

Proyecto fin Bootcamp Aprender a programar desde cero

Flask + vanilla JS

Índice

Índice	2
A zambullirse donde cubre	3
Sobre este documento	3
La aplicación web: Registro de movimientos de criptomonedas	4
Objetivo y funcionamiento de la aplicación:	4
Backend con flask	5
LISTA DE MOVIMIENTOS	5
ALTA DE MOVIMIENTO	6
ESTADO DE LA INVERSIÓN	7
Ejemplo de cálculo del estado de la inversión	8
Frontend con javascript	9
Listado de movimientos	10
Estado de la inversión	11
Compra/venta/tradeo de criptomonedas	12
Tabla de SQLite	13
Forma de trabajo	14

A zambullirse donde cubre

Hasta ahora habéis estado ejercitando vuestras habilidades de programadores (tanto las básicas de pensamiento computacional como las avanzadas de programación en python y desarrollo de distintos tipos de pequeñas aplicaciones - juegos, escritorio y web -). Os toca soltaros de la orilla y nadar en lo hondo.

La idea es que practiquéis lo aprendido y que seáis capaces de desarrollar una pequeña aplicación (un prototipo) lo más completa funcionalmente y sin errores. Tenéis tres semanas para hacerlo durante las cuales deberéis investigar, decidir, estudiar y aprender cosas nuevas (eso ocurre en cualquier proyecto de la vida real).

Los dos puntos fundamentales son aprender y disfrutar. Utilizad lo aprendido y el resto de cursos de la plataforma para completar vuestro prototipo.

El objetivo es entregar el proyecto lo más avanzado posible, valorándose tanto el modo de resolución como el alcance. Sois libres de modificar, añadir o reemplazar funcionalidades que mejoren la aplicación. Ahora bien, si la mejora sirve para escaquearse del trabajo sólo irá en contra de la valoración final del proyecto.

Sobre este documento

Cada funcionalidad a desarrollar aparece en una tabla con la siguiente estructura.

NOMBRE DE LA FUNCIONALIDAD
<ul style="list-style-type: none">• Qué debe hacer la app para cubrir esta funcionalidad• Esta funcionalidad se describe en tantos puntos como sean necesarios• Para ser considerada cubierta deben cumplirse todos los puntos
Restricciones
<ul style="list-style-type: none">• Indicaciones más concretas de que debe hacer la funcionalidad• Suelen ser puntos claves que debe cumplir la aplicación. Una funcionalidad que no cumpla las restricciones es una funcionalidad incompleta
Opciones
<ul style="list-style-type: none">• Esto son características para que el programa mejore o te luzcas. Totalmente opcionales

A no ser que se indique, debes desarrollar todas las funcionalidades

La aplicación web: Registro de movimientos de criptomonedas

Vamos a hacer un registro de inversiones y compra/venta de criptos para jugar con los valores a ver si podemos hacer crecer nuestra inversión en euros o no.

Para ello nos vamos a crear una aplicación en flask que consultará el valor real en euros de las siguientes criptomonedas.

- **Monedas y criptomonedas** posibles:

EUR,	BTC,
ETH,	USDT,
BNB,	XRP,
ADA,	SOL,
DOT,	MATIC

La consultas de valores relativos entre estas criptomonedas se harán utilizando la api siguiente

- <https://coinmarketcap.com/api/>

para ello habrá que obtener la api key según sus instrucciones y utilizaremos estos endpoints de los que están en el plan básico que es gratuito. A saber:

- <https://pro-api.coinmarketcap.com/v2/tools/price-conversion>, para las conversiones
- <https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest>, para obtener el valor en euros de las primeras 10, por ejemplo (están las 9 de arriba)

Permite 10000 peticiones al mes, con límite de 30 peticiones por minuto.

Objetivo y funcionamiento de la aplicación:

La aplicación nos permitirá realizar las siguientes conversiones, que en adelante llamaremos from/to:

- **COMPRA:** De euros a cualquier cripto. Se considerará inversión y necesitaremos saber el total de euros invertidos a lo largo del tiempo. Consideramos que tenemos una cantidad infinita¹ de euros y que siempre podremos invertir más euros en criptomonedas.
- **TRADEO:** De una cripto a otra cripto. Intercambiaremos unas criptos por otras en función de su valor relativo. En este caso consideraremos que sólo tenemos las criptos que hayamos conseguido bien por COMPRA o bien por TRADEO
- **VENTA:** De cualquier cripto a euros: Se considerará recuperación de la inversión. Si al vender conseguimos más euros de los que invertimos será considerado beneficio, si obtenemos menos euros de los que invertimos será considerado pérdida.

Hay dos formas de conseguir más euros de los invertidos, la primera es el tiempo, es decir, comprar más barato que al momento de la venta.

¹ Sí, infinita. Siempre podemos conseguir más euros trabajando

La segunda es intercambiar criptos entre ellas de manera que, al cambiar su valor relativo con tanta velocidad se incrementa el número de criptomonedas de las que compramos, de forma que si vendemos cuando sean más caras y, además, tenemos más criptomonedas de las que compramos, aseguraremos la ganancia.

Cada una de estas transacciones debe grabarse en la base de datos y para conocer el valor relativo de criptomonedas en euros y criptomonedas entre ellas utilizaremos coinAPI.io.

Backend con flask

La api de backend estará formada por tres endpoints, a saber:

- **/api/v1/movimientos**: devolverá un json con una lista de movimientos con los movimientos (compras, ventas y conversiones de criptomonedas) realizadas por el usuario. Cada movimiento contendrá todos los campos de la tabla. Sólo aceptará peticiones de tipo GET

LISTA DE MOVIMIENTOS

/api/v1/movimientos

GET

- devolverá un json con una lista de movimientos con los movimientos (compras, ventas y conversiones de criptomonedas) realizadas por el usuario. Cada movimiento contendrá todos los campos de la tabla.

Restricciones

El formato de la respuesta debe ser:

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "fecha": "2021-01-01",
      "hora": "10:00:00",
      "from_moneda": "EUR",
      "from_cantidad": 3675,25,
      "to_moneda": "BTC",
      "to_cantidad": 1
    },
    {...}
  ]
}
```

En caso de error se devolverá siempre la estructura

```
{
```

```
{
  "status": "fail",
  "mensaje": "Mensaje de error"
}
```

Código de respuesta HTTP 400

- **/api/v1/tasa/<moneda_from>/<moneda_to>**: Sólo aceptará peticiones de tipo GET y devolverá la tasa de conversión entre la moneda from y la moneda to

TASA DE INTERCAMBIO

/api/v1/tasa/<moneda_from>/<moneda_to>

GET

- Si la moneda from no es EURO, debe comprobar que existe saldo suficiente antes de grabar el movimiento.

Debe devolver, si todo va bien, lo siguiente

```
{
  "status": "success",
  "rate": <Precio de 1 moneda_to, en moneda_from>(*),
  "monedas": [EUR, ...]
}
```

Código de respuesta HTTP: 201
(*) moneda_to: BTC, moneda_from: EUR → rate:19863.985674673

En caso de error se devolverá siempre la estructura

```
{
  "status": "fail",
  "mensaje": "Mensaje de error"
}
```

Código de respuesta HTTP: 400

En esta aplicación no se pueden modificar ni borrar movimientos. Se supone que la grabación del movimiento implica compra/venta real. **/api/v1/movimiento**: Sólo aceptará peticiones de tipo POST y servirá para dar alta de nuevos movimientos

- **/api/v1/movimiento**: Sólo aceptará peticiones de tipo POST para dar de alta una transacción (compra/venta/tradeo)

ALTA DE MOVIMIENTO

/api/v1/movimiento

POST



Restricciones

Debe enviarse el movimiento completo al servidor en forma de JSON

```
{
  "fecha": "2021-01-02",
  "hora": "11:00:00",
  "from_moneda": "BTC",
  "from_cantidad": 3675,25,
  "to_moneda": "BTC",
  "to_cantidad": 1
}
```

- Si la moneda from no es EURO, debe comprobar que existe saldo suficiente antes de grabar el movimiento.
- Saldo de una criptomoneda:
 - a. Sumamos todas las cantidades_to en filas cuya moneda_to es la criptomoneda buscada
 - b. Sumamos todas la cantidades_from en filas cuya moneda_from es la criptomoneda buscada
 - c. El saldo es la resta (a - b)

Debe devolver, si todo va bien, lo siguiente

```
{
  "status": "success",
  "id": <Nuevo id creado>,
  "monedas": [EUR, ...]
}
```

Código de respuesta HTTP: 201

En caso de que no exista saldo suficiente debe devolver lo siguiente

```
{
  "status": "fail",
  "mensaje": "Saldo insuficiente"
}
```

Código de respuesta HTTP: 200

En caso de error se devolverá siempre la estructura

```
{
  "status": "fail",
  "mensaje": "Mensaje de error"
}
```

Código de respuesta HTTP: 400

En esta aplicación no se pueden modificar ni borrar movimientos. Se supone que la grabación del movimiento implica compra/venta real.

- **/api/v1/status**: devolverá 4 conceptos relativos al estado de la inversión en euros

ESTADO DE LA INVERSIÓN

/api/v1/status

GET

- Devolverá el estado de la inversión

Restricciones

Debe devolver, si todo va bien, los siguientes valores referidos al estado de nuestra inversión en euros

- **Invertido**: Es el total de euros con el que se han comprado criptos. Se calcula como
 - la suma de la columna *Cantidad_from* de todos los movimientos cuya *Moneda_from* es EUROS.
- **Recuperado**: Es el total de euros obtenidos con la venta de cualquier cripto (registrada en nuestro sistema). Se calcula como
 - la suma de la columna *Cantidad_to* de todos los movimientos cuya *Moneda_to* es EUROS.
- **Valor de compra**: Es el resultado de la operación **Invertido - Recuperado**
- **Valor actual** en euros de nuestras criptos: **Al existir varias posibles criptos debemos**
 - Para cada cripto obtener su total como: La suma de *Cantidad_to* de todos los movimientos cuya *Moneda_to* es la cripto en cuestión menos la suma de *Cantidad_from* de todos los movimientos cuya *Moneda_from* es la cripto en cuestión.
Así si calculas la suma de las cantidades_to de todos los movimientos cuya moneda_to es ETH y le restas la suma de las cantidades_from de todos los movimientos cuya moneda_from es ETH, tendrás el saldo actual de ETH según tus movimientos

<https://rest.coinapi.io/v1/exchangerate/{base}/{quota}?time={time}&apikey={apikey}>

 - obteniendo el valor en euros {quota} actual de cada cripto {base} que poseemos (según su cantidad)
Así si en el punto anterior tenemos 1.7 ETH, la api de coinmarket-cap nos devolverá cuanto valen 1.7ETH en euros hoy.
 - Sumamos los euros de cada una de las criptos y ese será el valor actual de nuestra inversión que mostraremos en el control.
Haciendo eso mismo con todas las criptos de las que tengamos movimientos tendremos el valor actual total en euros de nuestra inversión

(*) Ver [ejemplo](#) más abajo



```
{
  "status": "success",
  "data": {
    "invertido": 1000.00,
    "recuperado": 750.00,
    "valor_compra": 250.00,
    "valor_actual": 11328,8186413112
  }
}
```

Código de respuesta HTTP: 200

(*) Los datos viajan con toda la precisión posible, esta se pierde en la presentación pero no en los cálculos

En caso de error se devolverá siempre la estructura

```
{
  "status": "fail",
  "mensaje": "Mensaje de error"
}
```

Código de respuesta HTTP: 400

Ejemplo de cálculo del estado de la inversión

Se entenderá mejor con un ejemplo.

Partimos de la siguiente tabla de movimientos

Movimientos

Fecha	hora	Moneda_from	Cantidad_from	Moneda_to	Cantidad_to
2021-01-01	10:00:00	EUR	1000	BTC	0,1
2021-01-01	10:05:00	BTC	0,05	ETH	2
2021-06-01	09:00:00	ETH	1	ETH	750

Qué leída por un humano significar que:

- El uno de enero de 2021 se compró 0,1 BTC gastando para ello 1000 Euros
- Ese mismo día se tomó 0,05 BTC y se intercambió por dos ETH
- En junio de 2021 uno de los dos ETH que tenía el usuario se vendió y se recuperaron 750 euros

Los cuatro conceptos vistos en este endpoint valdrán para este ejemplo:

- **Invertido:** Suma de todos los cantidad From cuya moneda from es EUR → 1000 €

- **Recuperado:** Suma de todos los cantidad to cuya moneda to es EUR \rightarrow 750 €

Esto significa que aún tenemos 250 euros atrapados en criptomonedas, que en esta pantalla están representados por:

- **Valor de compra:** Invertido - Recuperado \rightarrow 250 €

y por último, el valor actual de nuestra cartera de criptomonedas. Para ello debemos determinar que criptomonedas tenemos y que cantidad de las mismas.

BTC: Suma de cantidades to - Suma de cantidades from \rightarrow 0,1 - 0,05 \rightarrow 0,05 BTC

ETH: Suma de cantidades to - Suma de cantidades from \rightarrow 2 - 1 \rightarrow 1 ETH

Nuestra cartera está compuesta medio BTC y un ETH. ¿Cuánto valen hoy?. Accediendo a coinAPI.io

- Precio 1BT: 19951.323013646091300880153816 Euros
- Precio 1ETH: 1353.1571344881282544015089492 Euros

Luego el valor de nuestra cartera es $19951,32 * 0,05 + 1353,16 * 1 = 2.350,73$ Euros

- **Valor actual:** 2.350,73 €

Por tanto la ganancia (si vendiéramos todo ahora) sería 2100,73 Euros

Frontend con javascript


Se trata de, utilizando la API desarrollada en el apartado anterior crear una página única que aúne las tres funciones necesarias

myCRYPTO

Fecha	Hora	From	Cantidad	From	Cantidad
DD/MM/YYYY	HH:MM:SS	EUR	7.625,15	BTC	1,0000
DD/MM/YYYY	HH:MM:SS	BTC	1	LTC	40,0000
DD/MM/YYYY	HH:MM:SS	LTC	40	BTC	1,6000

Invertido: 7.625,15 €
 Recuperado: 5.324,27 €
 V. de compra: 2300,88 €
 V. actual: 4897,32 €

+

From: ▼
 Q: 

To: ▼
 Q:
 P.U.:

OK

© 2021 - Todos los derechos reservados

Listado de movimientos

Fecha	Hora	From	Cantidad	From	Cantidad
DD/MM/YYYY	HH:MM:SS	EUR	7.625,15	BTC	1,0000
DD/MM/YYYY	HH:MM:SS	BTC	1	LTC	40,0000
DD/MM/YYYY	HH:MM:SS	LTC	40	BTC	1,6000

Se trata de informar la tabla utilizando la api LISTA DE MOVIMIENTOS



LISTA DE MOVIMIENTOS

Informar la lista de movimientos de la app web (frontend)

Peticiones (ciclo de peticiones)

- Realizar la petición a **api/v1/movimientos** nada más cargar la pantalla y cada vez que se produzca una modificación (compra, venta o tradeo) en la base de datos.

Restricciones

Si no hay movimientos debe aparecer la tabla vacía

Si se produce un error en la consulta debe informarse al usuario con un alert

Estado de la inversión

Invertido: 7.625,15 €
Recuperado: 5.324,27 €
V. de compra: 2300,88 €
V. actual: 4897,32 €

Como las peticiones a CoinAPI.io son solo 100 al día, podrás lanzar el cálculo de la imagen o bien con un botón calcular o bien al hacer clic sobre este área.

ESTADO DE LA INVERSIÓN

Se actualizará a la carga de la página y cada vez que el usuario quiera saber el estado de su inversión añadiendo un botón refrescar/calcular o haciendo clic en el texto

Peticiones (ciclo de peticiones)

- Realizar la petición a **api/v1/status** a la carga de la página y cada vez que el usuario quiera saber el estado de su inversión añadiendo un botón refrescar/calcular o haciendo clic en el texto

Restricciones

- Mostrar error si se produce con un alert
- Poner los resultados negativos (pérdidas) en rojo

Compra/venta/tradeo de criptomonedas


+

From:

To:

Q:

Q:



P.U.:

Cancelar

OK

COMPRA/VENTA/TRADEO DE CRIPTOMONEDAS

La funcionalidad principal de la página. Para grabar un movimiento primero tendrás que activar el formulario al pulsar el botón +. Alternativamente podrás dejar el formulario siempre abierto.

Peticiones (ciclo de peticiones)

El flujo es el siguiente.

1. El usuario elegirá una moneda de la lista From (todas las monedas indicadas arriba)
2. El usuario elegirá una moneda de la lista To (todas las monedas indicadas arriba)
 - a. El sistema no permitirá grabar un movimiento con moneda From y moneda To iguales
3. El usuario ingresará una cantidad a convertir (Q_From) y pulsará el boton calcular.
4. Al pulsar el botón calcular. El sistema llamará a la api de coinmarketcap
 - a. <https://rest.coinapi.io/v1/exchangerate/{base}/{quote}?time={time}&apikey={apikey}>
5. La api devolverá la cantidad de moneda To equivalente que será presentada en el campo Q_To. También se calculará el precio unitario y se mostrará en el campo P.U.:
6. Si el usuario ve conveniente realizar la compra pulsará el botón OK. Que llamará a la api api/v1/ movimiento POST para grabarlo.
7. Si no hay errores (hay saldo suficiente) y se ha grabado el movimiento, el navegador deberá refrescar la lista de movimientos llamando a api/v1/movimientos GET

Restricciones

- Debe existir saldo suficiente en la cuenta del usuario de cualquier cripto. Euros hay siempre infinitos
- Las monedas from y to deben ser diferentes.

Tabla de SQLite

Movimientos	
Columna	Tipo
id	integer, clave primaria
date	Text (formato YYYY-MM-DD)
time	Text (formato HH:mm:ss)
moneda_from	Text (la clave de la moneda según la lista de arriba)
cantidad_from	Real
moneda_to	Text (la clave de la moneda según la lista de arriba)
cantidad_to	Real

Forma de trabajo

El proyecto será individual, oficialmente tenéis dos semanas para hacerlo comenzando este periodo el **20 de octubre de 2025** y debe ser entregado el **2 de noviembre del 2025 antes de las 23:59 hora peninsular española (Madrid)**.

Se os indicará antes del inicio qué formulario utilizar para realizar la entrega.

La entrega consistirá en indicarnos el **enlace al repositorio remoto** en que habéis dejado vuestro programa. Este repositorio deberá incluir en el fichero **readme.md** las instrucciones especiales de instalación si las hubiera. No obstante sabéis utilizar entornos virtuales y el fichero requirements.txt con ello debería bastar. Debéis incluir instrucciones para crear la base de datos (preferible) o bien la base de datos vacía como parte del repositorio.

Los proyectos serán corregidos en los días siguientes a la entrega y os informaré por correo de vuestra calificación.