



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

Actividad 05

2º semestre 2017

7 de Septiembre de 2017

Excepciones y Testing

Introducción

El Departamento de Ciencia de la Computación se ha percatado de que muchos de sus cursos se quedan sin cupos y, por lo tanto, ha creado un formulario para que sus alumnos puedan indicar qué cursos necesitan tomar. Sin embargo, apareció un problema en la librería creada para analizar las respuestas del formulario. Esta librería no considera que algunos ponían puntos en su RUT, otros daban secciones que no existían o daban respuestas en palabras en vez de números. Todo esto provocó caos¹ en el departamento, por lo que se solicitó a los alumnos de Programación Avanzada que mejoren la librería y luego creen un nuevo código que la use pero permita esa clase de *inputs* sin generar errores.

Instrucciones

Esta actividad se divide en 3 partes: La primera es **identificar** los errores por input de las funciones de la librería y levantar excepciones con un mensaje que explique la excepción y porqué se levantó. La segunda es corregir un script que reciba diferentes *inputs* para que **capture** las excepciones generadas y arregle el input. Por último deberán crear un módulo de *testing* que verifique el correcto funcionamiento de la librería.

Parte 1: arreglar la librería

La librería posee los siguientes métodos:

- **check_rut(RUT)**: Chequea que el RUT sea válido y debe venir en el siguiente formato. *numeros_sin_punto_dígito_verificador*. Ejemplo “19644911-6”. Este método debe levantar una excepción cuando el número viene con puntos o en vez de guión hay un espacio. Ejemplo: “19.644.911-6” o “19.644.911 6”. Si el RUT es válido retorna **True**, en otro caso retorna **False**.
- **add_course(course, section)**: Añade a un diccionario llamado **courses** la demanda del curso indicado. El curso debe venir con la sigla del departamento y la sigla numérica sin espacios. Ejemplo: IIC2233 o IIC2115. Si existe un espacio entre la sigla del departamento y la numérica, se debe levantar una excepción. En caso de que el curso o la sección no exista en la base de datos, se debe levantar una excepción.

¹Para ver una descripción gráfica del caos, ver este link

- `register_people_info(student_name, gender, comment)`: Este método se llama después de chequear el RUT, en caso de que sea correcto se guarda la información del estudiante en una base de datos temporal.
- `save_data(path)`: Dado el `path`, genera un archivo con todos los usuarios registrados en la base de datos temporal y deja vacía esa base de datos. Si el `path` es de un archivo existente, se debe levantar una excepción.

Parte 2: usar la librería

Para la segunda parte, deben corregir el código `AC05.py`, que utiliza la librería que ya corrigieron, para analizar los *inputs* de un archivo de texto. Para esta parte, se entrega una descripción de los diversos *inputs* que recibirán y cual es la solución propuesta que deben implementar al momento de capturar las excepciones levantadas por la librería:

Descripción	Solución esperada	Ejemplo	Como queda
Los cursos ingresados pueden tener un espacio entre la sigla del departamento y la sigla numérica.	Eliminar ese espacio.	IIC 1103	IIC1103
Los RUT vienen con puntos o sin guión.	Eliminar los puntos o agregar el guión.	19.644.911-6, 196449116 o 19.644.9116	19644911-6
El número de la sección no existe para un curso.	Dejarla como sección 0.	888	0
Ingresa "section N" con N el número de la sección en vez de solo el número.	Eliminar la palabra para solo dejar el número.	section 5	5
En vez de poner 0 en sección, algunos escriben "todas".	Se reemplaza por 0.	todas	0

Parte 3: Testing

Para demostrar que la librería funciona, el departamento te ha solicitado que mandes un archivo de testing en Python que verifique el cumplimiento de lo esperado bajo ciertas circunstancias:

- Al verificar un RUT con un dígito verificador erróneo (es decir, un RUT incorrecto, pero escrito en el formato indicado), el método `check_rut` retorna `False`.
- Al ingresar con un formato equivocado el RUT de un usuario que se quiere registrar (con puntos y/o sin guión), verificar que se levanta una excepción, pero que al ingresarlo en el formato correcto, el método retorna `True`.
- Verificar que si se registra de manera correcta una persona y se guardan los cambios realizados en un archivo, los datos están bien ingresados en el archivo creado. Es decir, verificar que las primeras 4 líneas del archivo sean: (Considerar los nombres entre comillas como los *inputs* del usuario)

```
Student: "nombre_alumno"
Gender: "género"
Comment: "comentario"
#####
```

- Verificar que `register_people_info` guarde dentro de la base temporal la información de la persona. Para esto deben verificar que cada dato ingresado se encuentra en el último elemento de la base de datos temporal.

Notas

- Pueden suponer que todos los RUT serán mayores o iguales a 10 millones.
- Solo debe modificar las lineas donde deban usar `try`, `except` y con `raise`. **No modificar el `main`, `__init__` o `save.file`.**
- Toda excepción que no esté especificada no será considerada. Puede utilizar las excepciones de Python para esta actividad.

Requerimientos

- (2.40 pts) Manejo de Excepciones (identificación):
 - (0.80 pts) Se levantan las excepciones correspondientes cuando se ingresa RUT con punto y/o guión.
 - (0.80 pts) Se levantan las excepciones correspondientes cuando se ingresa texto como sección.
 - (0.40 pts) Se levantan las excepciones correspondientes cuando la sección no corresponde.
 - (0.40 pts) Se levantan las excepciones correspondientes cuando no se sigue el formato.
- (2.40 pts) Manejo de Excepciones (captura):
 - (0.80 pts) Se manejan las excepciones levantadas cuando se ingresa el RUT con puntos y/o guion.
 - (0.80 pts) Se manejan las excepciones levantadas cuando se ingresan texto.
 - (0.40 pts) Se manejan las excepciones levantadas cuando la sección no existe.
 - (0.40 pts) Se manejan las excepciones levantadas cuando no se sigue el formato.
- (1.20 pts) Testing:
 - (0.20 pts) Test 1: Verificar validez del RUT.
 - (0.20 pts) Test 2: Verificar que el RUT esté escrito en el formato incorrecto.
 - (0.20 pts) Test 3: Verificar escritura de datos en un archivo.
 - (0.20 pts) Test 4: Verificar base de datos local.
 - (0.40 pts) Uso de `tearDown` y `setUp`.

Entrega

- **Lugar:** En su repositorio de Github en la **carpeta** `Actividades/AC05/`
- **Hora:** 16:55