Clase 0

Introducción a la programación con Python

**Programación: un breve viaje por la historia**

La historia de la programación está fuertemente vinculada a la aparición y crecimiento exponencial de los computadores.

**La evolución de los lenguajes**

En sus inicios programar implicaba el uso del código binario con cadenas de 0s y 1s, el lenguaje que entiende directamente el computador. ¡Algo sumamente complejo!.

Más tarde se crea el lenguaje ensamblador, que en vez de números utilizaba letras, algo más fácil de recordar.

Finalmente, surgen los lenguajes de alto nivel, que suelen utilizar términos en inglés para dictar las órdenes a seguir. Este sería un proceso intermedio entre el lenguaje de la máquina (binario) y el “código fuente” (proceso por compilador o intérprete).

**¿Sabías qué…?**

Lady Ada Countess of Lovelace (1815-1852), matemática y escritora, es reconocida como la primera programadora de la historia. En su honor, Jean Ichbiah, en 1979, le puso ADA al lenguaje que había creado.

**Programación**

“La Programación es el proceso con el que se crea algún tipo de aplicación o software, para materializar un concepto o proyecto que requiere de la utilización de un lenguaje informático y así poder llevarse a cabo”. [NeoAttack](https://neoattack.com/neowiki/programacion/)

**De otro modo**

Es el proceso de escribir instrucciones para la computadora, en el formato y sintaxis que el lenguaje de programación requiere, a fin de que se realice determinada acción. Programar es comunicar.

**Lenguajes**

**Lenguajes de frontend**

Principales lenguajes de Frontend (lo que sucede en el cliente): JavaScript, HTML y CSS. HTML y CSS son lenguajes, pero no de programación. Sirven para dar estructura y estilo al sitio.

**Lenguajes de backend**

Principales lenguajes de Backend (lo que sucede en el servidor): Python, Java, Ruby, PHP.

**Frontend vs. Backend**

Si bien pareciera que al desarrollar solo el frontend de un sitio el mismo ya está completo, por ejemplo, no entenderíamos si realmente se concretó el pago de un producto, tampoco si terminó generando una orden para determinada cosa. Todo eso lo resuelve el backend.

| **FrontEnd** | **BackEnd** |
| --- | --- |
| Es la parte de un sitio web que interactúa con los usuarios, se le dice lado del cliente. | Es la parte que conecta con la base de datos y el servidor que usa el sitio web, se le dice lado del servidor. |
| Maquetación, lógica visible de un sitio web. Es lo que puede ver un cliente, y con lo que interactúa. | Desarrollo de API del lado de los servidores y bases de datos. Se conecta la base de datos con la interfaz de la web/app. |

En Python, nosotros vamos a dedicarnos más que nada a la programación BackEnd.

**Algunas tareas involucradas en el backend**

* Creación, integración y mantenimiento de bases de datos
* Comunicación del server, aplicación y base de datos
* Creación, mantenimiento, testeo y debugging del backend
* Creación de DB backups
* Conversión y devolución de datos
* Mantenimiento del sistema y deploy
* Integración con API’s
* Seguridad

**Frameworks**

Un framework, o marco de trabajo, es un conjunto de herramientas y clases que nos permiten solucionar un problema o funcionalidad. Están escritos en uno o más lenguajes de programación. Los frameworks establecen, además, una estructura determinada para el código y todos los archivos, como así también una metodología armada del proyecto.

**Programadores**

**¿Cuál es el trabajo del programador?**

Los programadores desarrollan aplicaciones y programas informáticos, sirviéndose de las bases de un software existente para crear una interfaz para los usuarios con fines comerciales, profesionales o recreativos.

**Consejos para motivarte**

* Nunca pierdas la ilusión
* Encuentra tu ritmo
* Trata de entender la lógica, no memorices código.
* Desconectar es importante
* Pide ayuda, y ayuda a quienes lo necesitan
* Ten presente lo que has logrado hasta el momento
* Enfócate en el día a día más que en el largo plazo
* Reaprende los fundamentos las veces que sean necesarias

**Metodologías Ágiles**

**¿Qué son las metodologías ágiles?**

Son las metodologías que permiten adaptar la forma de trabajo a las condiciones del proyecto, dando flexibilidad e inmediatez al amoldarse a las circunstancias del entorno.

**¿Qué metodología ágil usamos?**

Trabajaremos con Programación Extrema o eXtreme Programming (XP). Su objetivo principal es que un equipo de desarrollo pueda producir software de mejor calidad y promover buena calidad de vida al equipo.

**¿Por qué XP?**

* Desarrollo iterativo e incremental
* Programación en parejas
* Corrección de todos los errores
* Refactorización del código
* Propiedad de código compartida
* Simplicidad en el código

**Algunos conceptos claves**

**Datos vs. Información**

**Datos:** Característica o atributo sin procesamiento, el cual no informa nada por sí solo.

**Información**: Unión de datos procesados, que se complementan para informar un hecho.

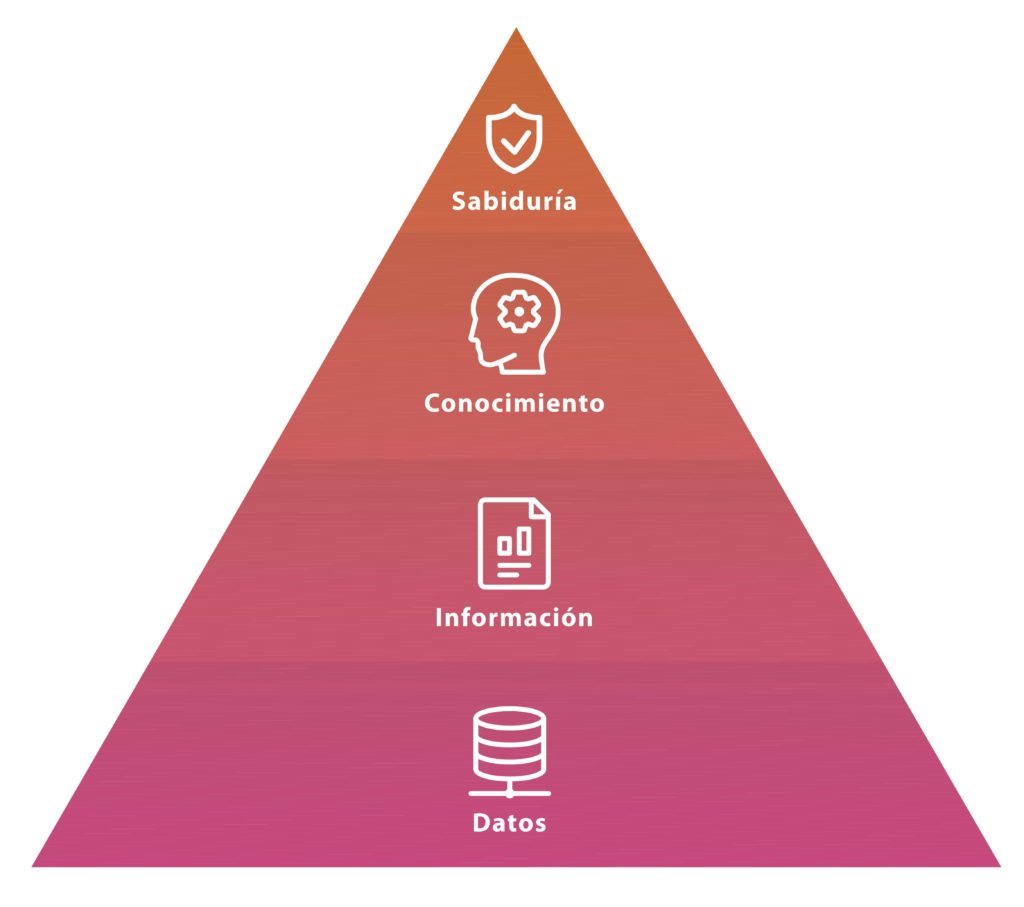
**Datos**

Un dato es un conjunto discreto, de factores objetivos sobre un hecho real. Dentro de un contexto empresarial, el concepto de dato es definido como un registro de transacciones. Un dato no dice nada sobre el por qué de las cosas y por sí mismo tiene poca o ninguna relevancia o propósito.

**Información**

A diferencia de los datos, la información tiene significado, relevancia y propósito. No solo puede formar potencialmente a quien la recibe, sino que está organizada para algún propósito. Los datos se convierten en información cuando su creador les añade significado. Transformamos datos en información añadiéndoles valor en diversos sentidos.

**Pirámide de la sabiduría**



Esta pirámide nos permite establecer una jerarquía entre datos, información, conocimiento y sabiduría, colocando los datos en su parte más baja y la sabiduría en la más alta, o de más valor, justamente aspirando a alcanzar la sabiduría dentro de una estrategia de aplicación de datos.

**Python**

**¿Qué es?**

Es un lenguaje de programación de código abierto, orientado a objetos, simple y fácil de entender.

Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único.

Es un lenguaje interpretado con tipado dinámico, multiparadigma y multiplataforma.

**¿Para qué se utiliza Python?**

* Inteligencia Artificial
* Big Data
* Data Science
* Frameworks de Pruebas
* Desarrollo Web

Nosotros en este curso nos centraremos en la parte web del desarrollo.

| **Ventajas** | **Desventajas** |
| --- | --- |
| Fácil de aprender | Problemas con hosting |
| Fácil de usar | Lentitud en multithread |
| Orientado a objetos | Lentitud de ejecución |
| Software libre |  |
| Portable |  |
| Multiplataforma |  |
| Poderoso |  |

**Entorno de programación**

**Google Colab**

Para programar con comodidad se suele instalar un IDE o editor de texto. En este curso les proponemos instalar [Visual Studio Code](https://www.youtube.com/watch?v=6npp93ZIQgM&ab_channel=v%C3%ADctorRomero); pero lo vamos a usar a partir de la clase 17 cuando empecemos a desarrollar nuestro proyecto.

Por ahora trabajaremos con un entorno de Google para enfocarnos solo en la programación, este entorno se llama Colab.

**¿Qué es Colab?**

Colab, o también conocido como “Google Colaboratory", nos permite escribir y ejecutar código de Python en nuestro navegador, con algunas características particulares como ser:

* No es requerido realizar ninguna instalación o configuración local del entorno de trabajo.
* Acceso gratuito a GPU que nos ofrece Google
* Facilidad para compartir y realizar programación “colaborativa”.