

# Historias de Usuario

■ Unidad	Unidad 1
■ tipo	teórico

## Caso de Uso vs Historia de usuario

Requerimientos de usuario y requerimientos funcionales

Los casos de uso describen requerimientos funcionales



Un caso de uso describe de forma técnica lo que tiene que hacer un software paso por paso. La user storie no se enfoca en como vamos a ir resolviendo paso a paso.

Partes de la historia de usuario:

- **Conversación**
- **Card - Tarjeta** → frase verbal , criterios de aceptación , pruebas de usuario, descripción : rol quien , que , para que , valor de negocio .
- **Confirmación**

El cliente es el que paga y puede que muchas veces coincidan usuario y cliente pero a veces no.

El cliente no debería ser el que tome las decisiones sino el usuario que es el que sabe la forma mas cómoda de trabajar.

## Card:

### Descripción



Como **Rol** quiero **que** para **valor de negocio**

La idea es que la card no sea circular sino que veamos realmente el valor que le otorga al negocio.

Cada historia de usuario tiene UN SOLO objetivo de negocio.

## **Criterios de aceptación**

Sirven para asegurar la calidad de la H.U

Todos los criterios que va a tener en cuenta nuestro cliente para decirnos que esa H.U le sirve.

De alguna manera nos pone límites y nos indica lo que la HU tiene que hacer. Si alguna de estos criterios falta probablemente la HU sea rechazada por nuestro cliente.

Ej : "el monto de la venta no debe de ser superior a ..."

La diferencia entre una prueba de usuario y un caso de prueba → los casos de prueba sirven para examinar la calidad del software y tienen pasos detallados con ejemplos puntuales de lo que hay que hacer. La ejecución de casos de prueba es anterior a las pruebas de usuario. Una prueba de usuario nos sirve que el usuario vea el software que estamos haciendo después de un caso de prueba y me diga si lo acepta o no.

Buscamos que las pruebas de usuario verifiquen los criterios de aceptación. Buscan probar cosas que el usuario pueda controlar.

La cantidad de pruebas de usuario que hagamos depende de la HU .

Cualquier error en las pruebas de usuario es carísimo y nos cuesta mucho mas que si encontramos un error en los casos de prueba.

Criterio de ready → es un criterio que cada equipo debe definir por si mismo. Busca definir cuando una HU esta lista para trabajar.

El modelo invest → es una base

**I → independiente**, en este momento de tiempo dado que yo voy a trabajar la HU tiene que ser independiente de cualquier otra HU que yo este trabajando . Sirve para evitar conflictos en el desarrollo

**N → Negociable**. Busca que se pueda negociar los criterios de aceptación con el cliente.

**V → Valuable o que da valor .**

**E → Estimable** . Cuanto tiempo me va a llevar

**S → Small** . Debe consumido en una iteración.

**T → Testeable**. Tengo que poder probar esa HU.

## ▼ TP 1

HU candidatas :

- ~~Registrar~~ **gastos** → guardar gastos (usamos lenguaje que se adecue mas al usuario)
- ~~Registrar~~ responsable de gastos → cargar responsable de gastos
- Visualizar planilla de gastos
- **Filtrar gastos**
- Iniciar Sesión
- Eliminar gasto
- Registrar un tipo de gasto
- Modificar gasto
- Registrar usuario
- Cerrar sesión

Las HU siempre van en lenguaje del cliente



~~Rol : Usuario~~ (No es un rol) Rol : Miembro familiar

### **Filtrar gastos**

Yo como usuario quiero que se puedan filtrar los gastos para facilitarme la visualización de los mismos .

Criterios de aceptación:

- Se debe poder filtrar por mes
- Se debe poder filtrar por tipo de gasto
- Se debe poder filtrar por rango de montos
- Se debe poder modificar el criterio de ordenamiento

Pruebas de usuario :

- Poder filtrar por un mes anteriores o iguales al actual (pasa)
- Poder filtrar por un mes posterior al actual (falla)
- Poder filtrar por un tipo de gasto existente (pasa)
- Poder filtrar por un tipo de gasto que no se encuentre registrado (falla)
- Poder filtrar por un rango valido (Pasa)
- Poder filtrar por un rango negativo (falla)
- Poder cambiar el criterio de ordenamiento por uno existe (pasa)
- Poder seleccionar un criterio de ordenamiento por uno inexistente (falla)