

## TAREA 2

**Orlando Chaparro 12-11499**

**María Magallanes 13-10787**

Para el momento de la entrega del presente informe, en el repositorio de GitHub que se envía como parte de la entrega, puede verse con facilidad la utilización de los commits, para modificar la rama dev con las actualizaciones del código fuente realizadas por el equipo. Para esta tarea se decidió utilizar una misma rama en la cuál ambos desarrollamos simultáneamente el código fuente, respetando el uso de TDD y asegurándonos de cada paso fue gestionado como nos fue indicado en las clases de la asignatura.

En un inicio se creó la clase Person que contiene los campos por los cuales seleccionaremos si es apto o no para recibir la pensión. En los commits realizados puede verse como estan siendo realizados los cambios a medida que transcurre la adaptacion de TDD a nuestro programa final. Cada caso de prueba esta comentado en los commits que van siendo realizados para que pueda darse cuenta fácilmente cual caso de prueba es forzado a salir mal, cual caso borde sale satisfactoriamente bien, y por lo tanto en base a esto refactorizar.

Analizando las horas en la que cada uno de los integrantes empezó y culminó sus labores de documentación, implementación y depuración de pequeños detalles en el código con fines de culminar la tarea, es posible promediar el tiempo de trabajo de ambos integrantes al rededor de 9 a 10 horas cada uno. El trabajo se realizó la mitad del tiempo a distancia y la segunda mitad juntos, por lo cuál no tuvimos problema en decidir trabajar en la misma rama desde el inicio, respetando los commits de cada uno y trabajando en función de ellos.

La estructura de la clase persona es implementada de la siguiente manera:

```
class Person:
    def __init__(self, name, genre, birth, quotation):
        self.name = name
        self.birth = birth
        self.genre = genre
        self.quotation = quotation

    def __str__(self):
        return "Persona de genero: " + self.genre + ", año: " + str(birthYear) + ",
y cotización: " + str(self.quotation)
```

Un primer caso de prueba es expuesto como sigue

```
class TestPension(unittest.TestCase):
    #Primer caso de prueba:
    #Hombre con 750 cotizaciones
    def testPensionMale750Quot(self):
        p=personGenerator("M",1996,750)
        self.assertTrue(p.AbleIVSSPension())
```

En este, se precisan los datos para que el caso sea definido como correcto, a pesar de que la edad no sea suficiente, y así posteriormente durante los commits realizados por cada uno se puede ver como se va modificando el código para que empiece a recibir los parámetros de edad necesarios, posteriormente estos se dividen mediante condicionales para poder aceptar la entrada de caballeros y damos al proceso de decision de la pension. Asimismo, se realizan casos de pruebas como el siguiente:

```
#Tercer caso de prueba:  
#Hombre de 60 y con 750 cotizaciones  
def testPensionMale750QuotAnd60Y(self):  
    p=personGenerator("M",1960,750)  
    self.assertTrue(p.AbleIVSSPension())
```

En el cual todas las variables empiezan a tener sentido según los requerimientos de la tarea. Posteriormente se refactoriza y así se continúa a lo largo del código.