

**Portafolio de Evidencias Testing Fundamentals**

**Software Testing y su relación con el SDLC**

Dalyn Yenisey Lara Cortés

07 de marzo de 2022

# Software testing y su relación en el ciclo de vida de desarrollo del software (SDLC)

Dado el siguiente requerimiento realiza los siguientes puntos:

1. Crea el programa en la herramienta online JDOODLE utilizando Java.
2. Identifica los defectos o ambigüedades que puedas encontrar en el requerimiento y llena la tabla mostrada más abajo.
3. Copia y pega el código.
4. Graba un video de 1 minuto demostrando su funcionalidad.

La pizzería Bella Napoli ofrece pizzas vegetarianas y no vegetarianas a sus clientes. Los ingredientes para cada tipo de pizza aparecen a continuación:

* Ingredientes vegetarianos: Pimiento, Hongos y Espinaca.
* Ingredientes no vegetarianos: Pepperoni, Jamón y Salchicha.

Escribir un programa que pregunte al usuario si quiere una pizza vegetariana o no, y en función de su respuesta le muestre un menú con los ingredientes disponibles para que elija. Sólo se puede elegir 3 ingredientes además de la mozzarella y el tomate que están en todas las pizzas. Al final se debe mostrar por pantalla si la pizza elegida es vegetariana o no y todos los ingredientes que lleva.

|  |  |
| --- | --- |
| Ambigüedades o defectos en el requerimiento | ¿Cómo se puede mejorar? |
| El requerimiento no especifica si es posible hacer una pizza con menos de 3 ingredientes, sin contar la mozzarela y el tomate | El enunciado podría indicar que el número máximo de ingredientes es 3 y el mínimo es 1. |
| No se especifica si es posible repetir un ingrediente más de una vez. | El enunciado podría indicar que se puede agregar un ingrediente más de una vez o que no es posible. |
| No se indica si es posible comprar más de una pizza por cliente. | El enunciado debería especificar si únicamente se evaluará una pizza por cliente o es posible hacer una evaluación de más de una pizza por cliente. |

Código:

|  |
| --- |
| import java.util.Scanner;  public class MyClass {  public static void main(String args[]) {  int[] ing = new int[3];  String[] vg = {"Pimientos", "Hongos", "Espinaca"};  String[] nvg= {"Pepperoni", "Jamon", "Salchica"};    Scanner sc = new Scanner(System.in);  System.out.println("Bienvenido a la Pizzeria Bella Napoli\n\n Por favor, elija el tipo que desea comprar:\n1.Vegetariana\n2.No Vegetariana");  String pz = sc.nextLine();    if(pz.equals("1")){  System.out.println("\n\nSeleccione los ingredientes que desea agregar a su pizza:\n1.Pimientos\n2.Hongos\n3.Espinaca");  for(int i = 0; i < 3; i++){  System.out.println("Ingrese el número que elige como ingrediente numero "+(i+1)+":");  ing[i]= sc.nextInt()-1;  }  System.out.println("\n\nLa pizza que eligió es Vegetariana");  System.out.println("Los ingredientes que eligio son:");  for(int i = 0; i < 3; i++){  System.out.println(vg[ing[i]]);  }      }else{  System.out.println("Seleccione los ingredientes que desea agregar a su pizza:\n1.Pepperoni\n2.Jamon\n3.Salchica");  for(int i = 0; i < 3; i++){  System.out.println("Ingrese el número que elige como ingrediente numero "+(i+1)+":\n\n");  ing[i]= sc.nextInt()-1;  }  System.out.println("La pizza que eligió No es Vegetariana");  System.out.println("Los ingredientes que eligio son:");  for(int i = 0; i < 3; i++){  System.out.println(nvg[ing[i]]);  }  }    }  } |

Del requerimiento anterior descomponlo en historias de usuario. Para este ejercicio se propone descomponer el requerimiento en al menos 5 historias de usuario con sus criterios de aceptación. Recuerda que una historia de usuario describe una pequeña parte de la funcionalidad de una especificación.

|  |  |
| --- | --- |
| Historia de usuario | Criterios de aceptación |
| Yo como cliente quiero que desde el inicio el programa me pregunte qué tipo de pizza deseo para saber lo que prefiero. | * El programa pregunta qué tipo de pizza desea el cliente. * Se almacena la información proporcionada por el usuario. |
| Yo como cliente quiero que el programa me muestre únicamente los ingredientes que prefiero para poder elegir entre ellos. | * El sistema muestra la información disponible basado en la preferencia del usuario. * Ingredientes vegetarianos: Pimiento, Hongos y Espinaca. * Ingredientes no vegetarianos: Pepperoni, Jamón y Salchicha. |
| Yo como cliente deseo que el programa me especifique si hay ingredientes por defecto en la pizza y cuantos puedo elegir. | El sistema muestra una leyenda indicando que:   * Todas las pizzas llevan mozzarela y tomate. * El cliente puede elegir máximo tres ingredientes para su pizza. |
| Yo como cliente quiero que el programa recuerde los ingredientes que elegí para saber cómo está conformada mi pizza. | * El sistema almacena cada una de las selecciones del usuario, tomando en cuenta el número máximo de ingredientes. |
| Yo como como cliente deseo que el sistema me muestre qué tipo de pizza elegí y los ingredientes que lleva para saber qué recibiré. | * El sistema muestra una leyenda con el tipo de pizza y los ingredientes que lleva sin contar el queso y el tomate. |

**Del ejercicio anterior:**

1. Implementa nuevamente el código utilizando una historia de usuario a la vez (preferentemente una función por historia de usuario).
2. Integra las partes del código.
3. Copia y pega el código.
4. Graba un video de 1 minuto demostrando su funcionamiento.

Código ver 2

|  |
| --- |
| import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  public class MyClass {  static int max =3;  static Scanner sc = new Scanner(System.in);    public static boolean val\_opt(String x, int max\_opt){    for(int i = 1; i <= max\_opt; i++){  if(x.equals(Integer.toString(i))){  return true;  }  }    System.out.println("\nHa ingresado una opción no válida. Intentelo de nuevo\n");  return false;    }    public static void inicio() {  List<Integer> sel = new ArrayList<Integer>();    System.out.println("Bienvenido a la Pizzeria Bella Napoli\n\n");  String type = pizza\_typ();  List<String> ing = pizza\_ing(type);  show\_ing(max,ing);  ing\_inf(max);  sel.addAll(select\_ing(max,ing.size()));  pizza\_inf(ing,sel,max,type);    }    public static String pizza\_typ(){  String pz="";  do{  System.out.println("Por favor, elija el tipo que desea comprar:\n\n1.Vegetariana\n\n2.No Vegetariana");  pz = sc.nextLine();  }while(!val\_opt(pz,2));    if(pz.equals("1")){  return "Vegetariana";  }else{  return "No Vegetariana";  }    }  public static List<String> pizza\_ing(String opt){  List<String> ingrd = new ArrayList<String>();  ingrd.add("Pimientos");  ingrd.add("Hongos");  ingrd.add("Espinaca");    if(opt.equals("Vegetariana")){  return ingrd;  }else{  ingrd.add("Pepperoni");  ingrd.add("Jamón");  ingrd.add("Salchica");  ingrd.add("Ninguno");  return ingrd;  }    }    public static void show\_ing(int n, List<String> ing){  System.out.println("\n\nEstos son los ingredientes disponibles para su pizza:");  for(int i = 0; i<ing.size(); i++){  System.out.println((i+1)+". "+ ing.get(i));  }  System.out.println();  }    public static List<Integer> select\_ing(int n, int n\_ing){  List<Integer> ing = new ArrayList<Integer>();  for(int i = 0; i < n; i++){  String aux = "";  do{  System.out.println("Ingrese el número del ingrediente que elige como número "+(i+1)+":");  aux = sc.nextLine();  }while(!val\_opt(aux,n\_ing));  ing.add(Integer.parseInt(aux));  }  return ing;  }    public static void ing\_inf(int n){  System.out.println("Todas las pizzas contienen mozzarela y tomate. Usted tiene derecho a elegir "+ n + " ingredientes además de los ya mencionados.\n");  }    public static void pizza\_inf(List<String> ing, List<Integer> sel, int n, String type){  System.out.println("\n\nLa pizza que eligió es: "+type);    System.out.println("Los ingredientes que eligio son:");  for(int i = 0; i < n; i++){  System.out.println(ing.get(sel.get(i)-1));  }  }  public static void main(String args[]) {  String pz="";  do {  inicio();  System.out.println("\n¿Desea hacer una nueva compra? 1. Sí / Cualquier otra tecla para salir");  pz = sc.nextLine();    }while(pz.equals("1"));  }  } |

Llena la siguiente tabla para responder las siguientes preguntas.

|  |  |
| --- | --- |
| Pregunta | Respuesta |
| ¿En cuál de los ejercicios fue más fácil identificar ambigüedades en las especificaciones? | En el segundo ejercicio. En el ejercicio uno identifiqué varios errores pero se notaron con más facilidad en el segundo. |
| ¿En cuál de los primeros dos ejercicios tuviste más claro cómo elaborar tu código y qué actividades desempeñar? ¿Y por qué? | En el segundo, porque las actividades a desarrollar estaban ya definidas. |
| ¿Por qué concluir que la comunicación es esencial en las actividades de testing y desarrollo? | Porque de esa forma se tiene claro lo que debe hacerse y se comprueba que el resultado es el correcto. |
| ¿Qué etapas del ciclo de vida del desarrollo de software identificas durante esta actividad? | Análisis: se realizó el análisis de cómo se debía abordar el problema a resolver.  Diseño: realizó el diseño lógico de la aplicación.  Implementación: realización del código.  Pruebas: Por cada función agregada se realizaron pruebas. |
| ¿Dónde piensas que se involucraría el proceso de testing de acuerdo a la actividad que se realizó? | En todas las etapas porque es necesario confirmar que se está haciendo bien el trabajo desde el inicio. En este caso específico, comenzando asegurar que se entendieron los requerimientos, el programa da el resultado esperado y cada parte funciona de forma adecuada. |