Vangrow

DECIDÍ CÓMO CRECER

DESARROLLO EN ODOO

Vangrow

DECIDÍ CÓMO CRECER

MODELS

Los módulos de Odoo pueden agregar una nueva lógica comercial a un sistema Odoo o modificar y ampliar la lógica comercial existente.

Cada módulo es un directorio dentro de un directorio de módulos . Los directorios de los módulos se especifican mediante la --addons-path opción.

Vangrow

DECIDÍ CÓMO CRECER

ESTRUCTURA DE CLASES

Vangrow

DECIDÍ CÓMO CRECER

Se declara una clase para empezar desarrollar el modelo

Declaramos que esto hereda de models.Model para poder utilizar sus métodos y propiedades.

```
class Mantenimiento(models.Model):

    _name = "garaje.Mantenimiento"
    _description = "Permite definir el mantenimiento rutinarios sobre
conjuntos de coches."
    _order = "Fecha"
    fecha = fields.Date("fecha", required = True, default =
fields.date.today())
       tipo = fields.Selection(string="Tipo", selection=[("l", "Lavar"),
("r", "Revision"), ("m", "Mecanica"), ("p", "Pintura")], default="l")
    coste = fields.Float("Coste", (8,2), help ="Esto es el coste del
mantenimiento total.")
```

Por cada clase se genera una tabla y consulta los datos requerido

Vangrow

DECIDÍ CÓMO CRECER

VARIABLES RESERVADAS

Vangrow

DECIDÍ CÓMO CRECER

VARIABLES RESERVADAS	SIGNIFICADO DE SU USO
_name= “”	Define al modelo real. Si queremos buscar un aparcamiento se debe buscar por garaje.aparcamiento, sería el nombre de aplicacion.laclase que sería el objeto.
_description	Permite definir una descripción acerca de ese objeto.
_order=“”	Permite identificar que orden va tener los datos de la tabla
name=“”	Se define un campo para cuando se busque o se navegue se va poder identificar el objeto.

Vangrow

DECIDÍ CÓMO CRECER

OBJETOS RESERVADOS

Vangrow

DECIDÍ CÓMO CRECER

OBJETOS RESERVADOS	SIGNIFICADO DE SU USO
fields.Char(string="Dirrecion", required=True)	fields.Char permite agregar contenido de texto, pero corto. Una dirección, un país, una ciudad, un nombre, etc..
fields.Text("Descripcion")	Asi como Char funciona para almacenar datos tipos string pero cortos, el Text permite almacenar el mismo dato con la diferencia que es para string largos. Ejemplo: Descripcion.
fields.Integer("Años", compute="_get_anual")	El Integer permite añadir datos numer. Este tipo de datos es un tipo de dato INT que es dato de numero entero. Ejemplo: [1],[2], ect,
fields.Float(string="Coste", (8,2), help ="Esto es el coste del mantenimiento total.")	Al igual que INT es un dato valor numérico. Pero con la diferencia que almacenara números decimales. Para eso es necesario agregarle dentro de float (1,2) una tupla. Ya que definimos la cantidad de numero entero que tendrá y decimales.

Vangrow

DECIDÍ CÓMO CRECER

OBJETOS RESERVADOS	SIGNIFICADO DE SU USO
fields.Date(string="fecha", required = True, default = fields.date.today())	Date es un campo de fecha. Solamente almacenara la fecha ejemplo [28/10/2022]
fields.date.today()	Funciona exactamente lo mismo que Data, solamente que estamos solicitando que nos brinde la fecha actual,
fields.Selection(string="Tipo", selection=[("l", "Lavar"), ("r", "Revision"), ("m", "Mecanica"), ("p", "Pintura")], default="l")	El Selection es un tipo de selección que el usuario estará eligiendo un dato especifico que se le proporciona. Para agregarle el contenido a la selección es necesario hacerlo mediante corchete y dentro de los corches dividir por tuplas cada seleccion, [(),(),()]

Vangrow

DECIDÍ CÓMO CRECER

VARIABLES RESERVADA PARA EL USO DE OBJETOS RESERVADOS

Vangrow

DECIDÍ CÓMO CRECER

VARIABLES RESERVADA PARA APLICAR A LOS OBJETOS RESERVADOS	SIGNIFICADO DE SU USO
default=“”	El default te permite que si no agregan un contenido, tomara el contenido que hayas elegido vos. Ejemplo si tengo un numero del 1 – 10, pero si no elijen ninguno se le aplica un default para que se le ponga 1. Puede tener cualquier valor.
string=“”	La etiqueta del campo en la interfaz de usuario (visible para los usuarios). En este caso se utiliza tipo de dato cadena de texto.
required=“”	El required permite que el campo que se le solicita la información para almacenar al usuario sea requerido. Para esto utiliza campo booleano “Verdadero” o “Falso”
size=“”	Coloca un limite de caracteres que puede llegar tener un campo. Recibe tipo de dato numero int (entero)
help =“”	Permite darle algo descriptivo al usuario para que el mismo no se pueda confundir.

Vangrow

DECIDÍ CÓMO CRECER

RELACIONES ENTRE TABLAS

Vangrow

DECIDÍ CÓMO CRECER

RELACIONES ENTRE TABLAS	SIGNIFICADO DE SU USO
fields.One2many	Significa cuando estamos consultando uno a muchos. Por ejemplo en un estacionamiento hay muchos coches (uno=estacionamientos / muchos=coches)
fields.Many2One	Significa cuando estamos consultando de muchos a uno. Ejemplos hay muchos autos pero solo un estacionamiento.
fields.Many2Many	Significa mucho a muchos, por ejemplo si tenemos muchos autos también abra muchos mantenimientos.

Para las relaciones permite la DB gestionarte una segunda tabla de datos para poder realizar consultas serias con respecto a las misma.