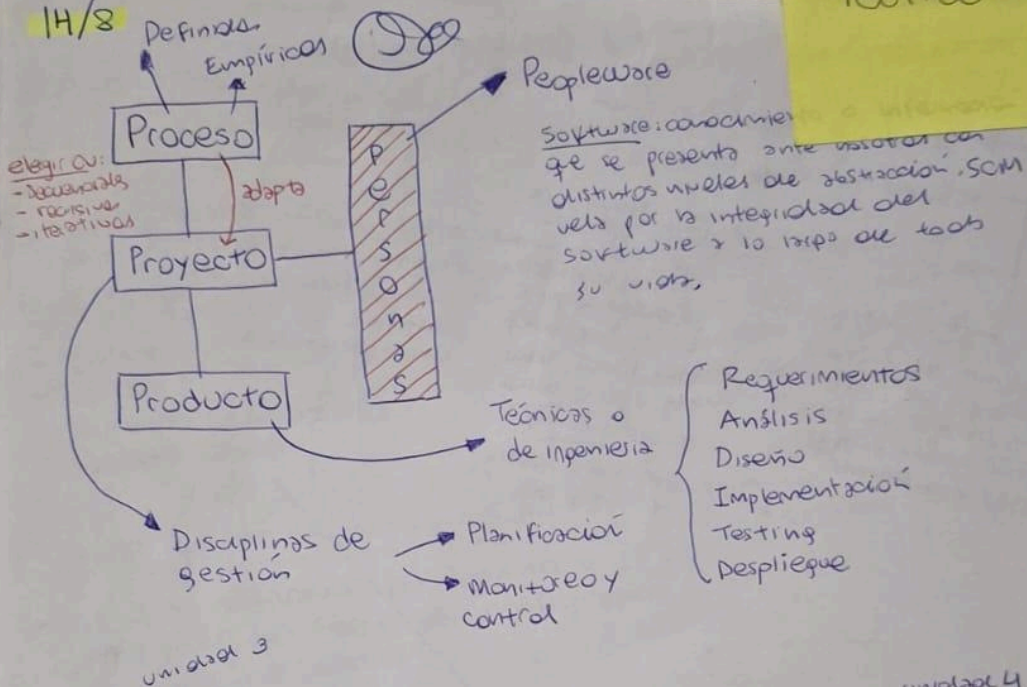


NOTAS TEÓRICO

14/8



Teórico

Disciplinas Protectoras:

- SCM software configuration management
- Aseguramiento de calidad de procesos y de productos
- Métricas → unidad 1 y 2

unidad 3

unidad 4

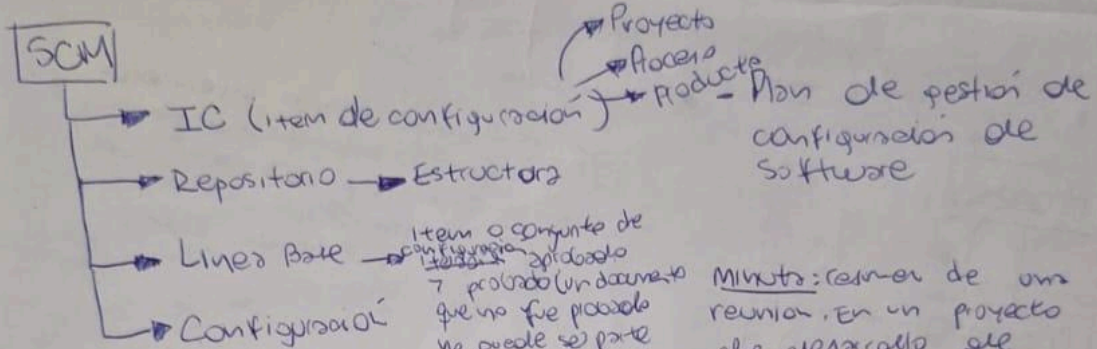
PPQA

Artículo de No Silver Bullet tiene contenido que se pregunta en el primer parcial

ingeniería de software: disciplina protectora y transversal que garantiza y vela por la calidad del producto

- Los procesos definidos permiten lograr la repetibilidad (PUD/RUP)
 - Los procesos empíricos buscan tener un equilibrio (Agil) → no es una metodología!!
- (Lean) Agil → Scrum
Lean → Kanban

Pr2



Como una foto que contiene todos los items de configuración en un instante de tiempo. No es lo mismo la línea base de configuración.

Item o conjunto de configuración aprobado y probado (un documento que no fue pasado no puede ser parte de la línea base). Hay líneas base de especificación y operacionales. Una tiene código y otra no.

Minsky: recordar de una reunión. En un proyecto de desarrollo de software se hace en varios momentos.

Práctico

21/08

SCM: Software Configuration Management

TP4: 21/08/25 - 28/08/25

TP5: 21/08/25 - Fin del cuatrimestre

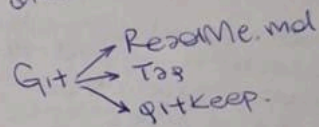
} Entrega por UV
(uno solo por grupo)

→ todo el equipo debe de tener commits (no todo al último).

Ejemplos de herramientas de scm:

- Github
- Gitlab
- Bitbucket (Atlassian)
- Google drive.

Linea base: es una versión del software aprobada y que funciona y que para cambiarla debe ser puesta a discusión.



Plan de gestión de configuración:

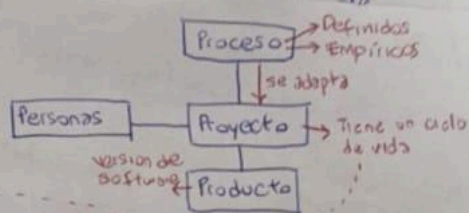
- Define la estructura del repositorio.
- Identificar los items de configuración
 - nombre - nomenclatura
 - que es
 - ubicación del archivo
- Glosario
- Criterio de linea base
- Crear propio repositorio (público)

Grupo 2

26/8

→ Proyecto

Filosofía ágil Manifiesto ágil



Tipos de ciclos de vida

- secuencial → cascada clásica
- iterativo → iterativo e incremental
- recursivo → espiral

Los procesos definidos pueden usar cualquier pero el empírico solo puede usar el iterativo (esto es por la retroalimentación)

Proyecto: medio por el cual administramos los recursos y las personas asignados a él para producir un producto. El producto se caracteriza por su resultado, el cual es único e irrepetible (cada versión es única y diferente por más de que el producto sea el mismo).

Características de un proyecto:

- Resultado Único
- Fecha Inicio/Fin
- Elaboración gradual
- Tareas interrelacionadas

En un punto se puede pedir hacer una infografía.

En el tema de retroalimentación ágil, el ciclo es una técnica.

Procesos empíricos vs. Procesos definidos:

Los procesos empíricos se diferencian de los definidos en que estos permiten predecir el comportamiento ya que definen de forma completa y previa por ejemplo los roles de las personas en el proceso. ~~Por ejemplo~~

Los procesos empíricos dicen que la experiencia no se extrapola y le sirve al equipo solamente (la experiencia cambia). El equipo genera su propia experiencia y lo debe hacer lo antes posible. ~~importante~~

El equipo no necesita pedir permiso a alguien fuera del equipo para definir ~~o~~ como trabajar. En un proceso definido el equipo no tiene que ser "preparado" o "muy preparado" ya que esto ~~se~~ se contrasta con lo definido pero en uno empírico es más importante tener un equipo preparado y que pueda tomar decisiones. Pilares del empirismo:

Transparencia: visibilidad que se puede hacer la inspección y adaptación. La información y el producto de trabajo es de todos. También implica saber pedir ayuda.
Inspección: ciclos cortos, control y adaptación en función de lo que considere.

Adaptación: correcciones/cambios realizados luego de un control

La filosofía Ágil y Lean trabajan con procesos empíricos.

Scrum (Framework) → Ágil
Lean → Kanban
Empíricos

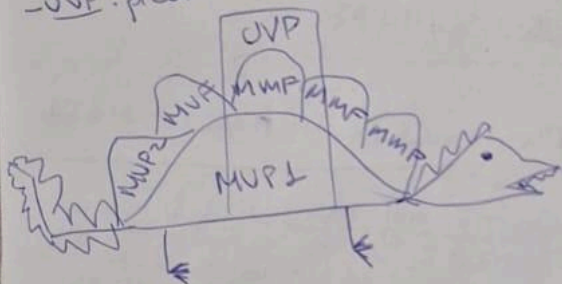
Ágil no es una metodología. Es un pensamiento. Por eso es una filosofía.

Práctico

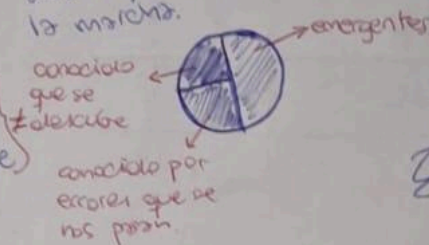
2/2

MVP

- MVP: producto mínimo viable.
- MVF: feature mínimo viable.
- MMP: producto mínimo comercializable.
- MMF: característica mínimo comercializable.
- UVP: producto único viable.



El 50% de los requerimientos se tienen en cuenta desde antes y el resto emergen sobre la marcha.

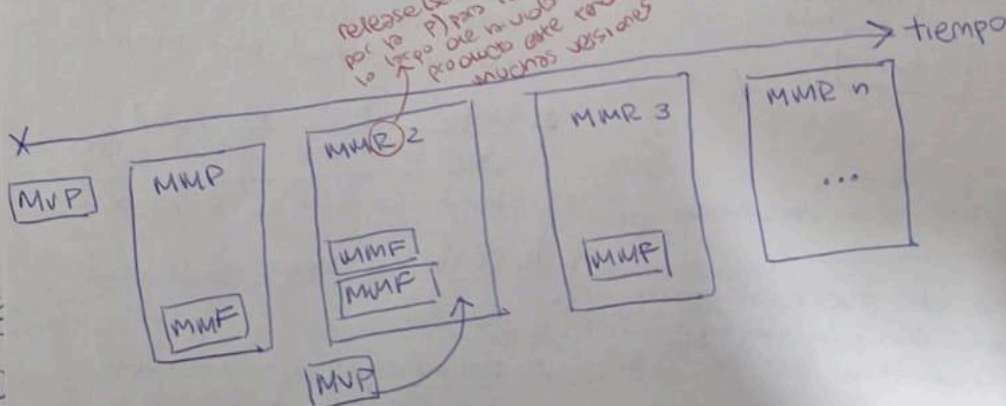


En el manifiesto agil lo importante es que se puede comenzar a trabajar con el sistema sin tener todos los requerimientos.

Viable: tengo una idea pero necesito validarla en el mercado (con una hipótesis)

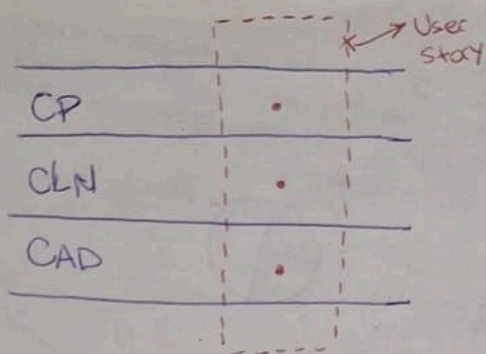
Mínimo: limitado en funcionalidad

release se suele cambiar por lo p/p para reflejar que lo largo de la vida del producto este rancho muchas versiones



Valor y Desperdicio: todo lo que no genera valor es un desperdicio. Se debe eliminar en lo posible la mayor cantidad de desperdicios. A veces se pueden eliminar y otras veces no.

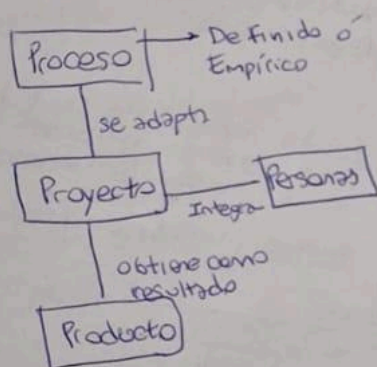
El MVP puede ser un video, ppt, etc pero es MMP ya porque este debe de ser comercializable



cuando desarrollo debo entregar al usuario una característica que pueda ponerse en producción por eso la user story se corta verticalmente.

8/8

Componentes de un Proyecto de Software



Planificación de un proyecto de SW

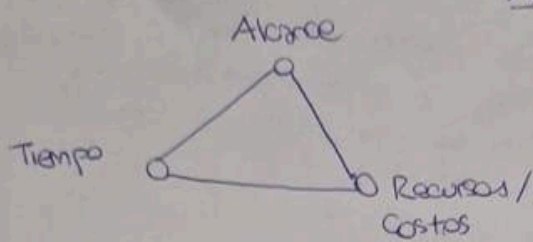
1. Objetivo del proyecto: ¿qué vamos a hacer?
 { Claro } un objetivo debe ser escrito
 { Abordable } cumpliendo esto
2. Alcance del proyecto: todo el trabajo y solo el trabajo que es necesario realizar para cumplir con el objetivo del proyecto.
3. Definir el proceso y el ciclo de vida del proyecto.
 ¿cómo lo vamos a hacer?
4. Definir el equipo de proyecto.
 ¿quién?
5. Estimación: Es una predicción/ supuesto que tiene una probabilidad (siempre) asociada.

Cuando estimamos debemos de estimar:

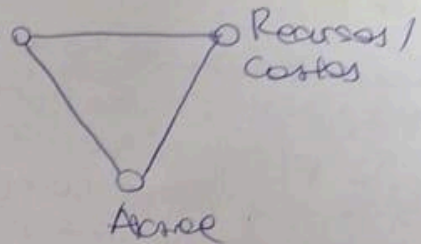
- Tamaño del producto (se suele medir usando la rcu con el "complejidad")
- Esfuerzo del proyecto (hrs personas)
- Tiempo/Calendario del proyecto
- Costos
- Recursos críticos

Proyecto

Triangulo de hierro o Triple restricción:

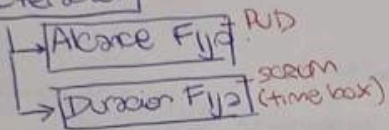


Apilismo:
Tiempo



Definición: periodo de tiempo que se debe en proyecto

Iteración



16/9

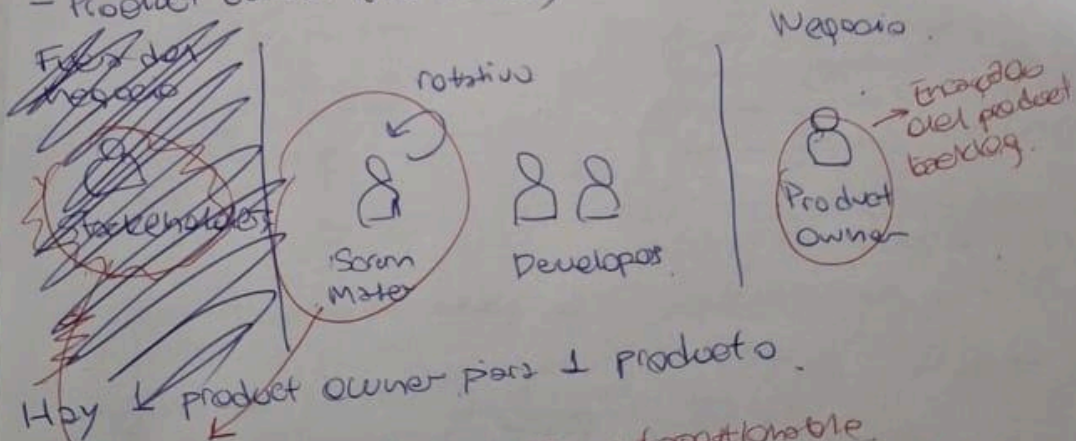
Scrum: framework que nos ayuda a organizar. Sirve para gestión ágil. El manifiesto ágil se conforma por 4 valores y 12 principios. Scrum adhiere a este manifiesto y propone cosas para cumplirlo, esto se materializa en prácticas. Scrum no sirve para hacer software.

Scrum respeta los 3 pilares del empirismo.

- Transparencia
- Inspección
- Adaptación

Involucrados (categoría de responsabilidad)

- Scrum master (técnico) - puede ser una responsabilidad rotativa
- creadores de productos de trabajo (Developers) (técnico)
- Product owner (no técnico)



Un equipo de scrum es autogestionable. La responsabilidad del scrum master es ser un líder de equipo. Garantizar que se cumplan las reglas del scrum.

Eventos (Spring es el contenedor de todas las cuentas)

- Planning (una iteración de duración fija. Como resultado, genera un incremento en el producto. La duración máxima es 1 mes)
- Daily
- Review
- Retrospective

- Refinamiento del product backlog.

Tech Stories

Douba tecnico

Defectos

US → Entrar a Sprint

Epico

Temas

Spikes

En el product backlog hay producto ~~no tarea~~ hay producto y no tareas. En el Sprint backlog no hay tareas.

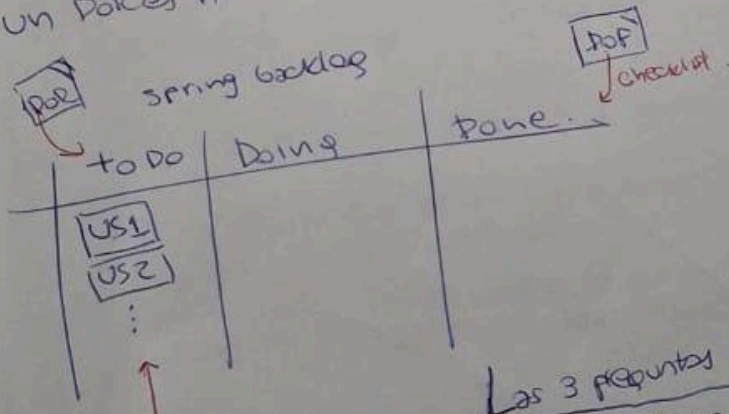
Sprint planning: define el momento del producto y el objetivo para lograrlo. Una planning tiene 2 parts

Como: ~~como se va a hacer~~ ~~este el product owner~~ ~~se va a hacer~~
Como: po planning, puede que el product owner no este si no lo necesitamos

Que: debe estar el product owner.

- Las US se estiman en SP
- Las tareas se estiman en Ho ideas

Un po es normalmente un checklist.



lo definido en el sprint backlog.

Las 3 preguntas de la daily

- Qué se hace
- Que se hace
- Que lo hace

La métrica de según es la velocidad (se calcula sumando los puntos de los historias que en el sprint se caben)

Se hace frente al tablero y se lo actualiza.

NOTAS PRÁCTICO

Práctico

23/8

User stories

→ Las 3 o de las user stories

Conversación (más importante)

Card - Tarjeta
Con firma

→ Frase verbal
→ Criterios de aceptación
→ Pruebas de usuario
→ Descripción

Rol, Quién, Qué,
Por qué, valor de negocio.

Ejemplo de una User story:

Como Rol quiero Que para valor de negocio

Casos de prueba y Pruebas de usuario.

Los casos de prueba son para buscar errores en el software y es una etapa anterior a las pruebas de usuario. Las pruebas de usuario le sirven para saber si el software es lo que los usuarios están buscando o no.

Pruebas de usuario:

- se hacen tantas como sean necesarias
- un error es muy grave y baja la confianza

Ejemplos de casos de prueba:

- 1) El usuario inicial sería
- 2) Model selección y operá cobrar al cliente

Criterio de Ready:


Modelo: se definen con el cliente


Independiente: la us que trabajo ahora es independiente de otros us

Requerible: los criterios de aceptación son negociables

Estimable: todos los us tienen que dar algún valor al negocio, cuánto tiempo/esfuerzo, etc me llevarán

Testable: todos los us debe poder testearse para evaluar si tiene errores o error bien planteado



Use stories TPI Gastos familiares

US candidatas:

- Guardar
- Registrar gasto ✓
- Registrar responsable de gastos ✓
- Visualizar plantilla de gastos ✓
- Filtrar gastos ✓
- Iniciar sesión ✓
- Registrar usuario ✓
- Modificar gasto ✓
- Registrar un tipo de gasto ✓
- Eliminar gasto ✓
- Cerrar sesión ✓

Estas se dejan con este lenguaje y se dejan de usar así si el dominio se registra:
 - Registrar con mail - Iniciar con mail
 - Registrar con google - Registrar con google

Consejos:

- las US son mas del lado del usuario (van en el lenguaje del cliente)
- una US tiene que dar valor por si misma

errores comunes

Detallar la user story de Filtrar gastos:

User story Filtrar gastos y registrar usuario
 Yo como usuario quiero poder filtrar los gastos para poder ~~hacer~~ gestionarlos de una manera mas eficiente mostrándolos segun ciertos criterios

muy poco común usar usuario como rol/es muy generados si se podría usar para finalizar sesión

USX Frase verbal

Yo como Rol quiero...
 Para valor de negocio

CA:

- se debe obligatorio
- se puede opcional

CA:

- se debe mostrar todos los gastos del mes ordenados del mas actual al mas antiguo.
- se debe permitir ver cualquier periodo de fechas para los gastos
- se debe poder filtrar por tipo de gasto
- se debe poder filtrar por ~~el~~ responsable de gastos
- se debe poder filtrar por ~~el~~ rango de monto
- se debe poder modificar el criterio de ordenamiento
- se debe mostrar el total de gastos segun el filtro aplicado

FU:

Probar frase verbal ... (Pass/Fails)

- Review
 - Retrospective

4/9

User stories, MVP, Estimaciones TP2

MVP \Rightarrow Todo el listado de las US
 \rightarrow Entran en el MVP
 \rightarrow No se incluyen en el MVP
Descripción
Hipótesis (Ho): ¿Que mercado voy a atacar?
¿Que necesidad quiero cubrir?

¿Que estamos?

Tradicional

- \rightarrow Tamaño
- \rightarrow Esfuerzo: se mide en Horas personas lineales
- \rightarrow Calendario
- \rightarrow Costo

En base al de arriba voy definiendo los de abajo

Agil:

\rightarrow Tamaño: características del producto } US [Story Points]

peso que tiene una US

Elegir: Escala de la estimación

- 1..10
- 2^{na}... 12481632...
- Fibonacci: 011235813...

Como elegir valor?

SP \Rightarrow Peso

- \rightarrow Complejidad (objetivo)
- \rightarrow Esfuerzo (subjetivo, depende del equipo)
- \rightarrow Incertidumbre: que tanto se atreva de implementar la US

Estimación:

La estimación es un proceso de opinión sobre uno de lo que tiene intentos predecir que es lo que tenemos que desarrollar. Nos ayuda a planificar y ver si lo que queremos construir lo vamos a poder construir

TP2 EcoHarmony Park

us candidato:

- Registrar sesión
- Iniciar sesión
- Cerrar sesión
- Consultar mapa
- Comprar entradas → efectivo o tarjeta
- Consultar horarios de alimentación
- ~~Consultar horarios de alimentación~~
- Inscribirse a actividad

Para que una us ses
us debe de
cumplir con el
modelo INVEST

Una us con una
puede ser una de
Abile

Comprar entradas:

Yo como visitante del parque quiero previamente poder
comprar mis entradas ~~en el aplicativo~~ en el
aplicativo para así ingresar más rápidamente. ~~en el aplicativo~~
~~en el aplicativo~~ autogestionandome. ~~en el aplicativo~~

CA:

- El visitante debe estar previamente registrado para comprar la entrada
- El visitante debe ingresar la fecha de visita deseada
- " " la cantidad de entradas y debe ser menor a 10
- " " la forma de pago en efectivo o tarjeta
- " " verificar su entrada en el parque

PU:

11/04

Estimaciones y MVP EcoPark

Hipotesis: Existen personas dispuestas a gestionar su experiencia en el parque mediante una aplicación móvil. Comprando entradas y reservando actividades.

la forma mas facil de validar una hipotesis es ver si el usuario esta dispuesto a usarla

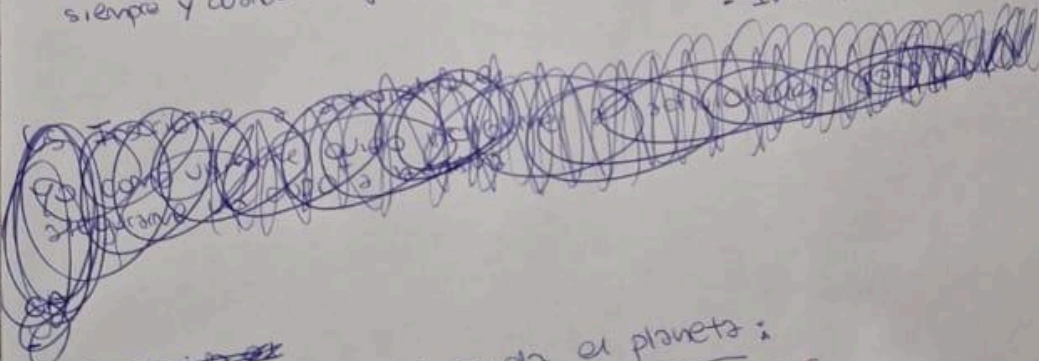
- US incluidas: → una US con la que no necesariamente va al MVP
- Comprar entradas [Justificación: Envío de mail]
 - Inscribirse a actividades [Justificación: Envío de mail]
 - Ver horarios de Alimentación
 - Ver mapa interactivo
 - Ver actividades (podría ir o no siempre y cuando este justificado)

Justificación: (Por que se incluye una US y por que no otra)

no necesariamente van US 200

No incluye:

- Parametrización del software
- Creación de usuarios
- Localización en tiempo real
- Recordatorios
- Validación de entradas
- Inscripciones



TP3 Reacta tus prendas cuida el planeta:

Yo se que hay gente que quiere comprar

Hipotesis: Hay personas dispuestas a comprar y vender ropa usada a través de una plataforma web.

Justificación:

Incluye:

- comprar una prenda X
- completar formulario de venta de prenda
- ver catalogo de prendas disponibles
- Registrar usuario vendedor por mail/google
- Aceptar propuesta de venta
- completar formulario del producto
- Enviar prenda para vender
- comprar prendas
- devolución de prendas

No incluye: