

Proyecto fin Bootcamp Aprender a programar desde cero

Flask classic

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
Routes	5
Funcionalidades	5
Ejemplo	10
Tabla de SQLite	11
Forma de trabajo	11

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://www.coinapi.io/>

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

Permite 100 peticiones diarias.

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 crypto. 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 crypto a otra crypto. 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 crypto 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.

Routes

La aplicación estará formada por tres páginas o routes, a saber:

- **/:** mostrará una tabla con los movimientos (compras y conversiones de criptomonedas) realizadas por el usuario
- **/purchase:** mostrará un formulario en el que realizar una compra, venta o intercambio de monedas. Se podrá comprar BTC en euros y vender BTC a euros, el resto de crypto monedas se intercambiarán por BTC y entre ellas
- **/status:** mostrará en pantalla el estado de la inversión, los euros gastados en comprar BTC y el valor actual del total de criptomonedas que existan en el stock del usuario según sus movimientos, así como una lista de todas las criptomonedas y sus saldos actuales y su valor en euros

Cómo hemos visto se puede dar algo de estilo a la aplicación con frameworks css. Un framework muy ligerito y fácil de manejar es pico.css. Sentíos libres de investigar y utilizar otros más complejos. Recordad que tenéis los recursos de la plataforma (cursos de HTML y CSS) y disponéis de estas semanas de navidad para investigar.

Funcionalidades

ESTRUCTURA DE LA WEB

- Todas las páginas tendrán el mismo header y footer.
- Todas las páginas tendrán el menú con las opciones.
 - **Inicio** → lleva a la página de movimientos
 - **Compra** → lleva a la página de nueva compra, venta o tradeo
 - **Estado** → lleva a la página de estado de la inversión

Restricciones

En cada página debe estar inactivo la opción en la que nos encontramos, así:

- **Inicio:** inactivo en página de listado de movimientos
- **Compra:** inactivo en página de compra / venta / tradeo de monedas
- **Estado:** inactivo en página de estado de la inversión

Opciones

- Pinta de un color a la opción de menú en la cual te encuentras.

myCRYPTO						
Inicio / Compra / Status						
MOVIMIENTOS						
Fecha	Hora	From	Q	To	Q	P.U
dd/mm/yyyy	hh:mm:ss	EUR	7.625,15	BTC	1,00	7625,15 €
dd/mm/yyyy	hh:mm:ss	BTC	0,13	LTC	20,00	0,0067 BTC
dd/mm/yyyy	hh:mm:ss	BTC	0,87	ETH	45,00	0,019 BTC
© 2020 - Todos los derechos reservados						

fig 1. El precio unitario es opcional

PORTADA
/
<ul style="list-style-type: none"> Mostrará el listado de movimientos
Restricciones
<ul style="list-style-type: none"> Si no hay movimientos, aparecerá el texto SIN MOVIMIENTOS.



myCRYPTO

Inicio / Compra / Status

COMPRA DE CRIPTOS

From:

To:

Q:

P.U.:^(*)

(*) Opcional

☐

© 2020 - Todos los derechos reservados

fig 2. El precio unitario es opcional, los botones calculadora y aceptar no necesitan hacerse con iconos, puedes hacerlos con texto

COMPRA/TRADEO/VENTA DE CRIPTOS

Usa la api con la petición que hay más abajo. La forma que pone en la página de coinApi.io es más compleja, pero puedes usarla como dice allí si quieres.

Route: /purchase [GET]

- Aparecerá un formulario con dos listas con las diferentes monedas que pueden comprarse, venderse o intercambiarse.
 - **Monedas o criptomonedas** posibles:

EUR,	BTC,
ETH,	USDT,
BNB,	XRP,
ADA,	SOL,
DOT,	MATIC
- Una de las listas será la **lista From**. Se elegirá una de las monedas y en el campo **Q** el usuario pondrá la cantidad de la moneda que se desea gastar.
- En la **lista To** se elegirá la moneda que se desea comprar con la cantidad Q de monedas From
- El sistema consultará con [coinAPI.io](https://coinapi.io) para obtener el total de monedas To que se

pueden conseguir al precio actual, para eso se utilizará el botón calculadora o <Calcular>

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

- Si se acepta la operación se grabará en la tabla MOVIMIENTOS

Route: /purchase [POST]

- Si el botón pulsado es la calculadora devolverá el mismo formulario con la cantidad de monedas To según la consulta de arriba. Este botón no tiene porque tener una calculadora, basta con el texto Calcular.



- Si el botón pulsado es aceptar validará el formulario e insertará en MOVIMIENTOS si los datos son correctos. No tiene porque tener el icono de check, basta con el texto Comprar o Aceptar.



Restricciones

1. Los movimientos se grabarán sobre una tabla con estructura mínima:
 - Fecha
 - Hora
 - Moneda_from
 - Cantidad_from
 - Moneda_to
 - Cantidad_to
2. El precio unitario no hace falta es siempre **Cantidad_from/Cantidad_to**
3. Si se pretende tradear o vender una criptomoneda, el sistema no permitirá vender más cantidad de la criptomoneda que se pretende vender que el saldo de la misma que figure en la tabla movimientos.
 - Saldo de una criptomoneda:
 - i. Sumamos todas las cantidades_to en filas cuya moneda_to es la criptomoneda buscada
 - ii. Sumamos todas la cantidades_from en filas cuya moneda_from es la criptomoneda buscada
 - iii. El saldo es la resta (i - ii)
4. Debe comprobarse que el movimiento (compra, venta o tradeo) no cambia en nada desde la pulsación de Calcular a la de aceptar. Investiga sobre los campos ocultos (HiddenField) de WTF.

myCRYPTO

Inicio / Compra / Status

ESTADO DE LA INVERSIÓN

Moneda	Cantidad	Valor en euros
BTC	0.0	0.0
ETH	0.095	144.98
USDT	35.9800	42.71

Valor actual: 187.69

Precio de compra: -180.00

Resultado de la Inversión: 7.69

ESTADO DE LA INVERSION EN EUROS

Route: /status

- Esta pantalla mostrará el estado de la inversión, como sigue
 - **Wallet:** Mostrará el saldo de todas las criptomonedas en poder del usuario, en la pantalla se muestra como una tabla.
 - **Precio de compra:** Son los euros invertidos en la compra de todas nuestras criptomonedas. Se calcula como la suma de la columna *Cantidad_to* de todas las filas con *Moneda_to* igual a EUR menos la suma de la columna *Cantidad_from* de todas las filas con *Moneda_from* igual a EUR. Un valor negativo indica que gastamos dinero en criptos que aún no hemos recuperado al venderlas. Un valor positivo indica que vendimos criptomonedas por un valor superior al de su compra.
 - **Valor actual:** Es el precio en euros de todas las criptomonedas que forman el wallet. **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
- **Resultado de la inversión:** La suma de los dos conceptos anteriores. Si es positiva, significa que nuestra cartera de criptomonedas (de venderla en este momento) nos daría ganancias, si es negativa, pérdidas.

OPCIONES

Mostrar el resultado de la inversión en color. Si es negativo es pérdidas (en rojo) si es positivo es ganancia (en verde)

(*) Este punto siempre da problemas. Léelo con mucho detalle y pregunta hasta que tenga sentido

Ejemplo

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 conceptos vistos en este endpoint valdrán para este ejemplo:

- **Precio de compra:** Suma de todos los cantidad to cuya moneda to es EUR $\rightarrow 750 \text{ €}$ menos la suma de todos los cantidad From cuya moneda from es EUR $\rightarrow 1000 \text{ €}$ es decir $\rightarrow - 250 \text{ €}$.

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

- **Precio de compra:** Invertido - Recuperado $\rightarrow - 250 \text{ €}$

Ahora a determinar el **Valor actual**, para ello debemos determinar que criptomonedas tenemos y que cantidad de las mismas.

BTC: Suma de cantidades to - Suma de cantidades from $\rightarrow 0,1 - 0,05 \rightarrow 0,05 \text{ BTC}$

ETH: Suma de cantidades to - Suma de cantidades from $\rightarrow 2 - 1 \rightarrow 1 \text{ ETH}$

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 $19951,32 * 0,05 + 1353,16 * 1 = 2.350,73 \text{ Euros}$

- **Valor actual:** 2.350,73 €

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

Y este es el valor que presentamos como **Estado de la inversión**, que es la suma de Valor actual (siempre positivo) y Precio de compra (casi siempre negativo).

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 tres semanas para hacerlo comenzando este periodo el **11 de julio de 2023** y debe ser entregado el **31 de julio del 2023 antes de las 23:59 hora peninsular española (Madrid)**.

Se os indicará antes de la fecha final de entrega 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.