PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Entrega Parte A

**Programación orientada a objetos**

**Francisco Andrés Muñoz Alarcón 20.242.456-2**

**Anais Monserrat Foix Monardes 20.834.761-6**

**Sofía Carolina Menzel Madrid 20.478.743-3**

**2022**

**Índice**

Contenido

[**A.1-**  **Se debe hacer un menú para el sistema con las siguientes funcionalidades: (1) Actualizar/Modicar del elemento (Update); (2) Eliminar (Delete) del elemento. Esto es para cada una de las 2 colecciones anidadas. opcional: hacer la funcionalidad de Buscar (Search) elemento en 1 o más niveles.** 3](#_Toc102587250)

[**A.2-**  **Se debe generar un reporte en archivo txt que considere mostrar datos de las 2 colecciones anidadas (ej: CSV). opcional: Se puede generar un archivo de salida de plantilla de cálculo (.xls o xlsx), opcional: Se puede utilizar un componente gráfico estadístico (JFreeChart u otro) en ventana.** 5](#_Toc102587259)

[**A.3-**  **El código fuente debe estar bien modularizado de acuerdo con lo descrito en el informe además de seguir las buenas prácticas de documentación interna y legibilidad.** 6](#_Toc102587260)

[**A.4-**  **Todas las funcionalidades pueden ser implementadas mediante consola. Opcional: La implementación de una o más interfaces gráficas de usuario, con la inclusión de ventanas con componentes AWT, SWING, JAVAFX u otro.** 6](#_Toc102587262)

[**A.5-**  **Utilización de GitHub (realización de al menos 3 a 4 commit)** 6](#_Toc102587264)

**A.1- Se debe hacer un menú para el sistema con las siguientes funcionalidades: (1) Actualizar/Modicar del elemento (Update); (2) Eliminar (Delete) del elemento. Esto es para cada una de las 2 colecciones anidadas. opcional: hacer la funcionalidad de Buscar (Search) elemento en 1 o más niveles.**

Se adjunta imagen del menú, en la cual se ven las opciones:

* Agregar elemento (alumno, profesor, curso, oferta laboral).
* Mostrar elemento (listado de alumnos, profesores, cursos,ofertas laborales).
* Actualizar elemento (estudiante, profesor, curso, oferta laboral).
* Borrar elemento (estudiante, profesor, curso, oferta laboral).
* Generar reporte de curso.
* Terminar programa.

**Tabla

Descripción generada automáticamente**

**A.2- Se debe generar un reporte en archivo txt que considere mostrar datos de las 2 colecciones anidadas (ej: CSV). opcional: Se puede generar un archivo de salida de plantilla de cálculo (.xls o xlsx), opcional: Se puede utilizar un componente gráfico estadístico (JFreeChart u otro) en ventana.**

Se adjunta imagen en la cual se ve en el código que se imprimen los cursos, se elige el curso y se crea un archivo txt de este con sus datos correspondientes.**![Texto

Descripción generada automáticamente]()**

**A.3- El código fuente debe estar bien modularizado de acuerdo con lo descrito en el informe además de seguir las buenas prácticas de documentación interna y legibilidad.**

El código fuente sigue todas las buenas prácticas, siendo modularizado, explícito en todo su contexto y legible, de forma que cuando se verifique el programa, se comprenda sin dificultad alguna, ya que presenta una arquitectura ordenada, metódico, perseverante, aplicado.

**A.4- Todas las funcionalidades pueden ser implementadas mediante consola. Opcional: La implementación de una o más interfaces gráficas de usuario, con la inclusión de ventanas con componentes AWT, SWING, JAVAFX u otro.**

**Texto, Carta

Descripción generada automáticamente**

**A.5- Utilización de GitHub (realización de al menos 3 a 4 commit)**

Captura de pantalla de computadora

Descripción generada automáticamente