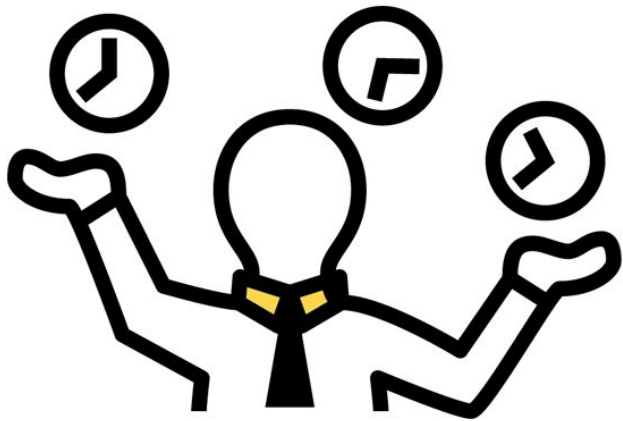


Modelamiento del idioma español en bases relacionales

Daniel Felipe Montenegro

Contenido



- Origen del proyecto
- Herramientas utilizadas
- Estructura de la base de datos
- Funcionalidades
- Uso de la herramienta

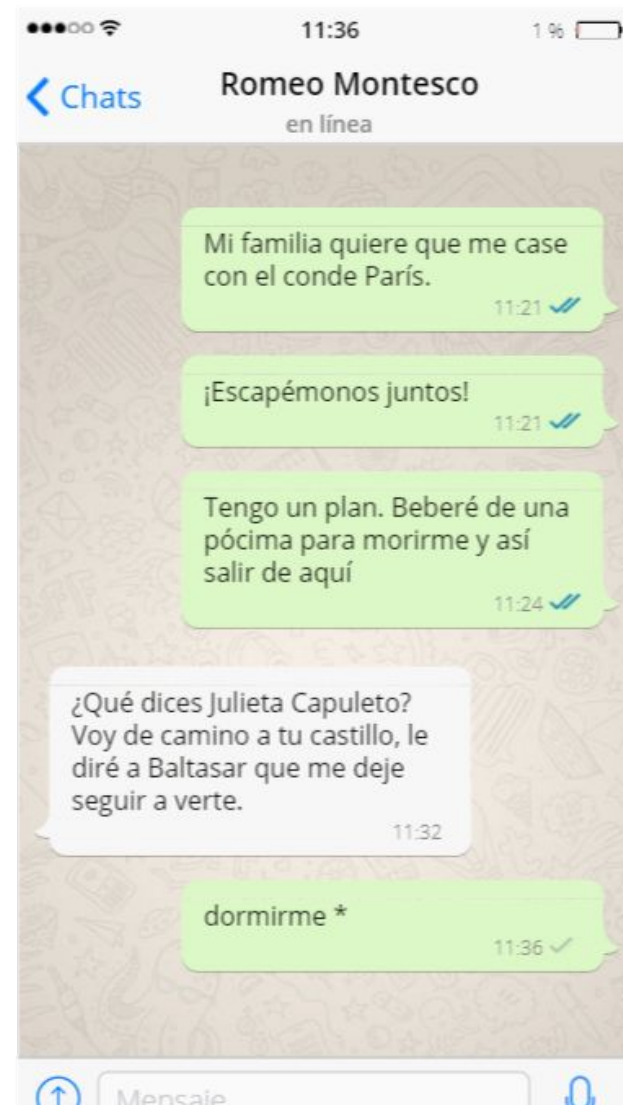
¡Todo nace

de una idea!

La palabra que hace la diferencia

Imagina que esos dos son Romeo y Julieta, el error en esa palabra hizo que sucediera el fatal desenlace que conocemos hoy en día.

Algunas veces las herramientas nos facilitan la vida, otra veces no lo hacen. Es por eso que el objetivo de este proyecto es crear una herramienta que haga de esta historia una con final feliz.



¿De dónde nace el error?

El error en la corrección de la palabra en la conversación no es una mera casualidad, realmente nace del modelo en el cual se esta basando el autocorrector para realizar las correcciones.



Lamentablemente ese modelo es ajeno a nosotros y es propiedad de quien nos esta prestando el servicio de escritura y corrección, es por esa razón que no podemos modificarlo ni personalizarlo. Una solución para esta limitación es crear nuestro propio modelo que se adapte de acuerdo a nuestro propio lenguaje.

“Los accidentes no existen”

¿Qué haremos?

El objetivo de este proyecto es modelar por medio de bases relacionales el idioma español, el programa principalmente será un analizador de texto el cual podrá recibir cualquier tipo de escrito; esto quiere decir que podrá recibir textos como un ensayo, un cuento, un párrafo, una frase o cualquier conjunto de palabras. La finalidad será analizar su orden, composición y estructura, para luego construir de acuerdo con estos parámetros un modelo el cual podrá tener futuras aplicaciones en áreas como ***Procesamiento del Lenguaje Natural y Redes Neuronales***.



¿Qué esperamos?



El resultado de este proyecto será una librería de código abierto para el análisis de texto en ***Python***, la librería podrá ejecutarse como un programa el cual permitirá crear scripts para agregar textos o frases a la base de datos y también visualizar su contenido. La biblioteca permitirá hacer uso de su información como de sus métodos para implementarlos en cualquier otro programa.



Herramientas utilizadas

Librerías Python

La librería para su correcto funcionamiento hace uso de otras cuatro librerías existentes en Python, estas librerías permiten la interconectividad de la herramienta, a continuación se detalla el uso de cada librería:

```
import os
import time
import requests
import mysql.connector
from bs4 import BeautifulSoup
```

La librería **Os** y la librería **Time** vienen por defecto distribuidas con **Python**, el resto se instala por medio de **pip**.

- **Os:** Permite la lectura y escritura de archivos
- **Time:** Genera pequeños retardos para que la interfaz sea mas amigable
- **Requests:** Crea las solicitudes a las paginas web
- **Mysql.connector:** Es la librería oficial de Mysql para conectarse desde Python
- **BeautifulSoup:** Facilita el análisis de documentos html

Corpus de Referencia del Español Actual

El **Corpus de Referencia del Español Actual (CREA)** provee un listado de frecuencias de todas palabras existentes en la lengua, un vistazo de la parte superior de la lista es este:

Orden	Frec.absoluta	Frec.normalizada
1. de	9,999,518	65545.55
2. la	6,277,560	41148.59
3. que	4,681,839	30688.85
4. el	4,569,652	29953.48
5. en	4,234,281	27755.16



REAL ACADEMIA ESPAÑOLA: Banco de datos (CREA) [en línea]. *Corpus de referencia del español actual*. <<http://www.rae.es>>

Web Scraping



Por defecto la base de datos se distribuirá con las primeras **121000 frecuencias del CREA**, sin embargo, para poner a prueba la herramienta procesamos con ella **3276 textos** (cuentos de diversos autores en el idioma español) que contenían **852302 frases** los cuales fueron recolectados de la pagina web www.ciudadseva.com, esto gracias a que en su documento **robots.txt** especificaba al momento del raspado que era posible extraer este contenido.

CIUDAD SEVA

Casa digital del escritor Luis López Nieves

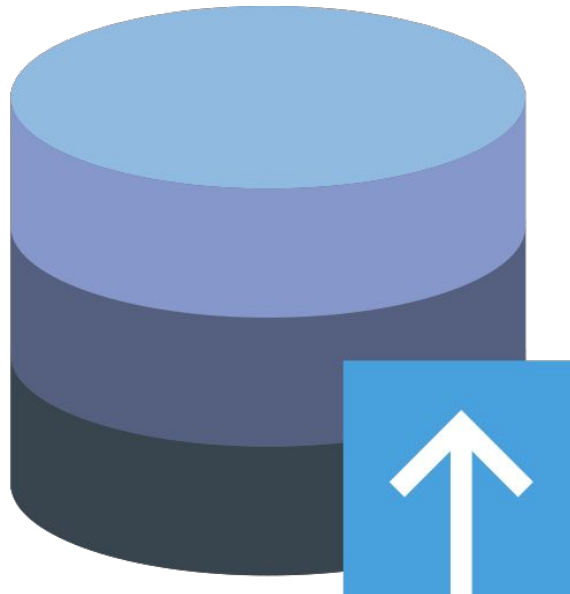
Esta biblioteca se formo por medio de una aplicación que esta integrada en la librería para hacer **Web Scraping** de cualquier pagina web.

MYSQL

Es el gestor de base de datos que utiliza el proyecto, los scripts de creación de la base de datos que se distribuyen con esta librería están implementados especialmente para su uso en MYSQL.



Estructura Base de datos



Resumen de la base de datos

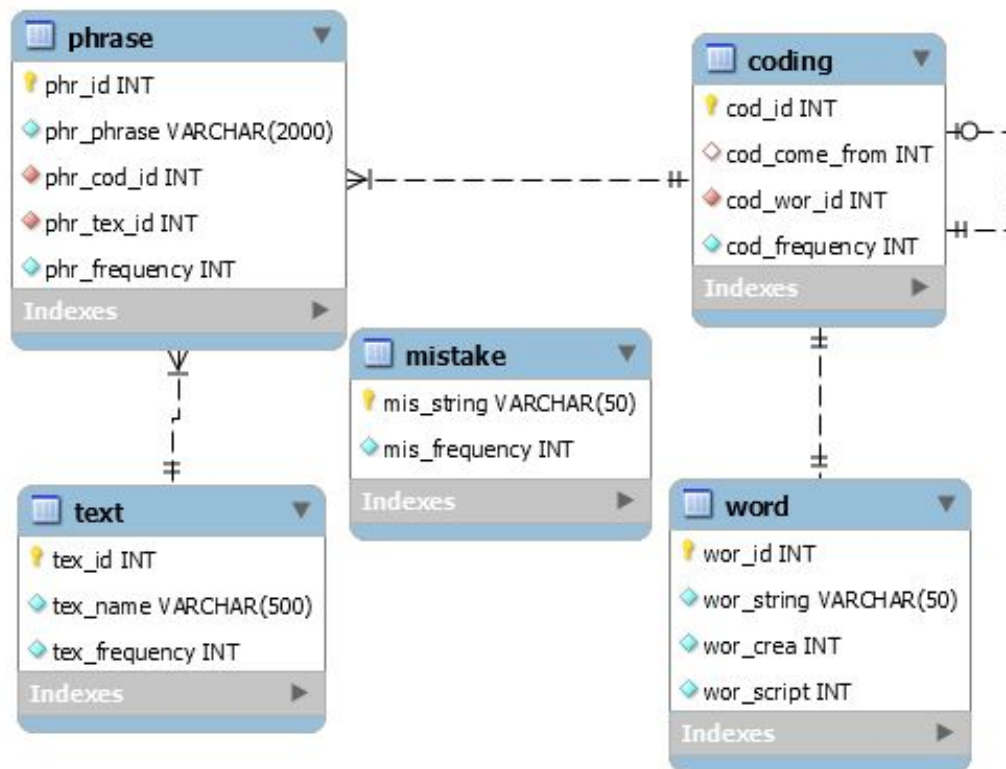
La base de datos que se distribuye con la librería por defecto esta compuesta de la siguiente manera:

- **Entidades (5):** word, coding, phrase, mistake y text
- **Triggers (1):** frequent_mistake
- **Views (7):** crea, scripts, total_words, mistakes, codings, texts y phrases
- **Stored Procedures (7):** insert_crea, insert_word, insert_mistake, insert_coding, insert_text, insert_phrase y id_word
- **Users (2):** root y guest
- **Values (121000):** Las 121.000 primeras frecuencias del CREA



Modelo de la base de datos

La base de datos que se distribuye por defecto es modificable, sin embargo, el script de creación incluido solo plantea la necesidad de cinco entidades para su correcto funcionamiento. Las entidades son las siguientes:



Entidades

phrase

	phr_id	phr_phrase	phr_cod_id	phr_tex_id	phr_frequency
▶	1	sombra imborrable del josco sobre la loma que d...	13	1	1
	2	la cabeza erguida	16	1	1
	3	las aspas filosas estoqueando el capote en san...	30	1	1
	4	la carrilluda en sombras	33	1	1
	5	el andar lento y rítmico	38	1	1
	6	la baba gelatinosa le caía de los belfos negros y...	48	1	1
	7	dejando en el verde enjoyado estela plateada d...	57	1	1
	8	era hosco por el color y por su carácter reconce...	69	1	1
	9	de peleador incansable	72	1	1
	10	cuando sobre el lomo negro del cerro farallón la...	87	1	1

coding

	cod_id	cod_come_from	cod_wor_id	cod_frequency
▶	1	NULL	1637	8
	2	1	32798	1
	3	2	10	1
	4	3	121004	1
	5	4	32	1
	6	5	2	1
	7	6	14927	1
	8	7	3	1
	9	8	7458	1
	10	9	4	1

mistake

	mis_string	mis_frequency
*	NULL	NULL

text

	tex_id	tex_name	tex_frequency
▶	1	Abelardo Díaz Alfaro - El Josco	224
	225	Abelardo Díaz Alfaro - Santa Clo va a La Cuchilla	166
	391	Adolfo Bioy Casares - El caso de los viejitos vola...	150
	541	Adolfo Bioy Casares - En memoria de Paulina	769
	1310	Adolfo Bioy Casares - La francesa	27
	1337	Adolfo Bioy Casares - La salvación	20
	1357	Adolfo Bioy Casares - La trama celeste	1554
	2911	Adolfo Bioy Casares - Las vísperas de Fausto	140
	3051	Adolfo Bioy Casares - Margarita o el poder de la...	85

word

	wor_id	wor_string	wor_crea	wor_script
▶	1	de	9999518	307276
	2	la	6277560	223394
	3	que	4681839	198327
	4	el	4569652	172276
	5	en	4234281	141858
	6	y	4180279	208262
	7	a	3260939	151158
	8	los	2618657	83058
	9	se	2022514	97289
	10	del	1857225	48950

Triggers

La base de datos tiene un trigger el cual se encarga de verificar cada vez que se inserta un error si la frecuencia del mismo es superior a **21**, si un error se repite mas de **21** veces el trigger se encarga de agregarlo como una palabra a la tabla **word** en la columna **script**.

```
DELIMITER $$  
CREATE TRIGGER frequent_mistake AFTER UPDATE ON mistake FOR EACH ROW  
BEGIN  
    IF NEW.mis_frequency = 21 THEN  
        INSERT INTO word(wor_string, wor_script) VALUES (NEW.mis_string, NEW.mis_frequency) ;  
    END IF;  
END$$  
DELIMITER ;
```

Vistas

crea

	Word	Frequency
►	de	9999518
	la	6277560
	que	4681839
	el	4569652
	en	4234281
	y	4180279
	a	3260939

codings

	Id	Come_from	Next_word	Frequency
►	2	sombra	imborrable	1
	3	imborrable	del	1
	4	del	josco	1
	5	josco	sobre	1
	6	sobre	la	1
	7	la	loma	1

texts

	Id	Text_name	Frequency
►	814820	Theodor Storm - El jinete del caballo blanco	5374
	485443	Herman Melville - Benito Cereno	4616
	838038	Voltaire - Cándido o el optimismo	4294
	533698	Jack London - El llamado de la selva	3845
	559129	Jean-Paul Sartre - La infancia de un jefe	3834
	13602	Albert Camus - El extranjero	3784
	822717	Thomas Mann - La muerte en Venecia	3708

scripts

	Word	Frequency
►	de	307276
	la	223394
	y	208262
	que	198327
	el	172276
	a	151158

mistakes

	Word	Frequency
--	------	-----------

phrases

	Id	Id_coding	Phrase	Frequency	Text_name
►	1	13	sombra imborrable del josco sobre la loma que d...	1	Abelardo Díaz Alfaro - El Josco
	2	16	la cabeza erguida	1	Abelardo Díaz Alfaro - El Josco
	3	30	las aspas filosas estoqueando el capote en san...	1	Abelardo Díaz Alfaro - El Josco
	4	33	la carrilluda en sombras	1	Abelardo Díaz Alfaro - El Josco
	5	38	el andar lento y rítmico	1	Abelardo Díaz Alfaro - El Josco
	6	48	la baba gelatinosa le caía de los belfos negros y...	1	Abelardo Díaz Alfaro - El Josco
	7	57	dejando en el verde enjoyado estela plateada d...	1	Abelardo Díaz Alfaro - El Josco
	8	69	era hosco por el color y por su carácter reconce...	1	Abelardo Díaz Alfaro - El Josco
	9	72	de peleador incansable	1	Abelardo Díaz Alfaro - El Josco
	10	87	cuando sobre el lomo negro del cerro farallón la...	1	Abelardo Díaz Alfaro - El Josco
	11	93	lo veía descender la loma majestuoso	1	Abelardo Díaz Alfaro - El Josco
	12	97	doblar la recia cerviz	1	Abelardo Díaz Alfaro - El Josco

Total_words

	Word	Frequency
►	de	10306794
	la	6500954
	que	4880166
	el	4741928
	y	4388541
	en	4376139
	a	3412097

Stored Procedures

La base de datos cuenta con siete procedimientos almacenados que detallaremos a continuación cada uno de ellos, seis de los siguientes se utilizan para insertar datos y el que mostramos en esta diapositiva se utiliza para encontrar el id de una palabra:

```
DELIMITER $$  
CREATE PROCEDURE id_word(IN w_string VARCHAR(50), OUT w_id INT)  
BEGIN  
    START TRANSACTION;  
    SELECT wor_id INTO w_id FROM word WHERE wor_string = w_string ;  
    COMMIT ;  
END$$  
DELIMITER ;
```

Stored Procedures

```
DELIMITER $$  
CREATE PROCEDURE insert_crea(IN w_string VARCHAR(50), IN w_frequency INT)  
BEGIN  
    START TRANSACTION;  
    INSERT INTO word(wor_string, wor_crea) VALUES (w_string, w_frequency) ON DUPLICATE KEY UPDATE wor_crea = w_frequency ;  
    COMMIT ;  
END$$  
DELIMITER ;  
  
DELIMITER $$  
CREATE PROCEDURE insert_word(IN w_string VARCHAR(50))  
BEGIN  
    START TRANSACTION;  
    INSERT INTO word(wor_string, wor_script) VALUES (w_string, 1) ON DUPLICATE KEY UPDATE wor_script = wor_script + 1 ;  
    COMMIT ;  
END$$  
DELIMITER ;  
  
DELIMITER $$  
CREATE PROCEDURE insert_mistake(IN m_string VARCHAR(50))  
BEGIN  
    START TRANSACTION;  
    INSERT INTO mistake(mis_string) VALUES (m_string) ON DUPLICATE KEY UPDATE mis_frequency = mis_frequency + 1 ;  
    COMMIT ;  
END$$  
DELIMITER ;
```

Stored Procedures

```
DELIMITER $$  
CREATE PROCEDURE insert_coding(INOUT come_from INT, IN next_word VARCHAR(50))  
BEGIN  
    DECLARE id INT ;  
    DECLARE id_word INT ;  
    START TRANSACTION;  
    SELECT wor_id INTO id_word FROM word WHERE wor_string = next_word ;  
    IF come_from IS NULL THEN  
        SELECT cod_id INTO id FROM coding WHERE cod_come_from IS NULL AND cod_wor_id = id_word ;  
    ELSE  
        SELECT cod_id INTO id FROM coding WHERE cod_come_from = come_from AND cod_wor_id = id_word ;  
    END IF ;  
    IF id IS NULL THEN  
        INSERT INTO coding(cod_come_from, cod_wor_id) VALUES (come_from, id_word) ;  
        SELECT MAX(cod_id) INTO come_from FROM coding ;  
    ELSE  
        UPDATE coding SET cod_frequency = cod_frequency + 1 WHERE cod_id = id ;  
        SET come_from = id ;  
    END IF ;  
    COMMIT ;  
END$$  
DELIMITER ;
```


Stored Procedures

```
DELIMITER $$  
  
CREATE PROCEDURE insert_text(IN t_name VARCHAR(500), OUT t_id INT)  
BEGIN  
    START TRANSACTION;  
    INSERT INTO text(tex_name) VALUES (t_name) ON DUPLICATE KEY UPDATE tex_frequency = tex_frequency + 1 ;  
    SELECT tex_id INTO t_id FROM text WHERE tex_name = t_name ;  
    COMMIT ;  
END$$  
DELIMITER ;  
  
DELIMITER $$  
  
CREATE PROCEDURE insert_phrase(IN p_phrase VARCHAR(2000), IN t_name VARCHAR(500), IN c_id INT)  
BEGIN  
    DECLARE t_id INT ;  
    DECLARE p_id INT ;  
    START TRANSACTION;  
    CALL insert_text(t_name, t_id) ;  
    SELECT phr_id INTO p_id FROM phrase WHERE phr_phrase = p_phrase AND phr_cod_id = c_id AND phr_tex_id = t_id ;  
    IF p_id IS NULL THEN  
        INSERT INTO phrase(phr_phrase, phr_cod_id, phr_tex_id) VALUES (p_phrase, c_id, t_id) ;  
    ELSE  
        UPDATE phrase SET phr_frequency = phr_frequency + 1 WHERE phr_id = p_id ;  
    END IF ;  
    COMMIT ;  
END$$  
DELIMITER ;
```

Usuarios

La base de datos cuenta con dos usuarios, el primero es el usuario por defecto “root” y el otro es el usuario “guest”. El script de creación para los usuarios invitados es el siguiente:

```
CREATE USER 'guest'@'localhost' IDENTIFIED BY 'guest';  
GRANT SELECT ON modeler.crea TO 'guest'@'localhost';  
GRANT SELECT ON modeler.scripts TO 'guest'@'localhost';  
GRANT SELECT ON modeler.total_words TO 'guest'@'localhost';  
GRANT SELECT ON modeler.mistakes TO 'guest'@'localhost';  
GRANT SELECT ON modeler.codings TO 'guest'@'localhost';  
GRANT SELECT ON modeler.texts TO 'guest'@'localhost';  
GRANT SELECT ON modeler.phrases TO 'guest'@'localhost';
```

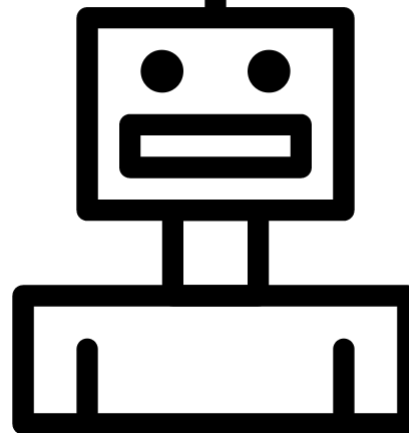
Values

```
CALL insert_crea('de', 9999518) ;  
CALL insert_crea('la', 6277560) ;  
CALL insert_crea('que', 4681839) ;  
CALL insert_crea('el', 4569652) ;  
CALL insert_crea('en', 4234281) ;  
CALL insert_crea('y', 4180279) ;  
CALL insert_crea('a', 3260939) ;  
CALL insert_crea('los', 2618657) ;  
CALL insert_crea('se', 2022514) ;  
CALL insert_crea('del', 1857225) ;  
CALL insert_crea('las', 1686741) ;  
CALL insert_crea('un', 1659827) ;  
CALL insert_crea('por', 1561904) ;  
CALL insert_crea('con', 1481607) ;  
CALL insert_crea('no', 1465503) ;  
CALL insert_crea('una', 1347603) ;  
CALL insert_crea('su', 1103617) ;  
CALL insert_crea('para', 1062152) ;
```

Por defecto la librería tiene una carpeta **SYSTEM** la cual contiene los scripts de creación de la base de datos, uno de esos scripts es **modeler_initialization.sql** el cual posee las sentencias para ingresar las primeras **121.000 del CREA**.

Un ejemplo de esas sentencias es el mostrado a la izquierda, el cual es el llamado al procedimiento almacenado **insert_crea** que recibe como parámetro la palabra y las veces que se repite en el **CREA**.

Funcionalidades



Resumen de las funcionalidades

La librería puede desempeñar las siguientes funcionalidades, ya sea por medio del llamado a sus métodos o por medio de su interfaz de consola:



- **Buscador de frases:** Dada una palabra o frase, busca todas las frases de la base de datos con esa frase o palabra, señalando en cada una el texto de su origen
- **Corrector de ortografía:** Dada una palabra o frase corrige su ortografía
- **Completar una frase:** A partir de una palabra o frase indica cual es la palabra mas probable a continuación
- **Entrenador corrector ortografico:** Funciona igual que el corrector ortografico, sin embargo, te indica las posibles correcciones y en caso de que no exista una correction guarda la palabra como un error (**exclusiva del usuario “root”**)

Buscador de frases

Creando una conexión con la base de datos es posible consultar la **vista phrases** y realizar una búsqueda sobre la misma, la sentencia a utilizar será **SELECT LIKE** para utilizar el comodín de búsqueda “%”. Un ejemplo de la búsqueda para todas las frases que contienen “**tu**” en la base de datos seria la siguiente:

```
SELECT Phrase, Text_name FROM phrases WHERE Phrase LIKE '%tu%'  
;
```

El resultado de esta búsqueda serán todas las tuplas **frase-texto** que contengan la palabra o frase indicada entre los porcentajes, de esta manera es como se implemento esta funcionalidad.

Corrector de ortografía

Para verificar cada uno de los ejemplos que se presentaran a continuación por favor considere como corrección ortográfica para cada tipo de error a la palabra:

Montenegro

Supongamos que al escribir la palabra de ejemplo cometemos un error, podemos clasificar cualquier posible error en alguno de estos tipos.

Error de omisión

Es la falta de uno o mas caracteres.

Ejemplo: Se omite la primer letra o

MIntenegro

Error de posición

Se intercambia uno o mas caracteres por uno diferente

Ejemplo: Se intercambio la r por la z

Monten**eg**zo

Error de adición

Se agrega un carácter de mas a la palabra

Ejemplo: Se escribe la palabra con dos t

Mont**t**enegro

Corrección ortográfica: virde

La palabra **virde** es realmente la palabra **verde** mal escrita, para corregirla ortográficamente veamos los posibles escenarios de acuerdo a los tipos de errores que presentamos anteriormente:

=

!irde
v!rde
vi!de
vir!e
vird!

-

?irde
v?rde
vi?de
vir?e
vird?

+

!virde
v!irde
vi!rde
vir!de
vird!e
virde!

El caso resaltado en **amarillo** es el que nos devolverá la corrección correcta de la palabra si intercambiamos '!' por 'e', para obtener la palabra **verde**

Símbolos: El símbolo ? representa la ausencia de un carácter en esa posición, la ! significa que existe un carácter cualquiera en esa posición, que es lo mismo que “_” al hacer una consulta en SQL.

Resultado de la corrección

Para encontrar la corrección de una palabra buscaremos en la base de datos todos los posibles escenarios que nombramos anteriormente con la sentencia **SELECT LIKE**. En la diapositiva anterior indicamos que el caso resaltado en amarillo es el que nos devolverá la corrección, es decir el caso **v!rde** que llevado a la sentencia SQL quedaría de la siguiente manera:

```
"SELECT * FROM " + table + " WHERE Word LIKE 'v_rde' ;"
```

Esta sentencia la hacemos con cada uno de los posibles escenarios y eso nos dará como resultado todas las posibles correcciones, ahora simplemente **solo elegimos la palabra que mas repeticiones** tenga en la base de datos. Si estamos corrigiendo mas de una palabra podemos utilizar la **vista codings** que nos indicara cual es el conjunto de palabras que puede continuar luego de una frase solo realizamos la intersección de estos dos conjuntos para afinar la corrección.

Completar una frase

Para completar una frase solo tenemos que realizar una consulta sobre la **vista codings**, pero antes es bueno explicar de que manera se almacenan los datos en esta vista.

- Para hacerlo supongamos la siguiente frase del autor **Abelardo Díaz Alfaro**: “sombra imborrable del josco sobre la loma que domina el valle del toa”

1. El programa divide la frase, es decir genera el arreglo: **[sombra, imborrable, del, josco, sobre, la, loma, que, domina, el, valle, del, toa]**
2. Luego analiza cada una de estas palabras y las agrega a la **tabla codings** teniendo en cuenta de que palabra viene y a que palabra va (agrega el id de la palabra, la vista solo convierte este id en la palabra en cuestión)
3. Si la combinación de palabras ya existe solo le suma uno a la frecuencia (esto lo hace el procedimiento almacenado)

	Id	Come_from	Next_word	Frequency
▶	2	sombra	imborrable	1
	3	imborrable	del	1
	4	del	josco	1
	5	josco	sobre	1
	6	sobre	la	1
	7	la	loma	1
	8	loma	que	1
	9	que	domina	1
	10	domina	el	1
	11	el	valle	1
	12	valle	del	1
	13	del	toa	1

Completar una frase

La **tabla codings** es una tabla que se relaciona consigo mismo, lo que mostramos anteriormente es la **vista codings** de esta tabla que realmente facilita la comprensión de la codificación, sin embargo, esta tabla solo esta compuesta por números y la columna **cod_come_from** es el id de la ultima fila de esta misma tabla que contiene la combinación de palabras anterior a la palabra de la columna **cod_wor_id**.

	cod_id	cod_come_from	cod_wor_id	cod_frequency
▶	1	NULL	1637	8
	2	1	32798	1
	3	2	10	1
	4	3	121004	1
	5	4	32	1
	6	5	2	1
	7	6	14927	1
	8	7	3	1
	9	8	7458	1
	10	9	4	1
	11	10	1904	1
	12	11	10	1
	13	12	51416	1

Nota: Abordar el problema de esta manera nos asegura poder crear relaciones entre las entidades que decidimos construir con anterioridad, ya que este tipo de modelos normalmente es construido bajo un paradigma **NOSQL**; el reto propuesto en este proyecto era proponer una alternativa funcional desde el paradigma **SQL**.

Entrenador corrector ortográfico

Esta es la misma funcionalidad que el corrector ortográfico, sin embargo, esta variación no nos mostrara una única corrección como resultado sino todas las posibles correcciones de una palabra y nos permitirá elegir cual es la correcta para nosotros, de esta manera una vez elegida la corrección correcta le sumara una frecuencia a esa palabra en la base de datos. Si no existe una corrección para esa palabra la agregara como un error a la base de datos.

Esta función es **exclusiva** del usuario **“root”** de la base de datos quien es el que posee todos los permisos sobre la base de datos. En el menú del usuario **“guest”** no aparecerá esta funcionalidad.

Entrenador corrector ortográfico

```
C:\Windows\py.exe

Modelador de textos

Entrenador corrector ortografico.

A continuación escriba por favor la palabra que desee revisar, para salir de esta funcionalidad
puede escribir el comando 'exit()' en lugar de una palabra.

- Escriba su palabra aqui: atlantico

Detalles de la corrección:

Las posibles correcciones para la palabra 'atlantico' son:

- 1. Atlántico
- 2. Atlantic
- 3. Ninguna de las anteriores

Elige un numero: 1

Se agrego una frecuencia a la palabra 'Atlántico' en la base de datos.

Presiona cualquier tecla para continuar...
```

Uso de la herramienta



Uso de las funcionalidades #1

Las funcionalidades se pueden usar de dos maneras, por medio del menú implementado en la librería o haciendo uso de los métodos, para llamar a los métodos primero hay que escribir **“import modeler”** en el script de Python en el que vayamos a utilizarlo, luego hacer uno de los siguientes llamados para la funcionalidad :

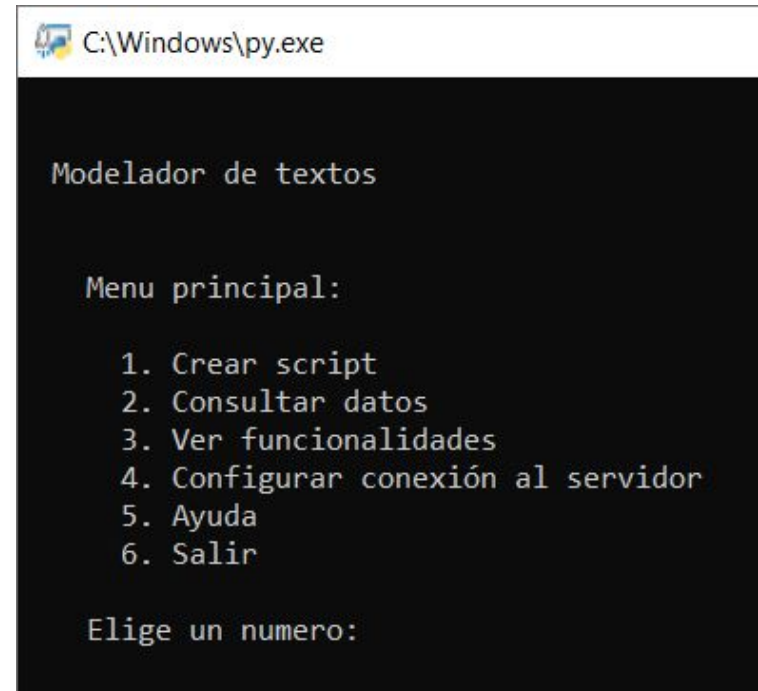
- **Buscador de frases:**
modeler.find_frase(phrase)
- **Corrector de ortografía:**
modeler.spell_checker (phrase, table)
- **Completar una frase:**
modeler.predictive_text(phrase)
- **Entrenador corrector ortografico:**
modeler.spell_checker_trainer(word, table)



Uso de las funcionalidades #2

La segunda alternativa es usar el menú integrado en la librería, para utilizar este menú existen dos maneras, la primera es ejecutar el script **modeler.py**, la segunda es ejecutar un **“import modeler”** y luego un **“modeler.Menu()”**. Al iniciar nos pedirá el usuario y contraseña de la base de datos, una vez ingresado y confirmada la conexión nos mostrara este menú:

Este menú principal es igual para ambos usuarios, su única diferencia es que en el sub menú **“Ver funcionalidades”** si nos conectamos con el usuario **“guest”** no nos aparecerá la funcionalidad **“Entrenador corrector ortográfico”**.



```
C:\Windows\py.exe

Modelador de textos

Menu principal:

1. Crear script
2. Consultar datos
3. Ver funcionalidades
4. Configurar conexión al servidor
5. Ayuda
6. Salir

Elige un numero:
```

¡MUCHAS GRACIAS!

