



DOCENTE:

AZAMAR PALMA IVAN.

ALUMNA:

XÓCHIHUA HERNÁNDEZ PERLA ESMERALDA.

MATERIA:

PROGRAMACIÓN EN TECNOLOGÍAS WEB PARA LA INDUSTRIA.

FECHA DE ENTREGA:

21 DE MARZO DE 2023.

INSTALACIÓN DE UN AMBIENTE DE DESARROLLO

UNIDAD 1

ÍNDICE:

[I. INSTALACIÓN DE UN IDE 2](#_Toc130257855)

[1.1 ¿Qué es un IDE? 3](#_Toc130257856)

[1.2 Características de un IDE 3](#_Toc130257857)

[1.3 Tipos y ejemplos de IDEs 3](#_Toc130257858)

[1.4. Ventajas y Desventajas 3](#_Toc130257859)

[1.5 Instalación de Visual Studio Code 3](#_Toc130257860)

[II. INSTALACIÓN DE UN LENGUAJE DEL LADO DEL SERVIDOR 2](#_Toc130257861)

[2.1 ¿Qué es un lenguaje del lado del servidor? 3](#_Toc130257862)

[2.2 Características del lenguaje del lado del servidor 3](#_Toc130257863)

[2.3 Tipos de lenguajes de lado del servidor 3](#_Toc130257864)

[2.3 Sitios estáticos y dinámicos 3](#_Toc130257865)

[2.4 Ejemplos de lenguajes del lado del servidor 3](#_Toc130257866)

[2.5 Instalación de Python 3](#_Toc130257867)

[III. INSTALACIÓN DE UN FRAMEWORK 2](#_Toc130257868)

[3.1 ¿Qué es un framework? 2](#_Toc130257869)

[3.2 Tipos de framework y Ejemplos 3](#_Toc130257870)

[3.3 Características de los frameworks 3](#_Toc130257871)

[3.4 Ventajas y Desventajas 4](#_Toc130257872)

[3.4 Instalación de Framework Angular 4](#_Toc130257873)

[IV. INSTALACIÓN Y CONFIGURACIÓN DE UN FRAMEWORK 5](#_Toc130257874)

[V. INSTALACIÓN Y CONFIGURACIÓN DE UNA BASE DE DATOS NoSQL 5](#_Toc130257875)

[VI. INSTALACIÓN DE DEPENDENCIAS 5](#_Toc130257876)

[VII. CONSTRUCCIÓN DE UN PROYECTO USANDO REPOSITORIOS Y VERSIONES 5](#_Toc130257877)

[VIII. INSTALACIÓN DE MONGODB 5](#_Toc130257878)

[IX. REFERENCIAS BIBLIOGRAFICAS 5](#_Toc130257879)

[Ilustración 1‑1 IDE Eclipce. 2](#_Toc130252214)

[Ilustración 1‑2 IDE Visual Studio Code. 2](#_Toc130252215)

[Ilustración 1‑3 NetBeans IDE. 2](#_Toc130252216)

[Ilustración 1‑4 IDE Code::Bloks. 2](#_Toc130252217)

[Ilustración 1‑5 IDE IntelliJ IDEA 2](#_Toc130252218)

[Ilustración 1‑6 IDE CodePen. 2](#_Toc130252219)

[Ilustración 1‑7 Descarga de IDE Visual Studio Code. 2](#_Toc130252220)

[Ilustración 1‑8 Ejecutar como administrador el IDE Visual Studio Code. 2](#_Toc130252221)

[Ilustración 1‑9 Términos del IDE Visual Studio Code. 2](#_Toc130252222)

[Ilustración 1‑10 Selección de tareas adicionales en el IDE. 2](#_Toc130252223)

[Ilustración 1‑11 Instalación de Visual Studio Code. 2](#_Toc130252224)

[Ilustración 1‑12 Proceso de instalación de visual studio code. 2](#_Toc130252225)

[Ilustración 1‑13 Instalación completa de visual studio code. 2](#_Toc130252226)

[Ilustración 1‑14 Visualización del IDE Visual Studio Code inicializado. 2](#_Toc130252227)

[Ilustración 2 ‑I‑1 Esquema de sitios estáticos. 2](#_Toc130252228)

# INSTALACIÓN DE UN IDE

## ¿Qué es un IDE?

Un IDE (Integrated Development Environment) es un software que proporciona un entorno integrado para el desarrollo de aplicaciones. Se utiliza para escribir, depurar y probar el código de programación en un solo lugar.

Un IDE suele incluir un editor de código, un depurador, un compilador y herramientas de automatización y gestión de proyectos, todo en una interfaz de usuario integrada. Además, puede ofrecer características como resaltado de sintaxis, autocompletado de código, asistencia para refactorizar código, integración con sistemas de control de versiones, entre otras. (Redác, 2022)

## Características de un IDE

* Editor de código integrado:

Un IDE proporciona un editor de código integrado que permite escribir, editar y guardar código fuente en un solo lugar.

* Depurador:

Un depurador es una herramienta integrada en un IDE que ayuda a encontrar y corregir errores en el código.

* Autocompletado de código:

Los IDEs ofrecen la funcionalidad de autocompletado de código, lo que significa que el IDE puede sugerir y completar automáticamente el código que está siendo escrito.

* Integración con sistemas de control de versiones:

Los IDEs pueden integrarse con sistemas de control de versiones como Git, SVN, etc. para facilitar la gestión y el seguimiento de los cambios en el código fuente.

* Herramientas de refactorización de código:

Los IDEs pueden ofrecer herramientas de refactorización de código para optimizar el código fuente y hacerlo más legible y mantenible.

* Pruebas unitarias integradas:

Los IDEs pueden ofrecer la capacidad de ejecutar pruebas unitarias integradas en el IDE, lo que facilita la ejecución de pruebas de forma rápida y sencilla.

* Compatibilidad con múltiples plataformas:

Los IDEs pueden ser compatibles con diferentes plataformas y sistemas operativos, lo que permite desarrollar aplicaciones en diferentes entornos.

* Integración con herramientas de compilación:

Los IDEs pueden estar integrados con herramientas de compilación, lo que facilita la creación de ejecutables y la distribución de aplicaciones. (Denied, s.f.)

## Tipos y ejemplos de IDEs

1. IDEs de lenguaje de programación específico:

Estos IDEs están diseñados específicamente para un lenguaje de programación en particular, por ejemplo, Eclipse para Java, Visual Studio para .NET, Xcode para iOS y macOS, etc.

Imagen que contiene Diagrama

Descripción generada automáticamente

Ilustración I‑1 IDE Eclipce.



Ilustración I‑2 IDE Visual Studio Code.

1. IDEs multi-lenguaje:

Estos IDEs permiten el desarrollo de aplicaciones en varios lenguajes de programación. Algunos ejemplos son NetBeans, que admite múltiples lenguajes de programación como Java, PHP, C++, etc., y Visual Studio Code, que admite múltiples lenguajes de programación como JavaScript, Python, C++, entre otros.

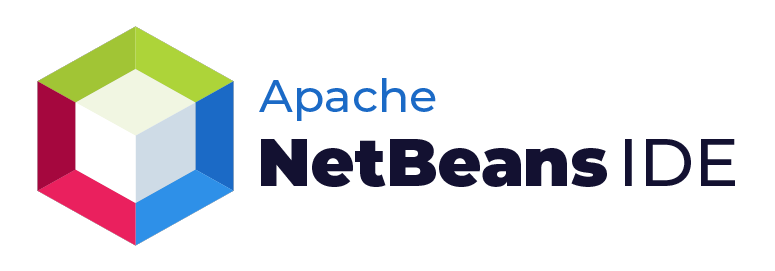


Ilustración I‑3 NetBeans IDE.

1. IDEs de código abierto:

Estos IDEs son de código abierto y están disponibles gratuitamente para su uso y modificación. Algunos ejemplos son Eclipse, NetBeans, Code::Blocks, etc.

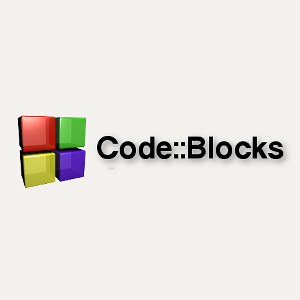


Ilustración I‑4 IDE Code::Bloks.

1. IDEs comerciales:

Estos IDEs son de pago y suelen ofrecer una amplia gama de características y herramientas. Algunos ejemplos son Visual Studio, IntelliJ IDEA, PyCharm, etc.



Ilustración I‑5 IDE IntelliJ IDEA

1. IDEs en línea:

Estos IDEs se ejecutan en línea en el navegador web y permiten el desarrollo de aplicaciones en línea. Algunos ejemplos son Cloud9, CodePen, JSFiddle, entre otros.

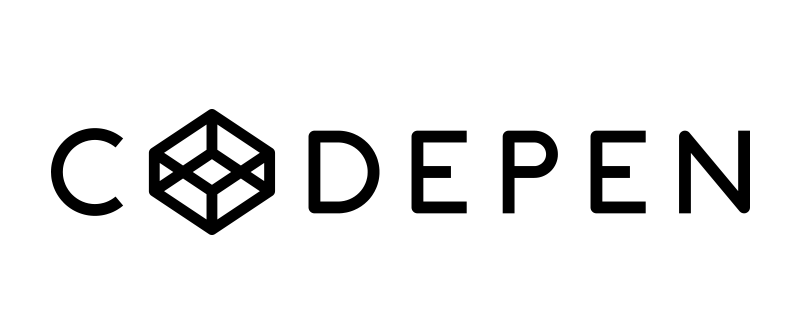


Ilustración I‑6 IDE CodePen.

## 1.4. Ventajas y Desventajas

Ventajas:

Mayor productividad: un IDE proporciona herramientas integradas para la edición de código, la depuración, la compilación y otras tareas relacionadas con el desarrollo de software, lo que puede aumentar la productividad del programador.

Facilidad de uso: la interfaz de usuario unificada de un IDE proporciona una experiencia de usuario más consistente y fácil de usar para los programadores.

Automatización: un IDE puede automatizar tareas como la generación de código, la refactorización y la generación de informes, lo que puede ahorrar tiempo y reducir errores.

Integración con otros servicios: un IDE puede integrarse con otros servicios de desarrollo de software, como sistemas de control de versiones y herramientas de análisis de código, lo que puede mejorar la calidad del software y reducir el tiempo de desarrollo.

Desventajas:

Aprendizaje: un IDE puede ser complicado para los programadores que no estén familiarizados con él. Puede llevar tiempo aprender a utilizar todas las funciones del IDE y aprovechar al máximo su potencial.

Sobrecarga de funciones: algunos IDE pueden proporcionar demasiadas funciones, lo que puede ser abrumador para algunos programadores y hacer que el IDE sea más lento.

Limitaciones en la personalización: algunos IDE pueden limitar la personalización del entorno de desarrollo, lo que puede dificultar la creación de un entorno de desarrollo óptimo para algunos programadores.

## Instalación de Visual Studio Code

1. Ingresar al siguiente enlace para descargar el IDE Visual Studio Code:

<https://visualstudio.microsoft.com/es/downloads/>.

En el cual existen dos apartados elegiremos el de nuestra preferencia en este caso descargaremos el de Windows x64.

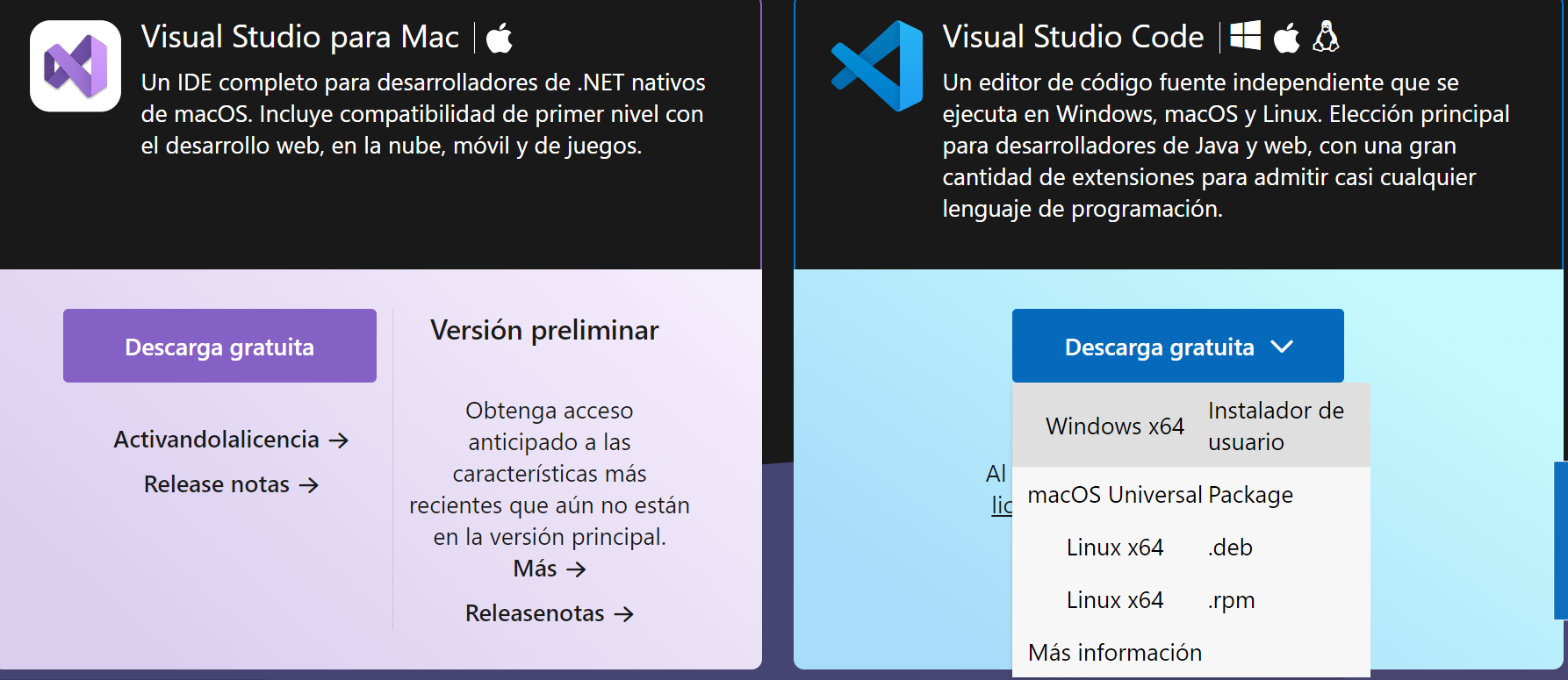
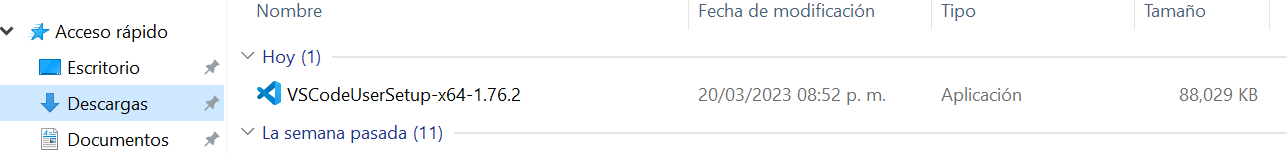


Ilustración I‑7 Descarga de IDE Visual Studio Code.

1. Cuando termine la descarga se verá de esta manera:



1. Dándole click derecho a la descarga aparece un recuadro en el cual aparece ejecutar como administrador proseguimos a oprimir ese apartado, aparecerá una ventana pidiendo permisos de administrador y se los otorgamos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración I‑8 Ejecutar como administrador el IDE Visual Studio Code.

1. Aparecerá una ventana la cual muestra los términos del IDE aceptamos para poder proseguir con la instalación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración I‑9 Términos del IDE Visual Studio Code.

1. Después de aceptar los términos aparecerá una selección de tareas que deseas tener en el IDE, sin embargo, en nuestro caso dejaremos las que ya están marcadas por este y por lo tanto le damos siguiente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración I‑10 Selección de tareas adicionales en el IDE.

1. Aquí nos iremos directo a realizar la instalación.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración I‑11 Instalación de Visual Studio Code.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración I‑12 Proceso de instalación de visual studio code.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración I‑13 Instalación completa de visual studio code.

1. Visualización del IDE inicializado para dar uso de este.

Captura de pantalla de un celular

Descripción generada automáticamente

Ilustración I‑14 Visualización del IDE Visual Studio Code inicializado.

# INSTALACIÓN DE UN LENGUAJE DEL LADO DEL SERVIDOR

## 2.1 ¿Qué es un lenguaje del lado del servidor?

Un lenguaje a lado del servidor (también conocido como server-side scripting language) es un lenguaje de programación que se utiliza para desarrollar aplicaciones web que se ejecutan en un servidor web. Estos lenguajes permiten crear aplicaciones web dinámicas que interactúan con bases de datos, realizan cálculos complejos y generan contenido dinámico en respuesta a las solicitudes del usuario.

## 2.2 Características del lenguaje del lado del servidor

Interacción con bases de datos:

Los lenguajes del lado del servidor tienen la capacidad de interactuar con bases de datos, lo que permite a las aplicaciones web almacenar y recuperar datos. Esto permite a las aplicaciones web del lado del servidor realizar cálculos complejos y personalizar el contenido en función de las preferencias del usuario.

Generación de contenido dinámico:

Los lenguajes del lado del servidor permiten a los desarrolladores generar contenido dinámico en respuesta a la interacción del usuario. Esto significa que las aplicaciones web pueden mostrar información personalizada para cada usuario, generar gráficos y tablas dinámicas, y realizar cálculos complejos en tiempo real.

Manejo de formularios y cookies:

Los lenguajes del lado del servidor pueden manejar formularios y cookies, lo que permite a las aplicaciones web interactuar con los usuarios de forma dinámica. Por ejemplo, las aplicaciones web pueden validar los datos de entrada del usuario y almacenar información personalizada en cookies para su uso posterior.

Implementación de seguridad:

Los lenguajes del lado del servidor permiten a los desarrolladores implementar medidas de seguridad para proteger las aplicaciones web y los datos de los usuarios. Esto puede incluir la validación de datos de entrada, la prevención de ataques de inyección de código y la implementación de autenticación y autorización.

Integración con otras tecnologías web:

Los lenguajes del lado del servidor pueden integrarse con otras tecnologías web, como HTML, CSS, JavaScript y bibliotecas y frameworks de desarrollo web. Esto permite a los desarrolladores crear aplicaciones web complejas y escalables utilizando una variedad de tecnologías. (Axarnet, s.f.)

## 2.3 Tipos de lenguajes de lado del servidor

Lenguajes de script: Estos lenguajes están diseñados para escribir pequeños scripts que se ejecutan en el servidor web en tiempo real. Ejemplos de lenguajes de script son PHP, Python, Ruby y Perl.

Lenguajes de programación de propósito general: Estos lenguajes son más potentes que los lenguajes de script y se utilizan para desarrollar aplicaciones web más complejas. Ejemplos de lenguajes de programación de propósito general son Java, C#, y Go.

Lenguajes de marcas: Estos lenguajes se utilizan para escribir páginas web y para estructurar el contenido. Ejemplos de lenguajes de marcas son HTML, XML y XHTML.

Lenguajes de consulta: Estos lenguajes se utilizan para realizar consultas en bases de datos. Ejemplos de lenguajes de consulta son SQL y XQuery.

Lenguajes de programación para aplicaciones específicas: Estos lenguajes se utilizan para desarrollar aplicaciones específicas, como aplicaciones para redes sociales o aplicaciones de comercio electrónico. Ejemplos de lenguajes de programación para aplicaciones específicas son ActionScript y ColdFusion.

## 2.3 Sitios estáticos y dinámicos

Un sitio web estático es una página web que se compone de archivos HTML, CSS y JavaScript que son entregados al cliente tal cual se encuentran en el servidor. Estos archivos se almacenan en el servidor web y se entregan al cliente en su forma original cada vez que el usuario accede al sitio. Los sitios web estáticos son relativamente simples y fáciles de crear, pero carecen de la capacidad de personalizar el contenido en función del usuario.

Por otro lado, un sitio web dinámico es una página web que se compone de archivos que son generados en tiempo real por un lenguaje de lado del servidor, como PHP, Python o Ruby. Estos archivos se generan en respuesta a las solicitudes del usuario, lo que significa que el contenido puede personalizarse en función de las preferencias del usuario o de otros datos dinámicos. Los sitios web dinámicos pueden incluir características como la autenticación de usuarios, la generación de contenido personalizado, el manejo de formularios y cookies, y la interacción con bases de datos. (Desconocido, 2022)

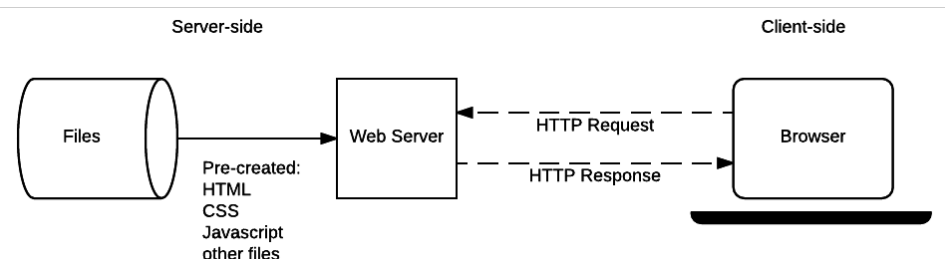


Ilustración II‑1 Esquema de sitios estáticos.

Diagrama

Descripción generada automáticamente

Ilustración II‑2 Esquema de sitios dinámicos.

## 2.4 Ejemplos de lenguajes del lado del servidor

* PHP:

Es uno de los lenguajes a lado del servidor más populares y es utilizado por una gran cantidad de sitios web. Es un lenguaje de código abierto y es fácil de aprender.

* Python:

Es otro lenguaje a lado del servidor popular y versátil que se utiliza en una amplia variedad de aplicaciones web. Tiene una sintaxis clara y concisa que lo hace fácil de leer y escribir.

* Ruby:

Es un lenguaje a lado del servidor que se utiliza con frecuencia en el desarrollo de aplicaciones web y móviles. Es conocido por su enfoque en la programación orientada a objetos y su marco de trabajo Ruby on Rails.

* Java:

Es un lenguaje a lado del servidor popular que se utiliza en muchas aplicaciones empresariales. Es conocido por su seguridad y escalabilidad.

* JavaScript:

Aunque se utiliza principalmente como un lenguaje del lado del cliente, también se puede utilizar como un lenguaje del lado del servidor a través de plataformas como Node.js.

## 2.5 Instalación de Python

1. Ingresar al siguiente enlace para descargar el lenguaje de Python: <https://www.python.org/downloads/>, descargamos la última versión que se encuentra que es la 3.11.2.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

Ilustración II‑3Descarga del lenguaje Python.

1. Al descargarse proseguimos a darle click izquierdo y así mismo comenzara la instalación,

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Ilustración II‑4 Instalación de python

1. En este apartado se encuentran las características opcionales que tiene el lenguaje en nuestro caso dejaremos seleccionadas las que da el mismo lenguaje y le daremos next.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración II‑5 Configuración opcional de Python.

1. Aquí se encuentran las opciones avanzadas y dejaremos marcadas las que da el lenguaje.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Ilustración II‑6 Opciones avanzadas de Python.

1. En el proceso de instalación no demora mucho máximo 5 minutos en la instalación.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración II‑7 Proceso de instalación.

1. Aparecerá la pantalla en donde ya finalizo la instalación y procemos a darle close.

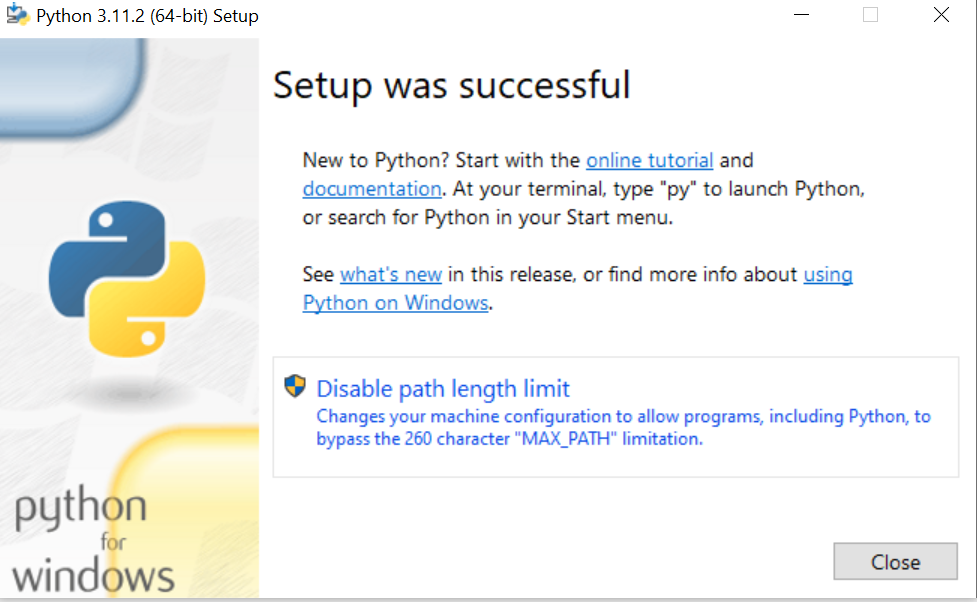


Ilustración II‑8 Instalación terminada de Python.

1. Así visualizamos Python en nuestra computadora.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración II‑9 Visualización de Python.

# INSTALACIÓN DE UN FRAMEWORK

## ¿Qué es un framework?

Un framework (marco de trabajo o estructura de trabajo en español) es un conjunto de herramientas, bibliotecas, convenciones y patrones de diseño que se utilizan para facilitar y agilizar el proceso de desarrollo de software.

Los frameworks proporcionan una base sólida y estructurada para construir aplicaciones, lo que permite a los desarrolladores enfocarse en la lógica de negocio y la funcionalidad de la aplicación, en lugar de tener que preocuparse por detalles técnicos de bajo nivel.

Además, los frameworks suelen promover buenas prácticas de programación, lo que puede ayudar a mejorar la calidad y la seguridad del software producido. Existen frameworks para diversos tipos de aplicaciones, desde aplicaciones web hasta aplicaciones móviles o de escritorio.

## Tipos de framework y Ejemplos

* Frameworks de desarrollo web:

son frameworks diseñados para facilitar la creación de aplicaciones web. Ejemplos incluyen Django (Python), Ruby on Rails (Ruby), Laravel (PHP), Express (Node.js), entre otros.

* Frameworks de front-end:

Se utilizan para desarrollar la interfaz de usuario (UI) de una aplicación web. Ejemplos incluyen React, Angular, Vue.js, Bootstrap, Foundation, entre otros.

* Frameworks de back-end:

Son diseñados para facilitar el desarrollo del lado del servidor de una aplicación web. Ejemplos incluyen Spring (Java), Flask (Python), Express (Node.js), Ruby on Rails (Ruby), entre otros.

* Frameworks de base de datos:

Se utilizan para facilitar el acceso y la manipulación de bases de datos. Ejemplos incluyen Hibernate (Java), Sequelize (Node.js), SQLAlchemy (Python), entre otros.

* Frameworks de pruebas:

Son diseñados para facilitar la escritura y ejecución de pruebas unitarias y de integración en una aplicación. Ejemplos incluyen JUnit (Java), pytest (Python), Mocha (Node.js), entre otros.

## Características de los frameworks

* Arquitectura definida: los frameworks suelen tener una arquitectura predefinida que se utiliza para estructurar la aplicación. Esto proporciona una base sólida para el desarrollo y ayuda a los desarrolladores a seguir buenas prácticas de programación.
* Abstracción de bajo nivel: los frameworks a menudo ocultan los detalles técnicos de bajo nivel para que los desarrolladores puedan enfocarse en la lógica de negocio de la aplicación. Por ejemplo, un framework de base de datos puede abstraer la complejidad de interactuar directamente con la base de datos.
* Reutilización de código: los frameworks suelen proporcionar un conjunto de herramientas y bibliotecas que los desarrolladores pueden utilizar para acelerar el proceso de desarrollo. Esto puede incluir código común para tareas como la validación de entradas de usuario o la autenticación de usuarios.
* Consistencia: los frameworks suelen imponer convenciones y patrones de diseño para garantizar que el código sea consistente en toda la aplicación. Esto facilita el mantenimiento y la colaboración entre desarrolladores.
* Comunidad: los frameworks a menudo tienen una comunidad activa de desarrolladores que contribuyen al desarrollo y mantenimiento del framework. Esto significa que hay una gran cantidad de recursos, documentación y soporte disponibles para los desarrolladores que utilizan el framework. (BootCamp, s.f.)

## 3.4 Ventajas y Desventajas

Ventajas:

1. Productividad: pueden acelerar el proceso de desarrollo de software al proporcionar una estructura y herramientas predefinidas que los desarrolladores pueden utilizar para evitar escribir código repetitivo y enfocarse en la lógica de negocio.
2. Consistencia: promueven la consistencia y las mejores prácticas de programación, lo que facilita el mantenimiento de la aplicación y la colaboración entre desarrolladores.
3. Comunidad: suelen tener una comunidad activa de desarrolladores que comparten recursos, documentación y soporte, lo que puede ser útil para resolver problemas y mejorar el conocimiento.
4. Seguridad: suelen implementar patrones de seguridad predefinidos, lo que ayuda a reducir el riesgo de vulnerabilidades de seguridad comunes.

Desventajas:

1. Limitaciones: pueden ser limitantes si se necesita una funcionalidad específica que el framework no soporta. Esto puede requerir trabajo adicional o la creación de soluciones personalizadas.
2. Curva de aprendizaje: pueden tener una curva de aprendizaje pronunciada, especialmente si los desarrolladores no están familiarizados con el framework específico.
3. Dependencia: pueden crear una dependencia en el framework específico, lo que puede ser un problema si se necesita cambiar a otra tecnología en el futuro.
4. Rendimiento: algunos frameworks pueden afectar negativamente el rendimiento de la aplicación, especialmente si se utilizan sin cuidado o en casos de uso específicos. (BootCamp, s.f.)

## Instalación de Framework Angular

# INSTALACIÓN Y CONFIGURACIÓN DE UN FRAMEWORK

# INSTALACIÓN Y CONFIGURACIÓN DE UNA BASE DE DATOS NoSQL

# INSTALACIÓN DE DEPENDENCIAS

# CONSTRUCCIÓN DE UN PROYECTO USANDO REPOSITORIOS Y VERSIONES

# INSTALACIÓN DE MONGODB

# REFERENCIAS BIBLIOGRAFICAS

Redác, T. (18 de Noviembre de 2022). *IDE : ¿Qué es un Entorno de Desarrollo Integrado?* Obtenido de Formation Data Science | DataScientest.com : https://datascientest.com/es/ide-que-es

Axarnet. (s.f.). *Lenguajes del lado del servidor* . Obtenido de https://axarnet.es/blog/lenguajes-del-lado-del-servidor

Denied, A. (s.f.). *Access Denied*. Obtenido de https://www.servicenow.com/es/now-platform/what-is-ide.html

Desconocido. (5 de Diciembre de 2022). *Introducción al lado servidor - Aprende sobre desarrollo web | MDN*. Obtenido de Introducción al lado servidor - Aprende sobre desarrollo web | MDN

Redác, T. (18 de Noviembre de 2022). *IDE : ¿Qué es un Entorno de Desarrollo Integrado?* Obtenido de Formation Data Science | DataScientest.com : https://datascientest.com/es/ide-que-es