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

**Alumno: victor Jair salmeron Gonzalez**

|  |  |
| --- | --- |
| **Ambiguedades o defectos en requerimientos** | **Como se podría mejorar** |
| Preguntarle si quiere una pizza vegetariana y al final imprimir si es vegetarian | Solo bastaría con preguntarle al principio |
| Y en función de su respuesta mostrarle un menu | El menú se muestra independiente de la respuesta del cliente |
| Mostrar todos los ingredientes que lleva | Solo mostrar los ingredientes que elegio el cliente ya que mozzarella y tomate llevan todas |

|  |  |
| --- | --- |
| Historia de usuario | Criterio de aceptacion |
| Como usuario quiero que el sistema muestre la opción de elegir si el cliente desea pizza vegetariana o no para almacenar lo que el cliente quiere | El programa debe mostrar en pantalla la opción de elegir tipo de pizza |
| Como usuario quiero que la aplicación muestre los ingredientes a elegir para que el cliente elija | El programa debe motrar un menú con los ingredientes disponibles |
| Como usuario quiero que el sistema muestre al usuario que todas las pizzas llevan mozzarella | El sistema debe mostrar que todas las pizzas incluyen tomate y mozzarella |
| Como usuario quiero que el sistema permite al usuario elegir hasta 3 ingredientes para conocer que llevara la pizza | El sistema solo permitirá elegir hasta 3 ingredientes |
| Como usuario de la aplicación de la pizzería, quiero que el sistema despliegue al cliente si la pizza es vegetariana | Dependiendo de las opciones seleccionadas por el usuario, se debe desplegar al cliente si la pizza es vegetariana. |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Pregunta | Respuesta |
| En cual de los 2 ejercicios fue mas fácil identificar ambigüedades en las especificaciones y porque | En el segundo ya que se desgloso por historia de usuario |
| En cual de los 2 ejercicios tuviste mas claro como elaborar tu código y que actividades a desempeñar y porque | En el segundo ya que al ir leyendo cada historia de usuario, se generaba el código mas exacto |
| Porque concluir que la comunicación es esencial en las actividades de testing y desarrollo | Porque si hay buena comunicación , habrá calidad en el producto y se entenderán mejor los requerimientos |
| Que etapas del ciclo de vida del desarrollo de software identificas en esta actividad | Planeación,codificación,testing,implementación |
| Donde piensas que se involucraría el proceso fundamental del testing de acuerdo a la actividad que se realizo | En la fase de requerimientos ya que al encontrar los defectos en etapas tempranas el costo es menor |
|  |  |

|  |
| --- |
| package proyecto\_junit; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| import java.util.Scanner; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public class MyClass { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public static void main(String[] args) { |
|  |

|  |
| --- |
| // TODO Auto-generated method stub |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| int max\_ingredientes = 3; |
|  |

|  |
| --- |
| PizzaOrden nuevaOrden = new PizzaOrden(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| nuevaOrden.preguntarPizzaVegetariana(); |
|  |

|  |
| --- |
| nuevaOrden.desplegarMensajeDefault(); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| if (nuevaOrden.getOpcion\_veg()) { |
|  |

|  |
| --- |
| nuevaOrden.desplegarIngredientesVegetarianos(); |
|  |

|  |
| --- |
| } else { |
|  |

|  |
| --- |
| nuevaOrden.desplegarIngredientesNoVegetarianos(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| nuevaOrden.guardarIngredientes(); |
|  |

|  |
| --- |
| nuevaOrden.desplegarResumenOrden(); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| class PizzaOrden { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private boolean opcion\_veg = false; |
|  |

|  |
| --- |
| private String ingrediente\_1 = "", ingrediente\_2 = "", ingrediente\_3 = ""; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public PizzaOrden() { |
|  |

|  |
| --- |
| System.out.println("Bienvenido a la aplicación para ordenar pizza.\n\n" |
|  |

|  |
| --- |
| + "Sigue las instrucciones por favor.\n\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public boolean getOpcion\_veg() { |
|  |

|  |
| --- |
| return opcion\_veg; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void setOpcion\_veg(String opcion\_veg) { |
|  |

|  |
| --- |
| if (opcion\_veg.equals("Y")) { |
|  |

|  |
| --- |
| this.opcion\_veg = true; |
|  |

|  |
| --- |
| } else { |
|  |

|  |
| --- |
| this.opcion\_veg = false; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public String getIngrediente\_1() { |
|  |

|  |
| --- |
| return ingrediente\_1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void setIngrediente\_1(String ingrediente\_1) { |
|  |

|  |
| --- |
| this.ingrediente\_1 = ingrediente\_1; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public String getIngrediente\_2() { |
|  |

|  |
| --- |
| return ingrediente\_2; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void setIngrediente\_2(String ingrediente\_2) { |
|  |

|  |
| --- |
| this.ingrediente\_2 = ingrediente\_2; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public String getIngrediente\_3() { |
|  |

|  |
| --- |
| return ingrediente\_3; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void setIngrediente\_3(String ingrediente\_3) { |
|  |

|  |
| --- |
| this.ingrediente\_3 = ingrediente\_3; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void preguntarPizzaVegetariana() { |
|  |

|  |
| --- |
| System.out.println("¿Deseas pizza vegetariana (Y/N)?"); |
|  |

|  |
| --- |
| Scanner input = new Scanner(System.in); |
|  |

|  |
| --- |
| setOpcion\_veg(input.next()); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void desplegarMensajeDefault() { |
|  |

|  |
| --- |
| System.out.println("Todas las pizzas contienen Salsa de Tomate y Queso Mozzarella.\n\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void desplegarIngredientesVegetarianos() { |
|  |

|  |
| --- |
| System.out.println("Selecciona 3 ingredientes de la siguiente lista: \n" |
|  |

|  |
| --- |
| + "[a] Pimiento \n" |
|  |

|  |
| --- |
| + "[b] Hongos \n" |
|  |

|  |
| --- |
| + "[c] Espinaca \n" |
|  |

|  |
| --- |
| + "[o] Ninguno \n\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void desplegarIngredientesNoVegetarianos() { |
|  |

|  |
| --- |
| System.out.println("Selecciona 3 ingredientes de la siguiente lista: \n" |
|  |

|  |
| --- |
| + "[a] Pimiento \n" |
|  |

|  |
| --- |
| + "[b] Hongos \n" |
|  |

|  |
| --- |
| + "[c] Espinaca \n" |
|  |

|  |
| --- |
| + "[d] Peperoni \n" |
|  |

|  |
| --- |
| + "[e] Jamon \n" |
|  |

|  |
| --- |
| + "[f] Salchicha \n" |
|  |

|  |
| --- |
| + "[o] Ninguno \n\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void guardarIngredientes() { |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Scanner input = new Scanner(System.in); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| System.out.println("Ingrediente 1: "); |
|  |

|  |
| --- |
| setIngrediente\_1(input.next()); |
|  |

|  |
| --- |
| System.out.println("Ingrediente 2: "); |
|  |

|  |
| --- |
| setIngrediente\_2(input.next()); |
|  |

|  |
| --- |
| System.out.println("Ingrediente 3: "); |
|  |

|  |
| --- |
| setIngrediente\_3(input.next()); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| System.out.println(getIngrediente\_1() + getIngrediente\_2() + getIngrediente\_3()); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| public void desplegarResumenOrden() { |
|  |

|  |
| --- |
| if (this.opcion\_veg) { |
|  |

|  |
| --- |
| System.out.println("La pizza a ordenar es vegetariana.\n"); |
|  |

|  |
| --- |
| } else { |
|  |

|  |
| --- |
| System.out.println("La pizza a ordenar no es vegetariana.\n"); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| System.out.println("Los ingredientes seleccionados son: \n"); |
|  |

|  |
| --- |
| imprimirIngrediente(this.ingrediente\_1, ","); |
|  |

|  |
| --- |
| imprimirIngrediente(this.ingrediente\_2, ","); |
|  |

|  |
| --- |
| imprimirIngrediente(this.ingrediente\_3, ""); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| private void imprimirIngrediente(String ingrediente, String separator) { |
|  |

|  |
| --- |
| if (ingrediente.equals("a")) { System.out.print("Pimiento"); System.out.print(separator); } |
|  |

|  |
| --- |
| else if (ingrediente.equals("b")) { System.out.print("Hongos"); System.out.print(separator); } |
|  |

|  |
| --- |
| else if (ingrediente.equals("c")) { System.out.print("Espinaca"); System.out.print(separator); } |
|  |

|  |
| --- |
| else if (ingrediente.equals("d")) { System.out.print("Peperoni"); System.out.print(separator); } |
|  |

|  |
| --- |
| else if (ingrediente.equals("e")) { System.out.print("Jamon"); System.out.print(separator); } |
|  |

|  |
| --- |
| else if (ingrediente.equals("f")) { System.out.print("Salchicha"); System.out.print(separator); } |
|  |

|  |
| --- |
| else { System.out.print(""); } |
|  |

|  |
| --- |
| } |
|  |

}