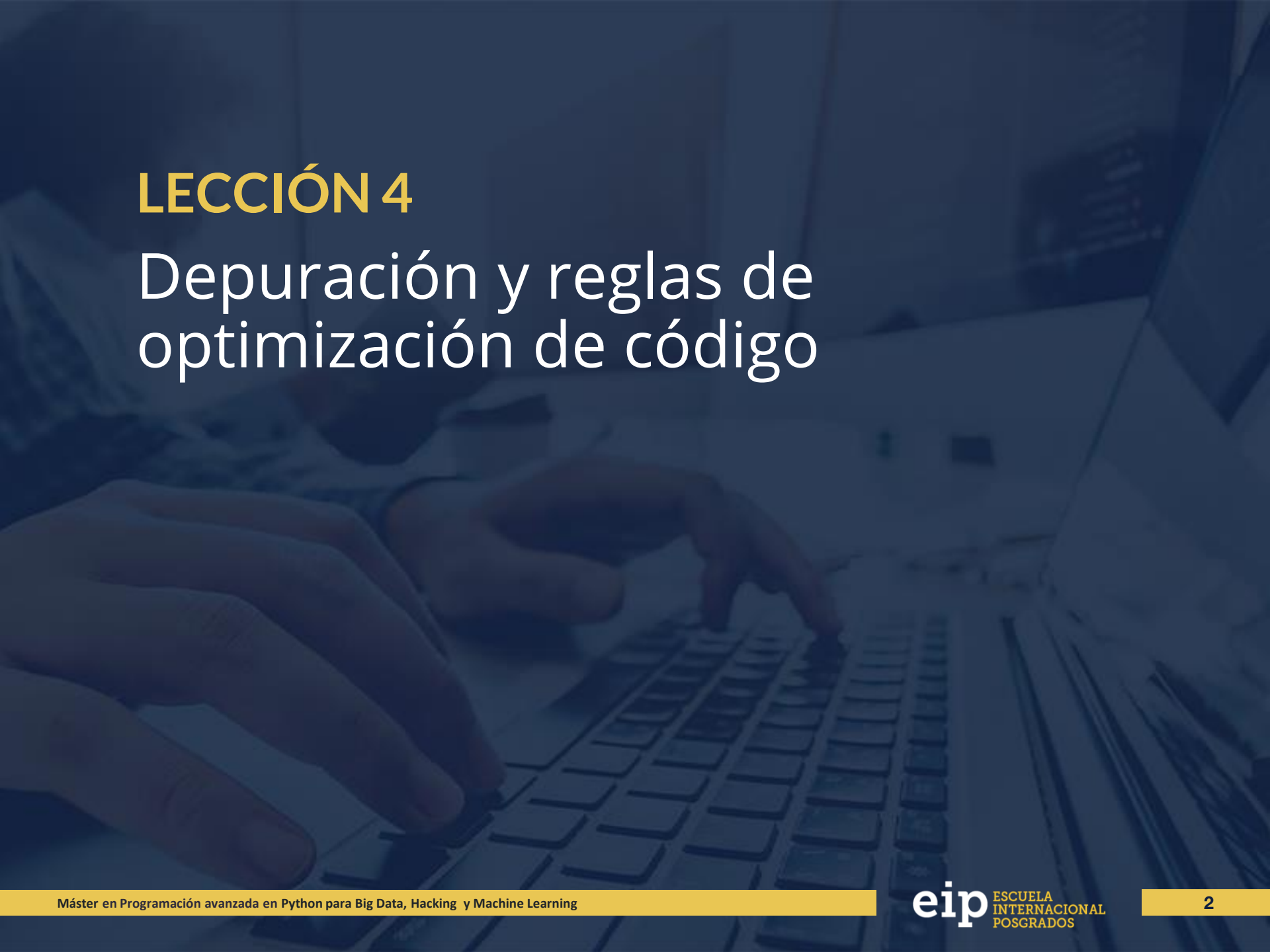
eip ESCUELA INTERNACIONAL POSGRADOS

Máster Avanzado de Programación en Python para Hacking, BigData y Machine Learning

BUENAS PRÁCTICAS DE PROGRAMACIÓN EN PYTHON



LECCIÓN 4

Depuración y reglas de optimización de código

ÍNDICE

INTRODUCCIÓN

En esta lección aprenderemos a utilizar el depurador de código nativo de Python pdb. Adicionalmente, estudiaremos una serie de funciones que nos permitirán optimizar nuestros códigos implementados en Python. Estas funciones o reglas no solo conseguirán optimizar nuestro código, sino que aumentarán la interpretabilidad y elegancia de nuestros programas.

OBJETIVOS

Al finalizar esta lección serás capaz de:

1. Comprender qué es el depurador de Python (pdb)
2. Estudiar cómo el uso del depurador pdb nos permite ejecutar nuestras implementaciones línea a línea para identificar y solucionar posibles errores.
3. Comprender qué es y cuál es la ventaja de integrar comprensión de listas en nuestras implementaciones.
4. Comprender el potencial de las funciones *map*, *filter* y *reduce*.

DEPURACIÓN DE CÓDIGO

Depurar un programa es el proceso a través del cual identificamos y corregimos errores de programación. En inglés, a este proceso se le denomina como debugging porque se asemeja al acto de eliminar bugs, manera en la que se conocen a los errores de programación.

Para depurar un programa, se hace uso de un depurador. En Python, el depurador por excelencia es el pdb (Python DeBugger).

DEPURACIÓN DE CÓDIGO

Con la herramienta pdb es posible insertar puntos de parada dentro del código fuente sin necesidad de usar un IDE o herramientas externas.

Como cualquier otro depurador, pdb está dotado de las capacidades necesarias para poder imprimir el valor de las variables en un determinado punto de la ejecución, ejecutar y evaluar el código línea a línea e incluso examinar el funcionamiento de las funciones de nuestro código.

DEPURACIÓN DE CÓDIGO

A continuación se muestra una lista del conjunto de comandos más utilizados con este depurador:

- Crear puntos de parada (breakpoints)
- Reanudación de la ejecución (continue).
- Mostrar código adyacente
- Avanzar línea a línea
- Inspector de variables
- Eliminar puntos de parada (breakpoints)

DEPURACIÓN DE CÓDIGO

Veamos un ejemplo práctico...

TÉCNICAS DE OPTIMIZACIÓN DE CÓDIGO

LISTAS DE COMPRESIÓN (veamos un ejemplo detallado)

TÉCNICAS DE OPTIMIZACIÓN DE CÓDIGO

FUNCIONES *MAP*, *FILTER*, *REDUCE* (veamos un ejemplo detallado)

CONCLUSIONES

1. El depurador de Python pdb es una herramienta fundamental que debe conocer todo programador de Python para identificar y solucionar posibles problemas en nuestras implementaciones.

2. La comprensión de listas es una forma de recorrer y operar con listas más eficientes que los métodos tradicionales estudiados hasta ahora.

3. El uso de las funciones *map*, *reduce* y *filter* nos permite operar sobre listas de una forma más eficiente y elegante.

MUCHAS GRACIAS POR SU ATENCIÓN

eip ESCUELA INTERNACIONAL POSGRADOS


rrueda@grupomainjobs.com


Ramón Rueda Delgado
https://www.linkedin.com/in/ramon-rueda/


twitter.com/eiposgrados


facebook.com/eiposgrados


instagram.com/eiposgrados