

**Proyecto Individual 1**

**Valor: 50%**

**Objetivo:** Programar una aplicación que responda a los requerimientos planteados en el caso dadoutilizando los principales elementos semánticos y sintácticos del modelo de programación estructurada.

**Descripción:** La persona docente asigna un caso a las personas participantes, para lo cual la personaparticipante realizará el análisis y programará una aplicación con el lenguaje de programación C++, y con el paradigma de programación estructurada. Se deberá de entregar un **Informe con el planteamiento del caso** y la **aplicación programada**. La entrega del archivo final debe ser en formato digital.

**Instrucciones para las personas participantes:**

**I Parte (Documento escrito):** El documento por elaborar debe contener los siguientes apartados:

1. Portada
2. Tabla de contenidos
3. Introducción
4. Objetivos
5. Problemas Empresariales (ejercicios que se requiere solucionar)
6. Análisis del ejercicio 1
   * + Descripción del problema empresarial
   * Datos de entrada
   * Datos de Salida
   * Suposiciones u observaciones
   * Alternativas de solución
   * Determinación de reglas de negocio
   * Algoritmo (descripción detallada y puntualizada de cada uno de los procesos requeridos para la solución del ejercicio, sin el uso de lenguaje técnico)
7. Análisis del ejercicio 2
   * + Descripción del problema empresarial
   * Datos de entrada
   * Datos de Salida
   * Suposiciones u observaciones
   * Alternativas de solución
   * Determinación de reglas de negocio
   * Algoritmo (descripción detallada y puntualizada de cada uno de los procesos requeridos para la solución del ejercicio, sin el uso de lenguaje técnico)

La fecha de entrega para esta **PRIMERA PARTE** es el 18 de noviembre, 2022

**II Parte: Código de la solución** (archivo del Proyecto en C++)

La fecha de entrega para esta **SEGUNDA PARTE** es el 09 de diciembre, 2022

**III Parte: Entrevista del proyecto**

La fecha de la entrevista será el 12 de Diciembre, 2022

Realice el trabajo de forma individual, limpia y ordena.

Sea ético para la elaboración de la aplicación y el documento.

**Cualquier indicio de trabajos copiados o segmentos del programa** entre compañeros o copiados de Internet u otra fuente, anula el proyecto, obteniendo una calificación de 0.

**Ejercicio 1:**

El juego de cartas conocido como **Blackjack** o 21 consiste en un sencillo juego cuyo objetivo es lograr obtener 21 puntos o lo más cercano posible, pero sin pasarse.

En la baraja cada carta numérica vale sus respectivos puntos, cada carta con dibujo (J, Q, K) valen 10 puntos y el As vale 1 u 11 puntos, el valor del As lo decide el jugador a su conveniencia.

Se reparten 2 cartas a cada jugador, y cada jugador decide si pide más cartas o se queda con su valor actual. Gana la persona que logre obtener 21 puntos, o lo más cercano posible, pero sin pasarse.

**Implementación:**

Realice una aplicación de consola que permita a 3 personas jugar una partida de Blackjack, no se utilizarán las reglas oficiales (dividir, doblar, rendirse, seguro, cobrar 1 a 1, etc.). Tampoco existirán las reglas de verse obligado a pedir si se tiene mayor o menor cantidad de ciertos puntos. No existirá la figura oficial del crupier (repartidor), ya que este tiene reglas diferentes.

Se realizará una partida sencilla en donde al iniciar se le reparten 2 cartas a cada una de las 3 personas. Cada persona puede decidir si quiere una carta más en cada ronda, pero si se pasa de 21 automáticamente pierde.

Si una persona tiene un As se tomará como 11. Pero cuando una persona se pase de 21 automáticamente se tomará el As como 1.

En cada ronda **cada persona decide** si desea tomar otra carta o no, y además cada vez que una persona toma una carta se comprueba si tiene un 21 para declarar que ganó.

Cuando **las 3 personas** ya hayan indicado que **no desean tomar más cartas**, entonces el programa calculará quién ganó, tomando como ganadora a la persona cuyos puntos estén más cerca de 21, pero sin pasarse, como se indicó anteriormente. Podría haber empates.

**Ejercicio 2:**

Crear una aplicación para el juego del dominó. El objetivo de un jugador es ganar la partida conforme criterios básicos de juego.

**Implementación:**

Realice una aplicación de consola utilizando el lenguaje de programación C++, que permita a 4 personas jugar una partida de dominó, no se utilizarán las reglas oficiales. Se propone una versión simplificada del juego. Se realizará una partida sencilla en donde las reglas básicas que regulan el juego del dominó se describen a continuación:

* El juego se desarrolla con cuatro jugadores, a los cuales hay que solicitarles el nombre.
* Las 28 fichas de dominó serán repartidas al azar **(7 fichas a cada jugador).**
* El jugador/a que inicia la partida será quien tenga la ficha 6/6 (doble seis), continuando el turno por el siguiente jugador según el orden de ingreso. En su turno, cada jugador colocará una de sus piezas con la restricción de que dos piezas solo pueden colocarse juntas cuando los cuadrados adyacentes son del mismo valor. Si un jugador no puede colocar ninguna ficha en su turno, tendrá que pasar el turno al siguiente jugador.
* El final de la partida puede venir determinado por dos situaciones: (1) un jugador colocó la última de sus fichas y, por tanto, ganó la partida; o (2) todos los jugadores tienen aún fichas, pero ninguno puede colocar ninguna de ellas. En este último caso, ganará la partida el jugador cuyas fichas sumen menos puntos.

**Proyecto Individual**

**Documento escrito I Parte - Valor: 15%**

**Escala de calificación numérica**

Participante:

|  |  |  |  |
| --- | --- | --- | --- |
| **Lista de rasgos observables** | **Criterios** | | |
| **CC** | **CI** | **II** | |
| **2** | **1** | **0** | |
| **En cuanto al documento escrito (I Parte – 15%)** |  |  |  | |
| 1. Incluye portada, tabla de contenidos y objetivos |  |  |  | |
| 1. La introducción y conclusión son acordes al tema |  |  |  | |
| 1. Descripción del problema empresarial (ejercicio 1) |  |  |  | |
| 1. Determina los datos de entrada del ejercicio 1 |  |  |  | |
| 1. Determina los datos de salida del ejercicio 1 |  |  |  | |
| 1. Establece suposiciones u observaciones del ejercicio 1 |  |  |  | |
| 1. Describe claramente las alternativas de solución del ejercicio 1 |  |  |  | |
| 1. Define las reglas de negocio del ejercicio 1 |  |  |  | |
| 1. Desarrolla el algoritmo de forma completa para el ejercicio 1 |  |  |  | |
| 1. El algoritmo no presenta errores de lógica para el ejercicio 1 |  |  |  | |
| 1. La lógica utilizada en el algoritmo es eficiente para el ejercicio 1 |  |  |  | |
| 1. El algoritmo está diseñado de forma funcional y modular para el ejercicio 1 |  |  |  | |
| 1. Descripción del problema empresarial (ejercicio 2) |  |  |  | |
| 1. Determina los datos de entrada del ejercicio 2 |  |  |  | |
| 1. Determina los datos de salida del ejercicio 2 |  |  |  | |
| 1. Establece suposiciones u observaciones del ejercicio 2 |  |  |  | |
| 1. Describe claramente las alternativas de solución del ejercicio 2 |  |  |  | |
| 1. Define las reglas de negocio del ejercicio 2 |  |  |  | |
| 1. Desarrolla el algoritmo de forma completa para el ejercicio 2 |  |  |  | |
| 1. El algoritmo no presenta errores de lógica para el ejercicio 2 |  |  |  | |
| 1. La lógica utilizada en el algoritmo es eficiente para el ejercicio 2 |  |  |  | |
| 1. El algoritmo está diseñado de forma funcional y modular para el ejercicio 2 |  |  |  | |
| **Puntos Totales** | **44 puntos** | | | |
| **Puntos Obtenidos** |  | | | |
| **Calificación** |  | | | |
| **Porcentaje Obtenido** |  | | | |

**Proyecto Individual**

**Aplicación en C++ II Parte - Valor: 25%**

**Escala de calificación numérica**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lista de rasgos observables** | **Criterios** | | |
| **CC** | **CI** | **II** | |
| **2** | **1** | **0** | |
| **En cuanto a la aplicación en C++ (II Parte – 25%)** |  |  |  | |
| **Del ejercicio 1 Juego del BlackJack:** |  |  |  | |
| 1. La aplicación compila sin errores |  |  |  | |
| 1. La aplicación no presenta errores conceptuales |  |  |  | |
| 1. La solución es eficiente |  |  |  | |
| 1. Incorpora comentarios para documentar el código |  |  |  | |
| 1. Asigna nombres adecuados a las variables |  |  |  | |
| 1. Realiza una declaración correcta de las variables y/o constantes |  |  |  | |
| 1. Incorpora el uso de arreglos o estructuras de datos a la aplicación |  |  |  | |
| 1. Utiliza programación modular de forma apropiada |  |  |  | |
| 1. Utiliza los tipos de archivos (.cpp, .h) de forma correcta |  |  |  | |
| 1. El código esta tabulado y legible |  |  |  | |
| 1. Realiza una correcta transferencia de parámetros a funciones y procesos |  |  |  | |
| 1. Utiliza las funciones de forma correcta (invocaciones y retornos) |  |  |  | |
| 1. Incorpora funciones predeterminadas |  |  |  | |
| 1. La interacción de la aplicación con el usuario es amigable |  |  |  | |
| 1. Maneja las excepciones de forma apropiada |  |  |  | |
| **Del ejercicio 2 Juego del Dominó:** |  |  |  | |
| 1. La aplicación compila sin errores |  |  |  | |
| 1. La aplicación no presenta errores conceptuales |  |  |  | |
| 1. La solución es eficiente |  |  |  | |
| 1. Incorpora comentarios para documentar el código |  |  |  | |
| 1. Asigna nombres adecuados a las variables |  |  |  | |
| 1. Realiza una declaración correcta de las variables y/o constantes |  |  |  | |
| 1. Incorpora el uso de arreglos o estructuras de datos a la aplicación |  |  |  | |
| 1. Utiliza programación modular de forma apropiada |  |  |  | |
| 1. Utiliza los tipos de archivos (.cpp, .h) de forma correcta |  |  |  | |
| 1. El código esta tabulado y legible |  |  |  | |
| 1. Realiza una correcta transferencia de parámetros a funciones y procesos |  |  |  | |
| 1. Utiliza las funciones de forma correcta (invocaciones y retornos) |  |  |  | |
| 1. Incorpora funciones predeterminadas |  |  |  | |
| 1. La interacción de la aplicación con el usuario es amigable |  |  |  | |
| 1. Maneja las excepciones de forma apropiada |  |  |  | |
| **Puntos Totales** | **60 puntos** | | | |
| **Puntos Obtenidos** |  | | | |
| **Calificación** |  | | | |
| **Porcentaje Obtenido** |  | | | |

CC: Correcto y completo CI: Correcto pero Incompleto II: Incorrecto e Incompleto

**Proyecto Individual**

**Entrevista III Parte - Valor: 10%**

**Escala de calificación numérica**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Participante: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Objetivo**: Comprobar mediante una entrevista individual el dominio de los contenidos y el avance obtenido en el transcurso del módulo. | | | | | | |
| **Lista de rasgos observables** | | | **Criterios** | | | |
| **LL** | | **EP** | **NL** |
| **2** | | **1** | **0** |
| **En cuanto a la presentación** | | |  | |  |  |
| 1. Se presenta puntual a la cita | | |  | |  |  |
| **En cuanto al desarrollo.** | | |  | |  |  |
| 1. Contesta las preguntas teóricas de forma satisfactoria | | |  | |  |  |
| 1. Comprende la lógica implementada en el proyecto | | |  | |  |  |
| 1. Comprende la declaración y uso de estructuras de datos | | |  | |  |  |
| 1. Demuestra dominio de la teoría desarrollada | | |  | |  |  |
| 1. Demuestra dominio en la elaboración de algoritmos en seudocódigo | | |  | |  |  |
| 1. Demuestra dominio en el análisis de problemas | | |  | |  |  |
| 1. Demuestra dominio en el manejo de funciones y procesos | | |  | |  |  |
| 1. Demuestra dominio en el uso eficiente de variables y parámetros | | |  | |  |  |
| 1. Demuestra dominio en el manejo de excepciones | | |  | |  |  |
| 1. Realiza el ejemplo asignado en la valoración individual | | |  | |  |  |
| 1. El ejemplo realizado funciona | | |  | |  |  |
|  | | |  | |  |  |
| **Total 24 puntos**  **Criterios**  **LL:** Lo logró **EP:** En proceso **NL:** No lo logró | | | | | | |
| **Puntos Obtenidos** | **Calificación** | **Porcentaje obtenido** | | **Calificación obtenida:** | | |
| **Observaciones:** | | | | | | |