



Universidad  
**Tecmilenio**®

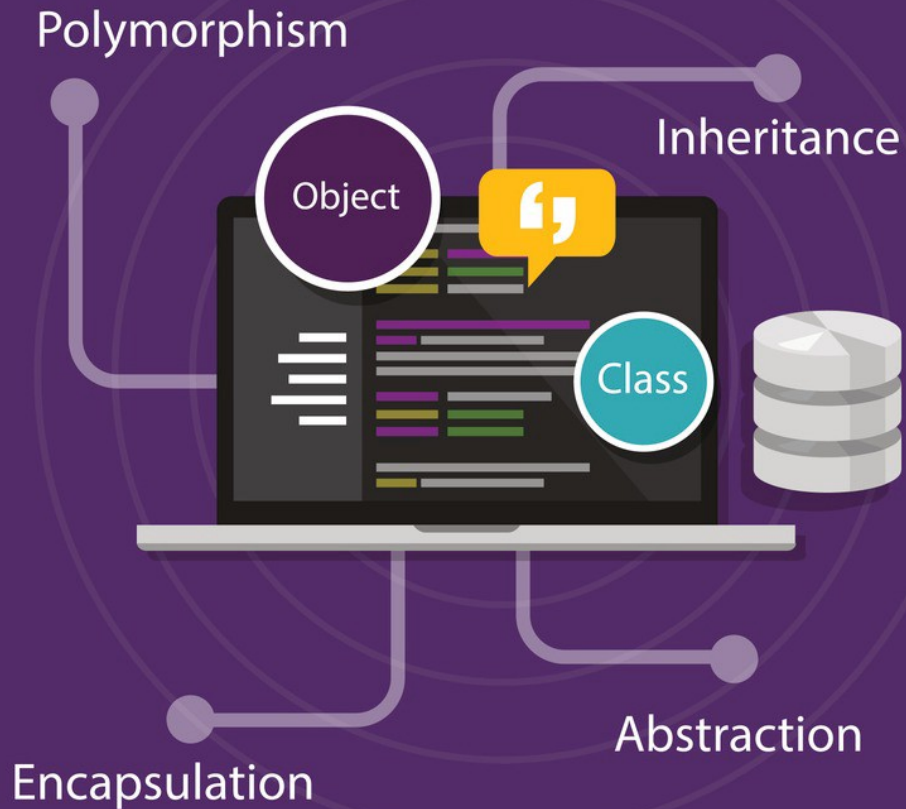


# Fundamentos de programación para Big Data

Orientación a objetos

Semana 3





Python es un lenguaje de alto nivel y está orientado a objetos como muchos otros lenguajes de programación.





Para entender a detalle la POO (programación orientada a objetos), revisa primero el siguiente material correspondiente a los objetos, atributos, métodos, clases e instancias en Python, de la página 191 a la 196:



Covantec. (2019). *Materiales del entrenamiento de programación en Python – Nivel básico. Versión 2.0.* Recuperado de <https://bit.ly/36zN4rU>





## Objeto String

Para conocer cómo crear cadenas en Python, te recomendamos ver el siguiente video:



Dirección de Producción de Contenidos. (2019, 4 de diciembre). *Tema 6. Cadenas* [Archivo de video]. Recuperado de <https://bit.ly/2N1JsqA>

Para saber sobre los caracteres especiales en Python, te recomendamos ver el siguiente video:



Dirección de Producción de Contenidos. (2019, 4 de diciembre). *Tema 6. Caracteres especiales en Python* [Archivo de video]. Recuperado de <https://bit.ly/2N5dlq7>

Para saber sobre las operaciones con cadenas en Python, te recomendamos ver el siguiente video:



Dirección de Producción de Contenidos. (2019, 4 de diciembre). *Tema 6. Operaciones con cadenas en Python* [Archivo de video]. Recuperado de <https://bit.ly/2sUtA2s>





El **módulo** *string* comprende funciones que permiten usar cadenas. Para utilizarlo, primero debes importarlo (Wentworth, Elkner, Downey y Meyers, 2019).

## Import string

Para poder visualizar lo que existe dentro de este, utiliza la función `Dir` con el nombre del módulo como un argumento.

## Dir (string)

Para saber más sobre un objeto específico de la lista, emplea el comando **type**. Es necesario especificar el nombre del módulo seguido del objeto, usando la notación punto.

```
type(string.digits)
<type 'str'>
type(string.find)
<type 'function'>
```

Puesto que `string.digits` es una cadena, se podrá mostrar para ver su contenido:

```
print string.digits 0123456789
```





## Objeto File

Python incluye la función **open()**, la cual permite abrir archivos. Esta regresa un objeto de archivo que permite leer o modificar el archivo.

Al abrir un archivo en Python debes indicarle el modo en que deseas hacerlo. Por defecto, Python usa el modo 'rt'.

Algo que debes tomar en cuenta es que después de utilizarlos es necesario cerrarlos usando el método **close()** para que Python libere los recursos que se ocuparon para abrirlo.



### Modo

### Descripción

't'	Modo texto.
'b'	Modo binario.
'r'	Lectura.
'w'	Escritura (si el archivo existe, lo trunca, y si no, lo crea).
'x'	Crea un archivo si no existe, y si ya existe, marca error.
'a'	Escribe al final de un archivo (si no existe, lo crea).
'+'	Lectura y escritura.

### Ejemplo:

```
# Abrir archivo
objeto_archivo = open('ejemplo.txt')

# Operaciones con el archivo
contenido = objeto_archivo.read()

# Cerrar archivo
objeto_archivo.close()

print (contenido)
```



Al usar la declaración **with** podrás asegurarte de que el archivo se cierre después de ejecutar el bloque de código correspondiente sin la necesidad de usar el método `close()`.

## Ejemplo:

```
with open ('ejemplo.txt') as objeto_archivo:  
    contenido = objeto_archivo.read()  
  
print(contenido)
```

**Los objetos de archivo cuentan con distintos métodos que podrás usar para leer o escribir en un archivo:**

### **read()**

Sirve para leer el contenido de un archivo. Requiere abrir el archivo en modo lectura.

### **readline()**

Funciona de manera similar, pero solo lee una línea del archivo. También requiere modo lectura.

### **readlines()**

Regresa una lista con el contenido de cada línea de un archivo como elemento. También requiere modo lectura.

### **write()**

Permite escribir en el archivo. Requiere modo escritura.







**PYTHON**

Esperamos que tengas una comprensión general de la programación en Python. Lo importante es practicar de manera continua con tareas sencillas que puedas automatizar y poco a poco tus habilidades mejorarán para realizar cosas más complejas.





- Covantec. (2019). *Materiales del entrenamiento de programación en Python – Nivel básico. Versión 2.0*. Recuperado de <https://bit.ly/36zN4rU>





# Fundamentos de programación para Big Data

Bases de datos

Semana 3





El origen de las bases de datos se debió a la necesidad de almacenar grandes cantidades de información de forma organizada y precisa. Hoy en día, llegan a ser increíbles colecciones de datos de un tamaño impresionante.

Una base de datos se puede definir como una colección de información estructurada y almacenada de forma electrónica para su posterior manipulación.





Las **bases de datos** permiten que, de manera simultánea, varios usuarios tengan acceso y puedan consultar información de una manera rápida y segura, utilizando una lógica y un lenguaje altamente complejos (Oracle, s.f.).



Una base de datos permite almacenar información de forma ordenada con diferentes propósitos y usos.

Existen diferentes tipos de bases de datos. Por ejemplo, las orientadas a objetos, orientadas a grafos, distribuidas, independientes, multimodelos, de código abierto, en la nube, relacionales, etc.





Uno de los lenguajes de programación más utilizados para las bases de datos relacionales es **SQL**.

**MySQL** es un sistema de gestión de bases de datos relacionales de código abierto basado en **SQL** para aplicaciones web. Puede ejecutarse en cualquier plataforma, procesando millones de consultas y miles de transacciones. Es popular para las empresas de comercio electrónico, convirtiendo a la flexibilidad bajo demanda en su característica principal.

MySQL es el **DBMS** (DataBase Management System) de algunos de los mejores sitios web y aplicaciones, incluyendo Airbnb, Uber, LinkedIn, Facebook, Twitter y YouTube (Oracle, s.f.).



MySQL





De acuerdo con el sitio Comunidad Python Argentina (s.f.) se recomiendan los siguientes **pasos para el uso de bases de datos en Python**:

- 1 **Importar** el conector.
- 2 **Conectar** a la base de datos (función connect del módulo conector).
- 3 **Abrir** un cursor (método cursor de la conexión).
- 4 **Ejecutar** una consulta (método execute del cursor).
- 5 **Obtener** los datos (método fetch sobre el cursor).
- 6 **Cerrar** el cursor (método close del cursor).





Un **ejemplo** elemental para conectar a una base de datos con MySQL es el siguiente:

```
>>> import MySQLdb
>>> db = MySQLdb.connect(host="localhost", user="root",
... passwd="mypassword", db="PythonU")
```

Una vez constituida la conexión, se crea un cursor. El método para crear uno se llama `cursor()`:

```
>>> cursor = db.cursor()
```

Realizados los pasos anteriores, se ejecutan algunos comandos SQL:

```
>>> cursor.execute("SELECT * FROM Students")
5L
```



El método `execute()` se utiliza para ejecutar comandos SQL. Para obtener un solo elemento se utiliza `fetchone()` y `fetchall()` para todos los elementos:

```
>>> cursor.fetchone()
(1L, 'Joe', 'Campbell', datetime.date(2006, 2, 10), 'N')

>>> cursor.fetchall()
((1L, 'Joe', 'Campbell', datetime.date(2006, 2, 10), 'N'),
 (2L, 'Joe', 'Doe', datetime.date(2004, 2, 16), 'N'),
 (3L, 'Rick', 'Hunter', datetime.date(2005, 3, 20), 'N'),
 (4L, 'Laura', 'Ingalls', datetime.date(2001, 3, 15), 'Y'),
 (5L, 'Virginia', 'Gonzalez', datetime.date(2003, 4, 2), 'N'))
```





Para observar un ejemplo de cómo conectarte a una base de datos en Python, te recomendamos ver el siguiente video:



Dirección de Producción de Contenidos. (2019, 4 de diciembre). *Tema 7. Conexión a una base de datos* [Archivo de video]. Recuperado de <https://bit.ly/2R04bfb>

Para observar un ejemplo de cómo conectarte a una base de datos MySQL en Python, te recomendamos ver el siguiente video:



Vicungoola Devs. (2017, 30 de noviembre). *16.- Bases de Datos MySQL en Python 3* [Archivo de video]. Recuperado de <https://bit.ly/2sXVESz>





SQL es un lenguaje estándar que permite administrar distintas bases de datos, mientras que MySQL es de licencia libre, de tal manera que es un software popular de los más utilizados y fáciles de manejar. Por otro lado, con ayuda de Python, puedes crear infinitas aplicaciones.

Finalmente, es importante saber que los datos son la materia prima dentro del Big Data, lo que hace fundamental y necesario utilizar correctamente una base de datos.





- Comunidad Python Argentina. (s.f.). *Base de datos con MySQL*. Recuperado de <http://www.python.org.ar/wiki/DbApi>
- Oracle. (s.f.). *Base de datos*. Recuperado de <https://www.oracle.com/mx/database/what-is-database.html>
- Vicungoola Devs. (2017, 30 de noviembre). *16.- Bases de Datos MySQL en Python 3* [Archivo de video]. Recuperado de <https://bit.ly/2sXVESz>




# Fundamentos de programación para Big Data

Introducción a lenguaje R

Semana 3





Desarrollado en 1993 por Robert Gentleman y Ross Ihaka, R es un lenguaje de programación inspirado en el lenguaje S. Además, es un ambiente de programación de libre acceso con código abierto para el análisis estadístico y gráfico.

El lenguaje R tiene soporte en una comunidad de desarrolladores, quienes continuamente aportan mejoras a la herramienta y la impulsan de manera constante para estar a la vanguardia.





Para conocer los conceptos básicos del lenguaje R, te recomendamos ver el siguiente video:



Dirección de Producción de Contenidos.  
(2019, 4 de diciembre). *Tema 8. Instalación del ambiente* [Archivo de video].  
Recuperado de <https://bit.ly/2s9MW3i>

- Multiplataforma (Mac, Windows, Linux e incluso en Playstation 3).
- Lenguaje de programación completo, permite el desarrollo de DSL.
- Promueve la investigación reproducible.
- Está actualizado gracias a que tiene una comunidad activa.
- Se puede combinar con otras herramientas.
- Tiene capacidades gráficas muy sofisticadas.

## Ventajas de utilizar lenguaje R



R es un programa estadístico y un lenguaje de programación de uso libre, de distribución gratuita y código abierto, desarrollado a partir de un proyecto colaborativo voluntario de investigadores y estadísticos de diversos países y disciplinas.

Es un programa basado en comandos, que permite acceder a todos los procedimientos y opciones a través de una sintaxis textual. Fue oficialmente presentado en 1997 bajo Licencia General Pública de la Fundación de Software Libre (Martínez y Losa, 2017).





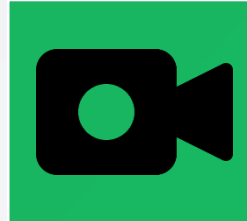
R es un entorno de programación para el análisis estadístico y gráfico de datos que cada vez se hace más popular entre los investigadores de todas las disciplinas.

Cada día, más universidades lo suman a sus planes de estudios, ya que es oportuno y pertinente para los investigadores de cualquier área del saber. Entre algunas de sus ventajas podemos mencionar las siguientes:

- ✓ **Estructuras de datos y operaciones vectorizadas.**
- ✓ **Valores faltantes.**
- ✓ **Ambientes interactivos.**

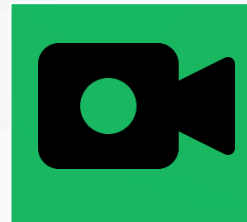


**Aquí podrás conocer acerca del ambiente y lenguaje R: lo que es, cómo entenderlo y por qué usarlo, entre otros aspectos importantes que debes considerar:**



Ortiz, T. (s.f.). *R para análisis de datos*. Recuperado de <https://bit.ly/35x4N1x>

**Para conocer cuestiones básicas, te recomendamos leer el segundo apartado:**



Andina, M. (2018). *Introducción a estadística con R*. Recuperado de <https://bit.ly/35yvqn4>





R es un lenguaje de programación libre, no tiene limitadas sus funciones, es colaborativo y sus librerías están disponibles y abiertas para todos.

R ha sido adoptado por diferentes plataformas como Azure, Visual Studio y Power BI, extendiendo su uso y popularidad entre los interesados en el área de la ciencia de los datos.





- Martínez, R., y Losa, A. (2017). *El procesamiento estadístico con R en la investigación científica*. Recuperado de <https://bit.ly/39xxpKN>

