

APLICACIÓN ToDoLIST

GRUPO-13

Pedro López Mellado

Jose Alberola Torres



Metodología

Product owner

- Dar valor

GitHub flow

- Rama principal
- Ramas de características
- Pull Request

Integración continua

- GitHub actions

Pair Programming

- 1 Historia de usuario
- Equipos con tareas



Flujo de trabajo



Nuevas funcionalidades

01

PERFIL

Modificación de datos y
transferencia de admin

02

TAREAS

Buscadores, fecha de
vencimiento y estado

03

TABLEROS

Clasificaciones, interfaz,
tareas y CRUD

04

EQUIPOS

Pertenencia al equipo,
tareas y gestión

Editar perfil

ToDoList Tareas Equipos Usuario Ejemplo ▾

Información de la cuenta

Email

Nombre

Contraseña anterior

Contraseña nueva

Fecha de nacimiento

Administrador

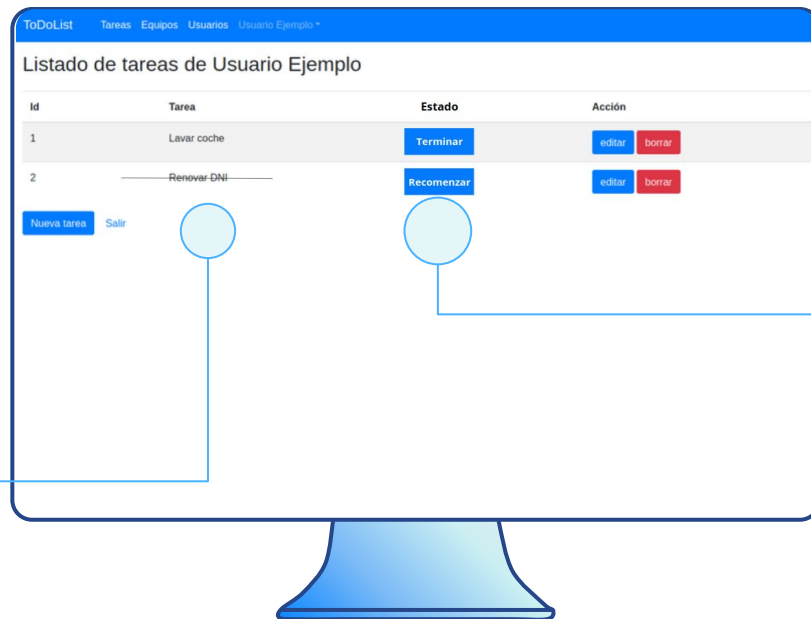
Datos personales

Actualización de valores

Administrador

Posibilidad de transferir el rol

Estado tareas



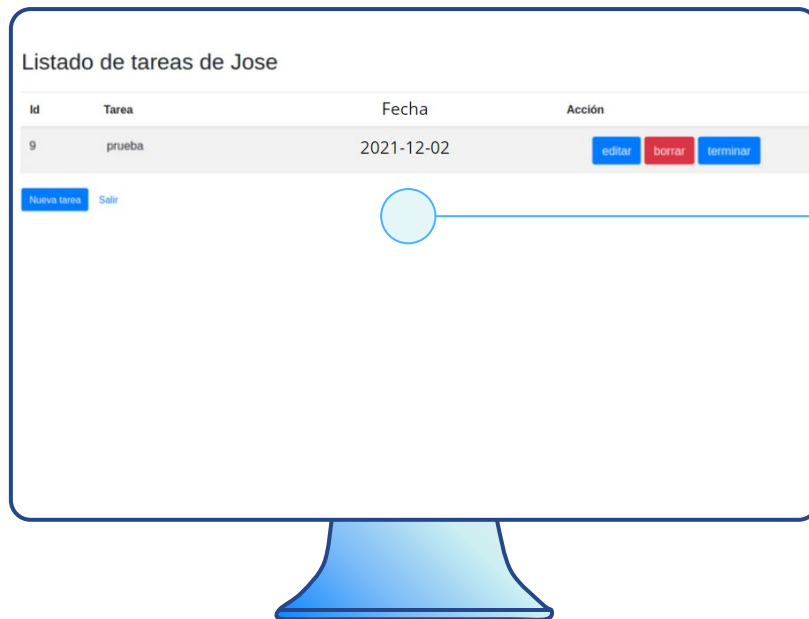
Tachado

Las tareas terminadas aparecen tachadas

Estado

Las tareas puede estar terminadas o no

Tareas con fecha



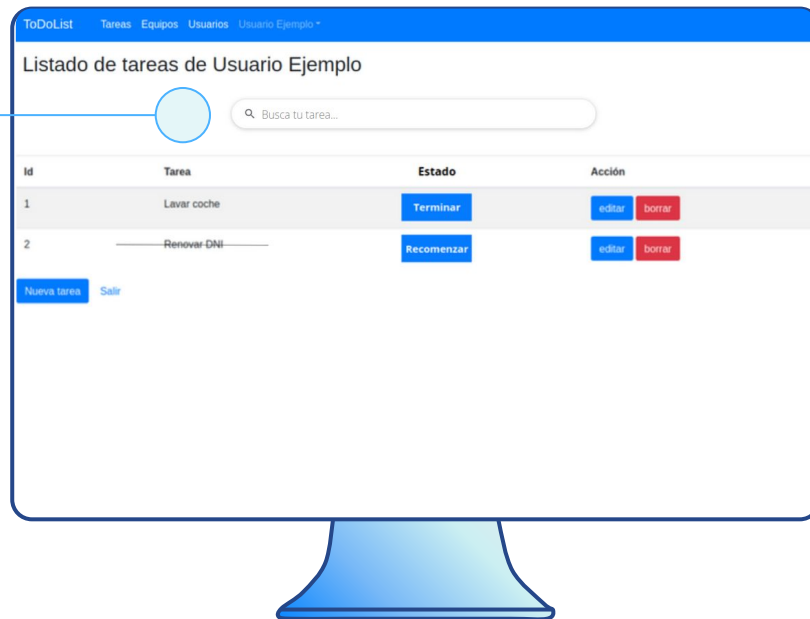
Fecha

Valor opcional para marcar
fecha de vencimiento

Buscadores de tareas

Buscador

Búsqueda por título dentro del marco de trabajo



Tableros

Añadir tablero

Nos abrirá el formulario de creación de tableros

Eliminar tablero

Enviaré las tareas a Bandeja de entrada



Creación de tareas

Se crearán las tareas en el tablero actual

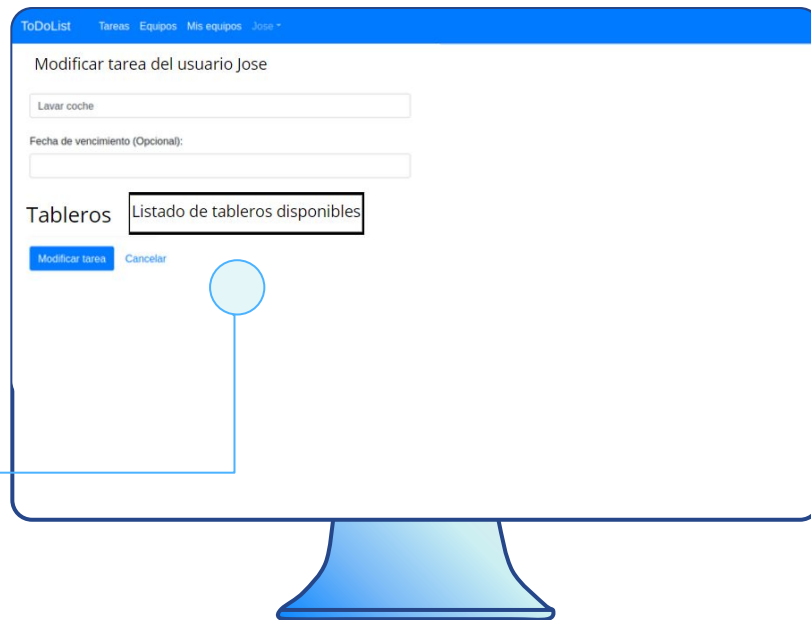
Contador de tareas

Contador

Dependiendo de la acción se incrementará o decrementará



Mover tareas de tablero



Listado de tableros

Aparecerán los tableros del usuario

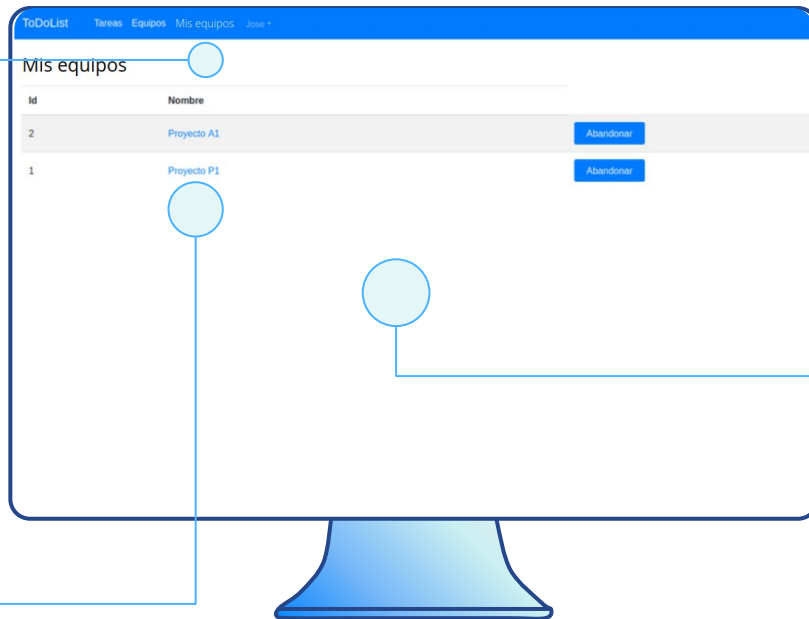
Interfaz 'Mis equipos'

Mis equipos

Nueva opción del menú superior

Detalles equipo

Vista con información acerca del equipo



Listado

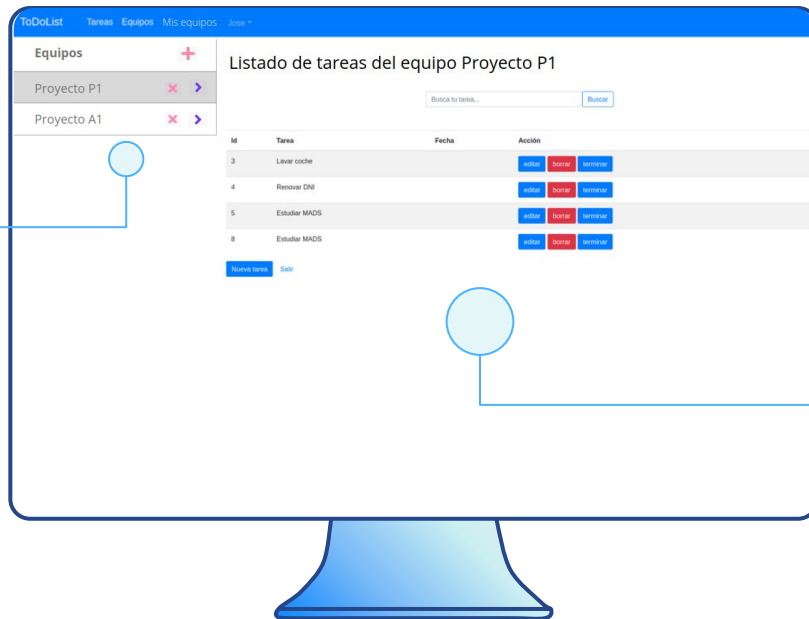
Equipos en los que el usuario es miembro

Equipos con tareas

Mis equipos

Listado de equipos en los que el usuario es miembro

- + Creación de equipo
- + Abandonar equipo
- + Información equipo



Tareas

Listado de tareas del equipo seleccionado

Puesta en producción

1

Backup datos

Realizar una copia de seguridad de los datos

2

Schema actual

Generar el schema de la BBDD actual

3

Schema nuevo

Generar el schema de la BBDD nueva

4

Migración

Construir el fichero de migración y ejecutarlo

5

Recuperar datos

Recuperar los datos antiguos de la BBDD

6

Ejecutar app

Ejecutar la nueva versión de nuestra aplicación

Migración (I)

```
43 110,111d74
44 < fecha date,
45 < terminada boolean,
46 113,114d75
47 < equipo_id bigint,
48 < tablero_id bigint,
```

```
108 CREATE TABLE public.tareas (
109     id bigint NOT NULL,
110     fecha date,
111     terminada boolean,
112     titulo character varying(255) NOT NULL,
113     equipo_id bigint,
114     tablero_id bigint,
115     usuario_id bigint NOT NULL
116 );
```

```
73 CREATE TABLE public.tareas (
74     id bigint NOT NULL,
75     titulo character varying(255) NOT NULL,
76     usuario_id bigint NOT NULL
77 );
```

diff-1.4.0-1.3.0

Diferencias entre ficheros
de schemas 1.4.0 y 1.3.0

schema-1.4.0.sql

Schema versión 1.4.0

schema-1.3.0.sql

Schema versión 1.3.0

Migración(II)

Creación del fichero de migración.
Ejecutaremos este script para actualizar la
base de datos.

```
20
21 ALTER TABLE public.tareas
22 ADD COLUMN fecha date,
23 ADD COLUMN terminada boolean,
24 ADD COLUMN equipo_id bigint,
25 ADD COLUMN tablero_id bigint;
26
27 ALTER TABLE ONLY public.tareas
28 ADD CONSTRAINT fk7a7bbe0arnbqnrekrryoab7je FOREIGN KEY (equipo_id) REFERENCES public.equipos(id) ON DELETE CASCADE;
29
30 ALTER TABLE ONLY public.tareas
31 ADD CONSTRAINT fksqj5mtew63jjlb55n4yfxqq0 FOREIGN KEY (tablero_id) REFERENCES public.tableros(id);
```




GRACIAS
