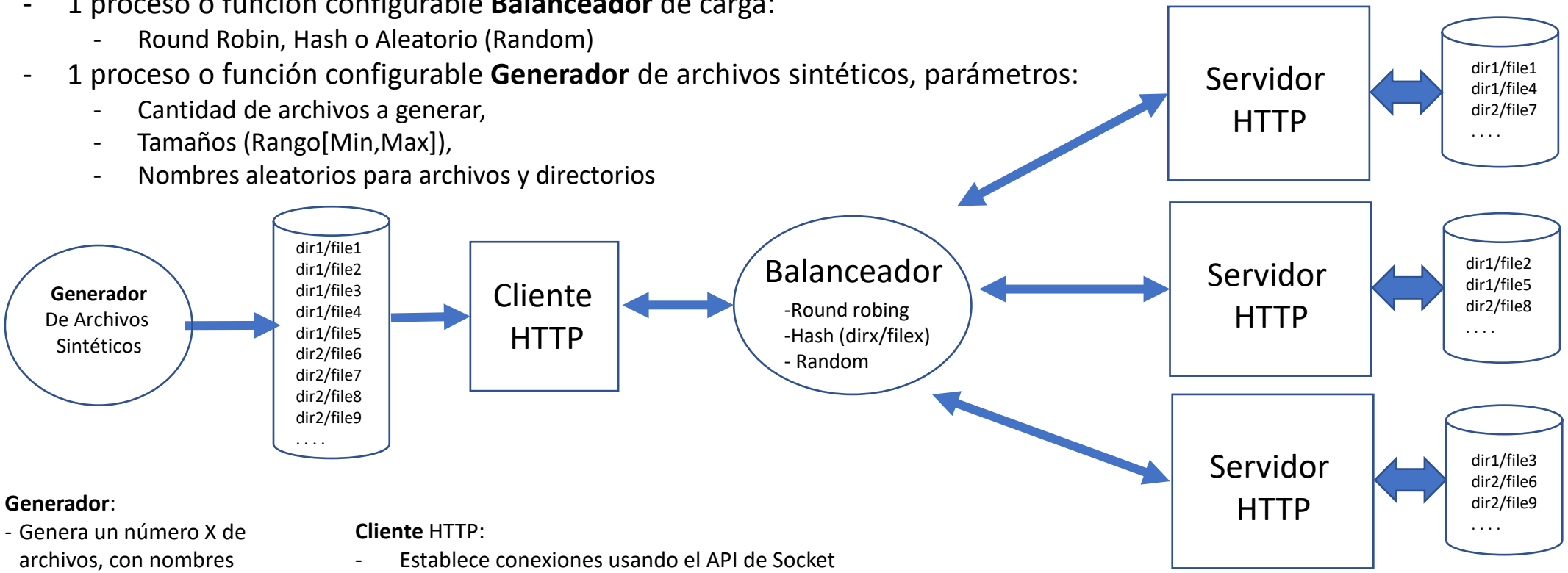
Procesos requeridos:

- 2 procesos extremos: **Cliente** y **Servidor**
- 1 proceso o función configurable **Balanceador** de carga:
  - Round Robin, Hash o Aleatorio (Random)
- 1 proceso o función configurable **Generador** de archivos sintéticos, parámetros:
  - Cantidad de archivos a generar,
  - Tamaños (Rango[Min,Max]),
  - Nombres aleatorios para archivos y directorios



Generador:

- Genera un número X de archivos, con nombres aleatorios y de tamaños aleatorios con rango (min,max).
- El contenido de los archivos es irrelevante.

Cliente HTTP:

- Establece conexiones usando el API de Socket de alguno de estos lenguajes: C, Java o Python. **NO** usa librerías de HTTP!
- Contiene dos funciones básicas: GET y PUT
- Ejemplo de operaciones:
  - **PUT dirx/filex** Crear el archivo filex dentro de dirx. Sino existe dirx lo crea.
  - **GET dirx/** Obtener la lista de archivos que residen en dirx.
  - **GET dirx/filex** Obtener el archivo filex.
- Limitación: Solo se usará un nivel de directorio, no se pedirán objetos del tipo: dirx/diry/filex (con subdirectorios)

Balanceador:

- Sirve solo de intermediario para balancear la carga de almacenamiento de los archivos.
- Se podrá configurar para usar balanceo tipo:
  - Round robin
  - Hash(dirx/filex)
  - Random

Servidor HTTP:

- Almacena/accede a los archivos del Cliente.
- Ejecuta las operaciones: GET y PUT que le envía el cliente.

Escenario de pruebas:

- 1 Cliente (más balanceador)
- 3 Servidores
- 3 grupos de archivos generados de manera sintética:
  - G1 (minis): 100 archivos de entre 1 Byte y 10KB
  - G2: (pequeños): 100 archivos de entre 10KB y 1MB
  - G3: (medianos): 100 archivos de entre 1MB y 10M

TRES BLOQUES DE PRUEBAS: BLOQUE 1, BLOQUE 2 Y BLOQUE 3

BLOQUE 1:
Uso de balanceo **Round Robin**.
Pruebas:
RRCG1: Carga de los archivos del grupo G1
RRCG2: Carga de los archivos del grupo G2
RRCG3: Carga de los archivos del grupo G3

RRDG1: Descarga de los archivos del grupo G1
RRDG2: Descarga de los archivos del grupo G2
RRDG3: Descarga de los archivos del grupo G3

BLOQUE 2:
Uso de balanceo **Hash**.
Pruebas:
HCG1: Carga de los archivos del grupo G1
HCG2: Carga de los archivos del grupo G2
HCG3: Carga de los archivos del grupo G3

HDG1: Descarga de los archivos del grupo G1
HDG2: Descarga de los archivos del grupo G2
HDG3: Descarga de los archivos del grupo G3

BLOQUE 1:
Uso de balanceo **Aleatorio**.
Pruebas:
ACG1: Carga de los archivos del grupo G1
ACG2: Carga de los archivos del grupo G2
ACG3: Carga de los archivos del grupo G3

ADG1: Descarga de los archivos del grupo G1
ADG2: Descarga de los archivos del grupo G2
ADG3: Descarga de los archivos del grupo G3

MÉTRICAS

Para cada tipo de balanceo de carga y para cada grupo de archivos (Bloque 1, Bloque 2 y Bloque 3), obtener:

- Tiempo promedio para hacer las **cargas** de archivos.
- Tiempo promedio para hacer las **descargas** de archivos.
- Cantidad de almacenamiento utilizado en cada Servidor.

PRIMERA ENTREGA (18/FEB):

Procesos Cliente y Servidor, sin el balanceador de carga y sin generador de archivos sintético:

- El proceso cliente envía un archivo al Servidor con el comando PUT /dirx/filex
- El proceso cliente solicita al servidor el archivo con GET /dirx/filex

Ejemplo 1: El Cliente carga un archivo en el Servidor

Cliente:

- Abrir socket TCP
- Enviar mensaje HTTP simplificado
- El mensaje contiene:

Ejemplo de mensaje de petición del Cliente:

cabecera
- **PUT dirx/filex**
- **Content-lenght: Tamaño del archivo**

Línea en blanco

cuerpo
- Contenido del archivo
- . . . . .
- . . . .
- ….

Servidor:

- Espera por conexiones
- Recibe el mensaje HTTP del Cliente
  - Primero la cabecera para saber los detalles de la solicitud.
  - Al ver que es un PUT, lee el archivo que le envía el Cliente después de la línea en blanco.
  - Guardar el archivo, creando el directorio si no existiera.
- Responder mensaje con OK o ERROR.

Ejemplo de mensaje de respuesta del Servidor:

cabecera
- **200 OK**

Línea en banco

Cuerpo: Estaría vacío.

Ejemplo 2: El Cliente solicita un archivo al Servidor

Ejemplo de mensaje de petición del Ciente:

GET dirx/filex
Línea en blanco
Contenido vacío

Ejemplo de mensaje de respuesta del Servidor:

cabecera
- **200 OK**
- **Content-lenght: Tamaño del archivo**

Línea en blanco

cuerpo
- Contenido del archivo
- . . . . .
- . . . .
- ….

ENTREGA FINAL (Fecha por definir):

Se espera la implementación de todo el proceso descrito en la lámina 1, programas y reporte de resultados.