

Trabajo Práctico Integrador Final Programación I.

Cohorte 2023, Comisión 1A.

Predicción de Resultados de Alumnos.

| Datos de entrada | Proceso/s principal/es | Proceso/s secundario/s | Salida |
|--|---|---|--|
| Se imprime por pantalla un mensaje pidiéndole al usuario que ingrese cuantos alumnos desea cargar. | El número ingresado por el usuario se lee y se guarda en el entero <code>num_alumnos</code> ; Este entero se utilizará para definir el tamaño del array de la estructura <code>alumnos</code> . | El entero <code>num_alumnos</code> se usará para varias funciones, por ejemplo, en varias declaraciones del tipo <code>for</code> o de tipo <code>while</code> . | El código procede hacia el menú de opciones. |
| Se imprime por pantalla el menú de opciones, el usuario elige la opción que desea usar. | Se lee el número cargado, y se guarda en el entero <code>opción</code> . | Se crea un archivo binario de nombre <code>alumnos.dat</code> . | Dependiendo la opción elegida, se procede con los casos dentro del menú de opciones. |
| <p><u>Caso 1:</u></p> <p>Siempre y cuando <code>opción</code> sea diferente a 6 (caso de salida), se procederá con la carga de los datos el el/los alumno/s. Se utiliza un bucle <code>for</code> para cargar todos los alumnos solicitados.</p> | <p>Se intenta abrir el archivo, si no se pudo abrir, o el archivo no se encontró/ no existe, se rompe el caso. Caso contrario:</p> <p>El usuario carga:</p> <ul style="list-style-type: none"> Nombre del alumno (se guardará en el apartado nombre de la estructura). Si el nombre es mayor a 50 caracteres, se pedirá que ingrese de nuevo hasta que cumpla con el requisito de ser menor o igual a 50 caracteres con un bucle <code>while</code>. DNI y Legajo del alumno, que se guardará en los respectivos enteros dentro de la estructura. El nombre de la materia que el alumno | <p>Se abre el archivo binario en modo "append". Todos los datos cargados mostrados en procesos principales se guardan dentro de la estructura y dentro del archivo binario, para poder usarse luego.</p> <p>Se utiliza un entero llamado <code>datos cargados</code> para indicar que la carga de datos concluyó. Este cambia su valor a 1.</p> | Los datos cargados por el usuario se guardan en el archivo binario. |

| | | | |
|--|--|--|--|
| | <p>está cursando.</p> <ul style="list-style-type: none"> • Faltas, que se guardará dentro de un entero. Si faltas es mayor o igual a 4, se cambia el dato booleano libre a true, para indicar que el alumno quedó libre • Se ingresa las calificaciones de tanto el primer como el segundo parcial del alumno. Dependiendo de las calificaciones obtenidas, ocurre una de tres opciones: <ul style="list-style-type: none"> - Si tanto la calificación de parcial 1 como de parcial 2 es mayor o igual a 6, se considera que el alumno promocionó, y se calcula el entero notaFinal sumando estas dos notas, y dividiendo el resultado entre 2. - Si la calificación de parcial 1, o la de parcial 2 es mayor o igual a 4, y menor o igual a 5, se considera al alumno aprobado pero no promocionado, y el usuario deberá cargar notaFinal que obtuvo el alumno rindiendo el examen final. - Si ninguno de los dos casos anteriores se cumple, se considera al | | |
|--|--|--|--|

| | | | |
|---|---|--|---|
| | <p>alumno desaprobado.</p> <p>Por último, si notaFinal es mayor o igual a 4, y el booleano libre se marcó en falso, se considera que el alumno aprobó la materia, y un nuevo booleano, llamado materiaAprobada se cambia a true, para indicar que aprobó.</p> | | |
| <p><u>Caso 2:</u></p> <p>Si no se selecciona el caso de salida, y se seleccionó la opción 2, se procederá con la baja de los datos cargados solicitados por el usuario.</p> | <p>Si el archivo no se encontró/ no existe, se rompe el caso.</p> <p>Caso contrario:</p> <p>En caso de que datos cargados sea diferente a 1, indica que no hay datos cargados, y no dejará al usuario operar con esta opción.</p> <p>Caso contrario, se abre el archivo en modo lectura y escritura binaria, y se procede:</p> <ol style="list-style-type: none"> 1. Se le pide al usuario que ingrese el número del alumno que desea borrar. Este número se guarda en el entero índice. A este número se le resta una unidad (ya que los índices de los arrays empiezan a contar de 0). 2. Con un for, se recorre el archivo de alumnos. 3. Se borra al alumno del archivo, y del arreglo en memoria. <p>Se cierra el archivo, y se rompe con el caso. Si no se encontró al alumno, muestra un mensaje de error.</p> | <p>El entero índice se vuelve a poner en valor 0, para poder volver a ser usado.</p> | <p>El alumno solicitado es eliminado del array.</p> |

| | | | |
|---|---|--|--|
| <p><u>Caso 3:</u></p> <p>Si no se seleccionó el caso de salida, se procederá con la modificación de los datos cargados.</p> | <p>Si el archivo no se encontró/ no existe, se rompe el caso. Caso contrario:</p> <p>Al igual que en el caso anterior, se verifica que haya datos cargados en primer lugar. Si no los hay, se rompe el caso. Si los hay, se abre el archivo binario en modo lectura y escritura binaria, y se procede:</p> <ol style="list-style-type: none"> 1. Con el entero índice, se procede de igual manera que en el caso anterior. 2. Con índice ya cargado, se posiciona en el lugar requerido del array. 3. Con un nuevo entero, llamado modificación, se abre otro menú de opciones. <p>Dependiendo de la opción elegida, se procede de manera similar al caso 1, para que el usuario elija que dato modificar del alumno. Con el caso 7 del segundo menú, se sale de este. Además, con un bucle do-while, se le consulta al usuario si desea continuar editando datos (Se continuará hasta que el usuario ya no desee modificar los datos).</p> <p>Se mueve el índice nuevamente a la posición indicada, y se sobrescriben los nuevos datos cargados. Se cierra el archivo y se rompe con el caso.</p> | <p>El entero índice vuelve a valer 0, para poder ser usado de nuevo.</p> | <p>Los nuevos datos son cargados en la posición donde está el alumno solicitado.</p> |
|---|---|--|--|

| | | | |
|---|--|--|---|
| | Si no se encontró el alumno, muestra un mensaje de error. | | |
| <u>Caso 4:</u> Si no se seleccionó el caso de salida, se procede con enlistar todos los datos cargados . | Si no se encontró/ no existe el archivo, se rompe con el caso. Caso contrario: Si no hay datos cargados, se rompe con el caso. Si los hay, se procede: Con un bucle for , se recorre el array para poder alcanzar a cada alumno cargado. Luego se imprime por pantalla los datos ya cargados dentro del archivo binario. | | Se muestran todos los datos cargados dentro del array. |
| <u>Caso 5:</u> | Se verifica tanto que el archivo exista/ se encuentre como que este tenga datos cargados. Luego: Se abre el archivo en modo lectura binaria. Con índice se le pide al usuario que elija el alumno cuyos datos quiera mostrar. Con un bucle for se recorre el array en busca del alumno, se posiciona dentro del índice del alumno, y se leen sus datos. Luego, estos datos se imprimen por pantalla. Si no se encontró el alumno, imprime un mensaje de error. | Índice vuelve a ser 0 para poder ser reutilizado. | Muestra solos los datos del alumno solicitado. |
| <u>Caso 6:</u> Si se selecciona el caso 6, se procede con el cierre del programa . | Se muestra un mensaje de despedida, y se corta el código. | | Se termina el bucle do-while que hace que se repita el menú de opciones siempre y cuando no se pida el caso 6 . |