

# Data Analysis con Python

Juan Felipe Acevedo Pérez  
Monitor Junior

Curso libre:  
Data Analysis con  
Python

Monitor encargado:  
Juan Felipe Acevedo  
Pérez

**Correo:** [uniic\\_bog@unal.edu.co](mailto:uniic_bog@unal.edu.co)

**Tel:** 3165000 **Ext:** 12301

# 3

## Sesión 3

*Pandas 1 (Funciones)*

Correo: [uniic\\_bog@unal.edu.co](mailto:uniic_bog@unal.edu.co)

Teléfono: 3165000 ext 12301

# Contenido



**Indexación y  
selección de  
datos**



**Modificaciones  
básicas en BD**



**Ufuncs en  
Pandas**

# Indexación

La indexación consiste en crear índices de localizaciones específicas x,y para una base de datos, esto, con el fin de facilitar la extracción y consulta de elementos. Es posible identificar dos “tipos” de indexación:

- **Explícita:** Asignada por el *usuario* para una serie de elementos.
- **Implícita:** Asignada de forma *automática* por el lenguaje basándose en la posición.

Nota: recordemos que Python usa indexación en base-0.

indexing: getting a specific element

0 1 2 3  
**grades = [88, 72, 93, 94]**

>>> grades[2]



# Opciones de indexación con DataFrame

Type	Notes
<code>df[val]</code>	Select single column or sequence of columns from the DataFrame; special case conveniences: boolean array (filter rows), slice (slice rows), or boolean DataFrame (set values based on some criterion)
<code>df.loc[val]</code>	Selects single row or subset of rows from the DataFrame by label
<code>df.loc[:, val]</code>	Selects single column or subset of columns by label
<code>df.loc[val1, val2]</code>	Select both rows and columns by label
<code>df.iloc[where]</code>	Selects single row or subset of rows from the DataFrame by integer position
<code>df.iloc[:, where]</code>	Selects single column or subset of columns by integer position
<code>df.iloc[where_i, where_j]</code>	Select both rows and columns by integer position
<code>df.at[label_i, label_j]</code>	Select a single scalar value by row and column label
<code>df.iat[i, j]</code>	Select a single scalar value by row and column position (integers)
reindex method	Select either rows or columns by labels
get_value, set_value methods	Select single value by row and column label

# Modificaciones básicas en BD

	total_bill	tip	smoker	day	time	size
0	16.99	1.01	No	Sun	Dinner	2
1	10.34	1.66	No	Sun	Dinner	3
2	21.01	3.50	No	Sun	Dinner	3
3	23.68	3.31	No	Sun	Dinner	2
4	24.59	3.61	No	Sun	Dinner	4

- La creación de columnas y/o filas con unos valores determinados es parte fundamental del manejo de bases de datos. Comprenderemos la **mecánica fundamental** de la interacción con los datos contenidos en una serie o DataFrame, nos centraremos en las características más importantes, dejando las cosas menos comunes (es decir, más esotéricas).

# Ufuncs en Pandas

- Una de las piezas esenciales de NumPy es la capacidad de realizar cambios rápidos de elementos. Operaciones tanto con aritmética básica (suma, resta, multiplicación, etc.) como con operaciones más sofisticadas (funciones trigonométricas, exponenciales y logarítmicas).
- Pandas hereda gran parte de esta funcionalidad de NumPy, y las **ufuncs** que presentamos en la sesión anterior son clave para esto.



# Mapeo entre operadores de Python y métodos de Pandas

Python operator	Pandas method(s)
+	add()
-	sub(), subtract()
*	mul(), multiply()
/	truediv(), div(), divide()
//	floordiv()
%	mod()
**	pow()