

Design Thinking

Es un método para generar ideas innovadoras que centra su eficacia en entender y dar solución a las necesidades reales de los usuarios. Proviene de la forma en la que trabajan los diseñadores de producto. De ahí su nombre, que en español se traduce de forma literal como "Pensamiento de Diseño", aunque nosotros preferimos hacerlo como "La forma en la que piensan los diseñadores".

<http://www.designthinking.es/inicio/>

Ejemplo de la utilización de una técnica por cada una de las fases



SCRUM

SCRUM : Metodologia agil para la creacion de productos en proyectos medianamente grandes.
Se centra en el trabajo en equipo y en obtener en el menor tiempo el mejor resultado posible

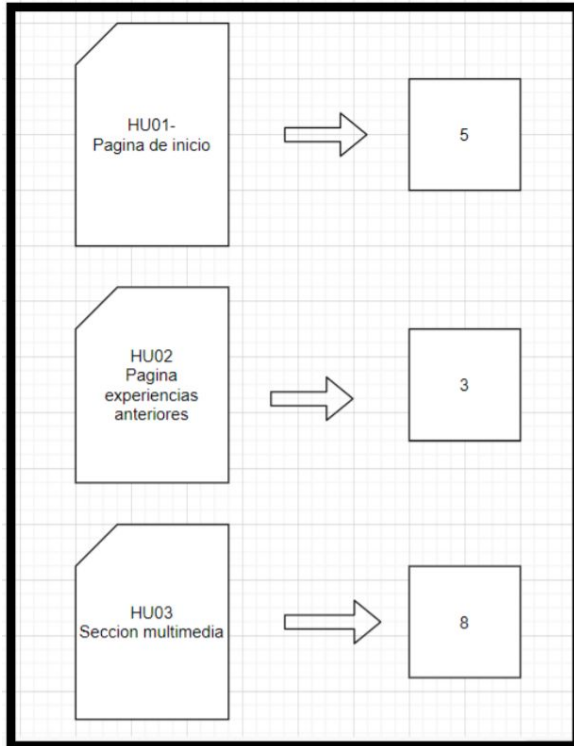
Backlog del producto.

Id	Siendo un	Necesito	Para lograr
1	Usuario no registrado	Ver la pagina de inicio	Tener informacion de que se trata la empresa
2	Usuario no registrado	Poder acceder a una seccion de experiencias de usuarios	Informarme de que opinion tuvieron los usuarios anteriores al contratar el servicio
3	Usuario no registrado	Poder ver videos de como trabaja el equipo	Ver si me sentiria comodo teniendo personas trabajando en mi casa
4	Usuario no registrado	ver la pagina de Login	Ingresar o registrarme al sistema
5	Ciente	Poder acceder a un calendario online	Fijar dias y horarios de contratacion del servicio
6	Ciente	Acceder a una parte de adaptacion del servicio	Que el servicio se adapte lo mejor posible a lo que necesito
7	Ciente	Acceder a una seccion de encuesta permanente	Informarle a la empresa como estan trabajando sus empleados
8	Ciente	Acceder a una seccion de pago del servicio	Pagar el servicio
9	Ciente	Acceder a una camara prendida constantemente	Ver como esta mi hijo en todo momento
10	Ciente	Poder contactarme con las personas que estan trabajando en mi casa	Comunicarme con ellas por el motivo que sea
11	Empleado	Poder acceder al calendario online	Ver que dias tengo que trabajar , en que horario , lugar y con que requisitos
12	Empleado	Poder contactar al cliente	Comunicarme con ellas por cualquier motivo
13	Empleado	Poder contactarme con otros compañeros de trabajo	Comunicarme con alguien para que me ayude en el trabajo a realizar
14	Gerente	Poder acceder a un diagrama de causa-efecto	Comprobar que las acciones realizadas tienen los resultados esperados
15	Gerente	Tener a mi disponibilidad un diagrama de dispersion	Mejorar la calidad de la empresa
16	Gerente	Tener un diagrama de flujo	Mejorar la eficiencia de la empresa
17	Gerente	Poder revisar el histograma	Mejorar la eficacia de la empresa
18	Gerente	Tener un diagrama de pareto	Tener todos los datos organizados
19	Gerente	Poder ver una hoja de recogida de datos	Revisar los datos cuando lo sienta necesario
20	Gerente	Tener disponible las graficas de control	Mejorar la calidad

El Backlog se crea a partir de las historias de usuario como se ve en la imagen anterior.

Una vez obtenido el Backlog podemos pasar a hacer un planning poker para darle puntaje a las tareas

Planning Poker

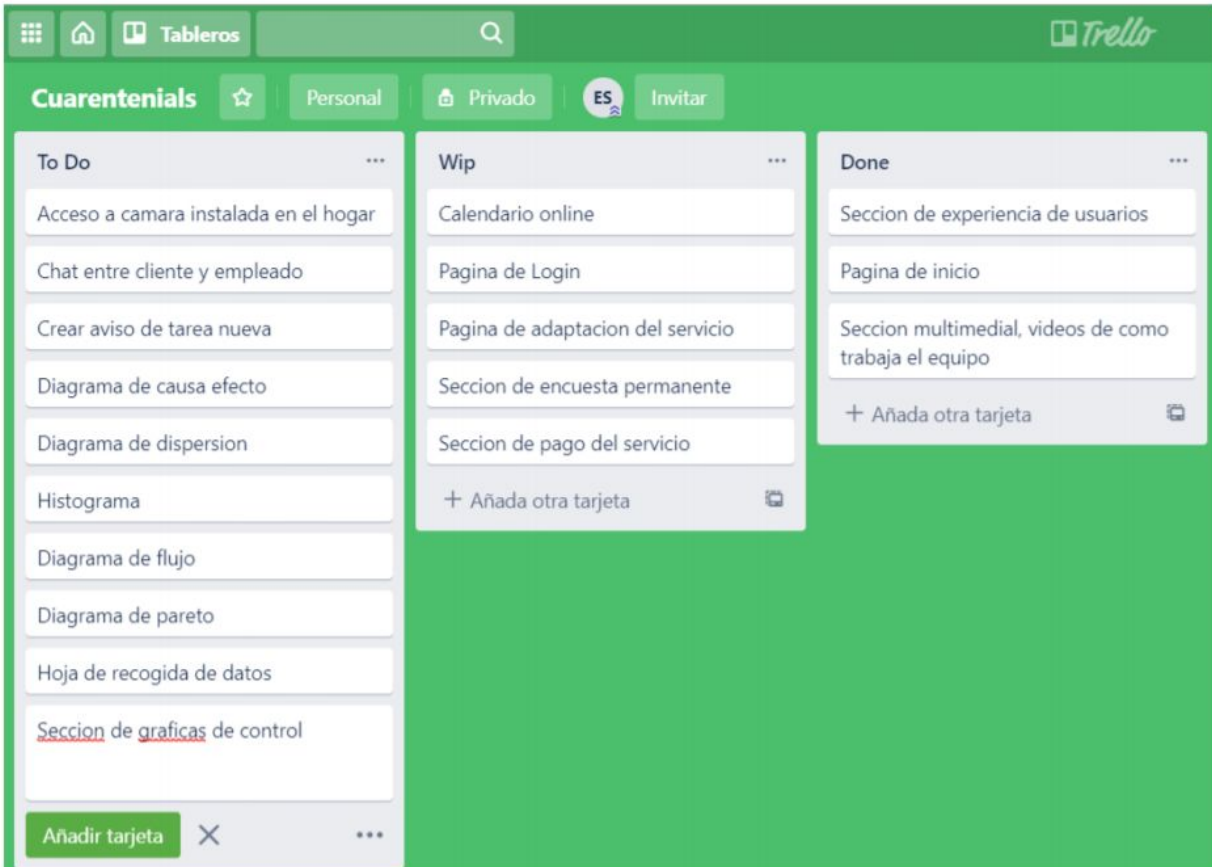


Tasking:

Aqui , con el Backlog creado anteriormente y con las historias de usuario valoradas en el planing poker , procedo a estimar la cantidad de horas corresponfientes a cada tarea. Todas las tareas duran 8 horas o menos... Las tareas que duraban mas tiempo fueron descomprimidas y subdivididas en subtareas agregadas al Backlog

Id	Siendo un	Necesto	Para lograr	Puntos	Estado	Horas	Release
1	Usuario no registrado	Ver la pagina de inicio	Tener informacion de que se trata la empresa	5	Done	7	Release 1
2	Usuario no registrado	Poder acceder a una seccion de experiencias de usuarios	Informarme de que opinion tuvieron los usuarios anteriores al contratar el servicio	3	Done	4	Release 1
3	Usuario no registrado	Poder ver videos de como trabaja el equipo	Ver si me sentiria comodo teniendo personas trabajando en mi casa	8	Done	8	Release 1
4	Usuario no registrado	ver la pagina de Login	Ingresar o registrarme al sistema	7	Wip	8	Release 1
5	Cliente	Poder acceder a un calendario online	Fijar dias y horarios de contratacion del servicio	5	Wip	5	Release 1
6	Cliente	Acceder a una parte de adaptacion del servicio	Que el servicio se adapte lo mejor posible a lo que necesito	3	Wip	4	Release 1
7	Cliente	Acceder a una seccion de encuesta permanente	informarle a la empresa como estan trabajando sus empleados	2	Wip	6	Release 1
8	Cliente	Acceder a una seccion de pago del servicio	Pagar el servicio	6	Wip	2	Release 1
9	Cliente	Acceder a una camara prendida constantemente	Ver como esta mi hijo en todo momento	4	To Do	5	Release 1
10	Cliente	Poder contactarme con las personas que estan trabajando en mi casa	Comunicarme con ellas por el motivo que sea	7	To Do	6	Release 2
11	Empleado	Poder acceder al calendario online	Ver que dias tengo que trabajar , en que horario , lugar y con que requisitos	9	To Do	8	Release 2
12	Empleado	Poder contactar al cliente	Comunicarme con ellas por cualquier motivo	5	To Do	6	Release 2
13	Empleado	Poder contactarme con otros compañeros de trabajo	Comunicarme con alguien para que me ayude en el trabajo a realizar	2	To Do	3	Release 2
14	Gerente	Poder acceder a un diagrama de cause-efecto	Comprobar que las acciones realizadas tienen los resultados esperados	8	To Do	8	Release 3
15	Gerente	Tener a mi disponibilidad un diagrama de dispersion	Mejorar la calidad de la empresa	7	To Do	8	Release 3
16	Gerente	Tener un diagrama de flujo	Mejorar la eficiencia de la empresa	5	To Do	5	Release 3
17	Gerente	Poder revisar el histograma	Mejorar la eficacia de la empresa	3	To Do	4	Release 3
18	Gerente	Tener un diagrama de pareto	Tener todos los datos organizados	2	To Do	6	Release 3
19	Gerente	Poder ver una hoja de recogida de datos	Revisar los datos cuando lo sienta necesario	6	To Do	2	Release 3
20	Gerente	Tener disponible las graficas de control	Mejorar la calidad	4	To Do	5	Release 3

Tablero Trello



<https://trello.com/b/H5fvj6F6/entrega-front-end>

Roles mas importantes de Scrum

Product Owner: Responsable de maximizar el valor del producto.

Scrum Master: Es el que se encarga de que todos en el equipo utilicen Scrum..Es el Facilitador

Team Members: El equipo de desarrollo. Son los encargados de crear el producto.

Equipo autoorganizado de entre 5 y 9 miembros.

Sprint: La creacion del producto se centra en Sprints , que son ciclos de trabajo de entre 2 y 4 semanas.

Reunion de planificacion: Es la que se lleva a cabo cuando comienza un Sprint.

Ahi el cliente presenta una lista de requisitos: El PRODUCT BACKLOG

Aca el product owner la ordena por prioridades.

Con ese product Backlog el equipo realiza el SPRINT BACKLOG

Reunion de planificacion: Cada dia se lleva adelante una reunion (de maximo 15 minutos) donde cada miembro dice

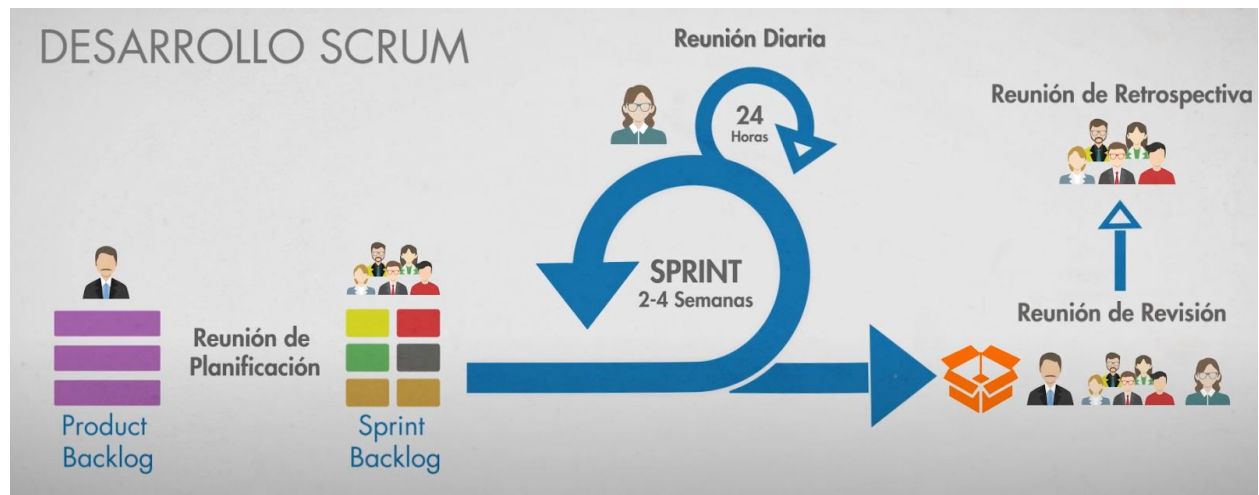
1-que hizo desde la ultima reunion que tuvieron

2-Que voy a hacer a partir de este momento

3-Que impedimentos tengo o voy a tener

Reunion de revision: Al finalizar el Sprint , se realiza una reunion para revisar lo realizado antes de entregarselo al cliente.

Reunion de retrospectiva: Terminado el Sprint y entregado esa parte del producto el equipo se reúne para ver que hizo bien , que hizo mal y como mejorarlo. La finalidad es mejorar la productividad.



<https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish-European.pdf>

eXtreme Programming

Introducción a la Programación eXtrema eXtreme Programming (programación extrema) también llamado XP, es una metodología que tiene su origen en 1996.

A diferencia de Scrum, XP propone solo un conjunto de prácticas técnicas, que aplicadas de manera simultánea, pretenden enfatizar los efectos positivos de en un proyecto de desarrollo de Software

eXtreme Programming propone 12 prácticas técnicas, simples, de fácil comprensión y que aplicadas en forma conjunta, garantizan un mejor resultado del Proyecto

PRÁCTICA #1: CLIENTE IN-SITU (ON-SITE CUSTOMER): La comunicación cara a cara con el cliente es fundamental, ya que a partir de ésta, el equipo avanzará más rápidamente en el proyecto,

PRÁCTICA #2: SEMANA DE 40 HORAS (40 HOUR WEEK): eXtreme Programming asegura la calidad del equipo, considerando que éste, no debe asumir responsabilidades que le demanden mayor esfuerzo del que humanamente se puede disponer

PRÁCTICA #3: METÁFORA (METAPHOR) A fin de evitar los problemas de comunicación que suelen surgir en la práctica, entre técnicos y usuarios, eXtreme Programming propone el uso de metáforas, intentando hallar un punto de referencia que permita representar un concepto técnico con una situación en común con la vida cotidiana y real.

PRÁCTICA #4: DISEÑO SIMPLE (SIMPLE DESIGN) Esta práctica, deriva de un famoso principio técnico del desarrollo de software: KISS. Cuyas siglas derivan del inglés "Keep it simple, stupid!" -¡Manténlo simple, estúpido!-

PRÁCTICA #5: REFACTORIZACIÓN (REFACTORING) La refactorización (o refactoring) es una técnica que consiste en modificar el código fuente de un software sin afectar a su comportamiento externo.

PRÁCTICA #6: PROGRAMACIÓN DE A PARES (PAIR PROGRAMMING) Otra de las prácticas técnicas fundamentales de eXtreme Programming, es la programación de a pares. Se estila (aunque no de forma constante) programar en parejas de dos desarrolladores, los cuales podrán ir intercambiando su rol, en las sucesivas Pair Programming. ¿En qué consiste la programación de a pares? Sencillo. Dos programadores, sentados frente a una misma computadora, cada uno cumpliendo un rol diferente. Las combinaciones y características de este rol, no solo son variables, sino que además, son inmensas, permitiendo la libertad de "ser originalmente creativos". Algunas de estas combinaciones (tal vez las más frecuentes), suelen ser: • Uno escribe el código, mientras que otro lo va revisando; • Un programador más avanzado, programa, mientras va explicando lo desarrollado a otro menos experto;

- Los dos programadores piensan en como resolver el código, y uno de ellos lo escribe. De esa forma y según las necesidades de cada equipo, las funciones de cada programador, podrán variar ilimitadamente.

PRÁCTICA #7: ENTREGAS CORTAS (SHORT RELEASES) Se busca hacer entregas en breves lapsos de tiempo, incrementando pequeñas funcionalidades en cada iteración. Esto conlleva a que el cliente pueda tener una mejor experiencia con el Software, ya que lo que deberá probar como nuevo, será poco, y fácilmente asimilable, pudiendo sugerir mejoras con mayor facilidad de implementación.

PRÁCTICA #8: TESTING En esta práctica, podemos encontrar tres tipos de test -pruebas- propuestos por eXtreme Programming:

1. Test Unitarios
2. Test de aceptación
3. Test de integración

PRÁCTICA #9: CÓDIGO ESTÁNDAR (CODING STANDARDS) Los estándares de escritura del código fuente, son esenciales a la hora de programar. Permiten hacer más legible el código y más limpio a la vez de proveer a otros programadores, una rápida visualización y entendimiento del mismo

PRÁCTICA #10: PROPIEDAD COLECTIVA (COLLECTIVE OWNERSHIP) Para eXtreme Programming no existe un programador “dueño” de un determinado código o funcionalidad. La propiedad colectiva del código tiene por objetivo que todos los miembros del equipo conozcan “qué” y “cómo” se está desarrollando el sistema, evitando así, lo que sucede en muchas empresas, que existen “programadores dueños de un código” y cuando surge un problema, nadie más que él puede resolverlo, puesto que el resto del equipo, desconoce cómo fue hecho y cuáles han sido sus requerimientos a lo largo del desarrollo.

PRÁCTICA #11: INTEGRACIÓN CONTINUA (CONTINUOUS INTEGRACIÓN) La integración continua de XP propone que todo el código (desarrollado por los miembros del equipo) encuentren un punto de alojamiento común en el cuál deban enviarse los nuevos desarrollos, diariamente, previo correr los test de integración, a fin de verificar que lo nuevo, no “rompa” lo anterior

Estos puntos de alojamiento en común, suelen ser repositorios, los cuales pueden manejarse ampliando las ventajas de la integración, mediante software de control de versiones como Bazaar, Git o SVN (entre otros).

PRÁCTICA #12: JUEGO DE PLANIFICACIÓN (PLANNING GAME) Con diferencias pero grandes similitudes con respecto a la planificación del Sprint en Scrum, la dinámica de planificación de eXtreme Programming llevada a cabo al inicio de la iteración, suele ser la siguiente: 1. El cliente presenta la lista de las funcionalidades deseadas para el sistema, escrita con formato de “Historia de Usuario”, en la cual se encuentra definido el comportamiento de la

misma con sus respectivos criterios de aceptación. 2. El equipo de desarrollo estima el esfuerzo que demanda desarrollarlas, así como el esfuerzo disponible para el desarrollo (las iteraciones en XP, suelen durar 1 a 4 semanas) . Estas estimaciones pueden hacerse mediante cualquiera de las técnicas de estimación, vista anteriormente. 3. El cliente decide que Historias de Usuario desarrollar y en qué orden.

LEAN

Los desperdicios de Lean, los 7 mudas

Muda = Desperdicio, despilfarro, todo aquello que no aporta valor al cliente.

- **Sobreproducción:** Consiste en producir más de lo demandado, antes de que el cliente lo requiera. Está considerado el peor de las mudas, ya que utilizamos esfuerzos y recursos que deberían haberse usado en otras cosas más necesarias.
- **Esperas:** Estos son los típicos cuellos de botella donde se generan esperas.
- **Movimientos innecesarios:** Movimientos que se pueden evitar como búsqueda de herramientas o documentos ya que disminuyen el tiempo dedicado a realizar lo que realmente aporta valor.
- **Transporte:** Debemos minimizar los movimientos innecesario de materiales, de nuevo, esto no aportan ningún valor.
- **Sobreprocesos:** Se trata de reducir fases que pueden ser innecesarias (trabajo extra).
- **No calidad (defectos):** Los defectos de producción y los errores de servicio provocan un consumo innecesario de recursos tanto materiales como de tiempo (atender quejas) provocando insatisfacción del cliente.
- **Inventario:** El stock acumulado que sobrepase lo necesario para cubrir las necesidades del cliente generan un impacto negativo tanto costes como en espacio.

Galletas Julieta

<https://youtu.be/cnA9kQJMwe4>

Como resolverlos: <https://es.slideshare.net/vxdominguez/guia-lean>