
ciuy
Versión 1.0.0

08 de junio de 2020

Contenido:

| | |
|---|-----------|
| 1. Instalación | 3 |
| 2. Uso | 5 |
| 3. Testing | 7 |
| 4. Línea de comandos | 9 |
| 4.1. Funciones y ejemplos | 9 |
| 4.2. Acerca de los números de identidad uruguayos | 10 |
| Índice | 13 |

Paquete para validar números de documentos de identidad Uruguayos.

Las funciones en este paquete funcionan con strings e ignoran cualquier caracter que no sea un número. También se pueden usar números. Entonces por ejemplo, las siguientes expresiones representan el mismo número de documento:

```
'1.234.567-2'  
'12345672'  
12345672
```


CAPÍTULO 1

Instalación

ciuy puede ser instalado con `pip`:


```
>>> import ciuy
>>> ciuy.validate_ci("1.234.567-2")
True
>>> ciuy.validation_digit("1.234.567")
'2'
>>> ciuy.random()
'82405816'
```

Ver *Funciones y ejemplos* para información más detallada acerca de cada función.

Este paquete incluye algunos doctests, así como también tests unitarios que pueden ser ejecutados con `pytest`.

Luego de clonar el repositorio, puedes ejecutar los doctests con:

```
$ python3 -m doctest ./ciuy/__init__.py -v
(several lines omitted)
11 passed and 0 failed.
Test passed.
```

Usa `pytest` para ejecutar los tests unitarios:

```
$ py.test
===== test session starts =====
platform linux -- Python 3.8.1, pytest-5.3.5, py-1.8.1, pluggy-0.13.1
rootdir: /home/ciuy
collected 98 items

tests\test_clean.py ..... [ 10%]
tests\test_command_line.py ..... [ 34%]
tests\test_random_ci.py .. [ 36%]
tests\test_validate_ci.py ..... [ 76%]
tests\test_validation_digit.py ..... [100%]

===== 98 passed in 1.09s =====
```

También es posible usar `pytest` para correr todos los tests, incluyendo los doctests, con:

```
$ py.test --doctest-modules
```

Línea de comandos

Luego de la instalación, los siguientes comandos estarán disponibles:

```
$ validate_ci 1.234.567-2
True
$ validation_digit 1.234.567
2
$ random_ci
82405816
```

4.1 Funciones y ejemplos

`ciuy.validate_ci` (*ci*)

Valida el número de documento pasado como parámetro.

Parámetros *ci* (*str*) – Número de documento a ser validado. Los caracteres que no sean números son ignorados. También se puede pasar un número como parámetro.

Devuelve `True` si el número de documento es válido, `False` en caso contrario.

Tipo del valor devuelto `bool`

Muestra `ValueError` – si *ci*, sin incluir el dígito verificador, es menor que 100.000 o mayor que 9.999.999.

```
>>> ciuy.validate_ci(1.234.567-2)
True
```

`ciuy.validation_digit` (*ci*)

Retorna el dígito verificador para un número de documento dado.

Parámetros *ci* (*str*) – Número de documento para el cual se desea hallar el dígito verificador. Los caracteres que no sean números son ignorados. También se puede pasar un número como parámetro.

Devuelve El dígito verificador.

Tipo del valor devuelto bool

Muestra ValueError – si `cies` menor que 100.000 o mayor que 9.999.999.

```
>>> ciuy.validation_digit(1.234.567)
'2'
```

`ciuy.random()`

Devuelve un número de documento aleatorio, incluido el dígito de validación. El número de documento está en el rango (100.000, 9.999.999).

Devuelve Un número de documento válido aleatorio.

Tipo del valor devuelto str

```
>>> ciuy.random()
'82405816'
```

4.2 Acerca de los números de identidad uruguayos

En Uruguay, a cada persona se le asigna un documento de identidad, el cual contiene toda la información necesaria para identificar a la persona inequívocamente.

Este documento es obligatorio para todos los habitantes de Uruguay, sean ciudadanos naturales y legales, o extranjeros residentes en el país, incluso para los niños a partir de los 45 días de nacidos.



Figura 1: Ejemplo de documento de identidad uruguayo.

A cada persona se le asigna un número de identidad (el cual es determinado al momento de nacimiento para ciudadanos naturales), el cual contiene 7 dígitos y un dígito verificador. Los números de identidad antiguos contienen solo 6 dígitos, y un dígito verificador.

4.2.1 Calculando el dígito verificador

Para calcular el dígito verificador para un número de identidad de 7 dígitos, el número debe ser multiplicado, dígito por dígito, por 8123476. Entonces, el resultado es añadido. Es decir, si el número es interpretado como un vector de 7 dimensiones, se debe calcular el producto escalar de dicho vector por $\langle 8, 1, 2, 3, 4, 7, 6 \rangle$.

Entonces, si el número resultante es n , el dígito verificador será n módulo 10.

Validando números de 6 dígitos

Los números de 6 dígitos deben ser multiplicados por $\langle 1, 2, 3, 4, 7, 6 \rangle$. Dado n , el número resultante, entonces el dígito verificador será n módulo 10.

Un ejemplo

Validando 1.234.567:

$$\begin{aligned} &\langle 1, 2, 3, 4, 5, 6, 7 \rangle \cdot \langle 8, 1, 2, 3, 4, 7, 6 \rangle = \\ &1 * 8 + 2 * 1 + 3 * 2 + 4 * 3 + 5 * 4 + 6 * 7 + 7 * 6 = 132 \end{aligned}$$

$$132 \bmod 10 = 2$$

Entonces el dígito verificador es 2. El número de identidad normalmente sería escrito como 1.234.567-8.

C

`ciuy.random()` (*función incorporada*), 10

`ciuy.validate_ci()` (*función incorporada*), 9

`ciuy.validation_digit()` (*función incorporada*), 9