC (FIDO2 o Duo). Para iniciar sesión con la clave API:

```Bash
bw login --apikey
```

Esto iniciará un aviso para tu personal `client_id` y `client_secret`. Una vez que tu sesión esté autenticada usando estos valores, puedes usar el comando `desbloquear`. Más información.

> **Tip**
> Si tu organización requiere SSO, aún puedes usar `--apikey` para iniciar sesión en la ILC.

Usando variables de entorno de clave API

En escenarios donde se realiza trabajo automatizado con el ILC de Bitwarden, puedes guardar variables de entorno para prevenir la necesidad de intervención manual en la autenticación.

Nombre de la variable de entorno	Valor requerido
BW_CLIENTID	`identificación del cliente`
BW_CLIENTSECRET	`secreto_del_cliente`

Usando SSO

Se recomienda iniciar sesión con SSO si una organización requiere autenticación SSO. Para iniciar sesión con SSO:

```Bash
bw login --sso
```

Esto iniciará el flujo de autenticación SSO en tu navegador web. Una vez que tu sesión esté autenticada, puedes usar el comando `desbloquear`. Más información.

> **Tip**
> Si su organización requiere SSO, puede usar alternativamente `--apikey` para iniciar sesión en la ILC.

Iniciar sesión en varias cuentas

Al igual que el uso de cambio de cuenta en otras aplicaciones de Bitwarden, la ILC tiene la capacidad de iniciar sesión en varias cuentas simultáneamente utilizando la variable de entorno `BITWARDENCLI_APPDATA_DIR` que apunta a la ubicación de un archivo de configuración `bw`, generalmente llamado `data.json`. Por ejemplo, puedes establecer alias en un perfil `.bashrc` para dos ajustes separados:

```bash
alias bw-personal="BITWARDENCLI_APPDATA_DIR=~/.config/Bitwarden\ CLI\ Personal /path/to/bw $@"
alias bw-work="BITWARDENCLI_APPDATA_DIR=~/.config/Bitwarden\ CLI\ Work /path/to/bw $@"
```

Usando este ejemplo, podrías entonces usar inicio de sesión en dos cuentas ejecutando primero `source /ruta/a/.bashrc`, seguido de `bw-personal inicio de sesión` y `bw-trabajo inicio de sesión`.

Desbloquear

Usar una clave API o SSO para iniciar sesión requerirá que sigas el comando de `inicio de sesión` con un `bw desbloquear` explícito si vas a trabajar directamente con los datos de la caja fuerte.

Desbloquear su caja fuerte genera una **clave de sesión** que actúa como una clave de descifrado utilizada para interactuar con los datos en su caja fuerte. La clave de sesión debe ser utilizada para realizar cualquier comando que toque los datos de la caja fuerte (por ejemplo, `listar`, `obtener`, `editar`). Las claves de sesión son válidas hasta que se invalidan usando `bw bloquear` o `bw cerrar sesión`, sin embargo, no persistirán si abres una nueva ventana de terminal. Genera una nueva clave de sesión en cualquier momento utilizando:

```bash
bw unlock
```

Cuando hayas terminado, siempre finaliza tu sesión utilizando el comando `bw bloquear`.

Opciónes de desbloqueo

Puedes usar las opciones `--passwordenv` o `--passwordfile` con `bw desbloquear` para recuperar tu contraseña maestra en lugar de ingresarla manualmente, por ejemplo:

1. Lo siguiente buscará una variable de entorno `BW_PASSWORD`. Si `BW_PASSWORD` no está vacío y tiene valores correctos, el ILC logrará desbloquear y devolverá una clave de sesión:

   ```bash
   bw unlock --passwordenv BW_PASSWORD
   ```

2. Lo siguiente buscará el archivo `~Users/Me/Documents/mp.txt` (que debe tener tu contraseña maestra como la primera línea). Si el archivo no está vacío y tiene un valor correcto, el ILC logrará desbloquear y devolver una clave de sesión:

   ```bash
   bw unlock --passwordfile ~/Users/Me/Documents/mp.txt
   ```

> ⚠ **Warning**
>
> Si utiliza la opción `--passwordfile`, proteja su archivo de contraseña al bloquear el acceso solo al usuario que necesita ejecutar `bw desbloquear` y proporcionando solo acceso de lectura a ese usuario.

Usando una clave de sesión

Cuando desbloqueas tu caja fuerte usando `inicio de sesión bw` con correo electrónico y contraseña o `desbloquear bw`, la ILC devolverá tanto un comando `exportar BW_SESIÓN` (Bash) como `env:BW_SESIÓN` (PowerShell), incluyendo tu clave de sesión. Copia y pega la entrada relevante para guardar la variable de entorno requerida.

Con la variable de entorno BW_SESSION establecida, los comandos `bw` harán referencia a esa variable y se pueden ejecutar de manera limpia, por ejemplo:

```
Bash

export BW_SESSION="5PBYGU+5yt3RHcCjoeJKx/wByU34vokGRZjXpSH7Ylo8w=="

bw list items
```

Alternativamente, si no estableces la variable de entorno, puedes pasar la clave de la sesión como una opción con cada comando `bw`:

```
Bash

bw list items --session "5PBYGU+5yt3RHcCjoeJKx/wByU34vokGRZjXpSH7Ylo8w=="
```

> 💡 **Tip**
>
> Cuando hayas terminado, siempre finaliza tu sesión utilizando los comandos `bw bloquear` o `bw cerrar sesión`. Esto invalidará la clave de la sesión activa.

Comandos Principales

crear

El comando `crear` crea un nuevo objeto (`elemento`, `adjunto`, y más) en tu caja fuerte:

```
Bash

bw create (item|attachment|folder|org-collection) <encodedJson> [options]
```

El comando `crear` toma JSON codificado. Un flujo de trabajo típico para crear un objeto podría parecerse a algo como:

1. Utilice el comando `obtener plantilla` (vea obtener comandos centrales para más detalles) para generar la plantilla JSON apropiada para el tipo de objeto.

2. Utiliza un procesador JSON de línea de comandos como jq para manipular la plantilla de salida según sea necesario.

3. Utilice el comando `encode` (vea los detalles) para codificar el JSON manipulado.

4. Utiliza el comando `crear` para crear un objeto a partir del JSON codificado.

Por ejemplo:

```bash
bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder
```

o

```bash
bw get template item | jq ".name=\"My Login Item\" | .login=$(bw get template item.login | jq '.use
rname="jdoe" | .password="myp@ssword123"')" | bw encode | bw create item
```

Tras su creación exitosa, el objeto recién creado se devolverá como JSON.

crear otros tipos de elementos

El comando create crea de forma predeterminada un elemento de inicio de sesión, pero puede usar un procesador JSON de línea de comandos como jq para cambiar un atributo `.type=` para crear otros tipos de elementos :

Nombre	Valor
Entrada	`.tipo=1`
Nota segura	`.tipo=2`
Tarjeta	`.tipo=3`
Identidad	`.tipo=4`

Por ejemplo, el siguiente comando creará una nota segura:

```Bash
bw get template item | jq '.type = 2 | .secureNote.type = 0 | .notes = "Contents of my Secure Note." | .name = "My Secure Note"' | bw encode | bw create item
```

> **Note**
> Observe en el ejemplo anterior que las Notas seguras requieren una subplantilla (`.secureNote.type`). Puedes ver el tipo de sub-plantillas de elemento usando `bw get template` (ver aquí para detalles).

crear adjunto

El comando `crear archivo adjunto` adjunta un archivo a un **existente** elemento.

A diferencia de otras operaciones de `crear`, no necesitas usar un procesador JSON o `codificar` para crear un adjunto. En su lugar, use la opción `--file` para especificar el archivo adjunto y la opción `--itemid` para especificar el elemento al que se adjuntará. Por ejemplo:

```Bash
bw create attachment --file ./path/to/file --itemid 16b15b89-65b3-4639-ad2a-95052a6d8f66
```

> **Tip**
> Si no conoces el `itemid` exacto que quieres usar, usa `bw get elemento` para devolver el elemento (ver detalles), incluyendo su `id`.

obtener

El comando `get` recupera un solo objeto (`elemento`, `nombre de usuario`, `contraseña`, y más) de tu caja fuerte:

```Bash
bw get (item|username|password|uri|totp|exposed|attachment|folder|collection|organization|org-collection|template|fingerprint) <id> [options]
```

El comando `get` toma un elemento `id` o cadena para su argumento. Si usas una cadena (por ejemplo, cualquier cosa que no sea un `id` exacto), `get` buscará en tus objetos de caja fuerte uno con un valor que coincida. Por ejemplo, el siguiente comando devolvería una contraseña de Github:

```Bash
bw get password Github
```

> **Note**
> El comando `get` puede **devolver solo un resultado**, por lo que deberías usar términos de búsqueda específicos. Si se encuentran múltiples resultados, la ILC devolverá un error.

obtener adjunto

El comando `obtener archivo adjunto` descarga un archivo adjunto:

```
Bash

bw get attachment <filename> --itemid <id>
```

El comando `obtener adjunto` toma un `nombre de archivo` y un **id** `exacto`. Por defecto, `obtener adjunto` descargará el adjunto en el directorio de trabajo actual. Puedes usar la opción `--output` para especificar un directorio de salida diferente, por ejemplo:

```
Bash

bw get attachment photo.png --itemid 99ee88d2-6046-4ea7-92c2-acac464b1412 --output /Users/myaccoun
t/Pictures/
```

> **Note**
> Al usar `--output`, la ruta **debe** terminar con una barra inclinada hacia adelante (`/`) para especificar un directorio o un nombre de archivo (`/Users/myaccount/Pictures/photo.png`).

obtener notas

El comando `obtener notas` recupera la nota de cualquier elemento de la caja fuerte:

```
Bash

bw get notes <id>
```

El comando `obtener notas` toma un exacto elemento `id` o cadena. Si usas una cadena (por ejemplo, cualquier cosa que no sea un `id` exacto), `obtener notas` buscará en tus objetos de caja fuerte uno con un valor que coincida. Por ejemplo, el siguiente comando devolvería una nota de Github:

```
Bash

bw get notes Github
```

obtener plantilla

El comando `obtener plantilla` devuelve el formato JSON esperado para un objeto (`elemento`, `elemento.campo`, `elemento.inicio de sesión`, y más):

```
Bash

bw get template (item|item.field|item.login|item.login.uri|item.card|item.identity|item.securenote|folder|collection|item-collections|org-collection)
```

Mientras puedes usar `obtener plantilla` para mostrar el formato en tu pantalla, el uso más común es canalizar la salida a una operación de `bw crear`, utilizando un procesador JSON de línea de comandos como jq y `bw codificar` para manipular los valores obtenidos de la plantilla, por ejemplo:

```
Bash

bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder
```

> **Note**
>
> Cualquier plantilla `elemento.xxx` debe usarse como un sub-objeto de una plantilla `elemento`, por ejemplo:
>
> ```
> Bash
>
> bw get template item | jq ".name=\"My Login Item\" | .login=$(bw get template item.login | jq '.username="jdoe" | .password="myp@ssword123"')" | bw encode | bw create item
> ```

editar

El comando de `editar` edita un objeto (`elemento`, `colecciones de elementos`, etc.) en tu caja fuerte:

```
Bash

bw edit (item|item-collections|folder|org-collection) <id> [encodedJson] [options]
```

El comando de `editar` toma un **exacto** `id` (el objeto a editar) y JSON codificado (ediciones a realizar). Un flujo de trabajo típico podría parecer algo así como:

1. Utilice el comando `get` (vea los detalles) para obtener el objeto a editar.
2. Utiliza un procesador JSON de línea de comandos como jq para manipular el objeto de salida según sea necesario.
3. Utiliza el comando `encode` (ver detalles) para codificar el JSON manipulado.

4. Utilice el comando `editar` (incluyendo el objeto `id`) para editar el objeto.

Por ejemplo, para editar la contraseña de un elemento de inicio de sesión:

```
Bash

bw get item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328 | jq '.login.password="newp@ssw0rd"' | bw encode | bw edit item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328
```

O, para editar la(s) colección(es) en la que se encuentra un elemento:

```
Bash

echo '["5c926f4f-de9c-449b-8d5f-aec1011c48f6"]' | bw encode | bw edit item-collections 28399a57-73a0-45a3-80f8-aec1011c48f6 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

O, para editar una colección:

```
Bash

bw get org-collection ee9f9dc2-ec29-4b7f-9afb-aac8010631a1 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5 | jq '.name="My Collection"' | bw encode | bw edit org-collection ee9f9dc2-ec29-4b7f-9afb-aac8010631a1 --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

El comando de `editar` realizará una operación de **reemplazo** en el objeto. Una vez completado, el objeto actualizado se devolverá como JSON.

lista

El comando `lista` recupera una matriz de objetos (`elementos`, `carpetas`, `colecciones`, y más) de tu caja fuerte:

```
Bash

bw list (items|folders|collections|organizations|org-collections|org-members) [options]
```

Las opciones para el comando `list` son **filtros** utilizados para dictar lo que se devolverá, incluyendo `--url` , `--folderid` , `--collectionid` , `--organizationid` y `--trash`. Cualquier filtro aceptará `nulo` o `nonulo`. Combinar múltiples filtros en un solo comando realizará una operación OR, por ejemplo:

```
Bash

bw list items --folderid null --collectionid null
```

Este comando devolverá elementos que no están en una carpeta o colección.

Además, puedes **buscar** objetos específicos usando `--buscar`. Combinar filtro y buscar en un solo comando realizará una operación Y, por ejemplo:

```
Bash

bw list items --search github --folderid 9742101e-68b8-4a07-b5b1-9578b5f88e6f
```

Este comando buscará elementos con la cadena `github` en la carpeta especificada.

eliminar

El comando `eliminar` elimina un objeto de tu caja fuerte. `eliminar` toma **solo un exacto** `id` para su argumento.

```
Bash

bw delete (item|attachment|folder|org-collection) <id> [options]
```

Por defecto, `eliminar` enviará un elemento a la Papelera, donde permanecerá durante 30 días. Puedes eliminar permanentemente un elemento utilizando la opción `-p, --permanent`.

```
Bash

bw delete item 7063feab-4b10-472e-b64c-785e2b870b92 --permanent
```

Para eliminar una `colección-org`, también necesitarás especificar `--organizationid`. Vea IDs de la Organización.

> ⚠ **Warning**
>
> Mientras que los elementos que se eliminan usando `eliminar` pueden ser recuperados usando el comando de `restaurar` durante hasta 30 días (ver detalles), los elementos que se eliminan usando `eliminar --permanente` **se eliminan completamente y son irrecuperables.**

restaurar

El comando `restaurar` restaura un objeto eliminado de tu basura. `restaurar` toma **solo un exacto** `id` para su argumento.

```
Bash

bw restore (item) <id> [options]
```

Por ejemplo:

```Bash
bw restore item 7063feab-4b10-472e-b64c-785e2b870b92
```

enviar

El comando `enviar` crea un objeto Bitwarden Send para compartir efímeramente. Esta sección detallará operaciones simples de `enviar`, sin embargo, enviar es una herramienta altamente flexible y recomendamos referirse al artículo dedicado a Enviar desde ILC.

Para crear un texto simple Enviar:

```Bash
bw send -n "My First Send" -d 7 --hidden "The contents of my first text Send."
```

Para crear un archivo simple Enviar:

```Bash
bw send -n "A Sensitive File" -d 14 -f /Users/my_account/Documents/sensitive_file.pdf
```

recibir

El comando `recibir` accede a un objeto Bitwarden Enviar. Para recibir un objeto Enviar:

```Bash
bw receive --password passwordforaccess https://vault.bitwarden.com/#/send/yawoill8rk6VM6zCATXv2A/9WN8wD-hzsDJjfnXLeNc2Q
```

Comandos de organización

IDs de organización

Acceder a una organización desde la ILC requiere conocimiento de un ID para su organización, así como IDs para miembros individuales y colecciones.

Recupere esta información directamente desde la ILC usando `bw list`, por ejemplo:

```Bash
bw list organizations
bw list org-members --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
bw list org-collections --organizationid 4016326f-98b6-42ff-b9fc-ac63014988f5
```

> **Tip**
>
> Puedes `bw list` tanto `colecciones` como `org-colecciones`. El comando `bw list collections` enumerará todas las colecciones, independientemente de a qué organización pertenezcan. `bw list org-collections` enumerará solo las colecciones que pertenecen a la organización especificada usando `--organizationid`.

mover

> **Note**
>
> **Agosto 2021**: El comando de `compartir` ha sido cambiado a `mover`. Saber más .

El comando `mover` transfiere un elemento de la caja fuerte a una organización:

```
Bash

bw move <itemid> <organizationid> [encodedJson]
```

El comando `mover` requiere que `codifiques` un ID de colección, y toma un **id** `exacto` (el objeto para compartir) y un **organizationid** `exacto` (la organización para compartir el objeto). Por ejemplo:

```
Bash

echo '["bq209461-4129-4b8d-b760-acd401474va2"]' | bw encode | bw move ed42f44c-f81f-48de-a123-ad010
13132ca dfghbc921-04eb-43a7-84b1-ac74013bqb2e
```

Una vez completado, se devolverá el elemento actualizado.

confirmar

El comando `confirmar` confirma a los miembros invitados a su organización que han aceptado su invitación:

```
Bash

bw confirm org-member <id> --organizationid <orgid>
```

El comando `confirmar` toma un **exacto** miembro `id` y un **exacto** `organizationID`, por ejemplo:

```
Bash

bw confirm org-member 7063feab-4b10-472e-b64c-785e2b870b92 --organizationid 310d5ffd-e9a2-4451-af87
-ea054dce0f78
```

Otros comandos

configuración

El comando `config` especifica los ajustes que el Bitwarden ILC debe usar:

```
Bash

bw config server <setting> [value]
```

Un uso principal de `bw config` es para conectar tu ILC a un servidor de Bitwarden autoalojado:

```
Bash

bw config server https://your.bw.domain.com
```

Tip

Conéctate al servidor de la UE de Bitwarden ejecutando el siguiente comando:

```
Bash

bw config server https://vault.bitwarden.eu
```

Pase `bw config server` sin un valor para leer el servidor al que está conectado.

Los usuarios con configuraciones únicas pueden optar por especificar la URL de cada servicio de forma independiente. Tenga en cuenta que cualquier uso posterior del comando de configuración sobrescribirá todas las especificaciones anteriores, por lo que esto debe ejecutarse como un solo comando cada vez que haga un cambio:

```
Bash

bw config server --web-vault <url> \
  --api <url> \
  --identity <url> \
  --icons <url> \
  --notifications <url> \
  --events <url> \
  --key-connector <url>
```

> **Note**
>
> El comando `bw config server --key-connector` es necesario si su organización utiliza Conector de clave y está utilizando la opción `--apikey` para iniciar sesión después de haber eliminado su contraseña maestra.
>
> Contacta al propietario de una organización para obtener la URL requerida.

sincronización

El comando de `sincronización` descarga tu caja fuerte encriptada desde el servidor de Bitwarden. Este comando es más útil cuando ha cambiado algo en su bóveda de Bitwarden en otra aplicación cliente (por ejemplo, bóveda web, extensión del navegador, aplicación móvil) desde que inició sesión en la CLI.

```Bash
bw sync
```

Puedes pasar la opción `--last` para devolver solo la marca de tiempo (ISO 8601) de la última vez que se realizó una sincronización.

> **Tip**
>
> Es importante saber que la `sincronización` **solo realiza una extracción** del servidor. Los Datos se envían automáticamente al servidor cada vez que haces un cambio en tu caja fuerte (por ejemplo, `crear`, `editar`, `eliminar`).

codificar

El comando `encode` codifica en Base 64 stdin. Este comando se utiliza normalmente en combinación con un procesador JSON de línea de comandos como jq al realizar operaciones de `crear` y `editar`, por ejemplo:

```Bash
bw get template folder | jq '.name="My First Folder"' | bw encode | bw create folder

bw get item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328 | jq '.login.password="newp@ssw0rd"' | bw encode |
bw edit item 7ac9cae8-5067-4faf-b6ab-acfd00e2c328
```

importar

El comando de `importar` importa datos de una exportación de Bitwarden o de otra aplicación de administración de contraseñas compatible. El comando debe apuntar a un archivo e incluir los siguientes argumentos:

```Bash
bw import <format> <path>
```

Por ejemplo:

```Bash
bw import lastpasscsv /Users/myaccount/Documents/mydata.csv
```

> **Tip**
>
> ¡Bitwarden también admite muchos formatos para importar, demasiados para enumerar aquí! Usa `bw importar --formatos` para devolver la lista en tu ILC, o mira aquí.
>
> Si está importando un archivo .json cifrado que ha creado con una contraseña, se le pedirá que ingrese la contraseña antes de que se complete la importación.

exportar

El comando de `exportar` exporta los datos de la caja fuerte como un archivo `.json` o `.csv`, o un archivo .json encriptado:

```Bash
bw export [--output <filePath>] [--format <format>] [--password <password>] [--organizationid <orgi
d>]
```

Por defecto, el comando de `exportar` generará un `.csv` (equivalente a especificar `--format csv`) en el directorio de trabajo actual, sin embargo, puedes especificar:

- `--format json` para exportar un archivo `.json`
- `--format encrypted_json` para exportar un archivo .json cifrado
  - `--contraseña` para especificar una contraseña para usar para cifrar `encrypted_json` exportaciones en lugar de tu clave de cifrado de cuenta
- `--producción` para exportar a una ubicación específica
- `--raw` para devolver la exportación a la salida estándar en lugar de a un archivo

exportar desde una caja fuerte de organización

Usando el comando de `exportar` con la opción `--organizationid`, puedes exportar una caja fuerte de una organización:

```Bash
bw export --organizationid 7063feab-4b10-472e-b64c-785e2b870b92 --format json --output /Users/myacc
ount/Downloads/
```

generar

El comando `generar` genera una contraseña fuerte o frase de contraseña:

```
Bash

bw generate [--lowercase --uppercase --number --special --length <length> --passphrase --separator <separator> --words <words>]
```

Por defecto, el comando `generar` generará una contraseña de 14 caracteres con caracteres en mayúsculas, caracteres en minúsculas y números. Esto es equivalente a pasar:

```
Bash

bw generate -uln --length 14
```

Puedes generar contraseñas más complejas utilizando las opciones disponibles para el comando, incluyendo:

- `--mayúsculas`, `-m` (incluir mayúsculas)
- `--minúscula`, `-l` (incluir minúscula)
- `--número`, `-n` (incluye números)
- `--especial`, `-s` (incluye caracteres especiales)
- `--longitud` (longitud de la contraseña, mínimo de 5)

generar una frase de contraseña

Usando el comando `generar` con la opción `--contraseña`, puedes generar una frase de contraseña en lugar de una contraseña:

```
Bash

bw generate --passphrase --words <words> --separator <separator>
```

Por defecto, `bw generar --frase de contraseña` generará una frase de contraseña de tres palabras separadas por un guión (`-`). Esto es equivalente a pasar:

```
Bash

bw generate --passphrase --words 3 --separator -
```

Puedes generar una frase de contraseña compleja utilizando las opciones disponibles para el comando, incluyendo:

- `--palabras` (número de palabras)
- `--separador` (carácter separador)

- `--mayúsculas`, `-c` (incluir para poner en mayúsculas la primera letra de cada palabra de la frase de contraseña)
- `--includeNumber` (incluye números en la frase de contraseña)

actualizar

El comando de `actualizar` verifica si tu Bitwarden ILC está ejecutando la versión más reciente. `Actualizar` **no actualiza automáticamente el ILC para ti.**

```
Bash

bw update
```

Si se detecta una nueva versión, necesitarás descargar la nueva versión de la ILC utilizando la URL impresa para el ejecutable, o utilizando las herramientas disponibles para el gestor de paquetes que usaste para descargar la ILC (por ejemplo, `npm install -g @bitwarden/cli`).

estado

El comando `estado` devuelve información de estado sobre el Bitwarden ILC, incluyendo la URL del servidor configurado, la marca de tiempo para la última sincronización (ISO 8601), el correo electrónico del usuario y la ID, y el estado de la caja fuerte.

```
Bash

bw status
```

El estado devolverá información como un objeto JSON, por ejemplo:

```
Bash

{
  "serverUrl": "https://bitwarden.example.com",
  "lastSync": "2020-06-16T06:33:51.419Z",
  "userEmail": "user@example.com",
  "userId": "00000000-0000-0000-0000-000000000000",
  "status": "unlocked"
}
```

`estado` puede ser uno de los siguientes:

- `"desbloqueado"` , indica que ha iniciado sesión y su bóveda está desbloqueada (una variable de entorno de clave `BW_SESSION` se guarda con una clave de sesión activa)
- `"bloqueado"` , indica que ha iniciado sesión pero su bóveda está bloqueada (**no** se guarda ninguna variable de entorno de clave `BW_SESSION` con una clave de sesión activa)

- `"no autenticado"` , indica que no has iniciado sesión

> **Tip**
> Cuando `"estado": "no autenticado"`, `últimaSincronización`, `correoElectrónicoUsuario`, y `IDusuario` siempre devolverán `nulo`.

servir

El comando `serve` inicia un servidor web express local que se puede utilizar para realizar todas las acciones accesibles desde la ILC en forma de llamadas API RESTful desde una interfaz HTTP.

```Bash
bw serve --port <port> --hostname <hostname>
```

Por defecto, `serve` iniciará el servidor web en el puerto 8087, sin embargo, puedes especificar un puerto alternativo con la opción `--port`.

Por defecto, `serve` vinculará tu servidor web API a `localhost` sin embargo, puedes especificar un nombre de host alternativo con la opción `--hostname`. Las solicitudes de API solo pueden hacerse desde el nombre de host vinculado.

Por defecto, `servirá` bloqueará cualquier solicitud con un encabezado de `Origen`. Puedes eludir esta protección utilizando la opción `--disable-origin-protection`, sin embargo, **no se recomienda**.

> **⚠ Warning**
> Puede especificar `--hostname all` para no vincular el nombre de host, sin embargo, esto permitirá que cualquier máquina en la red haga solicitudes de API.

Vea las especificaciones de API para obtener ayuda para realizar llamadas con `servicio` .

Apéndices

Opciones globales

Las siguientes opciones están disponibles a nivel mundial:

Opción	Descripción
`--bonito`	Formato de salida. JSON está tabulado con dos espacios.
`--en bruto`	Devuelve la salida en bruto en lugar de un mensaje descriptivo.

Opción	Descripción
`--respuesta`	Devuelve una versión en formato JSON de la salida de la respuesta.
`--silencio`	No devuelvas nada a stdout. Podrías usar esta opción, por ejemplo, cuando diriges un valor de credencial a un archivo o aplicación.
`--nointeracción`	No solicite la entrada interactiva del usuario.
`--sesión`	Pase la clave de sesión en lugar de leerla de una variable de entorno.
`-v, --versión`	Muestra el número de versión de Bitwarden ILC.
`-h, --ayuda`	Muestra el texto de ayuda para el comando.

Completado de shell ZSH

El ILC de Bitwarden incluye soporte para la finalización de shell ZSH. Para configurar el completado automático en la terminal, utiliza uno de los siguientes métodos:

1. **Vanilla ZSH:** Agrega la siguiente línea a tu archivo `.zshrc`:

```bash
eval "$(bw completion --shell zsh); compdef _bw bw;"
```

2. **Vanilla (completaciones de proveedor):** Ejecute el siguiente comando:

```bash
bw completion --shell zsh | sudo tee /usr/share/zsh/vendor-completions/_bw
```

3. zinit: Ejecute los siguientes comandos:

```bash
bw completion --shell zsh > ~/.local/share/zsh/completions/_bw
zinit creinstall ~/.local/share/zsh/completions
```

Usando certificados autofirmados

Si su servidor Bitwarden autoalojado expone un certificado TLS autofirmado, especifique la variable de entorno Node.js NODE_EXTRA_CA_CERTS:

Bash

```bash
export NODE_EXTRA_CA_CERTS="absolute/path/to/your/certificates.pem"
```

PowerShell

```bash
$env:NODE_EXTRA_CA_CERTS="absolute/path/to/your/certificates.pem"
```

Enumeraciones

Las siguientes tablas enumeran los valores requeridos en los escenarios documentados:

Métodos de inicio de sesión en dos pasos

Se utiliza para especificar qué método de inicio de sesión de dos pasos usar al iniciar sesión:

Nombre	Valor
Autenticador	0
Correo electrónico	1
YubiKey	3

> **Note**
> FIDO2 y Duo no son compatibles con la ILC.

Tipos de elementos

Usado con el comando `crear` para especificar un tipo de elemento de caja fuerte:

Nombre	Valor
Entrada	1
Nota segura	2
Tarjeta	3
Identidad	4

Tipos de coincidencias de URI de inicio de sesión

Usado con el comando de `crear` y `editar` para especificar el comportamiento de detección de coincidencias de URI para un elemento de inicio de sesión:

Nombre	Valor
Dominio	0
Servidor	1
Comienza Con	2
Exacta	3

Nombre	Valor
Expresión Regular	4
Nunca	5

Tipos de campos

Usado con los comandos de `crear` y `editar` para configurar campos personalizados:

Nombre	Valor
Texto	0
Oculto	1
Booleano	2

Tipos de usuarios de la organización

Indica el tipo de usuario:

Nombre	Valor
Propietario	0
Administrador	1
Usuario	2
Gestor	3

Nombre	Valor
Personalizado	4

Estados de usuario de la organización

Indica el estado del usuario dentro de la organización:

Nombre	Valor
Invitado	0
Aceptado	1
Confirmado	2
Revocado	-1