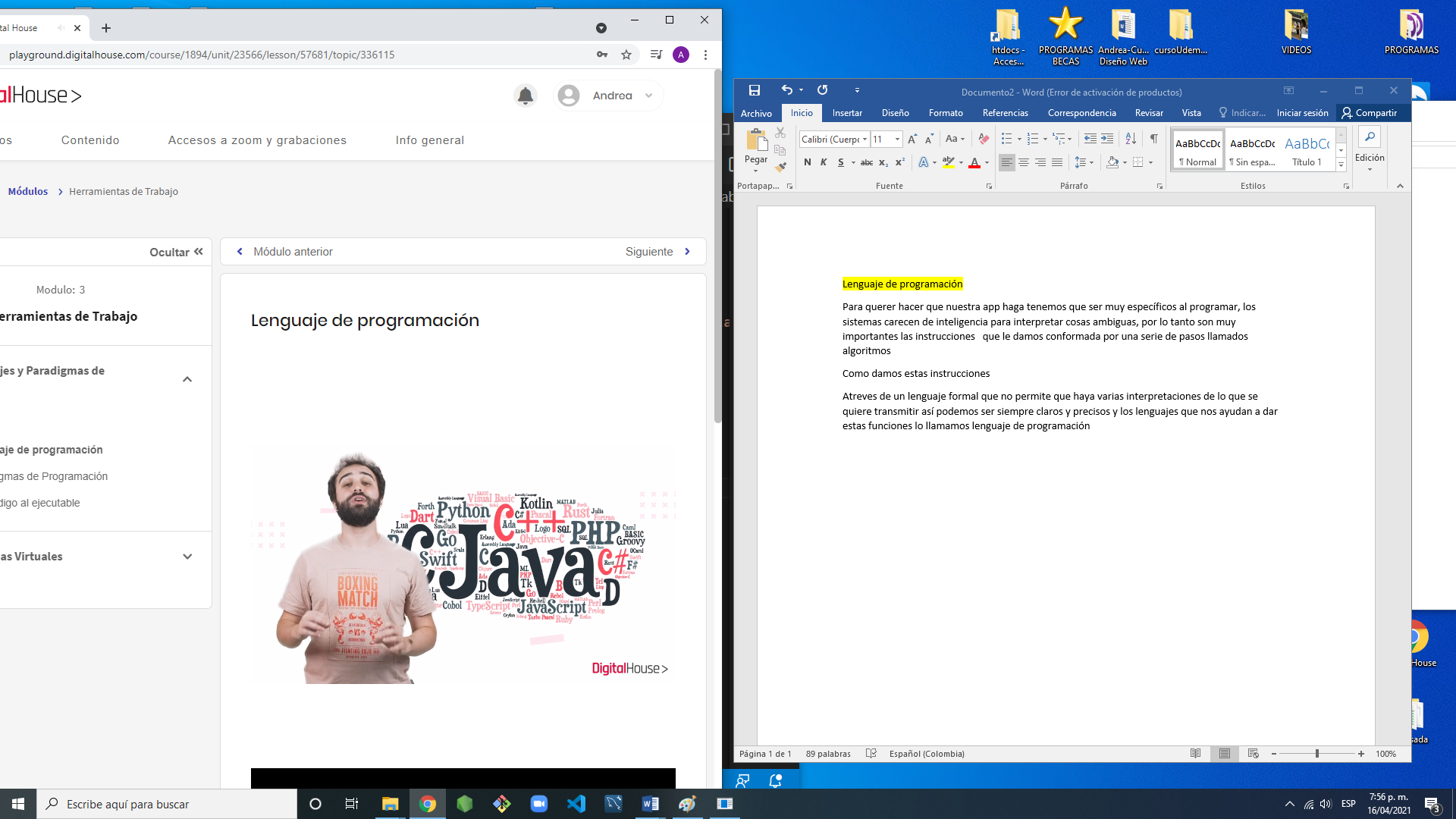
Lenguaje de programación

Para querer hacer que nuestra app haga tenemos que ser muy específicos al programar, los sistemas carecen de inteligencia para interpretar cosas ambiguas, por lo tanto son muy importantes las instrucciones que le damos conformada por una serie de pasos llamados algoritmos

Como damos estas instrucciones

Atreves de un lenguaje formal que no permite que haya varias interpretaciones de lo que se quiere transmitir así podemos ser siempre claros y precisos y los lenguajes que nos ayudan a dar estas funciones lo llamamos lenguaje de programación



Existen muchos lenguajes de programación y cada uno tiene su particularidad

Son dos grandes grupos específicos y generales

-específicos son para resolver problemas puntuales por ejemplo un programa de programación para resolver ejercicios matemáticos

-lenguajes de programación generales: que permite desarrollar una infinidad de aplicaciones distintas casi independiente del contexto como un sitio web de mascotas

Estos lenguajes son parecidos a los nuestros

Existen lenguajes de programación de alto y bajo nivel

Lenguaje de alto nivel: son más cercanos al lenguaje natural que al lenguaje 0 y 1 que permite escribir código de manera más natural y rápida, ejemplo javo scrip donde nos permite ser súper eficientes

Para esto debemos estar atentos a no solo la funcionalidad que queremos desarrollar si no tambien en que hardware

¿hay que saberse todos los lenguajes disponibles?

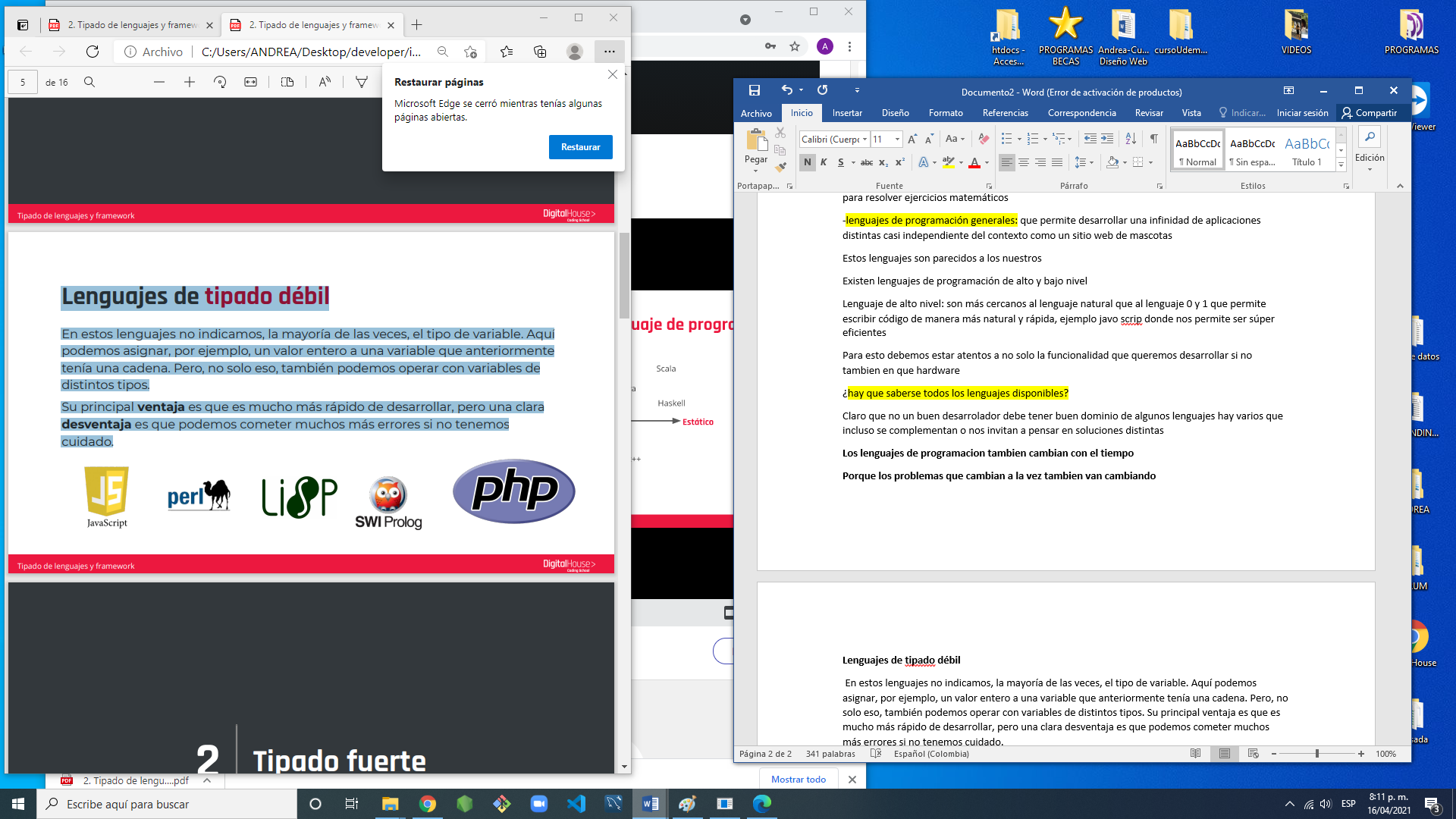
Claro que no un buen desarrolador debe tener buen dominio de algunos lenguajes hay varios que incluso se complementan o nos invitan a pensar en soluciones distintas

**Los lenguajes de programacion tambien cambian con el tiempo**

**Porque los problemas que cambian a la vez tambien van cambiando**

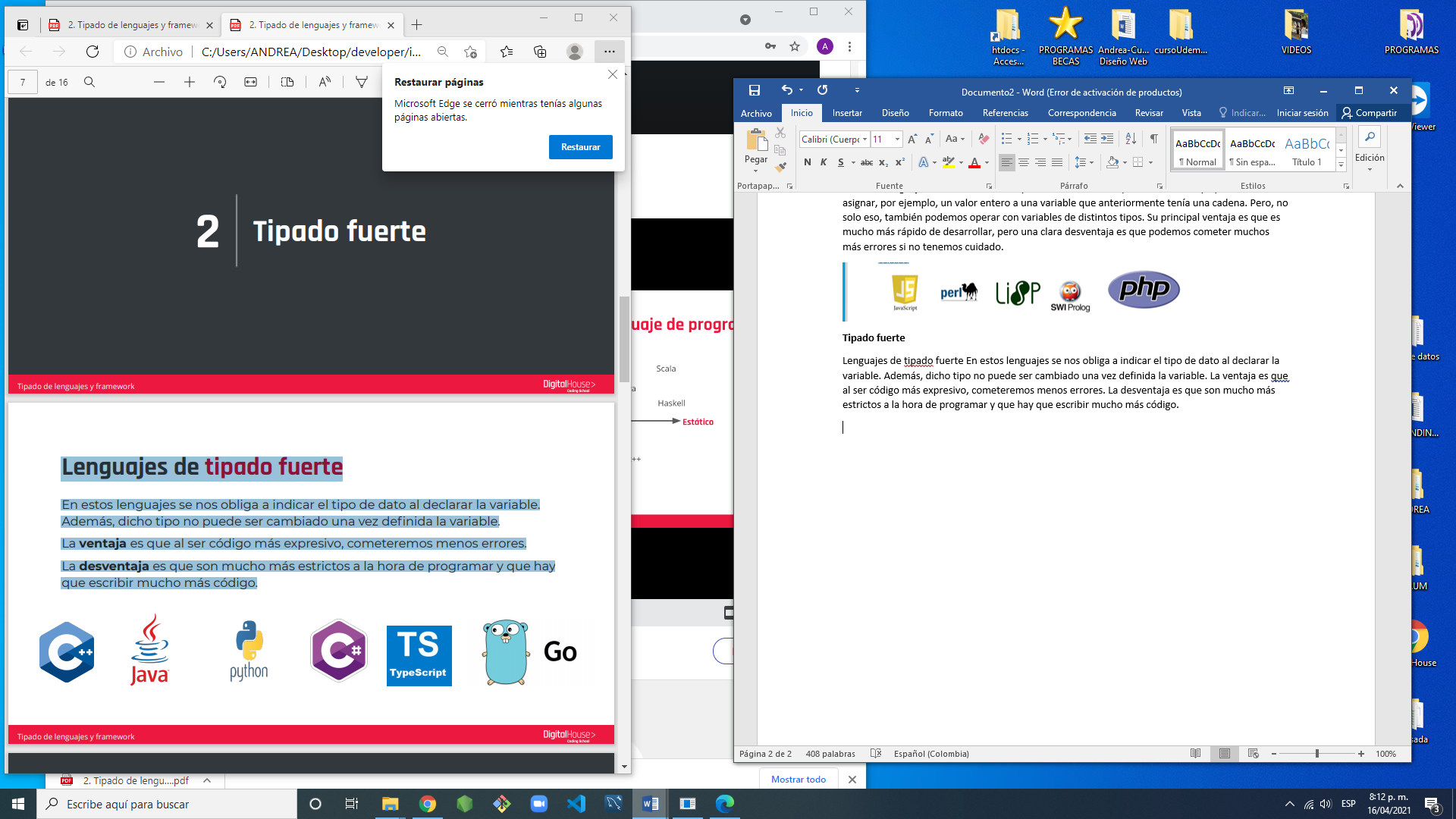
**Lenguajes de tipado débil**

En estos lenguajes no indicamos, la mayoría de las veces, el tipo de variable. Aquí podemos asignar, por ejemplo, un valor entero a una variable que anteriormente tenía una cadena. Pero, no solo eso, también podemos operar con variables de distintos tipos. Su principal ventaja es que es mucho más rápido de desarrollar, pero una clara desventaja es que podemos cometer muchos más errores si no tenemos cuidado.



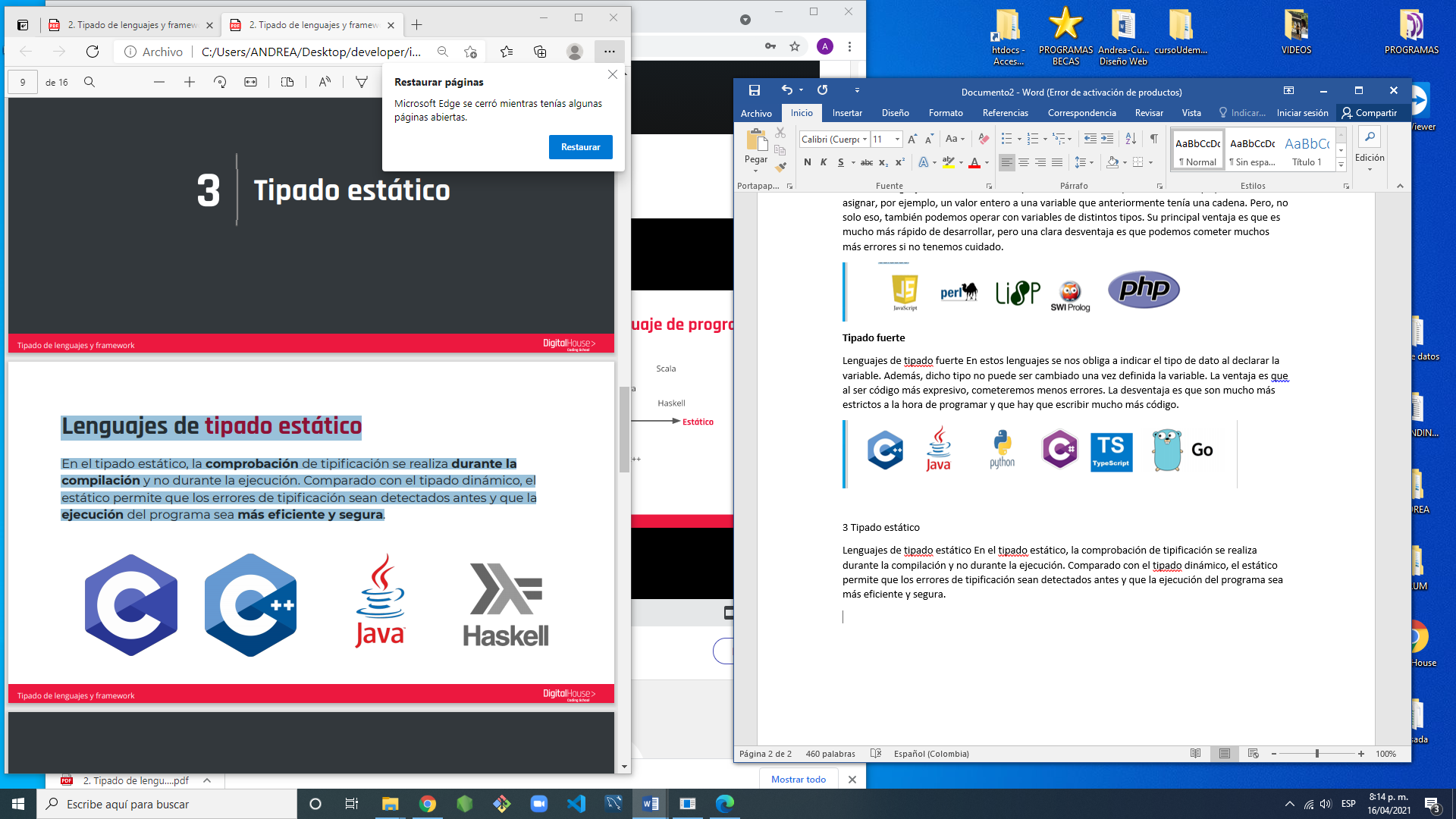
**Tipado fuerte**

Lenguajes de tipado fuerte En estos lenguajes se nos obliga a indicar el tipo de dato al declarar la variable. Además, dicho tipo no puede ser cambiado una vez definida la variable. La ventaja es que al ser código más expresivo, cometeremos menos errores. La desventaja es que son mucho más estrictos a la hora de programar y que hay que escribir mucho más código.



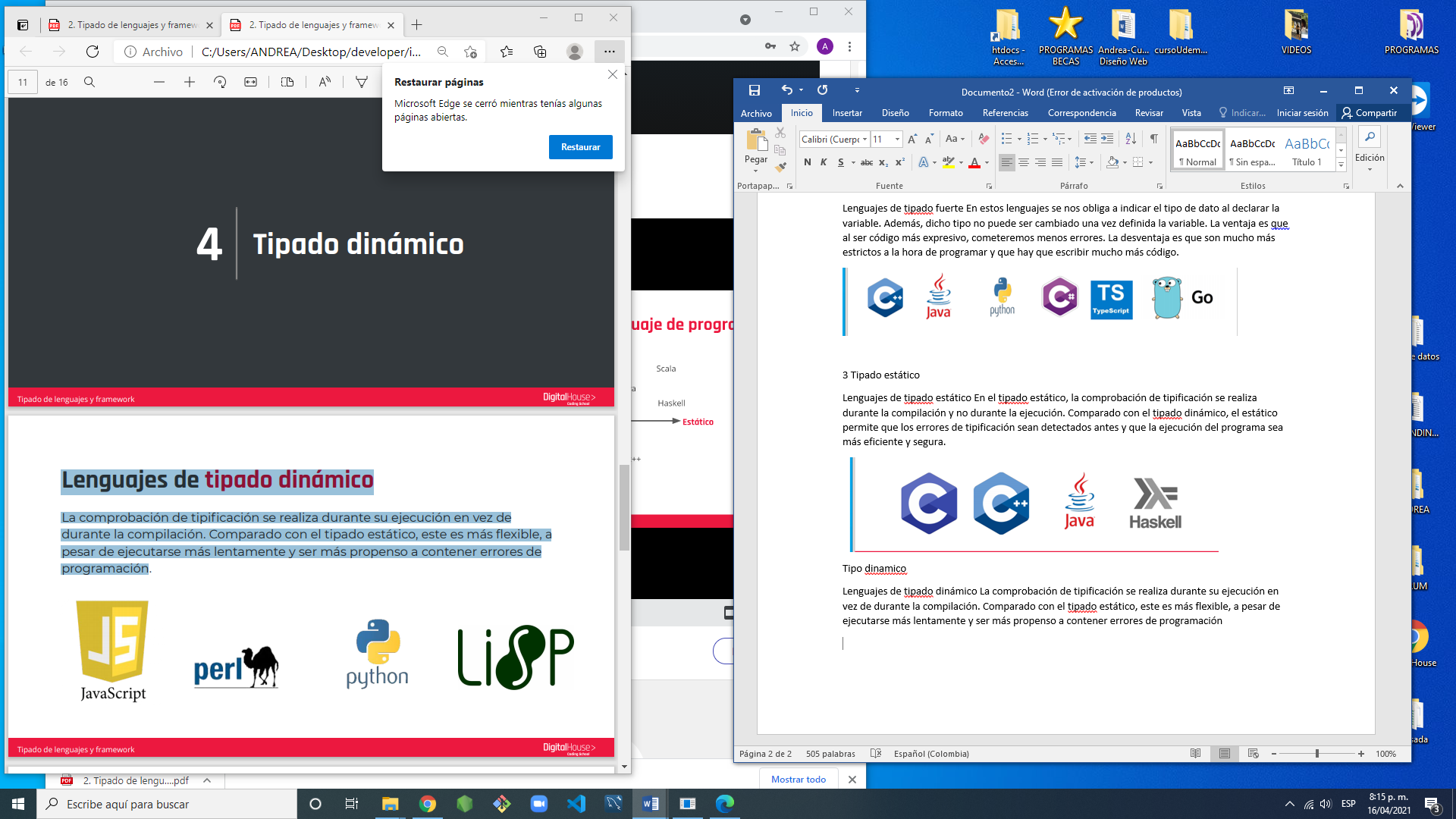
3 Tipado estático

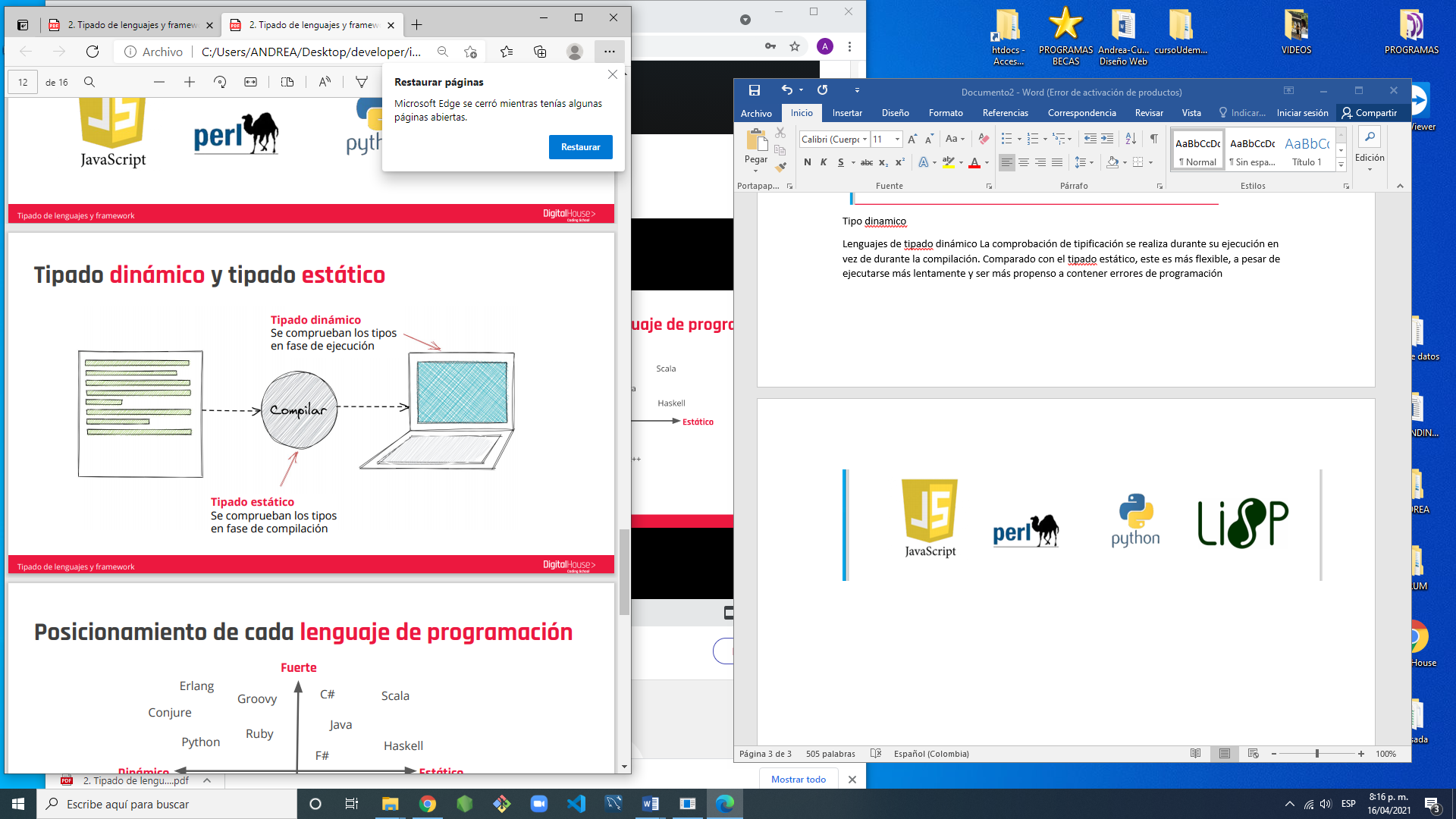
Lenguajes de tipado estático En el tipado estático, la comprobación de tipificación se realiza durante la compilación y no durante la ejecución. Comparado con el tipado dinámico, el estático permite que los errores de tipificación sean detectados antes y que la ejecución del programa sea más eficiente y segura.

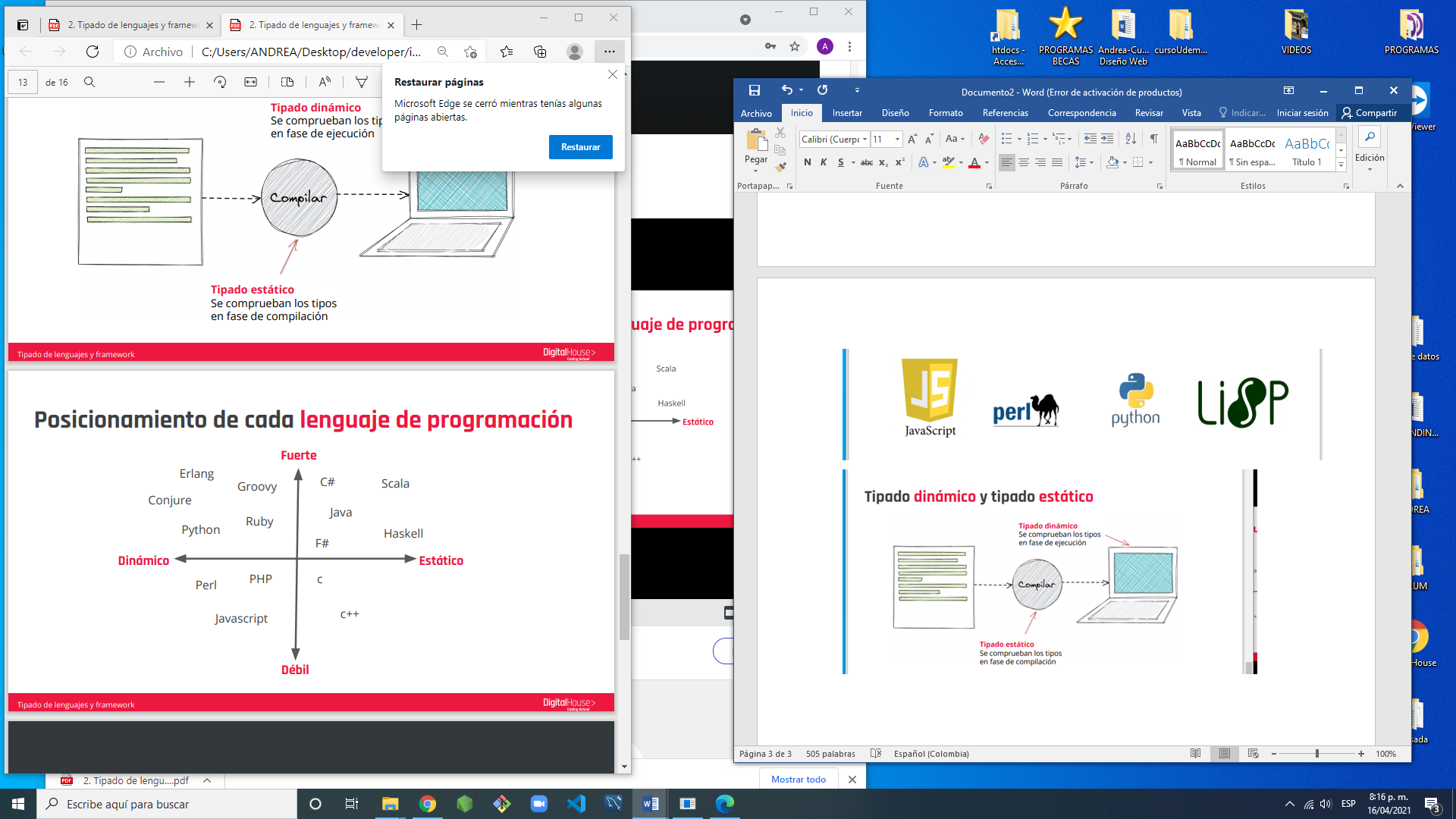


Tipo dinamico

Lenguajes de tipado dinámico La comprobación de tipificación se realiza durante su ejecución en vez de durante la compilación. Comparado con el tipado estático, este es más flexible, a pesar de ejecutarse más lentamente y ser más propenso a contener errores de programación

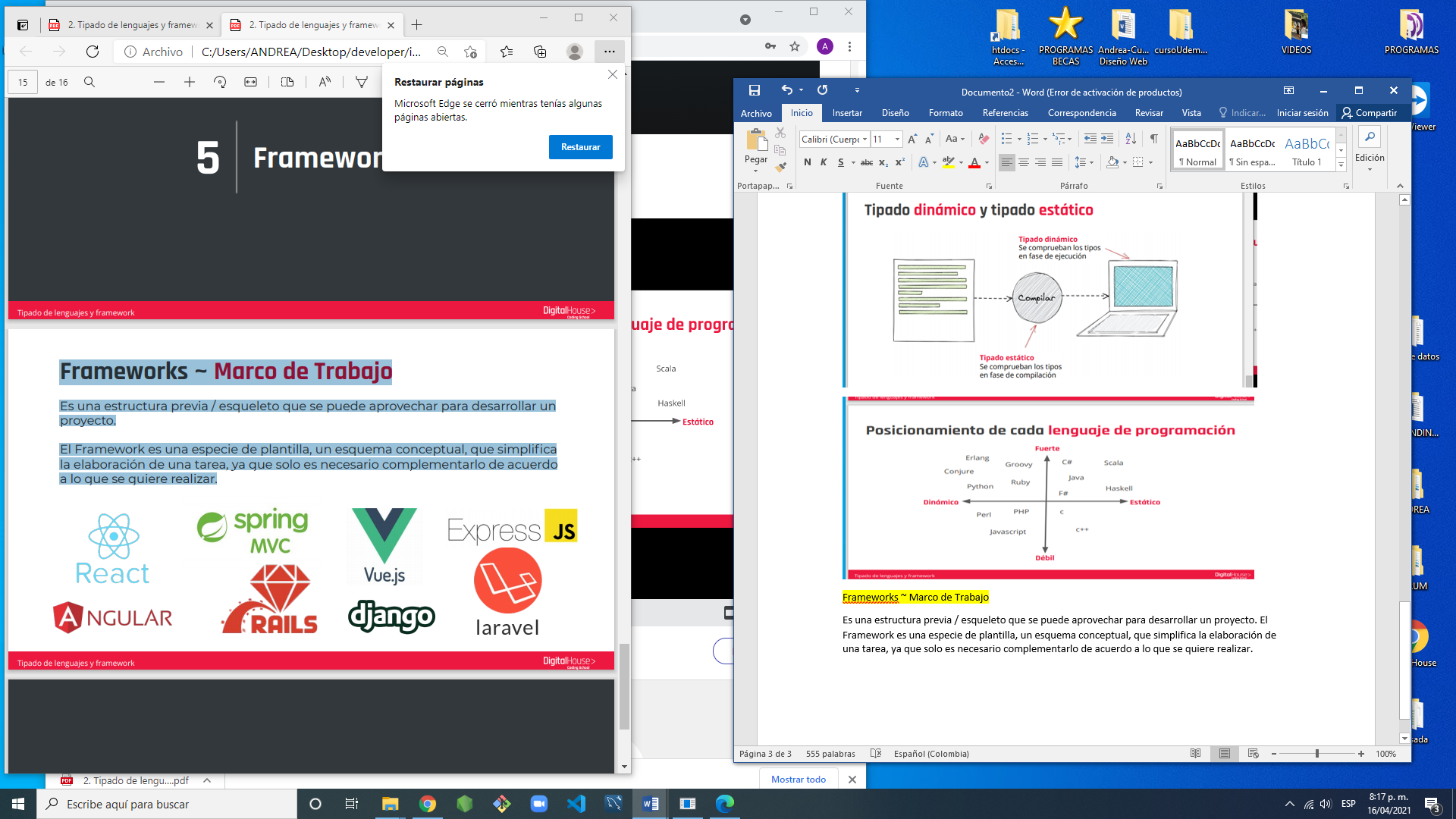






Frameworks ~ Marco de Trabajo

Es una estructura previa / esqueleto que se puede aprovechar para desarrollar un proyecto. El Framework es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea, ya que solo es necesario complementarlo de acuerdo a lo que se quiere realizar.



Cuestionario

1. **¿Cuál es el lenguaje más cercano al lenguaje natural?**

/lenguaje de alto nivel

### Para aprender a programar es necesario aprender todos los lenguajes posibles.

/falso

1. **Los lenguajes de tipiado débil:**

/ nos permite desarrollar mucho mas rápido

1. **En los lenguaje de tipado dinámico, la comprobación de sobre la tipificación de variables se realiza durante su ejecución y no durante la compilación.**

/verdadero

**Paradigmas de programación**

Década de los 60 encontrándonos en el centro de cómputo ole-johan dai y kristen my trabajan en simulaciones de naves para descubrir cómo se comportan cada una de sus partes al modificar sus cualidades

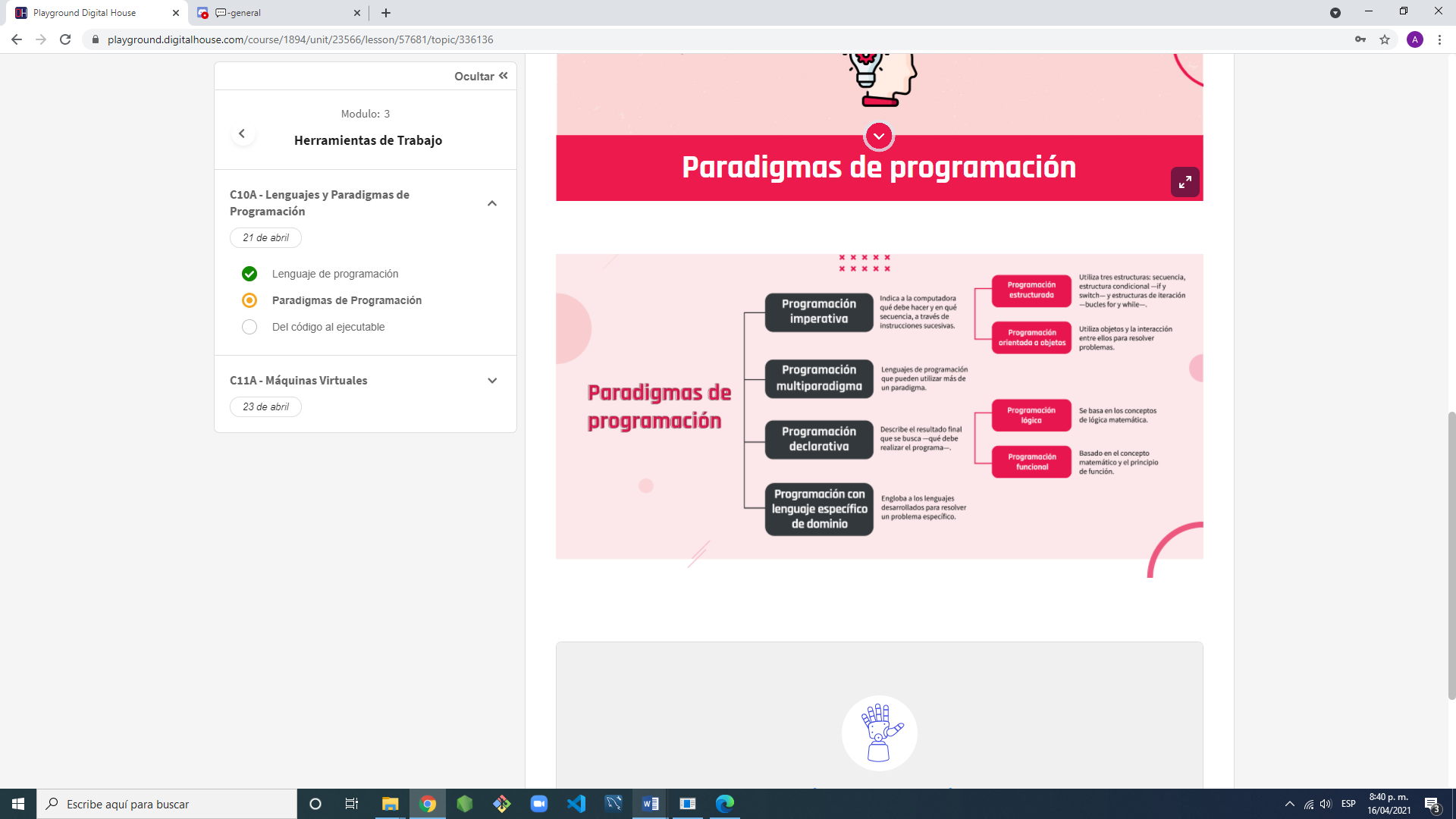
Antes la forma de programar era muy estructura, las simulaciones eran extremadamente complejas y abordarlas con el modelo de complejidad en vez de facilitar el desarrollo ahí es donde ellos dos pensaron el problema para mirar desde otra perspectiva

Ellos decidieron cambiar de modelo y escribir código, pero con otro sed de reglas donde pudieron literal repretar a los dintistos objetos

Cada objeto tenía un estado interno y funciones, como una nave puede tener un encendido y un apagado y una función que permite encenderla o apagarla

Aquí nació el lenguaje de programación simula programación orientada a objetos

**A la forma de pensar bajo un modelo se le llama paradigma**



1. **[Un paradigma es:](https://playground.digitalhouse.com/)**

/ Una forma de pensar bajo un modelo preestablecido.

1. **[¿Qué paradigma describe el resultado final?](https://playground.digitalhouse.com/)**

**/ Programación declarativa.**

1. [**Los lenguajes que encontramos en este paradigma tratan de resolver problemáticas específicas**](https://playground.digitalhouse.com/)

/ Programación con lenguaje específico de dominio.

1. **En el paradigma de programación orientada a objetos el código puede agruparse de tal forma que llegue a representar entidades que interactúen entre sí a través de mensajes.**

/Verdadero.

Del código al ejecutable

Escribimos programas utilizando programas de programación pero usualmente un programa puede estar compuesto por muchos archivos escritos en distintos lenguajes

¿Cómo interpreta la maquina leguajes distintos?

Las instrucciones tienen que pasar de código que escribimos a un código que la maquina entienda esta traducción se llama compilación

La compilación toma todo el código fuente y lo trasforma en el famoso programa ejecutable escrito código maquina

¿El código maquina no puede trabajar en otra? Esto dependerá del código maquina se puede hacer siempre y cuando la máquina que se compile sea similar a la que ejecute

Lo que quiere decir tener una arquitectura CPU similar y sistema operativo similar

**¿no podría mi código ser independiente de la plataforma donde corra?**

Hay dos formas para que los programas sean entendidos y ejecutados por una maquina

* Maquina virtuales
* Interpretes

**Del código al ejecutable**

El primer compilador de la historia, el A-0, fue desarrollado en 1952 por la científica en computación Grace Hopper.

**Código fuente**

El primer compilador de la historia, el A-0, fue desarrollado en 1952 por la científica en computación Grace Hopper.

**Código de máquina**

El código de máquina es una secuencia de sentencias en lenguaje de máquina o binario. Es el resultado obtenido después de que el compilador convierta el código fuente en un lenguaje que pueda ser comprendido por el procesador.

**Compilador**

Es una aplicación traduce (compila) el código fuente en un código que el procesador puede comprender y ejecutar. Este código de máquina se almacena en forma de archivo ejecutable.

**Intérprete**

Traduce el código fuente línea a línea y lo ejecuta directamente. El proceso de traducción funciona mucho más rápido que en un compilador, pero la ejecución es más lenta y se necesita una gran cantidad de memoria.

1. [¿Qué hace la compilación?](https://playground.digitalhouse.com/)

Toma el código fuente y lo transforma en un programa ejecutable escrito en código máquina.

1. **[¿Qué podemos utilizar para que nuestro código sea independiente de la plataforma donde corra?](https://playground.digitalhouse.com/)**

Podemos utilizar máquinas virtuales o intérpretes.

1. **[Un intérprete analiza línea por línea un código fuente en el momento de su ejecución.](https://playground.digitalhouse.com/)**

Verdadero

1. [La principal ventaja de compilar un código en una máquina virtual, es la alta performance.](https://playground.digitalhouse.com/)

Falso.