



Universidad del Valle

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Programación Interactiva

Proyecto

Fecha de entrega: Ver en el campus virtual



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 pueden desarrollar un examen en grupos hasta de cuatro integrantes. La aplicación consta de un servidor y varios clientes que interactúan con éste para resolver las preguntas.

El servidor:

En el servidor se debe poder configurar un examen que van a resolver los estudiantes. poder iniciar un examen, tener comunicación constante con los clientes que van a resolver el examen y finalmente mostrar información sobre los exámenes presentados. Para ello:

- Debe permitir que se cargue de un archivo de texto las preguntas que debe resolver el grupo de estudiantes. Para seleccionar el archivo a cargar, debe utilizar un JFileChooser.
- Compartir las preguntas con cada estudiante (cliente) miembro del grupo.
- Llevar la hora y enviar constantemente a los clientes el tiempo que falta para terminar el examen.
- El servidor debe permitir la definición de la duración del 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 y notificar a los otros clientes que esa pregunta está ocupada. Y cuando el cliente resuelva una

pregunta, cambia a estado respondida. Una pregunta solo puede ser editada por un estudiante al tiempo.

- Cuando el cliente resuelva una pregunta, cambia a estado respondida, y el servidor debe calificarla. El servidor mantiene un registro de las preguntas correctas, incorrectas y quien la resolvió.
- Al finalizar un examen el servidor envía a cada uno de los estudiantes que realizó el examen el informe con la nota, cada pregunta especificando para ellas quien la respondió si es correcta o incorrecta
- En el servidor se debe tener la opción de visualizar la información de los exámenes respondidos anteriormente. Esta información se debe cargar automáticamente cuando se abre la aplicación y se debe almacenar en archivo cada que se cierra la aplicación.

El cliente - (estudiante):

En la figura 1 se muestra una aproximación de lo que debe ser la GUI del cliente.

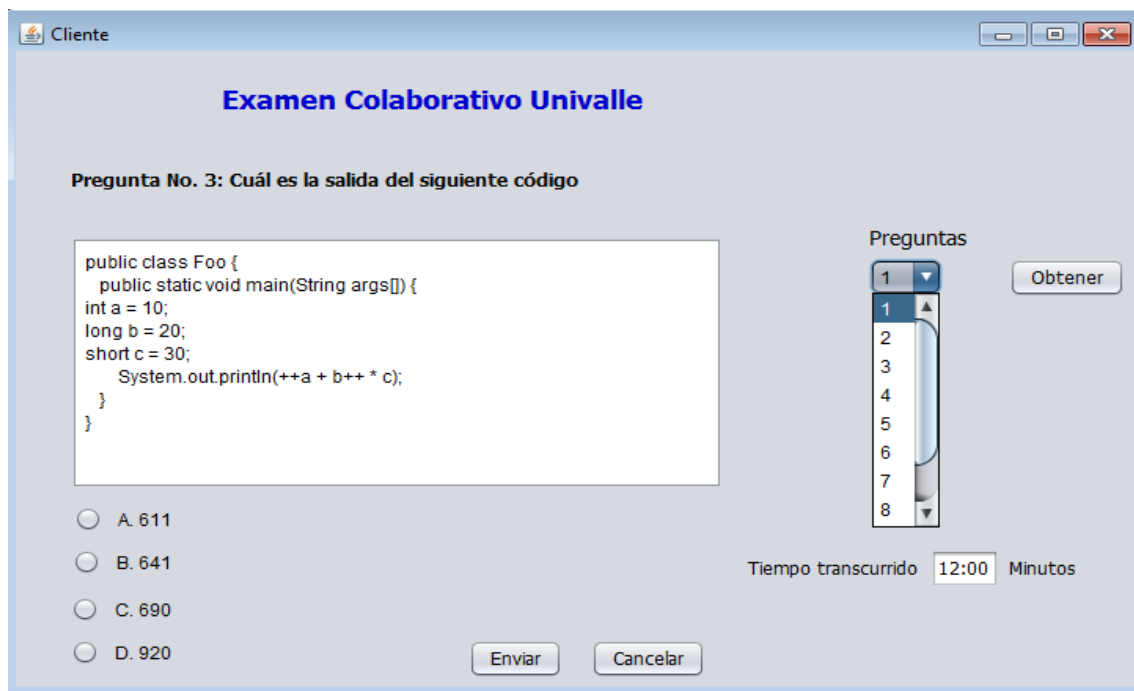


Figura 1. Interfaz gráfica del cliente

- Para comenzar a resolver un examen el cliente debe esperar a que el servidor dé inicio al mismo. Cuando el servidor dé inicio, en el cliente se debe desplegar la lista de preguntas a resolver.

- El estudiante (cliente) debe seleccionar de la lista la pregunta que quiere resolver y hacer clic en el botón "Obtener".
- El cliente le envía la solicitud al servidor y este la procesa. Si el estado de la pregunta es "Libre" envía un mensaje al cliente con la pregunta a resolver.
- En el cliente se 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.
- En el cliente el listado de preguntas se va modificando a medida que se van resolviendo las preguntas. (las respondidas se deben desaparecer del listado o deshabilitar del listado)
- Cuando el estudiante termine de resolver la pregunta debe hacer clic en el botón "enviar". El servidor recibe la respuesta enviada por el estudiante, cambia el estado de la pregunta a respondida, revisa y califica la pregunta.
- Una vez se resuelva la pregunta el cliente puede seleccionar otra pregunta para resolver. Si el cliente hace clic en "cancelar", el servidor actualiza el estado de la pregunta a "libre" e informa a todos los clientes para que actualicen la lista de preguntas disponibles.
- El cliente también contiene un campo que muestra la cantidad de minutos restantes para finalizar el examen. El servidor debe actualizar frecuentemente este reloj e informar a los clientes para que tengan el reloj actualizado.
- Cuando se termine de resolver todas las preguntas el servidor debe enviar un informe a todos los clientes con la siguiente información: Un listado con cada pregunta, si fue calificada correcta o incorrectamente y quién la resolvió. Finalmente debe tener la nota obtenida por todo el grupo. En la GUI del cliente debe tener la forma de visualizar ese informe.

Usted debe diseñar una GUI para el cliente y otra para el servidor, El diseño de la GUI queda a libertad de los desarrolladores, pero en ellas se deben poder visualizar todas las opciones descritas.

Porcentajes y Entregables:

Se debe entregar como solución al proyecto lo siguiente:

[20%] Diagrama de clases del modelado de la solución del problema.

[80%] Implementación de la solución (proyectos de Netbeans del cliente y el servidor)

[0-1] Sustentación con nota factor, sobre la nota obtenida en el proyecto. Es decir:

La nota del proyecto dependerá de lo implementado + modelado y de la sustentación, la cual será una nota factor que estará en el rango real [0 - 1]. Es decir, la nota que haya sacado en la implementación+modelamiento se multiplica por el factor sacado en la sustentación. Ejemplos:

1. Si saca 4.5 en implementación + modelamiento y saca 1 en sustentación la nota final del proyecto será:

$$4.5 * 1 = 4.5$$

2. Si saca 4.5 en implementación + modelamiento y saca 0.4 en sustentación la nota final del proyecto será:

$$4.5 * 0.4 = 1.8$$

3. Si saca 4.5 en implementación + modelamiento y saca 0 en sustentación la nota final del proyecto será:

$$4.5 * 0 = 0$$

¡Éxitos!