



Buenas prácticas de software



UNIDAD 3 Mejores prácticas para la ingeniería de diseño.



Buenas prácticas de software



3.3 Mejores prácticas para el diseño de experiencias de usuario.



Buenas prácticas de software



¿Qué es la experiencia de usuario?

Las experiencias de usuario (UX) (*user experience*) permiten identificar de manera general lo que la persona (usuario) expresa y se incluyen sus observaciones, emociones, acciones y percepciones y que se respuestan en el sistema.

La facilidad de uso, así como la accesibilidad son algunos de los criterios que requieren se tomen en cuenta durante el desarrollo.

Cuando nos referimos a las UX se deben considerar las interfaces de usuario que sean amigables.

La experiencia de usuario son cambiantes debido a las interacciones y las necesidades nuevas de los usuarios, así como a los cambios tecnológicos.



Buenas prácticas de software



Existen 4 tipos de diseño UX.

1. Diseño de interacción
2. Diseño visual
3. Investigación del usuario
4. Arquitectura de la información.

5.



Buenas prácticas de software



Diseño de interacción.

El diseño de interacción (IxD), el propósito es lograr la óptima relación entre el usuario y el software. Lo que se desea es lograr la mejor interactividad del software que permita retener la atención y el del usuario cuando interactúa con el sistema o software.

Se deben considerar cada uno de los elementos que lo integran: La uso, el diseño de interfaz (gráfico), la funcionalidad y por supuesto la legibilidad.

Diseño visual.

No es otra cosa que el diseño de la interfaz del sistema y en el se integran elementos de programación del frontend, para lo cual se puede hacer uso de herramientas como el CSS3 y se requiere de conocimientos de diseño gráfico.



Buenas prácticas de software



Investigación del usuario

Los desarrolladores deben de determinar qué se quiere así como cuales son las necesidades de los clientes y/o usuarios.

Es fundamental que el software resuelva un problema o mejore actividades y procesos del cliente o usuario. Por lo tanto es muy importante hacer la indagación puntual de qué requieren exactamente los clientes o usuarios.

Arquitectura de la información

Todo software o sistema está compuesto por datos e información, por lo que diseñar y desarrollar una arquitectura que permita el fácil acceso a la información es pertinente, al diseño y desarrollo de la arquitectura del software o sistema se le llama el backend.



Entregables en el diseño UX

Una buena practica en el desarrollo de sistemas deinformació o de software es la elaboración de la documentación y esta tambien debe incluir las actividades de las UX

La documentación es la evidencia de que los diseñadores indicaron de manera efectiva los requerimientos del diseño indicando claramente a los participantes el por qué de las recomendaciones de cambios y mejoras del software.

Investigación del usuario

El equipo de trabajo genera reuniones con el cliente o usuarios para identificar con claridad lo que necesita.

Con ello el equipo de diseño comprende y se identifica con las necesidades del usuario.



Evaluación de la competencia.

Evaluar las fortalezas y debilidades de tus competidores es una forma de mejorar tu propia estrategia de experiencia del usuario.

Puedes hacerlo mediante la creación de un informe de análisis competitivo, que detalle el diseño de interacción de tu competencia y proporcione un análisis de los riesgos y las oportunidades que perdió.

Diseño de interacción

Un diseño de interacción puede ser un prototipo para que las personas puedan revisar cómo ocurrirían las interacciones con el sitio.

Muestra cómo los usuarios completan las tareas clave, obtienen información, usan un software.

Cuál es el flujo de búsqueda de información y la facilidad con la que el Software puede utilizarse.

El prototipo debe contar con el máximo de requerimientos del software final para que se pueda tener una aprobación del diseño antes de comenzar a construirlo.



Buenas prácticas de software



3.3.1 El diseño amigable e intuitivo.

¿Qué es el diseño amigable?

El diseño amigable tiene como el objetivo lograr que el usuario accese fácilmente a la información.

El diseño hace referencia al frontend del software que indica hace referencia a la parte frontal o interfaz física del software o sistema, es muy común hablar de la estructura y estilos, colores, fondos, etc.

Características de un diseño amigable.

Navegación.

La navegación es uno de los elementos más importantes cuando hablamos de usabilidad del software.

Permite a los usuarios visitar las principales funciones del software o sistema.

Por lo que se debe de poner especial cuidado en el diseño de menús y barras de navegación, que debe ser sencilla.

Contenidos.

Los contenidos de información y datos los proporciona el usuario y es responsabilidad de él la calidad de los mismos.

Aunque el usuario o cliente es el que indica el diseño es importante que las imágenes que se diseñan sean de calidad y relacionadas con el propósito del software o sistema y que llamen la atención.

La comunicación mediante la imagen dice mucho de la información del software.

Recomendable utilizar estándares en textos.

Se deben organizar y categorizar las funciones y subfunciones tomando en cuenta el diseño funcional del software.



Buenas prácticas de software



¿Qué es el diseño intuitivo?

Intuitivo significa que no hay necesidad de indagar mucho sobre como acceder a la información.
La interfaz es intuitiva cuando la navegación es sencilla.

Un buen diseño de la interfaz se puede lograr con el uso de las hojas de estilo CSS, HTML, React, Redux, Angular, Bootstrap, PostCSS y algunos otros, y por lo tanto se garantiza una experiencia de navegación satisfactoria.

Recomendaciones.

- Utiliza formatos estandarizados.
- Evita la saturación.
- Y da legibilidad.
- Revisa la velocidad de carga.
- Diseño mobile-friendly.



Buenas prácticas de software



3.3.2 Herramientas abiertas para el diseño de experiencia UX/UI.

Las siguientes herramientas se pueden utilizar gratuitas o con un tiempo de prueba.

Consulta y revisa sus propiedades.

Herramienta	Página Web
Woopra	www.woopra.com
Appsee	www.appsee.com
Google Analytics	www.google.com



Buenas prácticas de software



3.3.3 Herramientas de pago para el diseño de experiencia UX/UI.

Figma

Permite crear prototipos de alta fidelidad.

Tiene una funcionalidad para el trabajo colaborativo.

Esta disponible en la nube para trabajos de documentos en línea en tiempo real.

Las funciones son:

Se logra la ilustración de las transiciones utilizando el Smart Animate.

Permite la alternación entre los archivos del diseño y del prototipo.

Se pueden hacer comentarios que van generándose en el desarrollo y esto lo hace dentro del prototipo lo que permite que todo el equipo esté informado.



Sketch

Se usa en el diseño de los prototipos.

La interfaz es muy intuitiva lo que facilita el inicio del diseño de una manera fácil y rápida.

Las funciones que tiene:

Permite crear componentes que son reutilizables los cuales se actualizan de inmediato durante el diseño.

Las bibliotecas se sincronizan y comparten ya se que guardan todos los cambio o desarrollos en el mismo repositorio.

Se lleva a cabo una entrega agíl a los desarrolladores con cuyas funciones se comparten.



Adobe XD

Se basa en vectores que permiten el desarrollo de apps web y móviles.

Se pueden crear los flujos de los usuario y los diseños hi-fi así como los prototipos de tipo interactivos utilizando una plataforma, lo que permite que el equipo se sincronise en cada una de las etapas del desarrollo.

Funciones:

Los documentos se pueden trabajar de manera simultánea lo que permite la colaboración.

Permite la revisión de como va evolucionando el diseño conjugado con el historial de las versiones que se van generando.

Los interesados pueden visualizar los diseño al ser estos compartidos en tiempo real.



Buenas prácticas de software



3.4 Mejores prácticas para el desarrollo de software.



Buenas prácticas de software



Desarrollo de software

- El desarrollo de software es etapa en la que se construye el software.
- Incluye la producción y entrega del software.
- Lo realiza los desarrolladores backend y frontend
- La gestión la realiza el DevOps



Buenas prácticas de software



¿Actividades que se llevan a cabo en el desarrollo del software?

1. Se lleva a cabo la gestión del proceso de desarrollo.
2. Se desarrolla la arquitectura del software.
3. Se diseña y se desarrollan las bases de datos.
4. Documentar el desarrollo.
5. Se realiza el testeo.



Buenas prácticas de software



3.4.1 Estandarizar las reglas del desarrollo.

¿Qué es un estandar en desarrollo del software?

Son pautas que se adoptaron para conseguir uniformidad en el desarrollo, facilitando el mantenimiento y la actualización del sw.



Buenas prácticas de software



Estándar ISO número 12207: corresponde a los modelos del Ciclo de Vida del Sw.

Indica una serie de requisitos que enumera paso a paso a seguir en el desarrollo de sw

El estándar esta compuesto por 17 procesos que se pueden clasificar en 3 categorias:

Procesos principales: en este se realiza:

- La adquisición

- El suministro

- El desarrollo

- La operación

- Mantenimiento.

Procesos de apoyo permite que se realizan las siguientes actividades:

- Documentación

- Gestión de la configuración

- Aseguramiento de la calidad

- La verificación y validación

Así mismo se realiza la revisión en conjunto

Se lleva a cabo la atuditoría y en su caso la soclución de problemas.



Buenas prácticas de software



3.4.2 Herramientas para la generación de código (IDE).

Las herramientas de generación de código, generan automáticamente el código de un programa.

Las herramientas ayudan a los desarrolladores a desarrollar rápidamente programas.



Buenas prácticas de software



¿Qué son los generadores de código?

En la etapa de programación se utilizan herramientas para la generación de código.



Buenas prácticas de software



Herramienta de desarrollo, revisión y documentación de código.

GitHub es una herramientas automatizada que permite y facilita la gestión de las versiones del codigo del sistema o software, así mismo se puede revisar llevar a cabo un control de los cambios los cuales quedan documentados.

Consulta su pagina web para que identifiques sus funcionalidades.



Buenas prácticas de software



3.4.3 Documentar o comentar el código fuente.

¿Qué es documentar el código?

Documentar el código de un programa es:

Cualquier programador o desarrollador entiendan qué están haciendo y por qué es la finalidad de documentar el código que permite punto por punto incluir en un documento.



Buenas prácticas de software



¿Cuáles son los beneficios de documentar?

1. Ahorro de tiempo de desarrollo a mediano plazo.
2. Agilidad en el desarrollo.
3. Identificación de errores.
4. Comprensión de las funciones.
5. Evitar creación de código duplicado.
6. Modularidad.
7. Escalabilidad.
8. Fácil identificación de errores.



Buenas prácticas de software



3.4.4 Utilizar y construir librerías.

Una librería es un archivo o conjunto de archivos que se utilizan para facilitar la programación.

También llamadas “Frameworks” consiste en archivos de código que desarrollan una función específica.

Una biblioteca o librería (del inglés library) es un conjunto de implementación de funciones, codificadas en un lenguaje, que ofrece una interfaz para ejecutar la funcionalidad que se invoca.



Buenas prácticas de software



Ejemplos de librerías.

En los sistemas UNIX,

`/usr/include`

Estos ficheros se suelen llamar "de cabecera"

`#include`

Que los incluirá en el fuente durante la fase de preprocesado.



Buenas prácticas de software



Librerías más utilizadas en Python

Pygame

Es una librería de código abierto la cual nos permite crear aplicaciones multimedia.

Su principal enfoque es crear videojuegos.

```
1 import pygame
2 pygame.init()
3
```

Request

Es una librería permite realizar peticiones HTTP.

Es de utilidad cuando se requiere consumir servicios web.

```
1 import requests
2
3 r = requests.get('https://www.google.com')
4 print(r.text)
5
```

<https://www.crehana.com/mx/blog/desarrollo-web/librerias-python/>



Buenas prácticas de software



3.4. Utilizar controles propios

Los 10 mejores IDE de Python para potenciar el desarrollo y la depuración.

¿Qué es IDE?

Un desarrollador necesita un entorno de desarrollo integrado (IDE) de calidad como espacio de trabajo para sus códigos.

El IDE es software.

El IDE consta de herramientas de desarrollo que se utilizan para desarrollar software y probarlo.

Proporciona un entorno de desarrollo donde todas las herramientas están disponibles en una única interfaz gráfica de usuario (GUI) fácil de usar.



Buenas prácticas de software



Todo programador puede desarrollar su controles apoyando en los IDE.

Un IDE incluye principalmente:

Editor de código para escribir los códigos de software.

Automatización de construcción local.

Depurador de programas.

Los IDE ayudan a acelerar la actividad de desarrollo, con las siguientes características.

Configuración más rápida.

Tarea de desarrollo acelerado.

Aprendizaje continuo.



Buenas prácticas de software



Ejemplos de IDE de Python.

PyCharm
Kdevelop
Thonny
Visual Studio
Atom
LiClipse
Spyder
Pyzo
Geany
Wing