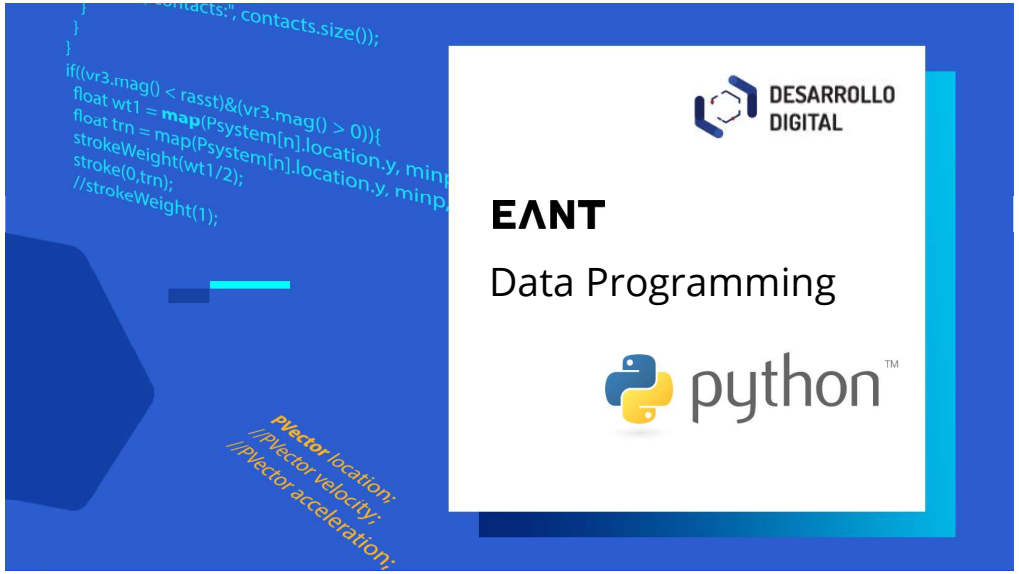


Presentación

Bienvenidos a la EANT

{Lo que nos gusta…

…el Barro}

…entrenar}

…ir al hueso}

…explorar}

¿Qué vamos a aprender?

A pensar, diseñar y construir soluciones de extracción, transformación y carga de datos

¿Cómo lo vamos a aprender?

Entrenando el uso de programación Python e infraestructura digitala través de distintos desafíos y casos típicos de Data Engineer

Pre requisitos

El curso supone un conocimiento básico pero concreto en programación con Python:

- Sintaxis básica
- Variables y tipos de datos
- Uso de estructuras de decisión if/else
- Manejo básico de objetos y métodos
- Manejo de listas
- Uso de estructuras de bucle (for)

2 Reglas de Oro en la EANT

1

Concepto nuevo? → Lo anoto → Lo googleo

2

No entendí? → Pregunto / Pregunto / Pregunto

Hablemos de Datos

Business Intelligence

Flow de Datos: del dato a la acción

El circuito de BI organiza y alinea todas las "capas" de trabajo que se requieren para conectar a una organización con los eventos que la caracterizan

Flow de Datos
Del dato a la información

FUENTES DE DATOS | PROCESOS ETL | MODELO DE DATOS

Datos → **Información**

Aplicaciones de Datos

Flow de Datos
Del dato a la información

FUENTES DE DATOS | PROCESOS ETL | MODELO DE DATOS

Aplicaciones de Datos

En TODO problema de BI, Data Analytics ó Aplicaciones de Datos en general los procesos **Extracción, Transformación y Carga** suelen significar el **80%** del esfuerzo del proyecto

Flow de Datos
Fuentes

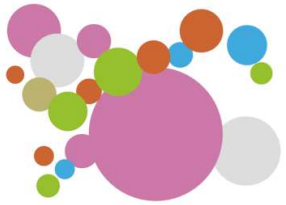
FUENTES DE DATOS | PROCESOS ETL | MODELO DE DATOS

Aplicaciones de Datos

Las capa de Fuentes de Datos se relaciona con nuestra capacidad para identificar orígenes de datos útiles, potables y extraíbles para nutrir a las aplicaciones de información

Fuentes de Datos
La materia prima

El estado de digitalización global actual en el que vivimos nos ofrece una gran oferta de datos de diversos orígenes:



- OLTP – Online Transactional Process: sistemas soporte a la operación (facturación, contabilidad, CRM, ventas, etc)
- Sistemas de bases de datos
- Servicios de Terceros (API)
- Informes de mercado, archivos de referencia, etc
- Logs (ej: Web Page, E-commerce, etc)
- Web Scraping
- Social Data (ej: Facebook, Twitter, etc)

Fuentes de Datos

Quién pega en las fuentes pega más rápido

En términos de esfuerzos relativos, una mejora en el tipo, cantidad y diversidad de las fuentes de datos (nuevas fuentes, mejor calidad, integración, enriquecimiento, etc) tendrá mejores chances de obtener un impacto certero en el potencial de aplicación:

Flow de Datos

Procesos ETL

FUENTES DE DATOS PROCESOS ETL MODELO DE DATOS

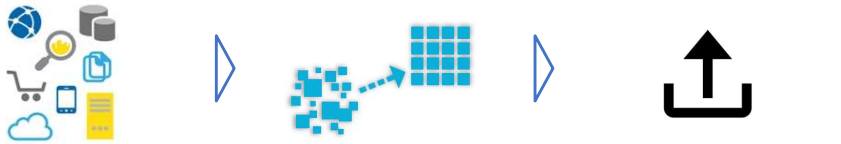
Aplicaciones de Datos

ETL comprende todos los procesos de acople, limpieza e integración que transforman los datos brutos en información modelada y disponible

Procesos ETL

Circuito de trabajo

"Extract-Tranform-Load" tiene a cargo la compleja tarea de empalmar los eventos diarios con los modelos de información que alimentan a las aplicaciones de datos



Extracción

Comprende los procesos de extracción/importación de los archivos o fuentes de datos que alimentan a todo el sistema

Transformación

Son todos los procesos que hacen a adaptar los datos originales al formato requerido por el modelo de datos del sistema

Carga

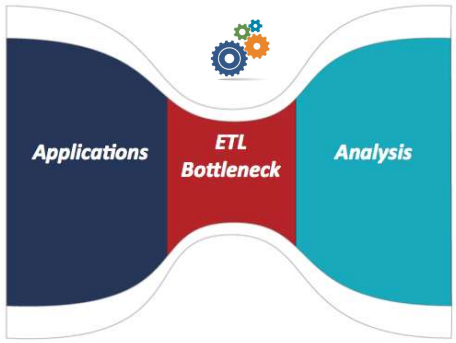
Son los procesos que se encargan de la carga final de los datos transformados al modelo de datos del sistema

Procesos ETL

Desafíos

Los procesos ETL son de los más críticos en el montaje de un sistema de Business Intelligence debido a que deben afrontar el suministro continuo de información satisfaciendo una gran demanda de requerimientos.

30-40%
data growth
per year
Source: 2013 IBM Briefing Book

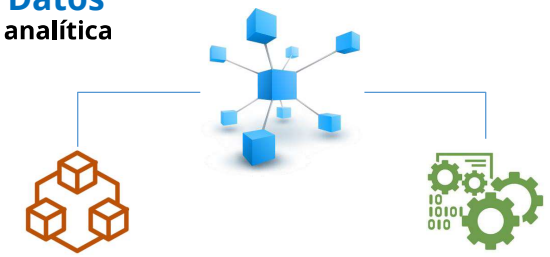


Flow de Datos
Procesos ETL

FUENTES DE DATOS → PROCESOS ETL → MODELO DE DATOS → Aplicaciones de Datos

El modelo de datos es la solución con que se ha pensado alimentar con información adecuada a las distintas aplicaciones de datos que se soporten en nuestro sistema

Modelo de Datos
El motor de la analítica



Modelo de Información

Son los modelos, formatos y métricas de información que se han definido como necesarios para el correcto funcionamiento de las aplicaciones de datos

Infraestructura de Datos

Comprende tas tecnologías seleccionadas para el almacenamiento, procesamiento y la disponibilización de la información

Modelo de Datos
Infraestructuras de Datos

Archivos de Datos

Son la elementos de almacenamiento independientes. Hoy en día existen gran variedad de protocolos

Bases de Datos

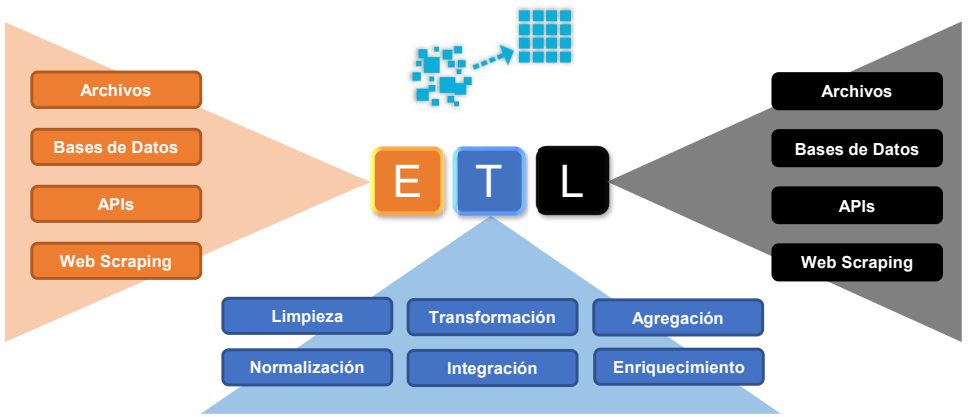
Son sistemas de unificación y organizacióndе datos. Suelen evolucionar a modelos relacionales

Servicio de Datos

Son aplicaciones de software en la nube que proveen métodos de descarga de datos

¿Para qué Data Programming?

A dónde vamos…



Necesitamos conocimientos fuertes en técnica y tecnología de datos

¡Vamos a Programar!

Vamos a programar!!

¿De qué estamos hablando cuando hablamos de Programar?

Desafíos de la Programación de Datos

Volumen/Velocidad

Debemos entrenar para enfrentar grandes procesamientos de datos donde la cantidad y la velocidad son requerimientos de alta demanda

Caos

Debemos aprender a lidiar con formatos, errores y omisiones todo el tiempo, en donde la "estandarización" es lo último que tendremos

Complejidad

Debemos aprender a desarrollar lógicas de procesamiento complejas capaces de organizar eficientemente el trabajo en múltiples capaz

En dónde vamos a poner el acento...

Los MUST de este curso:

1) Actitud de investigación

type( )

Identifica al tipo de elemento con el que estamos lidiando

help( )

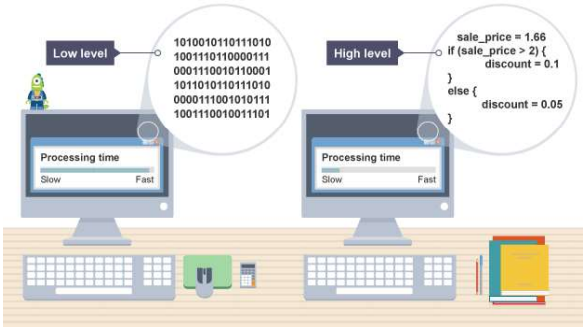
Devuelve la documentación disponible del elemento

Google( )

Devuelve todo lo demás...

Los MUST de este curso:

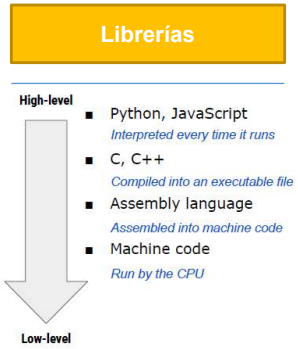
2) Eficiencia

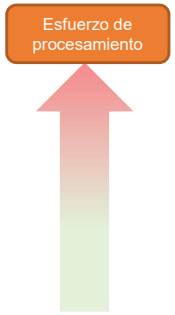


En el trabajo con datos es donde la eficiencia de un lenguaje más se pone a prueba

Los MUST de este curso:

2) Eficiencia: Librerías → el último recurso





Las librerías no deben corregir deficiencias de conocimiento, sino potenciarlos!

Es necesario aprender a explotar las posibilidades del lenguaje utilizando sus capacidades naturales

Y en esto Python esta especialmente preparado para el trabajo con Datos!!!

Los MUST de este curso:

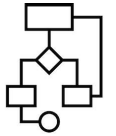
3) Código performante

Código Legible

Es necesario que tu código se ordenado, bien tabulado e identificable

Funciones!

Para construir grandes softwares necesitamos organizar bien la lógica de cada programa



Arquitectura

Poner el acento en la estética final de cada programa

Inmersión Digital con JavaScript

1.
Datos, protocolos y archivos

-Estructuras de datos
-Codificación
-Protocolos de datos
-Manipulación de archivos con Python

Datos

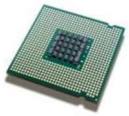
Del binario a su mesa!

Datos

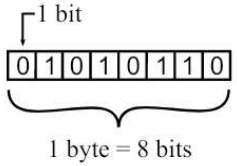
Estamos hechos de bits

En su forma más fundamental los datos se almacenan en elementos que sólo pueden tener 2 estados:

Prendido = 1
ó
Apagado = 0



En informática las combinaciones de 0s y 1s suelen estar organizadas en grupos de a 8 dígitos (8bit)



Datos

Codificación de Datos

Los caracteres son el resultado de crear sistemas de codificación en los que cada combinación de 0s1s se corresponde con un caractér

El American Standard Code for Information Interchange (ASCII) es un sistema de codificación de caracteres ampliamente extendido que se introdujo en 1963.

ASCii es parte del sistema de codificación UNICODE

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Datos

Código ASCii

Si bien los más comunes son los caratéres “imprimibles” existen también muchos tipos de caracteres especiales que viajan en la información:

ASCII control characters (character code 0-31)

Los caracteres de control se utilizan para transmitir información sobre cómo operar la secuencias de caracteres en cuestión

- \r matches carriage return.
- \n matches linefeed.
- \t matches horizontal tab.
- \v matches vertical tab.
- \0 matches NUL character.

https://theasciicode.com.ar/

https://www.ascii-code.com/

Datos

Unicode

El ASCii Extendido es una versión de 256 caracteres (el doble que el ASCii base) y ha servido a para que los distintos lenguajes incluyan caracteres específicos de su escritura

ASCII Extendido resuelve el problema para los idiomas que se basan en el alfabeto latino pero ... ¿qué pasa con los otros que necesitan un alfabeto completamente diferente? ¿Griego? ¿Ruso? Chino y similares… (+ 27.000 caracteres)

La codificación UNICODE define cómo son agrupados los binarios para el almacenamiento de cada set de caracteres

Encodings: UTF-8 vs UTF-16 vs UTF-32

La codificación de la información es uno de los elementos fundamentales del manejo de datos

Práctica

0s y 1s

-Exploración de **mensaje.txt**
-Podrías descifrar el mensaje?

Archivos

Almacenando datos

Los archivos son unidades de agrupación de datos

¿Qué es lo que hace la diferencia entre unos y otros tipos de archivos?

Archivos

Protocolos de Datos

Archivos de Texto

Es el formato plano (sín formato) por excelencia

Comma Separated Values

Son archivos planos con formato tabular

eXceL Spreadsheet

Son archivos que embeben datos planos junto con datos de aplicación

Protocolos

Texto Plano

Los archivos de texto plano se caracterizan por no estar enmarcados (necesariamente) en ningún formato específico, por lo que pueden usarse de manera libre para transportar casi cualquier protocolo de datos.

Los TXT se espera que estén libres de codificación no pero sin embargo hay una serie de “datos de aplicación” que suelen venir embebidos en los datos

¿Podrías decir cuáles?

Arándano
Frambuesa
Frutilla
Mandarina
Naranja
Pomelo
Melón
Sandía
Coco
Kiwi
Mango
Papaya
Piña
Ananá
Banana

Protocolos
CSV

Los CSV son archivos de texto plano que transportan datos bajo el protocolo de “Separación Por Comas”

En general el primer registro es el que lleva los nombre de los campos

En este caso la tabulación o separación se hace con el carácter de la cóma (,), pero pueden usarse otros cómo: **/ ; & % : tab**

```
Cliente, Fecha, Venta
ABC, 01/01/2011, "$1,630.00"
DEF, 02/01/2011, "$1,313.00"
GHI, 03/01/2011, "$1,230.00"
JKL, 04/01/2011, "$1,840.00"
MNO, 05/01/2011, "$1,566.00"
PQR, 06/01/2011, "$1,443.00"
STU, 07/01/2011, "$1,047.00"
VWX, 08/01/2011, "$1,581.00"
YZA, 09/01/2011, "$1,251.00"
```


Protocolos
XML

Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de marcado de propósito general.

Fue introducido como una solución al desarrollo de protocolos que permitieran el transporte de estructuras complejas de información.

Sin embargo se lo considera engorroso de manipular y pesado de transportar por lo que ha ido perdiendo relevancia frente a la agilidad de JSON.

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```


Protocolos
JSON

JavaScript Object Notation es un protocolo para el manejo de datos semi-estructurados que se caracteriza por su liviandad y flexibilidad para representar estructuras de datos variables y de gran complejidad

JSON es uno de los stándares más importantes de la actualidad debido a sus excelentes prestaciones para el trabajo con APIs y Big Data

```
{
 "squadName": "Super hero squad",
 "homeTown": "Metro City",
 "formed": 2016,
 "secretBase": "Super tower",
 "active": true,
 "members": [
  {
   "name": "Molecule Man",
   "age": 29,
   "secretIdentity": "Dan Jukes",
   "powers": [
    "Radiation resistance",
    "Turning tiny",
    "Radiation blast"
   ]
  },
  {
   "name": "Madame Uppercut",
   "age": 39,
   "secretIdentity": "Jane Wilson",
   "powers": [
    "Million tonne punch",
    "Damage resistance",
    "Superhuman reflexes"
   ]
  }
}
```


Cómo funciona un portal Web?
Trivia!

1) Cuál es la diferencia entre ?
2) Qué tipo de carácter es ‘/n’?
3) Para qué sirve la codificación de caracteres?
4) Qué es UTF-8? Cuándo se usa?
5) Qué son los “protocolos de datos”? Para qué se usan?
6) Cuál es la diferencia entre un archivo csv y uno xls?
7) Qué es JSON? Para qué sirve?

Manipulando Archivos

Método Open()

El método nativo **open()** de Python permite cargar cualquier tipo de archivo plano en un **buffer de datos**

Un **buffer** es un espacio virtual que se asigna a la **memoria** de la compu/server con que estamos trabajamos

```
archivo = open('nombre_archivo.extension')
```

Los datos son **almacenados en memoria** disponibles para ser operados

Los cambios en memoria pueden ser impactados en cualquier momento en el archivo de disco → archivo.write()

Los datos en memoria se trabajan hasta que se cierra el buffer con el método archivo.close() → archivo.close()

Práctica (part#1)

Manipulación de archivos de manera Nativa

El archivo 'frutas.txt' contiene una larga lista de frutas que nos gustan!

Cómo harías para imprimir todos los elementos de la lista de manera secuencial y con el lenguaje correcto

CONSEJO:

¿No funciona?

Investigá lo que no sepas

Manipulando Datos

Strings → Métodos útiles

Los **strings** son objetos! Y eso es una buena noticia!
Los objetos de tipo string saben hacer muchas cosas:

Te interesó alguna?
→ Investigala!

objeto.capitalize()	objeto.replace()	objeto.encode()
objeto.count()	objeto.split()	objeto.low()
objeto.find()	objeto.index()	objeto.zfill()
objeto.rfind()	objeto.startswith()	objeto.partition()

Práctica (part#2)

Manipulación de archivos de manera Nativa

El archivo 'subte.csv' contiene una serie de datos sobre cómo evolucionó el precio del boleto en los últimos años…

¿Cómo harías para extraer de la serie únicamente el campo de valores ($) ?

¿Qué métodos de Strings utilizarías para resolver el ejercicio?