



El futuro digital
es de todos

MinTIC

Ciclo 1

Fundamentos de programación con Python

Sesión 3: Introducción a Python

Programa Ciencias de la Computación e Inteligencia Artificial
Escuela de Ciencias Exactas e Ingeniería
Universidad Sergio Arboleda
Bogotá



UNIVERSIDAD
SERGIO ARBOLEDA

Mision
TIC2022



Agenda

1. Conceptos básicos de programación
2. Introducción Python
3. Entorno de desarrollo Visual Code





1. Conceptos básicos

- **Que es un Computador:** Es una herramienta que se utiliza para representar cualquier situación de la realidad en forma de datos, estos serán introducidos a la máquina, serán procesados y luego devolverán información.

Situación Real:

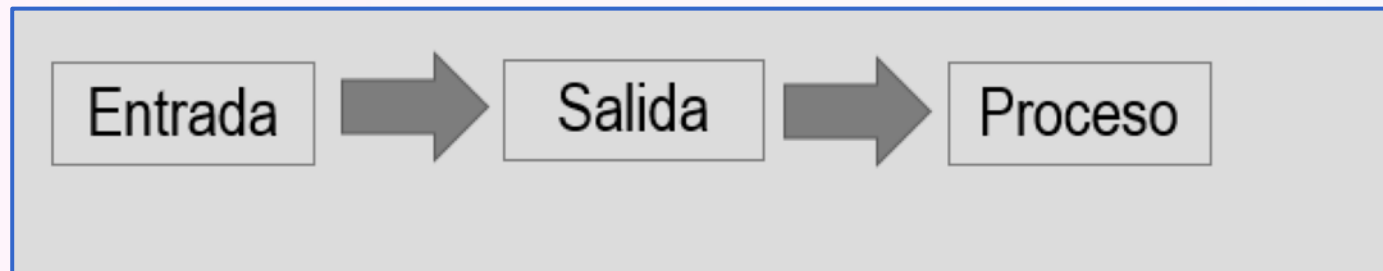
- Control de ventas en un concesionario
- Control de producción en una plantación de caña de azúcar
- Control de Inventario de una hipermercado





1. Conceptos básicos

- Toda situación que pueda ser abstraída y representada en forma de datos, puede ser manejada mediante una computadora.
- Esquema del proceso de datos.





1. Conceptos básicos

- *Ejemplo:* Una situación real que podríamos representar a través de datos, es el determinar el salario que obtiene un trabajador por su actividad realizada. Los datos serían: el nombre, la identificación, número de horas que trabajó y el valor de la hora, con estos datos luego se podrá calcular el salario que obtiene esta persona.
- Ejemplos planteados por los estudiantes (situaciones reales que pueden llevar a datos)





1. Conceptos básicos

- Del ejemplo anterior, es necesario ingresar a la máquina, los datos que representan al trabajador y el procedimiento que se debe realizar para generar la información requerida, en este caso la máquina, los procesará y devolverá como salida el nombre del trabajador y el sueldo.
- Ejemplos planteados por los estudiantes (identificar datos de salida y entrada y el procedimiento)





1. Conceptos básicos

- La forma en que vamos a ingresar tanto los datos de entrada, como los procedimientos a la máquina, es a través de lo que se conoce con el nombre de **Programa**.
- **¿Qué es un programa?**
Un programa hace que una computadora sea utilizable. Sin un programa, una computadora, incluso la más poderosa, no es más que un objeto.





1. Conceptos básicos

¿Qué es un programa?

Los computadores contemporáneos solo pueden evaluar los resultados de operaciones muy fundamentales como, sumar o dividir, pero pueden hacerlo muy rápido y pueden repetir estas acciones prácticamente cualquier cantidad de veces. Entonces, para llegar a la definición de programa veamos un ejemplo:

- Se quiere saber la velocidad promedio que se alcanzó durante un largo viaje, se sabe la distancia, el tiempo y se necesita la velocidad.





1. Conceptos básicos

El computador podrá calcular esto, pero no es consciente de cosas como la distancia, la velocidad o el tiempo. Por lo tanto, es necesario instruir a la computadora para que:

- Acepte un número que represente la distancia.
- Acepte un número que represente el tiempo de viaje.
- Divida el valor anterior por el segundo y almacene el resultado en la memoria.
- Muestre el resultado (representando la velocidad promedio) en un formato legible.





1. Conceptos básicos

Un **programa** es un conjunto de instrucciones que guían a la computadora para realizar alguna actividad o resolver algún problema. Un programa se compone de:

- Estructuras de datos
- Operaciones primitivas elementales
- Estructuras de control





2. Introducción Python

¿Qué es Python?

- Python es un lenguaje de programación sencillo, fácil de aprender, potente, de alto nivel y orientado a objetos.
- Python es también un lenguaje de scripting interpretado.
- Guido Van Rossum es conocido como el fundador de la programación en Python.





2. Introducción Python

- Python es un lenguaje de programación de propósito general, dinámico, de alto nivel e interpretado.
- Es simple y fácil de aprender y proporciona muchas estructuras de datos de alto nivel.
- Python es un lenguaje de scripting fácil de aprender pero potente y versátil, lo que lo hace atractivo para el desarrollo de aplicaciones.
- La sintaxis de Python y la tipificación dinámica, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para la creación de scripts y el desarrollo rápido de aplicaciones.
- Python soporta múltiples patrones de programación, incluyendo estilos de programación orientados a objetos, imperativos y funcionales o procedimentales





2. Introducción Python

- Se le conoce como un lenguaje multipropósito porque puede ser usado con varias finalidades.
- No se necesita usar tipos de datos para declarar variables porque está tipado dinámicamente.
- Python hace que el desarrollo y la depuración sean rápidos porque no hay ningún paso de compilación.
- El ciclo de edición-prueba-depuración es rápido.





2. Introducción Python



Características:

- Fácil de aprender y usar
- Es un lenguaje de programación de alto nivel y fácil de desarrollar.
- Lenguaje expresivo: significa que es más comprensible y legible.
- Lenguaje interpretado: el intérprete ejecuta el código línea por línea
- Lenguaje multiplataforma: puede funcionar igualmente en diferentes plataformas como Windows, Linux, Unix y Macintosh, etc.
- Lenguaje libre y de código abierto





Introducción Python



Áreas de aplicación:

- Aplicaciones web
- Aplicaciones GUI de escritorio
- Desarrollo de software
- Científicas y numéricas
- Aplicaciones empresariales
- Aplicaciones basadas en consolas
- Aplicaciones basadas en audio o vídeo
- Aplicaciones CAD 3D
- Aplicaciones empresariales





2. Introducción Python

¿Cómo empezar a utilizar Python?

El programa se puede instalar independientemente del sistema operativo que se tenga, el siguiente paso a paso nos permitirá descargar e instalar Python:

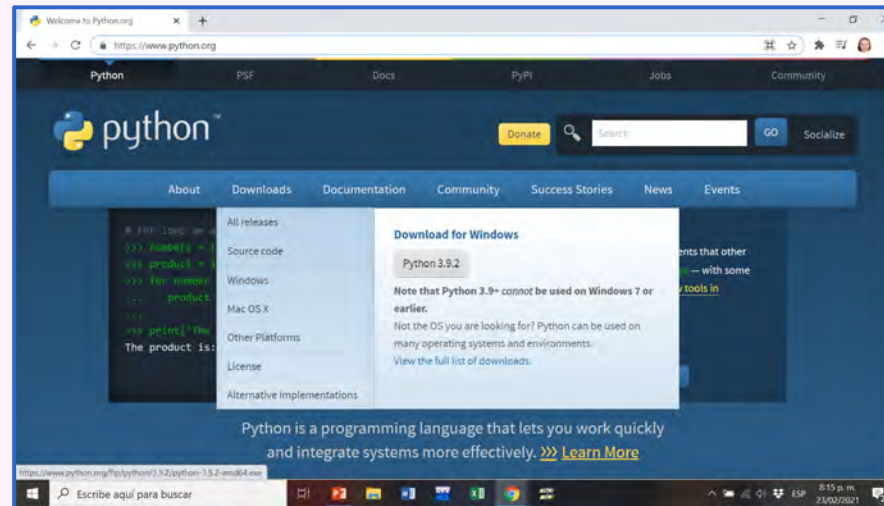
1. Dirigirse al siguiente dirección: <https://www.python.org/> Este es el sitio oficial del lenguaje de programación Python.
2. Seleccionar del menú la opción de Downloads, aquí encontraremos las diferentes opciones de descarga para los sistemas operativos.





2. Introducción Python

3. Automáticamente reconocerá el sistema operativo que tiene nuestra máquina y habilita la última versión disponible para descargarla.





2. Introducción Python

4. Ejecutar el archivo que se descarga y seguir todos los pasos.
5. Dejar las configuraciones predeterminadas que el instalador sugiere por ahora, con una excepción: mira la casilla de verificación denominada Agregar Python 3.x a PATH y selecciónala.





2. Introducción Python



Comprobación del funcionamiento de Python

Para probar si Python esta funcionando correctamente, es necesario hacer uso de él. Para comenzar el trabajo, es necesario disponer de las siguientes herramientas:

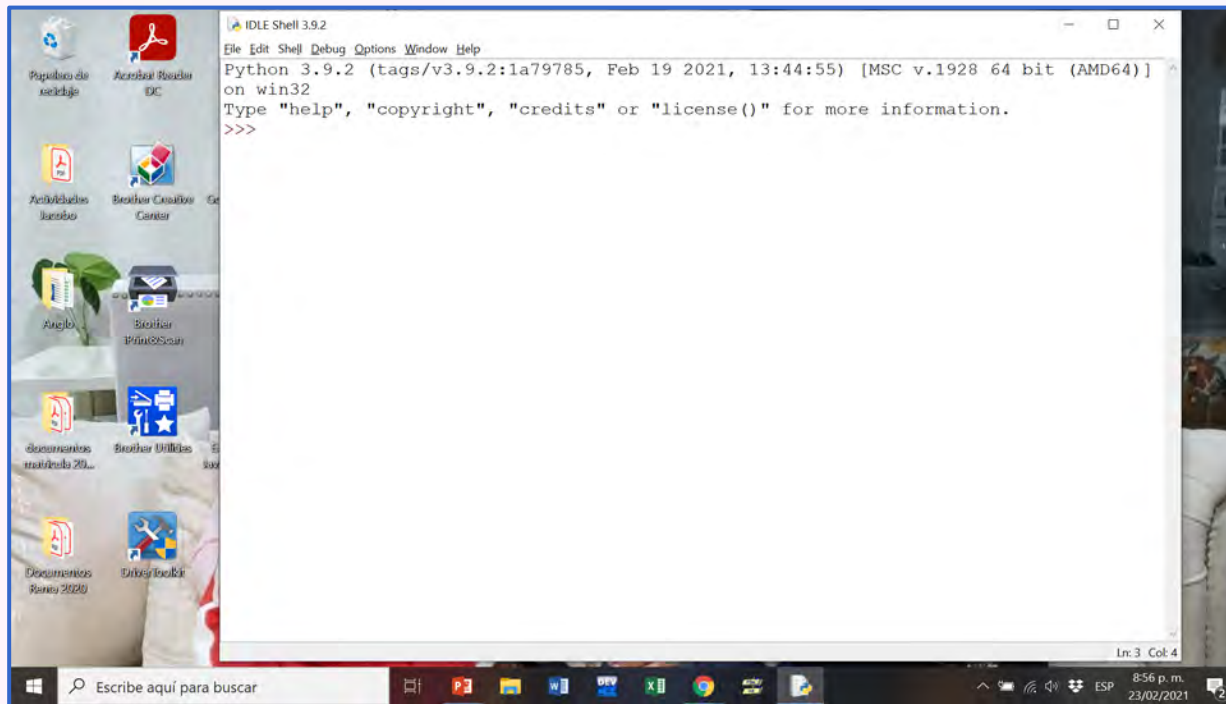
- Un editor que te ayudará a escribir el código.
- Una consola en la que puedes iniciar tu código recién escrito y detenerlo por la fuerza cuando se sale de control.
- Una herramienta llamada depurador, capaz de ejecutar tu código paso a paso.





2. Introducción Python

- Al ejecutarlo se debe visualizar la siguiente ventana





3. Visual Code

IDE

- Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI).





3. Visual Code

Características

- **Editor de código fuente:** editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales, el relleno automático específico del lenguaje y la comprobación de errores a medida que se escribe el código.
- **Automatización de compilación local:** herramientas que automatizan tareas sencillas e iterativas como parte de la creación de una compilación local del software para su uso por parte del desarrollador, como la compilación del código fuente de la computadora en un código binario, el empaquetado del código binario y la ejecución de pruebas automatizadas.
- **Depurador:** programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.





3. Visual Code

¿Por qué los desarrolladores utilizan los IDE?

- Los IDE permiten que los desarrolladores comiencen a programar aplicaciones nuevas con rapidez, ya que no necesitan establecer ni integrar manualmente varias herramientas como parte del proceso de configuración.
- Se encargan de ayudar a los desarrolladores a organizar su flujo de trabajo y solucionar problemas.
- Analizan el código mientras se escribe, así que las fallas causadas por errores humanos se identifican en tiempo real.





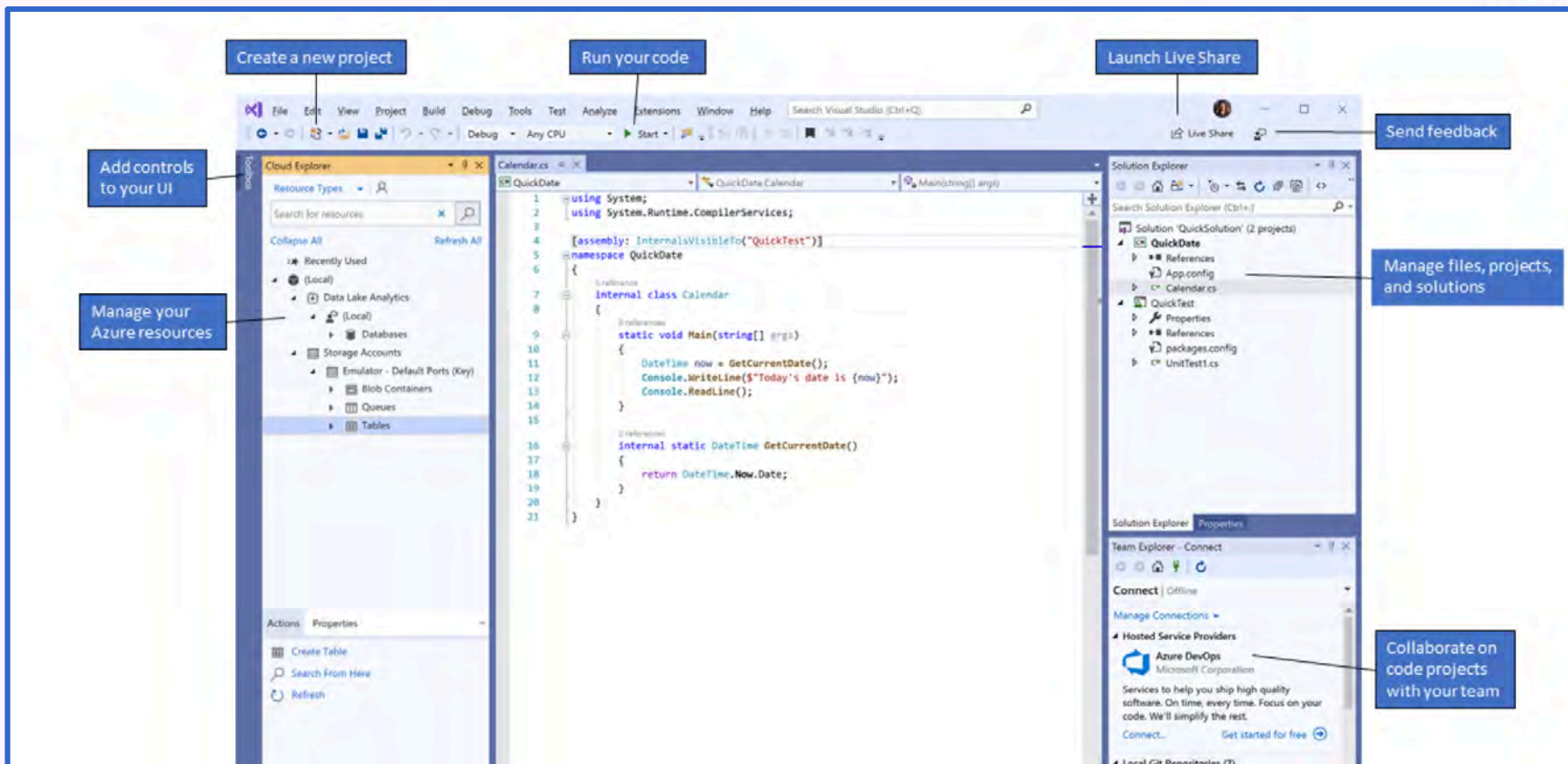
3. Visual Code

El entorno de desarrollo integrado de Visual Studio es una plataforma de lanzamiento creativa que puedes utilizar para editar, depurar y crear código, y luego publicar una aplicación. Un entorno de desarrollo integrado (IDE) es un programa rico en funciones que puede utilizarse para muchos aspectos del desarrollo de software. Además del editor y el depurador estándar que ofrecen la mayoría de los IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más funciones para facilitar el proceso de desarrollo de software.





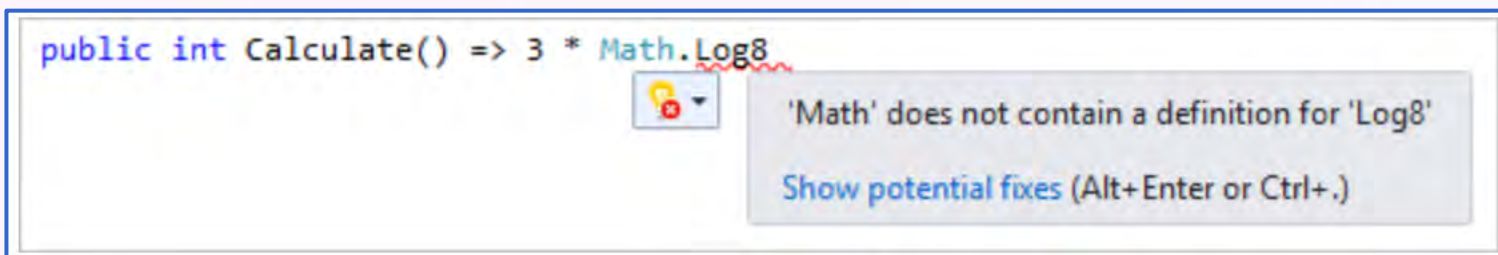
3. Visual Code





3. Visual Code

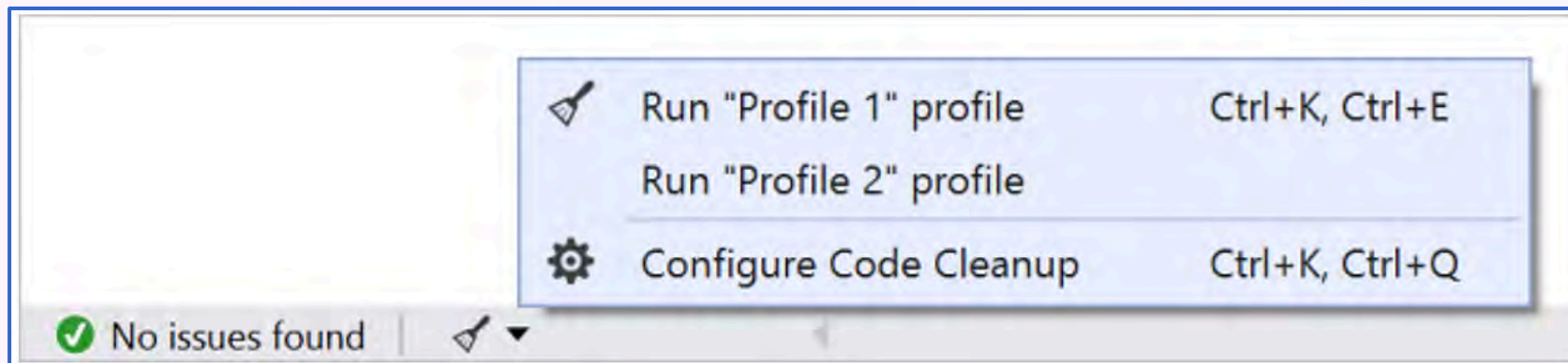
Guiones y acciones rápidas: Los guiones son subrayados ondulados que le avisan de errores o problemas potenciales en su código mientras escribe. Estas pistas visuales le permiten solucionar los problemas inmediatamente sin esperar a que el error se descubra durante la compilación o cuando ejecute el programa. Si pasa el ratón por encima de un guion, verá información adicional sobre el error. También puede aparecer una bombilla en el margen izquierdo con acciones, conocidas como Acciones Rápidas, para solucionar el error.





3. Visual Code

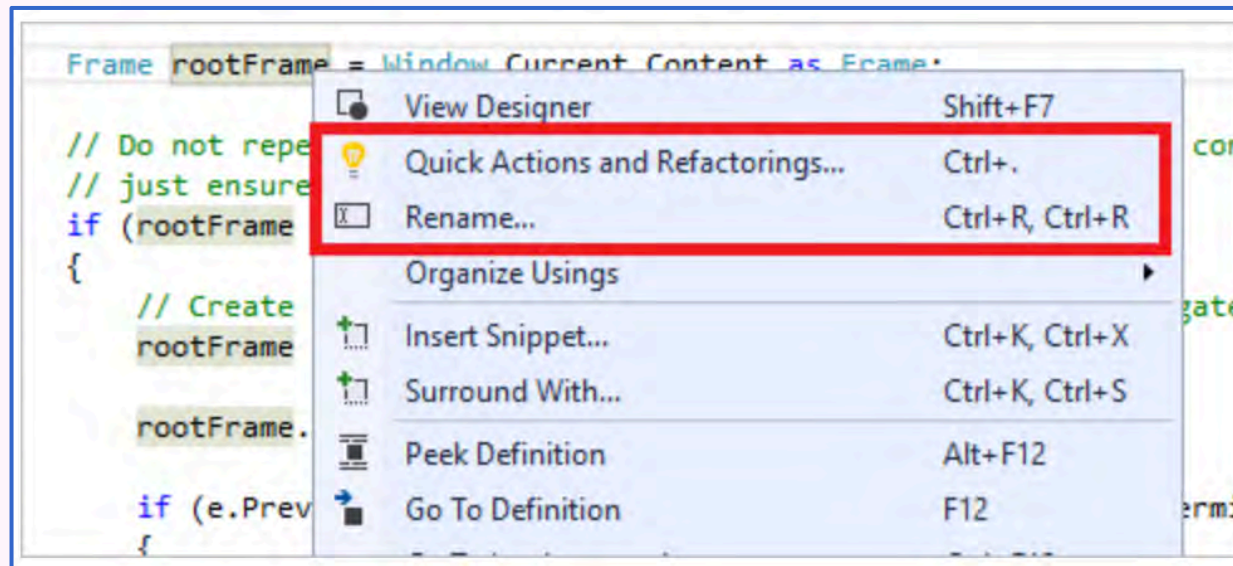
Limpieza de código: Con el clic de un botón, formatee el código y aplique las correcciones de código sugeridas por su configuración de estilo de código, las convenciones de `.editorconfig` y los analizadores de *Roslyn*. La limpieza de código le ayuda a resolver los problemas de su código antes de que pase a la revisión del mismo. (Actualmente sólo está disponible para código C#).





3. Visual Code

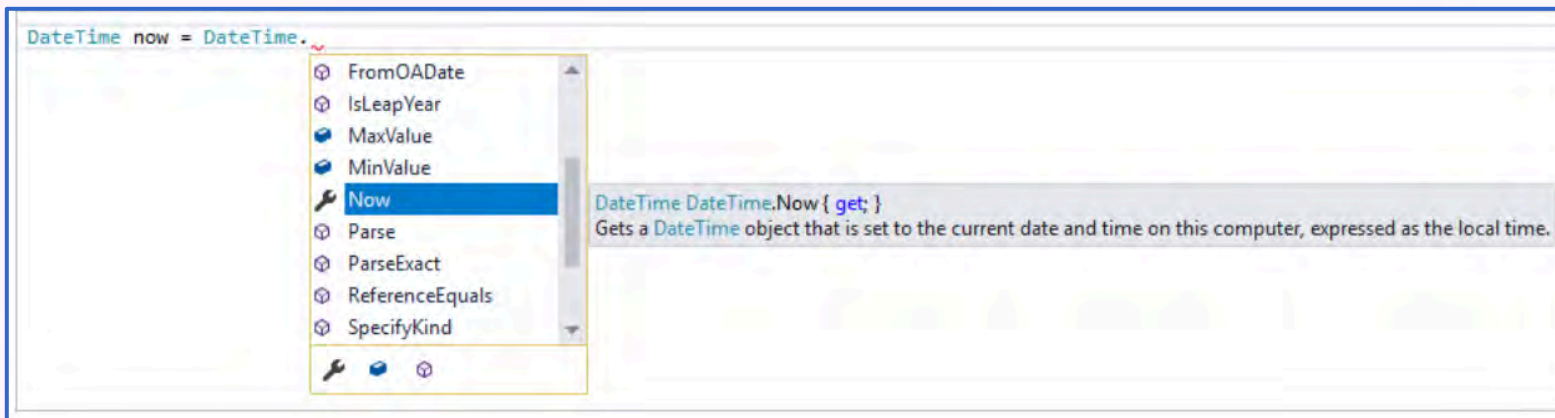
Refactorización: La refactorización incluye operaciones como el cambio inteligente de nombre de las variables, la extracción de una o varias líneas de código en un nuevo método, el cambio del orden de los parámetros del método, etc.





3. Visual Code

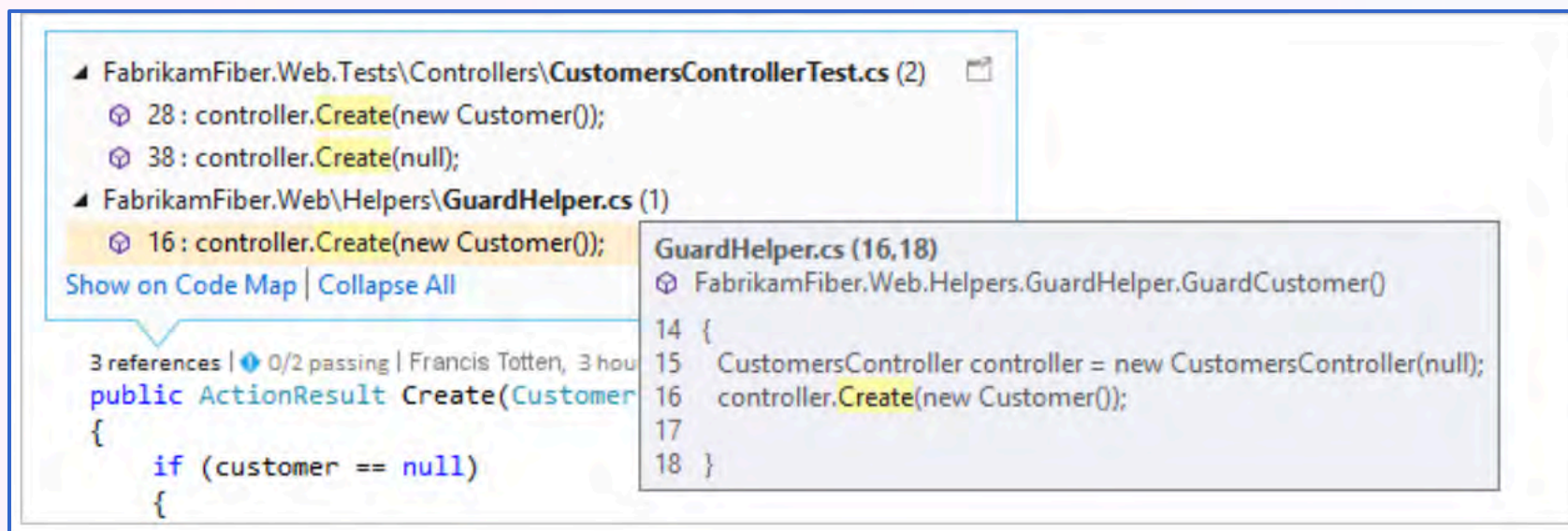
IntelliSense: IntelliSense es un término para un conjunto de características que muestra información sobre tu código directamente en el editor y, en algunos casos, escribe pequeños fragmentos de código por ti. Es como tener una documentación básica en línea en el editor, lo que te ahorra tener que buscar información de tipo en otro lugar. Las características de IntelliSense varían según el lenguaje. La siguiente ilustración muestra cómo IntelliSense muestra la lista de miembros de un tipo:





3. Visual Code

CodeLens: le ayuda a encontrar referencias a su código, cambios en su código, errores vinculados, elementos de trabajo, revisiones de código y pruebas unitarias, todo ello sin salir del editor.





3. Visual Code

➤ Python:

- Edición, depuración, desarrollo interactivo y control de código fuente para Python.
- Compatibilidad con la compilación de aplicaciones web de Python mediante marcos conocidos, como Django y Flask, y aplicaciones de ciencia de datos que admiten Conda y IPython.

➤ Aplicaciones de desarrollo:

- Aplicaciones de Python
- Aplicaciones web de Django
- Aplicaciones web de Flask





3. Visual Code

Componentes:

- Compatibilidad con entornos CPython, virtualenv, venv y conda
- Edición avanzada, IntelliSense y comprensión del código
- Ventana del entorno interactiva (REPL)
- Compatibilidad con Python en áreas de trabajo de carpetas abiertas
- Plantillas de proyecto y elemento para simplificar el proceso de creación de nuevos proyectos
- Generación de perfiles y herramientas de pruebas unitarias
- Administración de entornos Conda con Miniconda
- Live Share





Preguntas

