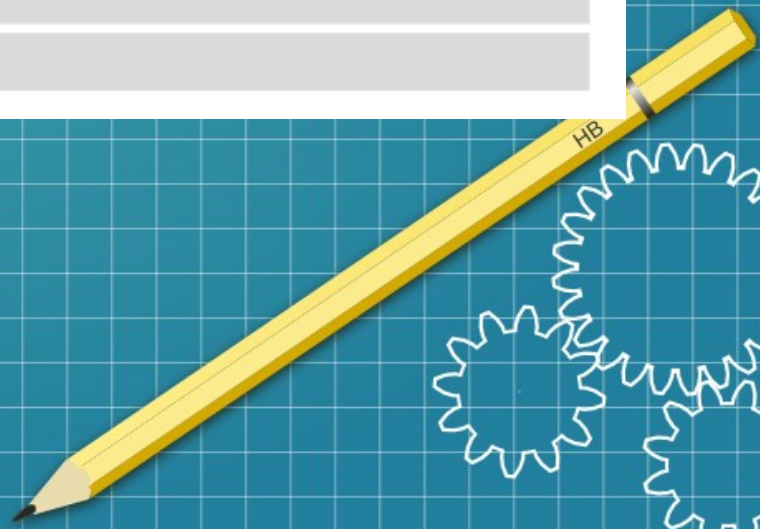
Minipresentación Modelos en Odoo

Algunas capturas del libro recomendado

Clase Model

```
class Book(models.Model):
    _name = "library.book"
    _description = "Book"
    _order = "name, date_published desc"
```



Campos Modelo

```
class Book(models.Model):
    _name = "library.book"
    _description = "Book"

    # String fields:
    name = fields.Char("Title")
    isbn = fields.Char("ISBN")
    book_type = fields.Selection(
        [("paper","Paperback"),
         ("hard","Hardcover"),
         ("electronic","Electronic"),
         ("other", "Other")],
        "Type")

    notes = fields.Text("Internal Notes")
    descr = fields.Html("Description")
```

Campos Modelo

```
    # Numeric fields:
    copies = fields.Integer(default=1)
    avg_rating = fields.Float("Average Rating", (3, 2))
    price = fields.Monetary("Price", "currency_id")
    # price helper
    currency_id = fields.Many2one("res.currency")

    # Date and time fields:
    date_published = fields.Date()
    last_borrow_date = fields.Datetime(
        "Last Borrowed On",
         default=lambda self: fields.Datetime.now())
```

Campos Modelo

```
    # Other fields:
    active = fields.Boolean("Active?")
    image = fields.Binary("Cover")

    # Relational Fields
    publisher_id = fields.Many2one(
        "res.partner", string="Publisher")
    author_ids = fields.Many2many(
        "res.partner", string="Authors")
```

Argumentos

```
    name = fields.Char(
        "Title",
        default=None,
        help="Book cover title.",
        readonly=False,
        required=True,
        index=True,
```

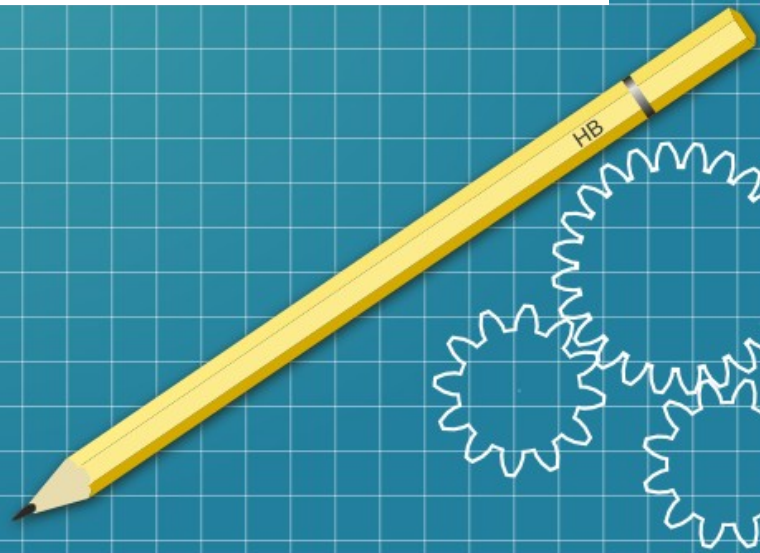
Valores defecto funciones

```
    def _default_last_borrow_date(self):
        return fields.Datetime.now()

    last_borrow_date = fields.Datetime(
        "Last Borrowed On",
        default=_default_last_borrow_date,
    )
```

```
    last_borrow_date = fields.Datetime(
        "Last Borrowed On",
        default="_default_last_borrow_date",
    )

    def _default_last_borrow_date(self):
        return fields.Datetime.now()
```

Campos reservados

- `name` or `x_name` of the `Char` type: These are used by default as the display name for the record. But a different field can be used for the display name by setting the `_rec_name` model attribute. Non-character field types are also known to work for this, and a number to text conversion will be forced for this.



Campos calculados

```
    publisher_country_id = fields.Many2one(
        "res.country", string="Publisher Country",
        compute="_compute_publisher_country",
    )

    @api.depends("publisher_id.country_id")
    def _compute_publisher_country(self):
        for book in self:
            book.publisher_country_id =
              book.publisher_id.country_id
```

database by adding the `store = True` attribute. They will be recomputed when any of their dependencies change. Since the values are now stored, they can be searched just like regular fields, and a search function is not needed.

Campos related

```
    publisher_country_id = fields.Many2one(
        "res.country",
        string="Publisher Country",
        related="publisher_id.country_id",
    )
```

Validación

```
from odoo.exceptions import ValidationError
```

```
    @api.constrains("isbn")
    def _constrain_isbn_valid(self):
        for book in self:
            if book.isbn and not book._check_isbn():
                raise ValidationError(
                    "%s is an invalid ISBN" % book.isbn)
```

Menús

Fichero “views.xml”:

```
<odoo>
  <data>
      <!-- actions opening views on models -->
      <!-- Acciones al abrir las vistas en los modelos
      https://www.odoo.com/documentation/14.0/developer/reference/addons/actions.html
      -->
      <record model="ir.actions.act_window" id="lista_tareas.action_window">
      <field name="name">Listado de tareas pendientes</field>
      <field name="res_model">lista_tareas.lista_tareas</field>
      <field name="view_mode">tree,form</field>
      </record>

      <!-- Top menu item -->
      <menuitem name="Listado de tareas" id="lista_tareas.menu_root"/>

      <!-- menu categories -->
      <menuitem name="Opciones Lista Tareas" id="lista_tareas.menu_1"
parent="lista_tareas.menu_root"/>

      <!-- actions -->
      <menuitem name="Mostrar lista" id="lista_tareas.menu_1_list"
parent="lista_tareas.menu_1"
            action="lista_tareas.action_window"/>
```

Permisos

When no access rights are defined on a model, Odoo determines that no users can access the data. It is even notified in the log:

```
WARNING rd-demo odoo.modules.loading: The models ['estate.property'] have no access rules in module estate, consider adding some
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
```

Access rights are defined as records of the model `ir.model.access`. Each access right is associated with a model, a group (or no group for global access) and a set of permissions: create, read, write and unlink[2]. Such access rights are usually defined in a CSV file named `ir.model.access.csv`.

Here is an example for our previous `test_model`:

```
id,name,model_id/id,group_id/id,perm_read,perm_write,perm_create,perm_unlink
access_test_model,access_test_model,model_test_model,base.group_user,1,0,0,0
```

- `id` is an **external identifier**.
- `name` is the name of the `ir.model.access`.
- `model_id/id` refers to the model which the access right applies to. The standard way to refer to the model is `model_<model_name>`, where `<model_name>` is the `_name` of the model with the `.` replaced by `_`. Seems cumbersome? Indeed it is...
- `group_id/id` refers to the group which the access right applies to.
- `perm_read,perm_write,perm_create,perm_unlink`: read, write, create and unlink permissions