

Configurador de PC

Primera entrega prácticas

Interfaces Persona-Computador

Una tienda de informática quiere ofrecer a sus clientes una aplicación para que se construyan su propio PC eligiendo los componentes entre los disponibles en el almacén. En principio, no se puede presuponer que los usuarios tengan experiencia previa seleccionando los componentes para un nuevo PC, por lo que la aplicación debería ayudar a los usuarios noveles. El objetivo de la primera entrega es construir una aplicación que permita al usuario preparar el presupuesto de un PC a partir de sus componentes.

La aplicación permitirá al usuario seleccionar los distintos componentes del PC mostrando información sobre su precio y disponibilidad. La aplicación debe facilitar la tarea de búsqueda de los componentes según distintos criterios (nombre, categoría, precio, disponibilidad, etc.)

El resultado de la configuración se mostrará en una ventana resumen que contendrá una línea por cada componente seleccionado, el subtotal sin IVA, el IVA y el importe total del presupuesto. Cada línea del presupuesto contendrá el componente seleccionado, el precio unitario sin IVA, la cantidad de componentes y el importe total sin IVA de la línea.

Un PC está compuesto, al menos, por los siguientes elementos:

- Una placa base
- Un procesador
- Memoria ram
- Tarjeta gráfica
- Disco duro (normal o SSD)
- Una torre

Como componentes opcionales, el usuario podrá adquirir:

- Teclado
- Ratón
- Monitor
- Altavoces
- Multilectores
- Grabadora DVD/Bluray
- Ventiladores
- Fuente de alimentación

Se pide implementar la siguiente funcionalidad:

- Al generar el presupuesto final, el sistema emitirá un aviso en el caso de que el usuario no haya seleccionado al menos una unidad de cada uno de los componentes no opcionales.
- El usuario podrá seleccionar un PC preconfigurado entre una serie de opciones predefinidas, o podrá empezar desde cero, seleccionando todos los componentes.
- Se podrá editar el presupuesto actual (añadir componentes, eliminarlos, modificar cantidades, etc.)
- Se podrá cargar y guardar la configuración actual del PC en un fichero.
- Se podrá visualizar el presupuesto final que, además de los componentes del ordenador y los precios tal y como se ha explicado arriba, tendrá en la cabecera una imagen con el logo y el nombre de la tienda. En el pie deberá aparecer la fecha de confección del presupuesto, y un aviso de que tiene una validez de 7 días.

Para acceder a la información de la base de datos se dispone de la clase `Database`. También se define la siguiente clase:

```
public class Product {
    public enum Category {
        SPEAKER, HDD, HDD_SSD, POWER_SUPPLY, DVD_WRITER, RAM, SCREEN,
        MULTIREADER, MOTHERBOARD, CPU, MOUSE, GPU,
        KEYBOARD, CASE, FAN
    }

    public Product(String description, double price, int stock, Category
category) {
        this.description = description;
        this.price = price;
        this.stock = stock;
        this.category = category;
    }

    public Category getCategory() {
        return category;
    }

    public String getDescription() {
        return description;
    }

    public double getPrice() {
        return price;
    }

    public int getStock() {
        return stock;
    }
    private final String description;
    private final double price;
    private final int stock;
    private final Category category;
}
```

La clase Database contiene varias funciones de acceso a la base de datos:

```
public class Database {  
    /**  
     * Devuelve todos los productos de la categoría  
     *  
     * @param cat la categoría buscada  
     * @return la lista de productos  
     */  
    public static List<Product> getProductByCategory(Product.Category cat)  
  
    /**  
     * Busca los productos de una categoría que se encuentra en el rango de  
     * precios indicado  
     *  
     * @param cat la categoría buscada  
     * @param minPrice el precio mínimo (en euros)  
     * @param maxPrice el precio máximo (en euros)  
     * @param available si true, sólo se devuelven productos con existencias  
     * @return la lista de productos que cumplen los criterios  
     */  
    public static List<Product> getProductByCategoryAndPrice(Product.Category cat,  
double minPrice, double maxPrice, boolean available)  
  
    /**  
     * Busca los productos en una categoría cuya descripción contiene la cadena  
     * indicada  
     *  
     * @param cat la categoría buscada  
     * @param substring la subcadena a buscar  
     * @param available si true, sólo se devuelven productos con existencias  
     * @return la lista de productos que cumplen los criterios  
     */  
    public static List<Product> getProductByCategoryAndDescription(Product.Category  
cat, String substring, boolean available)  
  
    /**  
     * Busca los productos en una categoría cuya descripción contiene la cadena  
     * indicada, su precio está en el rango indicado, y tiene o no  
     * disponibilidad  
     *  
     * @param cat la categoría buscada  
     * @param substring la subcadena a buscar  
     * @param minPrice el precio mínimo (en euros)  
     * @param maxPrice el precio máximo (en euros)  
     * @param available si true, sólo se devuelven productos con existencias  
     * @return la lista de productos que cumplen los criterios  
     */  
    public static List<Product>  
getProductByCategoryDescriptionAndPrice(Product.Category cat, String substring,  
double minPrice, double maxPrice, boolean available)  
}
```

Se aconseja definir la clase PC, que representa la configuración en un momento dado de un PC. La clase PC podrá tener un nombre opcional (por ejemplo, “Oferta del mes”, o “Para superjugones” o “Mi PC ideal”), y una lista de objetos Componente, donde la clase

Componente tiene una cantidad y un Producto (así, por ejemplo, podrás tener cuatro módulos de memoria o dos tarjetas gráficas). La clase PC se puede utilizar en varios lugares de la aplicación. Por ejemplo, los PC preconfigurados se podrán almacenar como una lista de objetos PC. El PC que se está editando en cada momento también puede utilizar esta clase. Los PC preconfigurados no se pueden modificar y vienen predefinidos por la aplicación. Por otra parte, el PC que está configurando el usuario sí permitirá modificar el objeto PC correspondiente.

Junto a la práctica podrás encontrar una librería llamada **DatabasePCStore.jar**. Dicho fichero contiene las clases Product y Database descritas arriba. Para usar dichas clases en tu proyecto, haz clic con el botón derecho sobre el apartado Libraries desde el explorador de proyectos, y selecciona la opción Add JAR/Folder. Selecciona el fichero anterior y ya podrás usar la clase Product y acceder a la base de datos en tu proyecto.

Ampliaciones:

Para obtener mayor puntuación puedes implementar las siguientes ampliaciones opcionales:

- Permitir trabajar con varios presupuestos.
- Ofrecer al usuario la opción de imprimir el presupuesto

Normativa de entrega:

- Exporta el proyecto NetBeans en un ZIP (<http://politube.upv.es/play.php?vid=68666>) y súbelo a la tarea de poliformaT correspondiente. Cada grupo de alumnos hará una sola entrega, incluyendo en el campo de comentarios los nombres de los dos alumnos.

Evaluación:

- Aquellos proyectos que no compilen o que no muestren la pantalla principal al arrancar se calificarán con un cero.
- Se deberán incluir los diálogos de confirmación, errores, etc. que se considere necesario.
- Para evaluar el diseño de la interfaz de la aplicación se tendrán en cuenta las directrices estudiadas en clase de teoría.
- Debe ser posible redimensionar la pantalla principal, cuyos controles se ajustarán adecuadamente para usar el espacio disponible (usa los contenedores vistos en clase).