



# Buenas prácticas de software



## Unidad 3. Mejores prácticas para la ingeniería de diseño.



# Buenas prácticas de software



## 3.4 Mejores prácticas para el desarrollo de software.



# Buenas prácticas de software



## ¿Qué es el desarrollo del software?

- El desarrollo del software es la actividad 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 (acrónimo inglés de development –desarrollo- y operations –operaciones-).



# Buenas prácticas de software



## ¿Qué actividades 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. Diseño y desarrollo de 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 estándar en el desarrollo de software?

Los estándares son pautas que se adoptaron para conseguir uniformidad en el desarrollo, facilitando el mantenimiento y la actualización del software.



# Buenas prácticas de software



Los estándares de software permiten la utilización de estándares y metodologías para el diseño, programación, prueba y análisis del software desarrollado, con el objetivo de ofrecer una mayor confiabilidad, mantenibilidad en concordancia con los requisitos exigidos, mejorando la productividad y el control en la calidad de software, el uso de estándares mejora la eficacia y eficiencia en el desarrollo.



# Buenas prácticas de software

Estándar ISO 12207: Modelos de Ciclos de Vida del Software.

Estándar para los procesos de ciclo de vida del software.

El estándar indica una serie de procesos desde la recopilación de requisitos hasta la culminación del software.





# Buenas prácticas de software



El estándar comprende 17 procesos lo cuales son agrupados en tres categorías:

**Procesos principales:** Adquisición, suministro, desarrollo, operación y mantenimiento.

**Procesos de apoyo:** Documentación, gestión de la configuración, aseguramiento de la calidad, verificación, validación, revisión conjunta, auditoría y solución de problemas.

**Procesos de organización:** Gestión, infraestructura, mejora y recursos humanos.





# Buenas prácticas de software



## 3.4.2 Herramientas para la generación de código (IDE).

¿Qué son las herramientas de generación de código?

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 y es la actividad en la cual un compilador convierte un programa sintácticamente correcto en una serie de instrucciones a ser interpretadas por una máquina.



# Buenas prácticas de software



## Herramienta de revisión de código estático.

1. Review Board, es una herramienta de código abierto basada en la web para la revisión de códigos.
2. El lenguaje de programación Python y sus instaladores, MySQL o PostgreSQL como base de datos y un servidor web son los requisitos previos para ejecutar Review Board en un servidor.
3. Se pueden integrar el Review Board para el control de versiones: Git, Mercurial, CVS, Subversion y Perforce.
4. También se puede vincular el Review Board a Amazon S3 para almacenar capturas de pantalla directamente en la herramienta.



# Buenas prácticas de software



## Herramienta de desarrollo, revisión y documentación de código.

### 1. GitHub

GitHub se utiliza para mantener los repositorios de Git en la nube, puede que ya hayas usado bifurcaciones y solicitudes de revisión de código.

GitHub tiene una herramienta de revisión de código incorporada en sus solicitudes de extracción.

La herramienta de revisión de código está incluida en el servicio principal de GitHub, que proporciona un plan gratuito para los desarrolladores.



# Buenas prácticas de software



## Herramienta de desarrollo, revisión y documentación de código.

El plan gratuito de GitHub limita el número de usuarios a tres en los repositorios privados.

Los planes de pago empiezan en 7 dólares al mes.

GitHub permite a un revisor con acceso al repositorio de códigos asignarse a la solicitud de extracción y completar una revisión.

Un desarrollador que ha enviado la petición pull también puede solicitar una revisión de un administrador.

La herramienta de revisión de código también te permite resolver conflictos simples de Git a través de la interfaz web.

GitHub sólo admite los repositorios de Git alojados en GitHub.



# Buenas prácticas de software



## Herramienta de desarrollo, revisión y documentación de código.

Para mayor información de GitHub consulta la guía para principiantes sobre GitHub y las diferencias entre Git y GitHub en Internet.





# 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:

Añadir suficiente información como para explicar lo que hace, punto por punto, de forma que no sólo los ordenadores sepan qué hacer, sino que además cualquier programador o desarrollador entiendan qué están haciendo y por qué.





# Buenas prácticas de software



## 3.4.3 Documentar o comentar el código fuente.

### ¿Qué es documentar el código?

La documentación de un programa puede ser interna y externa.

La documentación interna es la contenida en líneas de comentarios.

La documentación externa incluye análisis, diagramas de flujo y/o pseudocódigos, manuales de usuario con instrucciones para ejecutar el programa y para interpretar los resultados.



# Buenas prácticas de software



## ¿Cuáles son los beneficios de documentar?

Ahorro de tiempo de desarrollo a mediano plazo.

Agilidad en el desarrollo.

Identificación de errores.

Comprensión de las funciones.

Evitar creación de código duplicado.

Modularidad.

Escalabilidad.

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.



## Ejemplos de librerías.

En los sistemas UNIX,

`/usr/include`

Estos ficheros se suelen llamar "de cabecera"

`#include`

Que los incluirá en la 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
```



# 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 sus 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