

Curso UX/UI Design - Nuclio

Leonardo Rodenas E.

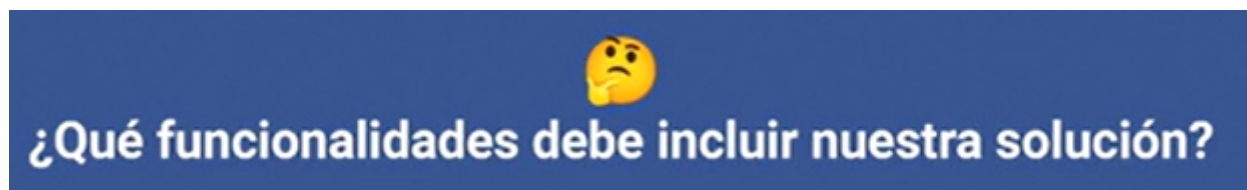
Temario

2. Estrategia de Producto - Definiendo el Producto Minimo Viable (MVP)

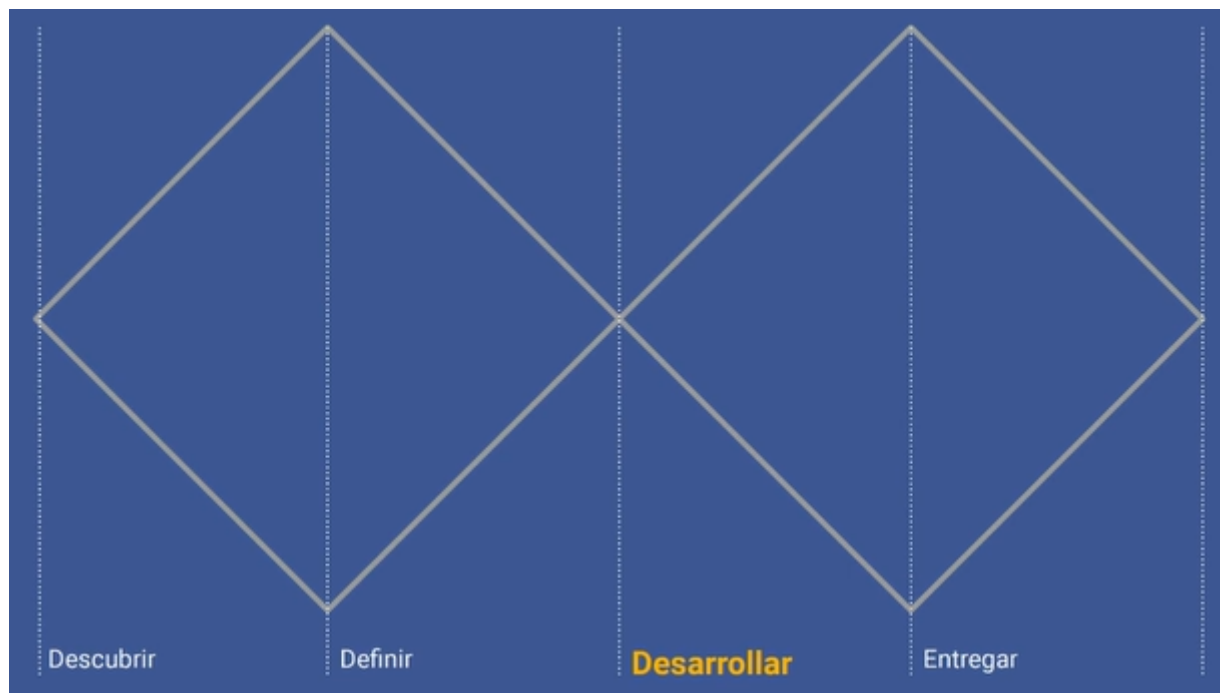
- 2.1 [Definiendo el MVP](#)
- 2.2 [Value proposition Canvas](#)
- 2.3 [User Stories: Jobs to be done](#)
- 2.4 [¿Qué es un MVP?](#)
- 2.5 [Priorización de tareas: MoSCoW](#)

2. Estrategia de producto: Definiendo el Producto Minimo Viable (MVP)

2.1 Definiendo el MVP (Minimum Viable Product):



Modelo Doble Diamante de Design Thinking



En este esquema a la izquierda (primer diamante) tenemos la parte del problema y a la derecha (segundo diamante) la parte de la solución. En el módulo anterior nos centramos más en el primero, descubriendo a nuestros usuarios, definiendo sus problemas y empatizando con el usuario para comprender su actitud y

trazar un plan de acción ante lo que lo problema o frustra. Ahora en este módulo profundizaremos un poco más en desarrollar las primeras ideas y soluciones, poder acotar que es lo que tiene sentido al momento de plantear la solución y generar el primer borrador de lo que será la entrega del prototipo/producto.

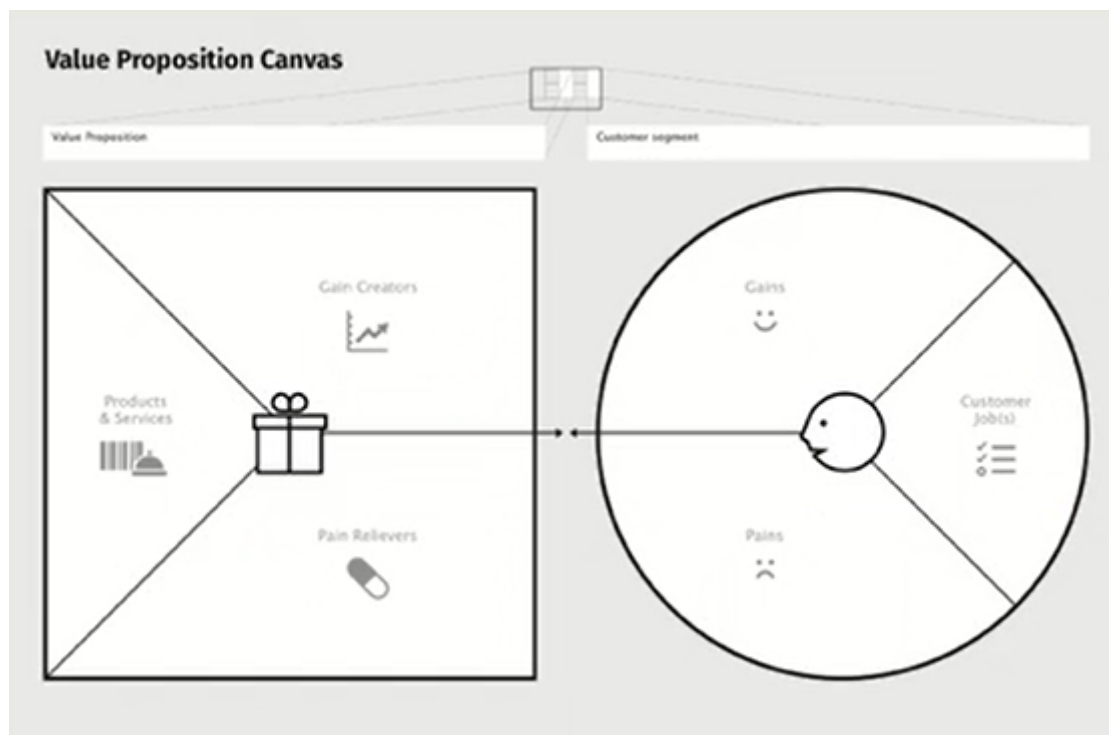
2.2 Value proposition Canvas:

Lo que se propone en este punto es definir la propuesta de valor, lo cual pretende explicar como tú producto o servicio resuelve los problemas del usuario, o mejora su situación a través de un beneficio o valor (ejemplo: Ipod, slogan "1000 canciones en tu bolsillo").

¿Cómo definimos una propuesta de valor?

Observar usuario > entenderlo > crear valor

Siempre tener en cuenta el no generar un producto que se sustente sólo en la idea y que no pretenda resolver un problema o generar un beneficio.



Este Canvas se usa para definir y/o optimizar propuestas de valor, y para crearlo se divide la técnica en dos partes (img de arriba): a la derecha se define el perfil del usuario (Customer profile) y a la izquierda donde se define el producto.

JOBS

JOBS

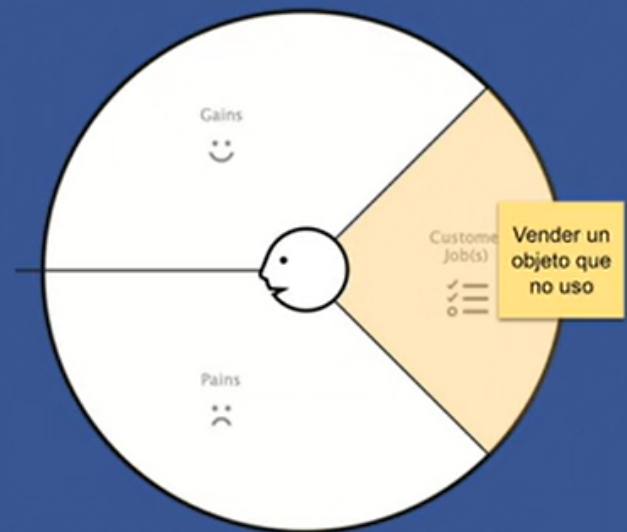
Tareas, actividades, acciones o “trabajos” que realiza nuestro usuario en relación al problema que pretende resolver nuestro producto.

Se deben priorizar:
Importantes - Insignificantes

Jobs funcionales: Completar una tarea concreta:
“Descargar un programa de videocalls”.

Jobs sociales: Cuando el usuario quiere mejorar la percepción que tienen los demás:
“Ser percibido como un buen profesional”.

Jobs emocionales: Cuando el usuario busca una sensación, un estado emocional:
“Buscar un momento de calma”.



EJEMPLO

Estudiantes master UX/UI

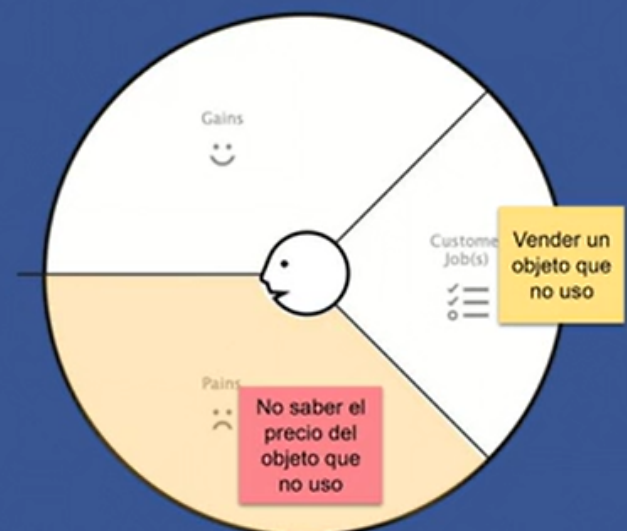
- Jobs funcionales

PAINS

PAINS

Aquí listamos lo que molesta al usuario, antes, durante y después de intentar llevar a cabo los Jobs.

Se deben priorizar:
Extremos - moderados



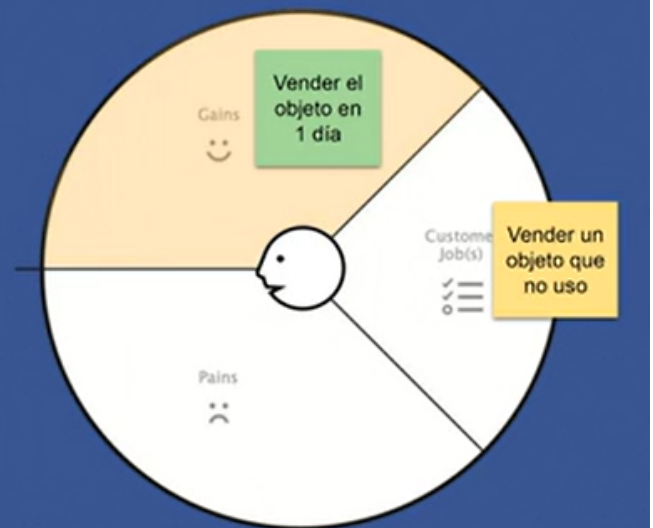
GAINS

GAINS

Aspectos que podrían **sorprender** de forma muy positiva a nuestros usuarios, y que ayudan a realizar el *Job* de forma evidente.

⚠ **No es lo contrario a Pain.**

Se deben priorizar:
Esenciales - "Nice to have"



Cuando ya tenemos realizado el Customer Profile, se comienza a generar el mapa de valor, que es en donde ya definimos nuestro producto. Acá se procede en el siguiente orden:

PAIN RELIEVERS - GAIN CREATORS - PRODUCTS & SERVICES

PAIN RELIEVERS

Aquí trataremos de solventar y/o aliviar un *Pain*. Cada *Pain reliever* hace referencia a uno o más *Pains*.

Ej: "No sé el precio del objeto que no uso". -> "Conocer el precio de objetos similares."

Se pueden priorizar:
Esenciales - "Nice to have"

GAIN CREATORS

Aquí trataremos de definir de qué forma podemos potenciar un *Gain*.

Ej: "Vender el objeto en muy poco tiempo" -> "Ponerme en contacto con gente interesada y cercana."

Se pueden priorizar:
Esenciales - "Nice to have"

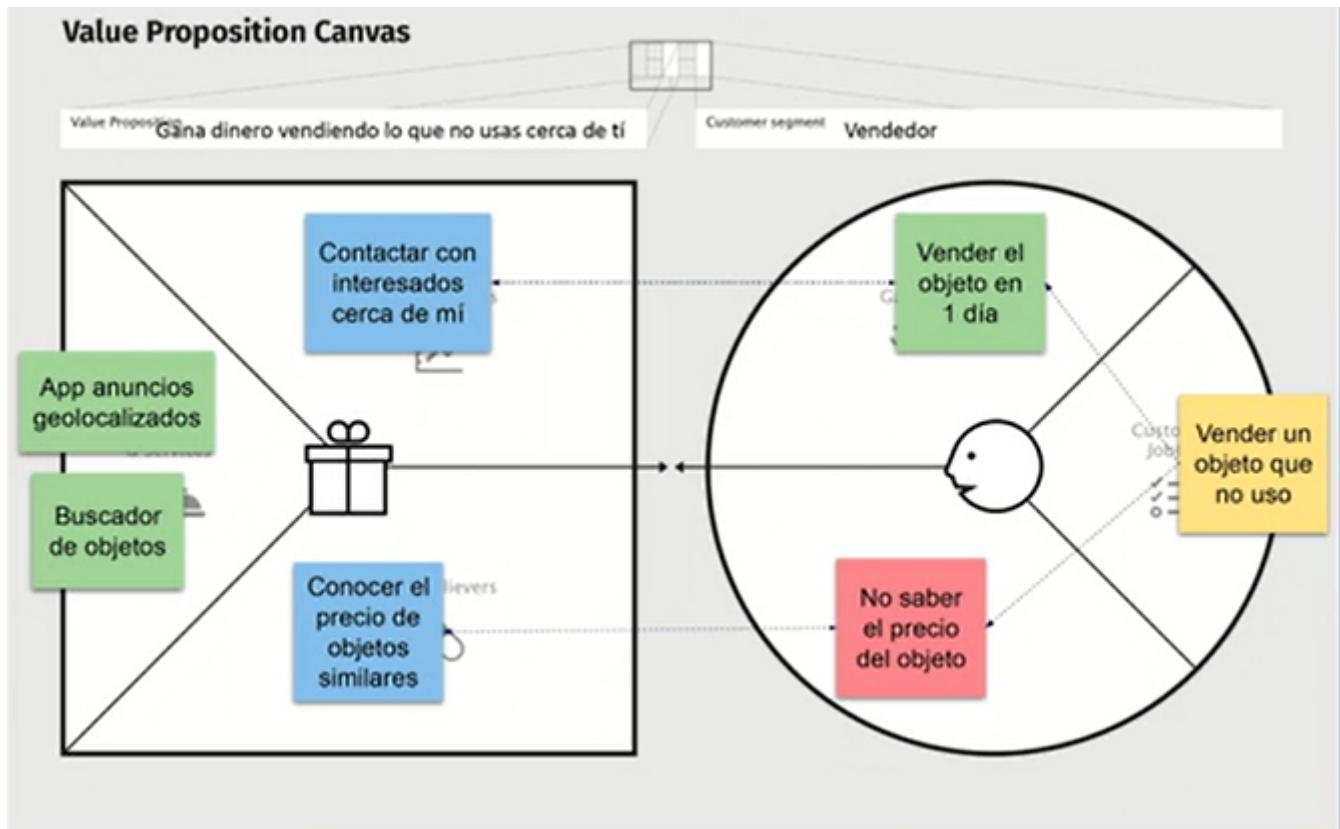
PRODUCTS & SERVICES

Aquí listamos todo lo que ofreceremos al usuario. El conjunto de servicios o funcionalidades que ayudarán al usuario a resolver sus *Jobs*.

Se pueden priorizar:
Esenciales - "Nice to have"



Resumiendo este ejemplo y juntando sus partes en un sólo Value Proposition Canvas (VPC) obtenemos lo siguiente:



Notar que arriba se ofrece una proposición de valor "Gana dinero vendiendo lo que no usas cerca de ti" (se soluciona algo que potencia el "Gain" y se le agrega un valor que solventa el "Pain").

¿Problemas comunes al rellenar el Canvas?:

1. Tener definido el producto, pero no el usuario.
2. Hemos llenado el Canvas... ¿Ahora qué?, ¿cómo uno ambas partes?, pues con el **"Market Fit"**, el cual debe analizar si es que ya existen antes soluciones al problema disponibles en el mercado, si podemos mejorarlas y de ser así... !!!CÓMO LE DAMOS VALOR A NUESTRA SOLUCIÓN!!!

2.3 User Stories: Jobs to be done:

Una feature es una parte del sistema o del producto que queremos construir, y la "User story" es la forma de describir esa parte. Estas historias de usuario nacen en el contexto de las metodologías ágiles, donde pretendemos desarrollar productos de forma iterativa y rápidamente. Las principales características que tienen que tener estas historias de usuario y features son:

1. **Independientes:** No dependen de otras para completarlas (así podemos subir a producción una parte de la funcionalidad y validar si ya agrega valor al usuario.).
2. **Negociables:** Se tienen que poder debatir (se bese seguir un consenso de negocio al desarrollar el producto).
3. **Valoradas:** hay que saber que valor aportan para poder priorizar (que hacemos antes o despues).
4. **Estimables:** Estimación de su coste (Tallas, historia de referencia, puntos de historia, etc).
5. **Pequeñas:** Si son muy grandes, dividir las en más pequeñas (pequeñas para poder validarlas rápido y ser más ágiles).
6. **Verificables:** Los criterios de aceptación verifican si se cumple lo esperado (ejemplo, con tarea terminada debe pasar X cosa, esto debe poder verificarse).

Historias de Usuario: ¿Cómo escribirlas?

Como <tipo de usuario> Quiero <acción> Para <Objetivo/beneficio>	Tener siempre claro y presente: <ul style="list-style-type: none"> - User personas. - Customer jobs. - Pains & Gains.
---	---

Una vez discutida la historia de usuario y validada, se redactan los criterios de aceptación que no es más que un listado de requisitos para que se considere la historia / tarea como cumplida (no hay una forma única, depende de la experiencia del team, del lenguaje, etc.).

Épica = Nivel de agrupación por encima de las historias de usuario. Conjunto de historias / tareas divisibles en unidades más pequeñas. Ejemplo: Épica: Crear un proceso de Checkout || Historia del usuario = pagar con Paypal).

Qué NO es una historia de usuario:

Como Diseñador de Instagram <- No es un usuario
Quiero Un registro al acceder a Instagram <- No es una acción -> **Es una funcionalidad**
Para conseguir más usuarios registrados <- No es un beneficio -> **usuario (Es un beneficio para la empresa, no para el usuario)**

- El usuario se tiene que poder registrar con facebook.
- El usuario se tiene que poder registrar con Google.
- El usuario tiene que poder crear una cuenta nueva.

<- Cada criterio es una Historia independiente. No es suficientemente preciso.

Qué SI es una historia de usuario:

Como usuario que instala Instagram por primera vez
Quiero crear mi perfil de usuario
Para empezar a compartir contenido personalizado con otros usuarios

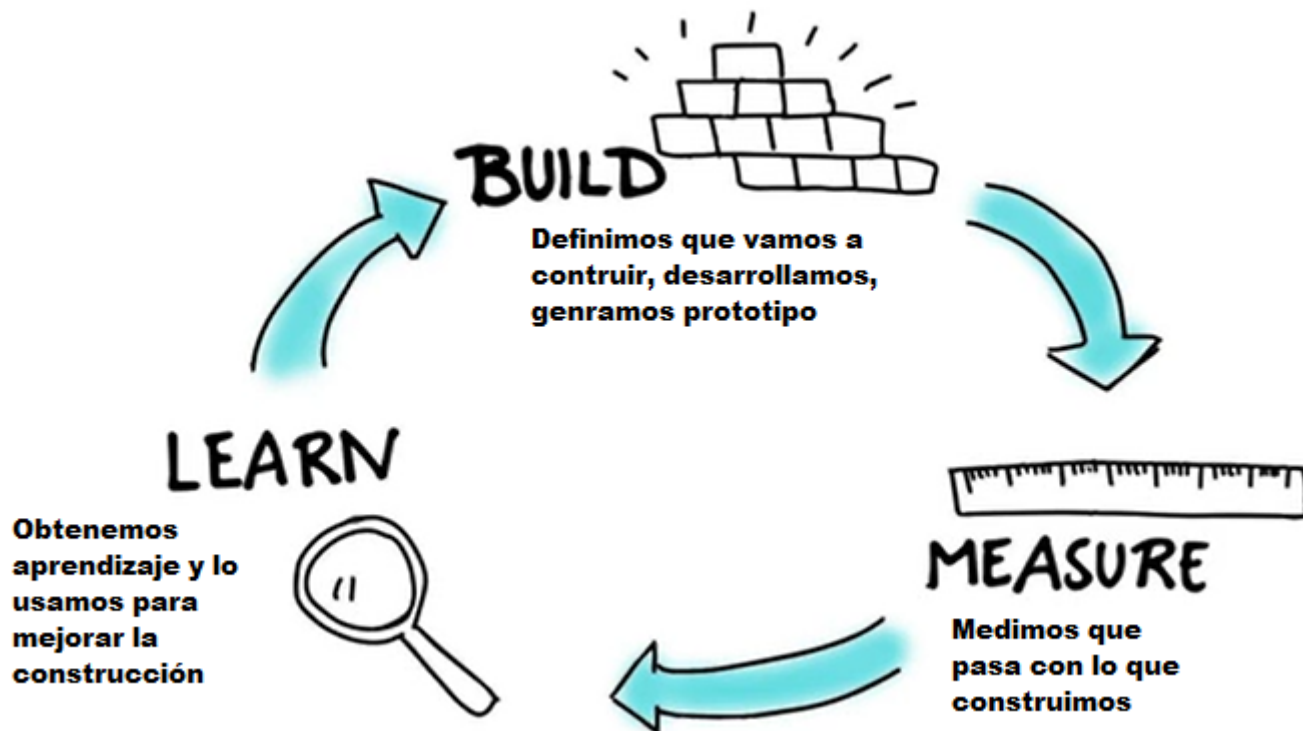
Criterios de aceptación:

- El usuario puede introducir: nombre, correo y contraseña en un formulario.
- Se mostrará formulario al acceder por 1ª vez después del splash screen.
- Si hay algún error al introducir los datos se mostrará un mensaje *inline*.
- Al click en aceptar, se mandará mail para la validación y se accederá a la app.

Una historia correcta es importante acompañarla de mockups o diseños que la respalden y ayuden a explicarla

2.4 ¿Qué es un MVP?:

Son las siglas de Producto Mínimo Viable (Minimum Viable Product = MVP). MVP nace dentro del contexto de **Lean Startup**, el cual tiene como objetivo reducir el riesgo a la hora de lanzar productos o servicios al mercado (se basa en aprendizaje, y cuanto más rápido y barato mejor).



Se busca testear la propuesta de valor lo antes posible ya aprender

Pivotar y Perseverar: Las hipótesis falsas nos obligan a **pivotar** (buscar otra forma de entregar valor al usuario) respecto a la idea inicial. Proponiendo alternativas cada vez más próximas a la estrategia óptima que nos permitan **perseverar**. Así, el mercado es el que guía nuestra estrategia (Ejemplo: Instagram antes de ser conocido era Burnbn, hacia muchas cosas pero no era especialmente bueno en ninguna).

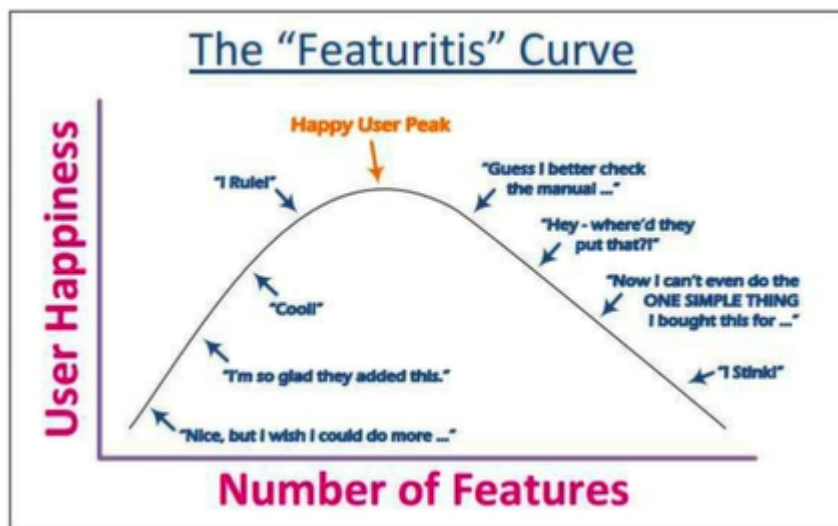
Tomando en cuenta esto, el MVP permite recabar la mayor cantidad de aprendizaje validado sobre los clientes con el menor esfuerzo posible. En este aspecto es importante tener y mostrar un producto completo, no sólo funcional, pues tal vez el producto falla en alguna otra etapa que no sabremos porque nunca la presentamos.



2.5 Priorización de tareas: MoSCoW:

¿Por qué historia o funcionalidad empezamos a desarrollar nuestro producto?: Es importante validar cuanto antes que realmente da valor al usuario, para eso debemos tener claras la features, pero hay un peligro asociado a esto, caer en riesgo de "Featuritis".

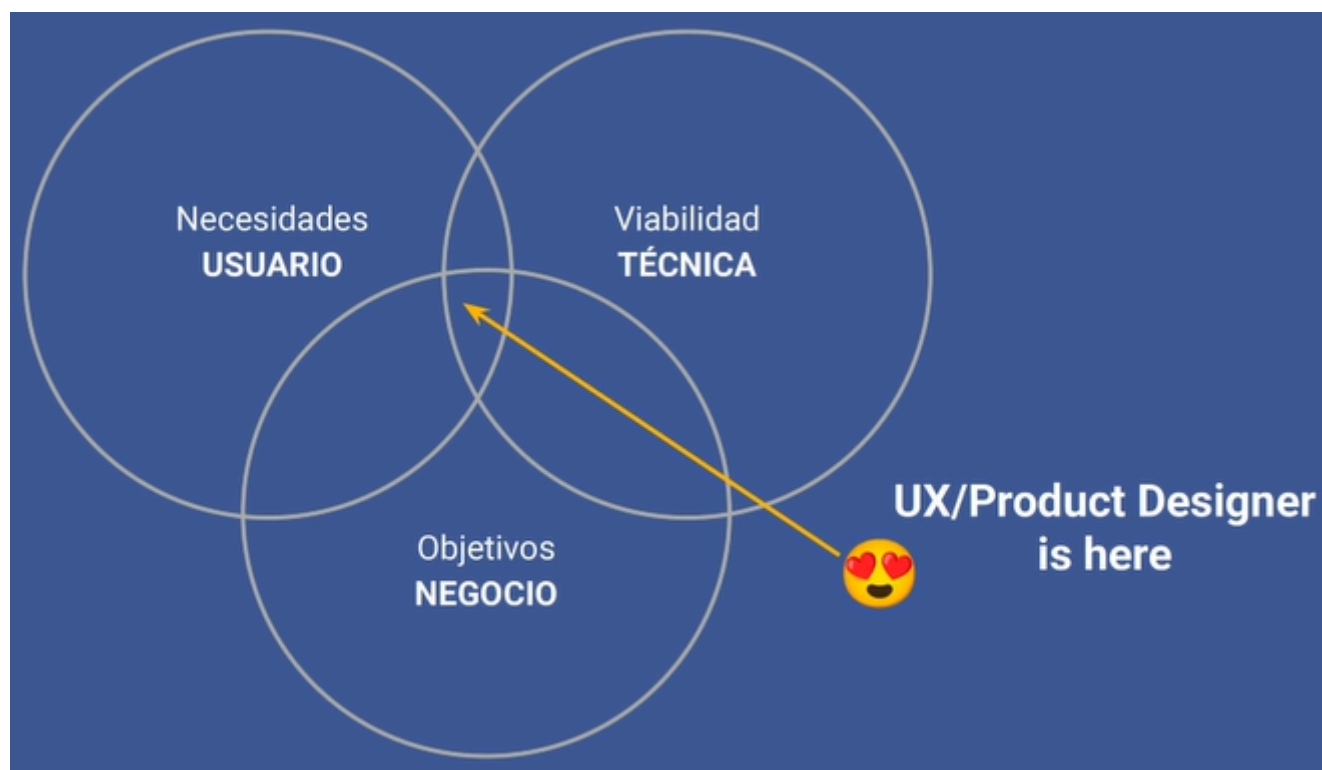
Riesgo de Featuritis:



Es intentar agregar valor al producto/servicio agregando más y más funcionalidades, hasta llegar a un punto en donde ofrecemos tanto que el usuario se pierde porque no entiende para que sirve el producto o que intenta solucionar.

La idea para solucionar esto es centrarse en lo que realmente valor al usuario y romper con el egocentrismo a la hora de crear el producto (no basarse en "A mi me gusta", "yo como usuario", "Mi mamá no lo usaría". "El CEO lo quiere así", etc) y ser más centrados en el usuario y en la tecnología que tenemos para crear el producto.

Criterio de priorización: Usuario / Negocio / Tech



¿Cómo llegamos al equilibrio en el diagrama?: Se puede hacer mediante una Matriz de Priorización, un Kano o un MoSCoW

MATRIZ DE PRIORIZACIÓN



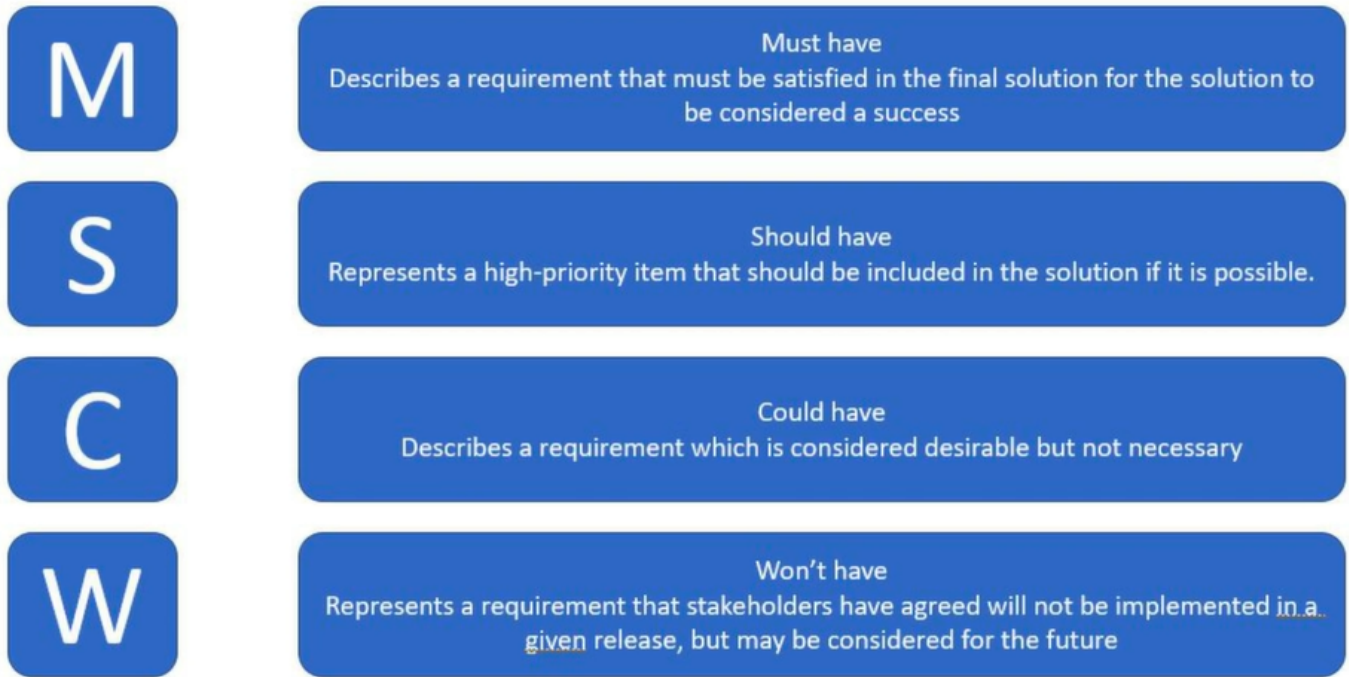
Notar los 4 cuadrantes y como relaciona el valor al negocio y al usuario, dando como resultado siempre lo mejor es un alto valor para ambos.

KANO - ¿Cómo te sientes teniendo la función v/s no teniendola?



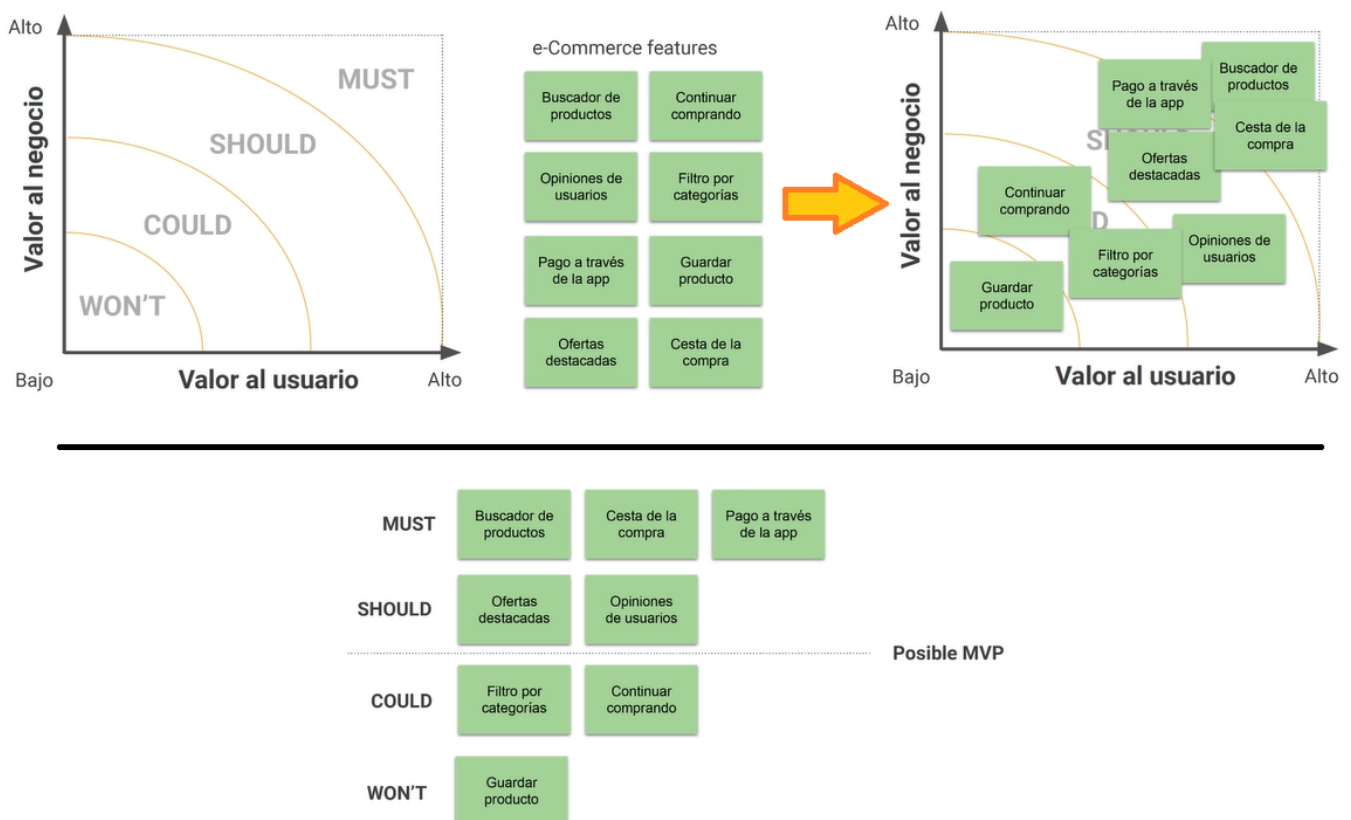
Se basa en una encuesta inicial respondible del 1 al 10, determina que SI O SI tiene que estar en la app y que no es necesario que esté. Identifica que quiere el usuario en la app (Delighters), funciones por funcionalidad en la app (Performance) y lo que no puede faltar en la app (Basics).

MoSCoW



Existe la posibilidad de mezclar métodos, como se ve a continuación (Ejemplo: e-Commerce):

Matriz + MoSCoW



Los criterios claros y la mentalidad del dato (data-driven, intentar tomar decisiones basadas en datos, no subjetivas) nos ayudan a ser objetivos (Aunque a veces es importante escuchar la intuición ;D).