

# CAPÍTULO 5

## PRUEBAS

## 5 Pruebas

Las pruebas son una de las partes más importantes del desarrollo de software y cualquier metodología las incluye. Estas son útiles para descubrir defectos, tanto de código, como en la lógica o estructura propia del sistema.

Las pruebas nos definirán si el sistema cumple con los objetivos para los cuales fue creado. No importa el tipo de sistema que se este creando este se puede probar mediante dos aspectos:

**Por código:** Es decir su lógica y estructura interna; se necesita conocer la función para la cual fue creado cada módulo e identificar que cada uno de estos funcione bien o tenga alguna utilidad.

**Producto terminado:** Se centra en la funcionalidad del sistema, que es lo que pide el sistema y cual es su respuesta.

A la prueba por código se le llama prueba de caja blanca y a la de producto terminado, prueba de caja negra.

### 5.1 Pruebas de Caja Blanca

**Pruebas de unidad y de integración:** La prueba de unidad, centra el proceso de verificación en la menor unidad del sistema. Los puntos que comúnmente se tocan para esta prueba son:

**Interfaz:** Que la información fluya de manera adecuada de un punto a otro.

**Estructuras de datos:** Asegurar que los datos temporales conservan su integridad durante la ejecución del algoritmo.

**Condiciones límite:** El módulo debe funcionar correctamente en los límites establecidos como restricciones.

**Caminos independientes:** Todas las sentencias del módulo se debe de ejecutar por lo menos una vez.

También se diseñan pruebas para detectar errores en las estructuras de datos locales como:

- Tipos de datos erróneos o inconsistentes
- Cantidades mal, variables no inicializadas
- Nombre de variables incorrectos
- Desbordamiento de memoria

Errores debidos a cálculos o comparaciones incorrectos o flujos de control inadecuados:

- Comparaciones entre tipos de datos diferentes
- Operadores lógicos o de precedencia incorrectos
- Variables o comparadores incorrectos
- Terminación inapropiada o inexistente de ciclos

Por lo que se refiere a las pruebas de integración, estas se establecen para construir la estructura del programa, mientras que, simultáneamente se realizan pruebas para detectar errores asociados con la interacción. Para unir los módulos se debe seguir la técnica denominada integración incremental, que consiste en probar pequeños segmentos para que los errores sean más fáciles de aislar y de corregir.

## 5.2 Pruebas de Caja Blanca realizadas

**Tipos de datos erróneos o inconsistentes:** en esta prueba, se envía un arreglo de valores y a la hora de recibirlos no se imprimen de manera adecuada.

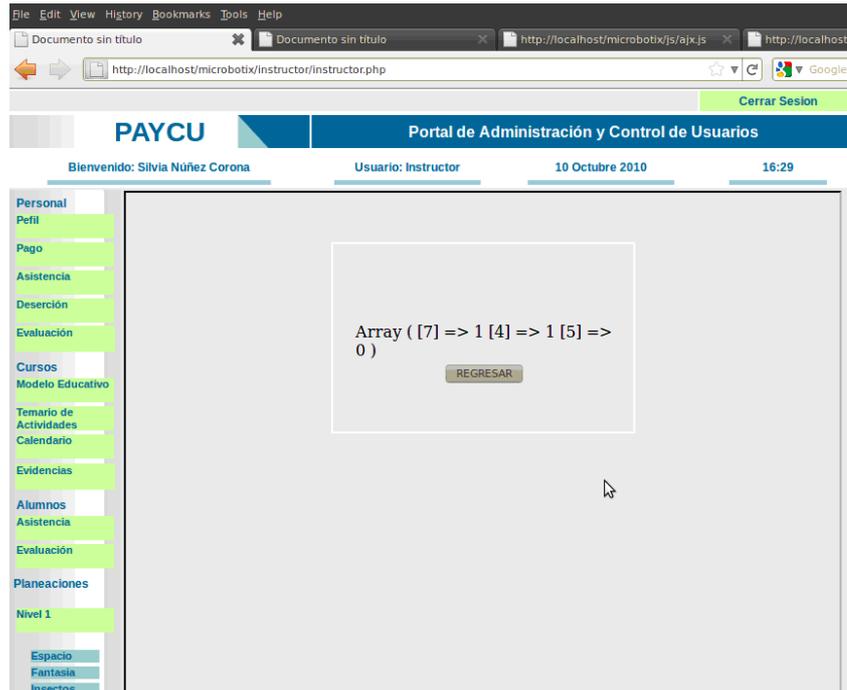


Figura 81 Dato erróneo

Nombre de variable incorrecta o no inicializada en el almacenamiento de los datos.

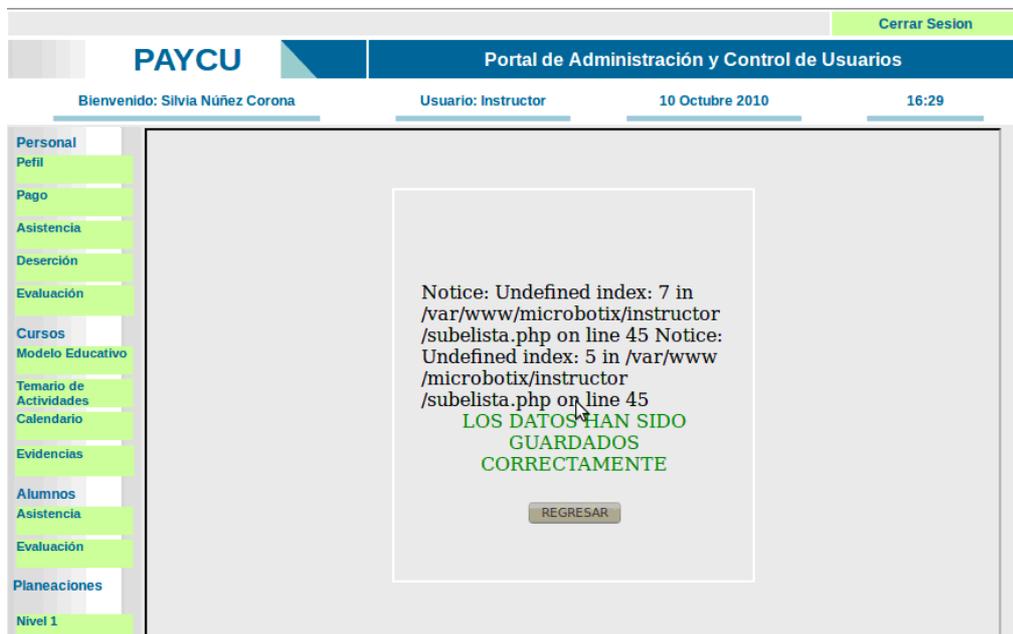


Figura 82 Variable incorrecta

Error de sintaxis en el reporte de asistencia, aparece cuando se escribe mal una sentencia o cuando PHP esperaba un parámetro y no se le pasa.

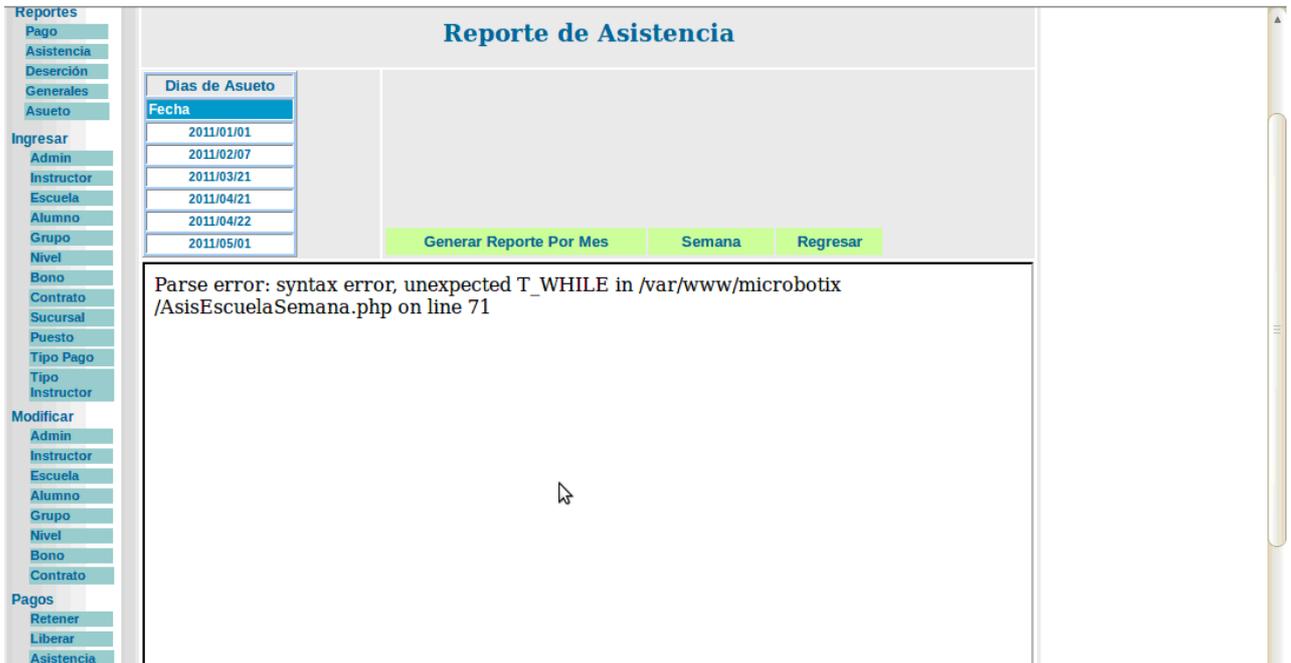


Figura 83 Error de Sintaxis

Variable no definida a la hora de subir la lista de asistencia, interfaz del instructor.

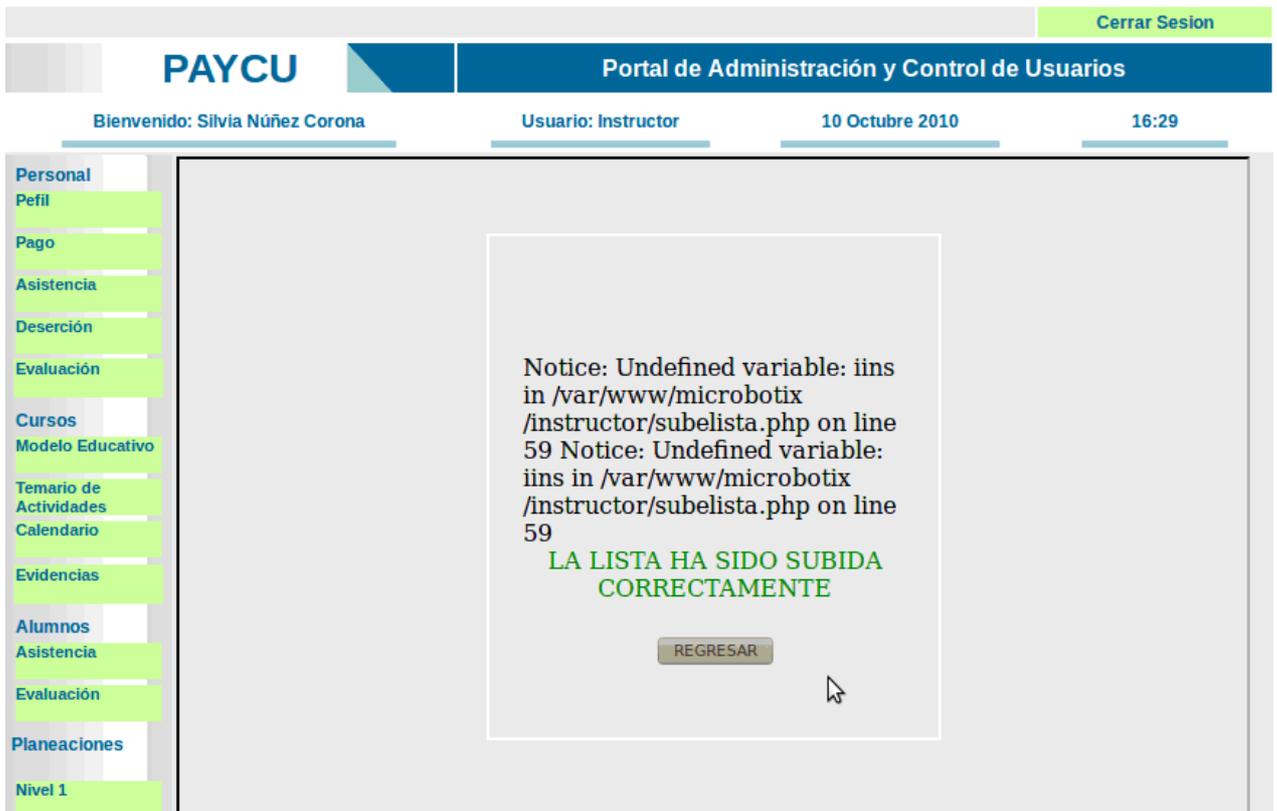


Figura 84 Variable no definida

## 5.2 Pruebas de Caja Negra

**Pruebas de Volumen:** La prueba de volumen intenta detectar errores en el software debidos a condiciones anormales, demandando recursos en cantidades, frecuencias o volúmenes fuera del rango que se especifico en el diseño. Algunos ejemplos de pruebas de volumen serían:

- Aumentar la cantidad de datos almacenados
- Aumentar el número de usuarios al portal y peticiones simultáneas
- Forzar el máximo de memoria o de otros recursos

**Pruebas Funcionales o de validación.** Se consigue al demostrar la conformidad con los requisitos, es decir, que el software funciona de acuerdo con las expectativas dadas por el cliente.

Cuando se termina un sistema es necesario que el cliente pruebe y revise el sistema con el fin de validar cada uno de los módulos y establezca que esta conforme con el y si satisfacen sus requisitos.

Para realizar esta prueba se necesita que el sistema ya este terminado e instalado para su uso, posteriormente se les capacita a los usuarios para que accedan al sistema e interactúen con el a fin de detectar posibles fallas lógicas, de concepto, funcionales de interpretación, faltantes y facilidad de uso, entre otras.

### Pruebas de caja Negra realizadas

Hay una liga rota, el archivo no se encuentra en esa dirección, o se escribió mal.

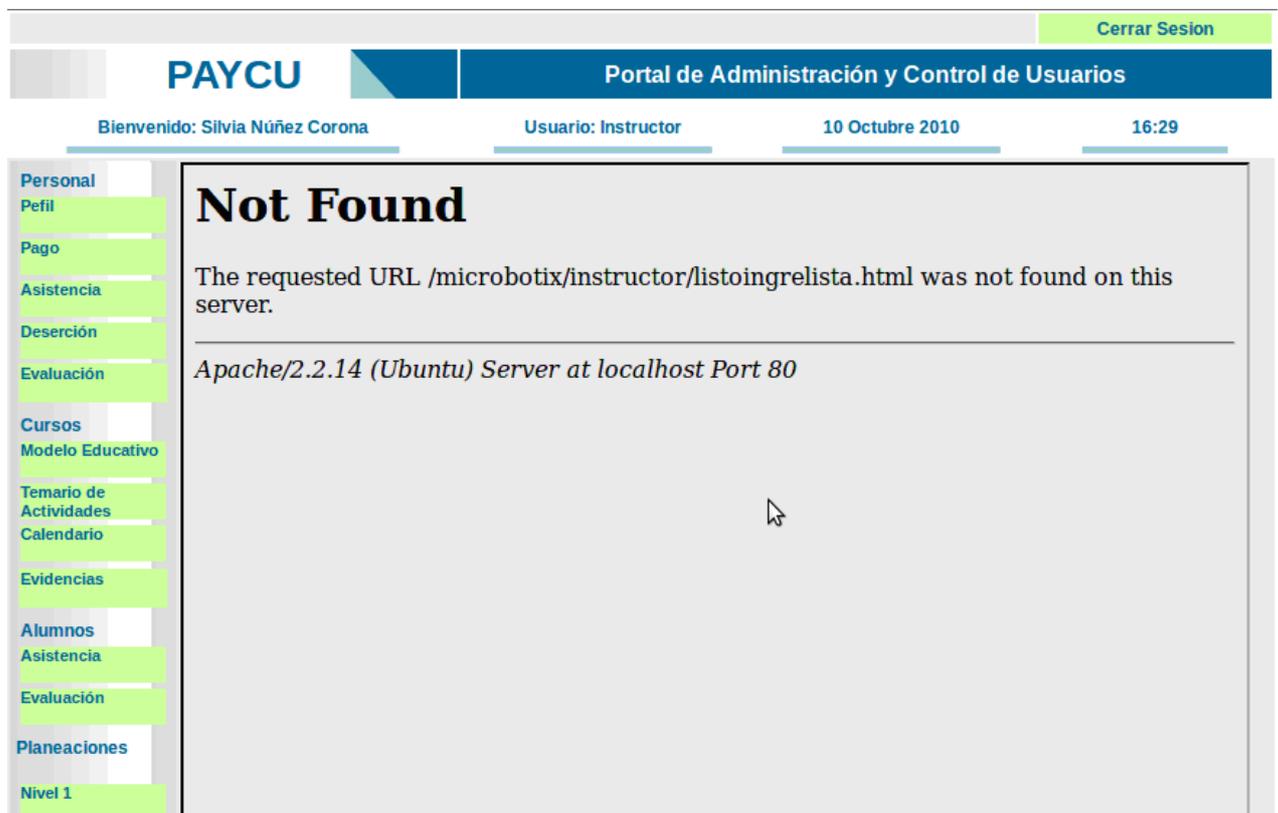


Figura 85 No se encuentra el Archivo

No inició sesión y aun así le muestra la información.

The screenshot shows the PAYCU portal interface. At the top right, there is a 'Cerrar Sesión' button. The header includes the PAYCU logo and the text 'Portal de Administración y Control de Usuarios'. Below the header, the user is identified as 'Bienvenido: Silvia Núñez Corona', 'Usuario: Instructor', and the date is '10 Octubre 2010' at '16:29'. A sidebar on the left contains various menu items like 'Personal', 'Cursos', and 'Alumnos'. The main content area displays a PHP error message: 'Notice: session\_start(): ps\_files\_cleanup\_dir: opendir(/var/lib/php5) failed: Permission denied (13) in /var/www/microbotix/instructor/pasandoList.php on line 2'. Below the error is a section titled 'Lista de asistencia' with a table showing attendance data for three students: ARIANZA PEREZ NATALIA, FLORES NUÑEZ MARIA, and ROBLEDO ESTRADA ROBERTO. The table includes columns for 'Grupo', 'Escuela', 'Horario', 'Nivel', 'Apellido Paterno', 'Apellido Materno', 'Nombre', 'Fecha', 'Asistencia', 'Falta', and 'Baja'. There are 'Limpiar' and 'Subir' buttons below the table.

Grupo	Escuela	Horario	Nivel
1	COLEGIO IRLANDES	14:00 - 15:00	2

	Apellido Paterno	Apellido Materno	Nombre	Fecha	Asistencia	Falta	Baja
1	ARIANZA	PEREZ	NATALIA	2011/03/21	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2	FLORES	NUÑEZ	MARIA	2011/03/21	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3	ROBLEDO	ESTRADA	ROBERTO	2011/03/21	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 86 Error de Sesión

No puede abrir el archivo al que hace referencia y no se tomo en cuenta esta advertencia para mandar un mensaje de error adecuado para el usuario.

The screenshot shows the PAYCU portal interface. At the top right, there is a 'Cerrar Sesión' button. The header includes the PAYCU logo and the text 'Portal de Administración y Control de Usuarios'. Below the header, the user is identified as 'Bienvenido: Silvia Núñez Corona', 'Usuario: Instructor', and the date is '10 Octubre 2010' at '16:29'. A sidebar on the left contains various menu items like 'Personal', 'Cursos', and 'Alumnos'. The main content area displays a PHP warning message: 'Warning: require\_once(limpiacadena.inc): failed to open stream: No such file or directory in /var/www/microbotix/instructor/subelista.php on line 34 Fatal error: require\_once(): Failed opening required 'limpiacadena.inc' (include\_path=.:usr/share/php:usr/share/pear) in /var/www/microbotix/instructor/subelista.php on line 34'. The error message is displayed in a white box with a black border.

Figura 87 Referencia no validada