

## Tarea 4: Red de Bitcoin

Entrega: Viernes – 14 – Noviembre – 2025 – (23:59 hora Chile continental)

### 1. Cosas administrativas

Cada tarea en este curso equivale a 20 % de la nota final. En total vamos a tener 6 tareas, pero la peor tarea que solucionan no cuenta para la nota final. Quiere decir que pueden botar una tarea.

El programa con su solución hay que subir por el buzón de canvas.

Uso de materiales (o soluciones) encontradas en Internet está permitido. Solo se les pide citar la fuente que utilizaron. No se aplica ninguna penalidad para su uso.

### 2. La tarea

En esta tarea simularemos lo que hace un full node de Bitcoin: conseguir bloques enteros.

Usando el código explicado en clases (`block.py` y `network.py`), deberían agregar soporte para los mensajes que nos permiten conseguir primeros 20 bloques de la testnet de Bitcoin (el genesis block ya está definido como una constante en nuestro código). Aquí se los pide descargar los bloques enteros, y no solo los headers (entonces tendrán que incluir todas las transacciones).

Para esto, necesitan hacer dos cosas. Primero, necesitan definir el mensaje que les permite conseguir un bloque entero desde un nodo. Como la respuesta a este mensaje, el nodo les enviará la serialización de un bloque. Para manejar este objeto, tendrán que implementar la clase `FullBlock` en `block.py`. En el código entregado ya existe un borrador de la clase con los métodos necesarios para su funcionamiento correcto. Para poder definir el método `parse` de la clase `FullBlock`, necesitan parsear todas las transacciones que ocurren en este bloque. Para parsear una transacción pueden ocupar el código disponible en `txP2PKH.py`.

Todos los detalles de serialización y de la lógica de mensajes que se mandan por la red de Bitcoin pueden encontrar en [https://en.bitcoin.it/wiki/Protocol\\_documentation](https://en.bitcoin.it/wiki/Protocol_documentation). Si no entienden cómo definir a un mensaje o cómo mandarlo a un nodo tienen que revisar el código entregado en la clase sobre la red de Bitcoin.

Una vez implementado, deberían hacer lo siguiente:

1. [4.5 puntos] Implementar el método `main` en el archivo `main.py` el cual debe descargar y retornar como una lista ordenada de `FullBlock` los primeros 20 bloques del testnet de Bitcoin, partiendo por el bloque original.
2. [1.5 punto] Una vez que tienen los bloques, intenten validar las transacciones de cada bloque usando la clase `Tx` definida en `txP2PKH.py`. La ejecución resultará en un error. Expliquen por qué ocurre este error. No es necesario corregirlo, solo se les pide por qué la validación falla.

### 3. Entrega

Su entrega debe ser un archivo `zip` (con el nombre que quieran) que contenga todos los archivos necesarios para ejecutar el archivo `main.py`, incluyendo este último. Adicionalmente, deben incluir un archivo en formato Markdown con su respuesta para el ítem 2.