

~~HENRY~~



# Procesamiento del lenguaje natural

Data Science





# Agenda



- Procesamiento del Lenguaje Natural
- Flujo de Trabajo en PLN
- Expresiones Regulares
- Pre-procesamiento
- Feature engineering



# **OBJETIVOS DE LA CLASE**

***Al finalizar esta lecture estarás en la capacidad de...***

- Entender los principales métodos de Procesamiento del Lenguaje Natural



Al **finalizar** cada uno de los temas,  
tendremos un **espacio de consultas**.



Hay un **mentor** asignado para  
responder el **Q&A**.

¡Pregunta, pregunta, pregunta! :D



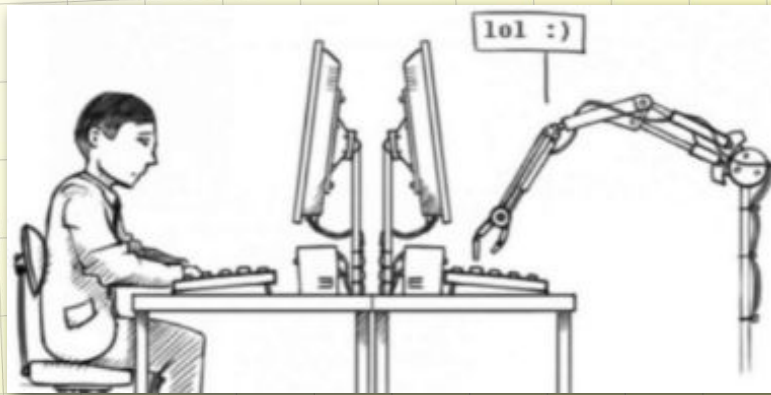
# Procesamiento del **Lenguaje Natural**





//

“Si una persona, en una conversación con una máquina, no es capaz de distinguirlo de un humano, ésta es aprobada como inteligente”



Alan Turing,  
Prueba de Turing,  
1950



# ¿Qué es?

Es el proceso de **analizar colecciones** de materiales de texto con el objeto de capturar los temas y conceptos clave y **descubrir las relaciones ocultas** y **las tendencias existentes** sin necesidad de conocer las palabras o los términos exactos que los autores han utilizado para expresar dichos concepto.





# Preguntas introductorias

- ¿Qué conceptos aparecen juntos?
- ¿A qué otras cosas están vinculados?
- ¿Qué categorías de nivel superior pueden crearse a partir de la información extraída?
- ¿Qué es lo que predicen los conceptos o las categorías?
- ¿Cómo predicen el comportamiento los conceptos o las categorías?





# Proceso

- **Identificar el texto** en el que se va a realizar la minería. Preparar el texto para el proceso de minería.
- **Minar el texto** y extraer datos estructurados. Aplicar los algoritmos de minería de textos al texto de origen.
- **Crear modelos de categoría y concepto**. Identificar los conceptos clave y/o crear categorías. El número de conceptos que se devuelven de los datos no estructurados suelen ser muy alto. Identificar los mejores conceptos y categorías para puntuar.
- **Analizar los datos estructurados**. Emplear técnicas de minería de datos convencionales, como el clúster, la clasificación y el modelado predictivo, con el objeto de descubrir las relaciones entre los conceptos.



# Expresiones Regulares

Una expresión regular es una secuencia de caracteres que determina un patrón de búsqueda.

Es un lenguaje muy flexible que sirve para identificar y extraer información de un cuerpo de caracteres no estructurado.

Elemento	Descripción
()	Capturing group
\w	carácter alfanumérico
\d	dígito
\s	espacio en blanco
[]	conjunto
(?:)	Non-capturing group
.	cualquier cosa menos \n
[m-z3-9]	rangos
+	uno o más del elemento anterior
*	cero o más del elemento anterior
{4,}	cuatro o más del elemento anterior



# ¡Atención!

El problema principal en la administración de todos estos datos de texto no estructurados radica en la **ausencia de reglas estándares** para escribir texto y que el ordenador pueda entenderlo.

Existen diversos métodos automáticos diferentes para extraer conceptos a partir de información no estructurada. Estos métodos pueden desglosarse en dos tipos: **lingüísticos** y **no lingüísticos**.



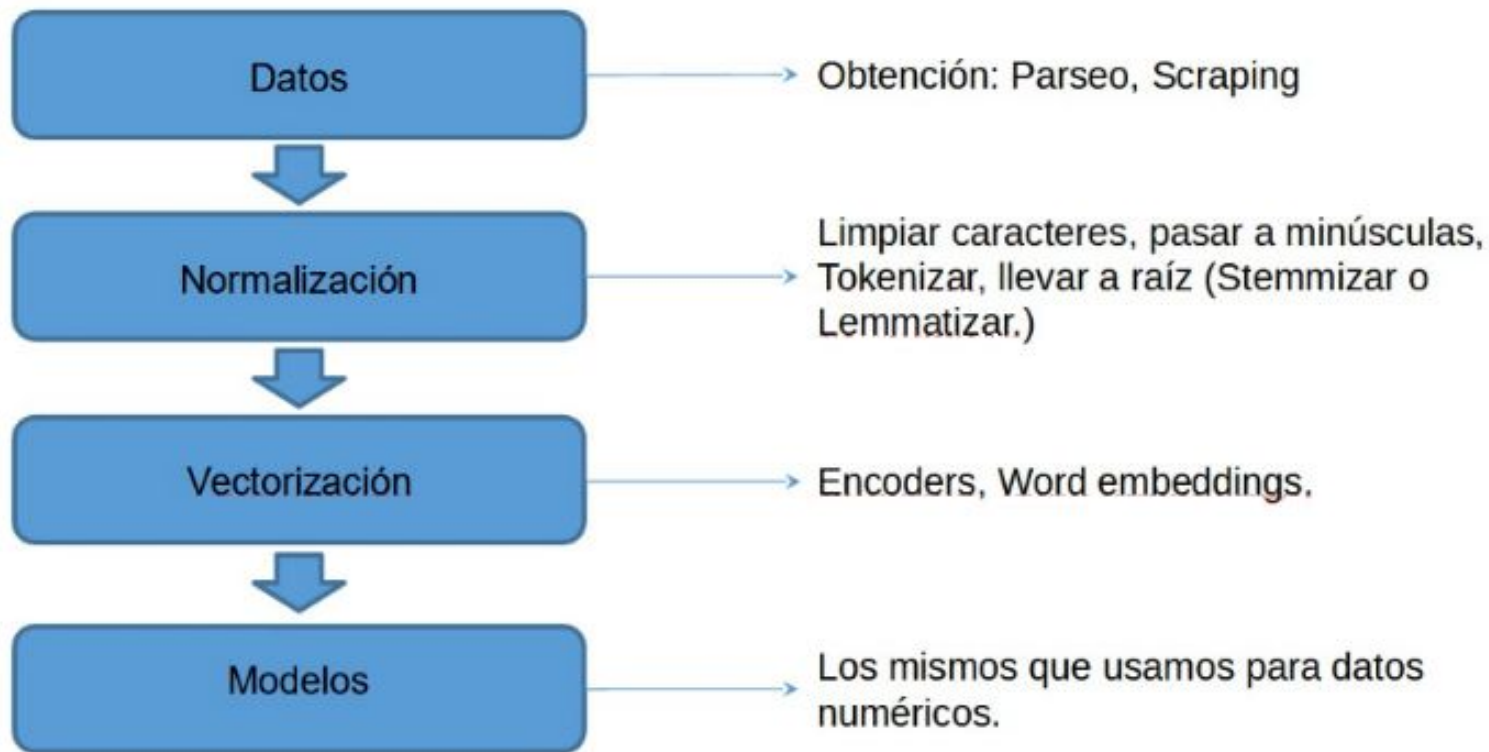


# Corpus

El corpus es una colección de textos escritos, especialmente las obras completas de un autor en particular o un cuerpo de escritura sobre un tema en particular (grupo de documentos, grupo de textos, grupo de tweets, etc.).

En resumen, es el conjunto de todos los **elementos que queremos analizar**.

# Flujo de Trabajo





# Tokenización

Consiste en dividir una oración compleja en palabras, comprender la importancia de cada palabra con respecto a la oración y finalmente producir una descripción estructural en una oración de entrada.







# **Pre- procesamiento**





# Preprocesamiento

1. **Signos de puntuación:** Eliminación de todos los signos de puntuación. Hay diferentes bibliotecas que se utilizan para eliminar los signos de puntuación.
2. **Números:** Eliminación de todos los números o transformación en palabras de los datos de texto que incluyen línea de tiempo, fechas, direcciones IP, etc.
3. **Minúsculas:** Conversión de todas las letras a minúsculas.





# Preprocesamiento

4. **Stop Words** (eliminación de palabras innecesarias): Por lo general, incluye la eliminación de palabras de alta frecuencia como (a, an, the, all pronombres, etc.).

No existe una lista universal de palabras vacías, pero se puede usar un conjunto estándar de palabras vacías en inglés de la biblioteca nltk. Además, se pueden agregar palabras vacías específicas del dominio.

5. **Espacios en blanco**: Eliminación de los espacios en blanco que pueden estar presentes al principio o al final de las oraciones/palabras o puede haber espacios adicionales en cualquier parte de la oración.



# Preprocesamiento


6. **Derivación** (stemming): Reducción de las fichas a formas de raíz para reconocer variaciones morfológicas. Elimina ciegamente cualquier prefijo o posfijo de manera iterativa.
7. **Lemmatization**: Conversión de formularios variantes a los formularios base. Esto impacta directamente en el tamaño del vocabulario. Esto evita la duplicación de datos al vincular palabras a la palabra raíz. Por ejemplo, "soy", "son" están vinculados a "ser". Para lograr esto, necesitamos una lista de reglas gramaticales y una lista de palabras irregulares.



# Preprocesamiento

## Stemming vs Lemmatization

change  
changing  
changes  
changed  
changer



chang

change  
changing  
changes  
changed  
changer



change



# Lematización

- **Lema**: forma canónica del diccionario; lexema; raíz o tema.
- **Métodos**: Porter, entropía, etc.

palabras gráficas	lemas
somos, soy, eres, fueron	ser
voy, fui, irán	ir
poder, pudieron, podrán	poder



# Tokenización: ejemplo

"In Brazil they drive on the right-hand side of the road. Brazil has a large coastline on the eastern side of South America"



Se "tokeniza" la frase anterior (palabras, comas y signos de puntuación)

['In', 'Brazil', 'they', 'drive', 'on', 'the', 'right-hand', 'side', 'of', 'the', 'road', '.', 'Brazil', 'has', 'a', 'large', 'coastline', 'on', 'the', 'eastern', 'side', 'of', 'South', 'America']



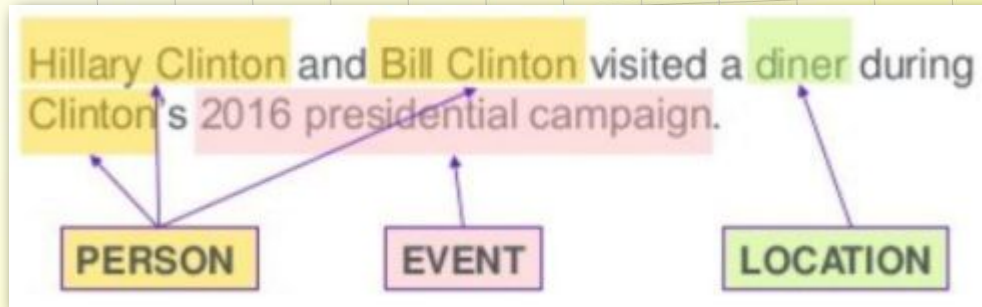
La frase tokenizada queda expresada como diccionario (llave: palabra, valor: frecuencia de la palabra)

FreqDist({'the': 3, 'Brazil': 2, 'on': 2, 'side': 2, 'of': 2, 'In': 1, 'they': 1, 'drive': 1, 'right-hand': 1, 'road': 1, ...})



# Librerías adicionales

- Hay librerías que nos ayudan a ver desde el tipo de vista lingüístico, qué función tiene la palabra (verbo, pronombre, etc.)



- También nos dicen qué significan contextualmente (persona, evento, etc.)





# Feature engineering

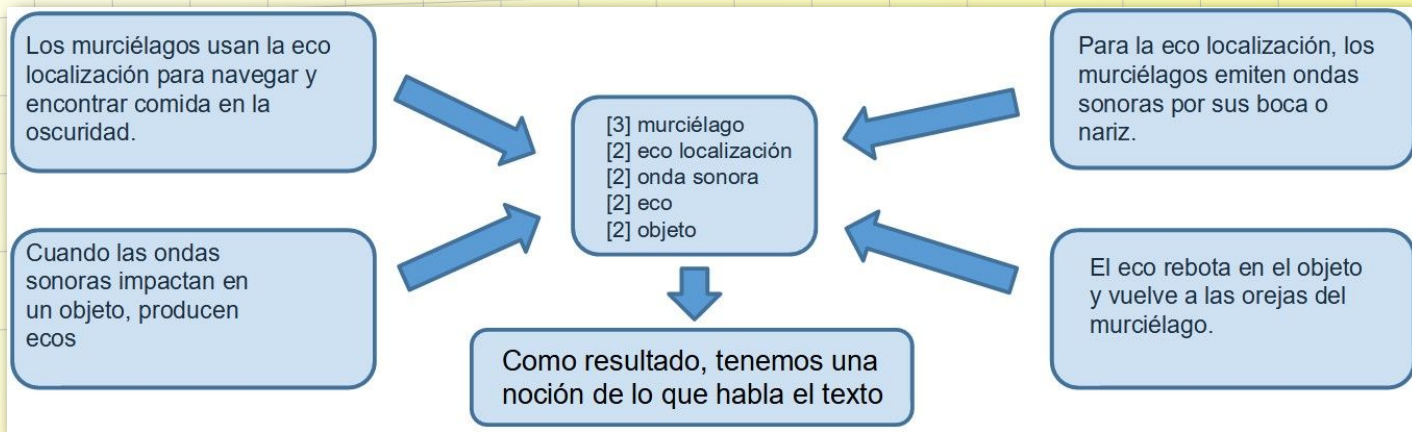




# N-Gramas

Hay palabras que cobran sentido cuando se las agrupa con otras, por ejemplo "eco localización". Además de cada palabra por separado, se agregan los grupos de 2 (ó N) palabras contiguas a nuestro vector de Features:

```
CountVectorizer(max_features=max_features, stop_words="english",  
                ngram_range=(1, 2))
```







# Bag of words

Cada documento va a ser un registro y palabra va a ser un atributo del dataset.

Básicamente, con bag of words, se le asigna un **número a cada palabra** de acuerdo a la **frecuencia** de la misma en un texto.

Doc 1: I love dogs.  
Doc 2: I hate dogs and knitting.  
Doc 3: Knitting is my hobby and passion.



	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1



# TF – IDF (Term Frequency – Inverse Document Frequency)

Se busca diferenciar cada documento (frase) por las palabras que lo componen, asumiendo que las palabras que están en TODOS ellos no aportan información.

Hay que medir no sólo cuanto aparece una palabra en una instancia (documento), sino también qué tan frecuente es esa palabra en todo el corpus del texto.



# TF (Term Frequency)

- sólo cuenta el número de palabras que aparecen en cada documento.
- El problema principal con esta frecuencia de términos es que dará más peso a los documentos más largos.
- La frecuencia de término es básicamente la salida del modelo Bag of words.

$$\text{TF}(\text{term}, \text{doc}) = \frac{\text{Número de veces que el } \textit{term} \text{ aparece en el } \textit{doc}}{\text{Número de } \textit{terms} \text{ diferentes en el } \textit{doc}}$$



# IDF (Inverse Document Frequency)

- Mide la cantidad de información que proporciona una palabra determinada en todo el documento.
- IDF es la relación inversa escalada logarítmicamente del número de documentos que contienen la palabra y el número total de documentos.

$$\text{IDF}(\text{term}, \text{corpus}) = \text{Log} \left( \frac{\text{Número total de docs}}{\text{Número de docs que tienen el term}} \right)$$



# Volviendo al ejemplo...

Volviendo al ejemplo que comenzamos en Bag of words, vemos la frecuencia inversa de cada palabra, considerando los tres documentos.

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1



	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	<b>0.48</b>	0.18							
Doc 2	0.18		0.18	<b>0.48</b>	0.18	0.18				
Doc 3					0.18	0.18	<b>0.48</b>	<b>0.95</b>	<b>0.48</b>	<b>0.48</b>



# DF y TF-IDF

- **Document Frequency**: Fracción de todos los documentos en el corpus que contienen el término.

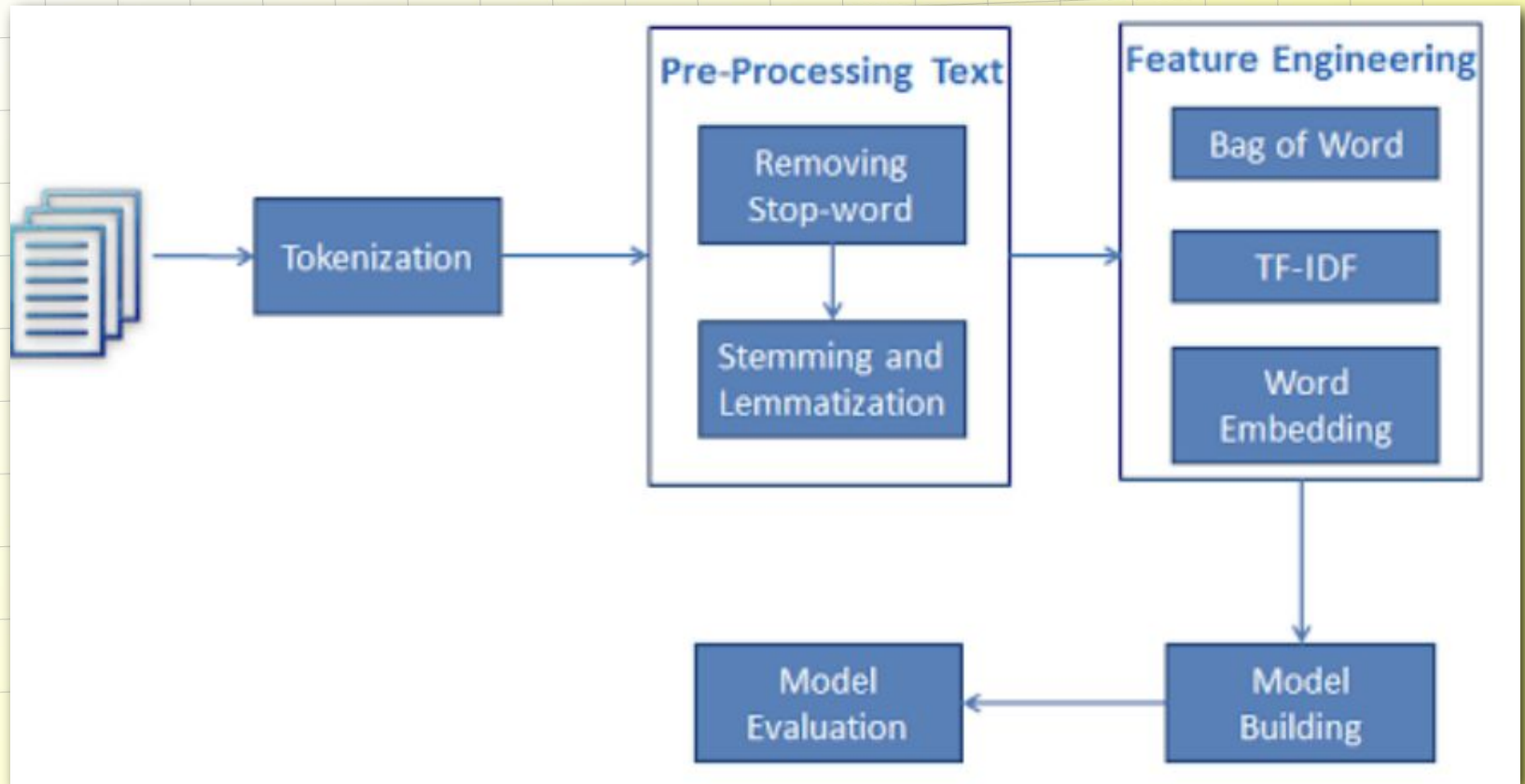
$$\mathbf{DF}(\text{term}, \text{corpus}) = \frac{\text{Número de docs que tienen el term}}{\text{Número total de docs}}$$

- **TF-IDF**: Producto del valor TF por el de IDF.

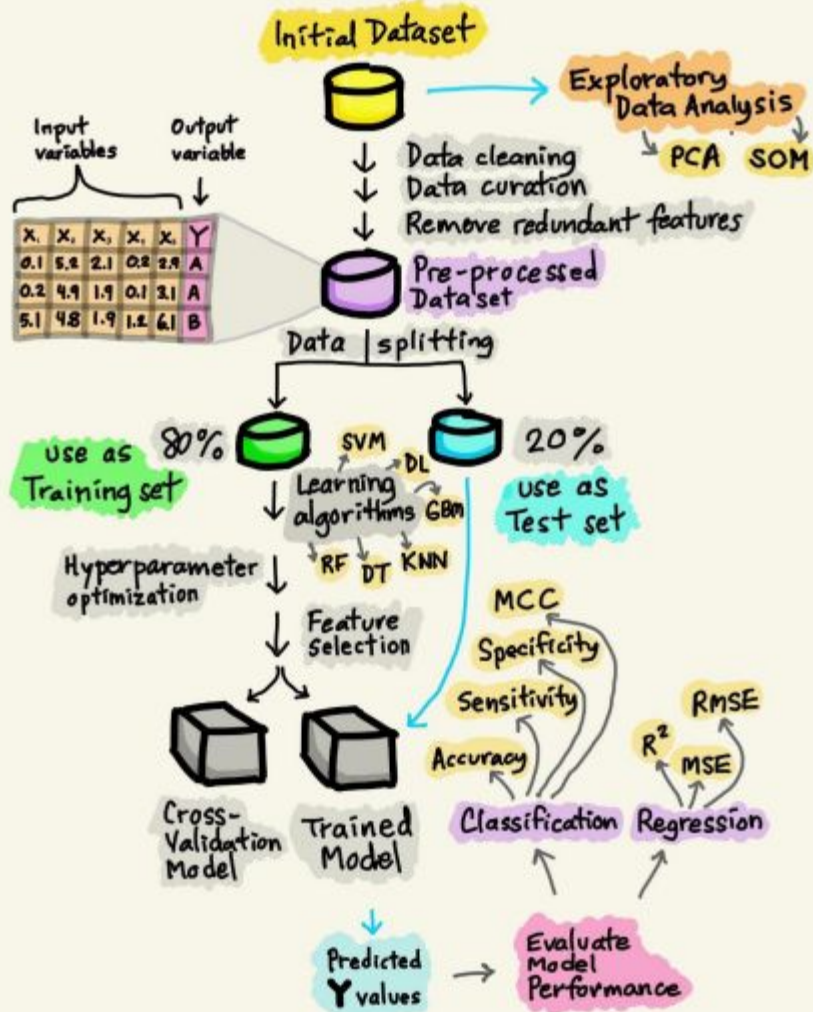
$$\mathbf{TF-IDF}(\text{term}, \text{doc}) = \frac{\text{Número de veces que el term aparece en el doc}}{\text{Número de terms diferentes en el doc}} \times \text{Log} \left( \frac{\text{Número total de docs}}{\text{Número de docs que tienen el term}} \right)$$



# SÍNTESIS



# SÍNTESIS





**¿PREGUNTAS?**



# ¿Alguien dijo Homework?



~~HENRY~~



Próxima lecture

# Redes neuronales





# ¡Feedback!

Click on me



Dispones de un **formulario** en:



Homeworks



Guías de clase



Slack

# HENRY

