Kevin Gómez Valderas 2ºDAM 24-25

**Ejercicios Vistas**

**Form Animales**

**1. En animales, cabecera donde aparezca el nombre del animal en grande y la imagen de la especie.**

| **Vista Form** |
| --- |
| **<div style="display: flex; align-items: center;**  **justify-content: flex-start; gap: 16px; height: 120px">**    **<field name="imagen\_especie"**  **widget="image"**  **style="width: 120px; height: 120px; "/>**  **<field name="name"**  **style="margin: 0; font-size: 40pt; height: 120px;"/>**  **</div>** |
| **Resultado** |
|  |

**2. Añadir el campo teléfono (char, los teléfonos siempre son de tipo char) para los cuidadores. Que se muestre en el formulario con el widget badge, junto con el nombre también con el widget badge.**

| **Modelo** |
| --- |
| **class Cuidadores(models.Model):**  **\_name = 'zoo.cuidadores'**  **\_description = 'Cuidadores del zoológico'**    **name = fields.Char(string='Nombre', required=True)**  **telefono = fields.Char(string='Teléfono')** |
| **Vistas** |
| **Tree** |
| **<tree>**  **<field name="name" widget='badge'/>**  **<field name="telefono" widget='badge'/>**  **</tree>** |
| **Form** |
| **<form>**  **<sheet>**  **<group>**  **<field name="name" />**  **<field name="telefono" />**  **</group>**  **</sheet>**  **</form>** |
| **Resultado** |
|  |

**3. Dividir los campos de Animal utilizando group para que ocupen dos columnas. Utiliza algún separador.**

| **Form** |
| --- |
| **<notebook>**  **<page string="Información Básica">**  **<group>**  **<group colspan="2">**  **<separator string="Datos del Animal" colspan="2" />**  **<group>**  **<field name="especie\_id" />**  **<field name="fecha\_nacimiento" />**  **<field name="fecha\_llegada" />**  **<field name="nacido\_zoo" readonly="1" />**  **</group>**  **<separator string="Hábitat y Cuidado" colspan="2" />**  **<group>**  **<field name="habitaculo\_id" />**  **<field name="cuidador\_id" />**  **</group>**  **</group>**  **</group>**  **</page>**  **</notebook>** |

**4. Los campos relacionados con el cuidador los ponemos en una pestaña (notebook)**

| **Form** |
| --- |
| **<notebook>**  **<page string="Información Básica">**  **<group>**  **<group colspan="2">**  **<separator string="Datos del Animal" colspan="2" />**  **<group>**  **<field name="especie\_id" />**  **<field name="fecha\_nacimiento" />**  **<field name="fecha\_llegada" />**  **<field name="nacido\_zoo" readonly="1" />**  **</group>**  **<separator string="Hábitat" colspan="2" />**  **<group>**  **<field name="habitaculo\_id" />**  **</group>**  **</group>**  **</group>**  **</page>**  **<page string="Cuidador">**  **<group>**  **<separator string="Información del Cuidador" colspan="2" />**  **<field name="cuidador\_id" />**  **</group>**  **</page>**  **</notebook>** |
| **Resultado** |
|  |

**5. Añadir un campo state al animal, Normal, Enfermo y En recuperación. El valor por defecto es Normal. Añade un widget statebar a la cabecera pero que no se pueda hacer click para cambiar estado. Añade dos botones. Un botón para decir que está enfermo (solo es visible si el estado es Normal), un botón para pasar de enfermo a en recuperación y otro para de en recuperación a normal (los botones solo son visibles cuando el estado es el adecuado).**

| **Modelo** |
| --- |
| **state = fields.Selection(**  **[('normal', 'Normal'), ('enfermo', 'Enfermo'), ('en\_recuperacion', 'En recuperación')],**  **string='Estado',**  **default='normal'**  **)**  **# Métodos para cambiar el estado**  **def action\_enfermo(self):**  **self.write({'state': 'enfermo'})**  **def action\_recuperacion(self):**  **self.write({'state': 'recuperacion'})**  **def action\_normal(self):**  **self.write({'state': 'normal'})** |
| **Vista** |
| **<header>**  **<field name="state" widget="statusbar" readonly="1" />**  **<button name="action\_normal" string="Normal" type="object"**  **class="btn-primary" invisible="state != 'recuperacion'" />**  **<button name="action\_recuperacion" string="Recuperación"**  **type="object" class="btn-primary" invisible="state != 'enfermo'" />**  **<button name="action\_enfermo" string="Enfermo" type="object"**  **class="btn-primary" invisible="state != 'normal'" />**  **</header>** |
| **Resultado** |
|  |

**6. Añadir un modelo con los posibles estados laborales de los empleados (prácticas, tiempo parcial, falso autonomo, etc). Lo hacemos así porque cambia mucho la legislación. Cada registro aparte del nombre tiene también un campo orden, que es el \_order del modelo.**

| **Modelo** |
| --- |
| **class EstadosLaborales(models.Model):**  **\_name = 'zoo.estados.laborales'**  **\_description = 'Estados laborales de los empleados'**  **\_order = 'orden'**  **name = fields.Char(string='Estado Laboral', required=True)**  **orden = fields.Integer(string='Orden', required=True)** |
| **Vista** |
| **<odoo>**  **<record id="action\_estados\_laborales" model="ir.actions.act\_window">**  **<field name="name">Estados Laborales</field>**  **<field name="res\_model">zoo.estados.laborales</field>**  **<field name="view\_mode">tree,form</field>**  **</record>**  **<menuitem id="menu\_estados\_laborales"**  **name="Estados Laborales"**  **parent="menu\_zoo\_config"**  **action="action\_estados\_laborales"**  **sequence="60" />**  **<record id="view\_estados\_laborales\_tree" model="ir.ui.view">**  **<field name="name">zoo.estados.laborales.tree</field>**  **<field name="model">zoo.estados.laborales</field>**  **<field name="arch" type="xml">**  **<tree>**  **<field name="name" widget='badge' />**  **<field name="orden" />**  **</tree>**  **</field>**  **</record>**  **<record id="view\_estados\_laborales\_form" model="ir.ui.view">**  **<field name="name">zoo.estados.laborales.form</field>**  **<field name="model">zoo.estados.laborales</field>**  **<field name="arch" type="xml">**  **<form>**  **<sheet>**  **<group>**  **<field name="name" />**  **<field name="orden" />**  **</group>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **CSV** |
| **access\_zoo\_estados\_laborales,zoo.estados.laborales,model\_zoo\_estados\_laborales,base.group\_user,1,1,1,1** |
| **Manifest** |
| **'views/EstadosLaborales.xml',** |
| **Resultado** |
|  |

**7. Añadir un campo de estado laboral para los cuidadores relacionado con el modelo de estados laborales.**

**1. El valor por defecto debe ser el estado laboral que tenga el valor 1.**

**2. Utilizamos un widget statebar para ver el estado laboral.**

**3. Añadir un botón que cambie de estado laboral al inmediatamente superior en orden. Si está en el estado 1 pasamos al 2, etc. No hay comprobaciones de que los valores son correctos y que no nos salimos del límite.**

| **Modelo** |
| --- |
| **estado\_laboral\_id = fields.Many2one(**  **'zoo.estados.laborales',**  **string='Estado Laboral',**  **default=lambda self: self.env['zoo.estados.laborales'].search([('orden', '=', 1)], limit=1),**  **)**  **def cambiar\_estado\_laboral(self):**  **for cuidador in self:**  **if cuidador.estado\_laboral\_id:**  **siguiente\_estado = self.env['zoo.estados.laborales'].search([**  **('orden', '=', cuidador.estado\_laboral\_id.orden + 1)**  **], limit=1)**  **if siguiente\_estado:**  **cuidador.estado\_laboral\_id = siguiente\_estado** |
| **Vista** |
| **Form** |
| **<header>**  **<field name="estado\_laboral\_id" widget="statusbar" options="{'clickable': 'True'}"/>**  **<button name="cambiar\_estado\_laboral" string="Siguiente Estado" type="object" class="oe\_highlight"/>**  **</header>** |
| **Tree** |
| **<field name="estado\_laboral\_id"/>** |
| **Resultado** |
|  |

**8. Añadir campo a los cuidadores con los estudios. Pueden ser Eso, Bachillerato, Ciclo Grado Medio, Ciclo Grado Superior o Universitarios. Utilizar un radiobox.**

| **Modelo** |
| --- |
| **estudios = fields.Selection([**  **('eso', 'ESO'),**  **('bachillerato', 'Bachillerato'),**  **('ciclo\_grado\_medio', 'Ciclo Grado Medio'),**  **('ciclo\_grado\_superior', 'Ciclo Grado Superior'),**  **('universitario', 'Universitario')**  **], string='Estudios', required=True)** |
| **Vista** |
| **Form** |
| **<field name="estudios" widget="radio"/>** |
| **Tree** |
| **<field name="estudios" />** |
| **Resultado** |
|  |

**9. Añadir un campo titulo (de estudios) a los cuidadores. En el formulario que solamente sea visible si los estudios son superiores y además debe de ser obligario (required) si los estudios son superiores.**

| **Modelo** |
| --- |
| **estudios = fields.Selection([**  **('eso', 'ESO'),**  **('bachillerato', 'Bachillerato'),**  **('ciclo\_grado\_medio', 'Ciclo Grado Medio'),**  **('ciclo\_grado\_superior', 'Ciclo Grado Superior'),**  **('universitario', 'Universitario')**  **], string='Estudios', required=True)**  **titulo = fields.Char(**  **string='Título',**  **compute='\_compute\_titulo',**  **readonly=False,**  **store=True,**  **)**  **hide\_titulo = fields.Boolean(**  **string='Ocultar Título',**  **compute='\_compute\_hide\_titulo'**  **)**  **# Método para comprobar el requerimiento del campo titulo**  **@api.constrains('estudios', 'titulo')**  **def \_check\_titulo\_required(self):**  **for record in self:**  **if record.estudios in ['ciclo\_grado\_superior', 'universitario'] and not record.titulo:**  **raise ValidationError(**  **'El título es obligatorio para estudios de Ciclo Grado Superior o Universitario.'**  **)**  **# Método controlar la visibilidad del campo titulo**  **@api.depends('estudios')**  **def \_compute\_hide\_titulo(self):**  **for record in self:**  **record.hide\_titulo = record.estudios not in ['ciclo\_grado\_superior', 'universitario']** |
| **Vista** |
| **<field name="hide\_titulo" invisible="1" />**  **<field name="titulo"**  **placeholder="Introduzca el título"**  **force\_save="1"**  **invisible="hide\_titulo" />** |
| **Resultado** |
|  |

**10. En cuidador personaliza las etiquetas de los campos para que aparezca tu nombre (Ej: Estudios Roberto). CAMPO STRING EN FIELD**

| **Vista** |
| --- |
| **<field string="Nombre Kevin" name="name" />**  **<field string="Teléfono Kevin" name="telefono" />**  **<field string="Estudios Kevin" name="estudios" widget="radio" />**  **<field string="Titulo Kevin" name="hide\_titulo" invisible="1" />**  **<field string="Titulo Kevin" name="titulo"**  **placeholder="Introduzca el título"**  **force\_save="1"**  **invisible="hide\_titulo" />** |
| **Resultado** |
|  |

**11. Crea un campo calculado que será algo similar a la urgencia en limpiar en los habitáculos donde divides el número de días que faltan desde hoy hasta el día que toca limpiar, entre el número de días desde la última limpieza hasta que vuelva a tocar a limpiar. Muestra este nuevo campo con un widget de barra de progreso. Si ya se ha sobrepasado la fecha límite mostramos la barra al 100%.**

| **Modelo** |
| --- |
| **urgencia\_limpieza = fields.Float(**  **string='Urgencia de limpieza',**  **compute='\_compute\_urgencia\_limpieza',**  **store=True,**  **help="Indica la urgencia de limpiar el habitáculo (100% = muy urgente, 0% = recién limpiado)"**  **)**  **@api.depends('fecha\_limpieza', 'fecha\_prox\_limpieza', 'tipo')**  **def \_compute\_urgencia\_limpieza(self):**  **today = fields.Date.today()**  **for record in self:**  **if not record.fecha\_limpieza or not record.fecha\_prox\_limpieza:**  **record.urgencia\_limpieza = 100.0**  **continue**    **# Si ya pasó la fecha de próxima limpieza, mostrar 100%**  **if today >= record.fecha\_prox\_limpieza:**  **record.urgencia\_limpieza = 100.0**  **else:**  **# Calcular días totales del ciclo de limpieza**  **dias\_ciclo = (record.fecha\_prox\_limpieza - record.fecha\_limpieza).days**  **# Días transcurridos desde la última limpieza**  **dias\_transcurridos = (today - record.fecha\_limpieza).days**    **# Evitar división por cero**  **if dias\_ciclo <= 0:**  **record.urgencia\_limpieza = 100.0**  **else:**  **# Porcentaje de urgencia (invertido para que 100% sea urgente)**  **record.urgencia\_limpieza = min(100.0, max(0.0, (dias\_transcurridos / dias\_ciclo) \* 100.0))** |
| **Vista** |
| **<field name="urgencia\_limpieza" widget="progressbar"/>** |
| **Resultado** |
|  |

**12. Añadir un botón al habitáculo para indicar que hemos limpiado hoy. Lo ponemos al lado de la fecha de última limpieza. Al pulsar el botón pide confirmación “Esta seguro?”. Añade un icono al botón (puedes ver los iconos a utilizar en la web** [**https://fontawesome.com**](https://fontawesome.com)**, y consultar los que hay disponibles en el fichero web/static/lib/fontawesome/css/font-awesome.css (depende de la instalación puede cambiar donde esté).**

| **Modelo** |
| --- |
| **def marcar\_limpieza\_hoy(self):**  **""" Método para actualizar la fecha de limpieza a hoy """**  **for record in self:**  **record.fecha\_limpieza = fields.Date.today()** |
| **Vista** |
| **<group string="Información General">**  **<field name="name" />**  **<field name="zona" />**  **<field name="superficie" />**  **<field name="tipo" />**  **</group>**  **<group string="Detalles de Limpieza">**  **<div style="display: flex; align-items: center; gap: 10px; width: 400px">**  **<p style="margin: 0;">Última Limpieza</p>**  **<field name="fecha\_limpieza" style="width: 100px" />**  **<button name="marcar\_limpieza\_hoy"**  **type="object"**  **class="oe\_highlight"**  **confirm="¿Está seguro de que desea marcar la limpieza de hoy?"**  **icon="fa-check"**  **style="height: 30px; width: 30px; padding: 5px;" />**  **</div>**  **<group>**  **<field name="dias\_desde\_limpieza" readonly="1" />**  **<field name="fecha\_prox\_limpieza" readonly="1" />**  **<field name="urgencia\_limpieza" widget="progressbar" />**  **</group>**  **</group>**  **<group string="Responsable">**  **<field name="responsable\_id" />**  **</group>** |
| **Resultado** |
|  |

**13. Utiliza el percentpie para la popularidad de una especie (es posible que necesites un calculado). Al ser un percentpie un widget que no se puede manipular, añade dos botones, uno para subir 1 punto la popularidad y otro para bajarla 1 punto. El campo original popularidad que no sea visible.**

| **Modelo** |
| --- |
| **popularidad = fields.Integer(string='Popularidad', compute='\_compute\_popularidad', store=True, readonly=True, default=50)**  **@api.depends('atractivo')**  **def \_compute\_popularidad(self):**  **for record in self:**  **record.popularidad = record.atractivo \* 10 # Escala de 1-10 a 10-100**  **def action\_aumentar\_popularidad(self):**  **for record in self:**  **record.atractivo = min(record.atractivo + 1, 10) # Limita el valor a 10**  **def action\_disminuir\_popularidad(self):**  **for record in self:**  **record.atractivo = max(record.atractivo - 1, 1) # Limita el valor a 1** |
| **Vista** |
| **<field name="atractivo" invisible="1" />**  **<p>Popularidad</p>**  **<div**  **style="width: 100px; display: flex; flex-direction: row; gap: 10px; align-items: center; justify-items: center;">**  **<field name="popularidad" widget="percentpie" />**  **<button name="action\_disminuir\_popularidad"**  **type="object"**  **class="oe\_stat\_button"**  **icon="fa-minus-circle" />**  **<button name="action\_aumentar\_popularidad"**  **type="object"**  **class="oe\_stat\_button"**  **icon="fa-plus-circle" />**  **</div>** |
| **Resultado** |
|  |

**14. Añade un campo peligrosidad a la especie de tipo selection con 4 valores (del 0 al 4). Utiliza un widget priority con estrellas para mostrar el campo.**

**Tree**

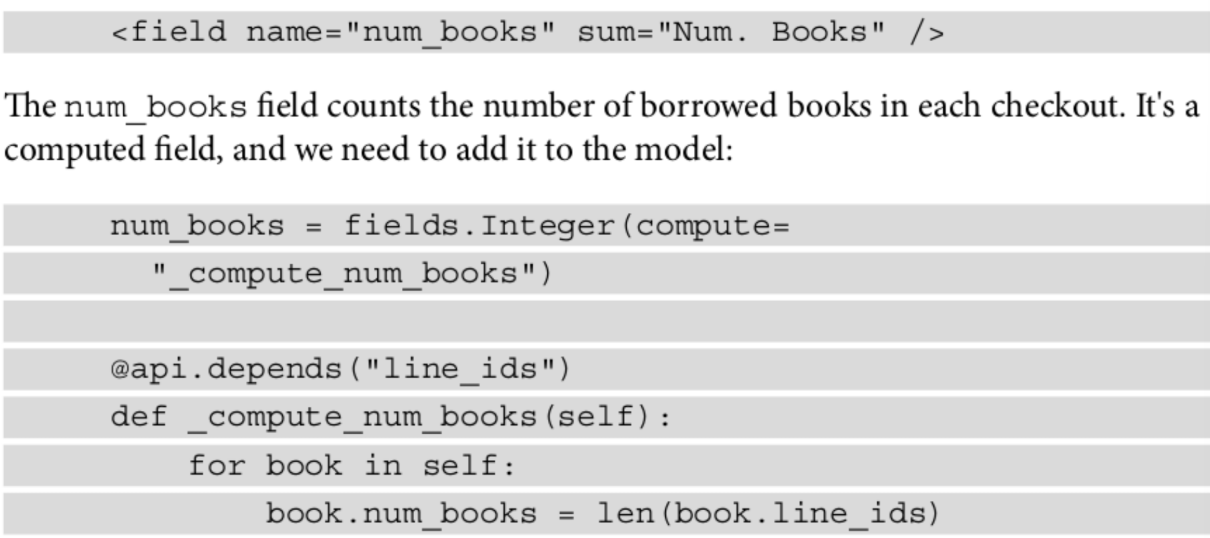
**1. Un tree para las especies.**

**1. Ordenadas por popularidad de mayor a menor y después por nombre.**

**2. Mostramos el nombre, los continentes en los que está, y la popularidad.**

**3. Añadir un campo total con el total de continentes en el que está la especie.**

**4. En rojo (danger) las especies peligrosas. En gris las especies con menos de 5 de popularidad.**

****

| **Modelo** |
| --- |
| **total\_continentes = fields.Integer(string='Total de Continentes', compute='\_compute\_total\_continentes', store=True)**  **peligroso = fields.Boolean(string='Especie peligrosa', default=False) # Nuevo campo**  **@api.depends('continente\_ids')**  **def \_compute\_total\_continentes(self):**  **for record in self:**  **record.total\_continentes = len(record.continente\_ids) # Cuenta los continentes** |
| **Vista** |
| **Tree** |
| **<record id="view\_especies\_tree" model="ir.ui.view">**  **<field name="name">zoo.especies.tree</field>**  **<field name="model">zoo.especies</field>**  **<field name="arch" type="xml">**  **<tree default\_order="popularidad desc, name asc"**  **decoration-danger="peligroso"**  **decoration-muted="popularidad &lt; 5">**  **<field name="name" />**  **<field name="continente\_ids" widget="many2many\_tags" />**  **<field name="popularidad" />**  **<field name="total\_continentes" />**  **<field name="peligroso" />**  **</tree>**  **</field>**  **</record>** |
| **Form** |
| **<field name="peligroso" />** |
| **Resultado** |
|  |

**2. Añade un tree nuevo para los cuidadores con el nombre, el telefono y los estudios.**

**1. En azul los cuidadores con estudios superiores (universidad o grado superior).**

**2. Mostramos el total de habitáculos que tiene a su cargo cada cuidador (tal vez sea útil añadir un onetomany con los habitaculos).**

| **Modelo** |
| --- |
| **habitaculos\_ids = fields.One2many(**  **'zoo.habitaculos',**  **'responsable\_id',**  **string='Habitáculos a cargo'**  **)**  **total\_habitaculos = fields.Integer(**  **string='Total Habitáculos',**  **compute='\_compute\_total\_habitaculos',**  **store=True**  **)**  **@api.depends('habitaculos\_ids')**  **def \_compute\_total\_habitaculos(self):**  **for record in self:**  **record.total\_habitaculos = len(record.habitaculos\_ids)** |
| **Vista** |
| **<record id="view\_cuidadores\_tree" model="ir.ui.view">**  **<field name="name">zoo.cuidadores.tree</field>**  **<field name="model">zoo.cuidadores</field>**  **<field name="arch" type="xml">**  **<tree decoration-info="estudios in ['ciclo\_grado\_superior', 'universitario']">**  **<field name="name" widget='badge' />**  **<field name="telefono" widget='badge' />**  **<field name="estado\_laboral\_id" />**  **<field name="estudios" />**  **<field name="total\_habitaculos" string="Habitáculos" />**  **</tree>**  **</field>**  **</record>** |
| **Resultado** |
|  |

**3. Los habitáculos los mostramos ordenados de mayor superficie a menos. Mostramos en warning aquellos a los que ya ha pasado la fecha de limpieza. Muestra el total de animales para cada habitáculo.**

| **Modelo** |
| --- |
| **total\_animales = fields.Integer(string='Total de animales', compute='\_compute\_total\_animales', store=True)**  **@api.depends('animal\_ids')**  **def \_compute\_total\_animales(self):**  **for record in self:**  **record.total\_animales = len(record.animal\_ids)**  **is\_late\_cleaning = fields.Boolean(**  **string="Limpieza Vencida",**  **compute="\_compute\_is\_late\_cleaning",**  **store=True**  **)**  **@api.depends('fecha\_prox\_limpieza')**  **def \_compute\_is\_late\_cleaning(self):**  **today = fields.Date.today()**  **for record in self:**  **record.is\_late\_cleaning = record.fecha\_prox\_limpieza and record.fecha\_prox\_limpieza < today** |
| **Vista** |
| **<record id="view\_habitaculos\_tree" model="ir.ui.view">**  **<field name="name">zoo.habitaculos.tree</field>**  **<field name="model">zoo.habitaculos</field>**  **<field name="arch" type="xml">**  **<tree default\_order="superficie desc, name asc"**  **decoration-warning="is\_late\_cleaning">**  **<field name="name" />**  **<field name="zona" />**  **<field name="tipo" />**  **<field name="superficie" />**  **<field name="responsable\_id" />**  **<field name="fecha\_prox\_limpieza" />**  **<field name="total\_animales" />**  **<field name="is\_late\_cleaning" />**  **</tree>**  **</field>**  **</record>** |
| **Resultado** |
|  |

**Search**

**1. En las especies**

**1. Se puede buscar por nombre, por continente o por las especies que tengan una popularidad mayor que la indicada,**

**2. Agrupar por continente o por peligrosidad.**

| **Vista** |
| --- |
| **<record id="view\_especies\_search" model="ir.ui.view">**  **<field name="name">zoo.especies.search</field>**  **<field name="model">zoo.especies</field>**  **<field name="arch" type="xml">**  **<search>**  **<!-- Campos de búsqueda -->**  **<field name="name" string="Nombre de la especie" />**  **<field name="continente\_ids" string="Continente de origen" />**  **<field name="popularidad" string="Popularidad" />**  **<!-- Filtro: Popularidad mayor que un valor indicado -->**  **<filter name="popularidad\_mayor\_que" string="Popularidad mayor que">**  **<field name="popularidad" operator=">" />**  **</filter>**  **<!-- Opciones de agrupación -->**  **<group expand="0" string="Agrupar por">**  **<filter name="group\_by\_continente" string="Continente"**  **context="{'group\_by': 'continente\_ids'}" />**  **<filter name="group\_by\_peligrosidad" string="Peligrosidad"**  **context="{'group\_by': 'peligroso'}" />**  **</group>**  **</search>**  **</field>**  **</record>** |
| **Resultado** |
| **Solo puede haber un Field para que funcione** |

**2. Habitáculos:**

**1. Podemos agrupar por fecha de limpieza.**

**2. Buscar por nombre y área (menor al valor indicado).**

**3. Panel lateral con los cuidadores (se muestra de cuantos habitáculos se encarga cada uno) y con los tipos de habitáculo.**

| **Vista** |
| --- |
| **Action** |
| **<record id="action\_habitaculos" model="ir.actions.act\_window">**  **<field name="name">Habitáculos</field>**  **<field name="res\_model">zoo.habitaculos</field>**  **<field name="view\_mode">kanban,tree,form</field>**  **<field name="context">{'search\_default\_group\_by\_fecha\_limpieza': 1}</field>**  **<field name="help" type="html">**  **<p class="o\_view\_nocontent\_smiling\_face">Configura los habitáculos del zoológico</p>**  **</field>**  **</record>** |
| **Search** |
| **<record id="view\_habitaculos\_search" model="ir.ui.view">**  **<field name="name">zoo.habitaculos.search</field>**  **<field name="model">zoo.habitaculos</field>**  **<field name="arch" type="xml">**  **<search>**  **<field name="name" />**  **<field name="fecha\_limpieza" />**  **<field name="superficie" string="Área menor a"**  **filter\_domain="[('superficie', '&lt;', self)]" />**  **<group expand="0" string="Agrupaciones">**  **<filter name="group\_by\_limpieza" string="Fecha de limpieza" domain="[]"**  **context="{'group\_by': 'fecha\_limpieza'}" />**  **</group>**  **</search>**  **</field>**  **</record>** |
| **Kanban** |
| **<record id="view\_habitaculos\_kanban" model="ir.ui.view">**  **<field name="name">zoo.habitaculos.kanban</field>**  **<field name="model">zoo.habitaculos</field>**  **<field name="arch" type="xml">**  **<kanban default\_group\_by="responsable\_id">**  **<field name="name" />**  **<field name="zona" />**  **<field name="superficie" />**  **<field name="tipo" />**  **<field name="fecha\_limpieza" />**  **<field name="fecha\_prox\_limpieza" />**  **<field name="responsable\_id" />**  **<field name="total\_animales" />**  **<field name="urgencia\_limpieza" />**  **<templates>**  **<t t-name="kanban-box">**  **<div class="oe\_kanban\_global\_click">**  **<div class="oe\_kanban\_content">**  **<div class="oe\_kanban\_card">**  **<strong>**  **<field name="name" />**  **</strong>**  **<div>Zona: <field name="zona" /></div>**  **<div>Superficie: <field name="superficie" /> m²</div>**  **<div>Tipo: <field name="tipo" /></div>**  **<div>Última limpieza: <field name="fecha\_limpieza" /></div>**  **<div>Próxima limpieza: <field name="fecha\_prox\_limpieza" /></div>**  **<div>Animales: <field name="total\_animales" /></div>**  **<div>Urgencia limpieza: <t**  **t-if="record.urgencia\_limpieza.raw\_value &gt;= 70">**  **<span style="color:red;">**  **<field name="urgencia\_limpieza" /> % </span>**  **</t>**  **<t**  **t-elif="record.urgencia\_limpieza.raw\_value &gt;= 40">**  **<span style="color:orange;">**  **<field name="urgencia\_limpieza" />%</span>**  **</t>**  **<t**  **t-else="">**  **<span style="color:green;">**  **<field name="urgencia\_limpieza" />%</span>**  **</t>**  **</div>**  **<div>Responsable: <field name="responsable\_id" />**  **</div>**  **</div>**  **</div>**  **</div>**  **</t>**  **</templates>**  **<sidebar>**  **<div class="o\_kanban\_sidebar">**  **<!-- Panel de cuidadores -->**  **<div class="o\_kanban\_group">**  **<h3>Cuidadores</h3>**  **<t t-foreach="widget.model.responsable\_id" t-as="cuidador">**  **<div class="o\_kanban\_record">**  **<div t-esc="cuidador.display\_name" />**  **<small t-esc="'Habitáculos: ' + cuidador.total\_habitaculos" />**  **</div>**  **</t>**  **</div>**  **<!-- Panel de tipos de habitáculo -->**  **<div class="o\_kanban\_group">**  **<h3>Tipos de Habitáculo</h3>**  **<t t-foreach="widget.model.tipo" t-as="tipo">**  **<div class="o\_kanban\_record">**  **<div t-esc="tipo.display\_name" />**  **<small t-esc="'Cantidad: ' + tipo.count" />**  **</div>**  **</t>**  **</div>**  **</div>**  **</sidebar>**  **</kanban>**  **</field>**  **</record>** |
| **Resultado** |
|  |

**Otros modelos**

**Añade tres campos en diferentes modelos que consideres interesantes. Genera las vistas (form, tree and search) para los otros modelos de zoo. Añade los elementos que consideres necesarios aplicando lo aprendido en los ejercicios y procurando que sean personales (que el parecido con otros trabajos no pase de lo razonable).**

| **Modelos** |
| --- |
| **Cuidadores** |
| **# Campo de fecha de nacimiento fecha\_nacimiento = fields.Date(string='Fecha de Nacimiento')**  **# Campo de dirección**  **direccion = fields.Char(string='Dirección')**  **# Campo de experiencia laboral**  **experiencia\_laboral = fields.Text(string='Experiencia Laboral')**  **def cambiar\_estado\_laboral(self):**  **for cuidador in self:**  **if cuidador.estado\_laboral\_id:**  **siguiente\_estado = self.env['zoo.estados.laborales'].search([**  **('orden', '=', cuidador.estado\_laboral\_id.orden + 1)**  **], limit=1)**  **if siguiente\_estado:**  **cuidador.estado\_laboral\_id = siguiente\_estado** |
| **Especies** |
| **# Nuevo campo: Nombre científico**  **nombre\_cientifico = fields.Char(string='Nombre científico', help='Nombre científico de la especie')**  **# Nuevo campo: Fecha de descubrimiento**  **fecha\_descubrimiento = fields.Date(string='Fecha de descubrimiento', help='Fecha en la que se descubrió la especie')**  **# Nuevo campo: Estado de conservación**  **estado\_conservacion = fields.Selection([**  **('en\_peligro', 'En peligro'),**  **('vulnerable', 'Vulnerable'),**  **('no\_amenazada', 'No amenazada')**  **], string='Estado de conservación', help='Estado de conservación de la especie')** |
| **Visitante** |
| **class Visitantes(models.Model):**  **\_name = 'zoo.visitantes'**  **\_description = 'Registro de visitantes del zoológico'**    **name = fields.Char(string='Nombre completo', required=True)**  **dni = fields.Char(string='DNI/Número de identificación', size=20)**  **fecha\_visita = fields.Date(string='Fecha de visita', default=lambda self: fields.Date.today())**  **tipo\_entrada = fields.Selection([**  **('general', 'Entrada General'),**  **('niño', 'Niño (3-12 años)'),**  **('senior', 'Senior (+65 años)'),**  **('discapacitado', 'Persona con discapacidad'),**  **('grupo', 'Grupo escolar')],**  **string='Tipo de entrada', required=True)**  **email = fields.Char(string='Email')**  **telefono = fields.Char(string='Teléfono')**  **origen = fields.Char(string='Ciudad/País de origen')**  **valoracion = fields.Integer(string='Valoración (1-5)', help='Valoración del visitante sobre su experiencia')**  **comentarios = fields.Text(string='Comentarios')**  **guia\_contratado = fields.Boolean(string='Contrató guía', default=False)**  **total\_gastado = fields.Float(string='Total gastado', digits=(10, 2))**    **precio\_entrada = fields.Float(string='Precio de Entrada', compute='\_compute\_precio\_entrada', store=True)**  **@api.constrains('valoracion')**  **def \_check\_valoracion(self):**  **for record in self:**  **if record.valoracion and (record.valoracion < 1 or record.valoracion > 5):**  **raise ValidationError('La valoración debe estar entre 1 y 5')**  **@api.depends('tipo\_entrada')**  **def \_compute\_precio\_entrada(self):**  **precios = {**  **'general': 15.00,**  **'niño': 8.00,**  **'senior': 10.00,**  **'discapacitado': 5.00,**  **'grupo': 6.00**  **}**  **for record in self:**  **record.precio\_entrada = precios.get(record.tipo\_entrada, 0.00)** |
| **Eventos** |
| **class Eventos(models.Model):**  **\_name = 'zoo.eventos'**  **\_description = 'Eventos especiales del zoológico'**    **name = fields.Char(string='Nombre del evento', required=True)**  **fecha\_inicio = fields.Datetime(string='Fecha y hora de inicio', required=True)**  **fecha\_fin = fields.Datetime(string='Fecha y hora de finalización', required=True)**  **descripcion = fields.Text(string='Descripción del evento')**  **ubicacion = fields.Char(string='Ubicación en el zoológico')**  **capacidad\_maxima = fields.Integer(string='Capacidad máxima de asistentes')**  **inscritos = fields.Integer(string='Número de inscritos', compute='\_compute\_inscritos')**  **estado = fields.Selection([**  **('planificado', 'Planificado'),**  **('confirmado', 'Confirmado'),**  **('cancelado', 'Cancelado'),**  **('completado', 'Completado')],**  **string='Estado', default='planificado')**  **tipo\_evento = fields.Selection([**  **('educativo', 'Educativo'),**  **('entretenimiento', 'Entretenimiento'),**  **('especial', 'Evento especial'),**  **('nocturno', 'Visita nocturna')],**  **string='Tipo de evento', required=True)**  **precio = fields.Float(string='Precio de entrada', digits=(10, 2))**  **responsable\_id = fields.Many2one('zoo.cuidadores', string='Responsable del evento')**  **imagen = fields.Binary(string='Imagen promocional')**  **requisitos = fields.Text(string='Requisitos para participar')**  **@api.depends('fecha\_inicio', 'fecha\_fin')**  **def \_compute\_duracion(self):**  **for record in self:**  **if record.fecha\_inicio and record.fecha\_fin:**  **delta = record.fecha\_fin - record.fecha\_inicio**  **record.duracion\_horas = delta.total\_seconds() / 3600.0**  **else:**  **record.duracion\_horas = 0.0**  **@api.constrains('fecha\_inicio', 'fecha\_fin')**  **def \_check\_fechas(self):**  **for record in self:**  **if record.fecha\_inicio and record.fecha\_fin and record.fecha\_fin < record.fecha\_inicio:**  **raise ValidationError('La fecha de finalización no puede ser anterior a la fecha de inicio')**  **def action\_confirmar\_evento(self):**  **self.write({'estado': 'confirmado'})**  **def action\_cancelar\_evento(self):**  **self.write({'estado': 'cancelado'})** |
| **Vista** |
| **Cuidadores** |
| **<field string="Fecha de Nacimiento" name="fecha\_nacimiento" />**  **<field string="Dirección" name="direccion" /> <field string="Experiencia Laboral" name="experiencia\_laboral" />** |
| **Especies** |
| **<field name="nombre\_cientifico" />**  **<field name="fecha\_descubrimiento" />**  **<field name="estado\_conservacion" />** |
| **Visitantes** |
| **<odoo>**  **<!-- Configuración de Visitantes -->**  **<record id="action\_visitantes" model="ir.actions.act\_window">**  **<field name="name">Visitantes</field>**  **<field name="res\_model">zoo.visitantes</field>**  **<field name="view\_mode">tree,form</field>**  **<field name="context">{'default\_fecha\_visita': context\_today()}</field>**  **<field name="help" type="html">**  **<p class="o\_view\_nocontent\_smiling\_face">**  **Gestiona el registro de visitantes del zoológico**  **</p>**  **</field>**  **</record>**  **<menuitem id="menu\_visitantes"**  **name="Visitantes"**  **parent="menu\_zoo\_config"**  **action="action\_visitantes"**  **sequence="70" />**  **<record id="view\_visitantes\_search" model="ir.ui.view">**  **<field name="name">zoo.visitantes.search</field>**  **<field name="model">zoo.visitantes</field>**  **<field name="arch" type="xml">**  **<search>**  **<field name="name" />**  **<field name="dni" />**  **<field name="fecha\_visita" filter\_domain="[('fecha\_visita', '=', self)]" />**  **<filter string="Visitas hoy" name="hoy"**  **domain="[('fecha\_visita','=',context\_today())]" />**  **<filter string="Con guía" name="con\_guia" domain="[('guia\_contratado','=',True)]" />**  **<separator />**  **<group expand="0" string="Group By">**  **<filter string="Tipo de entrada" name="group\_tipo\_entrada"**  **context="{'group\_by':'tipo\_entrada'}" />**  **<filter string="Por valoración" name="group\_valoracion"**  **context="{'group\_by':'valoracion'}" />**  **</group>**  **</search>**  **</field>**  **</record>**  **<record id="view\_visitantes\_form" model="ir.ui.view">**  **<field name="name">zoo.visitantes.form</field>**  **<field name="model">zoo.visitantes</field>**  **<field name="arch" type="xml">**  **<form>**  **<sheet>**  **<group>**  **<group>**  **<field name="name" />**  **<field name="dni" />**  **<field name="fecha\_visita" />**  **<field name="tipo\_entrada" />**  **<field name="valoracion" />**  **</group>**  **<group>**  **<field name="email" />**  **<field name="telefono" />**  **<field name="origen" />**  **<field name="guia\_contratado" />**  **<field name="total\_gastado" />**  **</group>**  **</group>**  **<notebook>**  **<page string="Comentarios">**  **<field name="comentarios" />**  **</page>**  **</notebook>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **<record id="view\_visitantes\_tree" model="ir.ui.view">**  **<field name="name">zoo.visitantes.tree</field>**  **<field name="model">zoo.visitantes</field>**  **<field name="arch" type="xml">**  **<tree>**  **<field name="name" />**  **<field name="dni" />**  **<field name="fecha\_visita" />**  **<field name="tipo\_entrada" />**  **<field name="valoracion" />**  **<field name="total\_gastado" />**  **</tree>**  **</field>**  **</record>**  **</odoo>** |
| **Eventos** |
| **<odoo>**  **<!-- Configuración de Eventos -->**  **<record id="action\_eventos" model="ir.actions.act\_window">**  **<field name="name">Eventos</field>**  **<field name="res\_model">zoo.eventos</field>**  **<field name="view\_mode">tree,form,calendar</field>**  **<field name="context">{'search\_default\_futuros': 1}</field>**  **<field name="help" type="html">**  **<p class="o\_view\_nocontent\_smiling\_face">**  **Organiza y gestiona los eventos especiales del zoológico**  **</p>**  **</field>**  **</record>**  **<menuitem id="menu\_eventos"**  **name="Eventos"**  **parent="menu\_zoo\_config"**  **action="action\_eventos"**  **sequence="80" />**  **<record id="view\_eventos\_search" model="ir.ui.view">**  **<field name="name">zoo.eventos.search</field>**  **<field name="model">zoo.eventos</field>**  **<field name="arch" type="xml">**  **<search>**  **<field name="name" />**  **<field name="tipo\_evento" />**  **<field name="responsable\_id" />**  **<filter string="Eventos futuros" name="futuros"**  **domain="[('fecha\_inicio','&gt;',context\_today())]" />**  **<filter string="Eventos confirmados" name="confirmados"**  **domain="[('estado','=','confirmado')]" />**  **<separator />**  **<group expand="0" string="Group By">**  **<filter string="Tipo de evento" name="group\_tipo\_evento"**  **context="{'group\_by':'tipo\_evento'}" />**  **<filter string="Por estado" name="group\_estado" context="{'group\_by':'estado'}" />**  **</group>**  **</search>**  **</field>**  **</record>**  **<record id="view\_eventos\_tree" model="ir.ui.view">**  **<field name="name">zoo.eventos.tree</field>**  **<field name="model">zoo.eventos</field>**  **<field name="arch" type="xml">**  **<tree>**  **<field name="name" />**  **<field name="tipo\_evento" />**  **<field name="fecha\_inicio" />**  **<field name="fecha\_fin" />**  **<field name="estado" />**  **<field name="capacidad\_maxima" />**  **<field name="precio" />**  **</tree>**  **</field>**  **</record>**  **<record id="view\_eventos\_form" model="ir.ui.view">**  **<field name="name">zoo.eventos.form</field>**  **<field name="model">zoo.eventos</field>**  **<field name="arch" type="xml">**  **<form>**  **<sheet>**  **<group>**  **<group>**  **<field name="name" />**  **<field name="tipo\_evento" />**  **<field name="fecha\_inicio" />**  **<field name="fecha\_fin" />**  **<field name="estado" />**  **<field name="responsable\_id" />**  **</group>**  **<group>**  **<field name="capacidad\_maxima" />**  **<field name="precio" />**  **<field name="ubicacion" />**  **<field name="imagen" widget="image" />**  **</group>**  **</group>**  **<notebook>**  **<page string="Descripción">**  **<field name="descripcion" />**  **</page>**  **<page string="Requisitos">**  **<field name="requisitos" />**  **</page>**  **</notebook>**  **</sheet>**  **</form>**  **</field>**  **</record>**  **</odoo>** |
| **Resultado** |
| **Cuidador** |
|  |
| **Especies** |
|  |
| **Visitantes** |
|  |
| **Eventos** |
|  |