**Prueba de Caja Negra**

*“Sistema automatizado de control de Inventario”*

**Versión 2.0**

**Integrantes:**

Caizapanta Muela Tammy Amarilis

Guaiguacundo Aguirre Valeria Naomi

Pincha Llanos Estefany Anahí

Robalino Zaldumbide Alejandro Benjamín

**Fecha**

2025-01-21

# ÍNDICE

[**ÍNDICE 2**](#_m8ehctgf1r3x)

[**1. REQ 001 - Agregar Productos 4**](#_bb24fs68ngoi)

[1.1. Historia de usuario 4](#_9cqu8xs3rfl2)

[1.2. Partición de clases equivalentes 4](#_ndx045ejym9h)

[1.3. Código 5](#_t2ym05gegvzm)

[1.3.1. Lenguaje C en Code::Blocks 5](#_g6rdwnyrs50v)

[1.4. Ejecución 7](#_ej40luj2h74j)

[1.5. Pass (OK) 7](#_torgwfhpnj9f)

[1.6. No Pass (No OK) 7](#_4fjz9jvg4o55)

[**2. REQ 002 - Buscar Productos 8**](#_j43vur8wazce)

[2.1. Historia de usuario 8](#_g5g2omwq2n39)

[2.2. Partición de clases equivalentes 8](#_i1qyigx1i3zi)

[2.3. Código 8](#_aw2g23fttw2d)

[2.3.1. Lenguaje C en Code::Blocks 8](#_i1eb7unzh48i)

[2.4. Ejecución 8](#_ej8x4i73f6l9)

[2.4.1. Pass (OK) 8](#_elne3z2d6s0v)

[2.4.2. No Pass (No OK) 8](#_rrg3eqhhwnul)

[**3. REQ 003 - Entrada de productos 8**](#_r488bqgupfh4)

[3.1. Historia de usuario 8](#_qzd5rho2plix)

[3.2. Partición de clases equivalentes 9](#_pmmkg0vmqqaq)

[3.3. Código 9](#_l8mxhizhss6g)

[3.3.1. Lenguaje C en Code::Blocks 9](#_j31amzez24qi)

[3.4. Ejecución 9](#_8f8pp6jjks68)

[3.4.1. Pass (OK) 9](#_1v9ahbbztkms)

[3.4.2. No Pass (No OK) 9](#_y2yfz0ihscj)

[**4. REQ 004 - Salida de productos 9**](#_btg71l52anrr)

[4.1. Historia de usuario 9](#_jmbtdgem84o)

[4.2. Partición de clases equivalentes 10](#_frf6jyrbmfxe)

[4.3. Código 10](#_nyc43dem58kq)

[4.3.1. Lenguaje C en Code::Blocks 10](#_y5z0snx9qakq)

[4.4. Ejecución 10](#_evq43yh3uesj)

[4.4.1. Pass (OK) 10](#_iritetfk38pi)

[4.4.2. No Pass (No OK) 10](#_x8gquiz5bnej)

[**5. REQ 005 10**](#_gw2mp5vd9340)

[5.1. Historia de usuario 10](#_stw8k3kvr092)

[5.2. Partición de clases equivalentes 11](#_wjx3dz9yc4g3)

[5.3. Código 11](#_fa1lyl6umq4x)

[5.3.1. Lenguaje C en Code::Blocks 11](#_m3r1raoslkd)

[5.4. Ejecución 12](#_ep1ntg9l8b3o)

[5.4.1. Pass (OK) 12](#_w62rcdsmfick)

[5.4.2. No Pass (No OK) 12](#_zcma09mg7rqr)

[**6. REQ 006 12**](#_3gjbzr85s7pk)

[6.1. Historia de usuario 12](#_cl508yfvn5ln)

[6.2. Partición de clases equivalentes 13](#_pgtec7p4407k)

[6.3. Código 13](#_mxt34c5v1qyd)

[6.3.1. Lenguaje C en Code::Blocks 13](#_84pp3mzbwml5)

[6.4. Ejecución 13](#_nfk077el2rw9)

[6.4.1. Pass (OK) 13](#_y8yd39ji96sb)

[6.4.2. No Pass (No OK) 13](#_1qrdz8bwkyn2)

# REQ 001 - Agregar Productos

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 001 | **Usuario:** Administrador |
| **Nombre de la Historia:** Agregar productos | |
| **Prioridad:** Alta | |
| **Programador responsable:** Tammy Caizapanta | |
| **Descripción:** Se selecciona la opción para digitar el código único de cada producto y su información: nombre, proveedor, cantidad, precio. | |
| **Validación:** Se registra el producto agregado con un mensaje de confirmación de "Producto agregado exitosamente". | |

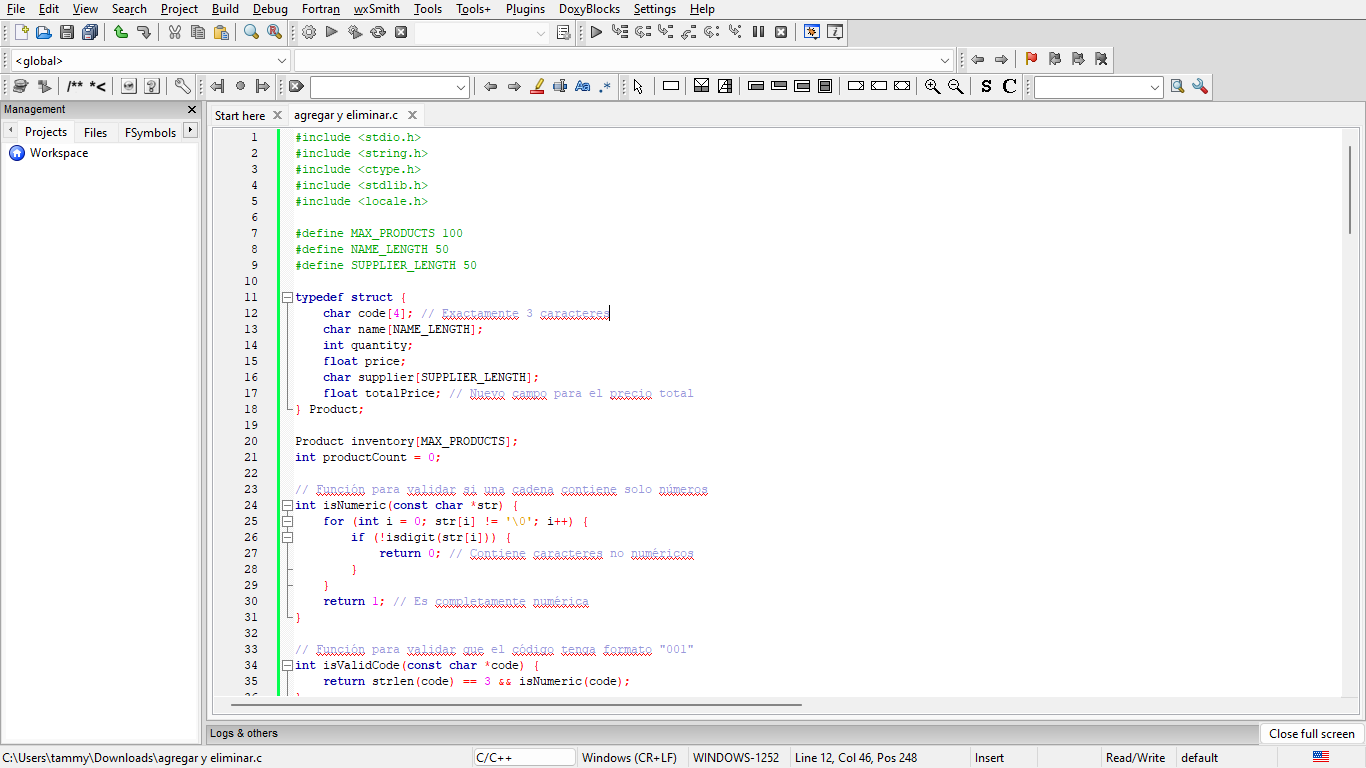
## 

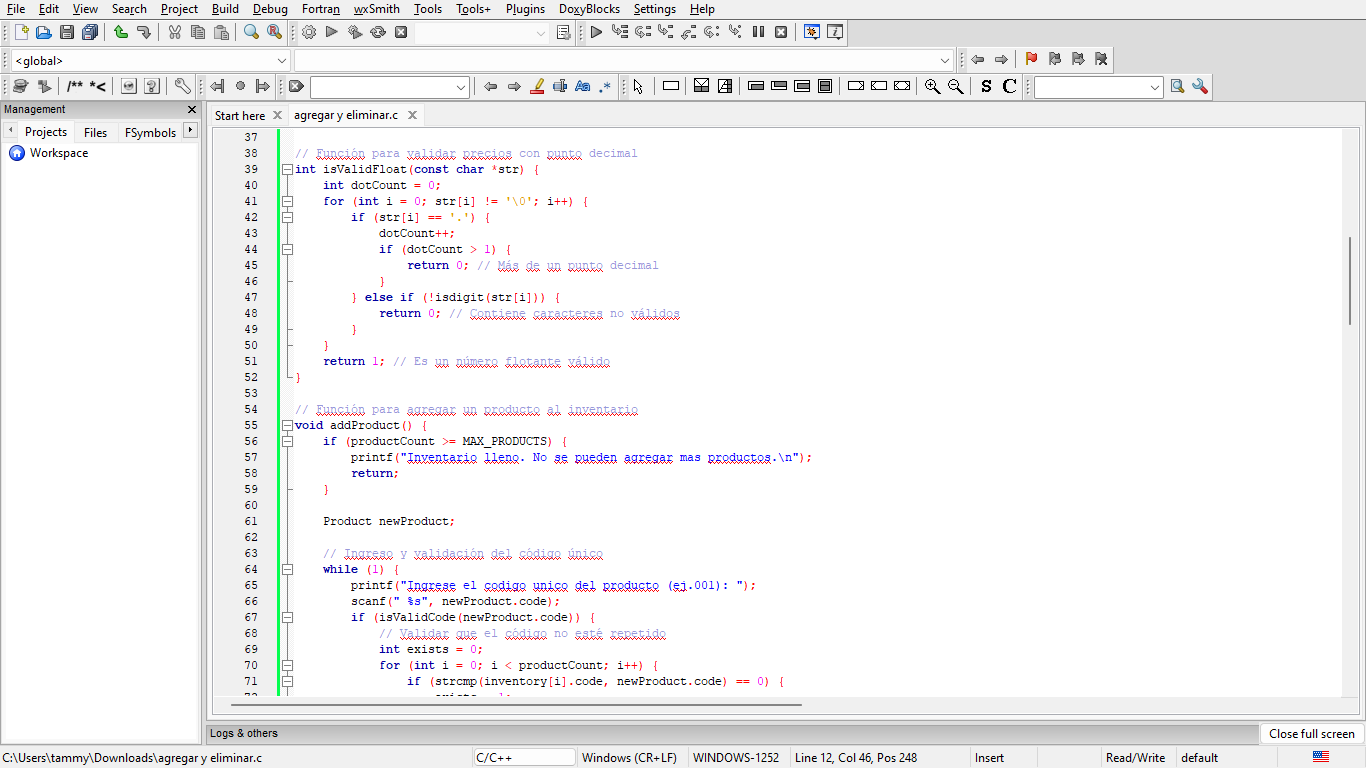
## *Partición de clases equivalentes*

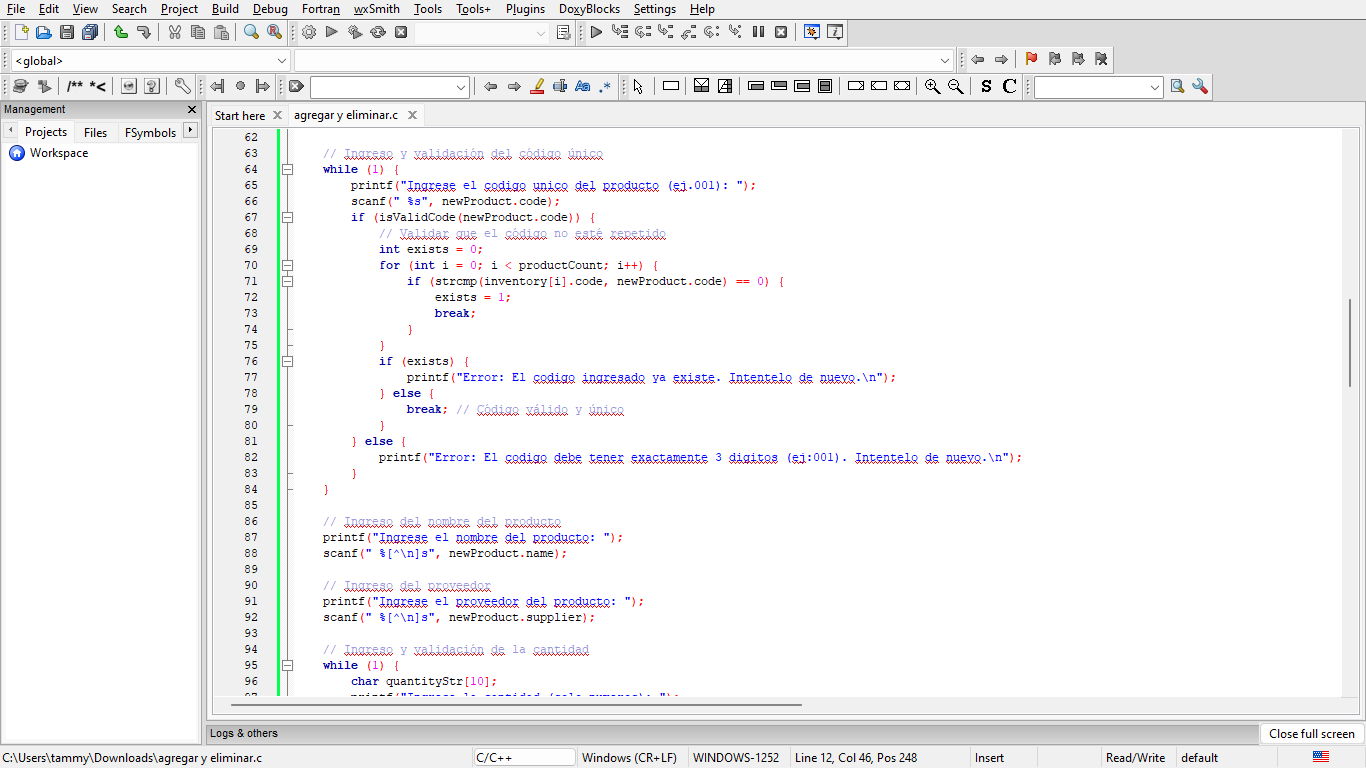
| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID | EC1:ID==ID | VÁLIDO | 001 |
| EC2:ID!=ID | NO VÁLIDO | \*\*\* |
| Nombre del producto | EC1:Name==Name | VÁLIDO | Queso |
| EC2:Name!=Name | NO VÁLIDO | - |
| Proveedor | EC1:Prov==Prov | VÁLIDO | KiosKo |
| EC2:Prov!=Prov | NO VÁLIDO | - |
| Cantidad | EC1:Cant==Cant | VÁLIDO | 7 |
| EC2:Cant!=Cant | NO VÁLIDO | siete |
| Precio | EC1:Precio==Precio | VÁLIDO | $ 2.50 |
| EC2:Precio!=Precio | NO VÁLIDO | $ 2,50 |

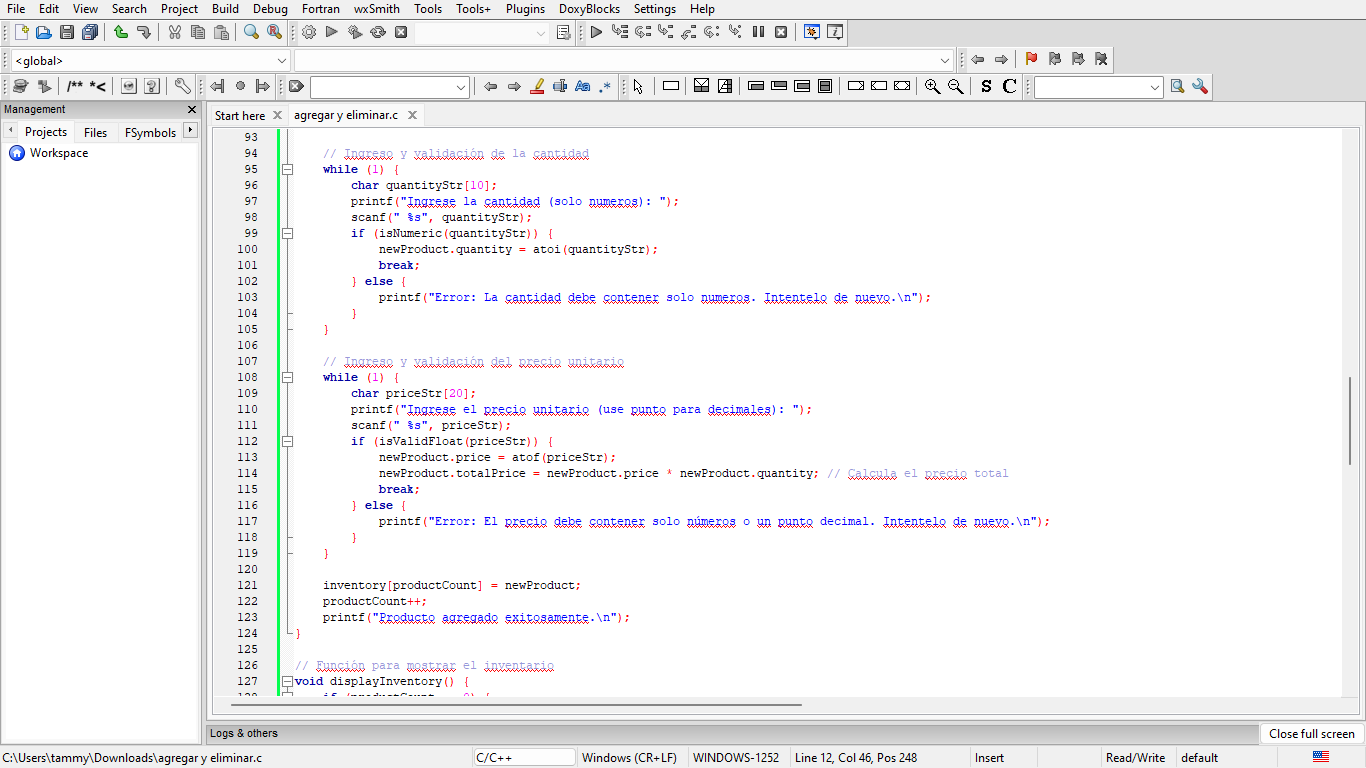
## *Código*

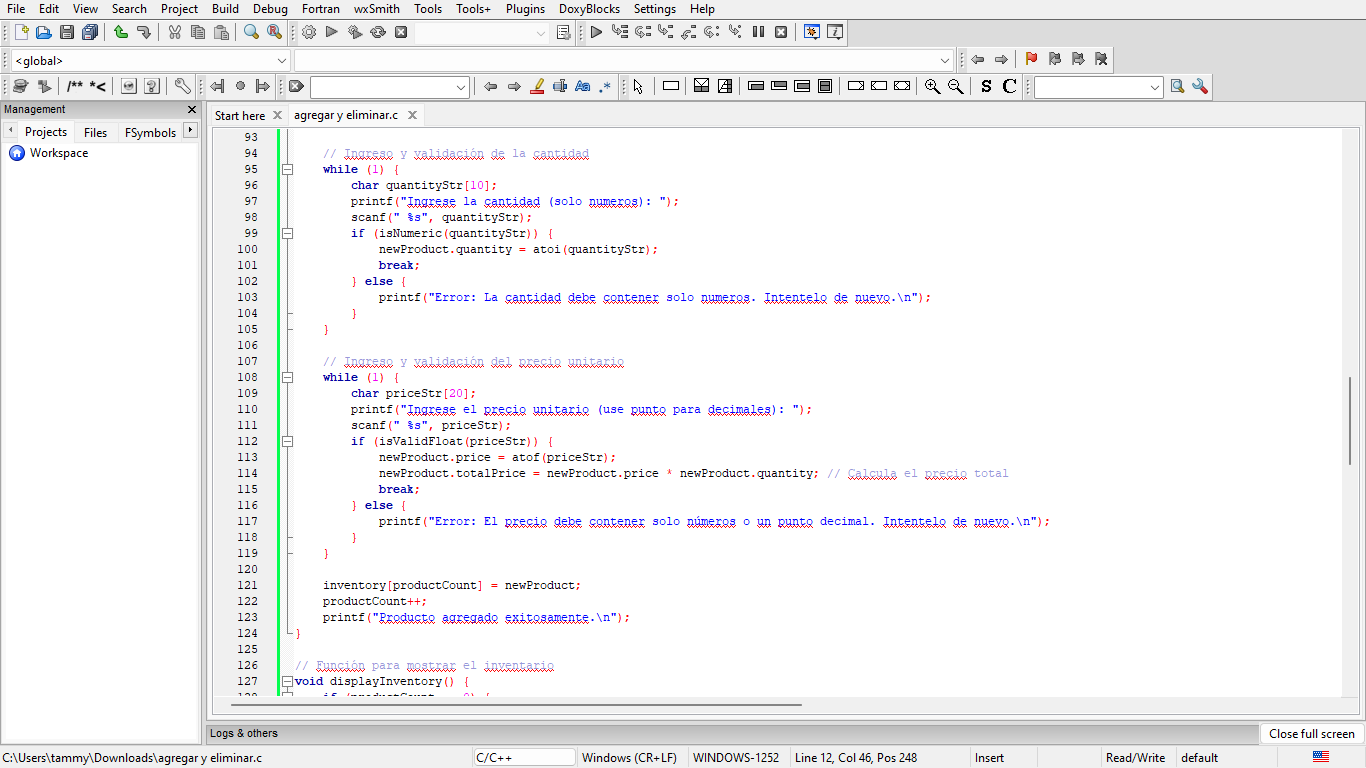
### Lenguaje C en Code::Blocks











## *Ejecución*

### Pass (OK)

## 

### No Pass (No OK)

## 

## 

## 

## 

## 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# REQ 002 - Buscar Productos

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 001 | **Usuario:** Administrador |
| **Nombre de la Historia:** | |
| **Prioridad:** | |
| **Programador responsable:** | |
| **Descripción** | |
| **Validación:** | |

## *Partición de clases equivalentes*

| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID | EC1:ID==ID | VÁLIDO | 001 |
| EC2:ID!=ID | NO VÁLIDO | \*\*\* |
| Nombre del producto | EC1:Nombre==Nombre | VÁLIDO | Queso |
| EC2:Nombre!=Nombre | NO VÁLIDO | imkthaily |
| Cantidad | EC1:Cant==Cant | VÁLIDO | 7 |
| EC2:Cant!=Cant | NO VÁLIDO | 1.5 |

## *Código*

### Lenguaje C en Code::Blocks

## *Ejecución*

### Pass (OK)

### No Pass (No OK)

# 

# REQ 003 - Entrada de productos

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 001 | **Usuario:** Administrador |
| **Nombre de la Historia:** | |
| **Prioridad:** | |
| **Programador responsable:** | |
| **Descripción** | |
| **Validación:** | |

## *Partición de clases equivalentes*

| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID | EC1:ID==ID | VÁLIDO | 001 |
| EC2:ID!=ID | NO VÁLIDO | \*\*\* |
| Nombre del producto | EC1:Nombre==Nombre | VÁLIDO | Queso |
| EC2:Nombre!=Nombre | NO VÁLIDO | imkthaily |
| Cantidad | EC1:Cant==Cant | VÁLIDO | 7 |
| EC2:Cant!=Cant | NO VÁLIDO | 1.5 |

## 

## *Código*

### Lenguaje C en Code::Blocks

## 

## *Ejecución*

### Pass (OK)

### No Pass (No OK)

# 

# 

# REQ 004 - Salida de productos

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 001 | **Usuario:** Administrador |
| **Nombre de la Historia:** | |
| **Prioridad:** | |
| **Programador responsable:** | |
| **Descripción** | |
| **Validación:** | |

## *Partición de clases equivalentes*

| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID |  | VÁLIDO |  |
|  | NO VÁLIDO |  |
| Nombre del producto |  | VÁLIDO |  |
|  | NO VÁLIDO |  |
| Cantidad |  | VÁLIDO |  |
|  | NO VÁLIDO |  |

## *Código*

### Lenguaje C en Code::Blocks

## 

## *Ejecución*

### Pass (OK)

### No Pass (No OK)

# 

# 

# REQ 005

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 005 | **Usuario:** Administrador |
| **Nombre de la Historia:** Eliminar productos | |
| **Prioridad:** Alta | |
| **Programador responsable:** Tammy Caizapanta | |
| **Descripción:** Se selecciona la opción de eliminar producto y se digita el código único de cada producto para su posterior eliminación con doble confirmación. | |
| **Validación:** Se registra la eliminación del producto con una doble confirmación de "¿Está seguro que desea eliminar este producto?" y si es afirmativo aparece el mensaje "Producto eliminado exitosamente". | |

## *Partición de clases equivalentes*

| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID | EC1:ID==ID | VÁLIDO | 001 |
| EC2:ID!=ID | NO VÁLIDO | \*\*\* |

## 

## *Código*

### Lenguaje C en Code::Blocks

## 

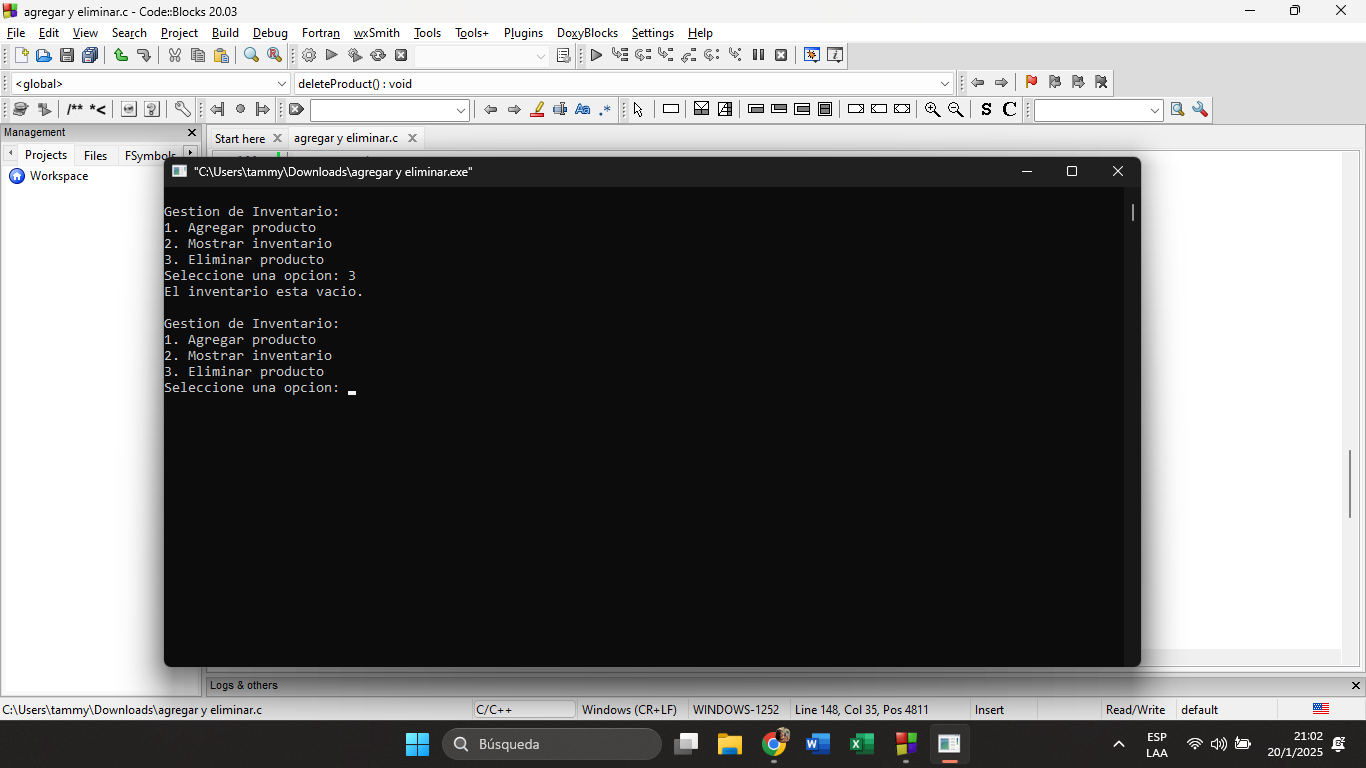
## *Ejecución*

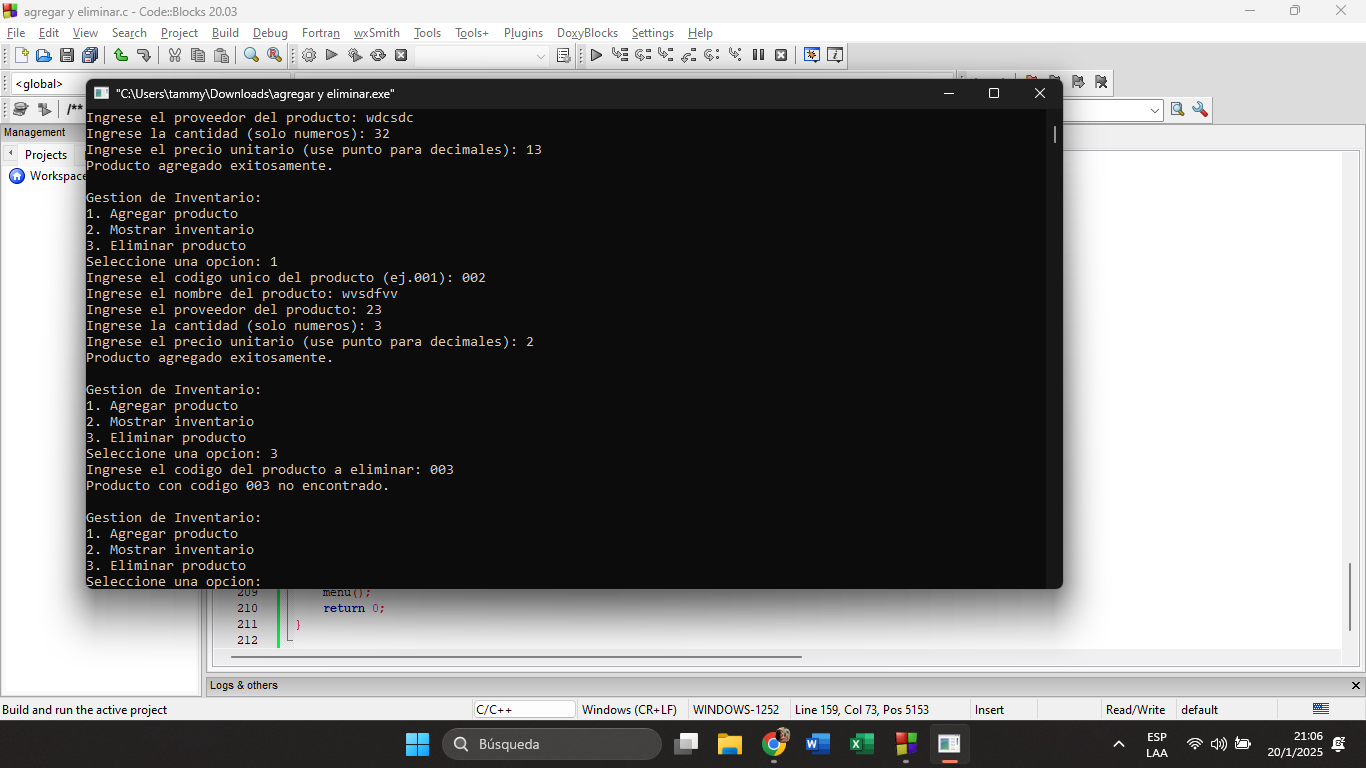
### Pass (OK)

## 

### 

### No Pass (No OK)





# 

# REQ 006

## *Historia de usuario*

| **Historia de usuario** | |
| --- | --- |
| **Número:** REQ 001 | **Usuario:** Administrador |
| **Nombre de la Historia:** | |
| **Prioridad:** | |
| **Programador responsable:** | |
| **Descripción** | |
| **Validación:** | |

## *Partición de clases equivalentes*

| **VARIABLE** | **CLASE DE EQUIVALENCIA** | **ESTADO** | **REPRESENTANTE** |
| --- | --- | --- | --- |
| ID | EC1:ID==ID | VÁLIDO | 001 |
| EC2:ID!=ID | NO VÁLIDO | \*\*\* |
| Nombre del producto | EC1:Nombre==Nombre | VÁLIDO | Queso |
| EC2:Nombre!=Nombre | NO VÁLIDO | imkthaily |
| Cantidad | EC1:Cant==Cant | VÁLIDO | 7 |
| EC2:Cant!=Cant | NO VÁLIDO | 1.5 |

## *Código*

### Lenguaje C en Code::Blocks

## 

## *Ejecución*

### Pass (OK)

### No Pass (No OK)