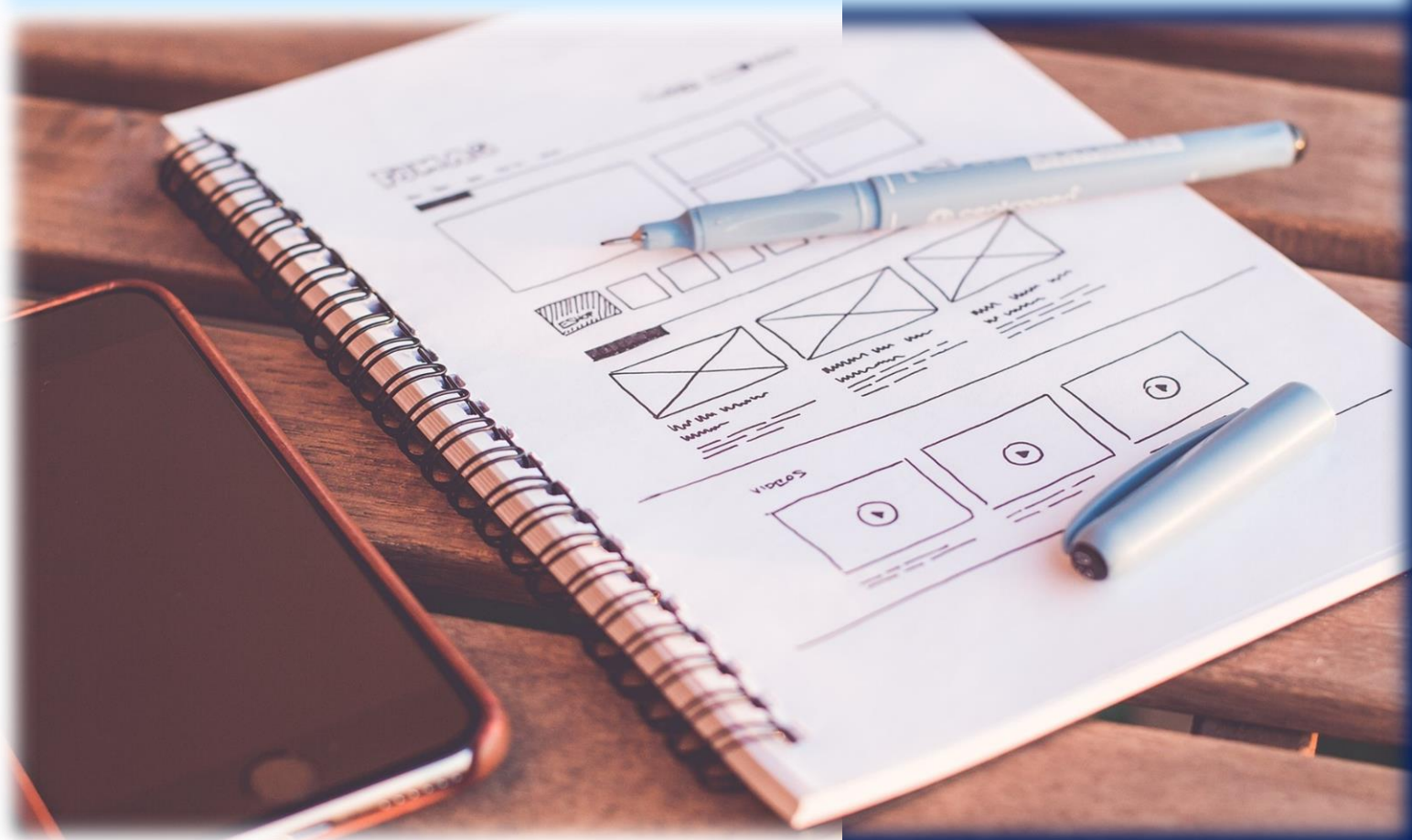


2023/24

U.D.1. PLANIFICACIÓN Y DISEÑO DE INTERFACES WEB.

1.3 Generación de documentos y sitios web.





ÍNDICE

1. Guías de estilo	3
1.1 ¿Cómo puedo crear una guía de estilo para un sitio web?	8
1.2 El checklist de una buena guía de estilo.	9
2. Metodología de trabajo (tradicional)	11
2.1 Estrategia y planificación	11
2.2 Análisis	15
2.3 Diseño	18
2.4 Desarrollo o construcción	23
2.5 Pruebas y Despliegue	25
2.6 Documentación	26
2.7 Mantenimiento	27
3. Anexo I - Metodología SCRUM	30
3.1 Desarrollo ágil de software	30
3.2 Principios de la metodología ágil	31
3.3 Sprints	32
3.4 Roles – Scrum	32
3.4.1 Product Owner	32
3.4.2 Scrum Master	33
3.4.3 Development team	34
3.5 Artefactos – Scrum	34
3.5.1 Product Backlog	34
3.5.2 Product backlog items	35
3.5.3 User story	35
3.5.4 Estimación	36
3.5.5 Planning poker o Scrum poker	36



3.5.5.1	Cartas basadas en la serie de Fibonacci	37
3.5.5.2	Cartas con talla de ropa	38
3.5.6	Orden de los PBIs	38
3.5.7	Tablero Scrum.....	39
3.6	Ceremonias o reuniones	39
3.6.1	Daily meeting (reunión diaria)	39
3.6.2	Sprint Planning (planificación del sprint)	40
3.6.3	Spring Review (revision de sprint).....	40
3.6.4	Sprint Retrospective (retrospectiva del sprint).....	40
3.6.5	Refinement (refinamiento).....	40
3.7	Ejemplo de proyecto con metodología SCRUM	41



1. Guías de estilo

Cuando nos enfrentamos a un nuevo proyecto Web debemos tener muy claro que **es tan importante el resultado final, visible para el usuario, como todo el trabajo previo de diseño, y toda la documentación** que habrá que realizar durante el proceso.



Así como en nuestro día a día existen normas de circulación, protocolos de actuación, etc, también en el diseño web está lo que se denominan **guías de estilo, o brand guideline** (en inglés).

Una **guía de estilo** es un documento que **establece las pautas** para el diseño y la presentación de contenido en una marca o proyecto. Su objetivo es **garantizar la coherencia y la uniformidad** en el uso de elementos visuales, como colores, tipografías, imágenes y otros elementos gráficos.

El **objetivo** principal de las guías de estilo es **establecer un estilo uniforme y consistente** para todo el sitio web. De esta forma se consigue un estilo homogéneo que favorece la navegación del usuario por el sitio. Desde el punto de vista del programador y diseñador, estas guías favorecen el desarrollo del sitio web.

Las guías de estilo recogen los **criterios y normas** que deben seguir los desarrolladores de un sitio web para que tenga una **apariencia uniforme y atractiva** para el usuario.

La guía de estilo está **dirigida a las personas encargadas del diseño y programación** de la interfaz web, con la misión de lograr una mejor toma de decisiones, tanto en el proceso de diseño como en la fase posterior de mantenimiento del sitio web.

Una guía de estilo se hace imprescindible cuando el sitio web es desarrollado y/o mantenido por un grupo de personas. Este aspecto es muy importante ya que el mantenimiento puede ser llevado cada vez por una persona.

En la guía de estilo se debe recoger aspectos como la introducción en la interfaz de fotografías e imágenes, los botones, la gama de colores, los tipos, tamaño y estilos de letra, el uso de logotipos, etc.



Guía de estilo de Google (<https://design.google/library/>).

En los siguientes enlaces podrás ver algunas guías de estilo:

- [Universidad Politécnica de Valencia.](#)
- [Museos de la región de Murcia.](#)
- [Universidad de Córdoba.](#)
- [Column Five.](#)
- [Universidad de Málaga.](#)
- [Apple.](#)
- [Youtube.](#)
- En el siguiente tablero de Pinterest podemos ver más ejemplos de guías de estilo:
<https://www.pinterest.es/pin/755056693761701615/>

No hay una regla exacta para la redacción de una guía de estilo. Aun así, los contenidos mínimos que deberá incluir una guía de estilo son los siguientes:

- Información general sobre **la marca, su misión, visión y valores.**
- Información sobre el **logo**: cual es el logo, sus variaciones y dónde se aplica.
- Información sobre los **colores**: cual es nuestra paleta de colores y dónde se aplican.
- Información sobre la **tipografía** contenida: estilos de tipografía y dónde se aplican.
- Información sobre las **imágenes e iconografía**, y dónde se aplican.
- Información sobre los elementos **multimedia e interactivos**, y dónde se aplican.

A continuación, se muestran algunos **ejemplos** adicionales de guías de estilo.



01. GRID

Landing Page - 1170px
Portal - 1100px

03. TYPOGRAPHY

Headers:
75 Bold, 72px
75 Bold, 32px
75 Bold, 20-24px

Body:
65 Medium, 12px - 16px

05. BUTTONS

Static

Static







- gradient 8%

Hover







Hover















- gradient 16%











06. AVATARS




  50px   35px  25px 



07. ICONS











Landing Page
     

Portal
             

Artists/Albums/Activity
         

Rank
  

Voting
 76  13

Social Media
(Portal) (Landing Page - footer)
         

02. COLOR PALETTE

#1ba9ed	#36bc74	#2d2926
#888d99	#c2c5cb	#eeeeee
#f35800	#18303d	#dbd0de
#8f737b	#fbc3e	

04. FORMS

Empty

Selected

Typing

Select

Choose

Error

Success

Alternative version of alert messages

Error

Success

08. UI ELEMENTS

Calendar

June 2014

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Slider/Tooltips

50 PLN

500 PLN

Paginator

1 2 3 ... 12 Next

Tag - Tooltip

New album, Jonny Bravo

Solidarity of Arts, Gdansk

Drop Down Menu

Activity

Friends

Messages 3

Edit 20

My profile

Widgets

Favorite albums

Log out

Read more

Move/Hide

Move module

Hide module

09. ALERT/ERROR MESSAGES

Info Message

Warning Message

Error Message

Success Message

10. LOGIN

Log in with Facebook


Log in with Google+

or

Forgotten Password? Reset Password

Log in



 **Postmen** | Styleguide

FONT
Open Sans
Aa Aa Aa Aa
Light Regular Semi Bold Bold
H1 - Open Sans Light (30 px)
H2 - Open Sans Regular (20 px)
H3 - Open Sans Bold (16 px)
Open Sans Regular (14 px)
Open Sans Bold (12 px)
Open Sans Regular (12 px)

COLOR PALETTE
Primary
Aa Aa Aa Aa Aa Aa Aa Aa
#283B53 40, 59, 83 #219AFF 33, 154, 255 #00C3F0 0, 195, 240 #DDE6EF 221, 230, 239 #E7EDF1 231, 237, 241 #333333 51, 51, 51 #999999 153, 153, 153 #DDDDDD 221, 221, 221 #FCFCFC 252, 252, 252
Secondary
Aa Aa Aa Aa Aa Aa
#F8C100 248, 193, 0 #EE001D 238, 0, 29 #FFCFCA 255, 207, 202 #00B258 0, 178, 88 #D4F3E3 212, 243, 227 #F67800 246, 120, 0 #FFE3C9 255, 227, 201

BUTTON
Normal Hover Disable
Normal Hover Disable
Text Hover
Normal Hover Normal

STATUS
Connected Fail Paid Fulfilled

FORM
Form
Title
Select Box
Title
Error
Title Option
Title* Option
Title* Error
Search
Search
Search something
Dropdown options (Shipper Account)
Option
Production
Hong Kong Warehouse
Hong Kong Post
Text
Hong Kong Post
DHL
Aramex
Date
2017-02-13 to 2017-02-13
January 2017
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

NOTIFICATION MESSAGE
Success, refresh to dismiss message
Fail, refresh to dismiss message
Success with button, can't dismiss
Timeout/Verify with button, can't dismiss
Success with cross, can dismiss message
Fail with cross, can dismiss message



UI KIT

PRODUCTPRICINGSUPPORTSIGN IN

TYPOGRAPHY

Headline Lato Light 36px - H1

Title Lato Light 24px - H2

Subheader Lato Regular 18px - H3

Body2/Menu Lato Semibold 16px

Body1 Lato Regular 14px

Caption Lato Regular 12px

SYSTEM ICONS - NAVIGATION

COLORS

#29B6F6 #0288D1
PRIMARY COLOR

#FFC107 #FFA000
ACCENT COLOR

#FF5252 #F44336
BUTTONS COLOR

#00BFA5 #03B098
ACCENT COLOR

#727272 #212121
PRIMARY TEXT

#FAF8FC #F5F7FA
BACKGROUND

#B3E5FC #B7BAC9
BUTTONS COLOR

#DDE6E9 #CFD8E2
DIVIDER

#FFB74D #EDF1F2

CHECKBOX & RADIO BUTTON

☐ Unchecked option

☐ Hover option

☒ Checked

☐ Disabled item

☐ Static

☐ Hover

☒ Checked

☐ Disabled

size: 20x20px

PAGINATION

SWITCH

ALERT STYLES

COMPONENTS

BUTTONS

Static

Hover

Pressed

Disabled

color: #ff5252; #f44336; #f44336; #ff8a65; + inner shadows

padding: 8px 16px; height: 40px; border-radius: 3px; font-size: 15px; font: LatoSemibold;

border-color: #e53935; border-size: 15px; icon-size: 13px;

Labeled:

Static

Hover

Pressed

padding: 8px 16px; border-radius: 3px; font-size: 15px; icon-size: 13px;

padding: 8px 16px; border-radius: 3px; font-size: 15px; icon-size: 13px;

padding: 8px 16px; border-radius: 3px; font-size: 15px; icon-size: 13px;

INPUT

Normal

Active

Alert

height: 40px; padding: 8px 15px; font-size: 16px; font-color: #7bac9; border: 1px solid #dde6e9; border-radius: 4px;

border: 1px solid #2986f6; font-color: #212121;

border: 1px solid #ff5252; font-color: #212121;

SELECT

Normal

Hover

height: 40px; padding: 8px 15px; font-size: 16px; font-color: #212121; border: 1px solid #dde6e9; border-radius: 4px;

font-color: #212121; border: 1px solid #2986f6;

DROPDOWN

Background-color #f5f7fa - header

Normal

Hover

Active

Group

Group

Group

height: 35px; padding: 6px 16px; font-size: 15px; font-color: #727272; border: 1px solid #dde6e9; border-radius: 4px;

Norvax, LLC

Norvax, LLC

Norvax, LLC

Norvax, LLC

Norvax, LLC

color: #dde6e9; font-color: #212121;

Background-color #ffffff

Group

Group

Group

Group

Group

Group

height: 35px; padding: 6px 16px; font-size: 15px; font-color: #212121; border: 1px solid #dde6e9; border-radius: 4px;

Norvax, LLC

Norvax, LLC

Norvax, LLC

Norvax, LLC

Norvax, LLC

color: #dde6e9; font-color: #212121;

FILE INPUT

Normal

Hover

Choose file

Choose file

height: 40px; padding: 8px 16px; font-size: 15px; font-color: #212121; border: 1px solid #dde6e9; border-radius: 4px; background-color: #edf1f2;

color: #ffffff;

color: #616161;

SELECT PERIOD

All

180 days

60 days

30 days

ELEMENTS FOR CHART

\$ 12.700

\$ 7.200

\$ 5.500

Product

Pricing

Support

Privacy

Terms of Use

Contact

Copyright © 2016 HealthJoy

Engage your employees in your Group

Contact Us



1.1 ¿Cómo puedo crear una guía de estilo para un sitio web?

Veamos los **pasos** a seguir para crear una guía de estilo para un sitio web pueden ser los siguientes:

- 1) **Define la identidad de tu marca.** Antes de comenzar a crear una guía de estilo, es importante tener una idea clara de la identidad de tu marca, el logo, y los valores que quieres transmitir a través del sitio web. Esto te ayudará a tomar decisiones sobre el estilo visual y el tono de voz que debes utilizar.
- 2) **Elige una paleta de colores.** Es importante elegir una paleta de colores que refleje la identidad de tu marca y sea coherente con el resto de tus elementos visuales. Asegúrate de incluir información sobre los colores principales y secundarios, así como sobre cómo deben utilizarse en diferentes elementos del sitio.
- 3) **Selecciona las tipografías.** Elige fuentes que sean legibles y estén en línea con la identidad de tu marca. Incluye información sobre las fuentes principales y secundarias, así como sobre cómo deben utilizarse en diferentes elementos del sitio, como títulos, subtítulos y texto del cuerpo.
- 4) **Establece pautas para el uso de imágenes.** Es importante establecer pautas claras sobre cómo deben usarse las imágenes. Incluye información sobre el tipo de imágenes que deben utilizarse, cómo deben editarse y cómo deben presentarse en el sitio.
- 5) **Define el tono y el estilo de escritura.** Establece pautas claras sobre el tono y el estilo que deben seguir los textos, incluyendo aspectos como el uso de la voz activa o pasiva, el nivel de formalidad y el uso de jerga o tecnicismos.
- 6) **Selecciona los iconos.** Los iconos son elementos gráficos que pueden ayudar a mejorar la navegación y la comprensión del contenido. Al crear tu guía de estilo, es importante establecer pautas claras sobre el uso de iconos, incluyendo información sobre el estilo y el tamaño de los iconos, así como sobre cómo deben utilizarse en diferentes partes del sitio.
- 7) **Diseña botones, formularios y otros componentes.** Es importante establecer pautas claras sobre el diseño y el uso de estos componentes, incluyendo información sobre el tamaño, el color, la forma y el estilo de los botones y formularios, así como



sobre cómo deben utilizarse en diferentes partes del sitio. Asegúrate de diseñar componentes que sean coherentes con la identidad visual de tu marca y que sean fáciles de usar para tus usuarios.

Una vez que hayas definido todos estos elementos, puedes compilarlos en un documento para crear tu guía de estilo. Asegúrate de presentar la **información de manera clara y concisa**, utilizando ejemplos visuales para ilustrar tus pautas. Además, ten en cuenta que, una guía de estilo **es un documento vivo** que puede evolucionar con el tiempo a medida que cambian las necesidades y preferencias de tu marca y tu audiencia.

1.2 El checklist de una buena guía de estilo.

El esfuerzo de crear la guía habrá valido la pena si consigues que sea de verdad útil, si las personas la consultan con frecuencia y encuentran en ella lo que necesitan. Para ello es importante cumplir con el siguiente checklist.

- ✓ **Fácilmente accesible** para cualquier persona tanto dentro como fuera de la organización.
- ✓ **Clara y fácil de comprender** incluso para un novato que desconozca todo sobre la marca.
- ✓ **Exhaustiva sí, pero breve.** De lo contrario será poco manejable y menos probable que las personas la utilicen.
- ✓ Todo lo **visual** que sea posible, para hacerla más atractiva y facilitar su comprensión.
- ✓ **¡Con estilo!** Tiene que estar impregnada de la personalidad de tu marca. ¿No está hablando de estilo? Pues que sirva de ejemplo.
- ✓ **Contiene reglas, pero es** lo bastante **flexible** para no obstaculizar el trabajo creativo del equipo.
- ✓ **¡Es conocida!** Todas las personas del equipo la conocen y saben dónde está.



- ✓ Siempre al día: para que sea verdaderamente efectiva tiene que estar **siempre actualizada**. Ayuda mucho agendar revisiones periódicas para evitar que se quede desactualizada.



2. Metodología de trabajo (tradicional)

En el **proceso de generación de un sitio web y de su interfaz** correspondiente, **es importante seguir una metodología de trabajo**, ya que trabajaremos de forma más ordenada y optimizada en tiempo, y daremos mejor impresión a nuestro cliente, lo que **se reflejará en la calidad del producto final** obtenido.

Existen diferentes metodologías de trabajo. La metodología de trabajo que se propone aquí es una de ellas, y es una propuesta basada en la experiencia profesional. Ésta está formada por las siguientes fases:

- 1) **Estrategia y planificación.**
- 2) **Análisis.**
- 3) **Diseño.**
- 4) **Desarrollo o construcción.**
- 5) **Pruebas y despliegue.**
- 6) **Documentación.**
- 7) **Mantenimiento.**

A continuación, se desarrolla cada una de las fases anteriormente mencionadas.

2.1 Estrategia y planificación

En esta fase se **recaba toda la información necesaria** para llevar a cabo el proyecto. Es una fase que está marcada por la **reunión inicial con nuestro cliente**, en donde tendremos que tomar todo el tiempo que sea necesario. Por tanto, en esta fase se debe:



A. DEFINIR EL PROYECTO.

A través de una **reunión** con nuestro **cliente**, tendremos que entender sobre qué trata el proyecto y definir una estrategia. En este tipo de reuniones se debe recabar toda la información que sea necesaria y **no dejar nada “al azar”**.

B. DEFINIR LA AUDIENCIA A LA QUE VA DIRIGIDA LA PÁGINA.

Esto determinará todo su diseño y contenido. Tenemos, por tanto, que definir los siguientes aspectos:

- El **género** de nuestra audiencia, es decir, si es masculino, femenino, o mixto.



- El rango de **edad**. Por ejemplo, si la página es para niños, los colores tienen que ser muy llamativos y variados para transmitir juego, creatividad, diversión... Por el contrario, si es para adultos, utilizaremos tonos claros para transmitir más seriedad, profesionalidad, confianza... Si va destinada a adolescentes, quizás debemos incluir fotografías con gente joven y atractiva...
- La **nacionalidad**. Esto es debido a que los gustos varían dependiendo de la procedencia de la audiencia.
- **Desde que dispositivos navega la audiencia**. Esto es importante a la hora de entender cuanta información acumular, de qué forma se estructura la página, y para optimizar la página en función del dispositivo que se utiliza.
- Que **navegador** utilizan. Esto nos orienta hacia qué tipo de especificaciones técnicas tenemos que optimizar más.
- Su **comportamiento**. Esto es, el **tiempo de permanencia** de la audiencia en la página, el número de páginas visitadas por sesión, si la audiencia se va rápido o ve diferentes secciones, etc. Entender estos aspectos nos hará ver en que tenemos que mejorar la página, que deficiencias tiene la audiencia, etc. Por ejemplo, si la audiencia dedica mucho tiempo, se puede añadir más contenido a la página; si dedica poco tiempo, entonces debemos poner lo importante a primera vista en la parte superior de la página.
- Alguna **otra información adicional** que consideremos relevante.

C. ESPECIFICAR LOS OBJETIVOS.

Se debe definir qué objetivos se buscan lograr a través de la página, además de darles prioridad. Tal y como se ha visto anteriormente, **lo ideal es definir un solo objetivo principal y dos o tres objetivos secundarios** (a pocos clics).

Por ejemplo, si el objetivo principal es “*vender*”, entonces podemos crear un botón de comprar muy llamativo, que destaque, que su acceso sea rápido, y que rellenar datos sea muy sencillo.

Otro ejemplo de objetivo principal sería “*reunir un gran volumen de emails en la base de datos para después enviar publicidad y conseguir así clientes en potencia*”. Para ello podemos superponer una ventana que tape el contenido de la página web, que contenga una caja de texto para que la audiencia ponga su dirección de correo electrónico, con la promesa de enviar un regalo o algo de forma gratuita.



Además, tal y como se comentó anteriormente, el objetivo principal tiene que ser ineludible **siempre evitando que sea forma abusiva y/o agresiva**.

D. DELIMITAR LAS ESPECIFICACIONES TÉCNICAS.

Tenemos que delimitar las especificaciones técnicas que debe tener la página web, tales como la **compatibilidad con navegadores y versiones**. Es decir, tenemos que reunir las características que va a tener nuestra página para el diferente soporte de visualización que vaya a haber entre dispositivos, navegadores, versiones, etc.

Tenemos que tener en cuenta que, normalmente, puede haber algún aspecto de la página web que no se vea bien desde un teléfono muy antiguo con un navegador web muy antiguo. Por ejemplo, si utilizamos html5 en nuestra web, va a haber etiquetas que un navegador antiguo no sabrá interpretar.

Un pequeño truco, si es posible, sería **observar el dispositivo sobre el que navega nuestro cliente y el navegador que utilizan sus empleados**.



E. DETALLAR EL INVENTARIO DEL CONTENIDO.

Esta es una tarea de nuestro cliente, el cual deberá acudir a la reunión con todo lo que quiere que su página web contenga.

Por tanto, el cliente tiene que detallar todo el contenido que quiere introducir en su web y en qué sección tiene que aparecer.

Es en este punto en donde **se definen las secciones** que va a tener el menú de la página web.

Consejo: antes de acudir a la reunión con el cliente, es bueno **analizar o examinar páginas similares**, no para copiar, sino para tener una idea de los contenidos que la competencia ya tiene.

F. REUNIR LOS RECURSOS DISPONIBLES.

Es importante reunir los **recursos disponibles** desde el inicio y que han sido **proporcionados por el cliente** como pueden ser **imágenes**, el **logotipo**, **bases de datos**, **tipografía**, **color corporativo**, etc.

También hay que analizar aquí, si los recursos proporcionados por el cliente ya están **“pulidos”** para presentación final de la página.

Otro aspecto a tener en cuenta es el **estado de la base de datos**. En el caso de que ya exista, debemos ver el trabajo que nos va a suponer convertirla en algo escalable, modular, sólida y estable.

G. ESTABLECER EL TIMELINE.

Una vez que ya hemos entendido la totalidad del trabajo que hay que realizar, se debe dar al cliente las fechas en las que presentaremos nuestro trabajo. **No dar fechas crea inseguridad en el cliente.**

Normalmente se suelen dar dos plazos en tiempo (como mínimo):

- 1) La primera fecha será aquella en la que estará lista la **primera versión inicial** de la página web, y es cuando, además, realizaremos el testeo con el cliente. Aquí pueden surgir cambios derivados de dicho testeo.



- 2) La segunda fecha será para la **publicación de la versión definitiva** de la página web.

H. PRESUPUESTO.

Establecer la cantidad de la que se dispone para definir el grado del diseño y el coste de servicios adicionales.

La cantidad se calculará **en base al tiempo** que se va a emplear en el desarrollo de la página, si se va a necesitar ayuda extra, etc.

Por ejemplo, si la página web es muy sencilla, se le podría cobrar al cliente un tercio del salario mínimo. Si es algo muy personalizado y complejo, se podría cobrar el doble del salario promedio.

2.2 Análisis

Después de reunir todos los datos que son necesarios para llevar a cabo nuestro proyecto, estableceremos los requisitos que deberá cumplir nuestro sitio web, su sistema de navegación y su funcionalidad.



A. CONCEPTOS INICIALES.

En este primer punto de la fase de planificación, habrá que llevar a cabo un **análisis preliminar** y establecer **conceptos iniciales a “grosso modo”** sin entrar en detalles. Esto es, se van a enumerar de forma muy global los conceptos más básicos de **cómo va a ser la página web, como va a ser la base de datos, como va a ser el diseño, de qué manera se van a cumplir los objetivos, etc.**

Por tanto, en este punto **no se concretan detalles.**

Los conceptos iniciales van a ser los “pilares” de la página web.

B. LLUVIA DE IDEAS (o BRAINSTORMING).

Es conveniente que **anotes toda la “avalancha” de ideas** que vengan a tu mente para el diseño de la página, a fin de cumplir la estrategia acordada. Por ejemplo, cómo



enfocar la página para cumplir los objetivos, posibilidades de como diseñar la página, cómo estructurarla, cómo hacer la arquitectura, que tipo de tipografías se podrían usar, los colores que creemos que nos encajan mejor, etc.

Algunos desarrolladores incluso crean una **base de datos de esas ideas** que les van surgiendo, con el objetivo de no perderlas, ya que existe la posibilidad de poder aplicarlas en proyectos futuros.

¿Cómo se sabe si una idea va a tener éxito o no? Nunca estaremos seguros por completo, aunque siempre podremos examinar el comportamiento de los usuarios en aquellas páginas en las que hayamos implantado dichas ideas.

C. ARQUITECTURA.

En este paso definiremos la **estructura que tendrá la página web**, es decir, cómo se organizarán los contenidos en las diferentes secciones de navegación. Nos centraremos en los siguientes puntos:

- 1) **Como va a ser la navegabilidad** de la página, esto es, cuáles van a ser los elementos definitivos que va a tener el **menú de navegación**, y cuáles van a ser las **opciones principales** y cuáles van a conformar el submenú. Para ello, será útil hacer un **sitemap interno**.
- 2) Como va a ser **la estructura de directorios** que vamos a tener en el hosting, y de qué manera lo vamos a planificar. Si está bien planificado, cuando se realicen enlaces internos (en caso de haberlos), todo va a funcionar bien. Si el árbol de directorios no está planificado y hay que cambiarlo o cambiar el nombre de las carpetas, los enlaces ya creados dejarán de funcionar y tendremos que modificarlos uno a uno.
- 3) La **arquitectura de la base de datos**. Si no se estructura bien y se hace “en bruto”, cuando tengamos que hacer cambios posteriores, puede ser que todo lo que teníamos hecho deje de funcionar, por ejemplo, consultas, procedimientos, triggers..., y tendremos que volver a acoplarlo a los cambios que ya se han hecho.



Los **nombres de las tablas deben ser intuitivas**, y que permita que las conexiones relacionales sean escalables y modulares.

- 4) La **profundidad de página o de clic** para que la página cumpla con el objetivo principal y secundarios.

La profundidad de clic es el número de clics que tiene que hacer un usuario desde que entra a la página principal hasta que entra al punto que estamos analizando.

Se aconseja que la **profundidad de clic para un objetivo principal sea uno**, o dos como máximo.

Por ejemplo, si el objetivo principal de una página web es “vender”, y queremos saber la profundidad de click debemos contar el número de clics que el usuario hace desde la página principal hasta llegar al formulario para completar sus datos de compra.

D. LAYOUTS.

Una vez tengamos la arquitectura definida, pasamos a especificar **qué posición** (absoluta o relativa) **va a ocupar cada elemento** dentro de la página. Por tanto, hay que diseñar como se presentarán los elementos de cada página **en distintas resoluciones de pantalla** (desktop, tablet y smartphone). Lo más habitual es diseñar los layouts para **desktop y smartphone**.

Una página web tiene que ser “**responsive**” absoluto, y por **tanto nunca puede aparecer la barra de scroll horizontal**, ya que tiene que adaptarse siempre a cualquier tamaño de pantalla. Para ello, **nunca se tienen que quitar los contenidos que cumplan con los objetivos** que se marcaron de la página principal.

E. USABILIDAD.

Tenemos que diseñar la página web para que sea **intuitiva, fácil de usar y dirigir la atención** a los elementos que cumplan el objetivo principal del sitio web.

Las partes importantes que los usuarios buscan, tienen que ser muy accesibles e intuitivas.



La usabilidad es muy importante, ya que, si los usuarios ven muy sencilla nuestra página web, éstos volverán y compartirán, los tiempos de permanencia serán más grandes, los porcentajes de interacción serán más elevados, etc.

En algunos casos puntuales, tendremos que desarrollar **una mala usabilidad en alguna parte de la página web**, ya que esto puede ser provechoso a nivel de estadísticas de rendimiento de eficacia empresarial. Un ejemplo de este caso, sería desarrollar una mala usabilidad en el botón de “dar de baja”, “borrar suscripción” o “cancelar suscripción”, o cualquier tipo de opción para eliminar usuarios.

F. SKETCHING Y WIREFRAMING.

Antes de escribir el código, es bueno hacer un diseño gráfico del aspecto final que pretendemos que tenga nuestra web. Para ello, podemos hacer un **sketch** en papel, y posteriormente un **wireframe**. Estos son producto de los **apartados D y E**, y se pueden llevar a cabo con las herramientas que ya hemos visto.

2.3 Diseño

En esta fase haremos el diseño completo y final de nuestra página web, teniendo como **resultado final**, varios **mockups o prototipos** alternativos que presentaremos al cliente.



A. PALETA DE COLORES.

Ahora elegimos una gama de colores con los que “pintar” toda nuestra web.

En el diseño de interfaces gráficas, la selección adecuada de la carta de colores es muy importante, puesto que ello puede condicionar la experiencia de navegación del usuario, determinándola por completo.

El uso de una escala coherente facilita la navegación, por lo que debemos ser coherentes y usar siempre un color para lo mismo, así como transmitir armonía.



Pero, **¿qué colores utilizar?** Para ello debemos tener en cuenta todos los aspectos que hemos visto al principio de la UD1.

B. LOGOTIPO.

Diseñamos un logotipo que refleje las características del cliente. Normalmente esta tarea se suele delegar en un diseñador gráfico.

Lo ideal es hacer dos o tres opciones de logotipo para presentar al cliente, y él pueda elegir la que más le guste.

C. AREAS VISUALES.

Diseñamos el **aspecto visual y final de las áreas más globales** de nuestra página web, y vemos cómo combinarlas entre sí. **De aquí en adelante empezamos a trabajar con nuestro mockup o prototipo**, tanto para desktops, smartphones y/o tablets.

En el proceso de diseño, tenemos que **empezar por las partes más genéricas** de la página, **para posteriormente concretar las partes o detalles** más pequeños, ya que si se hace al contrario, es muy posible que ante un mínimo cambio haya áreas que se descoloquen o choquen unas con otras, (sobre todo cuando se hace un responsive design).

D. ELEMENTOS VISUALES.

Una vez diseñadas las áreas genéricas, **se define el aspecto final de los elementos más pequeños o concretos**. Este es un trabajo **más tedioso** en el que hay que tener en cuenta diversos aspectos como:

- **Los botones**. Son el elemento principal de interacción en una página. Es importante que sean **llamativos** para que se cumplan las acciones. También es importante que el botón dé una **buena experiencia al usuario** mediante **padding**s lo suficientemente grandes que permitan que la persona que está navegando en la página no tenga que hacer “puntería” para darle justo en la parte del botón donde se acciona.



- **El formulario o formularios.** Hay que hacerlo de un **modo atractivo**, es decir, que sea **agradable rellenarlo**, que cada caja tenga un borde, que las letras que escriba el usuario sean grandes y que tengan un color bonito... En definitiva, que al usuario le guste escribirlo.
- **Los enlaces.** Hay que ver que estilo le ponemos a los enlaces, esto es, qué color le ponemos, si los queremos subrayados, etc.
- **Los iconos.** Tenemos que tener en cuenta los iconos que se van a poner en cada parte.

Es deseable que se mantenga una **apariciencia común entre todos los iconos** y entre todas las páginas y secciones del sitio web. Al utilizar estos iconos se evita el uso excesivo de textos. Una buena elección de estos iconos es muy importante, puesto que, si un usuario no es capaz de determinar su significado, no se habrá conseguido el propósito de ahorrarle tiempo en la visualización de la página.

- **Las imágenes.** Hay que definir cómo van a ser las imágenes. No todas las fotografías tienen criterios fotográficos buenos, así que podemos decir que no todas van a producir un buen diseño.

Las fotografías que se pongan en nuestra página web van a ser un factor clave en el resultado visual de nuestra página web. Por tanto, para obtener un buen resultado, se deberán incluir unas imágenes bien estudiadas y que cumplan con las reglas, ángulos y planos que ya hemos estudiado.

E. TIPOGRAFÍA.

Elige o crea una **tipografía que corresponda con el público, servicio o imagen** que el cliente desea proyectar.

El uso de fuentes familiares mejora, en gran medida, la calidad de la lectura. Es muy importante elegir un estilo funcional, que permita su visualización sin problema en



todos los navegadores. A la hora de elegir tipografía deben tenerse en cuenta la fuente, el estilo, el tamaño y el color, además de seguir las **recomendaciones** que hemos visto.

F. ANIMACIONES.

Decide si sería bueno añadir animaciones a la página web. En tal caso, se deberá definir como serán, su impacto en la experiencia de usuario, y el efecto que se desea lograr.

Las animaciones **nunca tienen que tener relación o incidencia sobre el contenido** de la página. Es decir, el contenido tiene que estar accesible por el usuario, sin incomodidades y sin juegos. Las animaciones **no pueden ocultar nunca el contenido**.

G. TEST DE USABILIDAD, o de UX.

Antes de empezar a desarrollar código debemos hacer un **test de usabilidad, mostrando el diseño a usuarios de pruebas**. Estos deberán indicar qué es lo que les llama la atención, si la interfaz es intuitiva, si es sencilla, etc.

Los usuarios de prueba deberán ser personas lo **más próximas posible al perfil de usuario real** que va a tener la página web y, además, cuantos más usuarios de prueba tengamos, mejor.

Para llevar a cabo este test, usando el mockup, deberemos imprimir la página diseñada en papel, (tanto la página principal como las secundarias), y por otro lado imprimiremos un formulario donde los usuarios de pruebas responderán a las preguntas formuladas. También se podría hacer con un prototipo.

Algunos ejemplos de dichas preguntas podrían ser:

- *Cuando ves el diseño, ¿qué elementos te han llamado más la atención?*
- *¿Encontró fácil la opción de compartir? ¿Se sintieron impulsados a compartir?*
- *¿Le gusta el diseño?*
- *¿Le resulta fácil de entender?* Para saber que el usuario de prueba está respondiendo correctamente, esta pregunta irá acompañada de una pequeña acción que tenga que realizar dentro de la página web.



- *Solicitar que todos los usuarios de prueba realicen alguna acción específica relacionada con los objetivos principales de la página web, y que no paren hasta conseguirlo.* Nosotros como desarrolladores debemos cronometrar el tiempo que tardan en hacerlo. Si los usuarios de prueba se enfadan o desesperan, debemos anotarlo, ya que esta sería la reacción real de los usuarios reales, los cuales cerrarán la ventana de nuestra página web.

Además, será bueno que los usuarios de prueba cubran un formulario anónimo con sus datos y saber así su perfil. Estos serán archivados.

Una vez hecho el test, se corregirán los errores encontrados, y **no será necesario realizar otro test.**

H. COMPATIBILIDAD.

Un factor que se debe examinar, es la **compatibilidad** de nuestro diseño web, ya que quizás, no todo lo que se haya diseñado sea posible llevarlo a cabo **en cualquier dispositivo y/o navegador web.**

Por tanto, habrá que **evaluar**, y según el presupuesto que se tenga o se haya presentado al cliente, **aquellas partes del diseño que puedan ser susceptibles a incompatibilidades**, como pueden ser las animaciones, las cuales puedan requerir de lenguajes de programación o de códigos que no sean funcionales en todas las versiones de todos los navegadores.

I. PRESENTACIÓN DE MOCKUPS o PROTOTIPOS.

Ahora que ya tenemos una versión final de lo que será la página web en diseño gráfico, es un buen momento para **presentárselo al cliente.** Esta reunión se llevaría a cabo **en el primer plazo de tiempo** que habíamos fijado con él.

Es mejor hacer la primera presentación con los **mockups o prototipos** que con la **página en código**, ya que cualquier cambio que haga el cliente, va a suponer más esfuerzo; o si hay algo que no es de su agrado, ya no será un cambio de código el que habrá que hacer, sino que puede ser un cambio de estructura de diseño.



Por tanto, en este punto, se presentarán al **cliente, dos o tres prototipos** del aspecto final de la página web, para que éste elija el que más le convenza. En dichos prototipos, la parte principal será igual (ya que sabemos que funciona debido al test UX que hemos hecho anteriormente), y las partes menos transcendentales serán distintas, por ejemplo, las tonalidades, texturas, etc.

2.4 Desarrollo o construcción

Una vez que ya sabemos el diseño final que tendrá la página web, entonces pasaremos a su desarrollo.



A. ESTÁNDARES WEB.

Debemos elegir con qué lenguajes y estándares desarrollar nuestra página web.

- Para el frontend, que es lo que nos ocupa, actualmente se utiliza **HTML5, CSS3 y Javascript**.
- Para el backend existen más opciones como **PHP, Python, Ruby**, etc. Debemos elegir aquel del que tengamos más dominio y tenga una gran comunidad detrás. Aun así, el estándar que elijamos tiene que cumplir tres **objetivos**:
 - Que permita una **buena usabilidad**.
 - Que permita una **buena interoperabilidad** (conexión entre todos los elementos como las bases de datos, otras páginas, etc).
 - Que proporcione **robustez**, esto es, la página tiene que ser sólida y **no dar errores** constantemente.

B. FRAMEWORKS.

Un **framework** es una librería o conjunto de librerías que nos permiten trabajar de una forma más rápida, ya que mucho del código interno, que tendríamos que haber escrito con el lenguaje directamente, ya está desarrollado. Además, también tiene bastantes opciones añadidas que quizás directamente con el código sería muy tedioso y muy extenso.



Por tanto, en este paso, debemos **elegir el framework y/o CMS** que se vaya a emplear para el desarrollo.

C. BASE DE DATOS.

Diseña la **estructura** de la base de datos de forma que permite un crecimiento **escalable y modular**.

D. FUNCIONALIDAD.

Antes de definir el estilo de la página web, debemos **primero implementar las operaciones** que tiene que realizar. Es decir, primero tiene que ser “antiestética” pero eficaz, y sin contenido.

El objetivo es asegurarnos de que el comportamiento de la página web corresponda con lo que realmente tiene que hacer, probando así los layouts y el responsive design. Por tanto, **empezaremos con HTML, creando primero las secciones más globales**, para pasar posteriormente a las más pequeñas.

E. CONTENIDO.

Ahora debemos **añadir todo el contenido** que se anotó durante la etapa de estrategia: texto, fotos, animaciones, etc.

F. EJECUCIÓN.

Una vez que se ha introducido a la página web todo el contenido, el volumen de datos y el volumen de carga será totalmente diferente, con lo cual incidirá en el **tiempo de carga** de la página, y el **tiempo de respuesta** de los scripts y las consultas. Además, debemos **probar la ejecución en una pantalla pequeña**, ya que si la página es pesada en un escritorio, lo será más en dicha pantalla pequeña.

También debemos **revisar** lo siguiente:

- Que las **imágenes** estén bien **reducidas** (sin pérdidas) y optimizadas, es decir, están en formatos optimizados para la web como jpeg o png.
- En cuanto a los datos, si la respuesta es más lenta de la que se quiere, habrá que rediseñar los **scripts**, los **procedures** o las **consultas SQL**.



- G. MARCADO.** Revisar que las palabras clave y los títulos de la página web están etiquetados correctamente para asegurar un buen **posicionamiento SEO**.

2.5 Pruebas y Despliegue

Una vez que la página web está desarrollada y finalizada, se **lanzará como pública**, previo test con el cliente.



A. TEST FINAL.

Realizaremos una **última revisión de la web** para comprobar su funcionalidad, la experiencia de usuario, su seguridad, etc.

Por tanto, en este punto nos **reuniremos con el cliente** para **comprobar** con él, el **resultado final** mediante la realización de diversas pruebas. Es en esta reunión cuando se le explicará todo lo relacionado con la página web, con el objetivo de que comprenda cómo funciona.

B. SUBIR ARCHIVOS.

Ahora que ya tenemos la versión definitiva de la página web, **tenemos que publicarla**. Para ello debemos hacer una transferencia de archivos al servidor del cliente o al servidor del **hosting**.

Una vez subidos los archivos al servidor, es recomendable **volver a probar la página web**, para comprobar que todo está funcionando correctamente. Estas pruebas, normalmente, se llevan a cabo de noche para que haya un número mínimo de usuarios conectados y así no afectar a su experiencia de usuario; y si además falla, tendremos tiempo para arreglarla.

Después de subir y publicar la página, es recomendable guardar el estado de la página inicial a **un sistema de control de versiones**, ya que es susceptible de modificaciones. En dicho sistema podremos ir guardando las versiones estables y las versiones beta.



2.6 Documentación

La fase de documentación **deberá llevarse a lo largo de todo el proceso** de planificación, análisis, diseño, desarrollo, despliegue y mantenimiento de una página web.

La **documentación es esencial** en cualquier proyecto de desarrollo web, ya que:

a) Ayuda a los **usuarios/cliente** a entender cómo funciona la página web que hemos desarrollado. Por ello, **después de la fase de despliegue** llevaremos a cabo las siguientes acciones:

- **ENTREGA DE ARCHIVOS.** Proveemos al cliente con todos los archivos que se han empleado en la fase de desarrollo, ya que ha pagado y tiene derecho a ello.
- **ENTRENAMIENTO DEL PERSONAL.** Es recomendable dedicar el tiempo necesario a **enseñar al personal** de la empresa a usar la web correctamente. Además, nos debemos asegurar de que éstos anotan los puntos clave.

En este punto, es recomendable preparar un **documento o dossier** con las instrucciones de al menos, los puntos más importantes del funcionamiento de la página web, y que vengan detallados como hacerlos paso a paso. También sería bueno, hacer uno o varios vídeos con ejemplos prácticos.

b) Ayuda al **propio desarrollador** del proyecto. Al crear una **buena documentación técnica** de la página web, así como una **buena guía de estilo**, el desarrollador será capaz de comprender mejor su funcionamiento y su estilo, lo que ayudará a explicarlo mejor a otros desarrolladores. Asimismo, se aprenderá a trabajar de manera más ordenada.

Otra situación en la que una documentación ayuda al desarrollador encargado del proyecto, es cuando se deja por un buen tiempo de trabajar en cierto código, ya que tiempo después, cuando se regresa al proyecto se pueden **entender o recordar más fácilmente**. Mientras más tiempo haya pasado sin trabajar en ese código o proyecto, más útil será la documentación.



- c) Ayuda a **otros desarrolladores** del mismo equipo de trabajo. Cuando se trabaja en equipo es probable que el código o proyecto no sea propiedad de un solo desarrollador y se tenga que enviar partes del mismo a otros miembros de equipo. Es aquí donde la **documentación técnica y la guía de estilo** entran en juego y cumplen un papel esencial en equipos de desarrolladores pues **sirve como material de referencia**.

Esta documentación, al ser dirigida a desarrolladores y no usuarios finales, debe contener la descripción de los **procedimientos técnicos**. De modo que se puede incluir términos propios del lenguaje y palabras técnicas. El uso de ejemplos sobre el funcionamiento del código también es bienvenido. Adicionalmente a una buena documentación, se pueden incluir comentarios que describan lo que hace el código dentro de su propio archivo.

2.7 Mantenimiento

La etapa de mantenimiento de la página web se lleva a cabo **durante toda la vida útil** de ésta.



A. CORRECCIÓN DE ERRORES.

Después de que la página web entre en funcionamiento, **es muy probable que algún punto del desarrollo tenga errores** y por tanto, debemos **corregirlo**.

Por otro lado, también puede haber **errores en la página web debido al mal uso**, es decir, dichos errores son provocados por el personal de la empresa. En este caso, **no estaremos obligados a dar soporte**, excepto si este tipo de errores están dentro de la garantía, o si el cliente nos ha contratado un servicio de soporte.

B. GARANTIZAR EL FUNCIONAMIENTO.

Dependiendo del tipo de página que se haya creado, hay páginas en las que se permite que los **usuarios** tengan un **grado muy elevado de participación en la creación de contenido** de la página, incluso pudiendo llegar a añadir secciones nuevas. En este caso, es bueno garantizar el funcionamiento de la página web mediante el



lanzamiento de tareas periódicas que garanticen su solidez, revisando así la exactitud de los nuevos datos introducidos. Un ejemplo sería comprobar o hacer una revisión de registros que se han creado con valores nulos.

C. SEGURIDAD.

Se deben implementar **medidas** para la seguridad: actualizar el CMS (en caso de utilizarlo), realizar copias de seguridad, tener un antivirus en el servidor de la web, actualizar el servidor, etc.

Si la seguridad se ve comprometida, no solo dejará de funcionar la página web, si no que los datos se pueden ver comprometidos a través de la base de datos.

D. SISTEMA DE ANALÍTICAS O ANALITYCS.

Saber cómo es el tráfico en nuestra página web, permitirá saber que partes de ella funcionan bien o mal, y nos ayudará también a definir las campañas.

Hay muchos **sistemas de analíticas**. Algunos de ellos son:

- 1) Las analíticas que nos da la empresa de **hosting**.
- 2) El sistema de analíticas que se ha introducido en la propia página web (por ejemplo, **Google Analytics**).
- 3) Las analíticas generadas de aspectos de la página web que no da Google, a **través de la base datos**.

Los datos que quitamos de estas analíticas nos pueden decir si realmente estamos cumpliendo con los objetivos de la página. También nos servirán para saber cuándo lanzar una campaña de un producto nuevo o con cuanta antelación hacer una publicidad de un producto, servicio o evento.

Mensualmente, se pueden analizar los datos y enviar un **informe** con datos como:

- Número total de visitas.
- El tiempo de permanencia medio.
- Porcentaje de usuarios que están volviendo a visitar la página.
- Datos relacionados con los objetivos de la página web.



- Datos relacionados con lo que nos diga el cliente que es relevante.

E. OPTIMIZACIÓN.

El tráfico puede variar o puede ser el cliente quien evolucione. Eso nos obligará a optimizar la web al nuevo perfil que ha surgido. Normalmente este tipo de actualizaciones se dará en el diseño, en funcionalidades o en contenido:

- **Actualizar el diseño.** Con el paso del tiempo, surgirán nuevas tendencias en el diseño web y el cliente quizás quiera estar al día.
- **Actualizar funcionalidades.** La evolución del cliente hará que los servicios que ofrece se amplíen o cambien. Esto se deberá reflejar en la página web.
- **Actualizar contenidos.** Si el sitio web se estanca en el contenido ofrecido, no habrá muchas visitas, y por tanto deberá renovarse periódicamente.

Optimizar también puede implicar cambiar el lenguaje dentro de la página, por ejemplo, cambiar de HTML5 a HTML6, o de CSS3 a CSS4.



3. Anexo I - Metodología SCRUM

La **metodología Scrum** es una **metodología de desarrollo ágil de software** que ayuda a los equipos a **desarrollar productos en periodos cortos**, permitiendo **obtener de forma rápida feedback por parte del cliente**, adaptaciones y una **mejora continuada**.

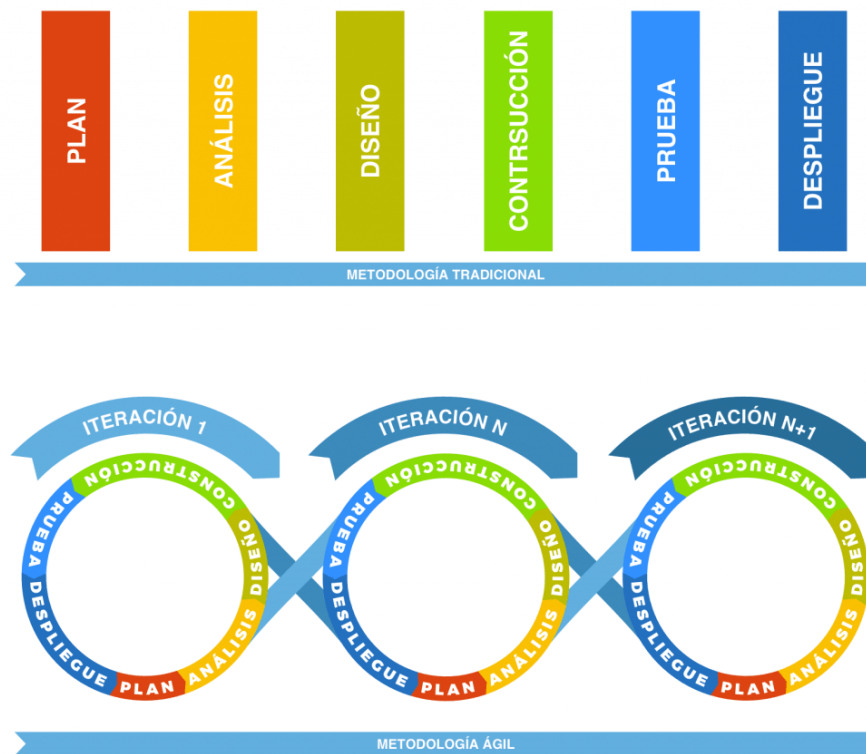
Para comprender a fondo qué es la metodología Scrum vamos a ver primero las características del desarrollo ágil de software y sus diferencias con respecto a la metodología tradicional.

3.1 Desarrollo ágil de software

El **desarrollo ágil de software** o **metodología ágil (Agile Methodology** en inglés) es una metodología de desarrollo de software basada en el **desarrollo iterativo e incremental**, donde los requisitos y soluciones van evolucionando según las necesidades del proyecto. En las metodologías ágiles la **adaptación al cambio** forma parte del proceso natural del proyecto.

Hay muchas **diferencias** entre el desarrollo ágil de software y el método de desarrollo tradicional. Si nos centramos en las diferencias fundamentales cabe destacar lo siguiente:

- En la **metodología tradicional**, se define el proyecto desde el inicio y se crea un plan detallado con **contratos estrictos**.
- Sin embargo, en la **metodología ágil** se comienza definiendo las **funcionalidades básicas** del proyecto garantizando la **entrega continua de software de valor** que se va mejorando de forma continuada en cada iteración. En lugar de contratos estrictos, se mantienen **reuniones constantes con el cliente** y se adapta el proyecto a sus necesidades.



Comparativa entre la metodología ágil y la tradicional.

3.2 Principios de la metodología ágil

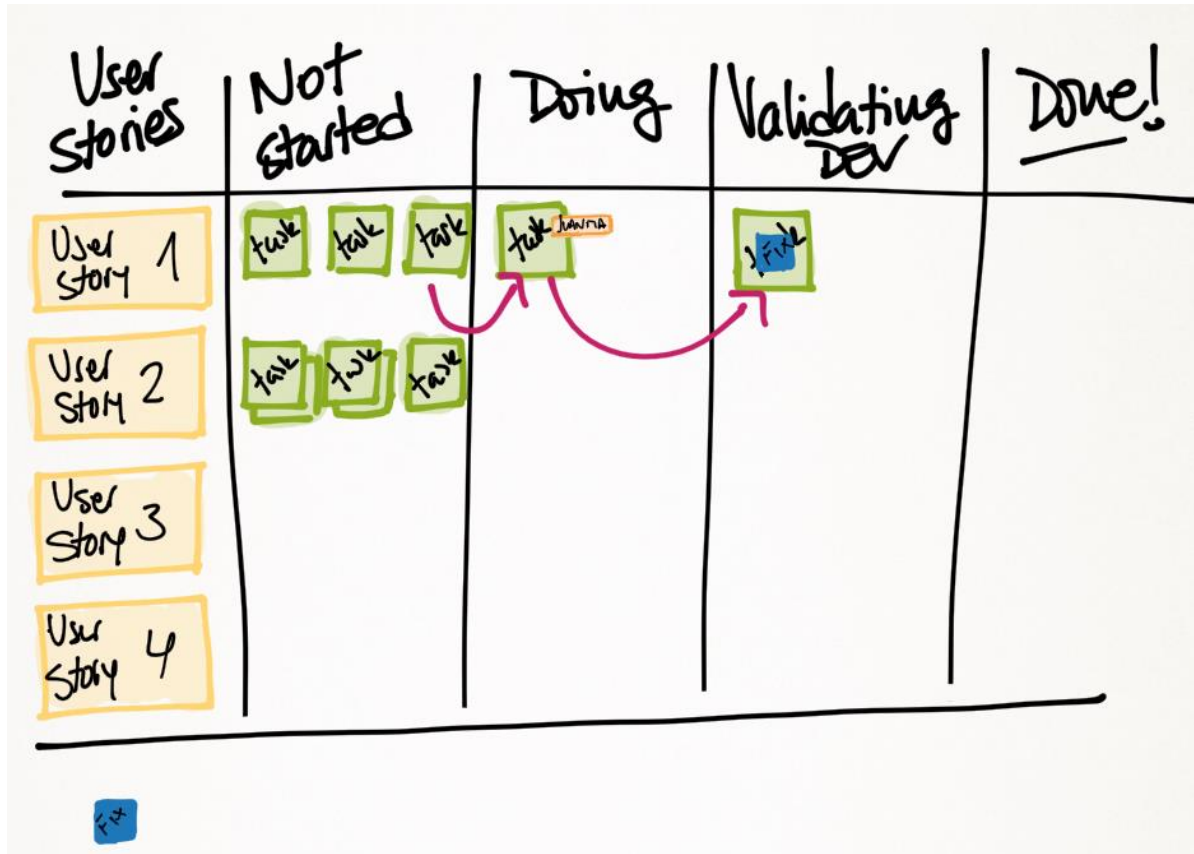
Las metodologías ágiles se basan en una serie de **principios**:

- 1) **Satisfacción del cliente** mediante la entrega temprana y continua de software de valor.
- 2) Gran **aceptación y respuesta al cambio** de los requisitos definidos.
- 3) **Entrega frecuente de software** que funcione, normalmente en periodos de dos semanas hasta dos meses.
- 4) **Trabajo conjunto** entre el **equipo de negocio** y los **desarrolladores**.
- 5) Creación de **equipos motivados** y respaldados.
- 6) **Comunicación** cara a cara.
- 7) **Equipos auto-organizados**.
- 8) **Reflexión continua** de formas para mejorar la efectividad y realizar los ajustes necesarios.



3.3 Sprints

Scrum es una **metodología ágil** que engloba una serie de principios que ayudan a los equipos a **desarrollar productos en periodos cortos (llamados sprints)**, permitiendo obtener de forma rápida feedback, adaptaciones y una mejora continua.



3.4 Roles – Scrum

Los roles que forman parte de este tipo de metodología son los siguientes.

3.4.1 Product Owner

Product Owner o **propietario del producto** trabaja con las partes interesadas, los usuarios finales y el equipo de Scrum para asegurarse de que el producto final cumpla con los requisitos del usuario y se alinee con los objetivos comerciales. En concreto:

- Es el más **cercano al negocio/cliente**.
- **Decide qué desarrollar** y qué no.



- Define buenas historias de usuario.
- Gestiona el **Product Backlog** (cartera de productos) y se asegura que esté claro para todo el equipo.
- Encargado del **retorno de inversión** del proyecto (**ROI**).
- Define el **producto mínimo viable** (MVP). El MVP es un producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro.
- **Valida las entregas** (sprint review).

3.4.2 Scrum Master

El Scrum Master es el **líder del equipo de Scrum**. Está a cargo de establecer la metodología Scrum y mantener a los miembros del equipo enfocados en los principios y las prácticas de Scrum:

- Planifica la implantación de Scrum y se **asegura de que todo el equipo lo entienda y lo aplique** correctamente.
- Ayuda a **resolver los impedimentos** que surjan durante el sprint.
- **Crea un buen ambiente** en el equipo y favorece la auto-organización.
- **Protege al equipo de interferencias externas**.
- **Ayuda al product owner** a comprender la metodología ágil.
- Se asegura de que se celebren las **reuniones** y de que se cumpla el **objetivo en el tiempo establecido**.



3.4.3 Development team

El Development Team o **equipo de desarrollo** se define como un conjunto de profesionales que realizan el trabajo de **desarrollar software “terminado”** (cumple la **Definition of Done**), que se pueda desplegar en producción al final de cada Sprint.

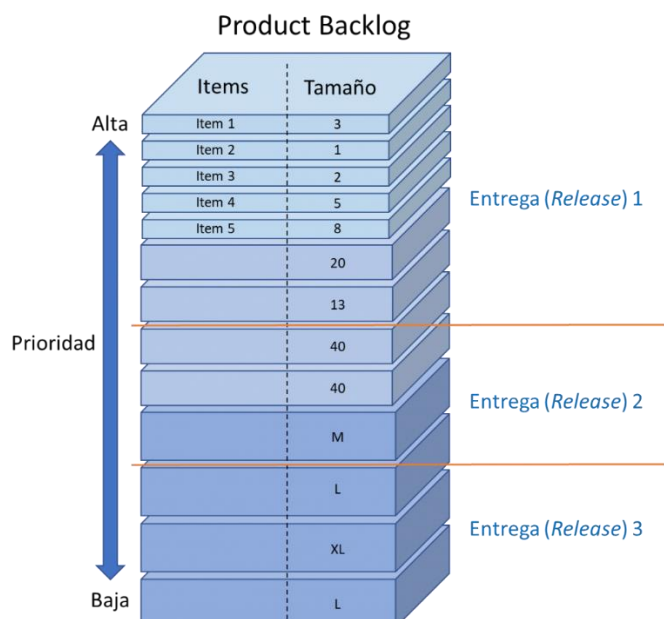
- Compuesto de 5 a 9 componentes (normalmente).
- Auto-organizado y auto-gestionado.
- Decide la cantidad de trabajo a realizar en un sprint (**sprint planning**).
- Estable, motivado y dedicado.
- Se obtienen mejores resultados cuando trabajan en la misma localización física.

3.5 Artefactos – Scrum

Los artefactos son aquellos elementos físicos que se producen como resultado de la aplicación de Scrum. Los tres principales artefactos o herramientas Scrum son: el **Product Backlog**, Sprint Backlog y el Incremento.

3.5.1 Product Backlog

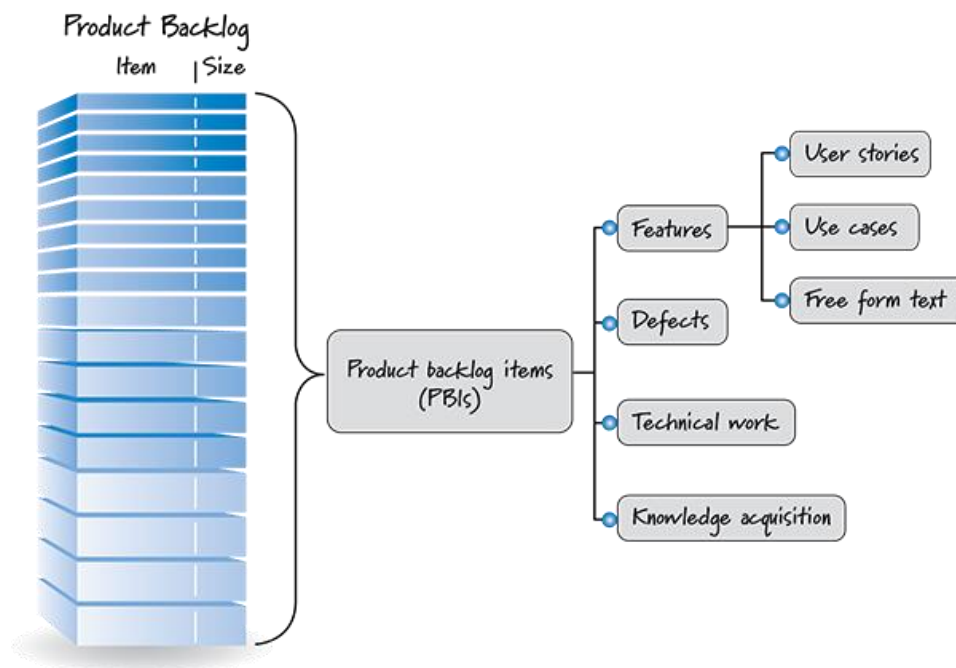
El **product backlog** o **pila del producto** es una **lista ordenada** y priorizada de **funcionalidades** definidas por el cliente.



3.5.2 Product backlog items

Dentro del product backlog definiremos los elementos de la pila de producto o **PBI** (de sus siglas en inglés **Product Backlog Item**).

Los elementos del product backlog suelen contener la **descripción, el orden, la estimación y el valor**. Además, también pueden disponer de otra información relevante como diagramas de uso, prototipos, datos de encuestas o gráficos. Sin embargo, lo más habitual es crear este listado de elementos con el formato de **Historia de Usuario**.



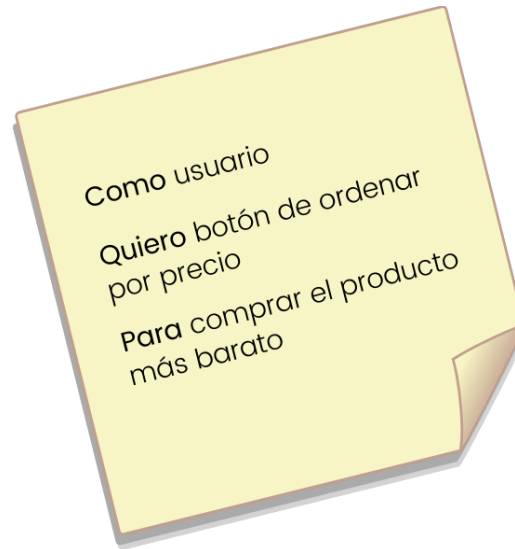
3.5.3 User story

Una **historia de usuario** o **user story** (elemento de la lista del product backlog) es una característica o funcionalidad que tiene **valor para el usuario final** y es lo suficientemente **pequeña** para que pueda ser desarrollada en un sprint. El **formato** de Historia de Usuario es el siguiente:

- **Como** <tipo de usuario>
- **Quiero** <necesidad a implementar>
- **Para** <funcionalidad>



Un **ejemplo** aplicado podría ser:



3.5.4 Estimación

Cada historia de usuario del product backlog debe ser estimada. Lo importante de la estimación es la realización de una **reunión** para ver qué tipo de **dificultades, problemas y riesgos** pueden surgir durante el sprint. Se suelen tener en cuenta las siguientes **características**:

- **Esfuerzo**: cuánto tiempo llevará su desarrollo.
- **Complejidad**: cómo de difícil es.
- **Incertidumbre**: grado de inseguridad para su implementación.

3.5.5 Planning poker o Scrum poker

Para la **realización de estimaciones** se ideó el planning poker o scrum poker que ayuda a la **estimación de la duración de las historias de usuario** de forma rápida evitando largas discusiones por parte del equipo de desarrollo.

El funcionamiento es muy sencillo: durante la estimación **cada miembro del equipo dispone de un juego de cartas**, y en la estimación de cada tarea, todos los miembros vuelven boca arriba la combinación que suma el esfuerzo estimado.



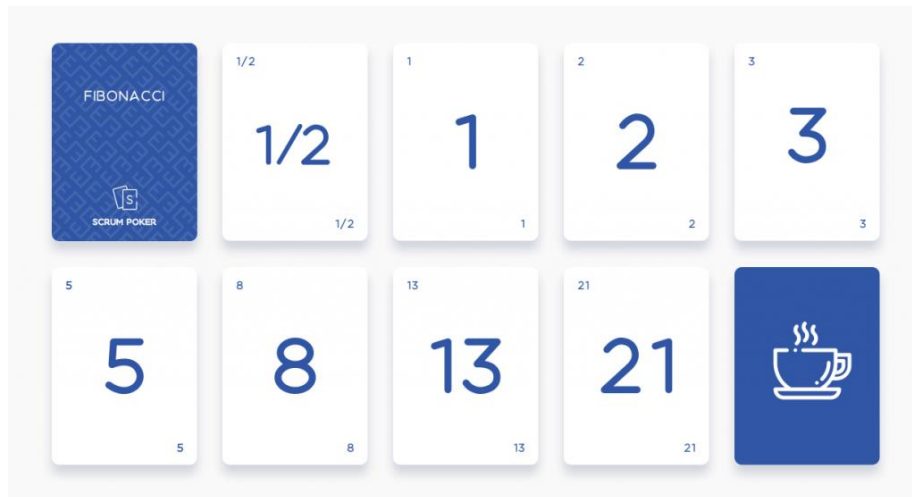
Cada equipo puede utilizar un juego de cartas con las numeraciones adecuadas a la unidad de esfuerzo con la que trabajan, y el tamaño máximo de tarea que se va a estimar. Las tareas que exceden el tamaño máximo definido deben descomponerse en tareas de menor tamaño.

Las dos variantes de **planning poker** más conocidas son las cartas con **serie de Fibonacci** y con **tallas de ropa**.

3.5.5.1 Cartas basadas en la serie de Fibonacci

A continuación, se propone la operativa que se llevaría durante una reunión de estimación mediante cartas basadas en la [serie de Fibonacci](#):

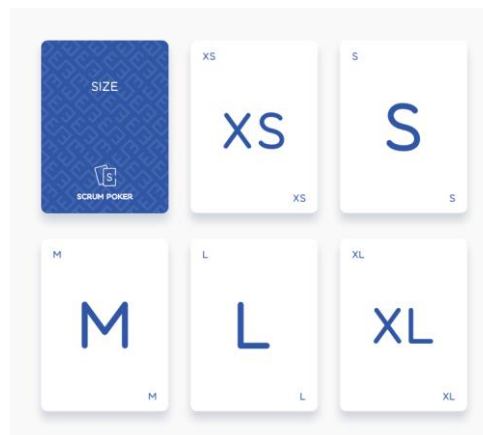
- 1) **Cada miembro dispone de un mazo de cartas.** Cada componente dispone de un mazo de cartas con puntuación basada en la serie Fibonacci: 0, $\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 21, infinito.
- 2) **Cliente o Product Owner lee la historia.** Las historias de usuario son explicadas por el cliente o por el *Product Owner*. El equipo de desarrollo debe realizar todas las preguntas necesarias para la estimación.
- 3) **El Development Team selecciona las cartas.** Cada miembro del equipo de desarrollo selecciona una carta, sin mostrársela al resto.
- 4) **Se muestran las cartas.** Todas las cartas se levantan a la vez.
- 5) **Argumentación de las diferencias.** Se argumentan las diferencias entre las estimaciones más extremas.
- 6) **Nueva estimación.** Se vuelve a realizar una nueva estimación hasta que se consiga un consenso



Scrum planning poker – Fibonacci

3.5.5.2 Cartas con talla de ropa

El mecanismo es el mismo que mediante cartas con serie de Fibonacci. La diferencia es que en vez de utilizar números, se utilizan **tallas** de camiseta o de ropa. De esta forma se pueden definir otros valores diferentes que se pueden asociar a cada una de las cartas durante en la estimación.



3.5.6 Orden de los PBIs

Los **PBIs prioritarios** deben estar **situados en la parte superior del listado** ya que son los primeros que se desarrollarán. De tal manera que, si al finalizar el sprint no se han completado todos los PBIs, serán los de menor importancia (los de abajo) los que no se hayan terminado.



Para priorizar la importancia de los elementos de la lista se pueden utilizar diferentes técnicas. Una de las más destacadas es el **método MoSCoW**:

- **M:** Must have, elementos importantes que se deben desarrollar.
- **S:** Should have, elementos importantes que pueden esperar a otra iteración y que no son necesarios en la entrega actual.
- **C:** Could have, elementos que solo se tendrán en cuenta si hay tiempo suficiente. Son elementos deseables, pero no necesarios.
- **W:** Won't have, elementos que seguro que no dará tiempo a desarrollar.

3.5.7 Tablero Scrum

El tablero Scrum es una herramienta para **monitorizar la carga de trabajo y facilitar la sincronización diaria**. El tablero se puede dibujar en una pizarra, colgar en un corcho o pegar en una pared. Además, existen numerosas **herramientas online** que permiten crear tableros Scrum de forma colaborativa y en tiempo real, como por ejemplo la herramienta [Trello](#).

3.6 Ceremonias o reuniones

3.6.1 Daily meeting (reunión diaria)

El objetivo de las reuniones diarias es la **comunicación del equipo para resolver posibles barreras y tomar decisiones** para asegurar la consecución de los objetivos del sprint. En las reuniones diarias cada miembro dispone de unos 2-3 minutos para responder a las siguientes preguntas:

- ¿Qué hiciste ayer?
- ¿Qué harás hoy?
- ¿Tienes algún impedimento?



3.6.2 Sprint Planning (planificación del sprint)

El objetivo de la reunión es el de planificar la **cantidad de trabajo que el equipo va a poder soportar** durante el próximo sprint.

La reunión de planificación responde a dos preguntas:

- **¿Qué puede entregarse en el sprint?** El product owner detalla las funcionalidades que hay que desarrollar en este sprint.
- **¿Cómo se realizará el trabajo?** El equipo de desarrollo estudia las historias de usuario y **estima el esfuerzo** que será necesario. Para ello, se suele usar la **técnica de planning poker**.

3.6.3 Spring Review (revision de sprint)

El objetivo de la reunión es realizar una revisión del sprint para **facilitar la retroalimentación de información y la colaboración**.

3.6.4 Sprint Retrospective (retrospectiva del sprint)

El objetivo de esta reunión es el de **analizar el trabajo realizado por el equipo y reflexionar sobre acciones de mejora**.

3.6.5 Refinement (refinamiento)

Es un **proceso continuo donde se revisan y examinan diferentes elementos de la Pila de Producto**. Este proceso puede hacerse en diferentes sesiones a lo largo del sprint o en una sola reunión. Sin embargo, resulta recomendable que se realice de forma frecuente.



3.7 Ejemplo de proyecto con metodología SCRUM

Generalmente este tipo de proyectos que utilizan Scrum, se gestiona con la ayuda de un **Software de Gestión Ágil**, como puede ser Jira, Monday, GanttPro u Wrike.

Un proyecto sencillo con metodología scrum sigue una serie de **pasos** que comúnmente llevaría a un equipo de desarrollo a trabajar en un proyecto web.

1) Definir el Product Backlog

El proceso inicia con una **reunión con el cliente**, donde se define qué es lo que espera y qué funcionalidades deberá tener el proyecto web. En esta reunión deberán participar el Product Owner o el Scrum Master. Si por **ejemplo**, se trata de una página web de **tienda virtual**, algunos requisitos podrían ser estos:

- Debe mostrar el catálogo de productos con fotos.
- Tiene que contar con la posibilidad de ver diferentes colores por cada diseño.
- Debe ser sencilla e intuitiva.
- Debe ser acorde en su diseño a la identidad de la marca
- Tiene que ofrecer la posibilidad de pagar con diferentes métodos.

Esta será la primera versión del **product backlog**, que se escribirá en forma de **Historias de usuario**.

2) Elegir al equipo

Estos nuevos **proyectos que utilizan Scrum** con características particulares puede requerir ciertos perfiles dentro del equipo Scrum. Diseño web, maquetación, programación web, programación de aplicaciones móviles Android, iPhone; serán actividades vitales. Estas personas son elegidas según las habilidades necesarias para hacer el proyecto, así que en muchas ocasiones proceden de departamentos diferentes.

3) Sprint Planning

La **reunión** donde se define la **duración de las iteraciones**, así como se organizan los requisitos más prioritarios es el Sprint Planning.



Con la participación de todo el equipo se redefine el Product Backlog, y se hacen los ajustes pertinentes para planificar **el cronograma** con la definición de la cantidad de tareas a realizar durante los **Sprints**. Esto puede hacerse con técnicas como el **Planning Poker**.

4) Sprint

Con la constante supervisión del Scrum Master, **comenzarán las iteraciones**. Este Sprint durará un tiempo determinado, en función de la complejidad del proyecto, normalmente 7, 15 ó 30 días.

La parte final del Sprint es revisar el progreso, y presentarlo al cliente, en forma de **entregas parciales**. Cada día, puede realizarse un **Daily Scrum**, donde se revise el avance diario y se realicen los ajustes pertinentes según los KPI como el **Burndown Chart**.

5) Sprint Review

Es una reunión donde todo el equipo Scrum se junta para hacer un **repaso del estado actual del Sprint, con el cliente**. Esta reunión permitirá redefinir el Product Backlog y continuar con el proceso.

Esta reunión tendrá una hora, por cada semana de trabajo.

6) Monitoreo y Progreso constante

Cuando finalizan los Sprints, e incluso cuando ya se finalizaron los proyectos con metodología Scrum, es necesario sacar **conclusiones** y **Know How**, para los próximos proyectos venideros.