|  |  |
| --- | --- |
| **Instituto Tecnológico de Costa Rica**  **Ingeniería en Computación**  **Sede Regional San Carlos** | Primer Tarea Programada  Lenguajes de Programación |
| Prof. Oscar Víquez Acuña. | SpotiCry |

**Descripción:**

Una aplicación que permita la reproducción de música en una computadora o dispositivo puede ser un proyecto de software retador debido a la cantidad de elementos que se deben considerar entre los cuales se pueden enumerar, el almacenamiento, búsqueda, organización y reproducción de canciones.

Se pretende que el programador diseñe e implemente una aplicación que permita la administración de canciones en un esquema donde un módulo se encargue de la atención sincrónica de solicitudes por parte de clientes y justamente el otro módulo implemente un cliente que en potencia podría instanciarse en múltiples ocasiones.

El módulo que atiende solicitudes deberá administrar canciones de forma local y deberá dar respuestas a solicitudes que el cliente le realice.

La implementación de dicho módulo debe permitir diversas técnicas de programación imperativa eficiente y concurrente, pero sobre todo debe generar un programa ejecutable en máquina real.

Es por esto que dicho módulo se implementará en una aplicación utilizando el lenguaje Go que permita el acceso a solicitudes en paralelo por parte de diversos clientes. Esta solución, para efectos de simplicidad y no propiamente porque sea la mejor solución posible, se hará utilizando sockets implementando una arquitectura simple de clientes y servidor a través del TCP-IP. En este esquema, esta aplicación funcionará como un servidor que atenderá una serie de solicitudes de clientes que no serán otros que instancias del módulo/aplicación que reproduzca las canciones y realice la interacción en general con el cliente implementada en otro lenguaje de programación siguiendo prácticas y estrategias de programación funcional.

**El Servidor:**

La implementación del servidor debe cumplir con las siguientes características/acciones:

* Listas de canciones según criterios de búsqueda definidos a gusto del grupo de programadores programador (al menos 3 criterios diferentes). Se espera para cada canción tener la información del nombre de la misma, que no necesariamente es el mismo nombre del archivo que la contiene. Si se desea algo más queda a criterio de los programadores
* Inicio y finalización de reproducción de canciones de forma local (que implicará decodificación y envío de paquetes referentes a los bytes que necesitará el cliente para enviar la canción a sonar)
* Funciones para administrar canciones donde se permita mínimamente agregar y eliminar las canciones a partir de archivos de audio (por ejemplo MP3)

Cabe recordar que el servidor es un cliente que atiende solicitudes por lo que cada grupo debe establecer el formato de las instrucciones que se envían de parte del cliente por el protocolo TCP-IP de manera que dicha comunicación y la decodificación de lo que se solicita y los parámetros de opciones sean claros y concisos.

Específicamente para las APIs de administración de canciones, lo que se quiere es que el servidor tenga su propia interfaz de texto donde se permita agregar canciones a partir de archivos de texto locales en la máquina y eliminar canciones existentes siempre que no se encuentren siendo reproducidas. Es importante mencionar que una canción que está siendo reproducida no es lo mismo que una canción que fue consultada, por lo que no es lo mismo que un cliente haya solicitado el nombre de la canción para por ejemplo ponerla en una lista a que esté reproduciendo dicha canción y las implicaciones deben considerarse para efectos prácticos del sistema.

Se espera que cada grupo defina sus estrategias de programación en este sentido y se da cierta libertad de implementación para obtener resultados diferentes aunque con el mismo propósito.

**El Cliente:**

El cliente será una aplicación desarrollada en otro lenguaje de programación pero siempre una aplicación que corra localmente. El lenguaje de programación que se solicita es el lenguaje F# por ser un lenguaje mayoritariamente de carácter funcional, aunque si el grupo de programación elige otro lenguaje, deberá justificar la elección y deberá realizar la programación siguiendo las especificaciones de carácter funcional que en su momento serán detalladas.

La aplicación del cliente 100% funcional debe cumplir con los siguientes requerimientos:

* Se debe contar con una interfaz gráfica de usuario que permita administrar listas de canciones y que permita reproducir dichas canciones.
* Se debe permitir buscar canciones con al menos 3 criterios diferentes. Cada criterio es definido por el grupo de programación pero deben ser suficientemente diferentes para no ser resueltos de la misma forma técnica del lado del servidor.
* Se debe permitir crear listas de reproducción y las canciones buscadas en el servidor podrían agregarse a más de una lista de reproducción
* La canción en reproducción debe contar con un buffer que permita adelantar y retroceder cuantas veces se quiera para la misma canción. Solamente la canción de turno que se esté reproduciendo

**Documentación:**

Para documentar el trabajo se requiere que se haga uso del formato establecido por la Carrera de Computación para este fin. Se solicita especial énfasis en la documentación de aspectos como:

* Manejo de archivos de música
* Administración en memoria de la lista de canciones del lado del servidor
* Detalles sobre manejo de conflictos y excepciones durante la ejecución de los módulos
* Hallazgos en torno a la implementación de sockets y procesos en paralelo. Cómo funciona el programa sin sincronización y cómo sería el escenario implementando cierta sincronización.
* Qué elementos consideran, deben tomarse en cuenta mejorar/cambiar, con miras en una implementación robusta de un sistema similar en producción a gran escala?

**Notas Finales para la Tarea:**

* Tarea grupal para máximo dos personas
* Fecha de entrega: Miércoles 20 de setiembre de 2023 antes de las 10:00 pm.
* **Los aspectos de programación funcional específicos a solicitar para el cliente serán especificados luego de cubrir el paradigma funcional en clase.**
* Se recomienda que se empiece a trabajar desde hoy.
* Cualquier tipo de fraude será severamente castigado.
* La entrega del trabajo se hará a través del Tec-Digital.