**Fundamentos: Intro Bienvenida - Pedro Nieto**

Conceptos a saber:

Bit y sus distintas capacidades y escalas. Cada letra almacenada corresponde a un encoding y en forma binaria. Por ejemplo, la A es encoding UTF-8 Y 01000001 en binario. El encoding es la cantidad de números a utilizar en binario para almacenar una letra. Cuantos unos y ceros para definir una palabra. Ejemplo, si es de la A a la Z, la UTF-8 es la valida, si contamos la A con acento o cualquiera variedad, haría falta mas combinaciones de 0 y 1 y por tanto mayor encoding.

Clock speed (MHz) Y – CPU Clock Speed X (gráfica)

¿Por qué en ceros y unos y no mas números?¿por que solo usamos dos números?

Viene de si o no y los primeros ordenadores se hacían con válvulas de vacio, con lo cual se consideraba dos situaciones: vacio (0) o lleno (1).

Cuando hay un problema se llama bug por que antiguamente el hecho de que un insecto se metiese podía estropear y no marcaba ni lleno ni vacío.

En la segunda generación (1952-1964) pasamos de válvulas a transistores.

Tercera generación (1964-1971), el circuito integrado.

Cuarta generación (1971-1981), el microprocesador (integración de toda la CPU).

Partes:

Disco, RAM, Placa Base, GPU(pizza), CPU(cocina) y BUSES(nosotros).

Quinta generación (1982-2000), DVD.

Sexta generación (2000-Qbit), procesadores en paralelo y miniaturización avanzada. Otra novedad es la incorporación de chips en procesadores especializados en las tareas de video y sonido.

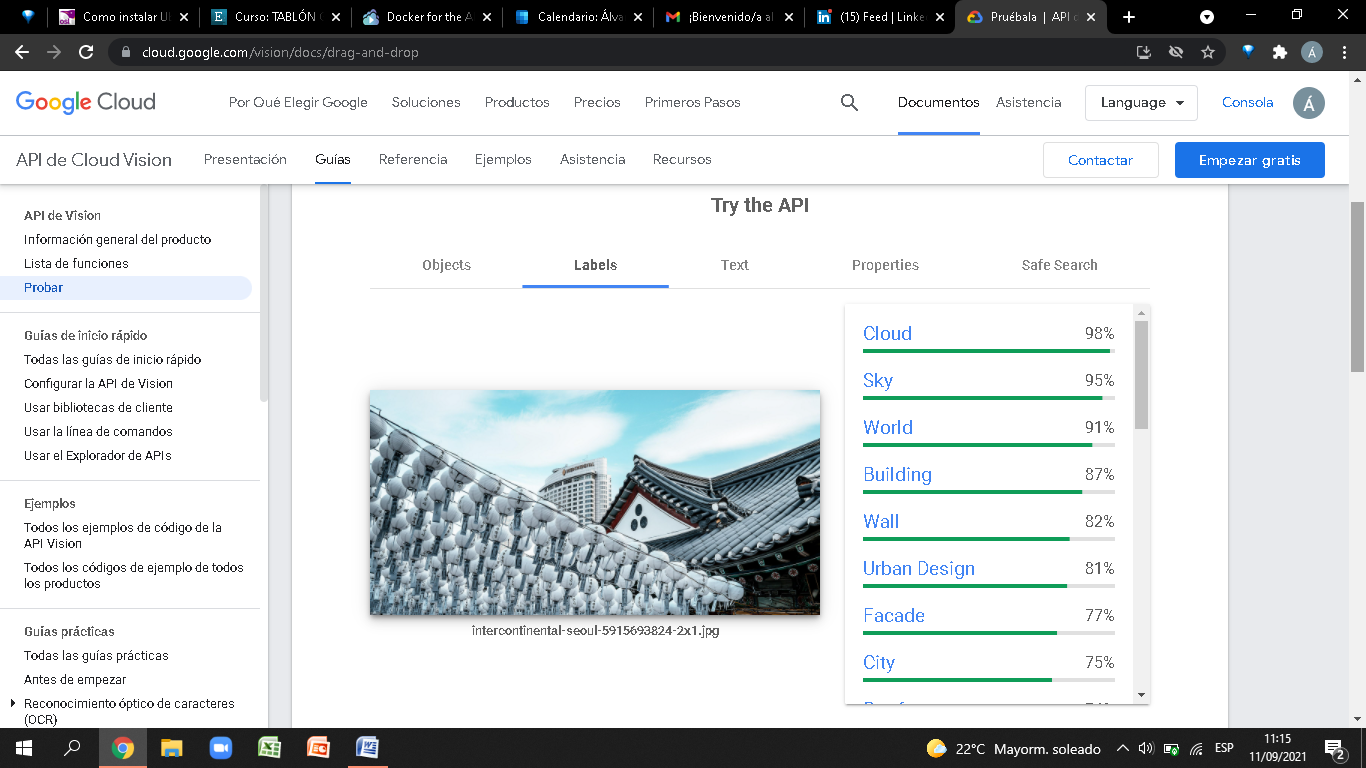
¿La alternativa a usar más que ceros y uno? Una base mayor, es decir, computación cuántica.

LOS BASICOS

Relacion entre ordenadores: Cliente-Servidor o Servidor-Servidor

Servidor si procesas y cliente si recibes la información. El proceso entre ambos es petición.

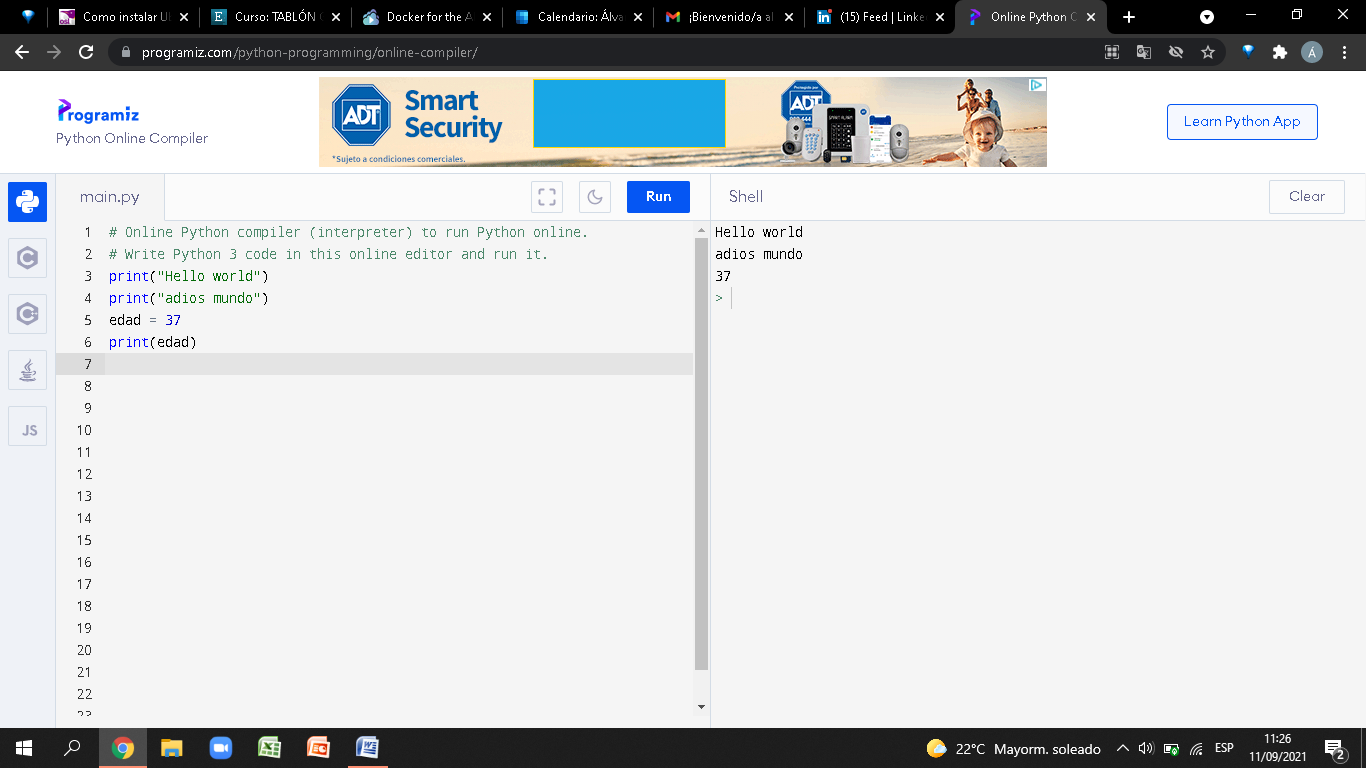
La voz; canal es internet; el lenguaje es el idioma (en nuestro caso de programación es http, el estandar); lo que se pide se conoce como **API** (que cosas puedo hacer/pedir, funciones limitadas), se piden con Rest (invoco-resultado); el okay son códigos: 200 (si), distinto de 200 (mal).

API EJEMPLO (https://cloud.google.com/vision/docs/drag-and-drop):

Lenguaje de programación:

3 tipos: compilado (por ejemplo C), interpretado (Python) e intermedio (Scala, Java, Coding...).

Python (<https://www.programiz.com/python-programming/online-compiler/> ):



HTML (<https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro> ):

Pagina recomendada, **codesignal** ([**https://app.codesignal.com/login**](https://app.codesignal.com/login)):

Caché, memoria de acceso rápido para optimizar un proceso.

Log, la manera que tiene un ordenador de decirnos que esta pasando. Si algo no funciona, revisar los logs para detectar el error. Deberíamos de poner siempre logs en nuestra aplicación (bugear), debugear es quitar logs incorrectos).

Encriptacion, ([**https://www.puttygen.com/download-putty**](https://www.puttygen.com/download-putty) ); https://www.puttygen.com/download.php?val=55

Clave pública (llave que dejas)

Clave privada (llave que no se la das a nadie)

Protocolos:

SSH: Comandos ssh, putty, pantalla negra

SFTP: trabajar con ficheros FILEZILLA

FTP: transmitir ficheros

HTTP:80

FTP Y SSH:22

SFTP: 443

S primero significa que está cifrada, es segura.

Puerto: de 0 a 65535

Pagina web debe estar en el puerto 80

CPU/GPU/TPU(termomix)

Sistema operativo, el capaz de interactuar entre usuario, aplicación y hardware.

Usuario<>Aplicación<>Sistema Operativo<> Hardware

Lista de software a instalar: GitKraken (gestión de código, hacer repositorios de código, donde guardar las practicas en una carpeta); Typora (visualización de un tipo de markdown, notas); Putty; Java (Java jdk versión); Docker (WSL 2 para Windows Home o Pro; <https://docs.docker.com/desktop/windows/install/> ).

Tipos de archivos: CSV (fichero separado por comas), Excel y TXT

Concepto librería: código de terceros que empleamos en nuestra aplicación.

Cheatsheet: usaremos hojas de chuletas