

## Tarea 1 - Traductor al Idioma Animalese

Fecha de entrega: 2 de octubre, 23h59

**Profesor:** Francisco J. Gutiérrez

**Auxiliares:** Konrad Ivelic – Sebastián Toro

Recuerde respetar la política de colaboración presentada al inicio del curso (y disponible en Material Docente).

Cada función **DEBE** seguir la receta de diseño: especifique contrato, propósito, ejemplos y tests, tanto para casos relevantes como de borde. Las funciones que retornen un archivo de audio NO deberán llevar tests

Se le sugiere trabajar con tiempo. NO DEJE LA TAREA PARA ÚLTIMO MINUTO.  
No se aceptarán tareas atrasadas ni entregas por cualquier otro medio distinto a U-Cursos.

### 1. Introducción

Animal Crossing<sup>1</sup> es una serie de videojuegos de simulación de vida de Nintendo. El último título dentro de la franquicia, *New Horizons*, es el juego más vendido en Nintendo Switch, lanzado “casualmente” al inicio de la pandemia mundial. Esto provocó que nuestros únicos amigos durante este distanciamiento social sean los pobladores de la isla de este juego. Lamentablemente, éstos se comunican mediante un lenguaje que no es familiar para nosotros, el Animalese, por lo que crearemos un traductor para así poder comunicarnos a la perfección.



Figura 1: Animal Crossing: New Horizons

<sup>1</sup>Podríamos haberlo llamado *Creature Intersection* para no ser acusados de violación de derechos de autor... y no tener que pagarle royalties a Tom Nook. Pero en esta línea declaramos que no haremos uso comercial de la marca y el contexto de esta tarea es con fines meramente educacionales.

El objetivo de esta tarea es **crear un traductor que logre recibir una frase por teclado y generar un audio con la frase traducida al lenguaje ficticio Animalese**. Esto se debe realizar únicamente con los contenidos trabajados durante el curso, en particular, recursión, manejo de condiciones, funciones y módulos.

## 2. Conocimientos Previos

En esta sección detallamos antecedentes relevantes para entender el contexto en el cual se enmarca esta tarea.

### 2.1. Dialectos de Animal Crossing

En el mundo de Animal Crossing existen 3 idiomas principales<sup>2</sup>. Animalese es el lenguaje hablado por defecto por los vecinos de la isla. Éste se habla pronunciando cada letra por separado, es decir, "Hello world" se transforma en "H e l l o w o r l d". Bebebese es un idioma hablado solamente por los hombres de nieve de la isla, el cual se habla usando ruidos "bip" repetitivamente y de forma aguda. El último idioma hablado es el Silencio, en el cual los vecinos simplemente no emitirán sonido, por lo que al hablar con ellos simplemente sonará la música de fondo.



Figura 2: Tom Nook dando el reporte de noticias diario a los vecinos de la isla.

### 2.2. Animalese

En esta tarea nos centraremos en crear un traductor que permita comunicarnos con los vecinos que hablan Animalese. La razón detrás de esto es que Bebebese es hablado solamente por los hombres de nieve y el Silencio, claramente, no necesita traductor.

<sup>2</sup>[https://nookipedia.com/wiki/List\\_of\\_languages](https://nookipedia.com/wiki/List_of_languages)

Cabe destacar que si bien el lenguaje Bebebese es hablado por muy pocos, se utiliza dentro del Animalese para tres casos:

- **Puntuación:** Para representar símbolos del estilo `?!*`
- **Mensajes fuera de la oración principal:** Cuando se realizan murmulos o cuando se habla consigo mismo, por ejemplo, al hablar entre paréntesis.
- **Reemplazar garabatos o palabras prohibidas:** Las palabras consideradas como groseras dentro de la isla se reemplazan por sonidos en Bebebese.

### 3. Creación del Traductor

Junto a este enunciado, se adjunta el módulo `animalese.py`, una carpeta `letters` y la carpeta `pydub`.

La carpeta `pydub` contiene los módulos necesarios para manejar el audio. Luego, **NO DEBEN** modificar nada contenido en esta carpeta. Tampoco deben importarlo directamente a su tarea, puesto que este módulo se utiliza dentro de `animalese.py`.

La carpeta `letters` posee 39 archivos de audio, de los cuales 26 representan las letras del alfabeto inglés (la ñ no está incluida), 5 representan dígrafos<sup>3</sup> ingleses, como por ejemplo `wh` o `th`, y 8 poseen sonidos que serán utilizados para representar el Bebebese.

El módulo `animalese.py` contiene las siguientes funciones que usted debe utilizar a lo largo de la tarea:

- `largo(palabra) :: str -> int` Retorna el largo de un String, incluyendo caracteres especiales como `' '`, `'*'`, `'.'` o `'.'`.
- `nuevoAudio() :: None -> Sonido` Retorna un segmento de audio vacío.
- `obtenerLetraEnPosicion(oracion, indice) :: str int -> str` Entrega el carácter del string `oracion` en la posición `indice`. El índice del primer carácter de la oración es `0`.
- `esLetra(letra) :: str -> bool` Indica si el string en `letra` está en la lista de sonidos de letras.
- `esDigrafo(digrafo) :: str -> bool` Indica si el string en `digrafo` está en la lista de sonidos de dígrafos.
- `esPuntuacion(letra) :: str -> bool` Indica si el string en `letra` es un signo de puntuación, como `'*'`, `'.'` o `'.'`.
- `obtenerSonido(letra) :: str -> Sonido` Entrega el sonido asociado al string `letra` en la carpeta `letters`. En caso de necesitar el sonido de en Bebebese puede ingresar el comando `obtenerSonido('bebebese')`.

<sup>3</sup><https://es.wikipedia.org/wiki/Digrafo>

- `reemplazarPalabra(oracion, palabra, nueva) :: str str str -> str` Reemplaza el string `palabra` por `nueva` en `oracion`. Un ejemplo de esta función es que al ejecutar el comando `reemplazarPalabra('Vamos a la playa', 'playa', 'casa')` retorna el string con la palabra reemplazada: `'Vamos a la casa'`
- `palabrasProhibidas() :: None -> int` Entrega la cantidad de palabras prohibidas.
- `palabraEnPosicion(i) :: int -> str` Entrega la palabra prohibida en la posición `i`.
- `posicionCaracter(oracion, letra) :: str str -> int` Entrega el índice con la posición del carácter `letra` en `oracion`.
- `existeEnOracion(letra, oracion) :: str str -> bool` Indica si el string en `letra` está en el string `oracion`.
- `reemplazarExtracto(oracion, reemplazo, inicio, fin) :: str str int int -> str` Inserta el string `reemplazo` en `oracion` entre las posiciones `inicio` y `fin`.
- `aMinusculas(oracion) :: str -> str` Pasa el string `oracion` a minúsculas.
- `cambiarVelocidadSonido(sound, velocidad) :: Sonido int -> Sonido` Cambia la velocidad del sonido afectando el tono del mismo, según el valor en `velocidad`. Si el valor ingresado es 1, el sonido no se modifica; si es mayor a 1, se hace más agudo; y si es menor a 1, se hace más grave.
- `agregarPalabraProhibida(palabra) :: str -> None` Agrega el string `palabra` a la lista de palabras prohibidas.
- `generarArchivoAudio(audio, nombre) :: Sonido str -> None` Exporta el segmento de audio ingresado a un archivo de extensión `.wav` cuyo nombre será el ingresado en la variable `'nombre'`.

Dichas funciones **NO** deben modificarse. Usted debe asumir que están correctas y que hacen lo anteriormente explicado. Puede revisar en el módulo las recetas de diseño de cada función, incluyendo ejemplos.

### 3.1. Funciones a programar

Usando las funciones provistas en el módulo `animalese.py`, se le pide que en un archivo llamado `traductor.py` implemente las siguientes funciones:

- a) Cree la función recursiva `reemplazarPalabrasProhibidas(oracion)` que reemplace en `'oracion'` todas las palabras prohibidas por asteriscos. Por ejemplo, si en la lista de las palabras prohibidas están `'malo'` y `'amigo'`, entonces `reemplazarPalabrasProhibidas('Mi amigo es muy malo')` debe retornar el string `'Mi ***** es muy *****'`. Cabe mencionar que la cantidad de asteriscos debe ser igual a la cantidad de letras de la palabra prohibida.

- b) Cree la función recursiva `reemplazarParentesis(oracion)` que reemplaza en el parámetro `'oracion'` todo el contenido que esté entre paréntesis por asteriscos. Por ejemplo, `reemplazarParentesis('Me gusta jugar (a veces) con mi amigo')` debe retornar el string `'Me gusta jugar ***** con mi amigo'`.
- c) Cree la función recursiva `construirAudio(oracion, audio)` que retorne en `'audio'` el elemento de sonido correspondiente al String `'oracion'`. Para esto debe encontrar el archivo de audio de cada carácter o dígrafo en `'oracion'` e ir almacenándolo en `'audio'`. Lea los caracteres en las posiciones `i` e `i + 1`, y verifique si forman o no dígrafo. Dependiendo del caso, debe recuperar el sonido correcto y generar el audio. Para concatenar los elementos de audio puede usar el operador `+`. Por ejemplo, si usted posee el elemento vacío de audio en la variable `audio_vacio` y el elemento de audio para la letra `'a'` en la variable `audio_a`, puede juntarlos haciendo `audio_vacio + audio_a`.
- d) Cree la función `procesarTexto(oracion)` que, utilizando las funciones anteriores, procese el texto del string `'oracion'`, eliminando palabras prohibidas y murmulos entre paréntesis, obtenga su elemento de sonido, le aumente la velocidad al doble y lo retorne.

### 3.2. Programa interactivo con el usuario

Finalmente se le pide que implemente un **programa interactivo** en un archivo de nombre `T1.py`, que le permita al usuario traducir texto ingresado por teclado a un archivo de audio `.wav` en Animalese.

Para esto, el programa debe dar un mensaje de bienvenida y luego permitir ingresar las palabras prohibidas de la aldea. El programa le debe preguntar recursivamente al usuario todas las palabras prohibidas hasta que ingrese la palabra clave `'fin'`. Luego, se le debe pedir al Usuario la frase que quiera traducir y el nombre del archivo. El programa deberá guardar en un archivo `.wav` la frase ingresada por el usuario traducida a Animalese.

A continuación se presenta un ejemplo de uso de este programa.

```
#####Bienvenido a Creature Intersection: New Adventures#####
"A continuación ingrese las palabras prohibidas en su Aldea, o 'fin' para terminar:"
Ingrese palabra prohibida: "
caca
"Palabra agregada!"
peo
"Palabra agregada!"
pichi
"Palabra agregada!"
fin
Ingrese mensaje a traducir: "
Introduccion a la programacion es el mejor ramo ever
Ingrese nombre de archivo: "
mejorramo
```

---

Esto debe provocar que se cree un archivo llamado `mejorramo.wav`, el cual contenga un sonido que representa la traducción del mensaje descrito en Animalese.

## 4. Entrega de la Tarea

La entrega de la tarea deben ser los archivos `traductor.py` y `T1.py`, indicando su nombre como comentario en la primera línea. Además, deberá documentar el proceso de diseño y desarrollo de su tarea en un informe de **no más de 3 páginas** en formato PDF. Se recibirán únicamente entregas por medio de U-Cursos en el espacio designado para ello. El plazo es hasta el **viernes 2 de octubre a las 23:59**.