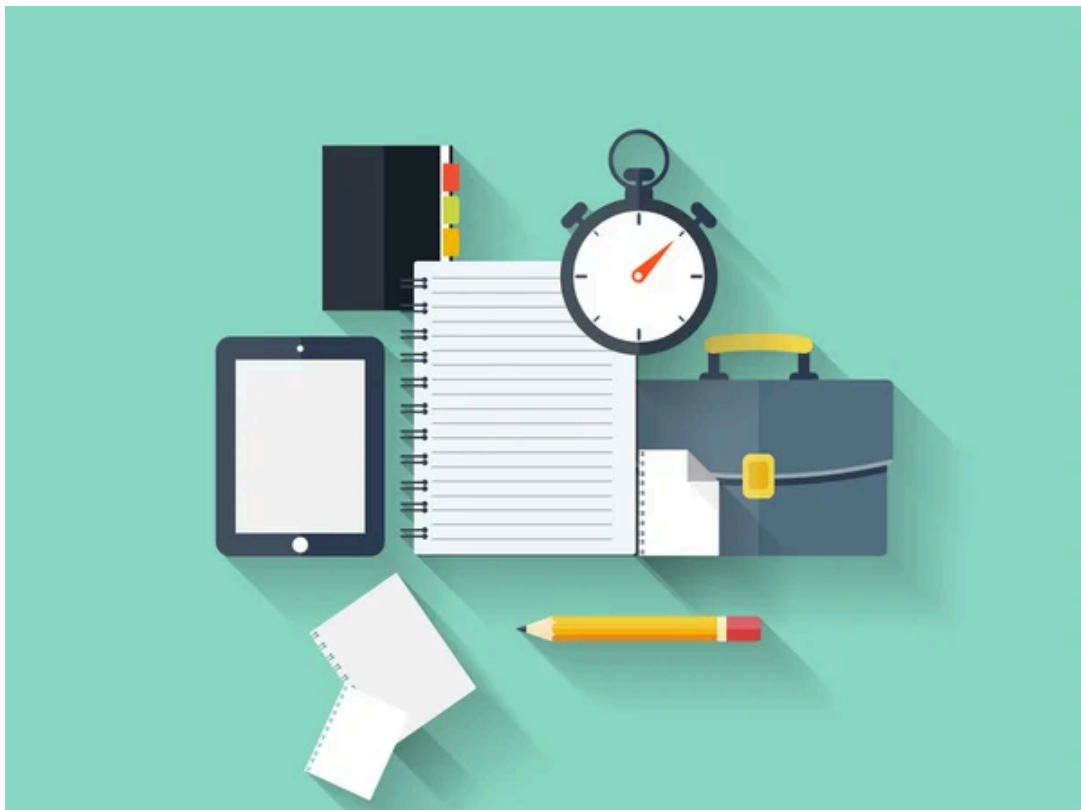


Técnicas de estudio para estudiantes

IPC: Herramienta de estudio.



Tú nueva herramienta de estudio creada por ti mismo

Todas las herramientas, tanto físicas como mentales, que sirven para mejorar los resultados en los estudios es lo que conocemos como **técnicas de estudio**. El objetivo **estas herramientas es ayudarnos y facilitarnos la comprensión y el entendimiento de las materias.**

Instrucciones

Crea un programa que servirá para el estudio y práctica del curso Introducción al pensamiento computacional.

Características

- Pedir información al estudiante: Nombres, apellidos, número de carné y carrera. Verifique que el número de carnet sea valido para estudiantes de primer año en 2022.
- El programa guardara un punteo basado en el número de respuestas correctas por usuario.
- 17 pantallas
 - 1 pantalla inicial donde se le pide los datos estudiante antes de empezar la prueba.
 - 15 pantallas interactivas con preguntas. ~~Cada vez que se haga una nueva prueba las preguntas serán escogidas al azar~~ Respetando que tienen que ser 5 de verdadero/falso, 5 de opción múltiple y 5 de ejemplo de c#.
 - 1 pantalla final para presentar estadísticas y terminar la prueba.
- En la pantalla final se debe mostrar: datos del usuario, número respuestas correctas, número de respuestas equivocadas y punteo final.

El usuario debe ser enviado a la pantalla inicial para que otro usuario pueda usar el programa.

~~Se pide que el programa tenga una librería de 6 preguntas de verdadero o falso, 6 preguntas de opción múltiple y 8 preguntas de código en c#. Las preguntas deben ser creadas por el equipo y deben cubrir todo el material visto en la clase teórica y práctica.~~

Se debe usar programación orientada a objetos.

Tipos de preguntas

- 5 preguntas de falso o verdadero. Cada respuesta correcta agrega 50 puntos.
- 5 preguntas de opción múltiple. Para considerar que una respuesta fue contestada correctamente debe seleccionar todas las opciones correctas. Cada respuesta correcta agrega 100 puntos.
- 5 preguntas con código de ejemplo. Se le presentará al estudiante una porción de código (en c#) que realiza una tarea e imprime a pantalla un resultado. El jugador debe escoger la respuesta correcta de entre 4 opciones. Siempre debe haber una respuesta correcta. Todas las porciones de código debe ser ejemplos creados por él el equipo. Cada respuesta correcta agrega 200 puntos.
 - ~~1 pantalla con ejemplo de una formula matemática~~
 - ~~1 pantalla con ejemplo manejo de cadenas~~
 - ~~1 pantalla con ejemplo de condicionales (if else)~~
 - ~~1 pantalla con ejemplo de ciclos~~

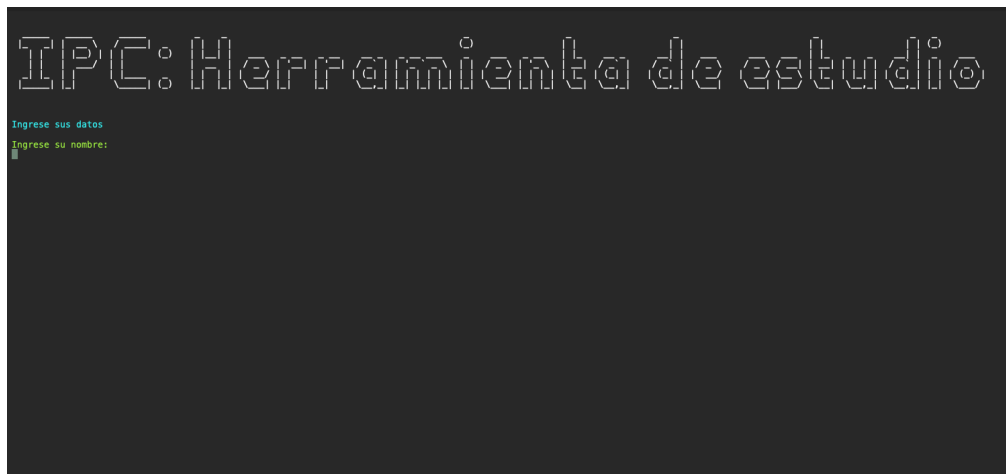
Estructura del programa

- Iniciar prueba
- Listado todos los usuarios que han hecho la prueba. Debe estar ordenado según cómo se fueron haciendo las pruebas (mostrar datos de los usuarios y sus notas).
- Datos estadísticos de las personas que han hecho la prueba.
 1. Usuario con la nota más alta
 2. Usuario con nota más baja
 3. **(Opcional) De la lista total muestre: promedio , mediana y la moda de las notas.**
 4. **(Opcional) Listado de usuarios que tengan menos del 65% de los puntos posibles. Ordenados en orden descendente de otra mayor a menor.**
- ~~Limpiar listado de pruebas.~~
- Créditos: Incluya nombre de integrantes, carnés y carrera.

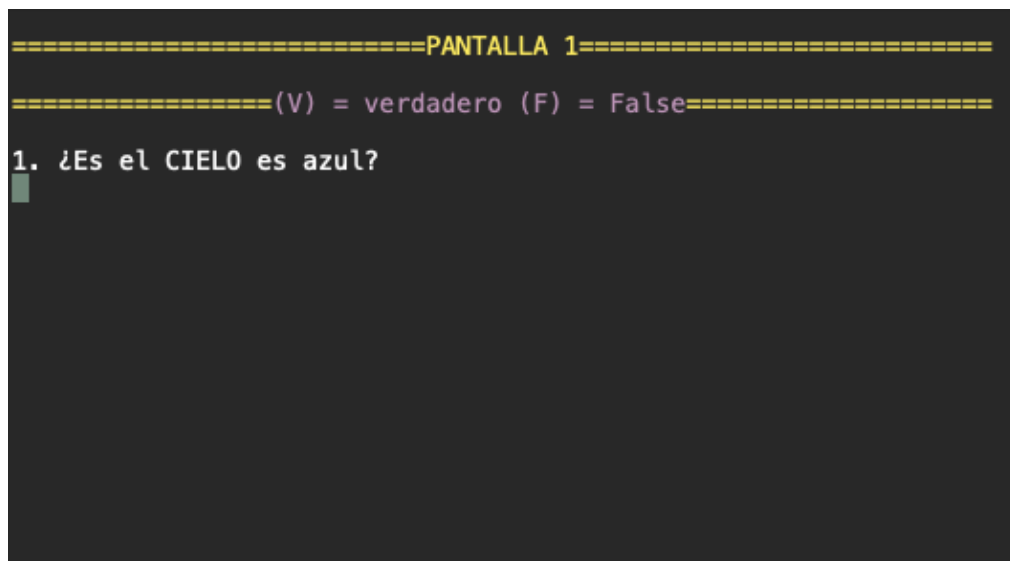
Ejemplo de estructura y orden de las preguntas



Ejemplo pantalla introductoria.



Ejemplo pantalla verdadero o falso



Ejemplo pantalla opción multiple

```
=====PANTALLA 2=====
Opción multiple
Escoga todas las verdaderas
Ejemplo: A,C,D
=====

1. ¿Medios de transporte?

    A. Tren   B. Avion
    C. Casa   D. Carro
```

Ejemplo pantalla código en c#

```
=====PANTALLA 3=====
Ejemplo de código en c#
Escoga la respuesta correcta
=====

1. Qué valor se imprime a pantalla?

int resultado =0;
int valor1 = 120;
int valor2 = 220;

if (valor1 >0 && valor2 >0) {
    resultado = calcular(valor1, valor2);
    Console.WriteLine(resultado);
}

int calcular(int primerNumero, int segundoNumero) {
    return ((primerNumero+segundoNumero)/4);
}

A. 85   B. 44
C. 22   D. 120
```

Rubrica de evaluación

Grabe un video mostrando cómo funciona su programa.

| Evaluación de programación (60 pts) | | |
|---|----------------|------------------|
| | Puntaje máximo | Puntaje Obtenido |
| Código de proyecto (comentarios, utilización de conceptos vistos en clase, orden etc) | 20 | |
| Validación de datos y gestión de errores | 10 | |
| Desarrollo de algoritmo y lógica de programación | 10 | |
| Funcionalidad | 20 | |
| Documentación (10 pts) | | |
| Presentación - manual de usuario | 5 | |
| Calidad, formatos y estructura de la documentación. | 5 | |
| Evaluación de exposición (video) (30 pts) | | |
| Calidad de exposición | 10 | |
| Demostración de código | 10 | |
| Dominio de temas en la exposición | 10 | |
| TOTAL | 100 | |

Documentación: Debe incluir diagramas de flujo de funcionalidades del programa y diagrama de clases. Agregar manual de usuario donde se detalle cómo utilizar el programa con ejemplos visuales y claros para un usuario final.

Entrega:

Cree una en ONE DRIVE carpeta llamada "Proyecto 2" suba su video, proyecto (en un archivo comprimido ZIP) y documentación. En su portal suba el link a esta carpeta.

Referencias

<http://programmingisfun.com/command-line-ascii-design/>

http://programmingisfun.com/command_line_window_properties/

<http://programmingisfun.com/console-application-color/>