

## **Pruebas Manuales**

Las pruebas manuales son un tipo de testing de software donde el tester ejecuta casos de prueba manualmente, sin el uso de herramientas automatizadas. Se usan para validar que una aplicación funciona según los requisitos establecidos.

### **Objetivo**

Detectar errores, defectos o comportamientos no deseados en el software, asegurando que el sistema cumple con lo esperado por el cliente o usuario final.

### **Características**

- No requiere programación.
- Se basa en la experiencia del tester.
- Requiere documentación clara: casos de prueba, pasos a seguir, datos de entrada, resultados esperados.
- Es flexible ante cambios de requisitos.

### **Ventajas**

- Buena opción para proyectos pequeños o en fases iniciales.
- Permite detectar errores visuales o de usabilidad.
- Favorece la validación subjetiva y experiencia del usuario.
- Útil para pruebas exploratorias o ad-hoc.

### **Desventajas**

- Es más lento que el testing automatizado.
- Mayor probabilidad de errores humanos.
- Difícil de escalar para grandes volúmenes de pruebas.
- No reutilizable ni repetible fácilmente.

## Tipos comunes de pruebas manuales

1. **Pruebas de humo (Smoke testing):** Validan si las funciones básicas del sistema funcionan.
2. **Pruebas de regresión:** Verifican que nuevas versiones del software no hayan introducido errores en funcionalidades previamente correctas. Desde mi experiencia personal normalmente suelen estar automatizadas.
3. **Pruebas de aceptación (UAT):** Hechas por el cliente o usuario para validar si el sistema cumple sus necesidades.
4. **Pruebas exploratorias:** No siguen un guion, se basan en la intuición y experiencia del tester.

## Ciclo básico de una prueba manual

1. Entender los requisitos del sistema.
2. Diseñar los casos de prueba.
3. Preparar el entorno de pruebas.
4. Ejecutar los casos manualmente.
5. Registrar resultados y capturar evidencias.
6. Reportar bugs con detalle y claridad.
7. Repetir pruebas si es necesario (verificación de corrección, solución de bugs, etc...).

## Herramientas útiles para pruebas manuales

- **Jira o Azure DevOps** para gestión de planes de pruebas y ciclos de vidas de los bugs. Debido a mi experiencia y la importancia de estas herramientas he decidido crear una documentación en especial para Azure.
- **TestRail o Xray** para gestionar casos de prueba.

**TestRail:**

- Definición y organización de casos de pruebas:

The screenshot shows the TestRail 'Test Cases' page. The top navigation bar includes tabs for OVERVIEW, TODO, MILESTONES, TEST RUNS & RESULTS, TEST CASES (active), and REPORTS. Below the navigation bar, there's a 'Test Cases' header with various action buttons like 'View Test Data', 'Shared Test Steps', 'Defects', 'Reports', and 'Run Test'. A filter bar allows sorting by 'Section' and includes a 'Display Deleted Test Cases' toggle. The main table lists test cases with columns for ID, References, Title, Automation Type, Template, and Assigned To. The 'Login' section is expanded, showing a sub-section 'Login - Unhappy Path' with a table of test cases. On the right, a sidebar shows a hierarchical tree of sections and subsections, including 'Login', 'Login - Unhappy Path', 'Download and Installation', 'Contacts', 'Messaging', 'Image Messages', 'Reactions', 'Replies', 'Voice Message', 'UI Testing', 'Hardware Integration', 'Security', 'Networking', 'Device Interactions', 'Battery Management', 'Data Management', 'Background Tasks', and 'Memory Management & Res'.

ID	References	Title	Automation Type	Template	Assigned To
C7322	TRM-42	Login page loads from homepage	Playwright	Test Case (Text)	Heather V.
C7453	TRM-1	Login page loads when app-switching	Playwright	Test Case (Text)	Heather V.
C7323	GMS-1, GMS-2, GMS-3	Username and password text boxes allow for text input	Playwright	Test Case (Text)	Heather V.
C4730	TRM-42	User can login with correct credentials	Playwright	Test Case (Text)	Heather V.

ID	References	Title	Automation Type	Template	Assigned To
C7066	TRM-42	Descriptive Error Message: Incorrect username	Playwright	Test Case (Text)	Heather V.

**Add Test Case**

Master

**Title \***

Login

Section *	Template *	Type *	Priority *
Test Cases	Test Case (Text)	Functional	Medium

Estimate	References	Automation Type
		None

**Preconditions**

The preconditions of this test case. Reference other test cases with [C#] (e.g. [C17]).

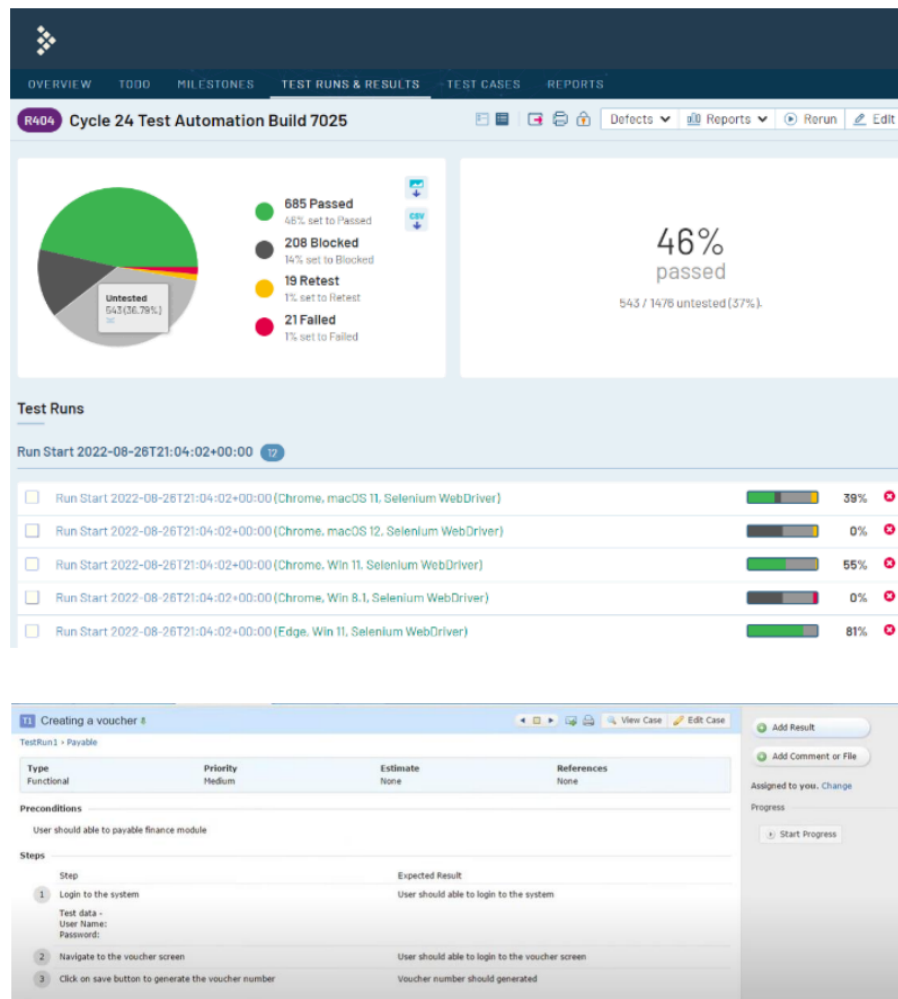
**Steps**

The required steps to execute the test case.

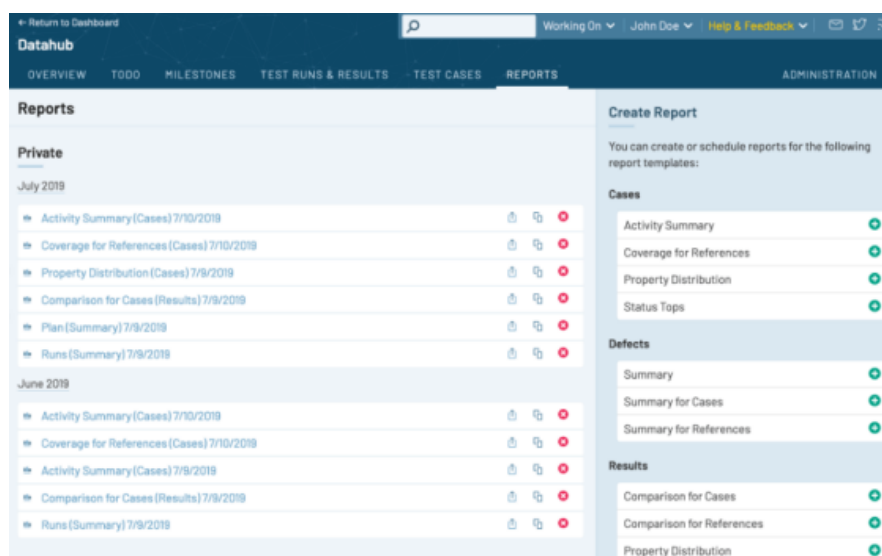
**Expected Result**

The expected result after executing the test case.

- Test Runs & Results: dentro de esta opción podemos observar las pruebas que vamos a realizar y que podemos ejecutar.



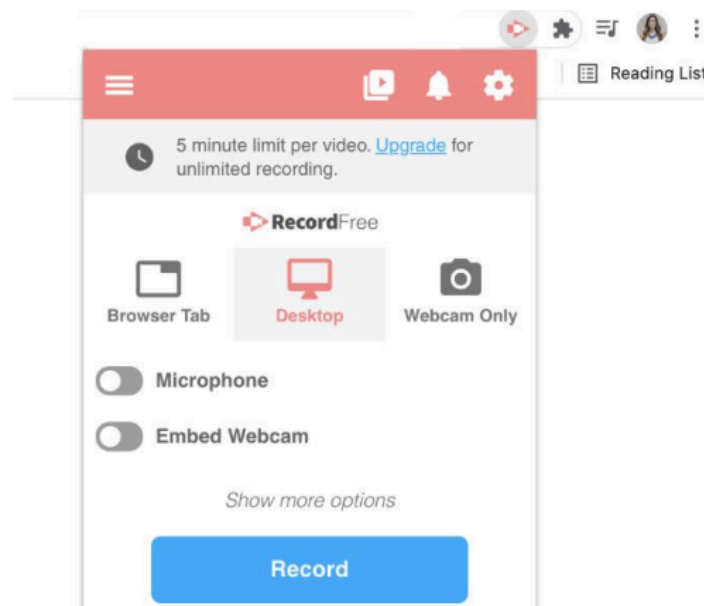
- Reports: Aquí podemos crear reportes con distintas plantillas en función de nuestras necesidades.



- **Herramientas de captura de pantalla y vídeo** como Lightshot o Screencastify.
- **Lightshot:** Es un plugin que podemos instalar en nuestro navegador Chrome que nos permite realizar capturas de pantalla de forma rápida y eficiente, además de poder editarla añadiéndole de señales y texto dando un plus a nuestra documentación.



- **Screencastify:** Es un plugin encargado de capturar pantallas de vídeo que podemos utilizar en nuestro navegador de Chrome. La versión gratuita solo puede hacer capturas de hasta 5 min



## **Buenas prácticas**

- Especificar bien los pasos de prueba, hacerlos de la forma más fácil y detallada posible, de forma que cualquier persona pueda .
- Usar datos representativos, recomendable crear baterías de usuarios.
- Probar tanto caminos esperados como situaciones límite.
- Documentar todo: resultados, errores, pasos seguidos.
- Colaborar con desarrolladores y analistas para entender bien el sistema, especial énfasis en este punto, ya que como QAs nuestra labor no es solo reportar defectos, sino prevenirlos .