

## 4/4 - Práctico

### Grupo N: 6

1er TP - No evaluable

Concepto de US: Descripción de una funcionalidad desde la perspectiva del usuario.

- no tiene conceptos técnicos, lenguaje elevado.
- 3 C:
  - Conversación: Acuerdo. El PO + equipo, para definir que se espera que se haga . Guardarlo en el acceptance criteria. Acceptance criteria, debe ser claro y determinar lo requerido. **PUEDE** se refiere a lo opcional, **DEBE** es obligatorio.

Card: tarjeta, COMO **Rol**, QUIERO **actividad/accion**, PARA **valor negocio**. Raramente se pone usuario en rol, el valor se entrega al rol.

COMO[usuario concreto]

QUIERO [realizar acción concreta]

PARA [resultado o beneficio]

Scenario: Como usuario existente y habilitado del cajero,  
quiero realizar una extracción para obtener dinero.

GIVEN Me autenticqué con una tarjeta habilitada

WHEN Introduzco la tarjeta en el cajero

THEN El cajero muestra la pantalla de opciones

- 
- Confirmación: Pruebas para detallar lo no especificado, para confirmar que está completa y correctamente hecha. Se realiza la prueba de aceptación.

Ready criteria: US clara y definida de forma completa. Cuando una US está lista para ingresar al Sprint.

Done criteria: Cuando la US está lista para ser integrada como incremento.

- **INVEST**: Independiente, negociable (define **que se espera**), valor, estimable, small (pequeña), testeable (tiene existencia y se puede demostrar).

US req de negocio, usuario.

CU, req de funcionalidad.

Registro, inicio de sesión

Rol: Responsable de gastos

UserStories:

- Registrar responsable de cuenta
- Iniciar sesión
- ABMC gastos (dividir en 4 US)
- Visualizar gastos (usar el plural para desambiguar)
- Consultar / Buscar / Filtrar gastos
- ABMC tipo de gasto (dividir en 4 US)
- Cerrar sesión

COMO responsable de gastos

QUIERO registrarme en la aplicación

PARA llevar control de los gastos

Criterios de aceptacion:

- Cuenta unica (username y movil)
- Debe ingresar nombre apellido, mail, contraseña, username, fecha de nacimiento.
- Debe ser mayor de edad (+18)
- La contraseña debe tener al menos 6 caracteres alfanumericos.

Pruebas de aceptacion de usuario:

- Probar registrar cuenta ingresando todos los datos requeridos.

- Probar registrar cuenta ingresando edad menor a 18 años (Falla).
  - Probar registrar cuenta ingresando contraseña menor a 6 caracteres.
  - Probar registrar cuenta ingresando user y mail válidos,
- completando todos los datos requeridos, con edad mayor a 18 años. (Pasa)

Ejercicio:

- Registrar gastos
- Registrar tipo de gastos
- Visualizar gastos
- Registrar Familiar

## 18/04 - Práctico

3 partes de la User Story

3 partes de

Complejidad: dificultad, validaciones, cantidad de datos que se necesita

Esfuerzo: cantidad de horas lineales a dedicar

Incertidumbre:

- Negocio
- Técnica

Us patron: Complejidad simple y esfuerzo bajo, sin incertidumbre ni técnica ni funcional.

Ponerle número, generalmente 1 o 2.

## Práctico - 28/4 - SCM - TP 4

SCM Herramienta que nos sirve para asegurar que todos los items de configs están claros y especificados. Y la posibilidad de integrar a un nuevo integrante de equipo.

## **Nomenclatura de items de config**

Armado de repositorio de documentacion

Definir linea base, cuando se genera

Armado de repositorio publico de docu y con la estructura definida.

MVP - Simulacro

Validar hipotesis unica: existen personas dispuestas a enviar sus envios mediante particulares.

## **Historias que entran al mvp**

Justificación: se toman estas US por ser el core de negocio, describir el core.

Queda fuera del mvp: GPS, zona de cobertura, forma de pago, notificaciones, liquidación, 2 formas de visualizar, envío con foto, etc

“Se va a validar para este mecanismo de pago, solo el pago con entrega” por ejemplo para justificar lo que no va.

- Poner users que van
- como se van a implementar esas user para validar el mvp
- justificación de selección de users
- poner las que no van al mvp

Roles:

- Dador de carga, cadete o persona que envía
- Transportista

User identificada: Basarse en todo lo que cumpla la hipótesis.

- Registrar usuario
- Comunicar el pedido de envio
- visualizar el estado
- cotizar el envio
- seleccion de transportista
- liberacion de datos de contacto, puede estar incluida en consulta de estado del viaje.
- iniciar viaje
- finalizar el viaje
- consultar el estado de viaje
- notificaciones push/email (no va)
- listado de cotizaciones

tiene que haber inicio de sesión, cada vez que no haya hardcodeado, revisar como se registra a la persona.

Poner peso en estimaciones

User Patron: Complejidad, esfuerzo, incertidumbre, no hace falta que la us esté en el mvp, pero si tiene que estar en el dominio, es de referencia y se asocia un peso de 1.

esfuerzo, bajo, medio, alto.

# Resolución ejercicio simulacro

## User Stories para el MVP

<aside> 💡 Nota: Puse todas las historias de usuario que entrarían en el MVP, independientemente del Rol que representan, las marco con DC (dador de carga), TT(transportista), dado que si hacemos un MVP para comprobar una hipótesis que solo que tenga que ver con uno de los roles, no tiene sentido sumar las Users del otro rol al MVP.

</aside>

- Publicar pedido de envío (TT)
- Notificación a transportista email servicio aprobado (TT)
- Visualizar listado pedidos pendientes de cotización (TT)
- Cotizar pedido (TT)
- Abonar pedido mediante mercadopago (DC)
- Visualizar cotizaciones (DC)
- Contratar cotización (DC)
- Visualizar datos contacto (DC) (TT)
- Actualizar estado pedido (TT)
- Consultar estado pedido (DC)
- Registrar usuario (DC) (TT)

US que los profes agregaron pero no puse:

- Iniciar viaje
- Finalizar viaje

<aside> 💡 No las incluí en el MVP porque el core del negocio son los envíos, mientras puedas controlar el estado del envío (en preparación-en camino-entregado) tenes un minimo de certeza del estado de tu paquete/envío y no haria falta info del viaje. Se pueden omitir si el envío es la entidad principal del negocio.

</aside>

## MVP

**Versión Dador de Carga (En el MVP solo irían las marcadas con DC)**

**Aplicación Web** que permita a los usuarios contratar servicios de envío de paquetes mediante particulares, realizando pagos mediante Mercado Pago. Para ello el usuario debe poder registrarse y asociarla a una cuenta su rol de dador de carga, también debe poder consultar el estado del envío, cotizaciones y contrataciones de transportistas.

No se contempla el mantenimiento, seguimiento por gps, calificación de transportista, zonas de cobertura, agregado de detalles o fotos al pedido, ni notificaciones push. (Ni todas las US de TT)

Justificación: se consideró el core funcional del producto para que pueda comenzar a utilizarse en un ciclo de entrega de carga y control de envío, con transportistas acotados a un grupo selecto de usuarios de prueba que dan soporte al MVP.

### **Versión Transportista (En el MVP solo irían las marcadas con TT)**

**Aplicación Web** que permita a los usuarios proveer servicios de envío de paquetes mediante particulares, realizando cobros mediante Mercado Pago. Para ello el usuario debe poder registrarse y asociarla a una cuenta su rol de transportista, también debe poder registrar y consultar el estado del envío, cotizaciones y contrataciones activas que posee.

No se contempla el mantenimiento, seguimiento por gps, calificación de transportista, zonas de cobertura, agregado de detalles o fotos al pedido, ni notificaciones push. (Ni todas las users de DC).

Justificación: se consideró el core funcional del producto para que pueda comenzar a utilizarse en un ciclo de envío de paquetes y control de envío, con dadores de carga acotados a un grupo selecto de usuarios de prueba que dan soporte al MVP.

### **Versión MVP para validar ambos**

**Aplicación Web** que permita a los usuarios contratar y proveer servicios de envío de paquetes mediante particulares, realizando pagos mediante Mercado Pago. Para ello el usuario debe poder registrarse y asociarla a una cuenta su rol, también debe poder registrar y consultar el estado del envío, cotizaciones y contrataciones de transportistas.

No se contempla el mantenimiento, seguimiento por gps, calificación de transportista, zonas de cobertura, agregado de detalles o fotos al pedido, ni notificaciones push.

Justificación: se consideró el core funcional del producto para que pueda comenzar a utilizarse en un ciclo de contratación de transportistas y envío de paquetes completo.

#### **Frase Verbal**

#### **Registrar Usuario**

#### **Descripción**

As a [user role], I want to [action], so that [value/business objective].

## Story Points

Acceptance Criteria: [List of specific criteria that must be met for the user story to be considered complete.]

Test Case: [Description of the specific test case.]

## Justificación

Tips: Todas las operaciones de ABMC o registro de usuario se pueden tomar como canónicas. Con incertidumbre nula (tanto técnica como de negocio), complejidad baja y esfuerzo bajo.

<aside> 💡 Mediante la estimación, se le asignó una puntuación de 1 a esta user story, posee una complejidad baja (que hace compleja a la tarea, interactuar con APIs, integrar servicios como MercadoPago, integración con otros módulos del mismo sistema, datos a recopilar y validar), incertidumbre técnica y de negocio nulas y un esfuerzo bajo debido a (tareas que vas a desarrollar, considerar el tiempo estimado para el desarrollo, las pruebas, la documentación y cualquier otra actividad relacionada)

</aside>

# 19/05 - Testing Caja Negra - TP 8

Testing: Disciplina del sw que busca las fallas del sistema. Falla vs Error, diferencia.

Actividades de Niveles de prueba. Test unitario, integración

Caja blanca vs negra.

Caja negra: Mirar entradas y salidas, dada X entrada, tenemos Y salida.

Caja blanca o de cristal: Permite dimensionar el código y las funciones. Lo importante es el algoritmo.

Las cajas, actividad costosa, la cobertura de testing al 100% es casi imposible de alcanzar, la idea es probar la mayor cantidad de cosas, al menor costo posible.

## Metodos de Caja Negra

Buscar metodos para hacer pruebas, elaborar casos de prueba. El QA elabora casos de prueba, casos a probar con un contexto, tiene los pasos a ejecutar y los resultados esperados, a modo de comparar la salida con la entrada. Como elaboramos la menor cantidad de casos de prueba

Basadas en las especificaciones:

- Particion de equivalencias: En un sistema tenemos entradas, salidas y contexto, contexto es cada campo que uno ingresa (user, contraseña), todos los inputs son entradas, todos los mensajes de error o acciones son salidas. La tecnica intenta agrupar clases en válido, buscar conjunto minimo de clases que produzca la misma salida. Es basado en la especificacion dado que no se tiene el sistema/desarrollo a mano, se arman antes de que exista el sistema.
- Valores límite: Instanciacion de particion de equivalencias. Sentido común. Evaluar los valores limite, edades, precios.

Basada en experiencia: Tests exploratorio.

A controlar: valores invalidos o inexistentes en campos requeridos, formato incorrecto.

Se definen las clases de equivalencia válidas/inválidas de entrada, en base a ellas se genera un caso de prueba.

Identificar 3 a 5 mensajes de error: No ingresa calle, no existe el pais/ciudad, etc.

## Armado de casos de Prueba

Cada caso tiene un uuid, un nombre que describe el escenario de lo que quiere probar.

Prioridad: Alta - Media - Baja. Mayor cobertura y valor de negocio.

Bajo para cuestiones cosmeticas, de estética.

Media, situaciones a presentar, me generan un perjuicio, alguna regla de negocio que no se respeta, (tiempo de espera mayor al esperable)

Precondiciones: Datos que tienen que estar en la DB para que se cumpla.

Mail: salida

Cliente: →Clase de equivalencia → Cliente logueado con email registrado.

| Condicion Externa | Clases de equivalencia válidas | Clases de equivalencias inválidas        |
|-------------------|--------------------------------|--|
|                   | Barrio valido seleccionado     | Barrio inexistente                       |
|                   | Barrio no seleccionado         |  |
|                   | Taxista seleccionado           | Taxista seleccionado sin chapa conectada |
|                   | Taxista no seleccionado        | Taxista inexistente                      |



|                     |   |   |
|---------------------|---|---|
|                     | Taxista seleccionado<br>mediante mapa           |   |
| Estado de taxi      | Estado de taxista existente                     | Estado distinto a los válidos                                       |
|                     | Estado no seleccionado                          |   |
| Usuario<br>logueado | Usuario logueado con perfil<br>de administrador | Usuario no logueado   |
|                     |   | Usuario logueado con cualquier otro rol<br>que no sea administrador |

Se considera como valida las clases de equivalencia validas a todas aquellas que representan un evento de interacción entre el usuario y el sistema.