**Kevin Gómez Valderas 2ºDAM 24-25**

**ZOO Parte 3**

**Añadir funcionalidad**

**1. Añadir un modelo tipocomidaanimales, con un campo nombre. Los animales comen pienso, carne, gusanos, zanahorias, etc.**

| **Modelo** |
| --- |
| **class TipoComida(models.Model):**  **\_name = 'zoo.tipo.comida'**  **\_description = 'Tipos de comida para animales'**  **name = fields.Char(string='Nombre', required=True)** |
| **Vista** |
| **<odoo>**  **<record id="action\_tipo\_comida" model="ir.actions.act\_window">**  **<field name="name">Tipo Comida</field>**  **<field name="res\_model">zoo.tipo.comida</field>**  **<field name="view\_mode">tree,form</field>**  **<field name="help" type="html">**  **<p class="o\_view\_nocontent\_smiling\_face">**  **Gestiona los tipos de comida para los animales.**  **</p>**  **</field>**  **</record>**  **<menuitem id="menu\_tipo\_comida"**  **name="Tipo Comida"**  **parent="menu\_zoo\_config"**  **action="action\_tipo\_comida"**  **sequence="90" />**  **<record id="view\_tipo\_comida\_search" model="ir.ui.view">**  **<field name="name">zoo.tipo.comida.search</field>**  **<field name="model">zoo.tipo.comida</field>**  **<field name="arch" type="xml">**  **<search>**  **<field name="name" />**  **</search>**  **</field>**  **</record>**  **<!-- Vista de Árbol -->**  **<record id="view\_tipo\_comida\_tree" model="ir.ui.view">**  **<field name="name">zoo.tipo.comida.tree</field>**  **<field name="model">zoo.tipo.comida</field>**  **<field name="arch" type="xml">**  **<tree>**  **<field name="name" />**  **</tree>**  **</field>**  **</record>**  **<!-- Vista de Formulario -->**  **<record id="view\_tipo\_comida\_form" model="ir.ui.view">**  **<field name="name">zoo.tipo.comida.form</field>**  **<field name="model">zoo.tipo.comida</field>**  **<field name="arch" type="xml">**  **<form>**  **<field name="name" />**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Resultado** |
|  |

**2. Añadir dieta recomendada especie. Se mostrará como lineas dentro de cada especie. Sus campos son el tipo de comida (referencia al modelo comida animales) y la cantidad en kilos. Con esto indicamos que le vamos a dar de comer a un animal de esa especie por defecto en un día. Por ejemplo a los camellos les podemos dar 5 kilos de zanahorias y 4 de puerros al día.**

| **Modelo** |
| --- |
| **Especies** |
| **dieta\_recomendada\_ids = fields.One2many('zoo.dieta.recomendada', 'especie\_id', string="Dieta recomendada")** |
| **DietaRecomendada** |
| **class DietaRecomendada(models.Model):**  **\_name = 'zoo.dieta.recomendada'**  **\_description = 'Dieta recomendada para cada especie'**  **especie\_id = fields.Many2one('zoo.especies', string="Especie", required=True, ondelete='cascade')**  **tipo\_comida\_id = fields.Many2one('zoo.tipo.comida', string="Tipo de comida", required=True)**  **cantidad\_kg = fields.Float(string="Cantidad (kg)", required=True, help="Cantidad diaria recomendada en kilogramos")** |
| **Vista** |
| **Especies** |
| **<notebook>**  **<page string="Dieta Recomendada">**  **<field name="dieta\_recomendada\_ids">**  **<tree editable="bottom">**  **<field name="tipo\_comida\_id" />**  **<field name="cantidad\_kg" />**  **</tree>**  **</field>**  **</page>**  **</notebook>** |
| **Resultado** |
|  |

**3. Un botón en los continentes que de un mensaje indicando de cuantas especies diferentes de este continente hay en el zoo (puedes mostrar un mensaje como con el siguiente ejempo**

**from odoo.exceptions import ValidationError, UserError**

**def boton\_aviso(self):**

**raise UserError('Cuidaaaoooo')**

| **Model** |
| --- |
| **especie\_count = fields.Integer(string='Número de especies', compute='\_compute\_especie\_count')**  **def \_compute\_especie\_count(self):**  **for record in self:**  **record.especie\_count = self.env['zoo.especies'].search\_count([('continente\_ids', '=', record.id)])**  **def boton\_aviso(self):**  **for record in self:**  **message = f'Hay {record.especie\_count} especies diferentes de este continente en el zoológico.'**  **raise UserError(message)** |
| **Vista** |
| **<div style="width: 200px;" >**  **<button name="boton\_aviso"**  **type="object"**  **string="Ver número de especies" />**  **</div>** |
| **Resultado** |
|  |

**4.**

**1. Añadir campo salario base al modelo con los estados laborales.**

**2. Mostrar un campo salario en el modelo de empleados donde el salario depende del estado laboral y de los estudios. El salario aumenta un 10% si se tiene la ESO y un 50 % si se tienen estudios superiores.**

| **Modelos** |
| --- |
| **EstadosLaborales** |
| **salario\_base = fields.Float(string='Salario Base', required=True)** |
| **Cuidadores** |
| **# Nuevo campo computado para el salario**  **salario = fields.Float(**  **string='Salario',**  **compute='\_compute\_salario',**  **store=True**  **)**  **@api.depends('estado\_laboral\_id', 'estudios')**  **def \_compute\_salario(self):**  **for record in self:**  **salario\_base = record.estado\_laboral\_id.salario\_base if record.estado\_laboral\_id else 0.0**  **incremento = 1.0 # Sin estudios, sin incremento**  **if record.estudios == 'eso':**  **incremento += 0.10 # Aumento del 10% si tiene ESO**  **elif record.estudios in ['ciclo\_grado\_superior', 'universitario']:**  **incremento += 0.50 # Aumento del 50% si tiene estudios superiores**  **record.salario = salario\_base \* incremento** |
| **Vista** |
| **EstadosLaborales** |
| **<field name="salario\_base" string="Salario Base"/>** |
| **Cuidadores** |
| **<field name="salario" string="Salario" />** |
| **Resultado** |
|  |

**3. Añadir un botón en el formulario de estados laborales que calcule la masa salarial total para todos los empleados (lo que cobran todos los empleados juntos teniendo el cuenta el incremento por estudios).**

| **Modelo** |
| --- |
| **masa\_salarial = fields.Float(string="Masa Salarial Total", compute="calcular\_masa\_salarial", store=False)**  **def calcular\_masa\_salarial(self):**  **"""**  **Método para calcular la masa salarial total de todos los cuidadores**  **teniendo en cuenta el incremento por estudios.**  **"""**  **# Obtener todos los cuidadores que tengan este estado laboral**  **cuidadores = self.env['zoo.cuidadores'].search([('estado\_laboral\_id', '=', self.id)])**  **# Calcular la masa salarial**  **masa\_salarial = sum(cuidador.salario for cuidador in cuidadores)**  **# Lanzar un error de usuario con el resultado**  **raise UserError("Masa Salarial Total: " + str(masa\_salarial) + "€")** |
| **Vista** |
| **<div style="width: 200px;">**  **<button name="calcular\_masa\_salarial" type="object" string="Calcular Masa Salarial" class="oe\_highlight"/>**  **</div>** |
| **Resultado** |
|  |

**4. Añadir campo en el modelo de estados laborales donde se muestre la media de los salarios de los empleados para ese estado laboral.**

| **Modelo** |
| --- |
| **# Campo de la media de salarios**  **media\_salarios = fields.Float(string="Media de Salarios", compute="\_compute\_media\_salarios", store=False)**  **def \_compute\_media\_salarios(self):**  **"""**  **Método para calcular la media de los salarios de los empleados**  **para el estado laboral actual.**  **"""**  **# Obtener todos los cuidadores que tengan este estado laboral**  **cuidadores = self.env['zoo.cuidadores'].search([('estado\_laboral\_id', '=', self.id)])**  **# Si no hay cuidadores, la media será 0**  **if not cuidadores:**  **self.media\_salarios = 0.0**  **else:**  **# Sumar los salarios de todos los cuidadores**  **total\_salarios = sum(cuidador.salario for cuidador in cuidadores)**  **# Calcular la media**  **self.media\_salarios = total\_salarios / len(cuidadores)** |
| **Vista** |
| **<field name="media\_salarios" />** |
| **Resultado** |
|  |

**5. Añadir botón cría en animales. Crea un nuevo animal de nombre “Cría de” y el nombre del animal activo. Todos los datos son iguales excepto el estado (normal) y la fecha de llegada y nacimiento (fecha de hoy)**

| **Modelo** |
| --- |
| **def action\_cria(self):**  **for animal in self:**  **# Crear el nuevo animal (cría)**  **cria = self.env['zoo.animales'].create({**  **'name': f'Cría de {animal.name}',**  **'especie\_id': animal.especie\_id.id,**  **'habitaculo\_id': animal.habitaculo\_id.id,**  **'fecha\_llegada': fields.Date.today(),**  **'fecha\_nacimiento': fields.Date.today(),**  **'nacido\_zoo': True,**  **'cuidador\_id': animal.cuidador\_id.id,**  **'state': 'normal',**  **})**  **# Retornar la acción para abrir la vista del formulario del nuevo animal**  **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.animales',**  **'res\_id': cria.id,**  **'view\_mode': 'form',**  **'view\_type': 'form',**  **'target': 'current', # Abrir en la ventana actual**  **}** |
| **Vista** |
| **<div style="width: 100px">**  **<button name="action\_cria" string="Crear Cría" type="object" class="oe\_highlight"/>**  **</div>** |
| **Resultado** |
|  |

**6.**

**1. Añadir un campo con el máximo de animales para el modelo habitáculo.**

**2. Al cambiar de habitáculo de un animal si sobrepasamos el total de animales en este habitáculo que nos de un warning. Hacerlo con read\_count (tener cuidado porque el animal en que acabamos de cambiar no está en la base de datos como en este habitáculo y no lo contará en el read\_count).**

| **Modelo** |
| --- |
| **@api.model**  **def create(self, vals):**  **# Crear el animal**  **animal = super(Animales, self).create(vals)**  **# Verificar si el habitáculo alcanza el límite**  **if animal.habitaculo\_id.total\_animales > animal.habitaculo\_id.max\_animales:**  **raise ValidationError('El habitáculo ha alcanzado el máximo de animales permitidos.')**  **return animal**  **def write(self, vals):**  **if 'habitaculo\_id' in vals:**  **# Obtener el habitáculo actual y el nuevo**  **habitaculo\_anterior = self.habitaculo\_id**  **habitaculo\_nuevo = self.env['zoo.habitaculos'].browse(vals['habitaculo\_id'])**  **# Comprobar si el cambio supera el máximo de animales**  **if habitaculo\_nuevo.total\_animales + 1 > habitaculo\_nuevo.max\_animales:**  **raise ValidationError('El habitáculo ha alcanzado el máximo de animales permitidos.')**  **# Realizar la escritura**  **return super(Animales, self).write(vals)** |
| **Vista** |
| **<field name="max\_animales" />** |
| **Resultado** |
|  |

**7. Wizard “Cambiar especie de habitáculo”. Indicamos una especie y un habitáculo de origen y uno de destino. Cambiamos el habitáculo a todos los animales de esa especie que estaban en el habitáculo de origen por el de destino.**

| **Modelo** |
| --- |
| **ZooCambioHabitaculoWizard** |
| **class CambioHabitaculoWizard(models.TransientModel):**  **\_name = 'zoo.cambio.habitaculo.wizard'**  **\_description = 'Wizard para cambiar el habitáculo de una especie'**  **especie\_id = fields.Many2one('zoo.especies', string="Especie", required=True)**  **habitaculo\_origen\_id = fields.Many2one('zoo.habitaculos', string="Habitáculo de Origen", required=True)**  **habitaculo\_destino\_id = fields.Many2one('zoo.habitaculos', string="Habitáculo de Destino", required=True)**  **def action\_cambiar\_habitaculo(self):**  **""" Cambia los animales de una especie del habitáculo de origen al de destino """**  **animales = self.env['zoo.animales'].search([**  **('especie\_id', '=', self.especie\_id.id),**  **('habitaculo\_id', '=', self.habitaculo\_origen\_id.id)**  **])**  **for animal in animales:**  **animal.habitaculo\_id = self.habitaculo\_destino\_id**  **return {'type': 'ir.actions.act\_window\_close'}**  **def action\_cancel(self):**  **""" Cierra el wizard sin hacer cambios """**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Habitaculos** |
| **def action\_abrir\_wizard\_cambio\_habitaculo(self):**  **""" Abre el wizard para cambiar el habitáculo de una especie """**  **return {**  **'name': 'Cambiar Especie de Habitáculo',**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.cambio.habitaculo.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **CambioHabitaculo** |
| **<odoo>**  **<record id="action\_zoo\_cambio\_habitaculo\_wizard" model="ir.actions.act\_window">**  **<field name="name">Cambiar Especie de Habitáculo</field>**  **<field name="res\_model">zoo.cambio.habitaculo.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<record id="view\_zoo\_cambio\_habitaculo\_wizard" model="ir.ui.view">**  **<field name="name">zoo.cambio.habitaculo.wizard.form</field>**  **<field name="model">zoo.cambio.habitaculo.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Cambiar Habitáculo de Especie">**  **<group>**  **<field name="especie\_id" />**  **<field name="habitaculo\_origen\_id" />**  **<field name="habitaculo\_destino\_id" />**  **</group>**  **<footer>**  **<button string="Cambiar Habitáculo" type="object"**  **name="action\_cambiar\_habitaculo" class="btn-primary" />**  **<button string="Cancelar" type="object" name="action\_cancel" class="btn-default" />**  **</footer>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Habitaculos** |
| **<div style="width: 300px;">**  **<button name="action\_abrir\_wizard\_cambio\_habitaculo"**  **string="Cambiar Especie de Habitáculo"**  **type="object"**  **class="btn-primary" />**  **</div>** |
| **CSV** |
| **access\_zoo\_cambio\_habitaculo\_wizard,zoo.cambio.habitaculo.wizard,model\_zoo\_cambio\_habitaculo\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**8. Wizard eliminar alimento de dieta de especies. Indicamos uno o varios alimentos y una o varias especies los eliminamos de todas las dietas de estas especies.**

| **Modelo** |
| --- |
| **EliminarAlimento** |
| **class EliminarAlimentoWizard(models.TransientModel):**  **\_name = 'zoo.eliminar.alimento.wizard'**  **\_description = 'Wizard para eliminar alimentos de la dieta de especies'**  **especie\_ids = fields.Many2many('zoo.especies', string="Especies", required=True)**  **tipo\_comida\_ids = fields.Many2many('zoo.tipo.comida', string="Tipos de comida", required=True)**  **def action\_eliminar\_alimentos(self):**  **""" Elimina los alimentos seleccionados de las dietas de las especies elegidas """**  **for especie in self.especie\_ids:**  **dietas\_a\_eliminar = self.env['zoo.dieta.recomendada'].search([**  **('especie\_id', '=', especie.id),**  **('tipo\_comida\_id', 'in', self.tipo\_comida\_ids.ids)**  **])**  **dietas\_a\_eliminar.unlink() # Elimina las dietas filtradas**  **return {'type': 'ir.actions.act\_window\_close'}**  **def action\_cancel(self):**  **""" Cierra el wizard sin hacer cambios """**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Especies** |
| **def action\_abrir\_wizard\_eliminar\_alimentos(self):**  **""" Abre el wizard para eliminar alimentos de la dieta """**  **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.eliminar.alimento.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **EliminarAlimento** |
| **<odoo>**  **<record id="action\_zoo\_eliminar\_alimento\_wizard" model="ir.actions.act\_window">**  **<field name="name">Eliminar Alimentos de Dieta</field>**  **<field name="res\_model">zoo.eliminar.alimento.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<record id="view\_zoo\_eliminar\_alimento\_wizard" model="ir.ui.view">**  **<field name="name">zoo.eliminar.alimento.wizard.form</field>**  **<field name="model">zoo.eliminar.alimento.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Eliminar Alimentos de la Dieta">**  **<group>**  **<field name="especie\_ids" widget="many2many\_tags" />**  **<field name="tipo\_comida\_ids" widget="many2many\_tags" />**  **</group>**  **<footer>**  **<button name="action\_eliminar\_alimentos" string="Eliminar" type="object"**  **class="btn-danger" />**  **<button name="action\_cancel" string="Cancelar" class="btn-secondary"**  **special="cancel" />**  **</footer>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Especies** |
| **<div style="width: 300px;">**  **<button name="action\_abrir\_wizard\_eliminar\_alimentos"**  **string="Eliminar Alimentos de Dieta"**  **type="object"**  **class="btn-danger" />**  **</div>** |
| **CSV** |
| **access\_zoo\_eliminar\_alimento\_wizard,zoo.eliminar.alimento.wizard,model\_zoo\_eliminar\_alimento\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**9. Wizard añadir continente. Seleccionamos una serie de especies y un continente. Se añada el continente a todas las especies.**

| **Modelo** |
| --- |
| **WizardAnadirContinente** |
| **class WizardAnadirContinente(models.TransientModel):**  **\_name = 'zoo.anadir.continente.wizard'**  **\_description = 'Wizard para añadir continente a especies'**    **especie\_ids = fields.Many2many('zoo.especies', string="Especies", required=True)**  **continente\_ids = fields.Many2one('zoo.continentes', string='Continente a añadir', required=True)**  **def action\_add\_continent(self):**  **"""Asocia el continente seleccionado a las especies seleccionadas sin borrar los continentes anteriores"""**  **for especie in self.especie\_ids:**  **# Verificar si el continente ya está asociado**  **if self.continente\_ids not in especie.continente\_ids:**  **# Añadir el continente al campo Many2many**  **especie.write({**  **'continente\_ids': [(4, self.continente\_ids.id)]**  **})**  **return {'type': 'ir.actions.act\_window\_close'}**  **def action\_cancel(self):**  **"""Cierra el wizard sin hacer cambios"""**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Si queremos sobreescribir los continentes** |
| **def action\_add\_continent(self):**  **"""Asocia el continente seleccionado a las especies seleccionadas"""**  **for especie in self.especie\_ids:**  **especie.continente\_ids = self.continente\_ids**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Especies** |
| **def action\_abrir\_wizard\_añadir\_continente(self):**  **""" Abre el wizard para añadir continente a las especies seleccionadas """**  **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.anadir.continente.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **<odoo>**  **<record id="view\_zoo\_anadir\_continente\_wizard\_form" model="ir.ui.view">**  **<field name="name">zoo.anadir.continente.wizard.form</field>**  **<field name="model">zoo.anadir.continente.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Añadir continente a especies">**  **<sheet>**  **<group>**  **<field name="especie\_ids" widget="many2many\_tags" />**  **<field name="continente\_ids" options="{'no\_create': True}" />**  **</group>**  **<footer>**  **<button string="Añadir" type="object" name="action\_add\_continent"**  **class="btn-primary" />**  **<button string="Cancelar" type="object" name="action\_cancel"**  **class="btn-secondary" />**  **</footer>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **<!-- Acción de menú para abrir el wizard -->**  **<record id="action\_abrir\_wizard\_añadir\_continente" model="ir.actions.act\_window">**  **<field name="name">Añadir continente a especies</field>**  **<field name="res\_model">zoo.anadir.continente.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **</odoo>** |
| **CSV** |
| **access\_zoo\_anadir\_continente\_wizard,zoo.anadir.continente.wizard,model\_zoo\_anadir\_continente\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**10. Wizard añadir alimento a la dieta. Indicamos una o varias especies, un alimento y una cantidad y añadimos este alimento con la cantidad a la dieta de las especies indicadas.**

| **Modelo** |
| --- |
| **AgregarAlimentoWizard** |
| **class AgregarAlimentoWizard(models.TransientModel):**  **\_name = 'zoo.agregar.alimento.wizard'**  **\_description = 'Wizard para agregar alimentos a la dieta de especies'**  **especie\_ids = fields.Many2many('zoo.especies', string="Especies", required=True)**  **tipo\_comida\_ids = fields.Many2many('zoo.tipo.comida', string="Tipos de comida", required=True)**  **cantidad = fields.Float(string="Cantidad", required=True)**  **def action\_agregar\_alimentos(self):**  **""" Agrega los alimentos seleccionados a las dietas de las especies elegidas """**  **dieta\_model = self.env['zoo.dieta.recomendada']**    **for especie in self.especie\_ids:**  **for tipo\_comida in self.tipo\_comida\_ids:**  **# Buscar si ya existe un registro de dieta para esta especie y alimento**  **dieta\_existente = dieta\_model.search([**  **('especie\_id', '=', especie.id),**  **('tipo\_comida\_id', '=', tipo\_comida.id)**  **], limit=1)**  **if dieta\_existente:**  **# Si ya existe, actualizar la cantidad**  **dieta\_existente.sudo().write({'cantidad': dieta\_existente.cantidad\_kg + self.cantidad})**  **else:**  **# Si no existe, crear una nueva dieta**  **dieta\_model.sudo().create({**  **'especie\_id': especie.id,**  **'tipo\_comida\_id': tipo\_comida.id,**  **'cantidad\_kg': self.cantidad**  **})**  **return {'type': 'ir.actions.act\_window\_close'}**  **def action\_cancel(self):**  **""" Cierra el wizard sin hacer cambios """**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Especies** |
| **def action\_abrir\_wizard\_agregar\_alimento(self):**  **""" Abre el wizard para agregar alimento a las especies seleccionadas """**  **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.agregar.alimento.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **<odoo>**  **<record id="action\_zoo\_agregar\_alimento\_wizard" model="ir.actions.act\_window">**  **<field name="name">Eliminar Alimentos de Dieta</field>**  **<field name="res\_model">zoo.agregar.alimento.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<record id="view\_zoo\_agregar\_alimento\_wizard" model="ir.ui.view">**  **<field name="name">zoo.agregar.alimento.wizard.form</field>**  **<field name="model">zoo.agregar.alimento.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Agregar Alimentos de la Dieta">**  **<group>**  **<field name="especie\_ids" widget="many2many\_tags" />**  **<field name="tipo\_comida\_ids" widget="many2many\_tags" />**  **<field name="cantidad" />**  **</group>**  **<footer>**  **<button name="action\_agregar\_alimentos" string="Agregar" type="object"**  **class="btn-danger" />**  **<button name="action\_cancel" string="Cancelar" class="btn-secondary"**  **special="cancel" />**  **</footer>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
|  |
| **access\_zoo\_agregar\_alimento\_wizard,zoo.agregar.alimento.wizard,model\_zoo\_agregar\_alimento\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**11. Wizard para dar de alta grupos de animales. Indicamos todos los datos de un animal y un número de animales que forman el grupo. Se añaden todos llamándolos Nombre 1, Nombre 2, etc. La fecha de llegada es la fecha de hoy. Se deja vacía la fecha de nacimiento.**

| **Modelo** |
| --- |
| **AltaAnimalesWizard** |
| **class AltaAnimalesWizard(models.TransientModel):**  **\_name = 'zoo.alta.animales.wizard'**  **\_description = 'Wizard para alta de grupos de animales'**  **name = fields.Char(string='Nombre base', required=True)**  **especie\_id = fields.Many2one('zoo.especies', string='Especie', required=True)**  **habitaculo\_id = fields.Many2one('zoo.habitaculos', string='Habitáculo', required=True)**  **num\_animales = fields.Integer(string='Número de animales', required=True, default=1)**  **def action\_crear\_grupo(self):**  **if self.num\_animales < 1:**  **raise ValidationError('Debe haber al menos un animal en el grupo.')**    **animales = []**  **for i in range(1, self.num\_animales + 1):**  **nombre\_animal = f"{self.name} {i}"**  **animal = self.env['zoo.animales'].create({**  **'name': nombre\_animal,**  **'especie\_id': self.especie\_id.id,**  **'habitaculo\_id': self.habitaculo\_id.id,**  **'fecha\_llegada': date.today(),**  **'fecha\_nacimiento': False, # Fecha de nacimiento vacía**  **'state': 'normal',**  **})**  **animales.append(animal.id)**    **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.animales',**  **'view\_mode': 'tree,form',**  **'domain': [('id', 'in', animales)],**  **'target': 'current',**  **}**    **def action\_cancel(self):**  **""" Cierra el wizard sin hacer cambios """**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Especies** |
| **def action\_abrir\_wizard\_alta\_animales(self):**  **""" Abre el wizard para alta animales a las especies seleccionadas """**  **return {**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.alta.animales.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **AltaAnimales** |
| **<odoo>**  **<record id="action\_zoo\_alta\_animales\_wizard" model="ir.actions.act\_window">**  **<field name="name">Alta animales de Dieta</field>**  **<field name="res\_model">zoo.alta.animales.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<record id="view\_zoo\_alta\_animales\_wizard" model="ir.ui.view">**  **<field name="name">zoo.alta.animales.wizard.form</field>**  **<field name="model">zoo.alta.animales.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Alta animales">**  **<group>**  **<field name="name" />**  **<field name="especie\_id" widget="many2one\_tags" />**  **<field name="habitaculo\_id" widget="many2one\_tags" />**  **<field name="num\_animales" />**  **</group>**  **<footer>**  **<button name="action\_crear\_grupo" string="Crear Grupo" type="object"**  **class="btn-danger" />**  **<button name="action\_cancel" string="Cancelar" class="btn-secondary"**  **special="cancel" />**  **</footer>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Especies** |
| **<div style="width: 300px; margin: 10px;">**  **<button name="action\_abrir\_wizard\_alta\_animales"**  **string="Alta Animal"**  **type="object"**  **class="oe\_highlight" />**  **</div>** |
| **CSV** |
| **access\_zoo\_alta\_animales\_wizard,zoo.alta.animales.wizard,model\_zoo\_alta\_animales\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**12.**

**1. Añadimos registro comida (la comida real que ha comido un animal ) como lineas dentro de cada animal. Tenemos los datos fecha, tipo y comida y cantidad.**

| **Modelo** |
| --- |
| **Comida** |
| **class RegistroComida(models.Model):**  **\_name = 'zoo.registro.comida'**  **\_description = 'Registro de comida de los animales'**    **animal\_id = fields.Many2one('zoo.animales', string='Animal', required=True, ondelete='cascade')**  **fecha = fields.Date(string='Fecha', required=True, default=fields.Date.context\_today)**  **tipo\_comida\_id = fields.Many2one('zoo.tipo.comida', string='Tipo de comida', required=True)**  **cantidad\_kg = fields.Float(string='Cantidad (kg)', required=True, help='Cantidad real consumida en kilogramos', default=0.0)**  **@api.onchange('animal\_id')**  **def \_onchange\_animal\_id(self):**  **""" Muestra solo los tipos de comida de la dieta recomendada y da una advertencia si no hay dieta definida """**  **if self.animal\_id:**  **dieta\_recomendada = self.env['zoo.dieta.recomendada'].search([**  **('especie\_id', '=', self.animal\_id.especie\_id.id)**  **])**  **comidas\_permitidas = dieta\_recomendada.mapped('tipo\_comida\_id')**  **if comidas\_permitidas:**  **# Asigna el tipo de comida permitido al campo tipo\_comida\_id**  **self.tipo\_comida\_id = comidas\_permitidas[0] # Selecciona el primer tipo de comida recomendado**  **# Asigna la cantidad recomendada al campo cantidad\_kg**  **if dieta\_recomendada:**  **self.cantidad\_kg = dieta\_recomendada[0].cantidad\_kg # Toma la cantidad recomendada de la dieta**  **return {'domain': {'tipo\_comida\_id': [('id', 'in', comidas\_permitidas.ids)]}}**  **else:**  **return {**  **'warning': {**  **'title': "Dieta no definida",**  **'message': "No hay dieta registrada para esta especie. Debes crear una nueva dieta antes de continuar."**  **}**  **}**  **@api.onchange('tipo\_comida\_id')**  **def \_onchange\_tipo\_comida\_id(self):**  **""" Da una advertencia si no hay cantidad recomendada para el tipo de comida seleccionado """**  **if self.animal\_id and self.tipo\_comida\_id:**  **dieta = self.env['zoo.dieta.recomendada'].search([**  **('especie\_id', '=', self.animal\_id.especie\_id.id),**  **('tipo\_comida\_id', '=', self.tipo\_comida\_id.id)**  **], limit=1)**  **if not dieta:**  **return {**  **'warning': {**  **'title': "Tipo de comida no recomendado",**  **'message': f"{self.tipo\_comida\_id.name} no está en la dieta recomendada para esta especie."**  **}**  **}**  **# Asigna la cantidad recomendada para este tipo de comida**  **self.cantidad\_kg = dieta.cantidad\_kg**  **@api.onchange('cantidad\_kg')**  **def \_onchange\_cantidad\_kg(self):**  **""" Da una advertencia si la cantidad de comida no es la recomendada o es 0 """**  **if self.animal\_id and self.tipo\_comida\_id:**  **# Verifica si la cantidad es 0 o está vacía**  **if not self.cantidad\_kg or self.cantidad\_kg == 0:**  **return {**  **'warning': {**  **'title': "Cantidad incorrecta",**  **'message': "La cantidad de comida no puede ser 0 o vacía. Por favor, introduce una cantidad válida."**  **}**  **}**    **dieta = self.env['zoo.dieta.recomendada'].search([**  **('especie\_id', '=', self.animal\_id.especie\_id.id),**  **('tipo\_comida\_id', '=', self.tipo\_comida\_id.id)**  **], limit=1)**  **if dieta and self.cantidad\_kg != dieta.cantidad\_kg:**  **return {**  **'warning': {**  **'title': "Cantidad incorrecta",**  **'message': f"La cantidad recomendada para {self.tipo\_comida\_id.name} es {dieta.cantidad\_kg} kg, pero has ingresado {self.cantidad\_kg} kg."**  **}**  **}** |
| **Animales** |
| **registro\_comida\_id = fields.One2many('zoo.registro.comida', 'animal\_id')** |
| **Vista** |
| **Animales** |
| **<group>**  **<separator string="Registro de Comida" colspan="2" />**  **<field name="registro\_comida\_id">**  **<tree editable="bottom">**  **<field name="fecha" width="100" />**  **<field name="tipo\_comida\_id" width="100" />**  **<field name="cantidad\_kg" width="100" />**  **</tree>**  **</field>**  **</group>** |
| **CSV** |
| **access\_zoo\_registro\_comida,zoo.registro.comida,model\_zoo\_registro\_comida,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**2. Wizard dar de comer animales en un habitáculo. Indicamos un habitáculo y una fecha de inicio y de fin. Para cada uno de los días indicados si no se ha dado de comer (no existen registros con esa fecha en su diario de alimentación) añadimos los alimentos de la dieta de su especie.**

| **Modelo** |
| --- |
| **AlimentacionHabitaculoWizard** |
| **class AlimentacionHabitaculoWizard(models.TransientModel):**  **\_name = 'zoo.alimentar.habitaculo.wizard'**  **\_description = 'Registrar alimentación en habitáculo'**  **habitaculo\_id = fields.Many2one('zoo.habitaculos', string='Habitáculo', required=True)**  **fecha\_inicio = fields.Date(string='Fecha de inicio', required=True)**  **fecha\_fin = fields.Date(string='Fecha de fin', required=True)**  **def action\_alimentar\_animales(self):**  **RegistroComida = self.env['zoo.registro.comida']**  **for wizard in self:**  **animales = wizard.habitaculo\_id.animal\_ids**  **fechas = [wizard.fecha\_inicio + timedelta(days=i) for i in range((wizard.fecha\_fin - wizard.fecha\_inicio).days + 1)]**  **# Obtener registros existentes en un solo query**  **registros\_existentes = RegistroComida.search([**  **('animal\_id', 'in', animales.ids),**  **('fecha', 'in', fechas)**  **])**  **fechas\_registradas = {(r.animal\_id.id, r.fecha) for r in registros\_existentes}**  **# Crear registros solo cuando falten**  **nuevos\_registros = []**  **for fecha in fechas:**  **for animal in animales:**  **if (animal.id, fecha) not in fechas\_registradas:**  **for alimento in animal.especie\_id.dieta\_recomendada\_ids:**  **nuevos\_registros.append({**  **'animal\_id': animal.id,**  **'fecha': fecha,**  **'tipo\_comida\_id': alimento.tipo\_comida\_id.id, # ← Corrección aquí**  **'cantidad\_kg': alimento.cantidad\_kg,**  **})**  **if nuevos\_registros:**  **RegistroComida.create(nuevos\_registros)**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **Habitaculos** |
| **def action\_alimentar\_animales(self):**  **""" Abre el wizard para cambiar el habitáculo de una especie """**  **return {**  **'name': 'Alimentar Animales del Habitaculo',**  **'type': 'ir.actions.act\_window',**  **'res\_model': 'zoo.alimentar.habitaculo.wizard',**  **'view\_mode': 'form',**  **'target': 'new',**  **}** |
| **Vista** |
| **AlimentacionHabitaculoWizard** |
| **<odoo>**  **<record id="action\_zoo\_alimentar\_habitaculo\_wizard" model="ir.actions.act\_window">**  **<field name="name">Alimentar Animales de los Habitáculos de Dieta</field>**  **<field name="res\_model">zoo.alimentar.habitaculo.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<record id="view\_zoo\_alimentar\_habitaculo\_wizard" model="ir.ui.view">**  **<field name="name">zoo.alimentar.habitaculo.wizard.form</field>**  **<field name="model">zoo.alimentar.habitaculo.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Alimentar Animales del Habitaculo">**  **<group>**  **<field name="habitaculo\_id" />**  **<field name="fecha\_inicio" />**  **<field name="fecha\_fin" />**  **</group>**  **<footer>**  **<button name="action\_alimentar\_animales" string="Alimentar Animales" type="object" class="btn-primary"**  **/>**  **<button name="action\_cancel" string="Cancelar" class="btn-danger"**  **special="cancel" />**  **</footer>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Habitáculos** |
| **<div style="width: 300px; margin: 10px;">**  **<button name="action\_alimentar\_animales"**  **string="Alimentar Animales del Habitáculo"**  **type="object"**  **class="btn-primary" />**  **</div>** |
| **Resultado** |
|  |

**13. Añade al menos un par de modelos más para el zoo que usen la estructura cabecera/lineas.**

| **Modelo** |
| --- |
| **HistorialMedico** |
| **class HistorialMedico(models.Model):**  **\_name = 'zoo.historial.medico'**  **\_description = 'Historial médico de los animales'**  **name = fields.Char(string='Referencia', required=True, copy=False, default=lambda self: self.env['ir.sequence'].next\_by\_code('zoo.historial.medico'))**  **animal\_id = fields.Many2one('zoo.animales', string='Animal', required=True, ondelete='cascade')**  **fecha\_inicio = fields.Date(string='Fecha de inicio', required=True, default=fields.Date.today)**  **fecha\_fin = fields.Date(string='Fecha de fin')**  **estado = fields.Selection([**  **('abierto', 'Abierto'),**  **('cerrado', 'Cerrado')**  **], string='Estado', default='abierto')**  **lineas\_historial = fields.One2many('zoo.historial.medico.linea', 'historial\_id', string='Detalles Médicos')**  **def action\_cerrar(self):**  **self.write({'estado': 'cerrado'})** |
| **HistorialMedicoLinea** |
| **class HistorialMedicoLinea(models.Model):**  **\_name = 'zoo.historial.medico.linea'**  **\_description = 'Entrada en el historial médico'**  **historial\_id = fields.Many2one('zoo.historial.medico', string='Historial Médico', required=True, ondelete='cascade')**  **fecha = fields.Date(string='Fecha', required=True, default=fields.Date.today)**  **descripcion = fields.Text(string='Descripción del tratamiento o diagnóstico')**  **veterinario\_id = fields.Many2one('zoo.cuidadores', string='Veterinario')** |
| **Vista** |
| **HistorialMedico** |
| **<odoo>**  **<!-- Action para Historial Médico -->**  **<record id="action\_historial\_medico" model="ir.actions.act\_window">**  **<field name="name">Historial Médico</field>**  **<field name="res\_model">zoo.historial.medico</field>**  **<field name="view\_mode">tree,form</field>**  **</record>**  **<menuitem id="menu\_historial\_medico"**  **name="Historial Médico"**  **parent="menu\_zoo\_config"**  **action="action\_historial\_medico"**  **sequence="60" />**  **<!-- Tree View -->**  **<record id="view\_historial\_medico\_tree" model="ir.ui.view">**  **<field name="name">zoo.historial.medico.tree</field>**  **<field name="model">zoo.historial.medico</field>**  **<field name="arch" type="xml">**  **<tree>**  **<field name="name" />**  **<field name="animal\_id" />**  **<field name="fecha\_inicio" />**  **<field name="fecha\_fin" />**  **<field name="estado" />**  **</tree>**  **</field>**  **</record>**  **<!-- Form View -->**  **<record id="view\_historial\_medico\_form" model="ir.ui.view">**  **<field name="name">zoo.historial.medico.form</field>**  **<field name="model">zoo.historial.medico</field>**  **<field name="arch" type="xml">**  **<form>**  **<header>**  **<field name="estado" widget="statusbar" readonly="1" />**  **<button name="action\_cerrar" string="Cerrar Historial" type="object" class="btn-primary" invisible="estado != 'abierto'" />**  **</header>**  **<sheet>**  **<group>**  **<group colspan="2">**  **<separator string="Información del Historial" colspan="2" />**  **<field name="name" />**  **<field name="animal\_id" />**  **<field name="fecha\_inicio" />**  **<field name="fecha\_fin" />**  **</group>**  **</group>**  **<notebook>**  **<page string="Detalles Médicos">**  **<group>**  **<separator string="Entradas del Historial Médico" colspan="2" />**  **<field name="lineas\_historial">**  **<tree editable="bottom">**  **<field name="fecha" width="100" />**  **<field name="descripcion" width="300" />**  **<field name="veterinario\_id" width="150" />**  **</tree>**  **</field>**  **</group>**  **</page>**  **</notebook>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **CSV** |
| **access\_zoo\_historial\_medico,zoo.historial.medico,model\_zoo\_historial\_medico,base.group\_user,1,1,1,1**  **access\_zoo\_historial\_medico\_linea,zoo.historial.medico.linea,model\_zoo\_historial\_medico\_linea,base.group\_user,1,1,1,1** |
| **Resultado** |
|  |

**14. Añade además al menos tres Wizards de tu invención. Se valorará la complejidad de los wizards.**

| **Modelo** |
| --- |
| **AsignarCuidadoresWizard** |
| **class AsignarCuidadoresWizard(models.TransientModel):**  **\_name = 'zoo.asignar.cuidadores.wizard'**  **\_description = 'Wizard para asignar cuidadores a múltiples habitáculos'**    **# Campos para selección**  **tipo\_habitaculo = fields.Selection([**  **('exterior', 'Exterior'),**  **('aviario', 'Aviario'),**  **('granja', 'Granja'),**  **('terrario', 'Terrario'),**  **('acuario', 'Acuario')],**  **string='Tipo de habitáculo')**    **zona = fields.Char(string='Zona del zoo')**  **superficie\_min = fields.Float(string='Superficie mínima (m²)')**  **superficie\_max = fields.Float(string='Superficie máxima (m²)')**    **# Campo para asignación**  **cuidador\_id = fields.Many2one(**  **'zoo.cuidadores',**  **string='Cuidador a asignar',**  **required=True,**  **domain="[('estado\_laboral\_id.orden', '<=', 3)]" # Solo cuidadores activos**  **)**    **fecha\_efectiva = fields.Date(**  **string='Fecha efectiva',**  **default=fields.Date.today,**  **help="Fecha a partir de la cual se aplicará el cambio"**  **)**    **# Métodos computados**  **total\_habitaculos = fields.Integer(**  **string='Total de habitáculos afectados',**  **compute='\_compute\_total\_habitaculos',**  **store=False**  **)**    **@api.depends('tipo\_habitaculo', 'zona', 'superficie\_min', 'superficie\_max')**  **def \_compute\_total\_habitaculos(self):**  **domain = self.\_prepare\_domain()**  **self.total\_habitaculos = self.env['zoo.habitaculos'].search\_count(domain)**    **def \_prepare\_domain(self):**  **domain = []**  **if self.tipo\_habitaculo:**  **domain.append(('tipo', '=', self.tipo\_habitaculo))**  **if self.zona:**  **domain.append(('zona', 'ilike', self.zona))**  **if self.superficie\_min:**  **domain.append(('superficie', '>=', self.superficie\_min))**  **if self.superficie\_max:**  **domain.append(('superficie', '<=', self.superficie\_max))**  **return domain**    **def action\_asignar\_cuidadores(self):**  **self.ensure\_one()**  **domain = self.\_prepare\_domain()**  **habitaculos = self.env['zoo.habitaculos'].search(domain)**    **if not habitaculos:**  **raise UserError("No hay habitáculos que coincidan con los criterios de búsqueda")**    **# Asignar nuevo cuidador, eliminando el anterior**  **habitaculos.write({'responsable\_id': self.cuidador\_id.id})**    **# Crear notificación**  **message = f"""**  **Se han actualizado {len(habitaculos)} habitáculos:**  **- Nuevo responsable: {self.cuidador\_id.name}**  **- Fecha efectiva: {self.fecha\_efectiva}**  **"""**  **return {**  **'type': 'ir.actions.client',**  **'tag': 'display\_notification',**  **'params': {**  **'title': 'Asignación completada',**  **'message': message,**  **'sticky': True,**  **'type': 'success',**  **}**  **}** |
| **ProgramarLimpiezasWizard** |
| **class ProgramarLimpiezasWizard(models.TransientModel):**  **\_name = 'zoo.programar.limpieza.wizard'**  **\_description = 'Wizard para programar futuras limpiezas'**  **habitaculo\_ids = fields.Many2many('zoo.habitaculos', string='Habitáculos', required=True)**  **nueva\_fecha\_limpieza = fields.Date(string='Nueva Fecha de Limpieza', default=fields.Date.today, readonly=True)**  **responsable\_id = fields.Many2one('zoo.cuidadores', string='Responsable', required=True)**  **def action\_confirmar\_limpieza(self):**  **""" Actualiza la fecha de limpieza en los habitáculos seleccionados """**  **for habitaculo in self.habitaculo\_ids:**  **habitaculo.write({**  **'fecha\_limpieza': self.nueva\_fecha\_limpieza,**  **'responsable\_id': self.responsable\_id.id**  **})**  **return {'type': 'ir.actions.act\_window\_close'}** |
| **AgregarVisitantesEvento** |
| **class AgregarVisitantesEventoWizard(models.TransientModel):**  **\_name = 'zoo.agregar.visitantes.evento.wizard'**  **\_description = 'Asistente para añadir visitantes a un evento'**    **evento\_id = fields.Many2one(**  **'zoo.eventos',**  **string='Evento',**  **required=True,**  **default=lambda self: self.\_context.get('active\_id')**  **)**  **visitantes\_ids = fields.Many2many(**  **'zoo.visitantes',**  **string='Visitantes a añadir',**  **required=True)**    **fecha\_visita = fields.Date(**  **string='Fecha de visita para los nuevos visitantes',**  **default=lambda self: fields.Date.today())**    **tipo\_entrada\_evento = fields.Selection(**  **related='evento\_id.tipo\_evento',**  **string='Tipo de Evento',**  **readonly=True)**    **precio\_evento = fields.Float(**  **related='evento\_id.precio',**  **string='Precio del Evento',**  **readonly=True)**    **capacidad\_disponible = fields.Integer(**  **string='Capacidad disponible',**  **compute='\_compute\_capacidad\_disponible')**    **@api.depends('evento\_id.capacidad\_maxima', 'evento\_id.visitante\_ids', 'visitantes\_ids')**  **def \_compute\_capacidad\_disponible(self):**  **for wizard in self:**  **if wizard.evento\_id:**  **inscritos = len(wizard.evento\_id.visitante\_ids)**  **wizard.capacidad\_disponible = wizard.evento\_id.capacidad\_maxima - inscritos - len(wizard.visitantes\_ids)**  **else:**  **wizard.capacidad\_disponible = 0**    **@api.constrains('visitantes\_ids', 'capacidad\_disponible')**  **def \_check\_capacidad(self):**  **for wizard in self:**  **if wizard.capacidad\_disponible < 0:**  **raise ValidationError('No hay suficiente capacidad para todos los visitantes seleccionados. Capacidad disponible: %d' % wizard.capacidad\_disponible)**    **def action\_add\_visitantes(self):**  **self.ensure\_one()**    **if not self.visitantes\_ids:**  **raise ValidationError('Debe seleccionar al menos un visitante')**    **# Actualizar los visitantes existentes**  **self.visitantes\_ids.write({**  **'fecha\_visita': self.fecha\_visita,**  **'tipo\_entrada': 'general', # O podrías añadir un campo para seleccionar tipo de entrada**  **'total\_gastado': self.precio\_evento,**  **})**    **# Relacionar los visitantes con el evento (asumiendo que hay una relación many2many entre eventos y visitantes)**  **# Si no existe, primero deberías añadirla al modelo zoo.eventos:**  **# visitante\_ids = fields.Many2many('zoo.visitantes', string='Visitantes inscritos')**  **self.evento\_id.write({**  **'visitante\_ids': [(4, visitante.id) for visitante in self.visitantes\_ids]**  **})**    **return {'type': 'ir.actions.act\_window\_close'}** |
| **Eventos** |
| **visitante\_ids = fields.Many2many('zoo.visitantes', string="Visitantes inscritos")**  **inscritos = fields.Integer(**  **string="Visitantes Inscritos",**  **compute="\_compute\_inscritos",**  **store=True**  **)**  **@api.depends('visitante\_ids')**  **def \_compute\_inscritos(self):**  **for evento in self:**  **evento.inscritos = len(evento.visitante\_ids)** |
| **Vista** |
| **AsignarCuidadoresWizard** |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<odoo>**  **<!-- Wizard: Asignar Cuidadores -->**  **<record id="action\_asignar\_cuidadores\_wizard" model="ir.actions.act\_window">**  **<field name="name">Asignar Cuidadores a Habitáculos</field>**  **<field name="res\_model">zoo.asignar.cuidadores.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<!-- Wizard: Asignar Cuidadores -->**  **<record id="view\_asignar\_cuidadores\_wizard\_form" model="ir.ui.view">**  **<field name="name">zoo.asignar.cuidadores.wizard.form</field>**  **<field name="model">zoo.asignar.cuidadores.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Asignar Cuidadores">**  **<sheet>**  **<div class="oe\_title">**  **<h1>Asignación Masiva de Cuidadores</h1>**  **</div>**  **<group>**  **<group string="Filtros de Habitáculos">**  **<field name="tipo\_habitaculo" />**  **<field name="zona" />**  **<field name="superficie\_min" />**  **<field name="superficie\_max" />**  **<field name="total\_habitaculos" widget="progressbar">**  **<progressbar**  **options="{'current\_value': record.total\_habitaculos, 'max\_value': 100, 'editable': false}" />**  **</field>**  **</group>**  **<group string="Asignación">**  **<field name="cuidador\_id" required="1" />**  **<field name="fecha\_efectiva" />**  **</group>**  **</group>**  **<footer>**  **<button name="action\_asignar\_cuidadores" string="Asignar" type="object"**  **class="btn-primary" />**  **<button string="Cancelar" class="btn-secondary" special="cancel" />**  **</footer>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **<menuitem id="menu\_asignar\_cuidadores"**  **name="Asignar Cuidadores"**  **parent="menu\_zoo\_config"**  **action="action\_asignar\_cuidadores\_wizard"**  **sequence="100" />**  **</odoo>** |
| **ProgramarLimpiezasWizard** |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<odoo>**  **<!-- Acción del Wizard -->**  **<record id="action\_programar\_limpieza\_wizard" model="ir.actions.act\_window">**  **<field name="name">Programar Limpieza</field>**  **<field name="res\_model">zoo.programar.limpieza.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<!-- Vista del Wizard -->**  **<record id="view\_programar\_limpieza\_wizard\_form" model="ir.ui.view">**  **<field name="name">zoo.programar.limpieza.wizard.form</field>**  **<field name="model">zoo.programar.limpieza.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Programar Limpieza">**  **<sheet>**  **<div class="oe\_title">**  **<h1>Programación Masiva de Limpiezas</h1>**  **</div>**  **<group>**  **<group string="Configuración">**  **<field name="habitaculo\_ids" widget="many2many\_tags"/>**  **<field name="nueva\_fecha\_limpieza"/>**  **<field name="responsable\_id"/>**  **</group>**  **</group>**  **<footer>**  **<button name="action\_confirmar\_limpieza" string="Programar" type="object" class="btn-primary"/>**  **<button string="Cancelar" class="btn-secondary" special="cancel"/>**  **</footer>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **<!-- Menú de Acceso -->**  **<menuitem id="menu\_programar\_limpieza"**  **name="Programar Limpieza"**  **parent="menu\_zoo\_config"**  **action="action\_programar\_limpieza\_wizard"**  **sequence="110"/>**  **</odoo>** |
| **AgregarVisitantesEvento** |
| **<?xml version="1.0" encoding="UTF-8"?>**  **<odoo>**  **<!-- Acción para abrir el asistente -->**  **<record id="action\_agregar\_visitantes\_evento\_wizard" model="ir.actions.act\_window">**  **<field name="name">Agregar Visitantes al Evento</field>**  **<field name="res\_model">zoo.agregar.visitantes.evento.wizard</field>**  **<field name="view\_mode">form</field>**  **<field name="target">new</field>**  **</record>**  **<!-- Vista del asistente -->**  **<record id="view\_agregar\_visitantes\_evento\_wizard\_form" model="ir.ui.view">**  **<field name="name">zoo.agregar.visitantes.evento.wizard.form</field>**  **<field name="model">zoo.agregar.visitantes.evento.wizard</field>**  **<field name="arch" type="xml">**  **<form string="Agregar Visitantes a un Evento">**  **<sheet>**  **<div class="oe\_title">**  **<h1>Agregar Visitantes al Evento</h1>**  **</div>**  **<group>**  **<group string="Detalles del Evento">**  **<field name="evento\_id" />**  **<field name="tipo\_entrada\_evento" readonly="1" />**  **<field name="precio\_evento" readonly="1" />**  **<field name="capacidad\_disponible" readonly="1" />**  **</group>**  **<group string="Visitantes">**  **<field name="visitantes\_ids" widget="many2many\_tags" />**  **<field name="fecha\_visita" />**  **</group>**  **</group>**  **<footer>**  **<button name="action\_add\_visitantes" string="Añadir Visitantes" type="object" class="btn-primary" />**  **<button string="Cancelar" class="btn-secondary" special="cancel" />**  **</footer>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **<menuitem id="menu\_agregar\_visitantes\_evento"**  **name="Agregar Visitantes"**  **parent="menu\_zoo\_config"**  **action="action\_agregar\_visitantes\_evento\_wizard"**  **sequence="120" />**  **</odoo>** |
| **Eventos** |
| **<page string="Visitantes">**  **<field name="inscritos" />**  **<field name="visitante\_ids" />**  **</page>** |
| **CSV** |
| **access\_zoo\_asignar\_cuidadores\_wizard,zoo.asignar.cuidadores.wizard,model\_zoo\_asignar\_cuidadores\_wizard,base.group\_user,1,1,1,1**  **access\_zoo\_programar\_limpiezas\_wizard,zoo.programar.limpiezas.wizard,model\_zoo\_programar\_limpiezas\_wizard,base.group\_user,1,1,1,1**  **access\_zoo\_agregar\_visitantes\_evento\_wizard,zoo.agregar.visitantes.evento.wizard,model\_zoo\_agregar\_visitantes\_evento\_wizard,base.group\_user,1,1,1,1** |
| **Resultado** |
| **AsignarCuidadoresWizard** |
|  |
| **ProgramarLimpiezasWizard** |
|  |
| **AgregarVisitantesEvento** |
|  |