

Tarea 3

Diseñada para practicar Express, Middlewares y base de datos.

Requerimientos

- [Exponer api para los animales](#)
- [Interfaz gráfica](#)
- [Base de datos](#)

Herramientas sugeridas

- Visual Studio Code
- DBeaver
- Postman
- PostgreSQL
- [node-postgres](#)

Exponer api para los animales

Crear las siguientes rutas (con manejo de excepciones):

Request	Ruta	Body	Esperado
GET	/animals	NA	Arreglo de animales con algunos atributos [{id(id), name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour) owner (fullname from user)}...]
POST	/animals	{name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour)} Validación: requerido: name, age name: string(5) breed: string, age:string, color: string	Animal creado {id(id), name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour)}
PUT	/animals/:id	{name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour)} Validación: requerido: name, age name: string(5) breed: string, age:string, color: string	Animal actualizado {id(id), name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour)}

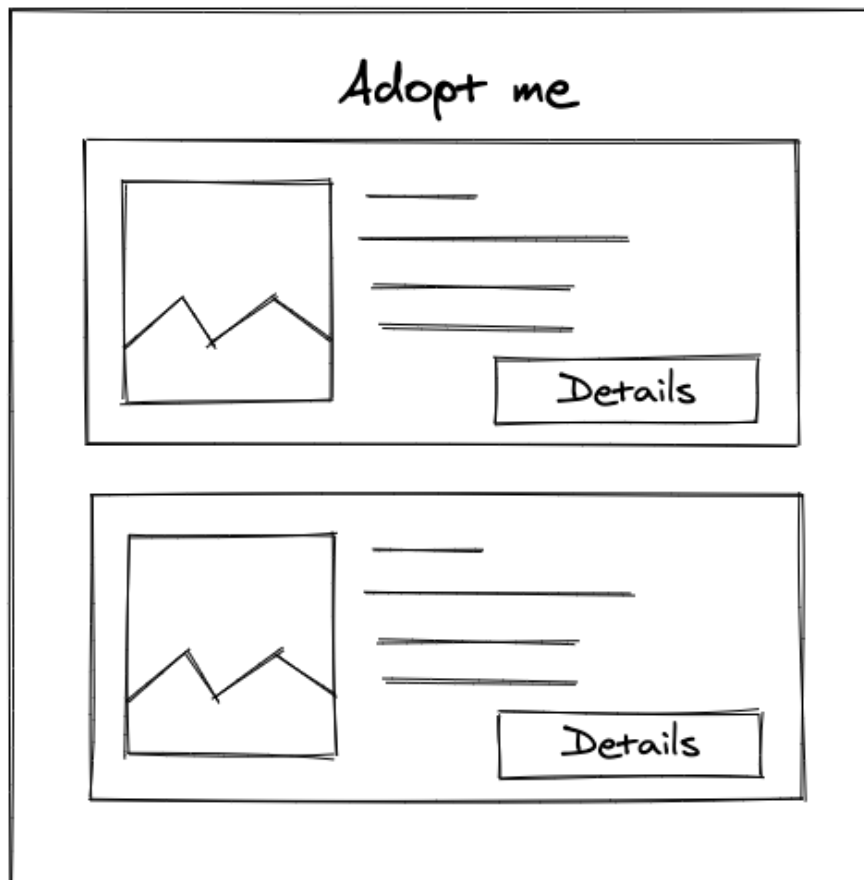
DELETE	/animals/:id	NA	Animal borrado {id(id), name(animalsname), breed (breedname), specie (speciesname), age (animalsage), color (basecolour), owner (fullname from user)*} * Si existe owner, mostrar nombre completo, si no, mostrar string vacío
--------	--------------	----	--

Request	Ruta	Body	Esperado
GET	/users	NA	Arreglo de usuarios con atributos [{id(id), name(fullname), age (age)}...]
POST	/users	{name(fullname), age (age)} Validación: requerido: name, age name: string(5) age:number > 0	Usuario creado {id(id), name(fullname), age (age)}
PUT	/users/:id	{name(fullname), age (age)} Validación: requerido: name, age name: string(5) age:number > 0	Usuario actualizado {id(id), name(fullname), age (age)}
DELETE	/users/:id	NA	Usuario borrado {id(id), name(fullname), age (age)} También se tiene que borrar de la tabla de Animals, es decir, poner como null la columna de owner

En caso de ir a una ruta no encontrada retornar 404 Not Found. (Utilizando middlewares)
En caso de no tener la información correcta para POST, PUT, DELETE retornar 400 Bad request. (Utilizando middlewares)

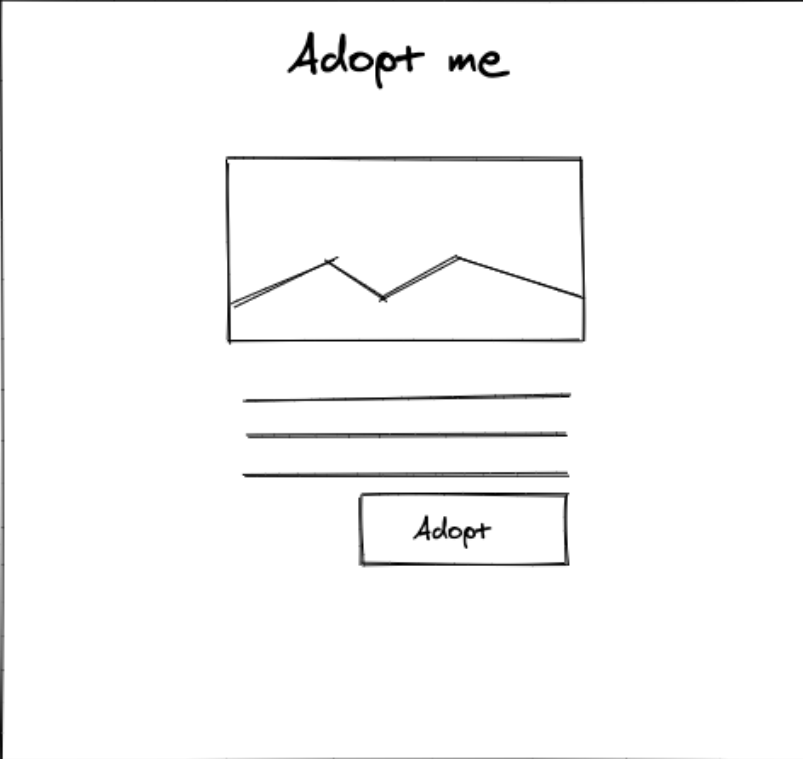
Interfaz gráfica

Pantalla principal



Mostrar todos los detalles mínimos utilizados en GET /animals

Pantalla de detalle (Se muestra al hacer clic en **Details**)



Adopt me

[Placeholder image of a mountain range]

Adopt

Mostrar todos los detalles mínimos de GET /animals

Pantalla de adopción (Se muestra al dar clic en **Adopt**)



Adopt me

You're adopting <Animal's name>
please write your user's id so we can
register you on our database.
Thank you for adopting!

id [input field]

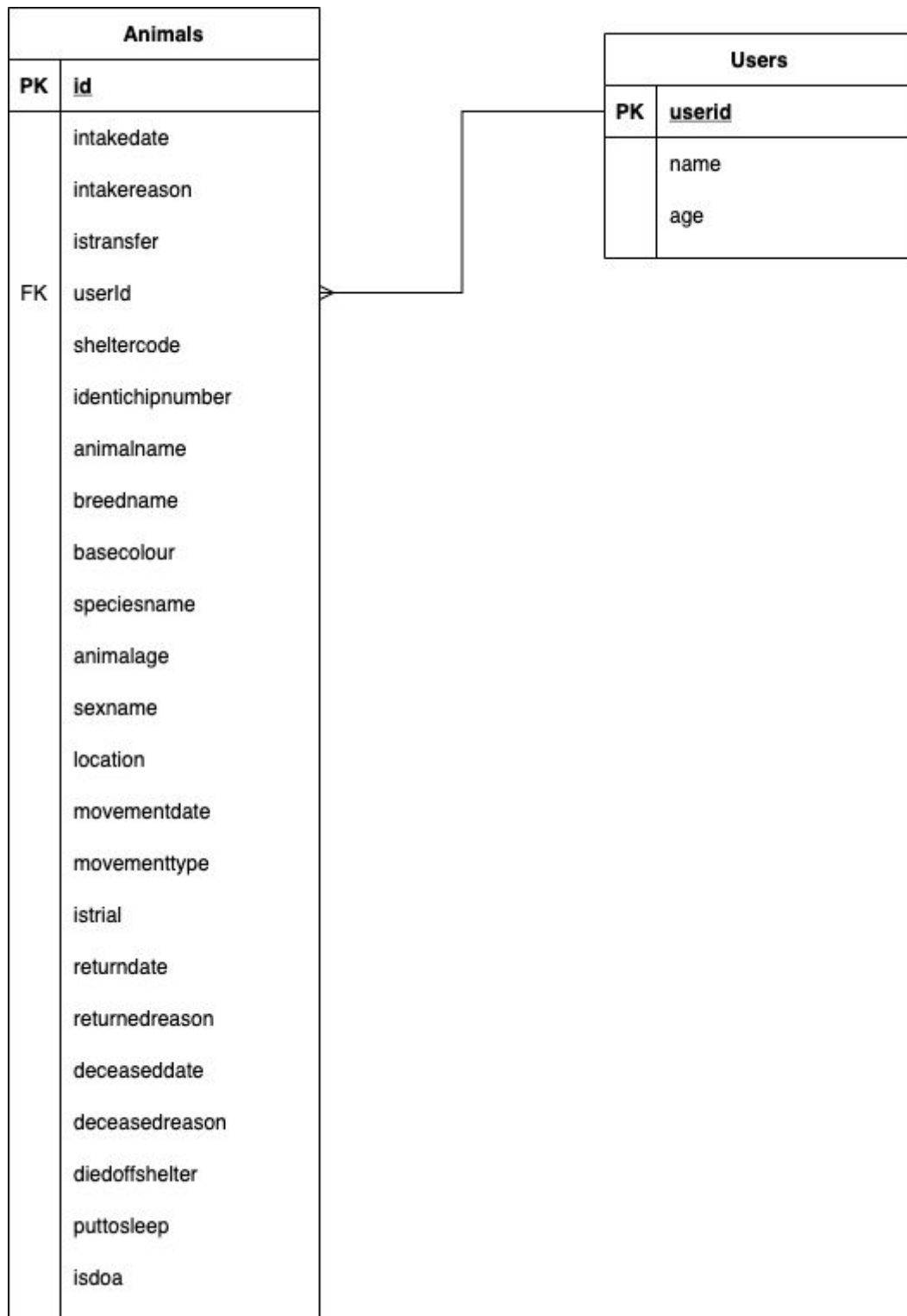
Adopt

Al hacer clic en Adopt te debe de llevar a la Pantalla principal

Base de datos

Puedes utilizar la base de datos que gustes siempre y cuando sea relacional.

- Base de datos con animals y users
- Animals tiene agregada la columna de owner como foreign key de users



Entregables

- Url de repositorio de github
- Base de datos utilizada