

# Odoo / React

Yassir Curbelo Haldi 1-DAM-T

# Índice

<i>Procedimiento para crear el Endpoint.....</i>	<i><b>1</b></i>
<i>Capturas de funcionamiento.....</i>	<i><b>3</b></i>

## Procedimiento para crear el Endpoint

El procedimiento usado para crear un endpoint al que poder atacar el módulo de odoo de 'project' fue primero heredar todos los elementos necesarios para poder crear un controlador en Python usando el ORM de odoo.

Los elementos heredados son los siguientes.

- project.project: para manipular los proyectos y poder asignarles etapas predefinidas o tareas.
- project.task: para poder crear tareas, asignarles estados a estas.
- project.task.type: para crear las etapas en los proyectos y añadirlas a odoo.

Tras esta tarea añadí los elementos autogenerados para que al crear un proyecto se añadan las tareas necesarias junto a sus estados y etapas, los estados de las tareas 'kanban\_state' fue creado desde el módulo heredado project.task añadiendo dos y modificando tres elementos existentes, las etapas de proyecto fueron añadidas con una carga previa de datos, este se puede ver en el archivo 'project\_task\_type.xml', por ultimo las tareas fueron añadidas de la misma forma que los estados del Kanban lo que directamente al módulo project.project.

Lo siguiente y ultimo para finalizar el desarrollo del backend fue crear un controlador web en el que tener las funciones básicas de un CRUD para Project.task.

En la siguiente enumeración se pueden ver los métodos creados y su función.

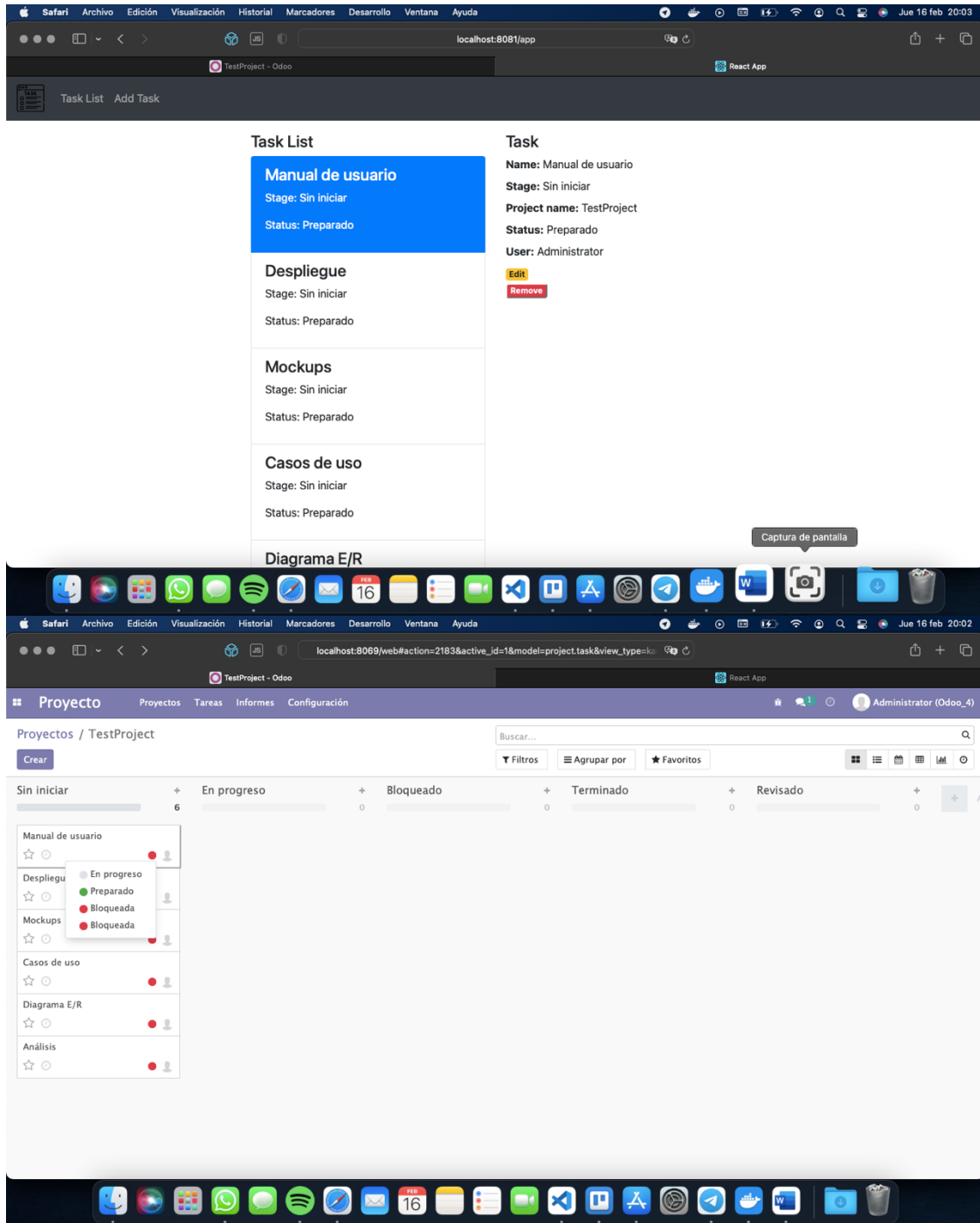
- List: retorna un listado de tareas con los siguientes datos
  - o Id de tarea
  - o Nombre de tarea
  - o Nombre de proyecto al que está asociada
  - o Nombre de usuario al que está asociada
  - o Id de la etapa en la que se encuentra la tarea
  - o Nombre de la etapa en la que se encuentra la tarea
  - o Nombre del estado en el que se encuentra la tarea
  - o Descripción de la tarea
- ListOne
  - o Id de tarea
  - o Nombre de tarea
  - o Nombre de proyecto al que está asociada
  - o Nombre de usuario al que está asociada
  - o Id de la etapa en la que se encuentra la tarea
  - o Nombre de la etapa en la que se encuentra la tarea
  - o Nombre del estado en el que se encuentra la tarea
  - o Descripción de la tarea
- Create
  - o Este método usa el modelo de projec.task así que se le puede pasar cualquier elemento siempre que cuadre con el modelo de la base de datos de odoo

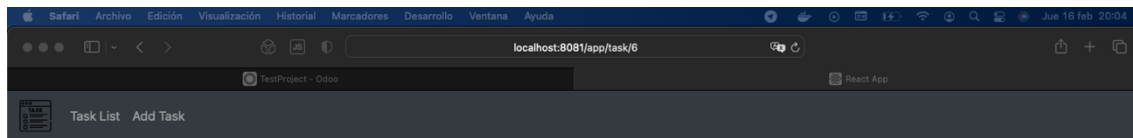
- Update
  - Este método usa el modelo Project.task de referencia solo que tiene dos elementos que se asignan con cuidado para no crear errores.
    - Stage\_id: variable que asigna en que etapa se crea la tarea
    - Kanban\_state: variable que asigna el estado de la tarea al ser creada
- Remove
  - Este método recibe en la URL un valor int que será el id de la task que se desea eliminar.
- deleteAll
  - Aunque en desuso recorre todos los elementos de project.task para eliminarlos uno a uno
- listProject
  - Este método es usado para obtener todos los stages disponibles en project.task.type devolviendo su id y su nombre para posteriormente usarse en un seleccionable del frontend a base de maestro de datos.

Con esto conseguimos que al crear una tarea nueva se creen los estados, etapas y tareas básicas para un proyecto además de brindar la posibilidad de ser manipulado externamente a odoo.

NOTA: Es necesario activar el cors en el servidor de odoo al igual que el endpoint para la sesión, todo esto usando el módulo auth\_session\_info.

# Capturas de funcionamiento





### Task

Name

Manual de usuario

Stage

Selecciona una opción: Revisado

Project Name

TestProject

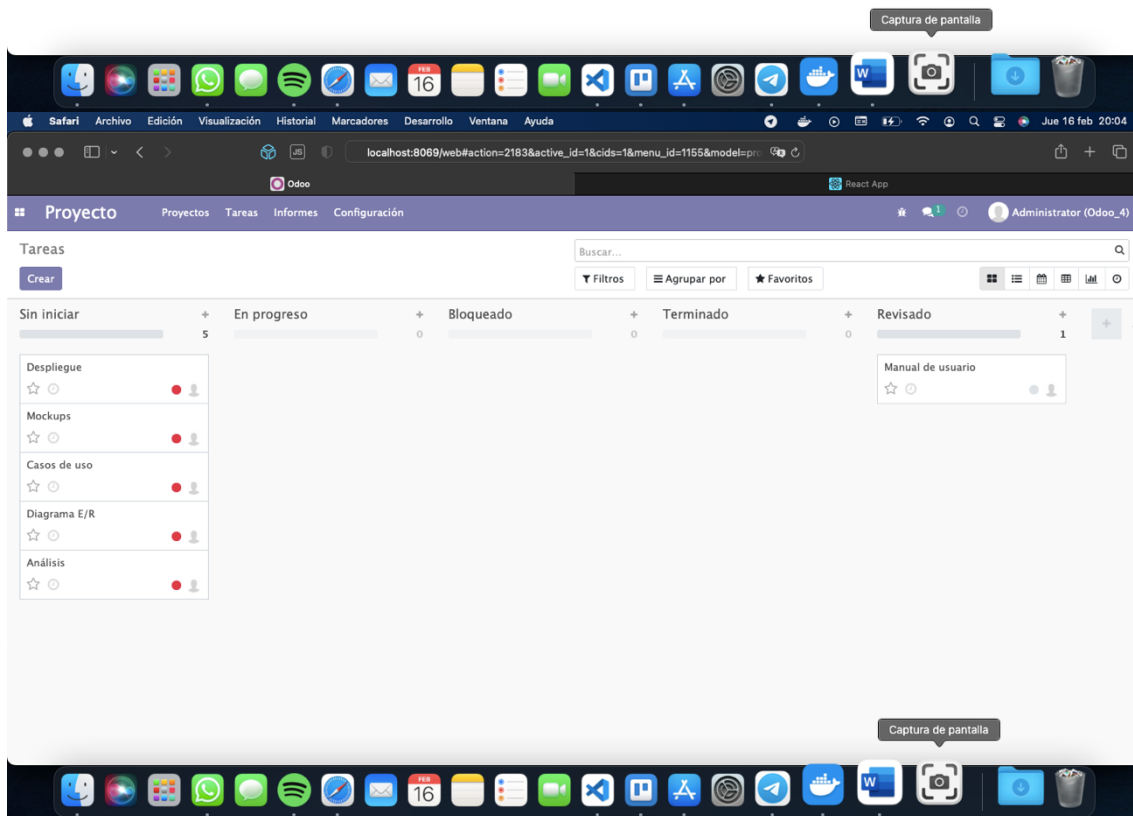
Status

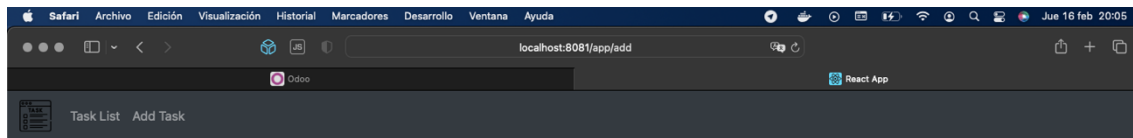
Selecciona una opción: En progreso

User

Administrator

[Delete](#) [Update](#)





## Task

Name

Stage

Selecciona una opción: Sin iniciar

Project Name

Status

Selecciona una opción: Bloqueada

User

Create

