



## 750015C - Fundamentos de Programación Orientada a Objetos (FPOO)

### Primer Examen Parcial 2022S02

Nombre : \_\_\_\_\_

#### Recomendaciones

1. Lea el enunciado completo, antes de empezar a plantear su solución.
2. Evite entablar conversación con sus compañeros; recuerde que la evaluación es individual. Evite hacer fraude o colaborar con el mismo.

#### Marcar el examen [0]

A continuación se describe el proceso para marcar el examen.

Cree un proyecto en Dev C++ con nombre

Apellido-Nombre-EP02

Donde reemplace Apellido y Nombre por su primer apellido y primer nombre, respectivamente. La carpeta del proyecto debe tener el mismo nombre

#### Entrega del proyecto [0]

De la carpeta del proyecto, borre (solamente) los archivos con extensión .o y .exe. No borre otros archivos.

Comprima la carpeta del proyecto en **formato ZIP (únicamente)** y haga su entrega a través del Campus Virtual. Para comprimir la carpeta, haga clic derecho sobre la carpeta y seleccione la opción comprimir.

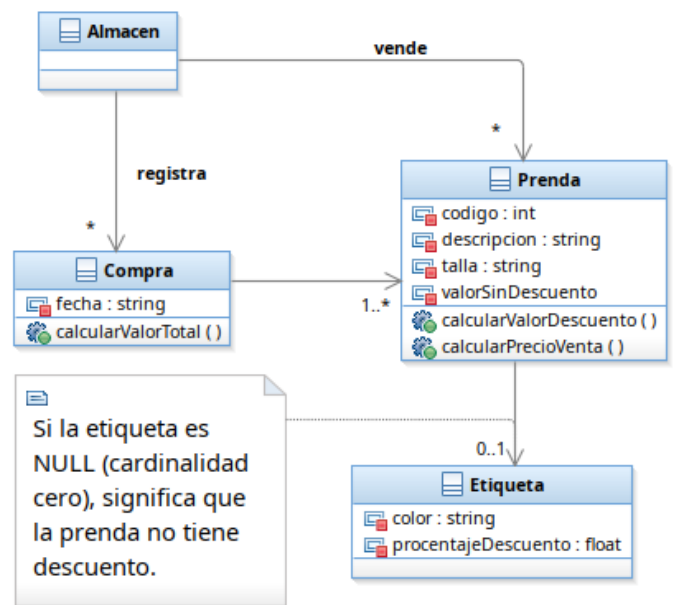
#### Punto No. 01 - Diagrama de clases [RA2-20]

Un reconocido almacén de la ciudad, ha lanzado una innovadora estrategia para vender en promoción las prendas (de ropa) de antiguas colecciones.

En las estanterías del almacén, las prendas están etiquetadas con el valor comercial de las mismas y con una etiqueta de color la cual indica al

comprador que dicha prenda tiene descuento. Las prendas sin etiqueta (NULL) no tienen descuento.

Para la implementación de dicho programa se ha elaborado el siguiente diagrama de clases.



#### Punto No. 02 - Métodos [0]

Implemente los métodos y verificación de valores válidos de acuerdo a la lógica que se describe a continuación.

#### Punto No. 03 - Programa principal [RA4-70]

En el programa principal:

- [0] Cree un objeto de tipo almacén.
- [10] Cree cinco objetos de tipo Prenda, con diferente código y relaciónelos con el almacén. No es necesario la verificación de valores válidos.

De los 5 objetos creados, cree dos sin etiqueta y los tres restantes, con etiquetas de diferente color, porcentaje de descuento de 10, 25 y 35% (respectivamente). Defina usted el color de cada etiqueta.

### Flujo básico de eventos (registrar compra)

Este flujo básico inicia cuando cliente se acerca a la caja registradora con las prendas que desea comprar

1. [5] El programa crea un objeto de tipo Compra con la fecha del día (establezca la fecha de hoy como cadena de texto - no es necesario lectura de datos).
2. Los siguientes pasos se repiten para cada una de las prendas que desea comprar el cliente:
  - a. [0] El cajero ingresa el código de la prenda
  - b. [20] El programa busca y muestra la información de la prenda (código, descripción, valor sin descuento, valor del descuento y valor con descuento)

Ayuda: Si la prenda no tiene etiqueta, el valor sin descuento y el valor con descuento será el mismo, por ende, el valor del descuento es cero.

- c. [5] El programa agrega la prenda a la compra.
- d. [10] El programa muestra el valor total a pagar.
- e. [10] El programa pregunta si desea registrar otra prenda. Véase el apartado “Ayuda” más adelante.

Si no logra implementar este paso, registre solo una prenda [-5].

### Flujo Alterno 01 - Prenda no encontrada [10]

En el paso 2b del flujo básico, si el programa no encuentra una prenda con el código ingresado, el programa muestra un mensaje indicando dicha situación.

El programa continúa en el paso 2a (ingresar el código de la prenda)

### Clase Excepción (Exception)

Puedes descargar la clase Excepción de los siguientes enlaces: [excepcion.h](#) y [exception.cpp](#)

### Ayuda

Para preguntar si desea registrar otra prenda, puede hacer uso del siguiente fragmento de código. Si digita un valor diferente de 1, el proceso finalizará.

```
// Lógica para repetir el proceso
// para cada una de las prendas a
// comprar
int opcion = 1;
while (opcion == 1) {
    // Lógica de registrar la compra

    // Preguntar si desea registrar otra
    // prenda
    cout << "Desea registrar otra prenda"
        << " 1:Si / 0:No ... ";
    cin >> opcion ;
}
```

### Utilización de listas en C++

A continuación se incluye a manera de apuntes el esquema básico para la utilización de listas (T es el nombre de la clase)

```
// Definición de la variable de tipo lista
list<T*> * variable ;

// Creación de una lista
variable = new list<T*> ( ) ;

// Forma de recorrer una lista usando while
list<T*>::iterator it = variable->begin() ;
T * objeto ;
while ( it != variable -> end () ) {
    objeto = * it ;
    ..... // lógica del método
    it ++ ;
}

// Forma de recorrer una lista usando for
list<T*>::iterator it = variable->begin() ;
T * objeto ;
for ( ; it != variable -> end () ; it ++ ) {
    objeto = * it ;
    ..... // lógica del método
}
```