SOLUCIÓN 7

```
//modificar Tª de congelacion de un producto buscando por su id
public boolean modificarTemperaturaCongelacion(String id, double temp){
    for (Producto producto : listaP) {
        if (producto!=null && producto.getIdProducto().equals(id) &&
                producto instanceof Congelado) {

            Congelado congelado =  (Congelado) producto;
            congelado.setTemperaturaCon(temp);
            return true;
        }   }   return false;
}
//modificar Tª de congelacion de un producto buscando por su id
public boolean modificarTemperaturaCongelacion2(String id, double temp){

    for (int i = 0; i < listaP.length; i++) {
        if (listaP[i]!=null && listaP[i].getIdProducto().equals(id) &&
                listaP[i] instanceof Congelado) {
            Congelado congelado =  (Congelado) listaP[i];
            congelado.setTemperaturaCon(temp);
            return true;
        }
    }       return false;
}
```

```
//retornar array de frescos de un país concreto
public Fresco [] consultarProductosPorPaisOrigen(String paisBuscado){
    int contador=0;
    for (Producto producto : listaP) {
        if (producto!=null && producto instanceof Fresco &&
                ((Fresco) producto).getPaisOrigen().equals(paisBuscado)) {
            contador++;
        }
    }
    //creo el array y lo retorno
    Fresco [] array = new Fresco[contador];
    int indice=0;
    for (Producto producto : listaP) {
        if (producto!=null && producto instanceof Fresco &&
                ((Fresco) producto).getPaisOrigen().equals(paisBuscado)) {

            Fresco fresco = (Fresco) producto;
            array[indice]=fresco;
            indice++;
        }
    }
    return array;
}
```

```java
//retornar array de productos caducados
public Producto [] consultarProductosCaducados(){

    Producto [] caducados;

    int contador=0;//cuenta los caducados
    for (Producto producto : listaP) {
        if (producto!=null &&
                producto.getFechaCad().isBefore(LocalDate.now()))
            contador++;
    }

    //System.out.println("Caducados" + contador);
    caducados = new Producto[contador];

    int indice=0;
    for (Producto producto : listaP) {
        if (producto!=null &&
                producto.getFechaCad().isBefore(LocalDate.now()))
            caducados[indice++]=producto;
    }
    return caducados;
}
```

```java
public Producto [] obtenerProductosDelTipo (String tipo) {
    //tipo = "C" o "R" o "F"
    int contador=0;//cuenta los productos
    for (Producto producto : listaP) {
        if (producto!=null && producto instanceof Fresco
                && tipo.equalsIgnoreCase("F"))
            contador++;
        else if (producto!=null && producto instanceof Refrigerado
                && tipo.equalsIgnoreCase("R"))
            contador++;
        else if (producto!=null && producto instanceof Congelado
                && tipo.equalsIgnoreCase("C"))  contador++;
    }Producto [] productosTipo = new Producto[contador];
    int indice=0;
    for (Producto producto : listaP) {
        if (producto!=null && producto instanceof Fresco
                && tipo.equalsIgnoreCase("F"))
            productosTipo[indice++]=producto;
        else if (producto!=null && producto instanceof Refrigerado
                && tipo.equalsIgnoreCase("R"))  productosTipo[indice++]=producto;
        else if (producto!=null && producto instanceof Congelado
                && tipo.equalsIgnoreCase("C"))  productosTipo[indice++]=producto;
    }   return productosTipo;
```