

## Sesiones de navegación

*Temas evaluados: Abstracción, estructuras de datos enlazadas, arrays, resolución de problemas.*

### Definición del contexto

Un servidor web requiere incorporar historial y análisis de tráfico de sesiones de navegación. Para lo cual establece las siguientes definiciones técnicas:

1. La cantidad de sesiones simultáneas sólo estará delimitada por la memoria asignada al programa.
2. Por cada nueva sesión, se debe agregar a la estructura de sesiones en memoria, un nuevo elemento con los siguientes datos:

*ID de sesión (entero), fechaHoraInicio (entero largo), fechaHoraFin (entero largo), códigoNavegador (0=Chrome, 1=Firefox, 2=Safari, 3=IE/Edge, 4=Otro).*

3. A su vez, cada sesión debe mantener el histórico de páginas visitadas por el usuario, de forma tal de poder retornar la dirección (URL) de la anterior visitada, cada vez que lo solicite el navegador.
4. Por cada solicitud de nueva página, el servidor asocia a la sesión los siguientes datos:

*URL (cadena de caracteres), fechaHora (entero largo).*

Además, cada 24 horas el sistema genera un informe sobre todas las sesiones finalizadas (*fechaHoraFin != 0*), detallando:

*Fecha informe (DDMMAAAA), cantidad de sesiones, duración promedio (en segundos) de las sesiones, cantidad de páginas navegadas en total y un array (N=5) con la cantidad de sesiones por código de navegador.*

Que se guarda en un archivo binario de registros.

### Objetivos:

1. Desarrollar la función *agregarPagina* que guarda la página visitada. Si la sesión no existe, se debe crear. Recibe por parámetros: las sesiones en memoria, ID de sesión, el código de navegador y la URL de la página.
2. Desarrollar la función *obtenerUltimaPagina* que recibe por parámetros: las sesiones en memoria, el ID de sesión y retorna la URL de la última página visitada. Se debe eliminar de memoria la página retornada.
3. Desarrollar la función *crearInforme* que recibe la estructura con todas las sesiones y crea un informe con las sesiones finalizadas. Recibe por parámetros: la estructura con las sesiones y un puntero al archivo.

*Dispone y puede utilizar (sin desarrollar) las funciones:*

- ✓ *long ahora()* que retorna el resultado a invocar a la función *time(NULL)* de la biblioteca *time.h*
- ✓ *long diferencia(long, long)* que recibe 2 resultados de invocar a *ahora()* y retorna la diferencia de tiempo en segundos.
- ✓ *string fecha(long)* que retorna el resultado de invocar a *ahora()* al formato DDMMAAAA.
- ✓ Todas las funciones que son parte del apunte de Cátedra (biblioteca de funciones).

### Se pide

1. (2 puntos) **Codifique la declaración** de todas las estructuras de datos necesarias.
2. (3 puntos) **Codifique la función del punto 1.**
3. (2 puntos) **Codifique la función del punto 2.**
4. (3 puntos) **Codifique la función del punto 3.**