

# Examen Final: *Ahorcado*

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería Mecánica Eléctrica  
Introducción a la Programación de Computadoras (0769) "N"

Ing. Iván René Morales  
2º Semestre 2014

## ¿Qué debe hacer?

Utilizando Python 2.7 implementar una variante del juego *Ahorcado*. Se debe jugar en función de un listado de palabras, las cuales pertenecen a categorías distintas. El programa mostrará al usuario los caracteres utilizados al intentar adivinar la palabra en cuestión, una representación gráfica simple del "ahorcado", los espacios disponibles para colocar las letras restantes y las oportunidades disponibles antes de perder. Al finalizar el juego el usuario podrá visualizar (ya sea que haya ganado, o perdido) su estado con un mensaje, mostrando la palabra que debía adivinarse.

## Restricciones y Limitaciones

El jugador debe elegir inicialmente la categoría en la que desea jugar, y de allí Python seleccionará al azar (utilizando *random*) una palabra perteneciente a dicha categoría. Las palabras y categorías deben obtenerse de un archivo de texto llamado **examenFinal.py**, que se encuentra en el mismo directorio que el *script*. La cantidad de categorías y de palabras depende únicamente del contenido del archivo de texto (Diccionario), por lo que el programa debe ser capaz de leer todos los datos que el usuario haya colocado en dicho fichero. El formato de éste debe ser exactamente igual al archivo de ejemplo que se encuentra en la sección **Documentos de ayuda**.

El programa debe ser capaz de discriminar caracteres inválidos ingresados por el usuario. Asimismo, debe hacer caso omiso a caracteres que hayan sido utilizados previamente por el usuario durante el juego actual. Asimismo, todo debe ser convertido a MAYÚSCULAS, con el fin de homogenizar la forma de escritura. Algo muy importante, es tomar en cuenta los espacios en blanco: por ejemplo, la palabra *COSTA RICA* no puede dejarse como se encuentra originalmente, ya que la casilla sin ocupar debe reemplazarse por un GUIÓN (—)

(u otro símbolo que se considere conveniente). Los espacios que el usuario tiene disponibles para colocar las letras deben simbolizarse con un GUIÓN BAJO (-)

La interfaz gráfica debe ser a través de sentencias **print** en consola.

Queda limitada la importación de módulos al uso de **random**.

Todo debe realizarse utilizando programación orientada a objetos. Si no se cumple con este requerimiento, el proyecto no será calificado.

## Documentos de ayuda

- Video en *YouTube* con un ejemplo del funcionamiento:  
<http://youtu.be/Lo1H5rfbBto>
- Ejemplo de Diccionario (examenFinal.txt):  
<http://goo.gl/s9wcVA>

## Metodología de entrega

Con el fin de agilizar el proceso de calificación y evitar hacinamiento de los alumnos, la entrega será a través de un formulario en la web, pero la calificación individual se realizará por bloques (divididos por números de carnet). Esta calendarización será publicada luego de la fecha de entrega.

El proyecto es trabaja de forma individual o en parejas, y cualquier indicio de copia (parcial o total) será motivo de anulación inmediata para todas las partes involucradas. El límite de la entrega del proyecto es el sábado 08/10/2014 a las 18:00 hrs. Este proceso será via web: <http://goo.gl/5STid3>. Puede enviar su examen solamente una vez, de lo contrario no será calificado. *Todo debe estar debidamente indentado y comentado.* Al momento de la calificación deberá lleva 2 COPIAS IMPRESAS de la última página de este documento. No es necesario que lleve computadora en este instante.

## Calificación

Carnet 1: \_\_\_\_\_ Carnet 2: \_\_\_\_\_

Descripción	Calificación
Uso adecuado de clases	/40
Interfaz del usuario	/20
Restricciones	/20
Manejo de archivo de texto	/20

Nota:

Calificó: