Universidad Andrés Bello
Conectar · Innovar · Liderar

Creación de una evaluación académica mediante carga de archivos: Programación orientada a eventos en Java

Integrantes:
Joaquín Fuenzalida C.
Benjamín Vallejos V.

Profesor:
Juan Calderón M.

Introducción

El presente informe muestra la creación de un programa mediante el lenguaje de Java, donde se le concede al usuario la creación de una prueba mediante la carga de archivos “CSV” en el programa. Si el usuario carga el archivo, este último podrá realizar la evaluación que consta de preguntas de verdadero/falso y opción múltiple encasilladas cada una con algún nivel de la taxonomía de Bloom. Si se finaliza la prueba, el usuario podrá ver los porcentajes de acierto, habiendo dos de este tipo, uno el cual muestra el porcentaje de respuestas correctas según la taxonomía de Bloom a la que pertenece y otra según el ítem al cual pertenezca.
La motivación detrás de la realización de este proyecto consiste en la digitalización de una evaluación buscando la facilitación y automatización en cuanto a la revisión de pruebas se refiere, ahorrando un tiempo significante a nuestro cliente.
El informe se encuentra estructurado de la siguiente manera:
Objetivo general:
-Desarrollar una aplicación mediante Java utilizando el framework Swing para la creación y administración de evaluaciones, dando a conocer los porcentajes de rendimiento del usuario respecto a la taxonomía de Bloom asociada a las preguntas.

Objetivos específicos:
-Implementar un sistema de carga de archivos desde un directorio, permitiendo su visualización y corrección en la interfaz gráfica
-Diseñar una interfaz gráfica intuitiva que permita la fácil navegación entre las preguntas y en los distintos paneles
-Entregar al usuario al finalizar la prueba, un resumen de sus respuestas ingresadas, indicando los porcentajes de respuestas correctas según su taxonomía y por tipo (alternativas/verdadero y falso).

Descripción de la solución

Para la realización de este programa, se crearon las siguientes clases:

1. **Evaluación** *(dentro de package “backend”):* Se crea la clase Evaluacion, esencial para la gestión y procesamiento de las colecciones de preguntas (alternativas/verdadero y falso), principalmente al cargarlas desde un archivo .csv
2. **Pregunta** *(dentro de package “backend”)*: Se crea la clase padre Pregunta, esta contiene los atributos de: enunciado, bloom, tiempo estimado. Estos últimos serán heredados a las clases “PreguntaAlt” y “PreguntaVoF”. Esta clase permite un manejo más sencillo de las preguntas en conjunto.
3. **PreguntaAlt** *(dentro de package “backend”)*: Se crea la clase PreguntaAlt, que hereda de Pregunta. Representa las preguntas de selección múltiple, teniendo de atributo una lista de alternativas y otro atributo llamado respuestaAlt, la que se encarga de almacenar la respuesta ingresada por el usuario de tipo alternativa.
4. **PreguntaVoF** *(dentro de package “backend”)*: Se crea la clase PreguntaVoF, que hereda de Pregunta. Representa las preguntas de verdadero o falso, teniendo de atributos respuestaVF lo que significa la respuesta ingresada por el usuario (de tipo String) y respuestaCorrectaVF, la que se encarga de almacenar la respuesta correcta de la prueba en el formato “CSV”.

5. **TableroVisual** *(dentro de package "frontend")*: Se crea la clase principal para la interfaz gráfica del programa. Permite la carga de archivos, la realización de las preguntas. la corrección de estas mismas y los resultados finales calculados en porcentaje, mediante la asociación de botones a estos eventos.
6. **Main** *(dentro de carpeta src)*: Permite que el tablero visual se inicialice y sea visible.

Detalles del modelo:

- (explicar contenido/métodos)
- **Métodos ubicados en Tablero Visual:**
    - **cargarPreguntasEnPanel()** : Método que muestra las preguntas de alternativas y verdadero/falso en el JPanel llamado "muestraPregunta" y crea dinámicamente los componentes de la interfaz de usuario para cada pregunta (jRadioButton's).
    - **mostrarPreguntasConRespuestasCorrectas()**: Método que muestra las preguntas de alternativas y verdadero/falso con su respuesta correcta en el JPanel "revisiónPrueba", además de los porcentajes según taxonomía de Bloom y por tipo de pregunta.
    - **mostrarPreguntaActual()**: Método que utiliza cardLayout para mostrar las preguntas como tipo "tarjeta", es decir, una pregunta por una e imprime un mensaje en consola cual es la pregunta en la que el usuario se encuentra.
    - **setListeners()**: Método en el que se encuentran almacenados los listeners de los botones, jLabel's.

- **Botones ubicados en setListeners():**
    - **cargarArchivo:** Botón con la función de subir un archivo al programa mediante la búsqueda por directorio, solo acepta la carga del archivo que sea formato "CSV", si este es subido, mostrará la cantidad de preguntas cargadas y el tiempo estimado de realización de prueba. En el caso de cancelar la selección muestra el aviso "Selección de archivo cancelada.". Si se selecciona un archivo que no es de formato "CSV", muestra lo siguiente "Error: Seleccione un archivo CSV válido". Por otro lado, si ocurre algún error, desplegará el siguiente mensaje "Ocurrió un error en la selección del archivo."
    - **inicioPrueba:** Botón que cumple con la función de inicializar la prueba, sólamente si ya se tiene cargado el archivo "CSV" y entrega un aviso en el terminal de que el botón está siendo presionado.
    - **finPrueba:** Botón que cumple con la función de dar término a la prueba, ya teniendo las respuestas almacenadas y permite cambiar al siguiente JPanel la cual entrega un resumen detallado de los resultados respecto a la evaluación.
    - **anterior:** Botón que cumple con la función de poder retroceder a la pregunta anterior.
    - **siguiente:** Botón que cumple la función de poder pasar a la siguiente pregunta (cambio de panel)
- **Métodos de clase Evaluacion:**
    - **preguntasCsv:** Método que cumple la función de leer el archivo "CSV" de acuerdo al formato ya establecido por nosotros.
    - **parsearLineaCsv:** Método que sirve para dividir una línea de texto en columnas, en este caso, al estar nuestro archivo separado por " ; ", respetando todo el contenido que están dentro de comillas

- **getTotalPreguntas:** “getter” que devuelve el total de preguntas almacenadas.
- **obtenerEstadisticasPorBloom:** Método que calcula los porcentajes de las respuestas según su taxonomía de Bloom (Conocimiento, Comprensión, Aplicación, Análisis, Síntesis, Evaluación).
- **obtenerEstadisticasPorTipo:** Método que calcula los porcentajes de respuestas correctas ingresadas por el usuario según el tipo de pregunta (Verdadero/Falso o Alternativas).

Conclusión

La realización de este trabajo nos permitió aplicar nuestros conocimientos aprendidos a lo largo del año, como también el conocimiento adquirido mediante la investigación en internet relacionados al lenguaje de Java.

Habiendo finalizado el código, se puede decir que se han cumplidos todos los objetivos, dándole al usuario la opción de poder cargar archivos “CSV” al programa, la realización de una evaluación de fácil navegación entre estas y al finalizar esta misma, dar los resultados en porcentaje de cada nivel de taxonomía de bloom y la respuesta correcta de cada pregunta.

Este trabajo nos ayudó a poder comprender de mejor manera el funcionamiento del lenguaje de Java y su biblioteca Swing para la creación de interfaces gráficas, entendiendo y aplicando la orientación a eventos y la modelación de una o más GUI.