22/08/2023 PRACTICO

Repaso User Stories

Estructura US  
Como: rol, no persona física, rol que está ejecutando.  
Quiero: funcionalidad que se espera que haga.  
Para: tiene que entregar valor de negocio.

3C -> Card – **Conversation** – Confirmation

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Frase verbal en **infinitivo**.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Criterios de aceptación: poner lo que se espera, lo que se puede hacer, restricciones, etc.

* Usar DEBE y PUEDE dependiendo de si el dato es opcional u obligatorio.
* Se busca evitar ambigüedades.
* OJO con los casos de datos “válidos”, aclarar en que contexto.

Imagen que contiene Diagrama

Descripción generada automáticamente

Pruebas de usuario: el PO te da casos de prueba, distintas combinaciones de escenarios, que deberían **pasar** o **fallar**.

* Caso de mínima, caso de máxima, errores posibles.
* Las fallas deben ser contraladas, para que el usuario tenga un plan de acción.
* Lenguaje para automatizar las pruebas.
* **Probar** + frase verbal + … (pasa/falla)
* De mínima una por criterio

TP1 Mis gastos familiares

Preguntas del enunciado:

Hay un responsable de gastos tipo único, no se puede acceder de distintas cuentas al mismo grupo familiar.

Se puede modificar gastos

No hay gastos a futuro, es actual y pasado.

Nombres y apellidos sin relaciones (no me acuerdo q es esto)

Los tipos de gastos son personalizables.

1. Roles: Administrador de gasto, usuario deslogueado, usuario deslogueado. **NUNCA** poner usuario solo.
2. Stories identificadas (frases verbales):
   * 1. Visualizar gastos (**PLURAL** IMPORTANTE)
     2. Cargar gasto (**No** usar “gestionar”, es muy amplio) (**un** gasto)
     3. Filtrar por tipo de gasto
     4. Registrar tipo de gasto
     5. Cambiar fecha
     6. Cambiar gasto
     7. Registrar responsable de gasto
     8. Registrar usuario (cuando son distintos roles tipo Cabify se deben hacer **historias separadas**)
     9. Iniciar sesión de usuario
     10. Cerrar sesión de usuario
     11. Modificar datos de usuario
     12. Borrar gasto
     13. Eliminar tipo de gasto
     14. Eliminar responsable de gasto.
3. User Story completa (card)

US: **Registrar Usuario**

Como [Administrador de gastos] quiero [registrarme en la app] para ~~poder~~ [comenzar/iniciar a administrar mis gastos.]

Criterios de aceptación:

* Debe registrarse con un mail válido (acá se puede usar *válido* porque hay un formato tipo estandarizado para los mail)
* Debe ingresar nombre, apellido y nombre de usuario no ocupado.
* Debe ingresar contraseña de 8 caracteres, alfanumérica (preguntar al PO como tiene que ser la contraseña)
* Puede agregar foto de perfil, formato, tamaño no mayor a 5MB (recordar puede = opcional)

Pruebas de usuario:

* Probar registrar usuario con un mail válido completando campos obligatorios [PASA].
* Probar registrar usuario con mail inválido [FALLA].
* Probar registrar usuario con contraseña invalida [FALLA].
* Probar registrar usuario con mail válido con obligatorios y opcional (foto) [PASA].
* Probar registrar usuario con solo nombre [FALLA].

US: **Registrar tipo de gasto**

Como [Administrador de gastos] QUIERO [registrar un tipo de gasto] PARA [clasificar/organizar mis registros de gastos]

Criterios de aceptación:

* Debe ingresarse un nombre no utilizado, máximo 20 caracteres (preguntar a PO).

Pruebas de usuario:

* Probar registrar un tipo de gasto con nombre válido [PASA].
* Probar registrar un tipo de gasto con nombre repetido [FALLA].
* Probar registrar un tipo de gasto con nombre mayor a 20 caracteres (esta es opcional pq esto está incluido arriba en *nombre válido.* [FALLA]
* Probar registrar … sin nombre [FALLA].

US: **Registrar gasto**

Como [Administrador de gastsos] quiero [registrar un gasto] para [administrar mis finanzas o las de mi grupo familiar].

Criterios de aceptación:

* Debe poner un nombre al gasto (descripción) de hasta 100 caracteres.
* Debe seleccionar un tipo de gasto.
* Debe seleccionar una fecha no mayor a la actual. Por defecto se pone la actual (preguntar al PO los por defecto)
* Debe ingresar un monto numérico positivo > 0, con formato de coma para decimales (preguntar PO formatos)
* Puede modificarse la fecha (opcional)
* Debe poner quien hizo el gasto, si es propio o alguien del grupo.
* Debe autocompletar el nombre y apellido al escribir parte de él si ya se ha usado.

Pruebas de usuario (min 6):

* Probar registrar un gasto y poner una descripción de hasta 100 caracteres [PASA].
* Probar reg poner una desc de más de 100 caracteres [FALLA].
* Probar reg no seleccionar tipo de gasto [FALLA].
* Probar reg seleccionar fecha mayor a la actual [FALLA].
* Probar reg seleccionar fecha anterior a la actual [PASA].
* Probar reg ingresar monto negativo [FALLA].
* Probar reg modificar fecha [PASA].
* Probar reg poner alguien del grupo familiar como gastador [PASA].
* Probar reg poner al mismo administrador como que hizo el gasto [PASA].
* Probar reg autocompletar nombre y apellido y lo hace [PASA].

US: **Visualizar planilla de gastos**

Como [Administrador de gastos]quiero [visualizar una planilla de mis gastos] para [analizar y exportar los gastos familiares según filtros de consulta].

Criterios de aceptación:

* Debe visualizarse similar a un formato Excel, con columnas para monto, tipo de gasto y fecha.
* Puede poner un filtro de fechas para elegir un período, por defecto se muestra la del mes actual con sus gastos ordenados desde el gasto más actual.
* Puede filtrar por un tipo de gasto.
* Puede filtrar por rango de montos.
* Puede filtrar por autor del gasto.
* Puede modificar el criterio de ordenamiento [] (poner cuales son los criterios)
* Debe mostrar total de gastos según filtro? (consultar?¿)

Pruebas de usuario ():

* Probar filtrar fecha desde mayor a fecha hasta [FALLA].
* Probar filtrar por un tipo de gasto previamente creado [PASA].
* Probar filtrar por un rango de montos negativo [FALLA].
* Probar filtrar por un autor (lista desplegable) [PASA].
* Probar visualizar con fecha por defecto y se ordenan los gastos desde el más actual [PASA].
* Probar modificar el criterio de ordenamiento a descendente por monto [PASA].

US: **Consultar gasto**

ISW 29/08/2023 PRACTICO

Tp6 Implementar User Stories

Documento de estilo de código PEP3

Cualquier tecnología

Martes 12/09 entrega

Si no compila = 2

Si no traemos el documento de código < 8

TP2

Story Points: Esfuerzo – Complejidad – Incertidumbre

[Nula/Baja/Media/Alta]

Frases:

podés pedir desde tu casa cualquier cosa que entre en una mochila, nosotros lo buscamos y te lo llevamos a tu domicilio = Pedido

podés descargarla desde ambas tiendas (Preguntar)

podés usar Facebook o Google para registrarte o bien mediante e-mail = registros

elegir de entre nuestros comercios adheridos o realizar un pedido a un lugar no listado = visualizar comercios adheridos

navegar por su carta de productos en base a su categoría = consultar carta

un carrito de productos, en donde podés ir agregando diferentes productos especificando la cantidad y si es necesario alguna observación = Agregar producto al carrito

carrito: modificar algún Producto del Carrito o bien quitarlo definitivamente = modificar y eliminar prod del carrito

confirmar el Pedido

el comercio como el cadete reciben notificaciones en sus teléfonos = recibir notificacion

ver el estado de tus pedidos en tiempo real y seguir a tu cadete = ver estado pedido, ver ubi cadete

realizar un pedido de “lo que sea” = pedido lo que sea

**Para desglosar una US revisar el PARA, y el valor que entrega la negocio, si es distinto deberían ser US distintas.**

* Realizar pedido a comercio adherido
* Realizar pedido “lo que sea”
* Visualizar comercios adheridos
* Agregar producto al carrito (2/3)
* Modificar carrito
* Eliminar producto del carrito (1)
* Recibir notificación de pedido realizado (comercio)
* Recibir notificación de pedido realizado (cadete)
* Visualizar estado pedido
* Visualizar la ubicación del cadete
* Registrar usuario con email
* Registrar usuario con Google
* Registrar usuario con facebook
* Consultar carta
* Recibir notificación de preparación de pedido (cliente)
* Recibir notificación de traslado de pedido (cliente)
* Notificar entrega de pedido (cadete)

US: **Recibir notificación de pedido realizado (comercio)**

Como [comercio adherido] quiero [recibir una notificación de pedido realizado] para [iniciar la preparación del pedido].

Criterios de aceptación:

* Debe recibir la notificación push al momento de confirmarse el pedido.
* Debe incluir el número de pedido, el nombre de usuario del cliente, los productos seleccionados con cantidades y observaciones, dirección cliente, nombre del cadete, hora de entrega.
* Debe recibir la notificación aun cuando esté cerrada la app
* La notificación debe ser compatible con iOS y Android.

Pruebas de usuario:

* Probar recibir notificación confirmando un pedido y llega en el momento con todos los datos (numero, nombre cliente, productos seleccionados con cantidades y obs, dirección, nombre del cadete y hora entrega) [PASA].
* Probar recibir una notificación sin permiso y llega [FALLA]. (cuestionable, puede ser PASA)
* Probar recibir notificación confirmando… con la app cerrada y llega [PASA]

Puesta en común

Estimación: **Registrar Usuario con mail (1)**

* Esfuerzo = Baja (pocas horas de trabajo, no hay que hacer muchas validaciones, simple registro a BD)
* Complejidad = Baja (no necesita algoritmos complejos, ni conexiones a sistemas externos)
* Incertidumbre = Nula (se tiene conocimiento de lo que quiere el negocio)

Se compara con la canónica de “Eliminar …” = 1, y también comparte el esfuerzo bajo, complejidad baja e incertidumbre nula. **Esta US tiene 1 SP** también.

Estimación: **Visualizar comercios adheridos**

* Esfuerzo = Medio (validaciones de filtros, varias consultas a BD)
* Complejidad = Baja/Media (se usan algoritmos y búsquedas ¿simples? y no consultas dinámicas a BD)
* Incertidumbre = Nula (de negocio nula, pero técnica baja por los algoritmos a utilizar o las consultas dinámicas)

ISW 5/09/2023 PRACTICO

TP3

* Comprar prenda
* Buscar prenda (filtros cats)
* Enviar prenda para la venta (vendedores envian)
* Publicar prenda (ellos la publican)
* Registrarse (incluye verificación?)
* Seleccionar prenda
* Leer QR? O va dentro de publicar prenda?
* Notificar al vendedor por mail prenda en venta
* Confirmar venta de prendas (y q hacer con las otras)(vendedor en web)
* Notificar al vendedor donación o retiro
* Registrarse cliente distinto?
* Notificar venta al vendedor
* Registrar pago al vendedor?
* Notificar pago al vendedor?

Hipótesis

MVP -> debe dar valor a los destinatarios

Al detallar un MVP debemos incluir:

* Hipótesis
* User stories incluidas en esa hipótesis
* Justificación de esas US: puede ser un párrafo mencionando todas o una por una, no es tan estricto. La idea es no hacer nada “porque si”

Hipótesis: Existen Usuarios dispuestos a **pedir** por la app.

US:

* Realizar pedido de lo que sea (solo se acepta efectivo al comenzar) (de esta forma recortamos la US para el MVP y es más fácil, acordado con el PO, justificando cada recorte).
* Registrar usuario (mail/Facebook/Google) (para la hipótesis usamos mail). [como consumidor y como cadete]
* Notificar entrega de pedido. [Registrados por BD, dándoles sus credenciales a mano]
* Iniciar sesión con mail.
* [El seguimiento lo vamos a dejar para después, este MVP puede hacerse solo de pedir y no darle importancia a la preparación y demás del cadete.]
* Anotar por que hacemos los cambios, recortes y por que no hacemos lo que no hacemos je.

Hipótesis: Existen comercios dispuestos a **~~ofrecer~~ vender** sus productos por medio de una aplicación.

US hipótesis:

* Registrar usuario con email (adherirse) (comercio)
* Iniciar sesión con email
* Agregar/registrar producto a la carta
* Recibir notificación de pedido realizado (comercio)
* Realizar pedido a comercio adherido
* Consultar carta
* Visualizar comercios adheridos
* Agregar prod a carrito
* Eliminar prod a carrito
* Visualizar pedidos
* Visualizar un pedido
* Notificar entrega de pedido (cadete)

Con estas US logramos que un comercio se adhiera (registrándose) y agregue productos para ofrecer en su carta. (agregar más de los nuevos US)

MVP de TP3 recircula tus prendas

* Comprar prenda
* Buscar prenda (filtros cats)
* Enviar prendaS para la venta (vendedores envian) (prendas pq se envia un pack)
* Publicar prenda (ellos la publican)
* Registrarse (incluye verificación?)
* Seleccionar prenda
* Leer QR? O va dentro de publicar prenda?
* Notificar al vendedor por mail prenda en venta
* Confirmar venta de prendas (y q hacer con las otras)(vendedor en web)
* Notificar al vendedor donación o retiro
* Registrarse cliente distinto?
* Notificar venta al vendedor
* Registrar pago al vendedor?
* Notificar pago al vendedor?

Hipótesis: Existen usuarios que quieren **comprar** todo tipo de ropa por medio de una aplicación.

* Registrar usuario con mail (cliente) [incluye verificación]
* Iniciar sesión con mail
* Visualizar catálogo de productos
* Confirmar compra con retiro [al comienzo solo será en efectivo]
* Recibir email de detalle de venta (vendedor)
* Recibir email de pedido listo p retirar (cliente)
* Carrito?

Justificación: El registro con mail será con tipo “cliente” el cual incluye la verificación de teléfono, a su vez se podrá iniciar sesión con esas credenciales, se podrá visualizar el catalogo de productos que será cargado por BD en este MVP, en este MVP solo se podrá pagar con efectivo, se reciben mails para notificar al vendedor de su venta y al cliente cuando su pedido este listo.

* Iniciar sesión con gmail como comprador: Busca poder iniciar sesión en la aplicación como comprador, en el caso de que no exista la cuenta previamente Google se encarga de crear una nueva. Es necesario para poder realizar las compras futuras
* Visualizar catálogo de productos
* Confirmar compra con retiro [al comienzo solo será en efectivo]
* Recibir email de detalle de venta (vendedor)
* Recibir email de pedido listo p retirar (cliente)
* Carrito?

Hipótesis: Existen usuarios dispuestos a **vender** sus prendas

FOTO RESOLUCION

ISW 12/9 PRACTICO

1. Describir user relacionada al req de “Enviar prendas a vender”
2. Definir user canónica y justificar
3. Estimar user 1) justificando

Complejidad AMB

* Cantidad transacciones/entidades
* Validaciones
* Rep graficas/pantalla/UX
* Cálculos complejos

Esfuerzo AMB

* Persistencia
* entidades
* pantallas
* integraciones (API/Librerias)

Incertidumbre

Técnico

* Integraciones API
* Tecnología/lenguaje
* Servidores
* Imágenes
* Libreria

Negocio

* Reqs incompletos/ambiguos
* Reglas de negocio no claras
* Reqs legales

US: **Enviar prendas a vender**

Como [vendedor] quiero [enviar prendas] para [~~obtener una ganancia~~ con la venta de las mismas/poner a la venta/sean seleccionadas para su venta].

Criterios de aceptación: DEBE (obligatorio) PUEDE (opcional)

* ~~Debe comprobar que sea un usuario registrado~~
* Debe indicar la cantidad de cada categoría de prenda.
* Debe indicar una descripción de cada prenda de hasta 200 caracteres (en el dominio del profe es de las prendas de toda la categoría).
* Puede subir una foto de hasta 512kb (¿de cada prenda? En el dominio del profe es de cada categoría).
* Debe seleccionar la forma de envío, (Retiro en domicilio o llevar a punto de recolección) en caso de punto de recolección debe seleccionar un punto de recolección (sucursales, comercios amigos)
* Debe ingresar como mínimo enviar 10 prendas.
* (Agregar “Confirmar envío de formulario” no suma ni resta)

Pruebas de usuario: PROBAR + FRASE VERBAL + CASO

* Probar enviar prendas a vender ingresando mas de 10 prendas, con cantidades de categoría y descripción en cada una, sin incluir foto, seleccionando forma de envío “Retiro en domicilio” [PASA]
* Probar enviar prendas a vender ingresando mas de 10 prendas, con los campos obligatorios, incluyendo una foto de mas del peso máximo [FALLA]
* Probar enviar prendas a vender
* Probar enviar prendas a vender ingresando menos de 10 prendas [FALLA]
* Probar enviar prendas a vender
* Probar enviar prendas a vender

Cada user debe ser independiente

Registrar con mail tiene menos incertidumbre que usar Google por las llamadas a API y ¿…?

2) canónica: registrar con mail

3) estimar user 1)

3/5

Complejidad: Media. Por entidades, validaciones *combinadas*, persistencia, transacción (al límite, casi Alto).

Esfuerzo: Medio. Por implementación de validaciones, persistencia, imagen

Incertidumbre: Media. No presenta incertidumbre de negocio. Si incertidumbre técnica para guardar imágenes + validación. (puede ser baja si tenemos conocimientos con lo de las imágenes, siempre ponerse en la piel de uno, no de un profesional)

Comparado con la canónica de registrar, que tiene 2 de story points y Baja, Bajo, Baja en CEI, se la estima con 5 por tener Media, Medio, Media en CEI.

4 User confirmar propuesta de venta

5 User seleccionar las prendas

6 User completar ficha de prendas

ISW P 19/09/2023

TP4 Armar plan de gestión de configuración

Si la gestión se hace a través de un mail, el mail puede ser un ítem de configuración.

Armar estructura de ítems de configuración con Reglas de nombrado.

No olvidar crear líneas de base

Estructura de repositorio

Carpetas

Listado de Items de Configuración con un glosario debajo de que significan las siglas

El tipo de ítem no va en este TP.

Parcial:

* MVP
  + Descripción-> ¿Que vamos a construir? Hipótesis y justificación
  + Listado de users para MVP (frase verbal)
  + Criterio de por que usamos esas users
  + Lo no incluido: frase de lo que no se incluye (sin justificar)
* Canonica: frase verbal, story point 1/2, por que esa user justificar Incer(nula)/Esfu/Complej. (no usar las de mvp pq son complejas? Preg)
* Descripcion de user completa (la que pide en el req): frase verbal, criterios acep, pruebas usuario, ¿comparando con canonica y estimando?.

-Pruebas de usuario: casos mas relevantes de PO que afecten al negocio (no por ejemplo números negativos, no es tan relevante). No ponerse como QA, sino como PO, evitar generalidades, probar casos que sirvan al negocio particulares que el PO quiera probar.

-Impresión de QR se puede hacer en user aparte

-Estimacion: los campos dinámicos requieren gran esfuerzo porque van apareciendo dependiendo de que categoría seleccionamos

Baja Baja Nula = 1 o 1/2

Baja Baja Baja = 1

Media Baja Baja = 2

Media Media Baja = 3

Alta Baja Baja = 3

Media Media Media = 5

Alta Media baja = 5

Alta media media = 8

Alta Alta Media = 13/8

Alta Alta Alta = 21/13

3/10/2023 ISW P

**TPE 8**

**Taxi Mobile**

**Release planning**

* ¿Cuánto tiempo?
* ¿Cuántos sprints? ¿Duración del sprint? (1 semana no se hace nada, 2-3 ideal, 4 se relajan)
* ¿Qué voy a trabajar en cada sprint?
  + Hacer tabla
* ¿Qué US son parte del reléase?
* Velocidad vs Capacidad

**Sprint planning**

* ¿Quienes trabajan? Contexto del equipo
* Capacidad de cada uno y del equipo
* Tareas
  + Tiempo
    - Ceremonias
  + Según cada US
* Criterio de Done: hacer propio no buscar uno.

**Release planning**

¿Qué US son parte del reléase?

Tomaremos las US que son parte del MVP para el primer reléase.

Taxista

* Loguear taxista: 2
* Ocupar taxi: 2
* Liberar taxi: 2
* Ver ubicación del pasajero: 5

Pasajero

* Buscar taxis cercanos: 3
* Pedir taxi: 5
* Notificar a taxista solicitud de taxi: 3

¿Qué voy a trabajar en cada sprint?

Sprint 1

|  |  |  |
| --- | --- | --- |
| **Sprint N** | **US** | **SP** |
| 1 | * Loguear taxista: 2 * Ocupar taxi: 2 * Liberar taxi: 2 | 2 + 2 + 2 = 6 |
| 2 | * Pedir taxi: 5 | 5 |
| 3 | * Ver ubicación del pasajero: 5 | 5 |
| 4 | * Buscar taxis cercanos: 3 * Notificar a taxista solicitud de taxi: 3 | 3 + 3 = 6 |

¿Cuánto tiempo? ¿Cuántos sprints y cuánto dura cada uno?

4 sprints de 2 semanas cada uno, por lo tanto 8 semanas en total para el release.

Velocidad vs Capacidad

**Capacidad** es la otra métrica fuerte de SCRUM, se estima, es una determinación de antemano del trabajo que creemos que el equipo puede hacer durante un sprint. Se mide de dos formas, o en puntos de historia a *quemar* o medirla en horas ideales (de trabajo efectivo real). Recomendando trabajar por rangos, es decir, entre tantas y tantas horas por día.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Persona** | **Días disponibles (sin tiempo personal)** | **Días para otras actividades Scrum** | **Horas por día** | **Horas de esfuerzo disponibles** |
| Debo | 8 | 2 | 2-3 | 12-18 |
| Ivan | 10 | 2 | 2-4 | 16-32 |
| Julian | 10 | 2 | 3-4 | 24-32 |
| Nico | 8 | 2 | 3-4 | 18-24 |
| Mile | 8 | 2 | 2-3 | 12-18 |
| Valen | 8 | 2 | 1-3 | 6-18 |
| Total |  |  |  | 88-142 |

**Sprint planning**

Sprint Nro. 1

Duración del Sprint en días: 8

Objetivo del Sprint: Al finalizar este sprint el objetivo es implementar las funcionalidades de loguear taxista, ocupar taxi y liberar taxi de forma que el PO pueda tener distintas funcionalidades para aprobar.

Equipo Scrum:

* Débora
* Iván
* Julián
* Nico
* Mile
* Valen

Capacidad del Equipo en Horas Ideales: << XXX>>

¿Quiénes trabajan? Contexto del equipo

Capacidad de cada uno y del equipo

Tareas

Criterio de done

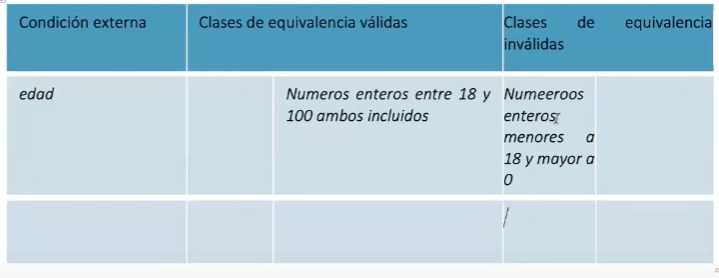
ISW P 10/10/2023

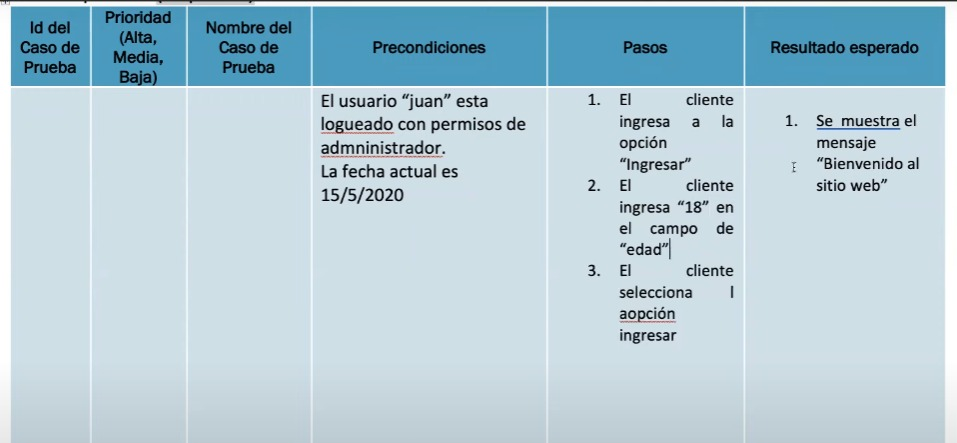
**Testing de caja negra**

<https://docs.google.com/document/d/1E7yVmIM66r8FL8fCJQdeiZlLBr23tKwMPIRKi97ZPSo/edit>

Condiciones externas -> entradas

Lo que produce el sistema es un estímulo de salida -> condiciones externas de salida



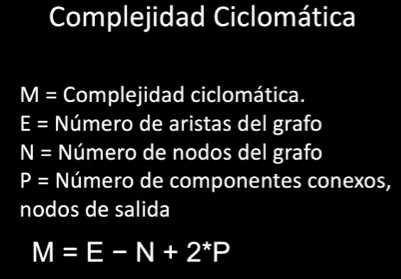


ISW P 17/10/2023

Caja Blanca

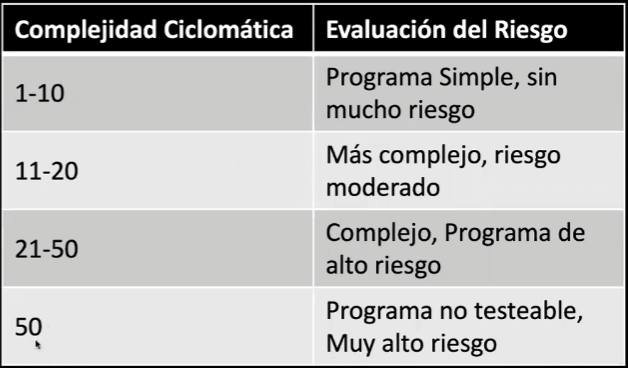
Cobertura: forma o estrategia de poder recorrer los distintos caminos que el código provee para desarrollar una funcionalidad. En un grafo, cada ramificación es un camino.

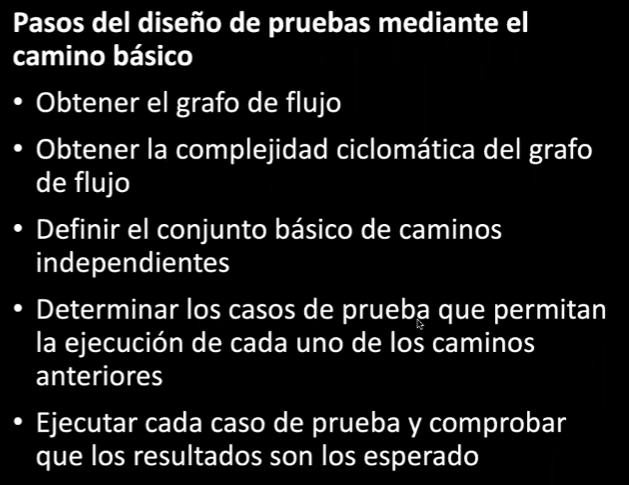
En el practico no vemos cobertura de enunciados o caminos básicos.



O







**Cobertura de sentencias (a partir de acá si se toma)**

Sentencia: cualquier instrucción, asignación o llamada a función. NO las condiciones.

Busca darle cobertura a todas las sentencias/instrucciones.

**Cobertura de decisión**

Decisión: estructura de control COMPLETA. Cada uno de los paréntesis del IF.

Busca darle cobertura a todas las decisiones por verdadero o falso cuando deban hacerlo con menor cantidad de casos de prueba (dándole valores a las variables). No se consideran los IF cortocircuitados donde un operador se cumple y el otro no, con ir una vez a cada verdadero o falso es suficiente.

**Cobertura de condición?** Anote arriba? Solo busca evaluar condiciones~~. O sea si contempla los cortocircuitos, toma todas las condiciones. (esta o la de abajo d/c?)~~

Evaluación lógica dentro de la decisión, conectadas con AND u OR, busca evaluar condiciones individuales por verdadero o falso. No considera cortocircuitos.

**Cobertura de decisión/condición**

Busca valuar todas las decisiones y condiciones tanto por verdadero como por falso.

**Cobertura de múltiple**

Busca valuar el combinatorio de todas las condiciones en todos sus valores de verdad posibles. TODOS. Tipo MAD tablas de verdad, no es variables^2.

Tener en cuenta caminos imposibles por restricciones de variables, por ejemplo si A = null, y otra condición es A = 5, que A = 5 sea verdadero es IMPOSIBLE, se elimina. Que A tenga 2 valores distintos y ambos sean verdaderos es IMPOSIBLE, se elimina.

Todos buscan tener la menor cantidad de casos de prueba para cubrir lo buscado.

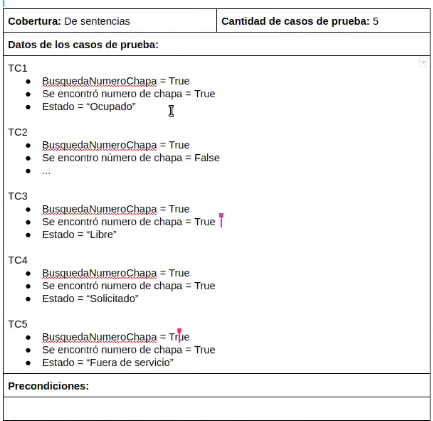
Micka recomienda hacer diagrama de flujo para ver decisiones y condiciones e ir marcando en el diagrama según los casos de prueba que hacemos lo que ya controlamos, para verlo más fácil que el pseudocodigo, nos puede evitar comer una decisión o duplicarla.

Decisión: al usar una variable para comprobar una rama, por ejemplo de falso, no hace falta asignar valores a las demás variables, no interesa.

Decisión/condición siempre va a tener mas casos de prueba que las coberturas individuales de decisión o condición.

Pegar foto de Micka y explicarla.

El cortocircuitado puede aparecer en caso de haber notas de la user story que deban tenerse en cuenta.



TP12 hacerle el entorno al grupo que nos testea

Por lo menos 10 casos de prueba

Tp11 armar casos de prueba, con las clases de equivalencia. Diferencias entre casos de prueba y lo realmente probado son los defectos a informar en la planilla.

Explicación Micka de caja blanca

Caso Sistema GPS

Pseudocodigo asociado a una user story, donde en esa user puede estar el formato o consideraciones especiales.

Parcial: Sentencia, Decisión, Condición, Dec/Cond, Múltiple.

TP10

Forma

Descripción generada automáticamente

**Sentencias**

Cobertura: De Sentencias Cantidad de Casos de Prueba: 1

Datos de los Casos de Prueba:

C1 (sentencia crearConvocatoria()):

* cliente = “Disney”
* descripción = “corto de Star Wars”
* cantidadPersonajes = 2

Precondiciones:

**Decisión**

Cobertura: De decisión Cantidad de Casos de Prueba:

Datos de los Casos de Prueba:

Precondiciones:

**Explicación precondiciones**: pueden ser textuales, por ítems como la foto del parcial. Arrays pueden hacerse por texto o [“así”,”poniendo”,”todas”].

**Condición** Evaluación lógica dentro de la decisión, conectadas con AND u OR, busca evaluar condiciones (cada variable = algo) individuales por verdadero o falso. No considera cortocircuitos.

Cobertura: Condición Cantidad de Casos de Prueba: 6

Datos de los Casos de Prueba:

1. cliente = null
2. cliente = “Disney”, descripción = “corto de Star Wars”, cantidadPersonajes = 2, nombres = [“Kylo Ren”, ”Rey”]
3. cliente = “Disney”, descripción = null
4. cliente = “Disney”, descripción = “corto de Star Wars”, cantidadPersonajes = null

Precondiciones:

**Cobertura de decisión/condición:** Busca valuar todas las decisiones y condiciones tanto por verdadero como por falso.

No es raro que queden iguales el decisión/condición con respecto al condición

**Cobertura múltiple**

Parcial: más de una cobertura seguro, más probable que sean 2.

Ultima multiple pag 106 guia

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t>=30 | t<=50 | Int(T) | Form valid | FHInicio <= FHAct+5 | D>=40 | D<=120 | Int(D) | Text !=null | Aspec  !=null | Maq.rep | Maq.carg |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

