|  |  |
| --- | --- |
| **Instituto Tecnológico de Costa Rica**  **Ingeniería en Computación**  **Sede Regional San Carlos** | Primer Tarea Programada  Lenguajes de Programación |
| Prof. Oscar Víquez Acuña. | Sistema para compra de entradas a eventos |

**Descripción:**

La compra de entradas a eventos masivos en Costa Rica es un proceso que puede ser agobiante en el tanto la cantidad de solicitudes alcance picos altos en periodos relativamente cortos de tiempo, tanto debido a temas de conectividad y saturación, como a eficiencia en la respuesta de las soluciones hardware-software del sistema de venta de entradas.

Se pretende que el programador diseñe e implemente una aplicación que permita la búsqueda, selección y compra de entradas a eventos. El núcleo de la aplicación se basa en el “motor de búsqueda y compra de entradas” que es un módulo que como su nombre lo dice, debe buscar las mejores entradas disponibles según ciertos parámetros que son solicitados al usuario y la posterior reserva y compra o cancelación de estas. Para la implementación de este motor de búsqueda, se requiere una solución rápida y eficiente (desde el punto de vista de ejecución del sistema) por lo que la implementación de dicho módulo se realizará en el lenguaje Rust, que permite diversas técnicas de programación imperativa eficiente, pero sobre seguro, que genera un programa ejecutable en máquina real.

Dicho módulo se implementará en una aplicación en Rust 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, la aplicación de búsqueda y eventual reserva o cancelación de boletos funcionará como un servidor que atenderá una serie de solicitudes de clientes que no serán otros que instancias simuladas de compras de clientes reales a partir de un sistema implementado en otra aplicación pero siguiendo prácticas y estrategias de programación funcional primordialmente.

**El Servidor:**

De esta forma se quiere que se diseñe e implemente una solución de este módulo, que permita contemplar las siguientes características:

* Mapear en memoria las zonas del inmueble en donde se realizará el evento mediante el uso de categorías, zonas y por organización de sillas (número de fila y número de silla). Cada categoría podría tener varias zonas, Ej: Categoría Platea Este, zonas A,B,C, etc y cada zona, una serie finita de asientos organizados por filas, Ej: (4 asientos consecutivos) 🡪 Platea Este, Zona A, fila 5, asientos 6,7,8,9. Se debe entonces generar una estructura en memoria que permita mapear las posibilidades totales de asientos para los eventos y que será el espacio de búsquedas en donde se podrá buscar según el criterio de los clientes.
* Contemplar un estado para cada silla del inmueble bajo las características de libre, reservada, comprada. Para la búsqueda de opciones, se tomarán como candidatos solo aquellas que estén libres. Las reservadas se marcan así porque se está en espera de que el cliente decida aceptarlas (mediante un pago hipotético) o cancelarlas. Una vez aceptado lo anterior, el estado de la silla pasaría a comprada.
* Resolver solicitud de compra donde el cliente solicita boletos para una categoría específica y una cantidad de boletos. El algoritmo debe escogen la mejor opción posible de asignación de boletos tratando de cumplir con los siguientes criterios:
  + Mejor vista del evento (criterio que define cada quien)
  + Asientos juntos primero, siempre que sea posible y si no, lo más cerca posible según filas (que, por ejemplo, si no hay 6 boletos juntos en ningún lado, pero si 5 juntos, el sexto esté lo más cerca posible de los otros 5.

Contemplar que existen varias zonas para la misma categoría y por tanto los mismos boletos juntos que se encuentran en una zona, podrían estar más cerca en otra zona de la misma categoría.

* Que no se puedan asignar más de N cantidad de boletos definidos por el programador. Quiere decir que se limite la búsqueda de boletos a un máximo definido previamente y que pueda ser cambiado en el momento que se requiera. Si un cliente quiere comprar más del máximo, se rechaza la compra y se le indica el por qué.

Al estar frente a un sistema que potencialmente puede recibir muchas solicitudes de clientes al mismo tiempo, se debe trabajar mediante el uso de programación concurrente utilizando hilos, que le permitan al servidor recibir en paralelo una cantidad potencialmente significativa de solicitudes de entradas de clientes.

*Se debe implementar una aplicación adicional que simule la eventual ejecución de varios cliente en diferentes tiempos. Esta aplicación tipo TESTER, deberá invocar varias pruebas quemadas para demostrar la funcionalidad y eficiencia del servidor por lo que debería además imprimir estadísticas de uso y de tiempo de respuesta.*

La interacción entre clientes y servidor en términos generales es la siguiente:

1. Se realiza una conexión al servidor y el cliente de turno solicita las entradas que quiera en la zona que quiera o que le sugieran al servidor.
2. El servidor recibe la solicitud y hace la búsqueda de las mejores opciones según los criterios mencionados.
3. El servidor le comunica al cliente el resultado de la búsqueda. Las entradas quedan en estado reservado
4. El cliente decide si las quiere o no (se puede tardar un tiempo hipotético pero con máximos definidos en este proceso) y le comunica al servidor su decisión
5. El servidor cambia el estado de las entradas a comprado, comunica al cliente la compra exitosa y cierra la conexión con el cliente

**El Cliente:**

Esta parte del proyecto será definida en el enunciado del segundo proyecto pero tomará como base este primer proyecto y por ende puede que sea necesario modificaciones al mismo para garantizar el mejor funcionamiento en conjunto de las dos aplicaciones.

**Documentación:**

Para documentar el trabajo se requiere que cree y adjunte el enlace a un video no mayor a 7 minutos en donde se solicita especial énfasis en los siguientes aspectos:

* Ejecución del programa
* Algoritmos de resolución de solicitudes y búsqueda de boletos.
* Mapeo del escenario a nivel de memoria
* Hallazgos en torno a la implementación de sockets y procesos en paralelo. Cómo funciona el programa mencionando elementos de hilos y si se usa o no sincronización de hilos y las razones del por qué si o por qué no
* 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:**

* La tarea se puede desarrollar de forma individual o en parejas (se deben mantener las parejas para el segundo proyecto)
* Fecha de entrega: Lunes 26 de setiembre de 2024 antes de las 10:00 pm.
* 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.

Ideas de croquis



