

# Ola Mundo

## @api.onchange

---

É similar a @api.depends, pero só e para a clase en curso e os cambios son a nivel de caché (a nivel de [GUI](#)), non se almacenan na Base de Datos.

Poden utilizarse para mensaxes de advertencia, xa que os cambios realizados no método non teñen efecto na BD

Xa que os cambios non se van almacenar na BD definimos en informacion.py, por exemplo un campo literal con store=False

```
literal = fields.Char(store=False)
```

Na vista informacion.xml, definimos o literal con **readonly** para que o usuario non poda modificar a mensaxe.

```
<field name="literal" readonly="1"/>
```

En informacion.py, definimos o método que será executado cando cambiemos na GUI o valor do campo alto\_en\_cms

```
@api.onchange('alto_en_cms')
def _avisoAlto(self):
    for rexistro in self:
        if rexistro.alto_en_cms > 7:
            rexistro.literal = 'O alto ten un valor posiblemente excesivo %s é maior' % rexistro.alto_en_cms
        else:
            rexistro.literal = ''
```

Vemos a mensaxe de advertencia, cando cambiamos o valor do campo 'alto\_en\_cms'

	Datos Persoais	Datos Numéricos	Moedas	Datas
Literal Alto en Cms.	O alto ten un valor posiblemente excesivo 12 é maior que 7			
	<input type="text" value="12"/>			

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)