

Maestría en Ciencia de Datos (MCD)

Programación II



Unidad 1 Ciencia de datos con python

Programación II

Objetivo particular de la unidad: Conocer acerca de bibliotecas de relacionadas con el análisis de datos en Python para crear software de manera mas rápida y eficiente

1. Bibliotecas de ciencia de datos para Python

1.1 Ciencia de datos

1.2 Biblioteca NumPy

1.3 Biblioteca Pandas

1.4 Biblioteca rpy2

1.5 Biblioteca Scikit-learn

1.6 Distribución Anaconda

1.7 Visual Studio Code



ANACONDA®



scikit-learn



rpy2



seaborn



Visual Studio Code



matplotlib



SciPy



Ciencia de Datos

Programación II

“La ciencia de datos es el estudio de datos con el fin de extraer información significativa para empresas. Es un enfoque multidisciplinario que combina principios y prácticas del campo de las matemáticas, la estadística, la inteligencia artificial y la ingeniería de computación para analizar grandes cantidades de datos.”

(AWS,2023)



Ciencia de Datos

Programación II

¿Por qué es importante la ciencia de datos?

La ciencia de datos es importante porque combina herramientas, métodos y tecnología para generar significado a partir de los datos.

(AWS, 2023)



Ciencia de Datos

Programación II

Futuro de la ciencia de datos

La inteligencia artificial y las innovaciones del machine learning han hecho que el procesamiento de datos sea más rápido y eficiente. La demanda del sector ha creado un ecosistema de cursos, grados académicos y puestos de trabajo en el campo de la ciencia de datos.

(AWS,2023)



Ciencia de Datos

Programación II

¿Para qué se utiliza la ciencia de datos?

La ciencia de datos se utiliza para estudiar los datos de cuatro maneras principales:

1. Análisis descriptivo

El análisis descriptivo examina los datos para obtener información sobre lo que ha ocurrido u ocurre en el entorno de datos. Se caracteriza por las visualizaciones de datos, como los gráficos circulares, de barras o líneas, las tablas o las narraciones generadas.

(AWS,2023)



Ciencia de Datos

Programación II

2. Análisis de diagnóstico

El análisis de diagnóstico es un examen profundo o detallado de datos para entender por qué ha ocurrido algo. Se caracteriza por técnicas como el análisis detallado, el descubrimiento y la minería de datos o las correlaciones. Se pueden llevar a cabo varias operaciones y transformaciones de datos en un determinado conjunto con el fin de descubrir patrones únicos en cada una de estas técnicas.

(AWS,2023)



Ciencia de Datos

Programación II

3. Análisis predictivo

El análisis predictivo utiliza los datos históricos para hacer previsiones precisas sobre los patrones de datos que pueden producirse en el futuro. Se caracteriza por técnicas como el machine learning, la previsión, la coincidencia de patrones y el modelado predictivo

(AWS,2023)



Ciencia de Datos

Programación II

3. Análisis predictivo

El análisis predictivo utiliza los datos históricos para hacer previsiones precisas sobre los patrones de datos que pueden producirse en el futuro. Se caracteriza por técnicas como el machine learning, la previsión, la coincidencia de patrones y el modelado predictivo

(AWS,2023)



Ciencia de Datos

Programación II

4. Análisis prescriptivo

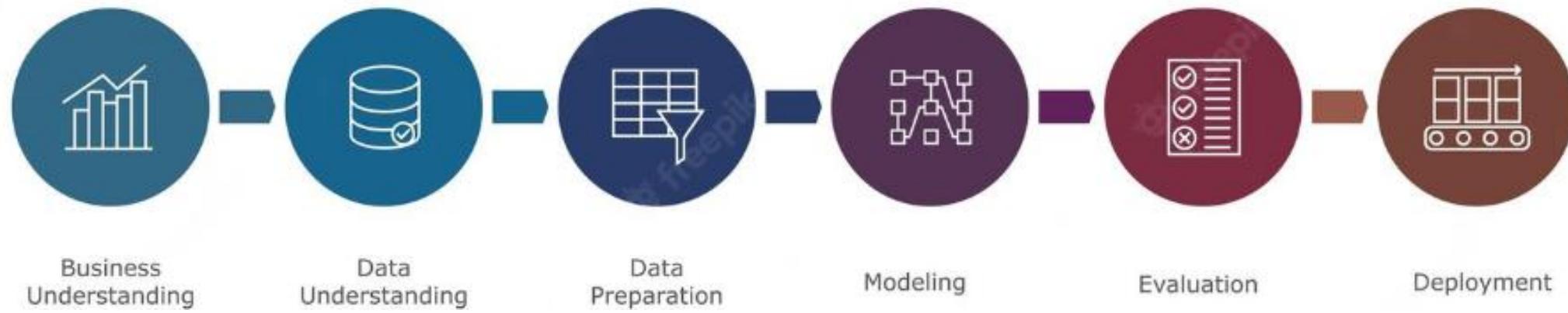
El análisis prescriptivo lleva los datos predictivos al siguiente nivel. No solo predice lo que es probable que ocurra, sino que sugiere una respuesta óptima para ese resultado. Puede analizar las posibles implicaciones de las diferentes alternativas y recomendar el mejor curso de acción.

(AWS,2023)



Ciencia de Datos

Programación II



Anaconda

Programación II

¿Qué es el Navegador Anaconda?

Anaconda Navigator es una interfaz gráfica de usuario (GUI) de escritorio incluida en la distribución de Anaconda® que le permite iniciar aplicaciones y administrar fácilmente paquetes, entornos y canales de conda sin usar comandos de línea de comandos. Navigator puede buscar paquetes en Anaconda.org o en un repositorio local de Anaconda. Está disponible para Windows, macOS y Linux.

(Anaconda, 2023)



ANACONDA®



Anaconda

Programación II

¿A qué aplicaciones puedo acceder usando Navigator?

Las siguientes aplicaciones están disponibles de forma predeterminada en Navigator:

- [JupyterLab](#)
- [Cuaderno Jupyter](#)
- [espía](#)
- [PyCharm](#)
- [Glueviz](#)
- [Aplicación naranja 3](#)
- [RStudio](#)
- Aviso de Anaconda (solo Windows)
- Anaconda PowerShell (solo Windows)



Anaconda

Programación II

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 621 MB

[Get Additional Installers](#)



<https://www.anaconda.com/>



Visual Studio Code

Programación II

ANACONDA.NAVIGATOR

All applications on base (root) Channels

 DataSpell	 CMD.exe Prompt 0.1.1	 JupyterLab 3.5.2	 Notebook 6.5.2	 PowerShell Prompt 0.0.1	 Qt Console 5.2.2
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.	Run a cmd.exe terminal with your current environment from Navigator activated	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Run a PowerShell terminal with your current environment from Navigator activated	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Install	Launch	Launch	Launch	Launch	Launch
 Spyder 5.2.2	 VS Code 1.74.3	 Datalore	 Deepnote	 IBM Watson Studio Cloud	 Oracle Cloud Infrastructure
Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features	Streamlined code editor with support for development operations like debugging, task running and version control.	Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.	Deepnote is a new kind of data notebook build. For collaboration - Jupiter compatible, in the cloud and sharing is easy as sending a link.	IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.	OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools
Launch	Launch	Launch	Launch	Launch	Launch
 console_shortcut_miniconda 0.1.1	 Glueviz 1.2.4	 Orange 3 3.32.0	 powershell_shortcut_miniconda 0.0.1	 PyCharm Professional	 RStudio 1.1.456
Multidimensional data visualization across files. Explore relationships within and	Component based data mining framework. Data visualization and data analysis for	A full-fledged IDE by JetBrains for both Scientific and Web Python development.	A set of integrated tools designed to help you be more productive with R. Includes R		

Documentation [Learn More](#)

Anaconda Notebooks Cloud notebooks with hundreds of packages ready to code.

Twitter YouTube GitHub



Visual Studio Code

Programación II

Qué es Visual Studio Code?



Visual Studio Code

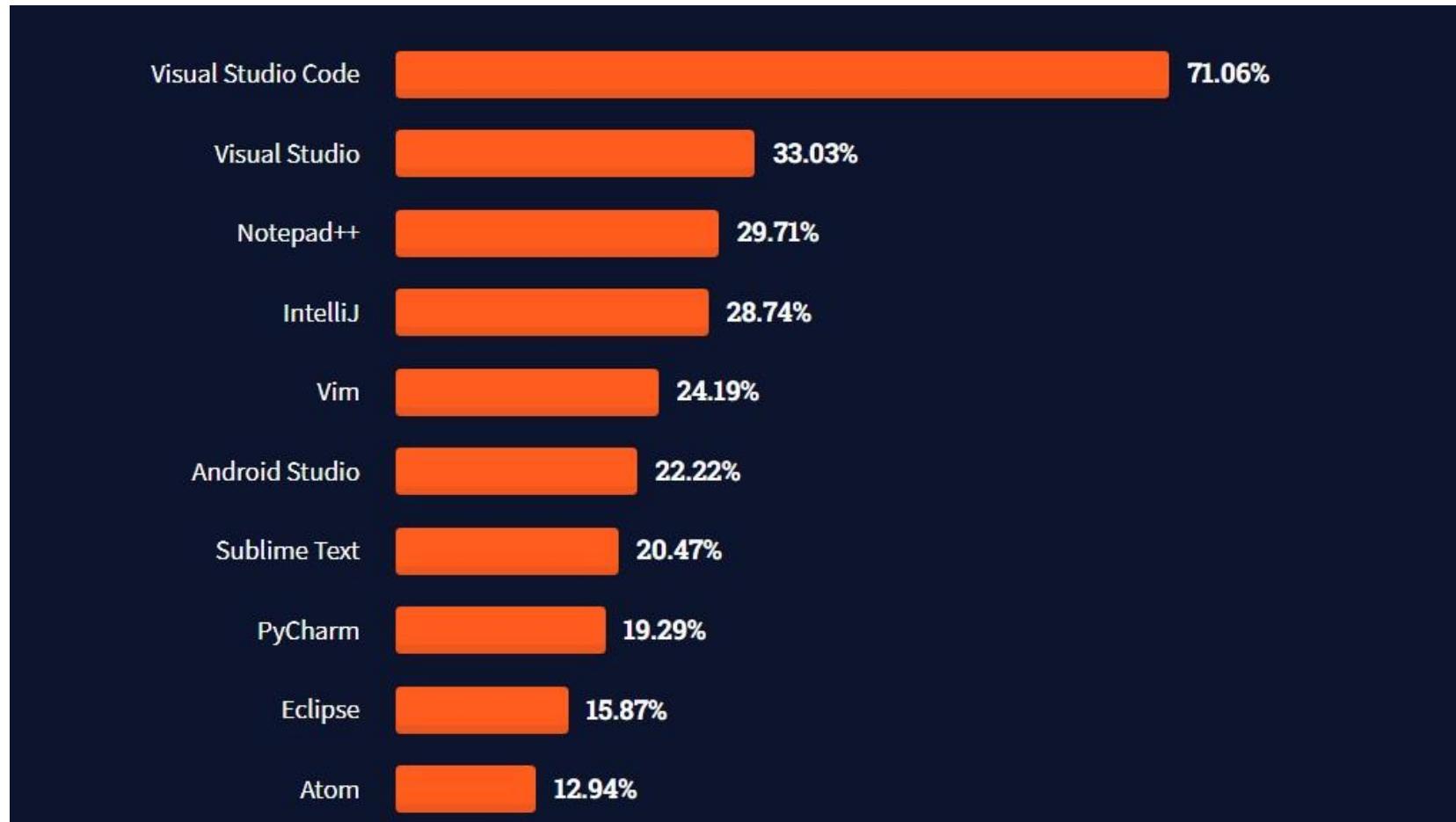
Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación

(OpenWebinars, 2023)



Visual Studio Code

Programación II

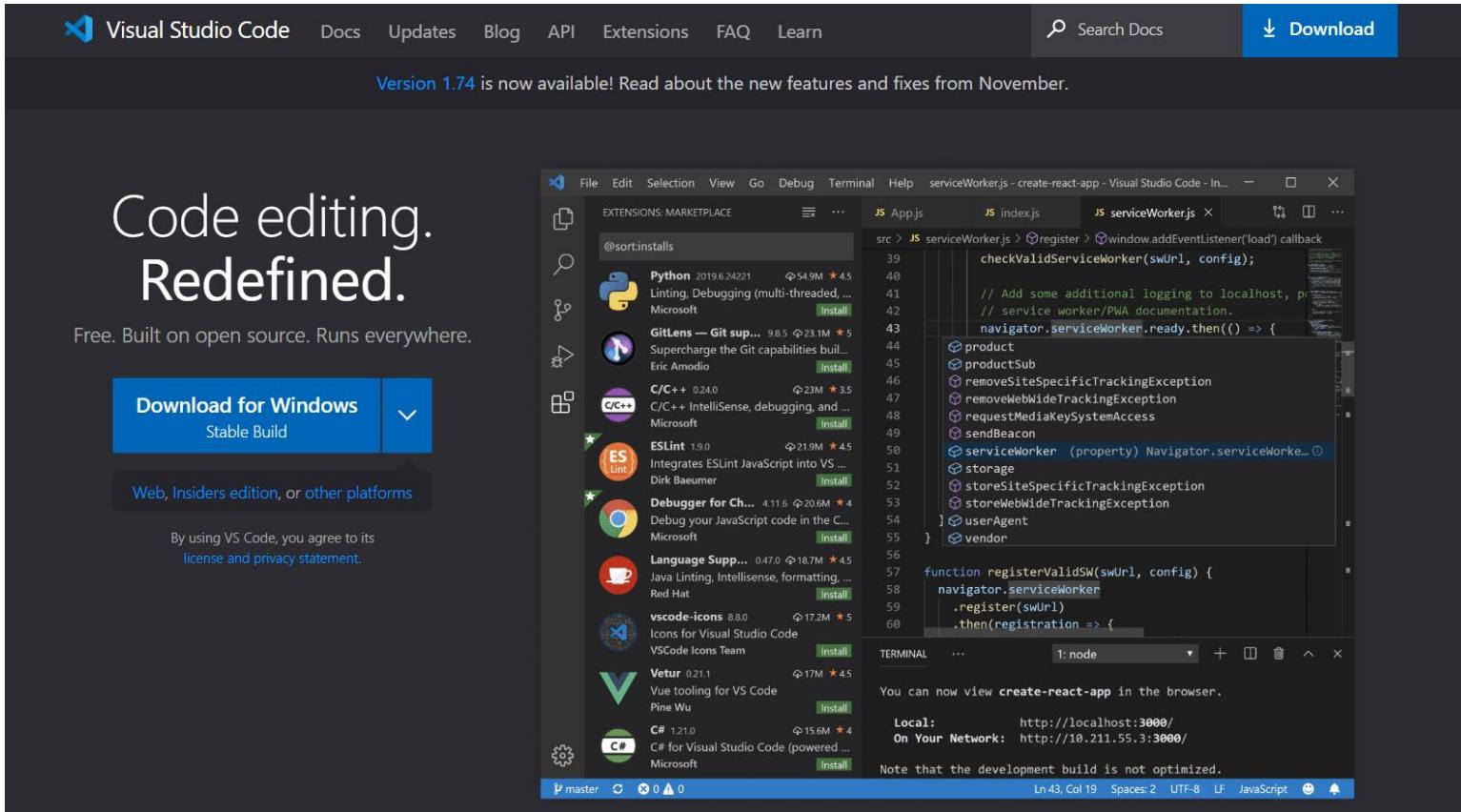


(OpenWebinars, 2023)



Visual Studio Code

Programación II



The image shows the official website for Visual Studio Code and a screenshot of the application itself.

Visual Studio Code Website:

- Header:** Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, Learn, Search Docs, Download.
- Text:** Version 1.74 is now available! Read about the new features and fixes from November.
- Section:** Code editing. Redefined. Free. Built on open source. Runs everywhere.
- Download Buttons:** Download for Windows (Stable Build), Web, Insiders edition, or other platforms.
- License Information:** By using VS Code, you agree to its license and privacy statement.

Visual Studio Code Interface:

- Left Sidebar:** EXTENSIONS: MARKETPLACE, showing installed extensions like Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support, vscode-icons, Vetur, and C#.
- Code Editor:** Three files are open: serviceWorker.js, App.js, and index.js. The code editor shows a snippet of JavaScript related to service workers.
- Terminal:** Shows the command "node" and the message "You can now view create-react-app in the browser."
- Status Bar:** master, 0 0 0, Ln 43, Col 19, Spaces:2, UTF-8, LF, JavaScript, Note that the development build is not optimized.

<https://code.visualstudio.com/>

Visual Studio Code

Programación II

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 8, 10, 11

User Installer

x64 x86 Arm64

System Installer

x64 x86 Arm64

.zip

x64 x86 Arm64

CLI

x64 x86 Arm64



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE



↓ Mac

macOS 10.11+

.zip Intel chip Apple silicon Universal

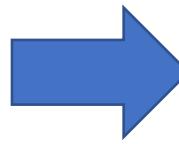
CLI Intel chip Apple silicon

<https://code.visualstudio.com/>



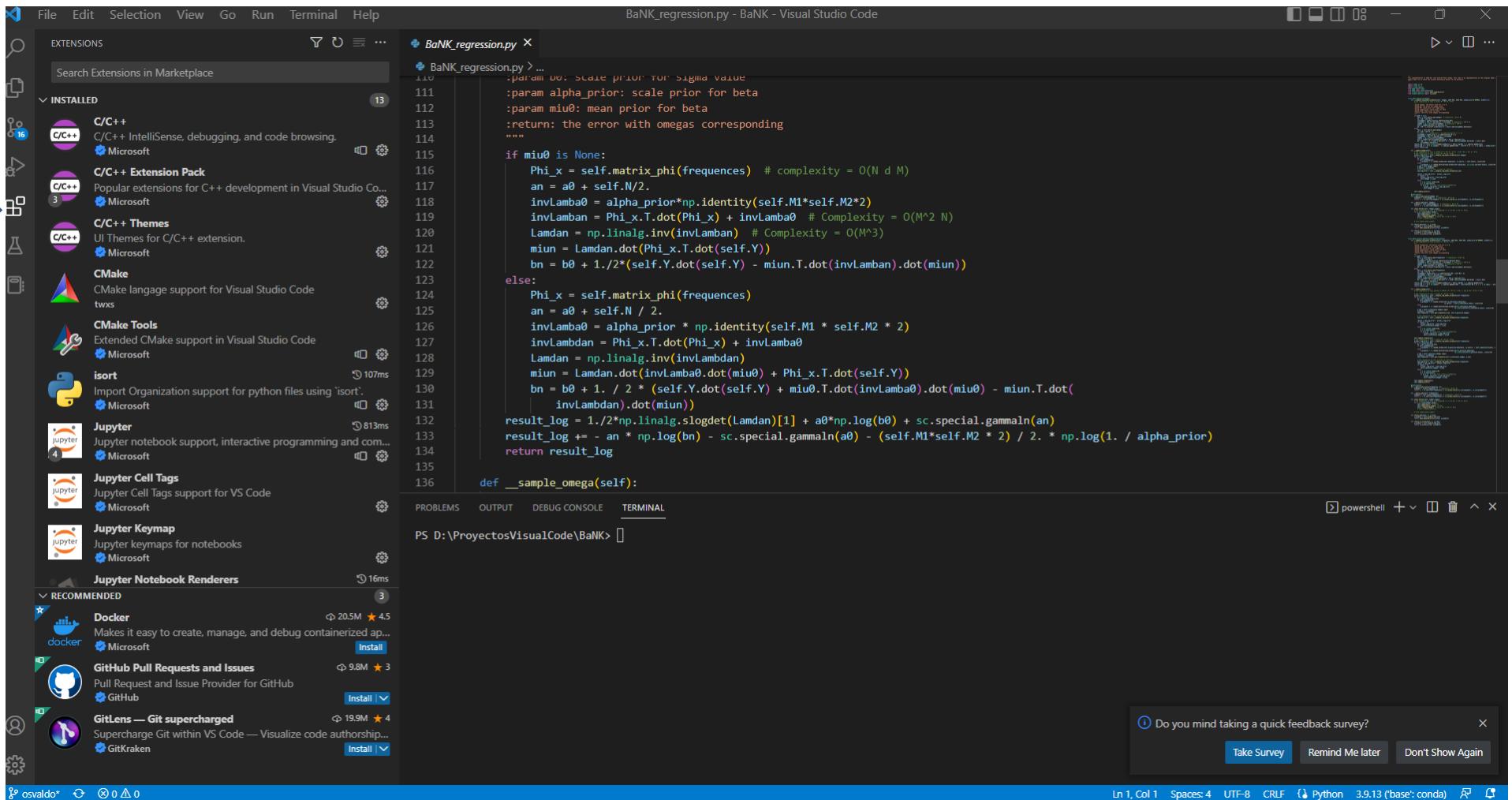
Visual Studio Code

Programación II



Actividades

- Instalar Python
- Instalar Jupyter
- Crear carpeta
- Abrir carpeta



BaNK_regression.py - BaNK - Visual Studio Code

```

110     :param do: scale prior for sigma value
111     :param alpha_prior: scale prior for beta
112     :param miu0: mean prior for beta
113     :return: the error with omegas corresponding
114     """
115
116     if miu0 is None:
117         Phi_x = self.matrix_phi(frecuencias) # complexity = O(N d M)
118         an = a0 + self.N/2.
119         invLamba0 = alpha_prior*np.identity(self.M1*self.M2*2)
120         invLamban = Phi_x.T.dot(Phi_x) + invLamba0 # Complexity = O(M^2 N)
121         Lamdan = np.linalg.inv(invLamban) # Complexity = O(M^3)
122         miun = Lamdan.dot(Phi_x.T.dot(self.Y))
123         bn = b0 + 1./2*(self.Y.dot(self.Y) - miun.T.dot(invLamban).dot(miun))
124     else:
125         Phi_x = self.matrix_phi(frecuencias)
126         an = a0 + self.N / 2.
127         invLamba0 = alpha_prior * np.identity(self.M1 * self.M2 * 2)
128         invLamban = Phi_x.T.dot(Phi_x) + invLamba0
129         Lamdan = np.linalg.inv(invLamban)
130         miun = Lamdan.dot(invLamba0.dot(miu0) + Phi_x.T.dot(self.Y))
131         bn = b0 + 1. / 2 * (self.Y.dot(self.Y) + miu0.T.dot(invLamba0).dot(miu0) - miun.T.dot(
132             invLamban).dot(miun))
133         result_log = 1./2*np.linalg.slogdet(Lamdan)[1] + a0*np.log(b0) + sc.special.gammaln(an)
134         result_log += - an * np.log(bn) - sc.special.gammaln(a0) - (self.M1*self.M2 * 2) / 2. * np.log(1. / alpha_prior)
135     return result_log
136
137     def __sample_omega(self):

```

File Edit Selection View Go Run Terminal Help

EXTENSIONS

Search Extensions in Marketplace

INSTALLED

- C/C++
- C/C++ Extension Pack
- C/C++ Themes
- CMake
- CMake Tools
- isort
- Jupyter
- Jupyter Cell Tags
- Jupyter Keymap
- Jupyter Notebook Renderers

RECOMMENDED

- Docker
- GitHub Pull Requests and Issues
- GitLens — Git supercharged

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS D:\ProyectosVisualCode\BaNK>

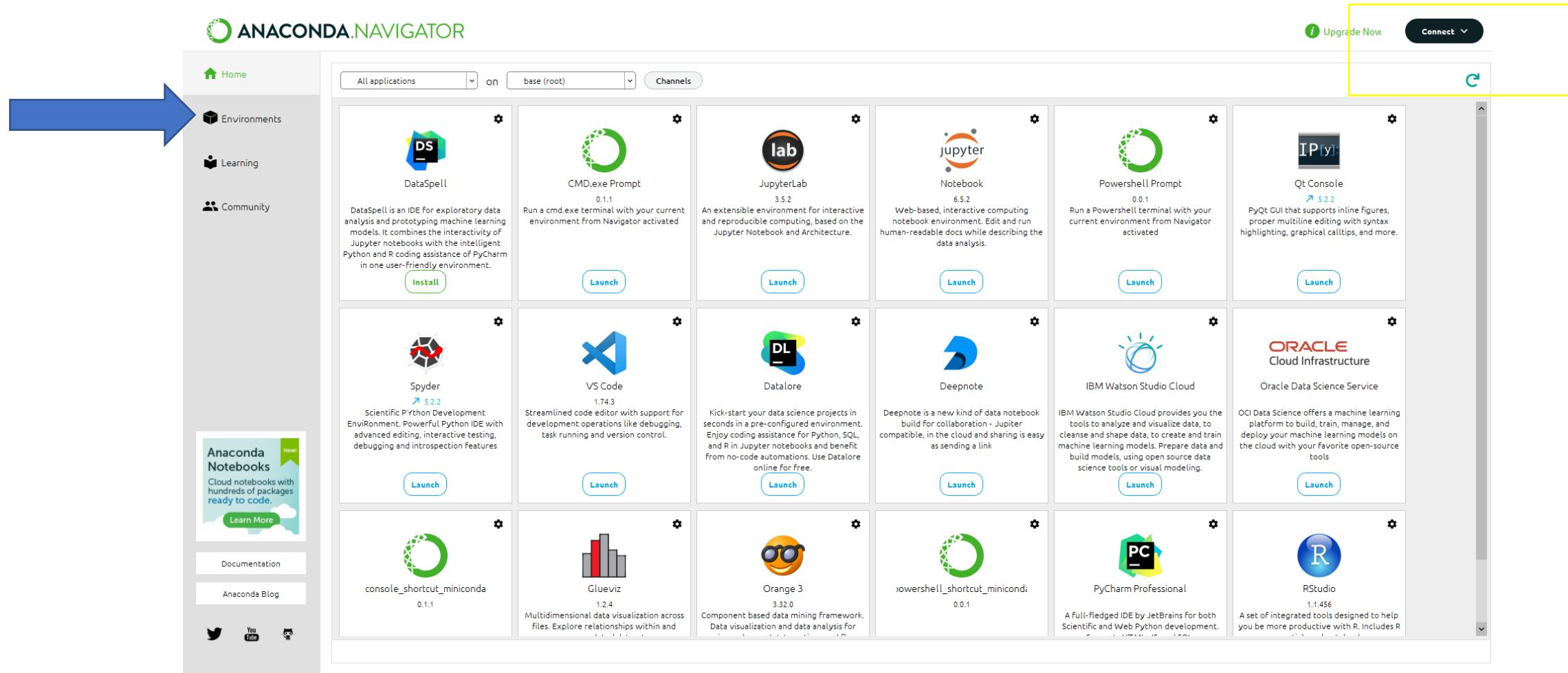
Do you mind taking a quick feedback survey?

osvaldo* 0 0 △ 0 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF (Python 3.9.13 ('base': conda) ⚡ ↗ ↘

Environments

Programación II

ANACONDA.NAVIGATOR



The screenshot shows the Anaconda Navigator interface. On the left, there's a sidebar with links for Home, Environments, Learning, and Community. A prominent blue arrow points from the left towards the main content area. The main area displays a grid of application cards. Some cards have a yellow border around them. The cards include:

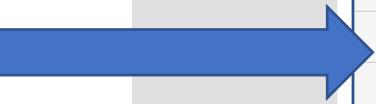
- DataSpell**: An IDE for exploratory data analysis and prototyping machine learning models.
- CMD.exe Prompt**: A cmd.exe terminal with your current environment from Navigator activated.
- JupyterLab**: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
- Notebook**: A web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
- Powershell Prompt**: A Powershell terminal with your current environment from Navigator activated.
- Qt Console**: A PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
- Spyder**: A scientific Python Development Environment. It's a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
- VS Code**: A streamlined code editor with support for development operations like debugging, task running and version control.
- Datalore**: A platform to kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.
- Deepnote**: A new kind of data notebook build for collaboration - Jupiter compatible, in the cloud and sharing is easy as sending a link.
- IBM Watson Studio Cloud**: IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
- ORACLE Cloud Infrastructure**: Oracle Data Science Service. OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools.
- console_shortcut_miniconda**: A miniconda environment for console users.
- Glueviz**: A multidimensional data visualization tool.
- Orange 3**: A component-based data mining framework for data visualization and data analysis.
- powershell_shortcut_miniconda**: A miniconda environment for Powershell users.
- PyCharm Professional**: A full-fledged IDE by JetBrains for both Scientific and Web Python development.
- RStudio**: A set of integrated tools designed to help you be more productive with R. Includes RStudio Server Pro.

At the bottom of the sidebar, there's a section for **Anaconda Notebooks** with a "Learn More" button, and links for Documentation and Anaconda Blog. Social media icons for Twitter, YouTube, and GitHub are also present.

Environments

Programación II

Environments



ANACONDA.NAVIGATOR

File Help

Upgrade Now Connect

Search Environments

Installed Channels Update index... Search Packages

base (root)

bank

computervision

gwp

gwp_tensor

gwp_tf_mlflow

latam

medicalimages

mlflow

objectdete

Anaconda Notebooks

Documentation

Anaconda Blog

Create Clone Import Backup Remove

Name Description Version

_ipyw_jlab_nb_ex... 0.1.0

alabaster 0.7.12

anaconda 2022.10

anaconda-client 1.11.0

anaconda-project 0.11.1

anyio 3.5.0

appdirs 1.4.4

argon2-cffi 21.3.0

argon2-cffi-bindings 21.2.0

arrow 1.2.3

astroid 2.11.7

astropy 5.1

atomicwrites 1.4.0

attrs 22.1.0

automat 20.2.0

autopep8 1.6.0

babel 2.11.0

backcall 0.2.0

Dependencias instaladas



Environments

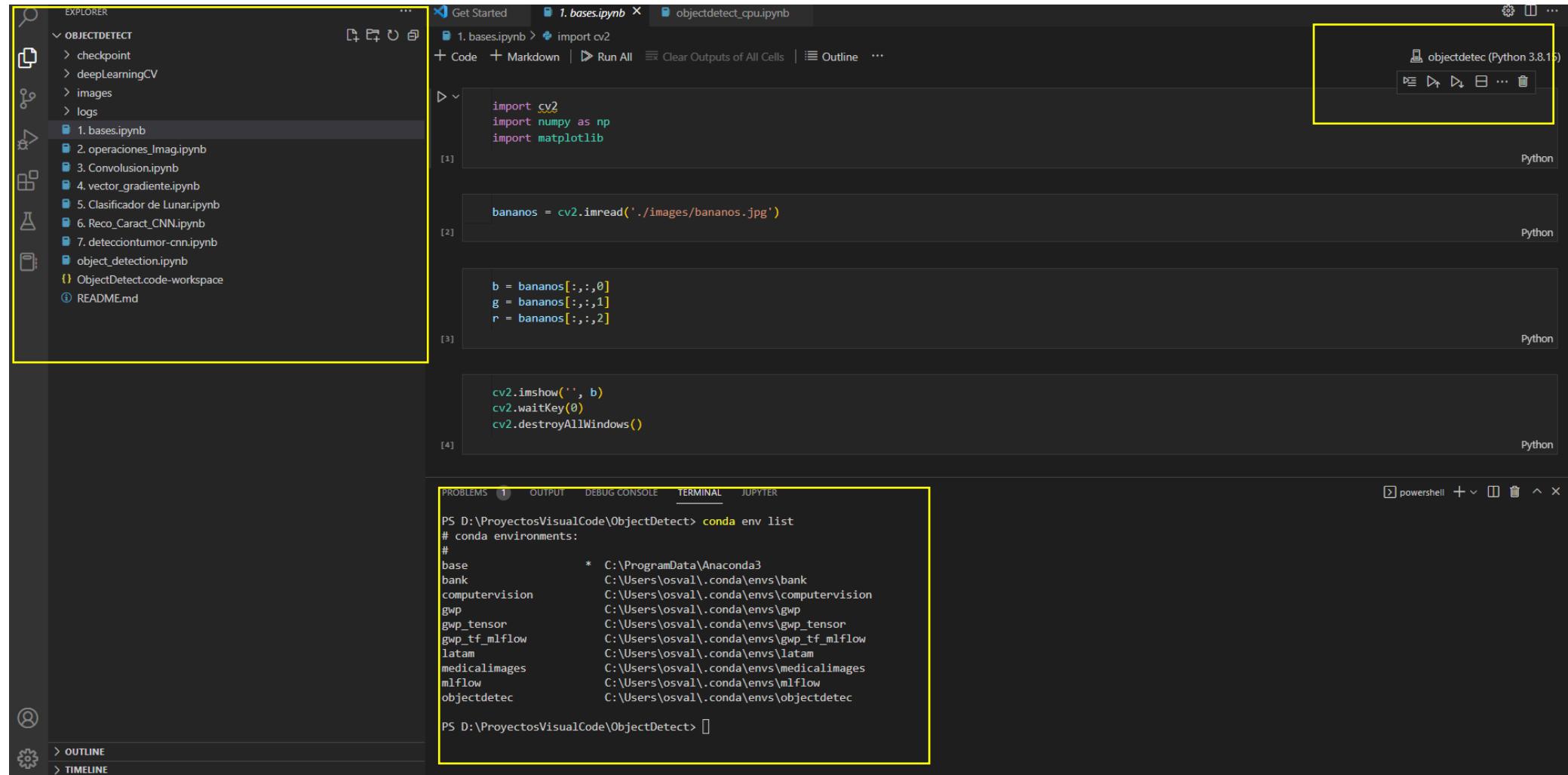
Programación II

- *conda create -n nombreenv python=3.x*
- *conda remove -n nombreenv -all*
- *conda env list*
- *conda activate nombreenv*
- *conda deactivate*



Environments (Problema común)

Programación II

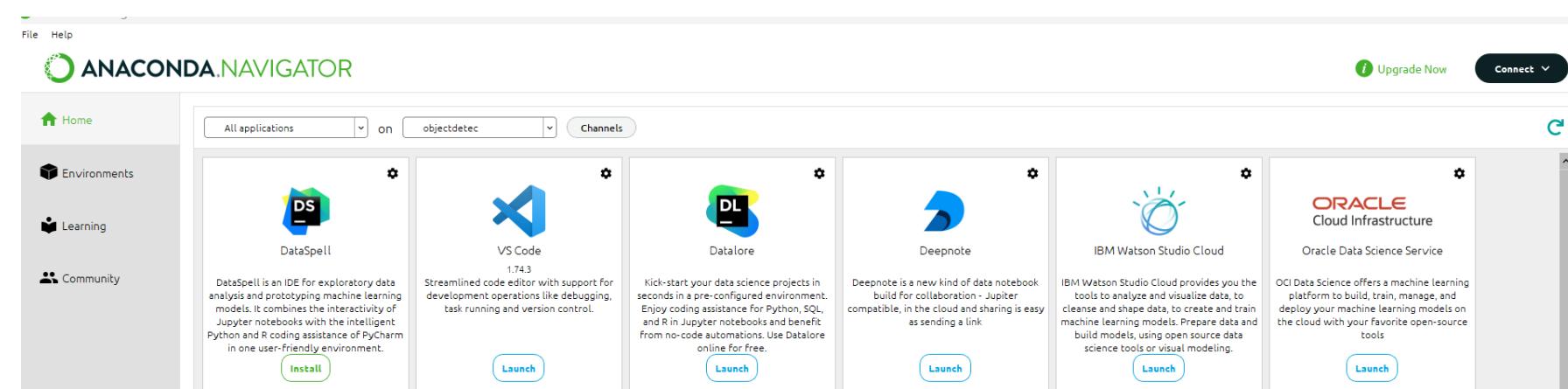
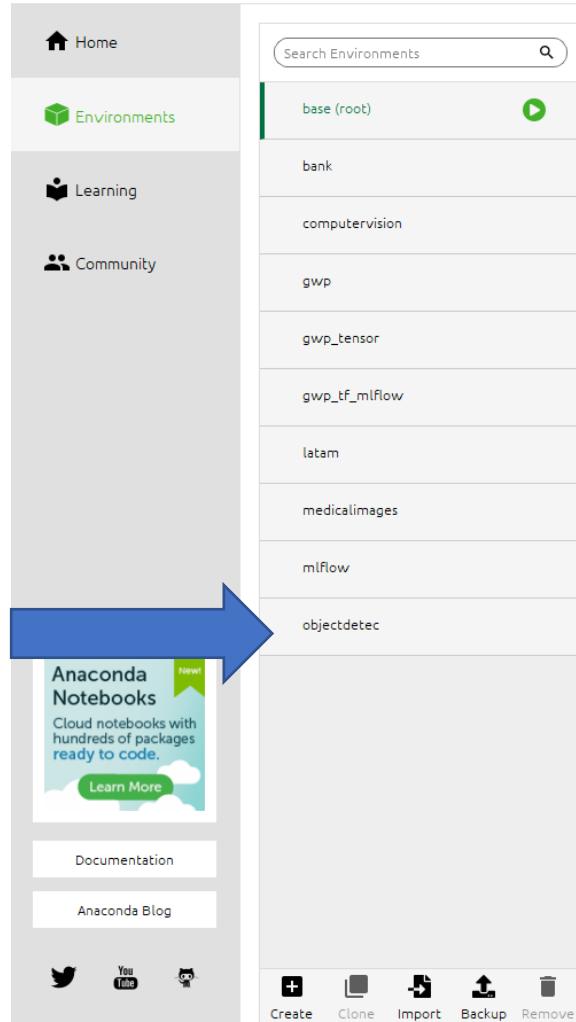
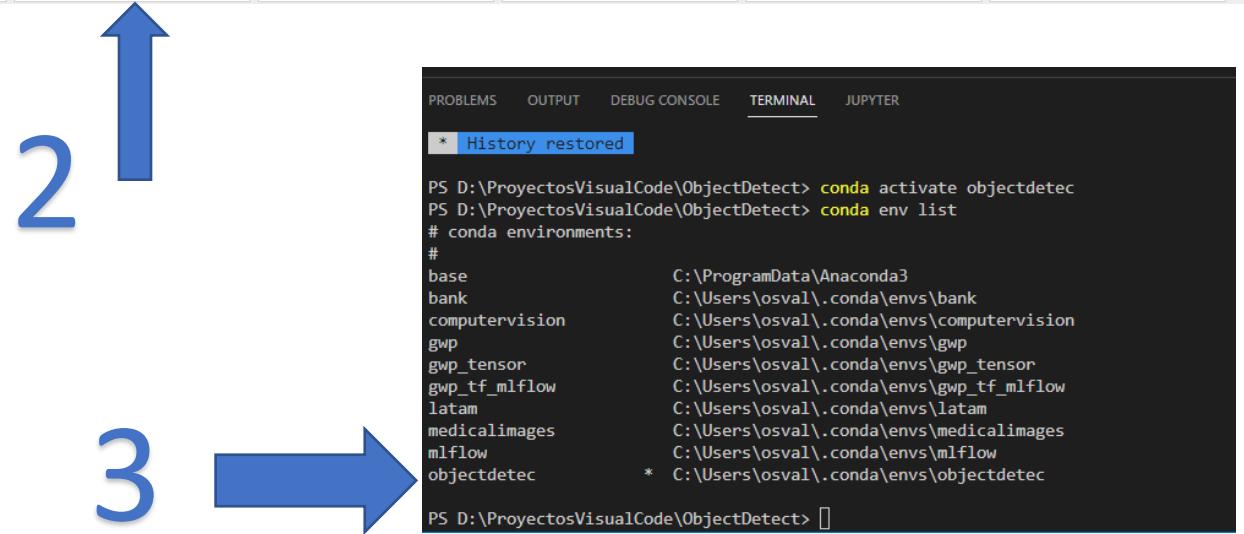


The screenshot displays a Visual Studio Code interface with several windows open:

- EXPLORER** sidebar: Shows a project structure under "OBJECTDETECT" containing files like "checkpoint", "deepLearningCV", "images", "logs", "1. bases.ipynb", "2. operaciones_Img.ipynb", etc.
- JUPITER** view: Shows four Jupyter notebook cells (1, 2, 3, 4) in Python 3.8.15 environment. Cell 1 imports cv2, numpy, and matplotlib. Cell 2 reads a banana image. Cell 3 extracts R, G, and B channels. Cell 4 displays the image and destroys all windows.
- TERMINAL** view: Shows the command "conda env list" being run in a PowerShell terminal. The output lists various conda environments and their paths, including "base", "bank", "computervision", "gwp", "gwp_tensor", "gwp_tf_mlflow", "latam", "medicalimages", "mlflow", and "objectdete".

Environments (Solución)

Programación II

The screenshot shows a terminal window with the following command history:

```

PS D:\ProyectosVisualCode\ObjectDetect> conda activate objectdete
PS D:\ProyectosVisualCode\ObjectDetect> conda env list
# conda environments:
#
base          C:\ProgramData\Anaconda3
bank          C:\Users\osval\.conda\envs\bank
computervision  C:\Users\osval\.conda\envs\computervision
gwp           C:\Users\osval\.conda\envs\gwp
gwp_tensor    C:\Users\osval\.conda\envs\gwp_tensor
gwp_tf_mlflow C:\Users\osval\.conda\envs\gwp_tf_mlflow
latam         C:\Users\osval\.conda\envs\latam
medicalimages C:\Users\osval\.conda\envs\medicalimages
mlflow        C:\Users\osval\.conda\envs\mlflow
objectdete    * C:\Users\osval\.conda\envs\objectdete

```

A large blue arrow labeled '3' points from the environments list towards this terminal window.