Caracas, 30 de enero 2020

República Bolivariana de Venezuela

Universidad Nacional Experimental Simón Bolívar

CI3715 - Laboratorio de Ingeniería de Software I

Profesor: Franco Gabriel Nori

Tarea 2

Alvaro Ávila

Carlos González

**Tabla de Contenidos**

Contenido

[Introducción 3](#_Toc31232234)

[Planteamiento del Problema 4](#_Toc31232235)

[Tecnología y Fundamentos Teóricos 6](#_Toc31232236)

[Solución del Problema 7](#_Toc31232237)

[Conclusión 10](#_Toc31232238)

# Introducción

A continuación, se presenta un proyecto de ingeniería de software que utiliza la metodología de trabajo TDD (Test Directed Development o Desarrollo Dirigido por Pruebas) como herramienta para implementar un sistema de verificación de pensionados según las leyes venezolanas. Dicho sistema permite consultar si una persona, masculina o femenina, es viable para optar por una pensión según sus semanas cotizadas y otras variables pertinentes.

# Planteamiento del Problema

Desarrolle en Python 3.x y pruebe con PyUnit en su IDE, usando el enfoque TDD (dirigido por casos de prueba), una función que devuelva como resultado si una persona cumple con los requisitos legales para recibir una pensión de vejez del Instituto Venezolano de los Seguros Sociales (IVSS).

Sabemos que, a tal fin, en Venezuela, la principal referencia legal es el artículo 27 de la Ley del Seguro Social (Decreto N° 6266,31 de julio 2008), el cual establece:

*“La asegurada o el asegurado, después de haber cumplido 60 años de edad si es varón o 55 si es mujer, tiene derecho a una pensión de vejez siempre que tenga acreditadas un mínimo de 750 semanas cotizadas.”*

Además, la ley también establece en el artículo 28:

*“La asegurada o el asegurado que realice actividades en medios insalubres o capaces de producir una vejez prematura, tiene derecho a una pensión por vejez a una edad más temprana a la que se refiere el artículo anterior y en la forma en que lo determine el Reglamento.”*

Este reglamento establece en el Artículo 162:

*“La pensión de vejez podrá concederse antes del cumplimiento de las edades señaladas en el artículo 27 de la Ley del Seguro Social, cuando el asegurado realice actividades en medios insalubres o capaces de producir vejez prematura, circunstancias éstas que deberán ser comprobadas por el Instituto.*

*A los efectos establecidos en este artículo, mediante reglamentación especial dictada por el Instituto y aprobada por el Ministerio del Trabajo, la edad límite podrá ser rebajada hasta en un (1) año por cada cuatro (4) trabajados en las condiciones antes dichas. Tal reducción no podrá exceder de cinco (5) años.”*

*Reforma Parcial del Reglamento General de la Ley del Seguro Social*

*Decreto N° 8.992 del 24 de abril 2012*

Debe aprovechar las funciones de librería de Python para manejar fechas. La función debe dar la respuesta a la fecha de hoy, donde hoy es el día en que se ejecute. Distribuya la labor de programación de la función y de los casos de prueba con su pareja y utilicen Git para guardar una versión nueva de su suite cada vez que alguno de los miembros culmine un caso de prueba nuevo. Recuerde ejecutar los casos previos para cerciorarse de que no se haya roto nada. Puede organizar las ramas de su repositorio de la manera que considere más conveniente, pero incluya al menos una rama master y una rama por programador, de modo que puedan practicar operaciones sencillas de commit, merge, pull y push. Una vez culminada la suite creada con la técnica de análisis de fronteras, agreguen algunos casos con esquinas y por malicia.

# Tecnología y Fundamentos Teóricos

Para el desarrollo de este proyecto se utilizó como lenguaje de programación Python 3.8, Visual Studio Code y Sublime Text 3 como editores de texto y Git como gestor de versiones. Además, se utilizaron las siguientes librerías para distintas utilidades:

* Para representar las fechas dentro del programa se utilizó la librería de datetime*[[1]](#footnote-1)*
* Para la ejecución de pruebas unitarias se utilizó el módulo de PyUnit unittest*[[2]](#footnote-2)*

Así mismo, se implementaron las prácticas de *git-flow[[3]](#footnote-3)* para el control de versiones del proyecto, el cual divide el proyecto en distintas ramas

Para el desarrollo del sistema se utilizó la técnica de Desarrollo Dirigido por Pruebas (TDD), por lo cual cada miembro del equipo se encargó de un rol distinto dentro del proyecto: uno de *Developer* y otro de *Tester*. Uno de los integrantes creaba el programa que diera solución al problema planteado mientras que el otro establecía los casos de prueba que considerara importantes y que el programa debía pasar sin problemas.

Por último, se implementó el archivo de .gitignore para mantener por fuera de git carpetas y archivos basura (como por ejemplo, la carpeta \_\_pycache\_\_).

# Solución del Problema

Se desarrolló un programa que, dado cuatro parámetros (fecha de nacimiento, fecha de inicio de período laboral, sexo, condiciones de salubridad en el trabajo) de una persona, debe determinar si la persona se encuentra, según las leyes anteriormente citadas, en condiciones de recibir una pensión del seguro social.

Para la definición de los casos de prueba se implementó una clase de prueba llamada TestPension extendiendo de la clase unittest.TestCase*.* El objetivo a probar de esta clase es la función pension que es la que calcula si la persona cumple o no las condiciones necesarias para ser pensionado. La función pension recibe los datos listados previamente con los siguientes tipos de datos:

* Fecha de nacimiento: Objeto de tipo datede la librería datetime
* Fecha de inicio laboral: Objeto de tipo datede la librería datetime
* Sexo (Masculino o Femenino): *Booleano* que evalúa a Truepara femenino, y False para masculino
* Indicar si trabajó en condiciones insalubres: *Booleano* que evalúa a Truepara condiciones insalubres, y False para condiciones saludables

La función pension busca calcular la edad actual de la persona y las semanas que ha trabajado, ambas respecto a la fecha actual. Dependiendo del sexo, la edad requerida según las leyes puede ser de 55 años si es mujer o 60 años si es hombre. A dicha edad se le resta la cantidad de años correspondiente en caso de que la persona haya trabajado en condiciones insalubres.

Finalmente, se calcula si trabajó el mínimo de 750 semanas estipuladas por la ley y se retornan ambas condiciones, es decir, la función devuelve un booleano que indica si cumple con el mínimo de tiempo necesario y tiene la edad estipulada por la ley.

El tiempo de desarrollo para cada integrante fue de aproximadamente 4 horas. En el siguiente enlace puede acceder al repositorio del curso y descargarlo para pruebas: <https://github.com/almiavicas/CI3715-Tarea2>. A continuación, se listan los casos de prueba con su respectivo valor esperado:

* Caso 1: Demostrar si un trabajador masculino con las semanas requeridas y la edad requerida para ser pensionado es determinado como pensionado.
  + Entrada:
    - Fecha de nacimiento: 61 años de edad recién cumplidos
    - Inicio de vida laboral: 15 años trabajando
    - Sexo: Masculino
    - Condiciones de insalubridad: no
  + Valor esperado: True
* Caso 2: Demostrar si un trabajador masculino sin las semanas requeridas y sin la edad requerida para ser pensionado es determinado como pensionado.
  + Entrada:
    - Fecha de nacimiento: 59 años de edad recién cumplidos
    - Inicio de vida laboral: 14 años trabajando
    - Sexo: Masculino
    - Condiciones de insalubridad: no
  + Valor esperado: False
* Caso 3: Demostrar si una trabajadora femenina con las semanas requeridas y la edad requerida para ser pensionado es determinada como pensionada.
  + Entrada:
    - Fecha de nacimiento: 56 años de edad recién cumplidos
    - Inicio de vida laboral: 15 años trabajando
    - Sexo: Femenino
    - Condiciones de insalubridad: no
  + Valor esperado: True
* Caso 4: Demostrar si una trabajadora femenina sin las semanas requeridas y sin la edad requerida para ser pensionada es determinada como pensionada.
  + Entrada:
    - Fecha de nacimiento: 54 años de edad recién cumplidos
    - Inicio de vida laboral: 14 años trabajando
    - Sexo: Femenino
    - Condiciones de insalubridad: no
  + Valor esperado: False
* Caso 5: Demostrar si un trabajador masculino, trabajando en condiciones insalubres, con las semanas requeridas y la edad requerida para ser pensionado es determinado como pensionado.
  + Entrada:
    - Fecha de nacimiento: 56 años de edad recién cumplidos
    - Inicio de vida laboral: 20 años trabajando
    - Sexo: Masculino
    - Condiciones de insalubridad: Si
  + Valor esperado: True
* Caso 6: Demostrar si un trabajador masculino, trabajando en condiciones insalubres, sin las semanas requeridas y sin la edad requerida para ser pensionado es determinado como pensionado.
  + Entrada:
    - Fecha de nacimiento: 54 años de edad recién cumplidos
    - Inicio de vida laboral: 14 años trabajando
    - Sexo: Masculino
    - Condiciones de insalubridad: Si
  + Valor esperado: False
* Caso 7: Demostrar si una trabajadora femenina, trabajando en condiciones insalubres, con las semanas requeridas y la edad requerida para ser pensionada es determinada como pensionada.
  + Entrada:
    - Fecha de nacimiento: 51 años de edad recién cumplidos
    - Inicio de vida laboral: 20 años trabajando
    - Sexo: Femenino
    - Condiciones de insalubridad: Si
  + Valor esperado: True
* Caso 8: Demostrar si una trabajadora femenina, trabajando en condiciones insalubres, sin las semanas requeridas y sin la edad requerida para ser pensionada es determinada como pensionada.
  + Entrada:
    - Fecha de nacimiento: 49 años de edad
    - Inicio de vida laboral: 14 años trabajando
    - Sexo: Femenino
    - Condiciones de insalubridad: Si
  + Valor esperado: False

# Conclusión

El desarrollo dirigido por casos de prueba (TDD) puede asegurar que un sistema mantiene su consistencia y confiabilidad en todo momento. Como las pruebas se hacen desde el inicio del desarrollo, la probabilidad de tener errores sin ser descubiertos se reduce al mínimo posible. Esto permite que el desarrollo siempre vaya hacia delante y haya pocos retrasos por problemas de acoplamiento, coherencia o diseño. Con esto en cuenta, esta metodología de trabajo nos ha brindado una eficiencia de tiempo para cada parte del equipo que nos permite avanzar con velocidad para alcanzar los objetivos planteados.

El sistema desarrollado, a pesar de ser pequeño, fue un excelente ejercicio de práctica para las herramientas utilizadas que serán implementadas a través de todo el curso y a futuro en el campo laboral.

1. <https://docs.python.org/3/library/datetime.html> [↑](#footnote-ref-1)
2. <https://wiki.python.org/moin/PyUnit> [↑](#footnote-ref-2)
3. <https://datasift.github.io/gitflow/IntroducingGitFlow.html> [↑](#footnote-ref-3)