# Documento de Pruebas – Creación de Usuario

**Nombre del Proyecto:** TravelDesk – Plataforma Web de Itinerarios Turísticos  
**Módulo:** Autenticación – Registro de Usuario  
**Versión:** 1.0  
**Responsable:** Equipo de Desarrollo  
**Fecha:** 09/10/2025

## Objetivo de la Prueba

Validar el correcto funcionamiento del módulo de **registro de usuarios** de la aplicación TravelDesk, comprobando que el formulario del frontend (React) se comunique correctamente con el backend (Express + MySQL), que las validaciones se apliquen de forma adecuada y que los datos se almacenen correctamente en la base de datos.

## Precondiciones

* El servidor backend (Node.js + Express) debe estar en ejecución en http://localhost:3000/.
* El frontend (Vite/React) debe estar en ejecución en http://localhost:5173/.
* La conexión con la base de datos MySQL debe estar activa y funcional (itinerarios\_turisticos).
* El usuario que se desea registrar **no debe existir previamente** en la base de datos.
* Se debe contar con conexión estable a Internet (para pruebas locales o remotas).

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Datos de Entrada

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo de Dato** | **Ejemplo de Valor** | **Validación Esperada** |
| Nombre | Texto | Shanks | Solo letras y espacios, no vacío. |
| Apellido | Texto | Dragon | Solo letras y espacios, sin números ni caracteres especiales. |
| Correo Electrónico | Email | shank@one.piece | Formato válido de correo electrónico. |
| Usuario | Texto | Shank | Entre 4 y 20 caracteres, sin espacios. |
| Contraseña | Texto (oculto) | Shank123- | Mínimo 8 caracteres, debe incluir mayúsculas, minúsculas, número y símbolo. |
| Confirmar Contraseña | Texto (oculto) | Shank123- | Debe coincidir con la contraseña. |
| Fecha de Nacimiento | Fecha | 01/01/2000 | El usuario debe tener más de 13 años. |
| Género | Lista | Masculino | Debe seleccionar un valor. |
| Términos y Condiciones | Checkbox | Marcado | Obligatorio para enviar el formulario. |

## Pasos para Ejecutar la Prueba

1. Iniciar el servidor backend con npm run dev desde la carpeta backend/.
2. Iniciar el frontend con npm run dev desde la carpeta frontend/.
3. Abrir el navegador y acceder a http://localhost:5173/registro.
4. Completar todos los campos del formulario con datos válidos.
5. Presionar el botón **“Registrarse”**.
6. Verificar en la consola del navegador si se envía la solicitud correctamente (fetch POST /api/registro).
7. Confirmar en la base de datos MySQL que se ha insertado el nuevo usuario.
8. Probar con valores inválidos (por ejemplo, sin contraseña o con correo duplicado) para validar mensajes de error.

## Resultado Esperado

* El sistema muestra el mensaje: **“Usuario registrado correctamente”** o **“Cuenta creada con éxito”**.
* El nuevo usuario aparece registrado en la tabla usuarios de la base de datos.
* No se producen errores de conexión ni fallos de validación inesperados.
* El formulario se limpia o redirige al login tras el registro exitoso.
* En casos de error (correo duplicado, contraseñas no coinciden, campos vacíos), el sistema muestra mensajes claros sin romper la interfaz.

## Resultado Obtenido

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Descripción del Caso** | **Resultado Obtenido** | **Estado** |
| 1 | Registro con datos válidos | El usuario se crea correctamente y se confirma en la BD. | Aprobado |
| 2 | Registro con correo existente | El sistema muestra “Correo ya registrado”. | Aprobado |
| 3 | Campos vacíos o inválidos | El sistema muestra errores de validación sin enviar datos. | Aprobado |
| 4 | Contraseñas no coinciden | Se muestra error y no se crea el usuario. | Aprobado |
| 5 | Omisión de términos | El sistema impide el envío y muestra error correspondiente. | Aprobado |

## Observaciones

* Se recomienda **mostrar un Toast o Alert visual** tras el registro exitoso.
* Incluir **validaciones visuales dinámicas** (colores o íconos) para mejorar la UX.
* Se sugiere **hashear la contraseña** con bcrypt antes del almacenamiento.
* Probar el flujo de registro en navegadores distintos (Chrome, Edge, Firefox).
* Realizar pruebas complementarias en dispositivos móviles (responsividad del formulario).

### **8. Evidencias de las Pruebas**

📷 **Caso 1 – Registro exitoso**

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.  
Figura 1.1: Formulario completado con datos válidos y mensaje “Registro exitoso”.

📷 **Caso 2 – Correo ya registrado**

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.  
Figura 1.2: Mensaje de error al intentar registrar un correo duplicado.

📷 **Caso 3 – Validaciones de campos**

Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.  
Figura 1.3: Campos resaltados en rojo cuando faltan datos obligatorios.

📷 **Caso 4 – Contraseñas distintas**

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
Figura 1.4: Mensaje de validación “Las contraseñas no coinciden”.

📷 **Caso 5 – Omisión de términos**  
Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

*Figura 1.5: Mensaje de validación “Debes aceptar los términos”.*