

# Ejercicios Vistas

## Form Animales

1. En animales, cabecera donde aparezca el nombre del animal en grande y la imagen de la especie.
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.
3. Dividir los campos de Animal utilizando group para que ocupen dos columnas. Utiliza algún separador.
4. Los campos relacionados con el cuidador los ponemos en una pestaña (notebook)
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).
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.
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.

```
@api.model
```

```
def _default_stage_id(self):
```

```
    Stage = self.env["library.checkout.stage"]
```

```
    return Stage.search([("state", "=", "new")],  
                        limit=1)
```

```
stage_id = fields.Many2one(
```

```
    "library.checkout.stage",
```

```
    default=_default_stage_id,
```

8. Añadir campo a los cuidadores con los estudios. Pueden ser Eso, Bachillerato, Ciclo Grado Medio, Ciclo Grado Superior o Universitarios. Utilizar un radiobox.
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.
10. En cuidador personaliza las etiquetas de los campos para que aparezca tu nombre (Ej: Estudios Roberto). CAMPO STRING EN FIELD
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%.
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>, y consultar los que hay disponibles en el fichero web/static/lib/fontawesome/css/fontawesome.css (depende de la instalación puede cambiar donde esté).
  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.
  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.

```
<field name="num_books" sum="Num. Books" />
```

The num\_books field counts the number of borrowed books in each checkout. It's a computed field, and we need to add it to the model:

```
num_books = fields.Integer(compute=
    "_compute_num_books")

@api.depends("line_ids")
def _compute_num_books(self):
    for book in self:
        book.num_books = len(book.line_ids)
```

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).
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.

## 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.
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.

## 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).