

La Secretaría Regional de Valparaíso del Ministerio de Salud cuenta con datos almacenados dentro de un archivo CSV que registra las camas UCI ocupadas diariamente a nivel regional. Este archivo representa los datos mediante la siguiente estructura de 3 campos: `fecha;codigo_hospital;numero_de_cama`

El valor del campo `fecha` corresponde a la fecha (en formato `aaaaammdd`) en que fue generado el registro. El valor del campo `codigo_hospital` corresponde a un identificador único para cada hospital de la región y el valor del campo `numero_de_cama` es la identificación de la misma. Al respecto, una cama es única en un hospital; si el mismo número se repite en dos hospitales distintos se trata, entonces, de dos camas diferentes.

Ejemplo del contenido del archivo:

<pre>fecha;codigo_hospital;numero_de_cama 20210101;17;122 20210101;14;218 20210101;15;127 20210101;16;122 20210101;17;122</pre>	<p>Puede descargar el archivo desde el siguiente enlace:  <a href="http://ping.appserver.science/eedd/camas_ocupadas.csv.zip">http://ping.appserver.science/eedd/camas_ocupadas.csv.zip</a></p> <p>Advertencia: en el archivo podrían haber duplicados, debido principalmente a errores de digitación.</p>
---	--

Usted debe implementar un programa en lenguaje C++ que dé cobertura a los siguientes requerimientos:

1. *leeProcesa(string, Lista)*: Al iniciarse la ejecución del programa, este invoca a la función **leeProcesa** que leerá el contenido del archivo CSV y almacenará su contenido completo en la memoria utilizando para su almacenamiento una lista de tipo LLS.
2. *nuevaListaSinDuplicados(Lista, Lista)*: Escribir el operador **nuevaListaSinDuplicados** que partir de la LLS genera una nueva LLS sin repetidos, dejando sólo una ocurrencia de cada información presente en la lista (A partir del punto 3, esta nueva lista será la que trabajaremos).
3. *seleccionar(Lista, int, Lista)*: Proveer la funcionalidad **seleccionar**, que a partir de los datos contenidos en una LLS como la creada en el punto 2 y un código de un hospital pasado como parámetro, creará una nueva LLS únicamente con los datos de las camas del hospital referido.
4. *estadistico(Lista)*: Proveer la funcionalidad **estadístico**, que dada una LLS permitirá mostrar en pantalla la cantidad de camas registradas por cada hospital. A modo de ejemplo:

```
hospital 17 tiene en total 112 cama(s) distintas.
hospital 14 tiene en total 16 cama(s) distintas.
hospital 15 tiene en total 18 cama(s) distintas.
...
```
5. *contar(Lista, int, int, int)*: Proveer la funcionalidad **contar**, que dada una LLS y una fecha (día, mes, año) retorna la cantidad de camas ocupadas existentes de esa fecha.
6. *eliminarHospital(Lista, int)*: Proveer la funcionalidad **eliminarHospital**, que dada una LLS y un identificador de hospital eliminará todas las ocurrencias contenidas en la LLS asociadas a dicho hospital.
7. Proveer de un menú de diálogo para el usuario que le permita acceder a las funciones definidas en los puntos anteriores.
  - a. Crear Lista desde archivo
  - b. Crear Lista sin duplicados
  - c. Crear Lista de un Hospital
  - d. Mostrar estadísticas de número de camas por Hospital
  - e. Cantidad de camas ocupadas por fecha especificada.
  - f. Eliminar registros de camas de un hospital
  - g. Salir

Nota: Al seleccionar cada opción, debe mostrar la lista respectiva.

Prepare un video donde aparezca su grupo explicando su estrategia de resolución de código de programa y lo muestra funcionando.

**Pauta de evaluación:**

Funcionalidad	Comentarios/indicaciones	Puntos
Declarativa	Crea y utiliza correctamente una declarativa de LLS de acuerdo al problema planteado	10
leeProcesa (*)	Implementa correctamente la lectura desde archivo hacia una LLS	10
nuevaListaSinDuplicados (*)	Implementa correctamente la creación de una nueva lista sin repetidos	10
Seleccionar (*)	Implementa correctamente la selección y crea una nueva lista	10
estadístico	Implementa correctamente la acción de mostrar la cantidad de camas registradas por cada hospital	10
contar	Implementa correctamente el operador que devuelve la cantidad de camas ocupadas que hay en la LLS	5
eliminarHospital (*)	Implementa correctamente la eliminación de todas las ocurrencias contenidas en la LLS	15
Menú	Implementa correctamente el menú para acceder a las funcionalidades descritas anteriormente	5
Código fuente	Se visualiza un código ordenado, indentado, documentado y que hace uso de nombres de variables representativas.  En la cabecera de su código de programa deberá indicar nombres de los integrantes, sección y nombre del profesor.	5
Video explicativo	Crea un video explicando uno de los operadores realizado por usted. Tiempo: 3 a 5 minutos de exposición. No se requiere edición especial de video, pero sí se debe escuchar la voz y ver el rostro de cada integrante del grupo. Coloque el link del video en Youtube al inicio de los comentarios de su código.	20
	<b>TOTAL PUNTOS</b>	<b>100</b>

**Observaciones:**

- Fecha de publicación: **lunes 19 de abril 2021, 20:00 hrs.**
- Fecha de entrega: **viernes 7 de mayo 2021, 20:00 hrs.**
- Nota: Tome las precauciones del caso para entregar la tarea dentro del plazo. No se aceptan tareas fuera de plazo.
- El trabajo es desarrollado **en parejas escogidas por el profesor o profesora** (de a 2), colocando la identificación de las o los autores, sección y profesor como comentario dentro del código.
- Deberá adjuntar el link del video en el mismo código fuente. Cada integrante del grupo (pareja) debe aparecer en el video.
- Solo uno de los integrantes entrega el link de su solución en onlinegdb con el proyecto completo y listo para ejecutar en un clic. Subir a <http://cursos.informatica.utem.cl>
- Si el proyecto no compila, o es copia, obtendrá nota mínima 1.0.