

Luisa Martinez e Ingrid Vallecilla

Contexto

Se nos pidió crear un aplicativo para el restaurante Macumba, restaurante de comida típica del pacífico Colombiano con el principal objetivo de que sus usuarios puedan solicitar a domicilio los platos del restaurante. El aplicativo solo permite solicitar domicilios a usuarios registrados. El aplicativo debe mostrar al menos 4 tipos de plato diferentes, una vez seleccionado el plato el usuario puede agregar adicionales como: papas, patacones y bebidas, entre otras adiciones

Finalmente el usuario puede pagar su orden a través del aplicativo y este darle un feedback si el pago se pudo realizar o no. Si el pago es exitoso deberá llevarlo a una pantalla de resumen con lo que se compró y en cuanto tiempo llega su pedido. El aplicativo le permite ver al usuario los pedidos históricos que ha realizado hasta la fecha, la lista de pedidos realizados podrá ser ordenado por valor del pedido y fecha del pedido. Finalmente el usuario podrá cerrar sesión.

Requerimientos funcionales

RF1	
Descripción	El programa debe poseer una pantalla de inicio de sesión
Entradas	-
Salidas	- userActive : boolean
Precondición	- Iniciar el Programa
Postcondición	- RF3

RF2	
Descripción	El programa debe poseer una pantalla de registrarse
Entradas	
Salidas	- userActive : boolean
Precondición	- crear usuario desde RF1
Postcondición	- RF3

RF3	
Descripción	El programa debe poseer una pantalla de visualización de productos
Entradas	- userActive : boolean - products : []
Salidas	-
Precondición	- RF9 o RF10
Postcondición	- RF4

RF4	
Descripción	El programa debe poseer una pantalla de producto
Entradas	<ul style="list-style-type: none"> - idProduct : string - productName : string - productImg : string - productPrice : string - productAdicion : []
Salidas	-
Precondición	- click en producto de RF3-input
Postcondición	-

RF5	
Descripción	El programa debe poseer una pantalla de pago
Entradas	- order
Salidas	<ul style="list-style-type: none"> - statusOrder: boolean - orderReceived :[]
Precondición	- order no debe ser diferente de null
Postcondición	- statusOrder pasa a ser verdadero

RF6	
Descripción	El programa debe poseer un feedback visual
Entradas	- statusOrder : boolean
Salidas	-
Precondición	- RF5
Postcondición	- RF7

RF7	
Descripción	El programa debe poseer una pantalla de resumen
Entradas	<ul style="list-style-type: none"> - order : [] - idUser : int
Salidas	- newOrder : []
Precondición	- RF6
Postcondición	- añadir newOrder a listProducts

RF8	
Descripción	El programa debe poseer una pantalla de históricos
Entradas	- userActive : boolean
Salidas	-
Precondición	- listProducts : []
Postcondición	- RF2

RF9	
Descripción	El usuario debe poder iniciar sesión en el programa

Entradas	<ul style="list-style-type: none"> - user : string - password : string
Salidas	<ul style="list-style-type: none"> - userActive : boolean
Precondición	<ul style="list-style-type: none"> - Iniciar el Programa
Postcondición	<ul style="list-style-type: none"> -

RF10	
Descripción	El usuario debe poder registrarse en el programa
Entradas	<ul style="list-style-type: none"> - user : string - password : string - confirmedPassword : string - name : string
Salidas	<ul style="list-style-type: none"> - user
Precondición	<ul style="list-style-type: none"> - Iniciar el Programa
Postcondición	<ul style="list-style-type: none"> - RF2

RF11	
Descripción	El usuario puede ordenar platillos en el programa
Entradas	<ul style="list-style-type: none"> - productName : string - productImg : string - productPrice : string - productAdicion : []
Salidas	<ul style="list-style-type: none"> - order : []
Precondición	<ul style="list-style-type: none"> -
Postcondición	<ul style="list-style-type: none"> - RF5

RF12	
Descripción	El programa debe poseer una lista de platillos
Entradas	<ul style="list-style-type: none"> - name : string - price : int - img : string
Salidas	<ul style="list-style-type: none"> - products : []
Precondición	<ul style="list-style-type: none"> - RF3
Postcondición	<ul style="list-style-type: none"> -

RF13	
Descripción	El programa debe poseer una lista de adicionales
Entradas	<ul style="list-style-type: none"> - name : string - price : int - img : string
Salidas	<ul style="list-style-type: none"> - adds : []
Precondición	<ul style="list-style-type: none"> - RF4
Postcondición	<ul style="list-style-type: none"> -

RF14	
Descripción	El usuario puede adicionar elementos a su platillo
Entradas	- idProduct : string - userActive : string
Salidas	- productAddiction : []
Precondición	- RF4
Postcondición	- RF2

RF15	
Descripción	El usuario debe puede ordenar sus pedidos realizado por fecha de pedido o valor del pedido
Entradas	- listProducts : [] - idUser : int
Salidas	-
Precondición	- RF8
Postcondición	-

RF16	
Descripción	El usuario debe poder cerrar sesión
Entradas	- userState : boolean - idUser : int
Salidas	-
Precondición	- RF3
Postcondición	- RF1 - userState = false

Requerimientos no funcionales

- El programa debe ser desarrollado en JAVA
- El programa debe implementar la libreria ControlIP5
- El programa debe usar el modelo de color RGB

Entidades

- Main
- Controller
- Button
- Logic
- Usuario
- Adiciones
- Producto
- Carrito
- Pantalla
- InicioSesionPantalla
- MenuPantalla

- HistorialPantalla
- ComprarPantalla
- ProductoPantalla
- PerfilPantalla

UML

https://lucid.app/lucidchart/invitations/accept/inv_a9136141-a78a-46e2-acc4-c246058cb3d4?viewport_loc=-10%2C-10%2C1707%2C823%2C0_0