

Simulador de Atención a Tickets

1. Introducción

El presente proyecto pretende poner en práctica los conceptos desarrollados durante el curso IC2101 Programación Orientada a Objetos del I Semestre del 2015.

2. ¿Qué se busca con esta Tarea?

- Practicar las habilidades de modelado de aplicaciones de software haciendo uso del lenguaje de modelado UML.
- Aumentar el conocimiento del estudiante sobre el lenguaje de programación orientado a objetos Java.
- Practicar la experimentación y la resolución de problemas bajo el paradigma orientado a objetos.
- Ejercitar la toma de decisiones sobre el dominio del problema y del dominio de la solución.
- Fomentar la investigación por parte del estudiante
 - Uso de sockets para la creación de una aplicación cliente-servidor.
 - Lectura y creación de archivos en Excel a partir del lenguaje Java.
- Fomentar el trabajo en equipo.

3. El contexto y las funcionalidades esperadas

El proyecto consta de la propuesta y construcción de una aplicación distribuida que se compone de dos partes:

- a. Una aplicación servidora que simula la recepción de “tickets” provenientes de clientes, los cataloga en categorías Roja, Amarilla, Verde y los coloca en colas de atención, las cuales serán la fuente de información de un proceso de atención por parte de empleados especializados en los distintos tipos de categorías.
- b. Una aplicación cliente que se activará en varios equipos que representan los empleados de la compañía que se conectan al servidor para ir atendiendo los tickets que ha recibido el servidor, los atiende y luego los reporta nuevamente al servidor para indicar que ya han sido atendidos.

Descripción de la aplicación servidora y cliente.**Recepción de tickets**

La aplicación servidora al iniciarse, procesa un archivo en Excel que deberá llamarse **TICKETS.xls**, en el cual se simula la recepción de un conjunto de registros que representan el ingreso de tickets o inconformidades registrados por clientes asociados a la empresa.

El archivo en Excel mantiene el siguiente formato:

	A	B	C	D	E	F	G	H	I	J
1	Fecha y Hora de Recepción	ID_CLIENTE	Asunto	Ticket_ID	Categoría	ID_EMPLEADO	Fecha y Hora de atención	Tiempo en segundos	Comentario	Estado
2					ROJA AMARILLA VERDE					PENDIENTE EN ATENCION ATENDIDA
3	Registrado desde el sistema	Aleatorio (Inventado)	Cualquiera	Consecutivo	Código de empleado	Registrado según simulación				

Al momento de leer los datos el servidor sólo se interesa por obtener la fecha y hora en la que se recibe el ticket, el id del cliente y el asunto.

La aplicación servidora cuenta con una “carpeta INBOX” donde se reciben todas las solicitudes de atención de tickets que provienen de clientes que se han conectado para reportar una inconformidad. Esta carpeta se alimenta por medio de una acción que tiene el servidor para cargar nuevos tickets y que sólo puede ser ejecutada por el administrador.

El id del ticket es un número secuencial generado automáticamente en el momento de recepción por parte del servidor.

Al momento de ingreso y recepción de los tickets, se registran con un estado de **PENDIENTE**.

Una vez recibidos y cargados por el servidor, se presentarán en una pantalla del servidor, donde un usuario tipo ADMINISTRADOR (que se encargó de levantar el servidor) puede clasificarlos en tres tipos de categorías representadas por un color:

CATEGORIA URGENTE
CATEGORIA MEDIA
CATEGORIA LEVE

Este proceso se puede realizar procesando los tickets por bloques (seleccionando uno o varios manualmente y asignando una categoría). Considere la posibilidad de generar este proceso automático.

La determinación del color o grado de intensidad del ticket es definida por el administrador y sólo puede ser realizada a los tickets recién ingresados (con estado de **PENDIENTE**).

Una vez asignada la categoría, este proceso provoca que los tickets se muevan de la carpeta INBOX a una de tres colas: COLA_ROJA, COLA_AMARILLA, COLA_VERDE dejando la carpeta INBOX únicamente con aquellos tickets que no se hayan categorizado.

Los empleados

En la aplicación servidora, existe registrado un conjunto de empleados que se conectarán al servidor.

Estos empleados serán cada uno de los miembros del equipo de trabajo y cada uno de ellos cuenta con:

- Un login (será su correo del TEC, ejemplo ersolano@tec.ac.cr)
- Una contraseña (su número de carnet, ejemplo 201403221)
- Nombre completo
- Categoría actual de trabajo (ROJA, VERDE, AMARILLA) la cual denota la especialidad de atención de los tickets.
- Estado (CONECTADO, DESCONECTADO)

Los servicios

El servidor puede atender las siguientes peticiones por parte de las aplicaciones Cliente:

1. Conectar empleado: Los empleados se conectan al servidor desde una aplicación cliente y si las credenciales son correctas, en el servidor el empleado se modifica como CONECTADO. Esta acción se identificará con el código **CONECTAR_EMPLEADO**.

2. Solicitar tickets que aún no han sido atendidos de acuerdo al color del cliente conectado (ROJO, VERDE, AMARILLO): En esta acción se recibe como parte de la petición el color y se devuelve al cliente una lista con los tickets solicitados. Esta acción se identificará por el código **SOLICITAR_TICKETS**.
3. Recibir la indicación de que un ticket se comenzará a atender, para lo cual recibe el número de ticket y procede a modificar su estado pasándolo de **PENDIENTE** a **EN ATENCION** retornando un valor de **TRUE** al cliente lo que le permite garantizarse que sólo el podrá modificar su información de atención. Si el ticket al momento de modificar su estado no se encuentra en estado de **PENDIENTE** entonces no podrá modificar su estado y por lo tanto retorna **FALSE**.
Esta acción se identificará por el código **RESERVAR_TICKET**.
4. Registrar los detalles de atención de ticket, para lo cual en la aplicación cliente se envía completa la queja atendida que reporta: el usuario que a atendió, el tiempo invertido en la resolución expresado en segundos, un comentario sobre la resolución, la fecha y hora de resolución. Esto provoca que los datos de atención se actualicen en la queja indicada y el estado pase de **EN ATENCION** a **ATENDIDO** por lo que ya no será seleccionada por ningún otro empleado de la misma categoría.
Esta acción se identificará por el código **ACTUALIZAR_TICKET**.
5. Liberar un ticket pues no se puede resolver: en este caso el cliente envía el número del ticket que anteriormente solicitó para darle atención, lo que provocó que estuviera en estado **EN ATENCION** pero por alguna razón se debe devolver a estado **PENDIENTE** para permitir que otro encargado la tramite.
Esta acción se identificará por el código **LIBERAR_TICKET**.
6. Solicitar reporte de atención, en el cual el servidor recibirá el login del empleado y una fecha específica y retornará la información necesaria al empleado conectado sobre su tasa de atención para la fecha indicada, esto es,

de la cantidad de tickets recibidos en esta fecha cuál la cantidad de tickets atendidos en forma satisfactoria por él (en cantidad y proporción, es decir si la totalidad de tickets recibidos en una fecha fueron 20 y él atendió satisfactoriamente 5, debería reportar los valores 20 recibidos, 5 atendidos, 25% de proporción). Esta acción se identificará como **REPORTE_TICKET**.

7. Desconectar empleado: En este servicio el servidor localiza el id del empleado indicado y procede a modificar su estado como DESCONECTADO por lo que no podrá hacer uso de los otros servicios del servidor. Esta acción se identificará como **DESCONECTAR_EMPLEADO**.

Cada vez que ocurre un cambio en las quejas cargadas en el servidor deberán refrescarse para tener la información lo más actualizada posible en la pantalla del mismo.

El monitoreo del servidor

La aplicación de atención a tickets al ser iniciada procede a levantar su servidor para permitir que las aplicaciones cliente que se encargarán de procesar los tickets recibidos puedan establecer contacto con él. Por lo tanto, esta aplicación cuenta con una pantalla de monitoreo donde se mostrará constantemente el estado de comunicación del servidor con los clientes conectados y las acciones que éstos están solicitando a cada instante.

El servidor además debe mostrar constantemente el estado gráfico de las listas de trabajo (INBOX, COLA_ROJA, COLA_AMARILLA, COLA_VERDE) y el estado de los tickets contenidos en ellas.

Las consultas en el servidor

Además el servidor puede ofrecer información significativa como:

- ¿Cuál es el tiempo promedio de atención de tickets en general y por cada empleado?
- Listar los tickets atendidas en un rango de fechas sin importar las categorías.
- Listar los tickets atendidos satisfactoriamente por un empleado particular.

Bajar el servidor

Al bajar el servidor y salir de la aplicación, deberá quedar en un archivo de Excel la información de cada una de las colas (ROJA, AMARILLA, VERDE) con los tickets tramitados al momento y en INBOX.

Cada uno de ellos deberá quedar en una pestaña distinta del archivo de Excel donde cada pestaña se llame ROJA, AMARILLA, VERDE, INBOX.

El nombre del archivo que se guarda sería TICKETS_Fecha.xls Ejemplo TICKETS_30042015.xls

La aplicación cliente

En el caso de la aplicación cliente, ésta inicia indicando el nombre del usuario que se conecta y cuenta con un selector de color (ROJO, AMARILLO, VERDE) que le permite indicar el tipo de queja que se dispone a atender.

Cuenta además con los espacios necesarios para indicar el host y el puerto al que se debe conectar para establecer comunicación con la aplicación servidora.

El encargado de operar la aplicación cliente tiene como acciones disponibles las siguientes:

1. Conectarse al servidor: Solicita el servicio **CONECTAR_EMPLEADO** enviado su login y contraseña y si son correctas según el servidor, la aplicación cliente ingresa al modo de sesión de trabajo indicando en pantalla su nombre completo, pudiendo marcar en un selector de color, la categoría actual de trabajo.
2. Solicitar la lista de tickets pendientes de atención según el color definido: esto provoca que se conecte al servidor por medio de la operación **SOLICITAR_TICKETS** y visualizar la lista de tickets con estado **PENDIENTE** en el servidor, las cuales se muestran en una tabla visual en pantalla.
3. Al seleccionar un ticket específico los datos del mismo se cargan en un espacio en la pantalla diseñado para este efecto.
4. El cliente tiene en pantalla un reloj o cronómetro que se inicializa en un cierto momento, se puede pausar y reanudar en cualquier momento.

5. Cuando decide comenzar a atender un ticket específico, el cliente envía una petición **RESERVAR_TICKET** al servidor para que lo marque como **EN ATENCION**. Si el servidor reporta éxito en la solicitud el cronómetro comienza a transcurrir midiendo el tiempo invertido en la atención de la misma. El cronómetro se pausa y se reanuda las veces que así lo requiera y el tiempo sigue acumulándose.
6. Si el cliente resuelve el ticket actual, envía los datos de la resolución por medio de la acción **ACTUALIZAR_TICKETS** en la que envía todos los datos necesarios para que el servidor pueda marcarlo como **ATENDIDO**.
7. Si decide que no puede resolverla puede solicitar la acción de **LIBERAR_TICKET** pero igual registra el tiempo dedicado aunque no haya sido resuelta y además envía una razón del porqué lo abandona sin éxito, mensaje que se mantiene en el comentario de resolución como un log de trámite de la misma.
8. Cada cierto tiempo y sin que el encargado lo solicite por medio de una acción la aplicación cliente deberá solicitar al servidor refrescar la lista de tickets del color actualmente atendido con el objetivo de poder visualizar los cambios que están realizando otros clientes conectados. Esto lo hace por medio de la acción **SOLICITAR_TICKET** que le devuelve la lista de quejas pendientes de manera que si en pantalla tenía una pendiente y ya fue atendida debería notarse el cambio en el estado y no debería permitirse su selección. Además debe mantener visible el ticket actualmente tramitado.
9. El usuario conectado puede solicitar al servidor un estado de su trabajo mediante el servicio **REPORTE_TICKET** en el cual por medio de su login y una fecha puede conocer su proporción de efectividad de acuerdo a la cantidad de tickets atendidos satisfactoriamente.
10. El usuario puede solicitar el abandono de la sesión por medio del servicio **DESCONECTAR_USUARIO**.

4. Puntos a ser evaluados

1. Correctitud y completitud de la solución computacional.
2. Deberá contar con una arquitectura de capas.
3. Debe ser implementado con una interfaz gráfica sumamente amigable y sencilla de utilizar tanto por el administrador del servidor como por el cliente.
4. Entregar un documento formal en formato digital con los siguientes apartados:
 - a) Esta especificación.
 - b) Deberá aportar la imagen completa del diagrama de clases utilizado para la implementación de toda la aplicación. **NO DEBERÁ** obviar ningún atributo ni método salvo los métodos set/get y toString ().Esta imagen deberá ser agregada como anexo a esta especificación. Si lo desea puede aportar el archivo original con la que fue construida.
 - c) Deberá aportar los diagramas de actividad y estado asociados al comportamiento de ambas aplicaciones.
 - d) Deberá entregar una tabla donde enumere todos los requisitos en este documento expuesto, su nivel de implementación y funcionalidad correcta, y en caso de que estos porcentajes no sean del 100% deberá justificar las razones del porqué no se implementaron o los problemas puntuales que se encontraron en su implementación. Esta tabla deberá ser agregada como anexo a esta especificación.
 - e) Deberá entregar en forma IMPRESA un “**brochure**” donde se muestren las bondades y funcionamiento a nivel de usuario de la aplicación. Este documento puede tener la forma de un tríptico donde se presente información de la proyecto, la aplicación y sus creadores, el desglose de funcionalidades con las que cuenta y muestras de los resultados que se pueden obtener con la aplicación.
 - f) Debe estar totalmente documentado a nivel interno en la parte de la definición de las clases, atributos y métodos, con el fin de aportar el Java Doc como parte de la documentación técnica.

Consideraciones administrativas

1. El proyecto está planteado para tener un tiempo de implementación de cuatro semanas por lo que de acuerdo al calendario estaría programado para el jueves el 28 de mayo (final de semana 16).
2. Las revisiones serán jueves y viernes de semana 16, se coordinará horario de revisión pues deberá ser en algún laboratorio.
3. Los detalles internos serán revisados por parte de la profesora.
4. Puede ser trabajado en grupos de 4 personas como máximo sin posibilidad de ampliar los grupos. **Favor no insistir!**
5. Debe entregar 2 carpetas llamadas: documentación y solución computacional, en la primera deberá incluir la documentación solicitada y en la segunda los archivos y/o carpetas necesarias para la implementación de esta tarea. El archivo comprimido debe llevar el nombre de los dos integrantes y la leyenda Proyecto2_POO_Nombre1_Nombre2_Nombre3_Nombre4.
6. El lenguaje de programación a utilizar es Java, y pueden utilizar la herramienta de desarrollo que deseen.