

---

# **impAr Documentation**

*Publicación 0.1*

**Juan BC - Nadia AL**

08 de January de 2014



<b>1. Installation Guide</b>	<b>3</b>
1.1. Ubuntu/Debian/Mint . . . . .	3
1.2. Windows u otro Xnix . . . . .	3
1.3. Desde el repositorio . . . . .	3
<b>2. Tutorial</b>	<b>5</b>
2.1. <code>datecontext</code> Module . . . . .	5
2.2. Impuestos . . . . .	7
<b>3. Indices and tables</b>	<b>9</b>
<b>Índice de Módulos Python</b>	<b>11</b>



Calcule impuestos argentinos según legislaciones de diferentes fechas

Contents:



---

## Installation Guide

---

### 1.1 Ubuntu/Debian/Mint

Ejecute en la terminal

```
$ sudo apt-get install python-setuptools python-pip
$ sudo pip install impar
```

### 1.2 Windows u otro Xnix

Primero instale las siguientes dependencias:

- Python 2.7 <http://www.python.org> (y agreguelo a su PATH)
- Setup tools <http://pypi.python.org/pypi/setuptools> (y agreguelo a su PATH)
- Mercurial (solo si quiere instalar desde el repositorio) <http://mercurial.selenic.com/>

Finalmente en la consola ejecute

```
> easy_install pip
> pip install impar
```

### 1.3 Desde el repositorio

Primero instale las dependencias y luego ejecute

```
$ hg clone http://bitbucket.org/leliel12/impar impar
$ cd impar
$ python setup.py install
```





### Implementaciones de impuestos argentinos

`impar` utiliza el concepto de contextos de fechas (`DateContext`) para separar implementaciones diferentes de los impuestos. Esto se debe a que los impuestos pueden calcularse de manera diferente dependiendo la fecha que se realizan.

Por ejemplo si quiere calcularse el **impuesto a las ganancias** al 29 de diciembre del 2013:

```
>>> import datetime
>>> import impar
>>> ctx = impar.DateContext(datetime.date(2013, 12, 29))
>>> ganancias = ctx.get("ganancias")
>>> ganancias.ganancias(20000, ganancias.CAT_EMPRESA)
700.0
```

Si por otro lado se quiere calcular la ganancia a fecha actual:

```
>>> import impar
>>> ctx = impar.DateContext()
>>> ganancias = ctx.get("ganancias")
>>> ganancias.ganancias(20000, ganancias.CAT_EMPRESA)
700.0
```

No hubo cambios en el impuesto por eso los valores y parametros son iguales.

#### Notas:

- El contexto se determina como la implementacion inmediatamente anterior o a la fecha del impuesto dado (puede verificar la fecha de implementacion del calculo con `ctx.date_of("ganancias")`).
- No todos los impuestos estan disponibles para todas las fechas.

## 2.1 datecontext Module

Manejador de contexto de fechas para los impuestos

```
class impar.datecontext.DateContext (date=None)
```

Define un contexto de calculo para impuestos. En la practica asume todas las legislaciones a la fecha dada en el contexto para cada impuesto

Si usted quiere calcular el impuestoA a con las legislaciones vigentes al dia 31 de octubre del 2012 deberia ejecutar el codigo de la siguiente manera

```
>>> import datetime, impar
>>> ctx = impar.DateContext(datetime(2012, 10, 31))
>>> impuestoA = ctx.get("impuestoA")
>>> impuestoA.calcular(...)
```

**Parámetros** `date` (*datetime.date*) – Fecha del contexto (hasta que fecha máxima se toman en cuenta las legislaciones para los impuestos). Si el valor es `None` se toma la fecha actual.

### **date**

Fecha del contexto

### **date\_of** (*impuesto*)

Retorna la fecha exacta de la implementación del impuesto utilizada por este contexto

**Parámetros** `impuesto` (*str*) – nombre del impuesto

### **get** (*impuesto*)

Retorna una implementación del impuesto dada la fecha del contexto

**Parámetros** `impuesto` (*str*) – nombre del impuesto

### **valid\_taxes**

Impuestos validos para este contexto

### **exception** `impar.datecontext.NotSuitableTax`

Si no existe un impuesto de fecha menor o igual a del contexto

### **exception** `impar.datecontext.TaxNotExists`

El impuesto no existe

### `impar.datecontext.parse_date` (*date*)

Convierte varios tipos de objetos a instancias de `datetime.date`

Todas estas funciones son equivalentes:

```
>>> # asumiendo que hoy es 25 de diciembre del 2013
>>> parse_date()
>>> parse_date(datetime.datetime.now())
>>> parse_date(datetime.datetime.now().date())
>>> parse_date(datetime.datetime(2013,12,25))
>>> parse_date(datetime.date(2013,12,25))
>>> parse_date("20131225")
>>> parse_date("2013-12-25")
>>> parse_date("2013_12_25")
>>> parse_date("2013_12-25")
>>> parse_date("2013-12_25")
>>> parse_date("2013/12/25")
>>> parse_date("2013/12-25")
>>> parse_date("2013-12/25")
>>> parse_date("2013/12_25")
>>> parse_date("2013_12/25")
>>> parse_date((2013, 12, 25))
```

### `impar.datecontext.same_implementation` (*date0, date1*)

Retorna un set con todos los impuestos que poseen la misma implementación en ambas fechas

### `impar.datecontext.valid_taxes` (*date=None*)

Retorna una tupla de todos los impuestos validos para una fecha dada, o todos si la fecha es `None`

## 2.2 Impuestos

### 2.2.1 Impuesto: 'ganancias'

#### Ganancias al 2013-12-29

Funciones de calculo de impuesto a las ganacias

`impar._impuestos.ganancias.ganancias_20131229.ALICUOTAS = [[0, 60000, <function <lambda> at 0x3ec09b0>]]`  
 Alicuota para cat 1, 2 y 4 (min, max, calculador)

`impar._impuestos.ganancias.ganancias_20131229.ALICUOTA_EMPRESA(x)`  
 Alicuota exclusiva para categoria empresas

`impar._impuestos.ganancias.ganancias_20131229.CAT_ALQUILERES = 1`  
 Categoria 1 (alquileres) de impuesto a las ganancias

`impar._impuestos.ganancias.ganancias_20131229.CAT_EMPLEADO = 4`  
 Categoria 4 (empleado) de impuesto a las ganancias

`impar._impuestos.ganancias.ganancias_20131229.CAT_EMPRESA = 3`  
 Categoria 3 (empresa) de impuesto a las ganancias

`impar._impuestos.ganancias.ganancias_20131229.CAT_TITULOS = 2`  
 Categoria 2 (títulos) de impuesto a las ganancias

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_FIJA_CONYUGUE = 8400.0`  
 Deducion por conyugue

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_FIJA_FAMILIAR = 7200.0`  
 Deducion por familiar

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_FIJA_HIJO = 7200.0`  
 Deducion por hijo

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_TOPE_SEGUROS_DE_RETIRO = 969.25`  
 Deducion máxima por seguro de retiro

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_TOPE_SEGUROS_DE_VIDA = 969.25`  
 Deducion máxima por seguro de vida

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_TOPE_SEPELIO = 969.25`  
 Deducion máxima por sepelio

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_VARIABLE_HONORARIO_MEDICO = 0.05`  
 Coeficiente de calculo de deduccion máxima por honorario médico

`impar._impuestos.ganancias.ganancias_20131229.DEDUCCION_VARIABLE_SEGURO_MEDICO = 0.05`  
 Coeficiente de calculo de deduccion máxima por seguro médico

`impar._impuestos.ganancias.ganancias_20131229.GANANCIA_NO_IMPONIBLE = 15960.0`  
 Valor de la ganancia no imponible

`impar._impuestos.ganancias.ganancias_20131229.GANANCIA_NO_IMPONIBLE_CATEGORIA_4 = 76608.0`  
 Valor de la ganancia no imponible para empleado

`impar._impuestos.ganancias.ganancias_20131229.MAX_QUEBRANTOS = 5`  
 Máximo número de quebrantos soportados en la ley

`impar._impuestos.ganancias.ganancias_20131229.TIEMPO_MIN_PERMANENCIA_MESES = 6`  
 Tiempo minimo de permanencia en el pais para aplicar el impuesto

```
impar._impuestos.ganancias.ganancias_20131229.ganancias (monto, cat, deduccio-  
nes_fijas=(), seguros_   
ros_medico=0, ho-   
norarios_medico=0,   
sepelios=0, segu-   
ros_de_vida=0, segu-   
ros_de_retiro=0, intere-   
ses_hipotecarios_vivienda_propia=0,   
tiem-   
po_permanencia_pais_meses=0)
```

Calcula el impuesto a las ganancias para una categoria dada

```
impar._impuestos.ganancias.ganancias_20131229.ganancias_empresa_con_quebrantos (quebrantos,   
*args,   
**kwargs)
```

Calcula el impuesto a las ganancias para empresas con impuestos con quebrantos

---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*



i

`impar.__init__`, 5

`impar._impuestos.ganancias.ganancias_20131229`,  
7

`impar.datecontext`, 5