

## Introducción

*Este obligatorio tiene 4 partes, hacen foco a distintas tecnologías vistas en el curso. Una primera parte a resolver con Bash, una segunda con POSIX, una tercera con ADA y cuarta con Docker.*

## Parte 1 Bash

### Cambio



Un cambio bancario necesita realizar un sistema con determinadas características para realizar sus actividades. Estas actividades son todas referidas a la compra venta de divisas.

En el sistema se debe poder:

le tengo que dar permisos de ejecucion  
`chmod +x ej1.sh`

`ej1.sh`  
`#!/bin/bash`  
comandos

**Cargar divisas:** Se le brindará al sistema un archivo .txt, de donde tomara el valor de las divisas. Este archivo puede contener cotizaciones de días anteriores, debiendo seleccionar la del día actual. Si no hay archivo o no hay cotización del día actual, debe informarlo (mediante mensaje en pantalla). En cualquier caso, se debe generar un archivo de nombre "carga.txt" indicando fecha y si la carga fue exitosa o no.

**Comprar Divisas:** Se puede comprar Euros, Dólares Americanos, Pesos Argentino y Reales. Se debe dar el importe en pesos uruguayos. Se debe especificar el monto que se compra. Todas las transacciones deben ser registradas en un archivo de nombre "Compra de Divisa.txt".

**Vender divisas:** Se puede vender Euros, Dólares Americanos, Pesos Argentino y Reales. Se debe dar el importe en pesos uruguayos. Se debe especificar el monto que se vende. Todas las transacciones deben ser registradas en un archivo de nombre "Compras de Divisas.txt".

**Buscar transacciones por fecha:** Se debe poder buscar transacciones por fecha.

**Buscar transacciones por divisa:** Se debe poder buscar transacciones por divisa.

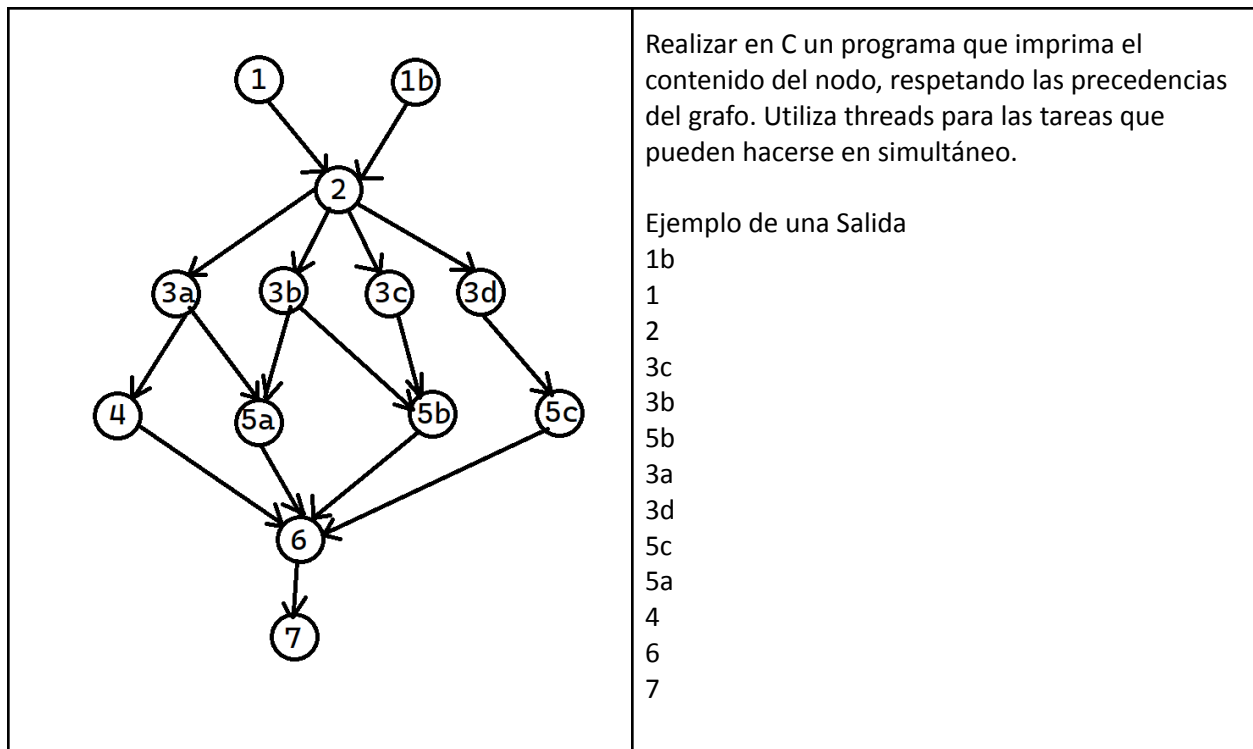
**Buscar transacciones por número de finalizaciones** Se debe poder buscar transacciones que finalicen su número en el dígito 3.  
~~número~~  
~~monto~~

**Bloquear transacciones:** Cuando se selecciona esta opción se debe pedir confirmación y si se bloquea no podrán realizarse más transacciones. Los archivos de estas transacciones deben de quedar sin permisos para escritura.

**Desbloquear transacciones:** Cuando se selecciona esta opción se debe pedir confirmación y si se desbloquea se podrán realizar transacciones. Los archivos de estas transacciones deben de quedar con permisos para escritura.

**Terminar:** Se termina el programa.

## Parte 2



### Parte 3



El CIE (Centro de Innovación y Emprendimientos de la Universidad ORT) le solicita a usted un software para automatizar sus rondas de inversión. Una ronda de inversión busca conectar inversores con emprendedores. Consiste en diferentes mesas (cada uno con inversores como espectadores) y un equipo emprendedor exponiendo su idea. Cada equipo tiene un límite de tiempo antes que suene la campana. Al sonar la campana deberán de rotar a otra mesa, hasta pasar por todas las mesas. Cuando todos los emprendedores pasan por todas las mesas, existen reuniones más concretas donde los emprendedores acuerdan con los inversores. Esta metodología requiere que los emprendedores puedan expresar sus ideas en poco tiempo, y sean capaces de defenderla ante las preguntas de los inversores.

Para emular este problema, se tiene en cuenta;

- En cada mesa se sientan 4 inversores y expone únicamente un equipo emprendedor.
- Existen 3 mesas de inversores. Los inversores permanecen en su mesa.
- Existen 7 equipos de emprendedores y 12 inversores.
- Cada mesa para que comience la exposición, debe tener a los inversores y un equipo emprendedor exponiendo.
- En una ronda de inversión (solo es necesario simular una ronda), termina cuando cada emprendedor pasa por todas las mesas una única vez.
- Mientras la mesa está ocupada, los inversores no pueden levantarse y tampoco cambiarse de expositor.
- Los tiempos de cada emprendedor en cada mesa serán aleatorios, de máximo 3 segundos aprox.
- Existe una sala de espera, donde entran los emprendedores mientras no están en una mesa. La misma tiene una capacidad de 6, quedando un grupo en el pasillo.
- Los inversores no pueden repetir la mesa durante el transcurso de la ronda.

Salida posible:

```
Equipo 1 entra a la sala de espera
Equipo 3 entra a la sala de espera
Equipo 5 entra a la sala de espera
Equipo 2 entra a la sala de espera
Equipo 4 entra a la sala de espera
```

Equipo 6 entra a la sala de espera  
Equipo 6 ingresa a la mesa 1  
Equipo 7 entra a la sala de espera  
Equipo 5 ingresa a la mesa 3  
Equipo 6 se va de la mesa 1  
Equipo 4 ingresa a la mesa 2  
Equipo 6 entra a la sala de espera  
Equipo 4 se va de la mesa 2  
Equipo 2 ingresa a la mesa 2  
Equipo 4 entra a la sala de espera  
Equipo 5 se va de la mesa 3  
...

#### **Parte 4**

Problema:

Tienes que configurar una aplicación de notas utilizando Node.js con Express y MongoDB como base de datos, la app Node.js y la DB deben estar en contenedores separados.

La aplicación ya tiene el código fuente y la estructura de carpetas preparada, incluyendo las funcionalidades para crear notas con título y descripción, traer todas las notas almacenadas en la base de datos y también traer una nota específica por su ID.

El usuario podrá interactuar con la app a través de la interfaz GraphQL, esto se hace entrando desde el navegador a:

<http://localhost:3001/graphql>

Su tarea es, crear los archivos Dockerfile, .dockerignore y docker-compose.yml para configurar los dos contenedores y conectarlos.

- El Dockerfile debe configurar un contenedor de Docker para la aplicación Node.js. Para esto, debes utilizar una imagen de Node.js y copiar los archivos necesarios al contenedor. Además, debes incluir los comandos npm i para instalar las dependencias necesarias, y npm run dev para ejecutar la aplicación.
- El .dockerignore debe excluir la carpeta data y los node\_modules, pues la primera sólo se usará para guardar la información de la base de datos, y la segunda se genera automáticamente.
- En cuanto al docker-compose.yml, debes asegurarte de que los puertos utilizados por los contenedores estén definidos como variables traídas del archivo .env.

- Por último, debes montar la base de datos MongoDB en un volumen de Docker para que los datos se conserven entre reinicios de contenedores, vinculados a la carpeta antes mencionada "data".

**Entrega:**

- La entrega debe indicar claramente de qué parte se trata.
- En el código debe estar los nombres y números de estudiantes de los integrantes del grupo.

El archivo zip debe contener también un pdf, que contenga carátula, presentación de los integrantes con foto, índice, documentación del trabajo realizado y el código en el anexo.