



Universidad del Valle

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Programación Orientada a Eventos
MINIPROYECTO - 3

Fecha de entrega: 4 de diciembre de 2023



Debido a la dificultad que tienen los estudiantes para responder los exámenes de manera individual, se ha planteado la idea de crear una aplicación que permita resolver un examen de manera colaborativa. Los estudiantes desarrollan un examen en grupos de tres integrantes. La aplicación consta de un servidor y varios clientes que interactúan con él para resolver las preguntas.

El servidor:

La aplicación servidor debe permitir a través de una GUI realizar las siguientes operaciones:

- **Creación de exámenes:** Un examen contiene un nombre, una lista de preguntas y un tiempo de duración (minutos y segundos). Las preguntas se deben cargar desde un solo archivo de preguntas del examen (puede utilizar para ello JFileChooser), todas las preguntas son de selección múltiple con cuatro opciones de respuesta. Se debe poder crear todos los exámenes que se desee.
- **Visualizar los exámenes:** debe permitir seleccionar un examen y ver las preguntas que hay en él y el tiempo de duración
- **Visualizar informes de exámenes realizados:** debe permitir seleccionar un informe y ver su contenido

El comportamiento del Servidor para la realización de exámenes debe ser el siguiente:

El servidor debe estar a la espera de que se conecten los tres integrantes del grupo para realizar el examen, cuando se conecte cada integrante debe mostrarse que se conectó y debe aparecer visible todo el tiempo del examen en la GUI (puede hacer uso label de colores que represente cada usuario conectado o imágenes). Si alguno se llega a desconectar también debe de reflejarse en la GUI.

Cuando los tres integrantes estén conectados se puede iniciar el examen. (debe asegurarse que el examen no se pueda iniciar si no están todos sus integrantes conectados).

Para iniciar el examen el usuario debe seleccionar uno de los exámenes creados en el sistema y seleccionar la opción de iniciar.

El servidor debe compartir las preguntas con cada estudiante (cliente) miembro del grupo y llevar la hora para saber cuánto tiempo falta para terminar el examen.

Cada pregunta tiene tres estados ("Libre", "Ocupada", "Respondida"), por defecto el estado es libre. Cuando un cliente selecciona una pregunta para resolver, el servidor debe cambiar el estado a ocupada. Una pregunta solo puede ser respondida por un estudiante a la vez.

Cuando el cliente resuelva una pregunta, cambia a estado respondida, y el servidor debe calificarla (la nota del examen es de 0.0 a 5.0 y cada pregunta tiene el mismo valor). El servidor mantiene un registro de las preguntas correctas, incorrectas y quien la resolvió.

El examen termina por cualquiera de estas dos causas:

- Porque se termine el tiempo del examen, o
- Porque los estudiantes hayan terminado todas las preguntas del examen.

Cuando termine el examen el servidor debe enviar el informe del examen realizado a todos los estudiantes, y almacenar el informe en un archivo de texto.

Cuando se arranque la aplicación servidor de manera automática debe cargar desde archivo la información de los informes de los exámenes realizados previamente.

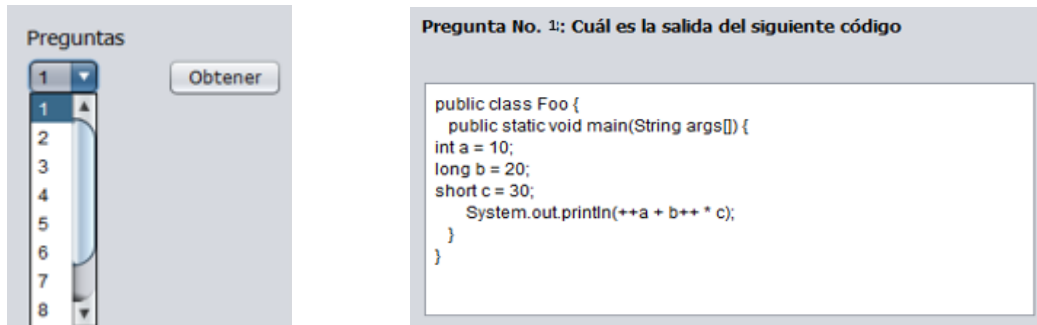
El cliente (estudiante):

El cliente debe tener una GUI que le permita:

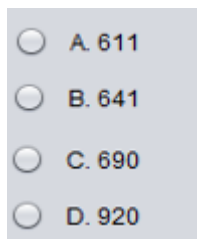
- Visualizar el número de preguntas del examen
- seleccionar una pregunta y visualizar su enunciado y sus opciones de respuesta
- visualizar el tiempo que resta para finalizar el examen
- Tener las opciones de responder o cancelar

Comportamiento del cliente durante el examen:

Cuando inicia el examen el servidor envía a todos los clientes las preguntas del examen que deben resolver y el cliente las debe visualizar en su GUI en un componente que permita seleccionar alguna de ellas (puede ser una lista) y pueda ver su enunciado, **por ejemplo**:



El estudiante (cliente) debe seleccionar de la lista la pregunta que quiere resolver y le informa al servidor que esa es la pregunta que va a resolver (**en el ejemplo**, haciendo clic en el botón "Obtener"). Si el estado de la pregunta es "Libre" el servidor envía un mensaje al cliente con las opciones de respuesta de la pregunta a resolver. El cliente actualiza la GUI con el contenido de la pregunta seleccionada:



El servidor actualiza el estado de la pregunta a "ocupada" y envía un mensaje a todos los clientes para que actualicen la lista de preguntas disponibles. El listado de preguntas se va modificando a medida que se van resolviendo cada pregunta (puede deshabilitar la pregunta de la lista, puede colocarle un mensaje ocupada o resuelta, cambiarle de color o eliminarla de la lista).

Cuando el estudiante termine de resolver la pregunta debe enviar la respuesta al servidor. El servidor recibe la respuesta enviada por el estudiante, cambia el estado de la pregunta a respondida, revisa, califica la pregunta, y avisa a todos que la pregunta fue resuelta para que modifiquen su lista de preguntas disponibles. Una vez se resuelva la pregunta el cliente puede seleccionar otra pregunta para resolver.

El cliente después de obtener una pregunta tiene la posibilidad de cancelarla, el servidor actualiza el estado de la pregunta a "libre" e informa a todos los clientes para que actualicen la lista de preguntas disponibles.

La forma que tiene el usuario para seleccionar una opción de respuesta es hacer clic sobre una opción de respuesta o digitando la letra correspondiente a la opción (haga uso de `KeyListener`). Las opciones de enviar, obtener y cancelar también deben funcionar dando clic sobre ellas y con atajos de teclado de la siguiente manera:

- para accionar la opción de responder oprima `ALT+R`
- para accionar la opción de cancelar oprimir `ALT+C` y
- para accionar la opción de obtener una pregunta debe oprimir la tecla `O` (mayúscula o minúscula)

El cliente también debe visualizar la cantidad de minutos restantes para finalizar el examen. El servidor le envía cuando inicia el examen ese tiempo y el cliente lo debe iniciar y visualizar. Cuando desde el servidor se termine el tiempo le avisará a los clientes y se terminará el examen.

Cuando se termine el examen el servidor debe enviar un informe a todos los clientes con la siguiente información: cada pregunta, la opción de respuesta escogida por el cliente, la respuesta correcta. Un mensaje indicando si es correcta o incorrecta, quién la resolvió y la nota obtenida en el examen.

REQUISITOS DE LA IMPLEMENTACIÓN:

La aplicación es cliente/servidor, se debe diseñar bajo el enfoque Orientado a Objetos utilizando MVC. El diseño de la interfaz, la selección de preguntas y las imágenes o componentes usados en la GUI son libres a criterio de los programadores

Debe haber un solo archivo de preguntas por examen, el cual debe ser un archivo de texto cuya estructura debe ser definida por los programadores.

ENTREGABLES:

La aplicación deberá de cumplir con la funcionalidad descrita anteriormente.

Las interfaces gráficas de usuario (GUI) deben ser construidas completamente a través de código Java teniendo en cuenta las recomendaciones hechas en clase, no se podrán construir a partir del uso de asistentes de construcción de GUI automático

PARA EL DESARROLLO DE ESTE MINIPROYECTO DEBERÁN TRABAJAR EN GRUPOS DE MÍNIMO 3 PERSONAS Y MÁXIMO 4

FECHA LIMITE DE ENTREGA: LUNES 4 DE DICIEMBRE 2023

PONDERACIÓN EN EL PORCENTAJE DE EVALUACIÓN: 25%

ENTREGABLES:

Como resultado del desarrollo del proyecto deberá entregar:

- Un archivo en formato .pdf donde se muestre el boceto de la GUI a desarrollar, donde se especifique además los Layout utilizados en cada panel
- El diagrama de clases de la aplicación
- Dos videos: uno de máximo 10 minutos donde muestren el funcionamiento de la aplicación y otro de máximo 15 minutos donde expliquen el código de su proyecto.
- El proyecto comprimido (.rar o .zip) del proyecto en Netbeans

¡Éxitos!