HITO 1 DEL 3º TRIMESTRE DE ENTORNOS DE DESARROLLO

Alejandro Cortés Díaz

CampusFP

Diseño de aplicaciones multiplataforma – ENTORNOS DE DESARROLLO  12/05/2025

Índice

[¿Qué son las pruebas estructurales? 2](#_Toc197507583)

[¿Qué son las pruebas funcionales? 2](#_Toc197507584)

[Comparación entre pruebas estructurales y funcionales 3](#_Toc197507585)

[Sistema elegido para pruebas: Aplicación de Gestión de Tareas 3](#_Toc197507586)

[Diseño de pruebas 3](#_Toc197507587)

[Pruebas estructurales (Caja Blanca) 3](#_Toc197507588)

[Pruebas funcionales (Caja Negra) 4](#_Toc197507589)

[**Enlace a GitHub** 5](#_Toc197507590)

[**Bibliografía** 6](#_Toc197507591)

## ¿Qué son las pruebas estructurales?

Las **pruebas estructurales**, también conocidas como **pruebas de caja blanca**, se centran en la **estructura interna** del código fuente. El objetivo principal es verificar que todos los caminos posibles dentro del código funcionen correctamente, es decir, **que la lógica del programa sea correcta**.

**Objetivos principales:**

* Verificar todos los flujos de control posibles del código.
* Garantizar la cobertura de instrucciones, decisiones y condiciones.
* Identificar errores en condiciones lógicas, bucles, llamadas a funciones, etc.

**Técnicas comunes:**

* **Cobertura de código:** mide qué porcentaje del código ha sido ejecutado durante las pruebas.
* **Pruebas de flujo de control:** analizan el flujo de ejecución (if, while, for, etc.).
* **Pruebas de caminos lógicos:** prueban todos los caminos posibles del programa.

## ¿Qué son las pruebas funcionales?

Las **pruebas funcionales**, también llamadas **pruebas de caja negra**, evalúan si el software **cumple con los requisitos funcionales especificados**, sin tener en cuenta cómo está implementado internamente.

**Objetivos principales:**

* Comprobar que las funcionalidades funcionan según lo esperado.
* Validar entradas, salidas y comportamiento del sistema.
* Identificar errores en la lógica de negocio desde el punto de vista del usuario.

**Técnicas comunes:**

* **Pruebas de caja negra:** validan la funcionalidad sin mirar el código.
* **Pruebas de aceptación del usuario:** verifican que el software satisface las necesidades del cliente.
* **Pruebas de regresión:** aseguran que nuevas funcionalidades no rompen las existentes.

## Comparación entre pruebas estructurales y funcionales

* Mientras que el enfoque de la caja blanca es la lógica interna del código, la caja negra se centra en las funcionalidades externas del sistema.
* La caja blanca tiene acceso al código fuente mientras que la negra no.
* La caja blanca está basada en la implementación mientras que la negra en los requisitos de usuario.
* La caja blanca hace uso de detalles referentes al código fuente mientras que la caja negra no tiene mucho conocimiento sobre el diseño interno del elemento.

## Sistema elegido para pruebas: **Aplicación de Gestión de Tareas**

La app permite:

* Crear tareas con título, descripción y fecha límite.
* Editar tareas existentes.
* Marcar tareas como completadas.
* Eliminar tareas.

## Diseño de pruebas

### Pruebas estructurales (Caja Blanca)

#### 🧪 Caso 1: Cobertura de instrucciones

* **Objetivo:** Asegurar que todas las líneas del método crearTarea() se ejecutan.
* **Procedimiento:** Ejecutar el método con datos válidos (título y fecha).
* **Resultado esperado:** Todas las instrucciones dentro del método deben ejecutarse sin errores.

#### 🧪 Caso 2: Cobertura de decisiones

* **Objetivo:** Verificar todos los caminos de un if (fecha.isAfter(hoy)).
* **Procedimiento:** Probar el método con una fecha pasada y otra futura.
* **Resultado esperado:** Se debe ejecutar tanto el camino verdadero como el falso.

#### 🧪 Caso 3: Cobertura de bucles

* **Objetivo:** Verificar que el bucle que recorre una lista de tareas funciona con 0, 1 y varias tareas.
* **Procedimiento:** Ejecutar el método con listas de diferentes tamaños.
* **Resultado esperado:** El bucle debe ejecutarse correctamente en todas las condiciones.

### Pruebas funcionales (Caja Negra)

* Caso de prueba funcional: Ingresar correo electrónico válido en formulario
* El correo electrónico ha de tener;
  + Entre 3 y 20 caracteres.
  + Sólo contener letras, números, y guión bajo “\_”.
    - El dominio debe tener entre 5 y 18 caracteres.
      * La extensión del dominio debe ser .com, .es, o .net
* La contraseña ha de tener entre 5 y 15 caracteres
* La contraseña ha de incluir una mayúscula, una minúscula, y un número
* TABLAS:

|  |  |  |
| --- | --- | --- |
| Clases de equivalencia | Clases válidas | Clases inválidas |
| Correo de 3 a 20 carac | Correo>=3 && Correo<=20 1 | Correo <3 || Correo >20 7 |
| Correo Solo puede contener letras, números y guion bajo | Correo =[a-z,A-Z,0-9,\_] 2 | Correo!=[a-z,A-Z,0-9,-] 8 |
| Dom de 5 a 18 carac | Dom >=5 && Dom <=18 3 | Dom <5 || Dom >18 9 |
| La extensión del dominio debe ser ".com", ".es" o ".net" | Ext=[.com||.es||.net] 4 | Ext!=[.com||.es||.net] 10 |
| pass de 5 a 15 carac | pass>=5 && pass<=15 5 | pass<5 || pass>15 11 |
| pass minimo 1 mayus, 1 minus y 1 num | pass==[A\_Z,a-z,0-9] 6 | pass!==[A\_Z,a-z,0-9] 12 |

|  |  |  |
| --- | --- | --- |
| Prueba | Clases que cumple | Resultado |
| [Cortes512@hotmail.net](mailto:Cortes512@hotmail.net)  Password;  Ct43xl | 1,2,3,4,5,6 | OK |
| [Cortes125?@gmail.com](mailto:Cortes125?@gmail.com)  Password;  Cortessxl6 | 1,8,3,4,5,6 | ERROR |
| [CortesL@mail.es](mailto:CortesL@mail.es)  Password;  Op24koi | 1,2,9,4,5,6 | ERROR |
| [CortesL@gmail.camion](mailto:CortesL@gmail.camion)  Password;  Op24koi | 1,2,3,10,5,6 | ERROR |
| [CortesL@gmail.camion](mailto:CortesL@gmail.camion)  Password;  Ahnehnlaieuyui123412 | 1,2,3,4,11,6 | ERROR |
| [CortesL@gmail.camion](mailto:CortesL@gmail.camion)  Password;  OpiKaxm | 1,2,3,4,5,12 | ERROR |
| [C2@hotmail.net](mailto:C2@hotmail.net)  Password;  Op24koi | 7,2,3,4,5,6 | ERROR |

# **Enlace a GitHub**

<https://github.com/Cortes-cmd/Entornos-de-Desarrollo.git>

# **Bibliografía**

Bibliografía

Boada, D. (2022, diciembre 6). *¿Qué es un entorno de desarrollo y en qué se diferencia de un entorno de desarrollo integrado (IDE)?* Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-un-entorno-de-desarrollo>

*ChatGPT*. (s/f). Chatgpt.com. Recuperado el 25 de enero de 2025, de <https://chatgpt.com>

(S/f). Amazon.com. Recuperado el 25 de enero de 2025, de <https://aws.amazon.com/es/what-is/ide/#:~:text=Un%20entorno%20de%20desarrollo%20integrado%20(IDE)%20es%20una%20aplicación%20de,una%20aplicación%20fácil%20de%20usar>.