

Documentación API.

1. [Lenguaje y librerías utilizadas.](#)
2. [Esquema de la API.](#)
3. [Método de vinculación.](#)

Lenguaje y librerías utilizadas.

Se decidió realizar el desarrollo utilizando **Python** junto al framework **Flask**. Este framework nos permite crear una API en pocas líneas mientras que **Python** nos permite enfocarnos directamente en las tareas que deberá realizar el *endpoint*.

Se utilizaron también las siguientes librerías externas, para agilizar varias sub-tareas:

- **pandas**. Con esta librería se crea un *dataframe* que contiene los *ingresos/egresos* y luego nos permite, de una manera muy sencilla, ordenar dichos registros en base al campo que deseemos (*fechaDeOperacion*, *montoTotal*)
- **flask api**. Esta librería, al igual que **Flask**, nos permite crear una API con la salvedad de que también nos proporciona una interfaz web para poder realizar las operaciones *REST*. Por el momento solo se proporciona un *endpoint* a modo de ejemplo (<http://127.0.0.1:5000/example/>).

Esquema de la API.

Se evaluó desarrollar una API *REST* pero creímos que el TP, al menos esta entrega, no lo justificaba ya que la API no tiene nunca contacto con la base de datos. Por lo que dichas operaciones, si se desean realizar, deberán hacerse mediante el *ORM* con **Java**, al menos por el momento.

La API dispone de un único *endpoint*. Si bien al principio se realizó un *endpoint* para cada operación luego decidimos unificar la llamada a la API y que las operaciones se ajusten acorde a los parámetros enviados en *request*.

Dicho *endpoint* es el siguiente <http://127.0.0.1:5000/link/> y posee los siguiente parámetros:

- *rangoFecha*: Un *integer* que representa la cantidad de días para validar los egresos. Por defecto **30 días**.
- *criterioOrden*: Criterio por el cual se ordenarán los ingresos y egresos. Debe ser uno de '*montoTotal*' o '*fechaDeOperacion*'. Por defecto '*montoTotal*'.
- *vincularPor*: Criterio por el cual se procederá a realizar la vinculación. Debe ser uno de '*ingresos*', '*egresos*', '*fecha*' o '*mix*'. Por defecto '*egresos*'.
- *criteria*: Una lista de strings que representa en que orden ejecutar las operaciones. Dichos elementos deberán ser uno/s de '*0*', '*1*', '*2*'. Por defecto *None*.
- *Ingresos*: lista de ingresos, **requerido**.
- *Egresos*: lista de egresos, **requerido**.

Método de vinculación.

A continuación se explicarán las funcionalidades intervinientes en el proceso de vinculación.

Ordenar eventos: aquí se ordenarán los eventos mediante la librería **pandas**, como se indicó anteriormente. El criterio por defecto será *montoTotal*, si se desea cambiar se deberá especificar en el *request* el parámetro *criterioOrden* y asignarle la otra opción, *fechaDeOperacion*. Decidimos que ambos conjuntos de datos, ingresos y egresos, compartan el mismo orden por lo cual el criterio aplicará a ambos por igual. Además, ambos conjuntos serán ordenados al momento de instanciar la clase *LinkEvents*.

Filtrar por fecha: esta funcionalidad se encargará de filtrar los egresos en base a la fecha de un ingreso. El enunciado indicaba que se puede proporcionar una *fecha desde* y una *fecha hasta*. Nosotros decidimos que la *fecha hasta* sea siempre la fecha del ingreso, de esta manera solo se debe pasar la cantidad de días retroactivos para tener en cuenta los egresos posibles. Por defecto esta cantidad son 30 días. Por ejemplo si tenemos un ingreso con fecha 25-09-2020 entonces solo se tendrán en cuenta aquellos egresos que figuren en el rango de 25-08-2020 a 25-09-2020. Llegamos a esta decisión ya que pensamos que no es correcto vincular ingresos y egresos de períodos distintos.

Vincular por egreso: consiste en comenzar a iterar los ingresos para luego comenzar con los egresos.

Vincular por ingreso: similar al anterior, pero se comienza por egresos y luego siguen los ingresos.

Vincular por fecha: idéntico a vincular por egreso con la salvedad que en este caso se ordena el conjunto por fecha.

Por defecto se vincula por egreso, sin embargo esto se puede especificar mediante el parámetro *vincularPor*.

Ejecutar mix: consiste en ejecutar las tres vinculaciones mencionadas en un orden establecido. Este orden es pasado mediante el parámetro *criteria*. Los ids disponibles son:

- '0': representa la vinculación por ingreso.
- '1': representa la vinculación por egreso.
- '2': representa la vinculaciones por fecha.

Si bien por el momento solo hay tres ids, se pueden repetir dentro de una lista (todos en *String*) en el orden en que se desea ejecutar. Ejemplo: *criteria* : ['0','2','0','1']