



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2233 - PROGRAMACIÓN AVANZADA

# Actividad 12

1º semestre 2018

7 de junio

## I/O: Serialización

### Introducción

¡Oh no! el Malvado Dr. Mavrakis ha conseguido aprender los **6 Lenguajes del Infinito** y está decidido en lograr el balance en el *IIC2233 Syllabus Universe*. Para esto, el malvado rufián tiene pensado serializar a la mitad de los usuarios del curso Programación Avanzada, con el objetivo de debilitar el universo de los versados estudiantes que lo componen y así después poder serializar a la mitad de la toda la comunidad de *Github*.



6 Lenguajes del Infinito <sup>1</sup>

Después de correr 14.000.605 simulaciones DES, te das cuenta que el único escenario posible en el que salvas tu semestre es ayudando al Dr. Mavrakis. Por esta razón, para esta actividad deberás hacer todo lo posible para que este vil villano lleve a cabo su plan.

### Instrucciones

Como habrás notado, el Dr. Mavrakis te entregó el *front-end* de su malévola aplicación, pero necesita que completes el *back-end* de ésta. Las funciones que están en el archivo `back_end.py` y que debes completar son las siguientes:

- **agregar\_estilo:** Lamentablemente el Dr. Mavrakis no tiene conocimientos de *CSS*, por lo que te pide que le des estilo a su interfaz. En el *back-end* encontrarás la variable `ESTILO` que contiene los estilos de los *widgets* de la interfaz. Éstos están en el formato: `((widget1, estilo1), (widget2, estilo2), ...)`. Tu función debe retornar un objeto `json` con el formato: `{widget1: estilo1, widget2: estilo2 ...}`

---

<sup>1</sup>C++, Ruby, C#, Javascript, Python, HTML respectivamente

- **cargar\_personas**: El archivo `personas.json` contiene la información de todos los alumnos del curso. Cada alumno contiene atributos que no le interesan al Dr. Mavrakis, por esta razón te entrega el archivo `caracteristicas.json` con los atributos que sí necesita. Tu función deberá cargar el archivo `personas.json` utilizando un `object_hook`, el cual debe recibir una **función auxiliar** que se encargue de cargar el archivo `caracteristicas.json`, filtrar los atributos necesarios y retornar un objeto `Persona`. Finalmente, tu función deberá retornar una lista de `Personas`.
- **generar\_personas(personas)**: Tu función deberá crear la carpeta `Personas` que contenga a las `personas` serializadas en formato JSON según el siguiente formato:

```
{
  "Nombre": nombre del alumno,
  "Apellido": primer apellido del alumno,
  "Numero de Alumno": numero de alumno
}
```

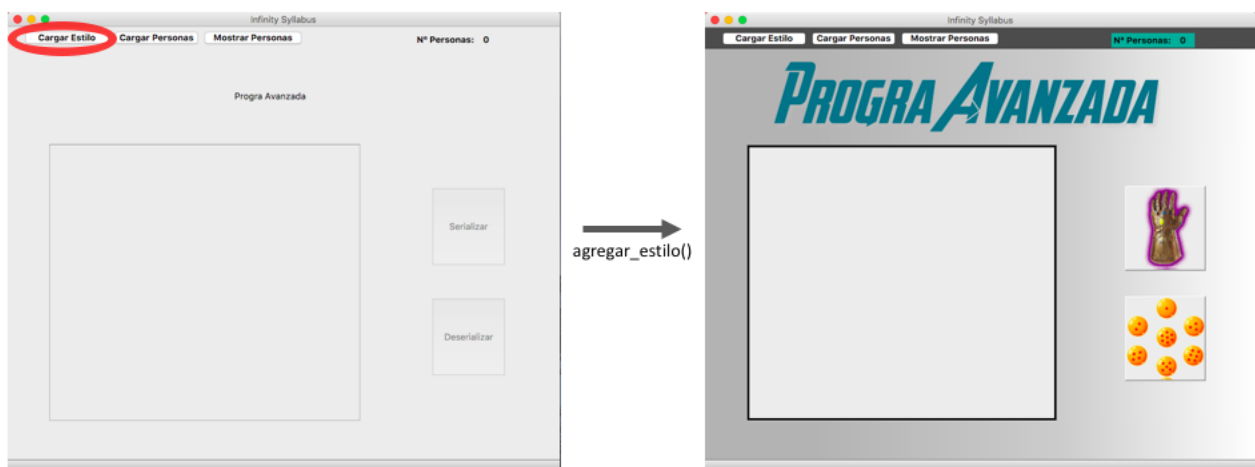
Para ello se deberá personalizar la serialización de cada persona mediante un `JSONEncoder`. El nombre de los archivos debe seguir el siguiente formato: `<codigo_genetico>.json`, donde `codigo_genetico` corresponde al código genético de cada persona.

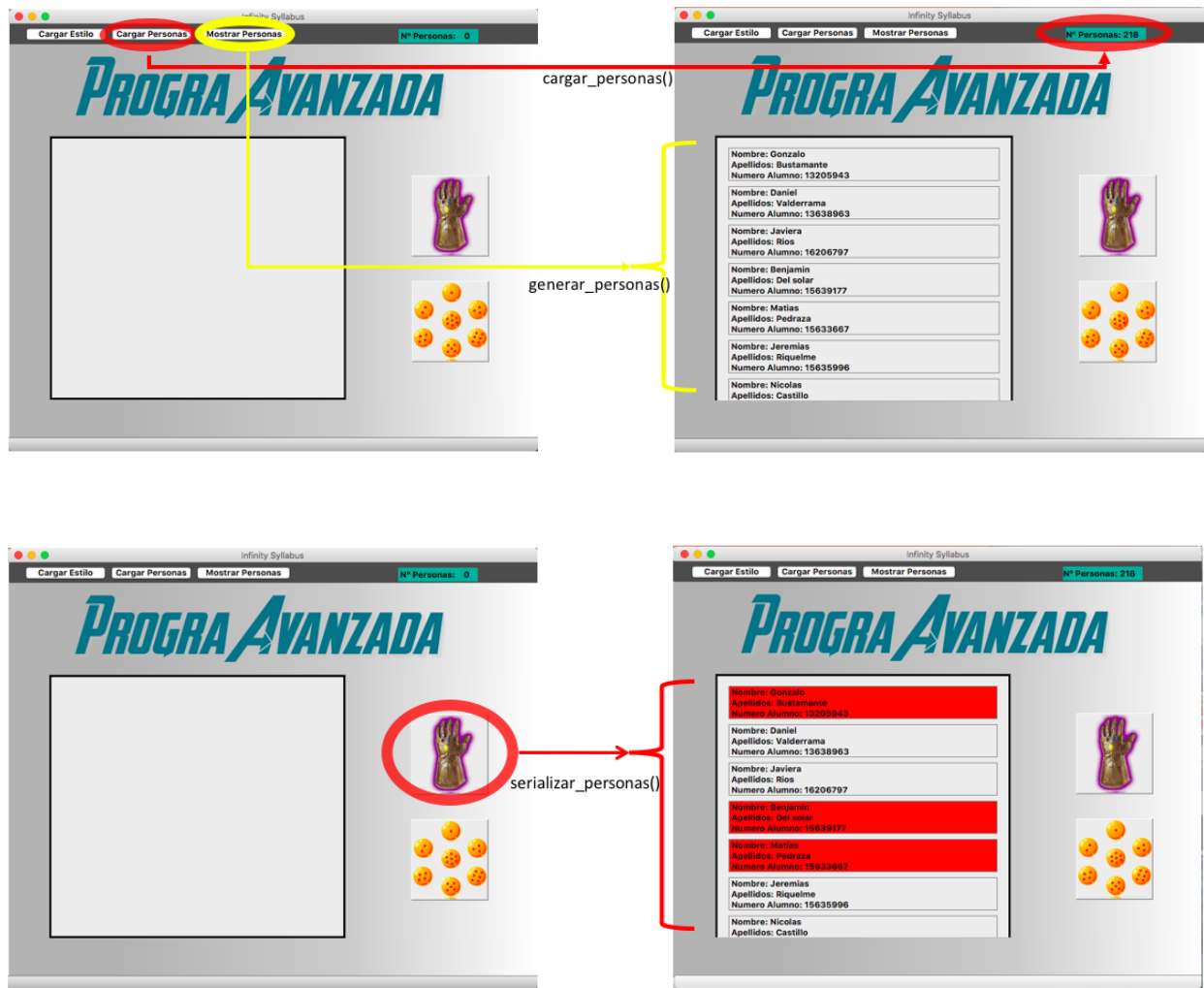
- **serializar\_personas(personas)**: Tu función debe seleccionar a la mitad de las personas al azar y serializarlas utilizando `pickle`. Para ello se deberán guardar los archivos generados en una carpeta llamada `Serializados` según el formato: `<numero_alumno>.rip`, donde `numero_alumno` corresponde al número de alumno de cada persona.

Como la serialización mediante `pickle` transforma a las `Personas` en bytes, éstas alcanzan a decir algo antes de ser serializados. Es por esto que debes modificar el `__getstate__` de la clase `Persona` para que cuando alguien sea serializado se le agregue el atributo `ultimas_palabras` y este sea igual a alguna frase emotiva (e.g: "Señor Nebil no me quiero ir..."). Deberás también modificar el `__setstate__` en caso de que la persona sea resucitada (deserializada), donde se borre este atributo.

## Anexo: Comportamiento de la GUI

Para verificar si las funciones anteriores están correctas, a continuación se muestra el comportamiento esperado en la interfaz según las acciones realizadas:





Una vez completada las funciones anteriores, el botón de las esferas del dragón te permite deserializar a las personas para volver a serializar y así ver quienes serían las nuevas víctimas.

## Notas

- Sólo puedes modificar el archivo `back_end.py`.
- La clase `Persona` ya está implementada en el `back_end`, sólo hay que modificar el `__getstate__` y `__setstate__` de ésta.
- La interfaz también funciona si no tiene agregado el estilo.
- Para crear carpetas puedes utilizar el método `mkdir` de la librería `os`.
- No subir a GitHub las carpetas y archivos generados (será penalizado con **\*\*0.2 pts\*\***). Se recomienda el uso de un `.gitignore`.

## Requerimientos

- (0,80 pts.) Función `agregar_estilo`: Retorna el estilo de la interfaz usando librería `json` con el formato solicitado.
- (0,60 pts.) Función auxiliar:
  - (0,40 pts.) Implementa función auxiliar para ser utilizada como parámetro del `object_hook` (en función `cargar_personas`).
  - (0,20 pts.) Carga el archivo `caracteristicas.json` utilizando librería `json`.
- (0,80 pts.) Función `cargar_personas`:
  - (0,60 pts.) Carga el archivo `personas.json` utilizando un `object_hook`.
  - (0,20 pts.) Retorna una lista de instancias de la clase `Persona`.
- (1,60 pts.) Función `generar_personas`:
  - (0,60 pts.) Se implementa un `JSONEncoder` propio y con el formato pedido.
  - (0,40 pts.) Se crea la carpeta `Personas`
  - (0,60 pts.) Dentro de ésta se guardan las personas serializadas utilizando la librería `json` y el `JSONEncoder` personalizado.
- (1,00 pts.) Función `serializar_personas`:
  - (0,40 pts.) Se crea la carpeta `Serializados`.
  - (0,60 pts.) Dentro de ésta se guardan las personas serializadas (escogidas al azar) utilizando la librería `pickle` en el formato pedido.
- (0,60 pts.) Correcta sobrescritura del método `__getstate__`.
- (0,60 pts.) Correcta sobrescritura del método `__setstate__`.

## Entrega

- **Lugar:** En su repositorio personal de GitHub en la **carpeta** `Actividades/AC12/`
- **Hora:** 16:40 hrs.
- Si está trabajando en pareja y ambos miembros suben la carpeta `Actividades/AC12/`, se corregirá una entrega al azar. Si ambos desean tener el código en sus repositorios, la carpeta que no se debe corregir tiene que llamarse `Actividades/AC12-pareja/`.