



Minutas del proyecto

Lógica:

- Decidimos implementar la clase $\text{Pair}\langle K, V \rangle$.
- Decidimos manejar las excepciones dentro de la clase Programa.
- Para calcular la nota mínima y ordenar las notas de mayor a menor, decidimos implementar nuestros propios comparadores $\text{MinComparator}\langle \text{Integer} \rangle$ que compara los enteros de menor a mayor y $\text{MaxComparator}\langle \text{Integer} \rangle$ que compara los enteros de mayor a menor, ambos redefinen el método `compare` de la clase `Comparator`.
- Cuando un alumno ya existe en el registro, si se intenta volver a agregarlo con una nota diferente, no hay modificaciones en el registro. Es decir, si se quiere modificar la nota de un alumno hay que eliminarlo y volverlo a agregar.

GUI:

- Siempre se puede aceptar presionando Enter.
- El diseño no es adaptable, es decir, si se modifica el tamaño de las ventanas puede afectar a la visualización de la aplicación.
- Cada funcionalidad abre una ventana nueva, pero si se quieren utilizar varias a la vez los cambios no se actualizan.
- Se puede visualizar una materia a la vez, es decir, si desea cambiar de materia debe cerrar y volver a abrir la aplicación.
- Al cerrar la aplicación los datos ingresados se borran.
- Si se cierra el menú de opciones se cierran todas las ventanas.
- Referido al nombre de la materia, sólo se corrobora que no sea vacío.
- El usuario sólo puede ingresar números en los campos referidos al LU o las notas.
- Cuando el usuario deja algún campo vacío pero hace click en el botón aceptar, se chequea mediante el método `isBlank()` de la clase `String` y se muestra un cartel de aviso.
- Cuando no hay alumnos en el registro, las opciones de visualizar la nota mínima y el promedio, muestran un cartel de aviso pero igualmente generan el panel vacío.
- Las funcionalidades ver los aprobados y ver los desaprobados sólo muestran el LU de los alumnos, no sus notas.
- La funcionalidad ver nota mínima no muestra el alumno que obtuvo la nota mínima.