

Table of contents

TaskLynx / Nest Hotel – Docs	2
API – Index	4
Trabajo	8
Trabajador	38
Trabajador - Views	67
Tutorial	77
Business – Index	87
Task Management	88
Employee Management	99
Report Generation	106
Conclusion	110
Mobile – Index	111
Nest Hotel – Index	112
Despliegue de Gestión Hotelera	113
Despliegue de Hotel Nest	118
Resumen Tests	126

TaskLynx / Nest Hotel – Docs



TaskLynx-logo-oval.svg

Introduction

TaskLynx is an ecosystem composed of TaskLynx Api, TaskLynx Business and TaskLynx Mobile. TaskLynx Api (<https://github.com/DanielAlmazan/TaskLynx-SpringBoot>) is a REST API that allows communication between TaskLynx Business (<https://github.com/DanielAlmazan/TaskLynx-JavaFX>) and TaskLynx Mobile (<https://github.com/DanielAlmazan/TaskLynx-Mobile>).

Nest Hotel (<https://github.com/DanielAlmazan/hotel-nest>) is a REST API that allows communication with TaskLynx Business (<https://github.com/DanielAlmazan/TaskLynx-JavaFX>) in order to manage the cleaning of hotel rooms.

TaskLynx Mobile (<https://github.com/DanielAlmazan/TaskLynx-Mobile>) is a mobile application that allows hotel staff to manage the cleaning of hotel rooms. It communicates with TaskLynx Api (<https://github.com/DanielAlmazan/TaskLynx-SpringBoot>) to get the tasks to be done. "Authentication" is done by using the endpoint `/api/trabajadores/{idTrabajador}/{contraseña}`. Didn't use JWT because it was not required.

TaskLynx Business (<https://github.com/DanielAlmazan/TaskLynx-JavaFX>) is a desktop application that allows hotel managers to manage the cleaning of hotel rooms and other

tasks. It communicates with TaskLynx Api (<https://github.com/DanielAlmazan/TaskLynx-SpringBoot>) to get the tasks to be done and with Nest Hotel (<https://github.com/DanielAlmazan/hotel-nest>) to manage the rooms.

API – Index



TaskLynx-API-oval-logo.svg

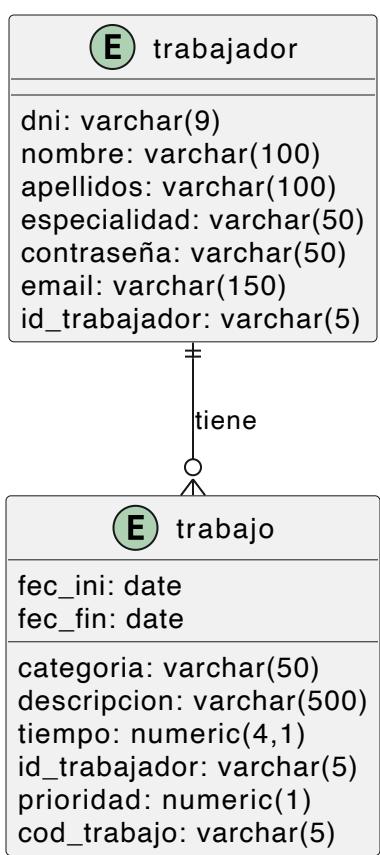
Introduction

TaskLynx API (<https://github.com/DanielAlmazan/TaskLynx-SpringBoot>) is a REST API that allows communication between TaskLynx Business (<https://github.com/DanielAlmazan/TaskLynx-JavaFX>) and TaskLynx Mobile (<https://github.com/DanielAlmazan/TaskLynx-Mobile>).

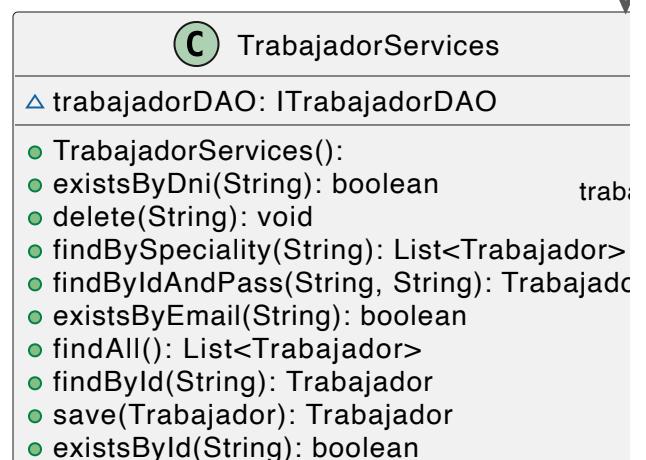
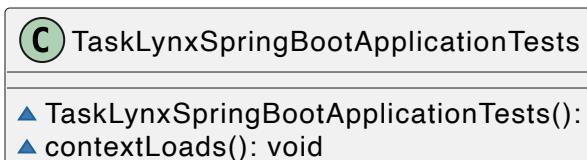
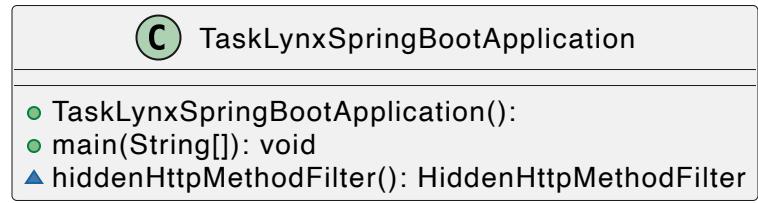
Technologies

- Java 21
- Spring Boot 2.5.4
- Apache Maven 3.9.6
- PostgreSQL

Entity-Relationship Diagram

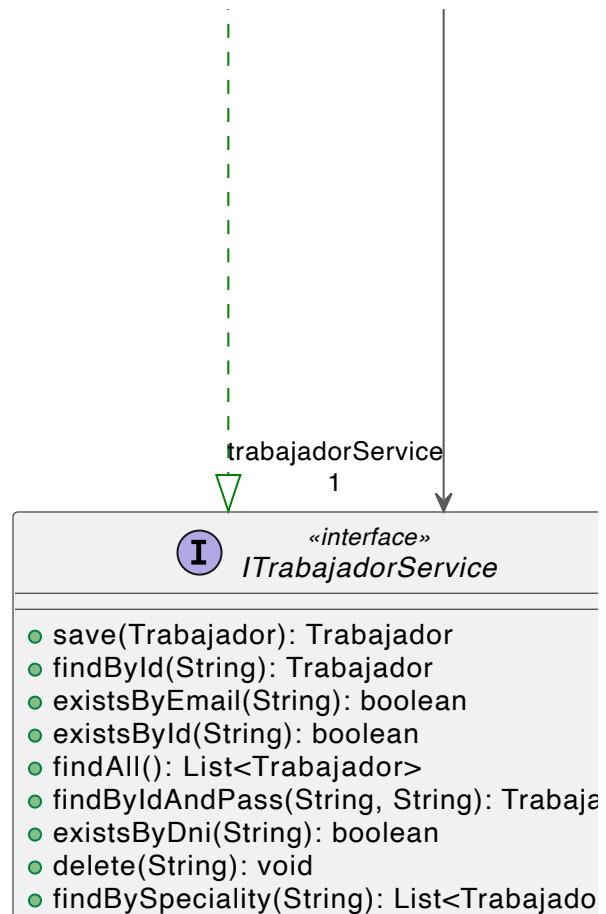


Class Diagram



trabajadorDAO

trabajadorServices



Authors

TRABAJADOR CRUD:

Aitor Moreno Iborra (<https://github.com/LtVish>)

TRABAJO CRUD:

Miguel Collado (<https://github.com/MiguelColl>)

API Development:

Daniel Enrique Almazán Sellés (<https://github.com/DanielAlmazan>)

Trabajo



Trabajo

- categoria: String
- idTrabajador: Trabajador
- fecIni: LocalDate
- fecFin: LocalDate
- descripcion: String
- codTrabajo: String
- tiempo: BigDecimal
- prioridad: BigDecimal

- Trabajo():
- getDescripcion(): String
- getTiempo(): BigDecimal
- getIdTrabajador(): Trabajador
- setFecFin(LocalDate): void
- getCodTrabajo(): String
- setDescripcion(String): void
- toString(): String
- setFecIni(LocalDate): void
- setCategoría(String): void
- setCodTrabajo(String): void
- setIdTrabajador(Trabajador): void
- getFecIni(): LocalDate
- setPrioridad(BigDecimal): void
- setTiempo(BigDecimal): void
- getPrioridad(): BigDecimal
- getCategoría(): String
- getFecFin(): LocalDate

BASE: /api/trabajos

GET BASE



Returns all the tasks in the database. If there are no tasks, it will return an empty list.

Successful response

```
{  
  "result": [  
    {
```

```
"codTrabajo": "J001",
"categoria": "Mecánica",
"descripcion": "Reparación de tuberías",
"fecIni": "2024-01-01",
"fecFin": null,
"tiempo": null,
"idTrabajador": null,
"prioridad": 2
},
{
  "codTrabajo": "J002",
  "categoria": "Fontanería",
  "descripcion": "Reparación de tuberías",
  "fecIni": "2022-01-02",
  "fecFin": null,
  "tiempo": 5.0,
  "idTrabajador": {
    "idTrabajador": "T002",
    "dni": "23456789B",
    "nombre": "Ana",
    "apellidos": "Gómez",
    "especialidad": "Fontanería",
    "contraseña": "contraseña2",
    "email": "ana.gomez@example.com"
  },
  "prioridad": 2
},
{
  "codTrabajo": "J003",
  "categoria": "Electricidad",
  "descripcion": "Instalación de luces",
  "fecIni": "2022-01-03",
  "fecFin": null,
  "tiempo": 8.0,
  "idTrabajador": {
    "idTrabajador": "T003",
    "dni": "34567890C",
    "nombre": "Carlos",
    "apellidos": "Pérez"
  }
}
```

```
        "apellidos": "Martínez",
        "especialidad": "Electricidad",
        "contraseña": "contraseña3",
        "email": "carlos.martinez@example.com"
    },
    "prioridad": 3
}
],
"error": false
}
```

Empty list response

```
{
  "result": [],
  "error": false
}
```

GET BASE/:id

⚠ Returns a task by its ID. If the task does not exist, it will return an error message.

Successful response

⚠ /api/trabajos/J001

```
{
  "result": {
    "codTrabajo": "J001",
    "categoria": "Mecánica",
    "descripcion": "Reparación de tuberías",
    "fecIni": "2024-01-01",
    "fecFin": null,
  }
}
```

```
        "tiempo": null,  
        "idTrabajador": null,  
        "prioridad": 2  
    },  
    "error": false  
}
```

Non-existent task response

⚠ /api/trabajos/J025

```
{  
    "errorMessage": "El trabajo con ID 'J025' no existe en la  
    base de datos",  
    "error": true  
}
```

POST BASE

⚠ Create a new task in the database and returns the task created. If there are any errors, it will return an error message with the list of errors. If the worker already exists, it will return an error message.

All fields are required except:

- fecFin: task completion date
- tiempo: duration of the task
- idTrabajador: worker assigned

Successful body

```
{  
    "codTrabajo": "J007",  
    "categoria": "Pintura",  
    "descripcion": "Pintar las paredes",  
    "fecIni": "2024-01-09",  
    "prioridad": 4  
}
```

Response

```
{  
    "result": {  
        "codTrabajo": "J007",  
        "categoria": "Pintura",  
        "descripcion": "Pintar las paredes",  
        "fecIni": "2024-01-09",  
        "fecFin": null,  
        "tiempo": null,  
        "idTrabajador": null,  
        "prioridad": 4  
    },  
    "error": false  
}
```

Other successful body

```
{  
    "codTrabajo": "J008",  
    "categoria": "Pintura",  
    "descripcion": "Pintar las paredes",  
    "fecIni": "2024-01-09",  
    "fecFin": "2024-01-11",  
    "tiempo": 8.5,
```

```
"idTrabajador": {  
    "idTrabajador": "T001",  
    "dni": "12345678A",  
    "nombre": "Pepe",  
    "apellidos": "Gomez",  
    "especialidad": "Fontanería",  
    "contraseña": "123",  
    "email": "wefwefesadf.com"  
},  
"prioridad": 4  
}
```

Response

```
{  
    "result": {  
        "codTrabajo": "J008",  
        "categoria": "Pintura",  
        "descripcion": "Pintar las paredes",  
        "fecIni": "2024-01-09",  
        "fecFin": "2024-01-11",  
        "tiempo": 8.5,  
        "idTrabajador": {  
            "idTrabajador": "T001",  
            "dni": "12345678A",  
            "nombre": "Pepe",  
            "apellidos": "García",  
            "especialidad": "Fontaneria",  
            "contraseña": "123",  
            "email": "wefwefewf@ergerg.com"  
        },  
        "prioridad": 4  
    },  
    "error": false  
}
```

Error body

```
{  
  "codTrabajo": "J009",  
  "fecIni": "2024-01-09",  
  "prioridad": 4  
}
```

Response

```
{  
  "errorsList": [  
    "La categoría no puede estar vacía",  
    "La descripción no puede estar vacía"  
,  
  "error": true  
}
```

Other error body

```
{  
  "codTrabajo": "J008",  
  "categoria": "Pintura",  
  "descripcion": "Pintar las paredes",  
  "fecIni": "2024-01-09",  
  "prioridad": 4  
}
```

Response

```
{  
  "errorsList": [  
]
```

```
        "El trabajo con id 'J008' ya existe en la base de datos"
    ],
    "error": true
}
```

PUT BASE/:id

- ⚠ Update a task by its ID and returns the task updated. If the worker does not exist, it will return an error message. If any of the fields are empty or invalid, it will return an error message with the list of errors.

Successful body

⚠ /api/trabajos/J007
We add an assigned worker

```
{
  "codTrabajo": "J007",
  "categoria": "Pintura",
  "descripcion": "Pintar las paredes",
  "fecIni": "2024-01-09",
  "prioridad": 4,
  "idTrabajador": {
    "idTrabajador": "T001",
    "dni": "12345678A",
    "nombre": "Pepe",
    "apellidos": "García",
    "especialidad": "Fontaneria",
    "contraseña": "123",
    "email": "wefwefewf@ergerg.com"
  }
}
```

Response

```
{  
  "result": {  
    "codTrabajo": "J007",  
    "categoria": "Pintura",  
    "descripcion": "Pintar las paredes",  
    "fecIni": "2024-01-09",  
    "fecFin": null,  
    "tiempo": null,  
    "idTrabajador": {  
      "idTrabajador": "T001",  
      "dni": "12345678A",  
      "nombre": "Pepe",  
      "apellidos": "García",  
      "especialidad": "Fontanería",  
      "contraseña": "123",  
      "email": "wefwefewf@ergerg.com"  
    },  
    "prioridad": 4  
  },  
  "error": false  
}
```

Error body

⚠ /api/trabajos/J020

```
{  
  "codTrabajo": "J020",  
  "categoria": "Pintura",  
  "descripcion": "Pintar las paredes",  
  "fecIni": "2024-01-09",  
  "fecFin": null,  
  "prioridad": 4  
}
```

```
    "prioridad": 4  
}
```

Response

```
{  
  "errorsList": [  
    "No se pudo editar, el trabajo con id 'J020' no existe en  
    la base de datos"  
,  
  "error": true  
}
```

DELETE BASE/:id

⚠ Delete a task by its ID. Returns a response without errors. If the task does not exist, it will return an error message.

Successful response

⚠ /api/trabajos/J008

```
{  
  "error": false  
}
```

Non-existent task response

⚠ /api/trabajos/J020

```
{  
  "errorMessage": "El trabajo con id 'J020' no existe en la  
base de datos",  
  "error": true  
}
```

GET BASE/pendientes

⚠ Returns all the tasks that are pending. If there are no pending tasks, it will return an empty list.

200 – Results

```
{  
  "result": [  
    {  
      "codTrabajo": "J001",  
      "categoria": "Carpintería",  
      "descripcion": "Construcción de armario",  
      "fecIni": "2022-01-01",  
      "fecFin": null,  
      "tiempo": 10.0,  
      "idTrabajador": {  
        "idTrabajador": "T001",  
        "dni": "12345678A",  
        "nombre": "Juan",  
        "apellidos": "Pérez",  
        "especialidad": "Carpintería",  
        "contraseña": "contraseña1",  
        "email": "juan.perez@example.com"  
      },  
      "prioridad": 1  
    },  
    {
```

```
"codTrabajo": "J002",
"categoria": "Fontanería",
"descripcion": "Reparación de tuberías",
"fecIni": "2022-01-02",
"fecFin": null,
"tiempo": 5.0,
"idTrabajador": {
    "idTrabajador": "T002",
    "dni": "23456789B",
    "nombre": "Ana",
    "apellidos": "Gómez",
    "especialidad": "Fontanería",
    "contraseña": "contraseña2",
    "email": "ana.gomez@example.com"
},
"prioridad": 2
},
{
    "codTrabajo": "J003",
    "categoria": "Electricidad",
    "descripcion": "Instalación de luces",
    "fecIni": "2022-01-03",
    "fecFin": null,
    "tiempo": 8.0,
    "idTrabajador": {
        "idTrabajador": "T003",
        "dni": "34567890C",
        "nombre": "Carlos",
        "apellidos": "Martínez",
        "especialidad": "Electricidad",
        "contraseña": "contraseña3",
        "email": "carlos.martinez@example.com"
    },
    "prioridad": 3
},
{
    "codTrabajo": "J800",
    "categoria": "Espacio-Tiempo",
```

```
        "descripcion": "Visitar a Sarah Connor",
        "fecIni": "2029-10-20",
        "fecFin": null,
        "tiempo": null,
        "idTrabajador": null,
        "prioridad": 1
    },
    {
        "codTrabajo": "J004",
        "categoria": "Electricidad",
        "descripcion": "Desinstalación de luces",
        "fecIni": "2022-01-03",
        "fecFin": null,
        "tiempo": null,
        "idTrabajador": {
            "idTrabajador": "T003",
            "dni": "34567890C",
            "nombre": "Carlos",
            "apellidos": "Martínez",
            "especialidad": "Electricidad",
            "contraseña": "contraseña3",
            "email": "carlos.martinez@example.com"
        },
        "prioridad": 4
    }
],
"error": false
}
```

200 – Empty

```
{
    "result": [],
    "error": false
}
```

GET BASE/sinTrabajador

- ⚠ Returns all the tasks that are pending and have no worker assigned. If there are no pending tasks without a worker assigned, it will return an empty list.

200 – Results

```
{  
  "result": [  
    {  
      "codTrabajo": "J800",  
      "categoria": "Espacio-Tiempo",  
      "descripcion": "Visitar a Sarah Connor",  
      "fecIni": "2029-10-20",  
      "fecFin": null,  
      "tiempo": null,  
      "idTrabajador": null,  
      "prioridad": 1  
    },  
    {  
      "cod_trabajo": "J008",  
      "categoria": "Pintura",  
      "descripcion": "Pintar las paredes",  
      "fec_ini": "2024-01-09",  
      "fec_fin": null,  
      "tiempo": null,  
      "id_trabajador": null,  
      "prioridad": 4  
    }  
}
```

200 – Empty

```
{  
    "result": [],  
    "error": false  
}
```

GET BASE/completados

- ⚠ Returns all the tasks that are completed. If there are no completed tasks, it will return an empty list.

200 – Results

```
{  
    "result": [  
        {  
            "codTrabajo": "J799",  
            "categoria": "Espacio-Tiempo",  
            "descripcion": "Conseguir ropa",  
            "fecIni": "2029-10-20",  
            "fecFin": "1984-10-20",  
            "tiempo": -999.9,  
            "idTrabajador": {  
                "idTrabajador": "T800",  
                "dni": "80080080T",  
                "nombre": "Arnaldo",  
                "apellidos": "Charcheneguer",  
                "especialidad": "Espacio-Tiempo",  
                "contraseña": "Dame tu ropa",  
                "email": "sayonara@volvere.com"  
            },  
            "prioridad": 2  
        }  
    ],  
}
```

```
        "error": false  
    }  
}
```

200 – Empty

```
{  
    "result": [],  
    "error": false  
}  
}
```

GET BASE/:id/trabajador

- ⚠** This endpoint is not necessary since the worker is already included in the task object, but it doesn't hurt anyone.
- ⚠** Returns the worker assigned to a task by its ID. If the task doesn't have a worker assigned, it will return an error message. If the task does not exist, it will return an error message.

200 – Result

```
{  
    "result": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
    },  
}
```

```
        "error": false  
    }  
}
```

404 – Not Found

```
{  
    "errorMessage": "El trabajo con ID 'J800' no tiene un  
    trabajador asignado",  
    "error": false  
}  
}
```

PUT BASE/asignar/:codTrabajo/:idTrabajador

- ⚠** Assigns a worker to a task by their IDs. If the task does not exist, it will return an error message. If the worker does not exist, it will return an error message. If the task already has a worker assigned, it will return an error message.

200 – Success

```
⚠ /api/trabajos/asignar/J800/T800  
  
{  
    "errorMessage": {  
        "codTrabajo": "J800",  
        "categoria": "Espacio-Tiempo",  
        "descripcion": "Visitar a Sarah Connor",  
        "fecIni": "2029-10-20",  
        "fecFin": null,  
        "tiempo": null,  
        "idTrabajador": {  
            "idTrabajador": "T800",  
            "dni": "80080080T",  
            "nombre": "Arnaldo",  
        }  
    }  
}
```

```
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 1
},
"error": false
}
```

404 – Task Not Found

⚠ /api/trabajos/asignar/J8000/T8000

```
{
    "errorMessage": "Trabajo no encontrado",
    "error": true
}
```

404 – Worker Not Found

⚠ /api/trabajos/asignar/J800/T8000

```
{
    "errorMessage": "Trabajador no encontrado",
    "error": true
}
```

400 – Worker already assigned

⚠ /api/trabajos/asignar/J001/T800

```
{  
    "errorMessage": "El trabajo ya tiene un trabajador  
asignado",  
    "error": true  
}
```

PUT BASE/finalizar/:id

- i** Accepts from 1 to 3 parameters:

- id (@PathVariable) (required): task ID

⚠ Mandatory parameter.

- fec_fin (@RequestParam) (not required): task completion date
- tiempo (@RequestParam) (not required): duration of the task

- ⚠** Marks a task as completed by setting the completion date and the duration of the task. Checks if the task exists. Checks if the task has a worker assigned. Checks that tiempo is not greater than 8 hours per day. Checks that fec_fin is not before fecIni. Checks that fec_fin is not after the current date. If fec_fin is not provided, it will set the current date. If tiempo is not provided, it will set the duration to 8 hours per day (not the best algorithm, as no one works 8 hours every day). If the task is already completed, just updates following the same rules mentioned above.

200 – No parameters

⚠ /api/trabajos/finalizar/J8

```
{  
  "result": {  
    "codTrabajo": "J8",  
    "categoria": "Carpintería",  
    "descripcion": "Lijar maderos",  
    "fecIni": "2024-01-09",  
    "fecFin": "2024-05-15",  
    "tiempo": 48,  
    "idTrabajador": {  
      "idTrabajador": "T001",  
      "dni": "12345678A",  
      "nombre": "Juan",  
      "apellidos": "Pérez",  
      "especialidad": "Carpintería",  
      "contraseña": "contraseña1",  
      "email": "juan.perez@example.com"  
    },  
    "prioridad": 4  
  },  
  "error": false  
}
```

200 – Same date as start date, no time

⚠ /api/trabajos/finalizar/J8?fec_fin=2024-01-09

```
{  
  "result": {  
    "codTrabajo": "J8",  
    "categoria": "Carpintería",  
    "descripcion": "Lijar maderos",  
    "fecIni": "2024-01-09",  
    "fecFin": "2024-01-09",  
    "tiempo": 8,  
    "idTrabajador": {
```

```
        "idTrabajador": "T001",
        "dni": "12345678A",
        "nombre": "Juan",
        "apellidos": "Pérez",
        "especialidad": "Carpintería",
        "contraseña": "contraseña1",
        "email": "juan.perez@example.com"
    },
    "prioridad": 4
},
"error": false
}
```

200 – Next day, no time

A /api/trabajos/finalizar/J8?fec_fin=2024-01-10

```
{
  "result": {
    "codTrabajo": "J8",
    "categoria": "Carpintería",
    "descripcion": "Lijar maderos",
    "fecIni": "2024-01-09",
    "fecFin": "2024-01-10",
    "tiempo": 8,
    "idTrabajador": {
      "idTrabajador": "T001",
      "dni": "12345678A",
      "nombre": "Juan",
      "apellidos": "Pérez",
      "especialidad": "Carpintería",
      "contraseña": "contraseña1",
      "email": "juan.perez@example.com"
    },
    "prioridad": 4
  }
}
```

```
    },
    "error": false
}
```

200 – No day, acceptable time

A /api/trabajos/finalizar/J8?tiempo=10

```
{
  "result": {
    "codTrabajo": "J8",
    "categoria": "Carpintería",
    "descripcion": "Lijar maderos",
    "fecIni": "2024-01-09",
    "fecFin": "2024-05-15",
    "tiempo": 10,
    "idTrabajador": {
      "idTrabajador": "T001",
      "dni": "12345678A",
      "nombre": "Juan",
      "apellidos": "Pérez",
      "especialidad": "Carpintería",
      "contraseña": "contraseña1",
      "email": "juan.perez@example.com"
    },
    "prioridad": 4
  },
  "error": false
}
```

400 – Acceptable day, unacceptable time

A /api/trabajos/finalizar/J8?fec_fin=2024-01-10&tiempo=10

```
{  
  "errorMessage": "El tiempo no puede ser mayor a 8 horas por  
d\u00f3a trabajado",  
  "error": true  
}
```

404 – Not Found

⚠ /api/trabajos/finalizar/J009

```
{  
  "errorMessage": "Trabajo no encontrado",  
  "error": true  
}
```

400 – Worker not assigned

⚠ /api/trabajos/finalizar/J008

```
{  
  "errorMessage": "El trabajo no tiene un trabajador  
asignado",  
  "error": true  
}
```

400 – Date before start date

⚠ /api/trabajos/finalizar/J8?fec_fin=2024-01-04

```
{  
  "errorMessage": "La fecha de finalización no puede ser menor  
  a la fecha de inicio",  
  "error": true  
}
```

400 – Date after current date

⚠ /api/trabajos/finalizar/J8?fec_fin=2094-01-10

```
{  
  "errorMessage": "La fecha de finalización no puede ser mayor  
  a la fecha actual",  
  "error": true  
}
```

400 – More than 8h/day

⚠ /api/trabajos/finalizar/J8?fec_fin=2024-01-10&tiempo=200

```
{  
  "errorMessage": "El tiempo no puede ser mayor a 8 horas por  
  día trabajado",  
  "error": true  
}
```

400 – Negative time

⚠ /api/trabajos/finalizar/J8?fec_fin=2024-01-10&tiempo=-2

```
{  
    "errorMessage": "El tiempo no puede ser menor a 0",  
    "error": true  
}
```

PUT BASE/editarFechaFin/:id

⚠ This endpoint is necessary, as simple PUT BASE /:id does not check anything, and PUT BASE /finalizar/:id will calculate `tiempo` when it's not provided.

⚠ Same as PUT BASE /finalizar/:id but only for the completion date.

200 – Acceptable date

⚠ `/api/trabajos/editarFechaFin/J8?fec_fin=2024-01-10`

```
{  
    "result": {  
        "codTrabajo": "J8",  
        "categoria": "Carpintería",  
        "descripcion": "Lijar maderos",  
        "fecIni": "2024-01-09",  
        "fecFin": "2024-01-10",  
        "tiempo": 2.0,  
        "idTrabajador": {  
            "idTrabajador": "T001",  
            "dni": "12345678A",  
            "nombre": "Juan",  
            "apellidos": "Pérez",  
            "especialidad": "Carpintería",  
            "contraseña": "contraseña1",  
            "email": "juan.perez@example.com"  
        },  
    },  
}
```

```
        "prioridad": 4
    },
    "error": false
}
```

200 – No parameter

⚠ /api/trabajos/editarFechaFin/J8

```
{
  "result": {
    "codTrabajo": "J8",
    "categoria": "Carpintería",
    "descripcion": "Lijar maderos",
    "fecIni": "2024-01-09",
    "fecFin": "2024-05-15",
    "tiempo": 2.0,
    "idTrabajador": {
      "idTrabajador": "T001",
      "dni": "12345678A",
      "nombre": "Juan",
      "apellidos": "Pérez",
      "especialidad": "Carpintería",
      "contraseña": "contraseña1",
      "email": "juan.perez@example.com"
    },
    "prioridad": 4
  },
  "error": false
}
```

400 – Date before start date

⚠ /api/trabajos/editarFechaFin/J8?fec_fin=2024-01-05

```
{  
  "errorMessage": "La fecha de finalización no puede ser menor  
  a la fecha de inicio",  
  "error": true  
}
```

400 – Date after current date

⚠ /api/trabajos/editarFechaFin/J8?fec_fin=2094-01-10

```
{  
  "errorMessage": "La fecha de finalización no puede ser mayor  
  a la fecha actual",  
  "error": true  
}
```

PUT BASE/editarTiempo/:id

⚠ This endpoint is necessary, as simple PUT BASE /:id does not check anything, and PUT BASE /finalizar/:id will assume today's date when fec_fin is not provided.

⚠ Same as PUT BASE /finalizar/:id but only for the duration of the task. Also checks that fec_fin is not null

200 – Acceptable time

⚠ /api/trabajos/editarTiempo/J8?tiempo=4

```
{  
  "result": {  
    "codTrabajo": "J8",  
    "categoria": "Carpintería",  
    "descripcion": "Lijar maderos",  
    "fecIni": "2024-01-09",  
    "fecFin": "2024-01-10",  
    "tiempo": 4.0,  
    "idTrabajador": {  
      "idTrabajador": "T001",  
      "dni": "12345678A",  
      "nombre": "Juan",  
      "apellidos": "Pérez",  
      "especialidad": "Carpintería",  
      "contraseña": "contraseña1",  
      "email": "juan.perez@example.com"  
    },  
    "prioridad": 4  
  },  
  "error": false  
}
```

200 – No parameter

⚠ /api/trabajos/editarTiempo/J8

```
{  
  "result": {  
    "codTrabajo": "J8",  
    "categoria": "Carpintería",  
    "descripcion": "Lijar maderos",  
    "fecIni": "2024-01-09",  
    "fecFin": "2024-01-10",  
    "prioridad": 4  
  }  
}
```

```
"fecIni": "2024-01-09",
"fecFin": "2024-05-15",
"tiempo": 48.0,
"idTrabajador": {
    "idTrabajador": "T001",
    "dni": "12345678A",
    "nombre": "Juan",
    "apellidos": "Pérez",
    "especialidad": "Carpintería",
    "contraseña": "contraseña1",
    "email": "juan.perez@example.com"
},
"prioridad": 4
},
"error": false
}
```

400 – Date before start date

⚠ /api/trabajos/editarTiempo/J8?fec_fin=2024-01-05

```
{
    "errorMessage": "La fecha de finalización no puede ser menor
a la fecha de inicio",
    "error": true
}
```

400 – Date after current date

⚠ /api/trabajos/editarTiempo/J8?fec_fin=2094-01-10

```
{
    "errorMessage": "La fecha de finalización no puede ser mayor
```

```
a la fecha actual",
  "error": true
}
```

400 – Null date

⚠ /api/trabajos/editarFechaFin/J8?tiempo=3

Assuming fec_fin is null

```
{
  "errorMessage": "El trabajo no tiene una fecha de
finalización asignada",
  "error": true
}
```

Trabajador

Model

C Trabajador	
□ contraseña: String	
□ trabajos: Set<Trabajo>	
□ idTrabajador: String	
□ especialidad: String	
□ nombre: String	
□ email: String	
□ dni: String	
□ apellidos: String	
● Trabajador(String, String, String, String, String, String):	
● Trabajador():	
● Trabajador(String, String, String, String, String, String, String):	
● setDni(String): void	
● setEspecialidad(String): void	
● setTrabajos(Set<Trabajo>): void	
● toString(): String	
● setApellidos(String): void	
● getEspecialidad(): String	
● setContraseña(String): void	
● getApellidos(): String	
● getIdTrabajador(): String	
● getEmail(): String	
● setNombre(String): void	
● setEmail(String): void	
● getNombre(): String	
● getTrabajos(): Set<Trabajo>	
● getDni(): String	
● setIdTrabajador(String): void	
● getContraseña(): String	

BASE: /api/trabajadores

GET BASE

- ⚠ Returns all the workers in the database. If there are no workers, it will return an empty list.

Successful response

```
{  
  "result": [  
    {  
      "idTrabajador": "T003",  
      "dni": "34567890C",  
      "nombre": "Carlos",  
      "apellidos": "Martínez",  
      "especialidad": "Electricidad",  
      "contraseña": "contraseña3",  
      "email": "carlos.martinez@example.com"  
    },  
    {  
      "idTrabajador": "T001",  
      "dni": "12345678A",  
      "nombre": "Juan",  
      "apellidos": "Pérez",  
      "especialidad": "Carpintería",  
      "contraseña": "contraseña1",  
      "email": "juan.perez@example.com"  
    }  
,  
  "error": false  
}
```

Empty list response

```
{  
  "result": [],  
  "error": false  
}
```

GET BASE/:id

⚠ Returns a worker by its ID. If the worker does not exist, it will return an error message.

Successful response

⚠ /api/trabajadores/T003

```
{  
  "result": {  
    "idTrabajador": "T003",  
    "dni": "34567890C",  
    "nombre": "Carlos",  
    "apellidos": "Martínez",  
    "especialidad": "Electricidad",  
    "contraseña": "contraseña3",  
    "email": "carlos.martinez@example.com"  
  },  
  "error": false  
}
```

Non-existent worker response

⚠ /api/trabajadores/asdtg

```
{  
  "errorMessage": "El trabajador con ID: 'asdtg' no existe en  
  la base de datos",  
  "error": true  
}
```

POST BASE

⚠ Create a new worker in the database. Returns the worker created. If there are any errors, it will return an error message with the list of errors. If the worker already exists, it will return an error message.

All the fields are required. ID, DNI and email are unique fields.

Successful body

```
POST /api/trabajadores
Content-Type: application/json
{
  "idTrabajador": "T001",
  "dni": "12345678A",
  "nombre": "Pepe",
  "apellidos": "García",
  "especialidad": "Fontanería",
  "contraseña": "123",
  "email": "wefwefewf@ergerg.com"
}
```

Response

```
{
  "result": {
    "idTrabajador": "T001",
    "dni": "12345678A",
    "nombre": "Pepe",
    "apellidos": "García",
    "especialidad": "Fontanería",
    "contraseña": "123",
    "email": "wefwefewf@ergerg.com"
  },
}
```

```
        "error": false  
    }
```

Error body

```
POST /api/trabajadores  
Content-Type: application/json  
{  
    "idTrabajador": "T001",  
    "dni": "",  
    "nombre": "Pepe",  
    "apellidos": "García",  
    "especialidad": "Fontanería",  
    "contraseña": "123",  
    "email": "holaHolita"  
}
```

Error response

```
{  
    "errorsList": [  
        "El campo DNI no puede estar vacío",  
        "El email no es válido"  
    ],  
    "error": true  
}
```

Existent worker body

```
POST /api/trabajadores  
Content-Type: application/json  
{  
    "idTrabajador": "T001",
```

```
"dni": "12312312T",
"nombre": "Pepe",
"apellidos": "García",
"especialidad": "Fontanería",
"contraseña": "123",
"email": "pepe@garcia.com"
}
```

Existent worker response

```
{
  "errorsList": [
    "Ya existe un trabajador con el ID: T001"
  ],
  "error": true
}
```

PUT BASE/:id

⚠ Update a worker by its ID. Returns the worker updated. If the worker does not exist, it will return an error message. If any of the fields are empty or invalid, it will return an error message with the list of errors.

Successful body

⚠ /api/trabajadores/T001

```
PUT /api/trabajadores
Content-Type: application/json
{
  "idTrabajador": "T001",
  "dni": "12345678A",
  "nombre": "Pepe",
```

```
        "apellidos": "Gomez",
        "especialidad": "Fontanería",
        "contraseña": "123",
        "email": "wefwefewf@ergerg.com"
    }
```

Response

```
{
  "result": {
    "idTrabajador": "T001",
    "dni": "12345678A",
    "nombre": "Pepe",
    "apellidos": "Gomez",
    "especialidad": "Fontanería",
    "contraseña": "123",
    "email": "wefwefewf@ergerg.com"
  },
  "error": false
}
```

Error body

⚠ /api/trabajadores/T001

```
PUT /api/trabajadores
Content-Type: application/json
{
  "idTrabajador": "T001",
  "dni": "",
  "nombre": "Pepe",
  "apellidos": "García",
  "especialidad": "Fontanería",
  "contraseña": "123",
```

```
        "email": "holaHolita"  
    }
```

Error response

```
{  
    "errorsList": [  
        "El campo DNI no puede estar vacío",  
        "El email no es válido"  
    ],  
    "error": true  
}
```

Non-existent worker body

⚠ /api/trabajadores/T999

```
PUT /api/trabajadores  
Content-Type: application/json  
{  
    "idTrabajador": "T999",  
    "dni": "12344321T",  
    "nombre": "José",  
    "apellidos": "Jimenez",  
    "especialidad": "Ebanistería",  
    "contraseña": "123",  
    "email": "jose@jimenez.com"  
}
```

Non-existent worker response

```
{  
    "errorsList": [
```

```
        "No se pudo editar, el trabajador con ID 'T999' no existe  
        en la base de datos"  
    ],  
    "error": true  
}
```

DELETE BASE/:id

⚠ Delete a worker by its ID. Returns a response without errors. If the worker does not exist, it will return an error message.

Successful response

```
⚠ /api/trabajos/T001  
  
{  
    "error": false  
}
```

Non-existent worker response

```
⚠ /api/trabajos/T999  
  
{  
    "errorMessage": "El trabajador con ID 'T999' no existe en la  
    base de datos",  
    "error": true  
}
```

GET BASE/:id/:contraseña

⚠ Returns the pending tasks of a worker by its ID and password. If the worker does not exist, it will return an error message. If the credentials are incorrect, it will return an error message. If the worker has no pending tasks, it will return an empty list.

200 – empty

⚠ /api/trabajadores/T800/Dame%20tu%20ropa

```
{  
    "result": [],  
    "error": false  
}
```

200 – results

⚠ /api/trabajadores/T003/contrase%C3%B1a3

```
{  
    "result": [  
        {  
            "codTrabajo": "J003",  
            "categoria": "Electricidad",  
            "descripcion": "Instalación de luces",  
            "fecIni": "2022-01-03",  
            "fecFin": null,  
            "tiempo": 8.0,  
            "idTrabajador": {  
                "idTrabajador": "T003",  
                "dni": "34567890C",  
                "nombre": "Carlos",  
                "apellidos": "Martínez",  
                "especialidad": "Electricidad",  
                "correo": "carlos.martinez@correo.com",  
                "telefono": "987654321",  
                "direccion": "Calle 123, Oficina 456",  
                "ciudad": "Bogotá",  
                "pais": "Colombia",  
                "edad": 30,  
                "genero": "Masculino",  
                "status": "Activo",  
                "password": "Contraseña123" // Observe que el password es una constante de ejemplo  
            }  
        }  
    ]  
}
```

```

        "contraseña": "contraseña3",
        "email": "carlos.martinez@example.com"
    },
    "prioridad": 3
},
{
    "codTrabajo": "J004",
    "categoria": "Electricidad",
    "descripcion": "Desinstalación de luces",
    "fecIni": "2022-01-03",
    "fecFin": null,
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T003",
        "dni": "34567890C",
        "nombre": "Carlos",
        "apellidos": "Martínez",
        "especialidad": "Electricidad",
        "contraseña": "contraseña3",
        "email": "carlos.martinez@example.com"
    },
    "prioridad": 4
}
],
"error": false
}

```

404 – Not Found

⚠ /api/trabajadores/T009/Dame_tu SONRISA

```
{
    "errorMessage": "El trabajador con ID: 'T009' no existe en
la base de datos",
```

```
        "error": true  
    }
```

404 – Not Found

⚠ /api/trabajadores/T800/Dame_tu SONRISA

```
{  
    "errorMessage": "Las credenciales son incorrectas",  
    "error": true  
}
```

GET BASE/especialidad/:especialidad

⚠ Returns all the workers with a specific specialty. If there are no workers with that specialty, it will return an empty list.

200 – results

⚠ /api/trabajadores/especialidad/espacio-tiempo

```
{  
    "result": [  
        {  
            "idTrabajador": "T800",  
            "dni": "80080080T",  
            "nombre": "Arnaldo",  
            "apellidos": "Charcheneguer",  
            "especialidad": "Espacio-Tiempo",  
            "contraseña": "Dame tu ropa",  
            "email": "sayonara@volvere.com"  
        }  
    ]
```

```
],  
    "error": false  
}
```

200 – empty

⚠ `/api/trabajadores/especialidad/hghfgtgcghvuhbk`

```
{  
    "result": [],  
    "error": false  
}
```

GET BASE/:id/trabajos

⚠ Returns a list of all the tasks of a worker by its ID. If the worker has no tasks, it will return an empty list. If the worker does not exist, it will return an error message.

200 – results

⚠ `/api/trabajadores/T800/trabajos`

```
{  
    "result": [  
        {  
            "codTrabajo": "J799",  
            "categoria": "Espacio-Tiempo",  
            "descripcion": "Conseguir ropa",  
            "fecIni": "2029-10-20",  
            "fecFin": "1984-10-20",  
            "tiempo": -999.9,  
            "ubicacion": "Calle de la Constitución, 123, Madrid, España",  
            "status": "En Proceso",  
            "prioridad": "Alta",  
            "recomendaciones": "Usar ropa ligera y protegerse del sol.",  
            "detalles": "Este trabajo consiste en recoger una gran cantidad de ropa usada para su posterior venta en un puesto callejero. Se requiere una fuerza física considerable y se recomienda usar ropa ligera y protegerse del sol."  
        }  
    ]  
}
```

```
"idTrabajador": {  
    "idTrabajador": "T800",  
    "dni": "80080080T",  
    "nombre": "Arnaldo",  
    "apellidos": "Charcheneguer",  
    "especialidad": "Espacio-Tiempo",  
    "contraseña": "Dame tu ropa",  
    "email": "sayonara@volvere.com"  
},  
    "prioridad": 2  
},  
{  
    "codTrabajo": "J800",  
    "categoria": "Espacio-Tiempo",  
    "descripcion": "Conseguir ropa",  
    "fecIni": "2029-10-20",  
    "fecFin": null,  
    "tiempo": null,  
    "idTrabajador": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
    },  
    "prioridad": 2  
}  
],  
    "error": false  
}
```

200 – empty

⚠ /api/trabajadores/TDUDE/trabajos

```
{  
  "result": [],  
  "error": false  
}
```

404 – ID Not Found

⚠ /api/trabajadores/T900/trabajos

```
{  
  "errorMessage": "El trabajador con ID: 'T9' no existe en la  
  base de datos",  
  "error": true  
}
```

GET BASE/:id/trabajos/pendientes

⚠ Returns a list of all the pending tasks of a worker by its ID. If the worker has no pending tasks, it will return an empty list. If the worker does not exist, it will return an error message.

200 – Results

⚠ /api/trabajadores/T800/trabajos/pendientes

```
{  
  "result": [  
    {
```

```
"codTrabajo": "J800",
"categoria": "Espacio-Tiempo",
"descripcion": "Visitar a Sarah Connor",
"fecIni": "2029-10-20",
"fecFin": null,
"tiempo": null,
"idTrabajador": {
    "idTrabajador": "T800",
    "dni": "80080080T",
    "nombre": "Arnaldo",
    "apellidos": "Charcheneguer",
    "especialidad": "Espacio-Tiempo",
    "contraseña": "Dame tu ropa",
    "email": "sayonara@volvere.com"
},
"prioridad": 1
},
{
    "codTrabajo": "J803",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Sonreír",
    "fecIni": "2029-10-20",
    "fecFin": null,
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 4
},
{
    "codTrabajo": "J801",
    "categoria": "Espacio-Tiempo",
```

```
"descripcion": "Conseguir un amoto",
"fecIni": "2029-10-20",
"fecFin": null,
"tiempo": null,
"idTrabajador": {
    "idTrabajador": "T800",
    "dni": "80080080T",
    "nombre": "Arnaldo",
    "apellidos": "Charcheneguer",
    "especialidad": "Espacio-Tiempo",
    "contraseña": "Dame tu ropa",
    "email": "sayonara@volvere.com"
},
"prioridad": 2
},
],
"error": false
}
```

200 – Empty

⚠ /api/trabajadores/T800/trabajos/pendientes

```
{
  "result": [],
  "error": false
}
```

404 – Worker not found

⚠ /api/trabajadores/T099/trabajos/pendientes

```
{  
  "errorMessage": "El trabajador con ID: 'T099' no existe en  
  la base de datos",  
  "error": true  
}
```

GET BASE/:id/trabajos/pendientes/prioridad

- ⚠ Returns an ascendant sorted list of all the pending tasks of a worker by its ID. If the worker has no pending tasks, it will return an empty list. If the worker does not exist, it will return an error message.

200 – Results

⚠ /api/trabajadores/T800/trabajos/pendientes/prioridad

```
{  
  "result": [  
    {  
      "codTrabajo": "J800",  
      "categoria": "Espacio-Tiempo",  
      "descripcion": "Visitar a Sarah Connor",  
      "fecIni": "2029-10-20",  
      "fecFin": null,  
      "tiempo": null,  
      "idTrabajador": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
      },  
    },  
  ]
```

```
        "prioridad": 1
    },
{
    "codTrabajo": "J801",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Conseguir un amoto",
    "fecIni": "2029-10-20",
    "fecFin": null,
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 2
},
{
    "codTrabajo": "J803",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Sonreír",
    "fecIni": "2029-10-20",
    "fecFin": null,
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 4
}
```

```
],  
    "error": false  
}
```

200 – Empty

⚠ /api/trabajadores/T800/trabajos/pendientes/prioridad

```
{  
    "result": [],  
    "error": false  
}
```

404 – Worker not found

⚠ /api/trabajadores/T099/trabajos/pendientes/prioridad

```
{  
    "errorMessage": "El trabajador con ID: 'T099' no existe en  
    la base de datos",  
    "error": true  
}
```

GET BASE/:id/trabajos/pendientes/:prioridad

⚠ Returns a list of all the pending tasks of a worker by its ID and the specified priority. If the worker has no pending tasks, it will return an empty list. If the worker does not exist, it will return an error message.

200 – Results

⚠ /api/trabajadores/T800/trabajos/pendientes/4

```
{  
  "result": [  
    {  
      "codTrabajo": "J803",  
      "categoria": "Espacio-Tiempo",  
      "descripcion": "Sonreír",  
      "fecIni": "2029-10-20",  
      "fecFin": null,  
      "tiempo": null,  
      "idTrabajador": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
      },  
      "prioridad": 4  
    }  
  ],  
  "error": false  
}
```

200 – Empty

⚠ /api/trabajadores/T800/trabajos/pendientes/3

```
{  
  "result": [],
```

```
        "error": false
    }
```

404 – Worker not found

⚠ /api/trabajadores/T099/trabajos/pendientes/3

```
{
    "errorMessage": "El trabajador con ID: 'T099' no existe en
    la base de datos",
    "error": true
}
```

GET BASE/:id/trabajos/completados

i Accepts two optional query parameters: fechAlni and fechaFin

⚠ Returns a list of all the completed tasks of a worker by its ID. If the worker has no completed tasks, it will return an empty list. If the worker does not exist, it will return an error message.

200 – No param | Results

⚠ /api/trabajadores/T800/trabajos/completados

```
{
    "result": [
        {
            "codTrabajo": "J799",
            "categoria": "Espacio-Tiempo",
            "descripcion": "Conseguir ropa",
            "fechaAlni": "2023-09-15T10:00:00Z",
            "fechaFin": "2023-09-15T12:00:00Z"
        }
    ]
}
```

```
"fecIni": "2029-10-20",
"fecFin": "1984-10-20",
"tiempo": -999.9,
"idTrabajador": {
    "idTrabajador": "T800",
    "dni": "80080080T",
    "nombre": "Arnaldo",
    "apellidos": "Charcheneguer",
    "especialidad": "Espacio-Tiempo",
    "contraseña": "Dame tu ropa",
    "email": "sayonara@volvere.com"
},
"prioridad": 2
},
{
    "codTrabajo": "J803",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Sonreír",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-21",
    "tiempo": -999.9,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 4
},
{
    "codTrabajo": "J801",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Conseguir un amoto",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-20",
```

```
"tiempo": null,
"idTrabajador": {
    "idTrabajador": "T800",
    "dni": "80080080T",
    "nombre": "Arnaldo",
    "apellidos": "Charcheneguer",
    "especialidad": "Espacio-Tiempo",
    "contraseña": "Dame tu ropa",
    "email": "sayonara@volvere.com"
},
"prioridad": 2
},
{
    "codTrabajo": "J800",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Visitar a Sarah Connor",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-23",
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 1
}
],
"error": false
}
```

200 — No param | Empty

A /api/trabajadores/TDUDE/trabajos/completados

```
{  
  "result": [],  
  "error": false  
}
```

200 – fechalni | Results

A /api/trabajadores/T800/trabajos/completados?fechalni=1984-10-21

```
{  
  "result": [  
    {  
      "codTrabajo": "J803",  
      "categoria": "Espacio-Tiempo",  
      "descripcion": "Sonreír",  
      "fecIni": "2029-10-20",  
      "fecFin": "1984-10-21",  
      "tiempo": -999.9,  
      "idTrabajador": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
      },  
      "prioridad": 4  
    },  
    {  
      "codTrabajo": "J800",  
      "categoria": "Espacio-Tiempo",  
      "descripcion": "Rendirme",  
      "fecIni": "2029-10-20",  
      "fecFin": "1984-10-21",  
      "tiempo": -999.9,  
      "idTrabajador": {  
        "idTrabajador": "T800",  
        "dni": "80080080T",  
        "nombre": "Arnaldo",  
        "apellidos": "Charcheneguer",  
        "especialidad": "Espacio-Tiempo",  
        "contraseña": "Dame tu ropa",  
        "email": "sayonara@volvere.com"  
      },  
      "prioridad": 4  
    }  
  ]  
}
```

```

    "categoria": "Espacio-Tiempo",
    "descripcion": "Visitar a Sarah Connor",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-23",
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 1
}
],
"error": false
}

```

200 — fechaFin | Results

⚠ /api/trabajadores/T800/trabajos/completados?fechaFin=1984-10-21

```
{
  "result": [
    {
      "codTrabajo": "J799",
      "categoria": "Espacio-Tiempo",
      "descripcion": "Conseguir ropa",
      "fecIni": "2029-10-20",
      "fecFin": "1984-10-20",
      "tiempo": -999.9,
      "idTrabajador": {
        "idTrabajador": "T800",

```

```
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 2
},
{
    "codTrabajo": "J803",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Sonreír",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-21",
    "tiempo": -999.9,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 4
},
{
    "codTrabajo": "J801",
    "categoria": "Espacio-Tiempo",
    "descripcion": "Conseguir un amoto",
    "fecIni": "2029-10-20",
    "fecFin": "1984-10-20",
    "tiempo": null,
    "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnaldo",
```

```

        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
    },
    "prioridad": 2
}
],
"error": false
}

```

200 – fechalni + fechaFin | Results

A /api/trabajadores/T800/trabajos/completados?fechalni=1984-10-21&fechaFin=1984-10-22

```

{
  "result": [
    {
      "codTrabajo": "J803",
      "categoria": "Espacio-Tiempo",
      "descripcion": "Sonreír",
      "fecIni": "2029-10-20",
      "fecFin": "1984-10-21",
      "tiempo": -999.9,
      "idTrabajador": {
        "idTrabajador": "T800",
        "dni": "80080080T",
        "nombre": "Arnoldo",
        "apellidos": "Charcheneguer",
        "especialidad": "Espacio-Tiempo",
        "contraseña": "Dame tu ropa",
        "email": "sayonara@volvere.com"
      },
      "prioridad": 4
    }
  ]
}

```

```
    },
],
"error": false
}
```

404 – Worker not found

⚠ /api/trabajadores/T099/trabajos/pendientes/3

```
{
  "errorMessage": "El trabajador con ID: 'T099' no existe en
la base de datos",
  "error": true
}
```

Trabajador - Views

GET /trabajadores

- ⚠ Returns a view with all the workers in the database
Trabajadores/indexTrabajadores (<http://localhost:8080/trabajadores>)

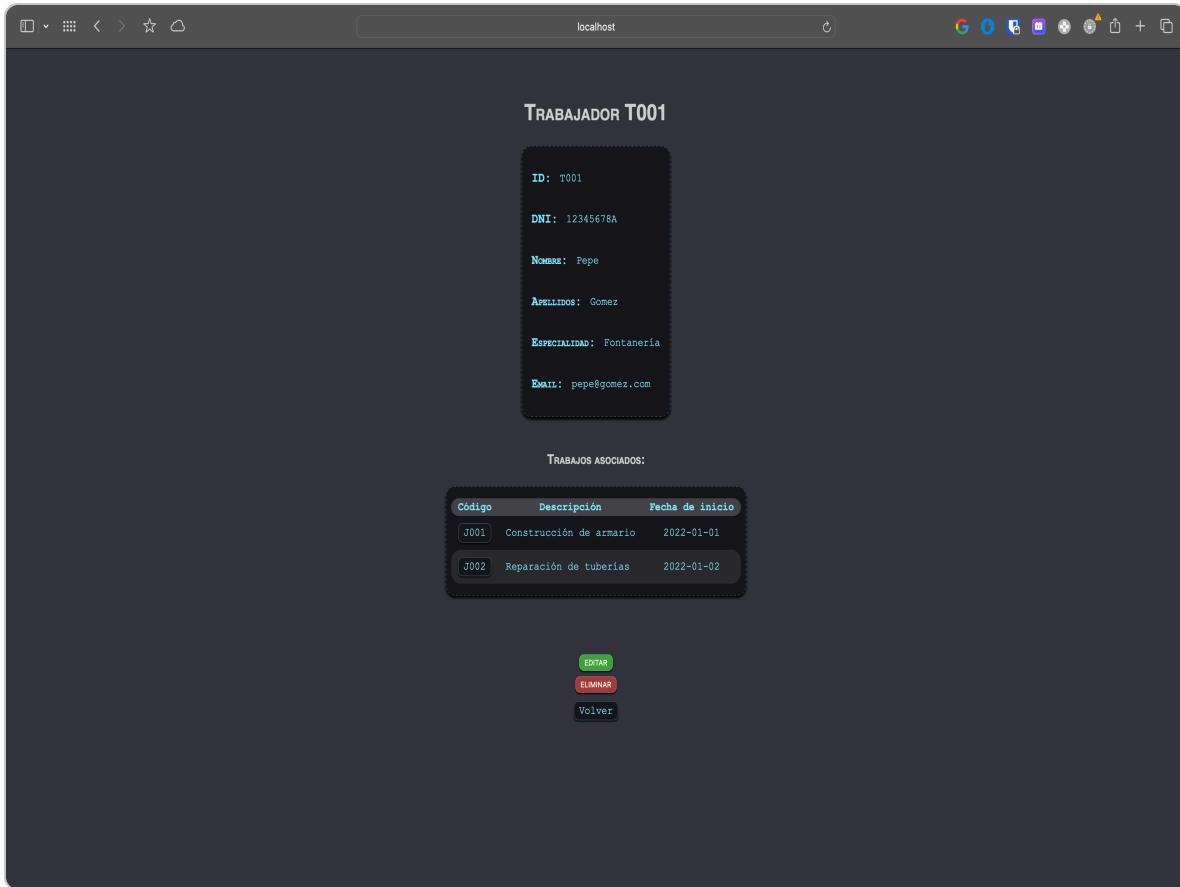
The screenshot shows a web application interface titled "TRABAJADORES". At the top, there is a header bar with a search field containing "localhost" and various browser icons. Below the header, a button labeled "Añadir" is visible. The main content area displays a table with the following data:

ID	Dni	Nombre	Apellidos	Especialidad	Email	Trabajos asociados	Opciones
T003	34567890C	Carlos	Martinez	Electricidad	carlos.martinez@example.com	J003	<button>EDITAR</button> <button>ELIMINAR</button>
T001	12345678A	Pepe	Gomez	Fontaneria	pepe@gomez.com	J001 J002	<button>EDITAR</button> <button>ELIMINAR</button>
T002	12341234R	Ana	Gómez	Carpintería	ana@gomez.com	-	<button>EDITAR</button> <button>ELIMINAR</button>

trabajadores-table.png

GET /trabajadores/:id

- ⚠ Returns a view with the detailed information of a specific worker
Trabajadores/trabajadoresDetalle
(<http://localhost:8080/trabajadores/T001>)



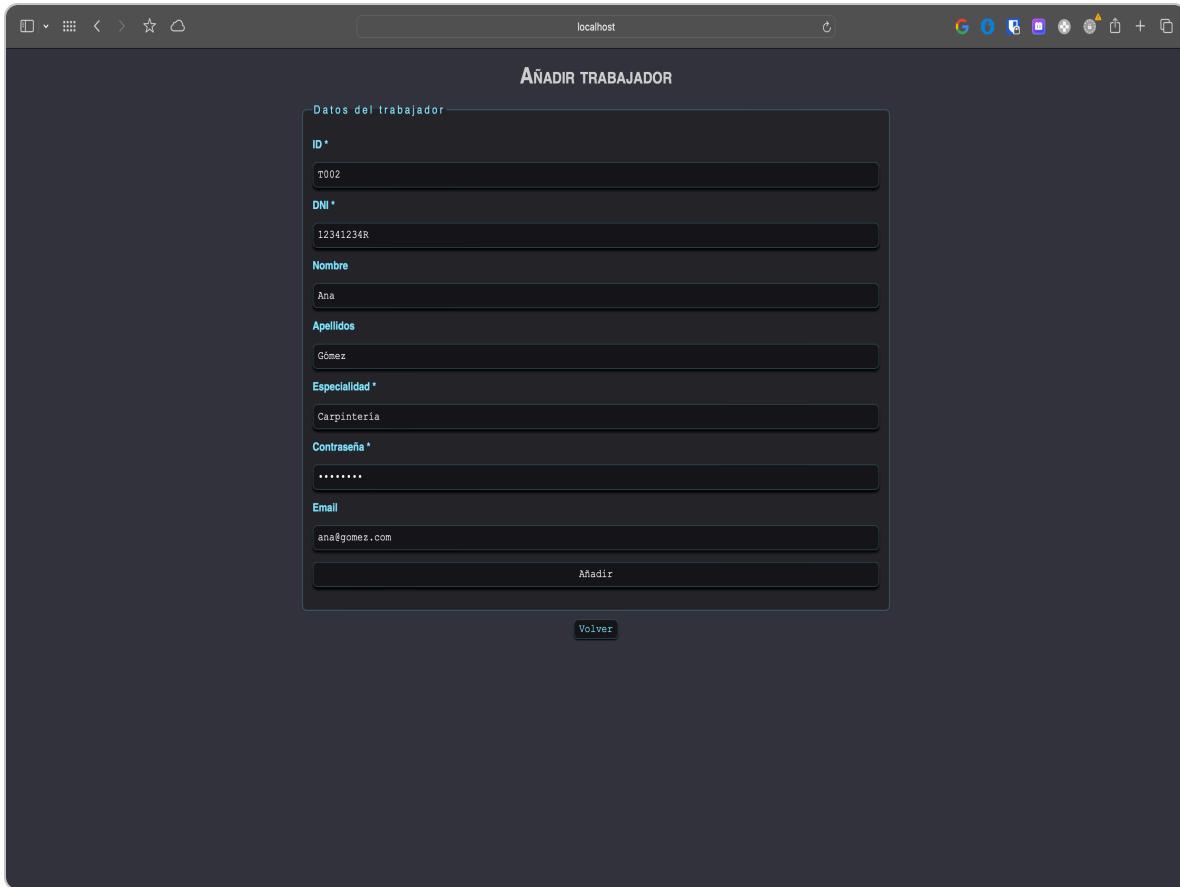
trabajadores-detail.png

GET /trabajadores/nuevo



Returns a view with a form to create a new worker

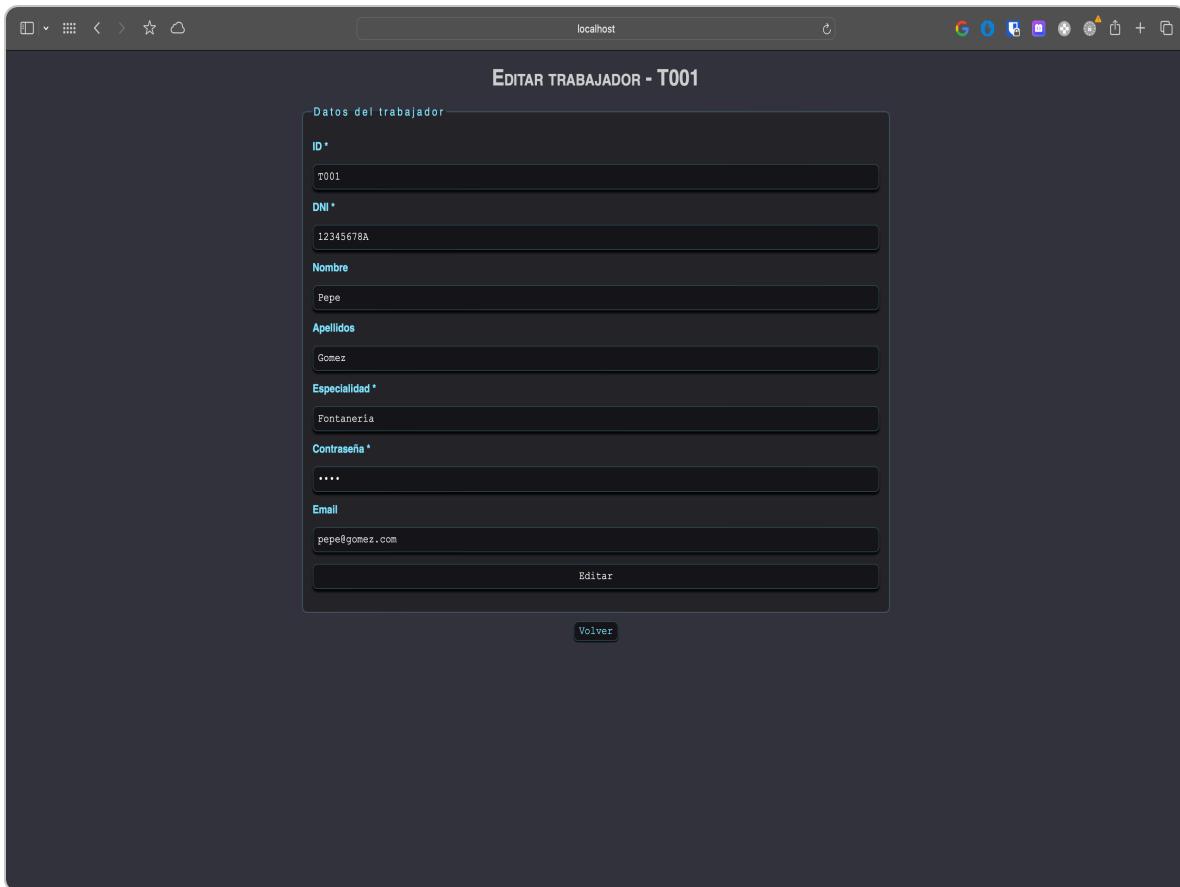
Trabajadores/trabajadoresForm (<http://localhost:8080/trabajadores/nuevo>)



trabajadores-add.png

GET /trabajadores/editar

⚠ Returns a view with the same form that 'trabajadores/nuevo', but modified to edit a worker Trabajadores/trabajadoresForm
(<http://localhost:8080/trabajadores/editar?idTrabajador=T001>)



trabajadores-edit.png

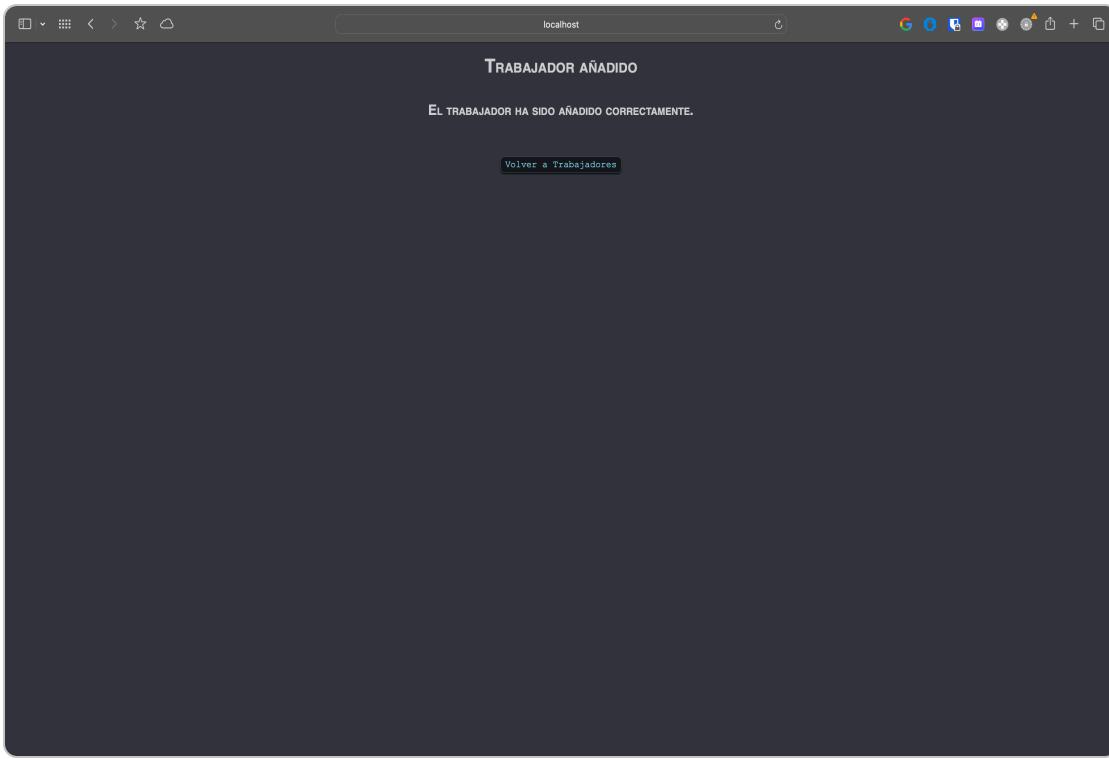
GET /trabajadores/processForm

⚠ This endpoint is not meant to be accessed directly, but to process the form in 'trabajadores/nuevo' and 'trabajadores/editar'. It checks if the method is POST or PUT and redirects to the corresponding endpoint

POST /trabajadores/create

⚠ Creates a new worker in the database. If there is no error, redirects to the 'ready' view with a success message. If there is an error, redirects to the 'trabajadores/nuevo' view with an error message

Successfully worker added



Add success screenshot

Error adding a worker

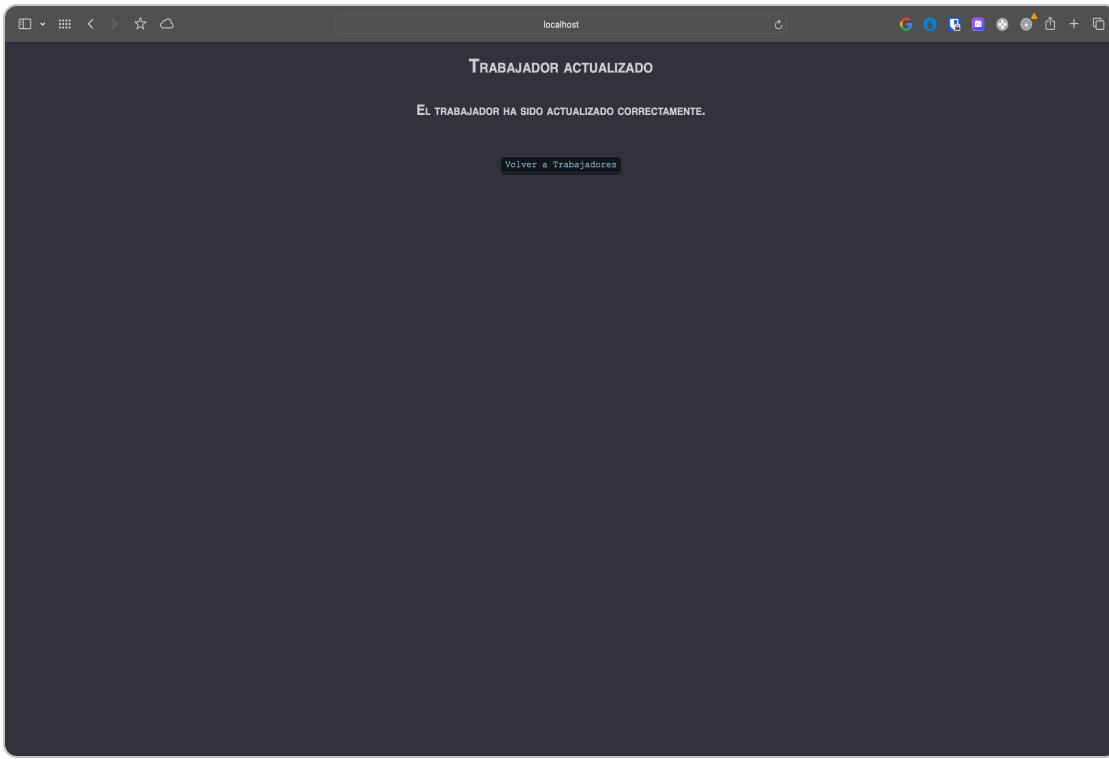
The screenshot shows a dark-themed web application window titled "AÑADIR TRABAJADOR". A red-bordered box labeled "ERRORES:" contains four validation errors: "El campo Id no puede estar vacío", "El campo especialidad no puede estar vacío", "El campo contraseña no puede estar vacío", and "El campo DNI no puede estar vacío". Below this, a form titled "Datos del trabajador" is displayed. It includes fields for "ID *", "DNI *", "Nombre", "Apellidos", "Especialidad *", "Contraseña *", and "Email". At the bottom right of the form is a "Añadir" button. At the very bottom of the page is a "Volver" button.

Add error screenshot

PUT /trabajadores/edit

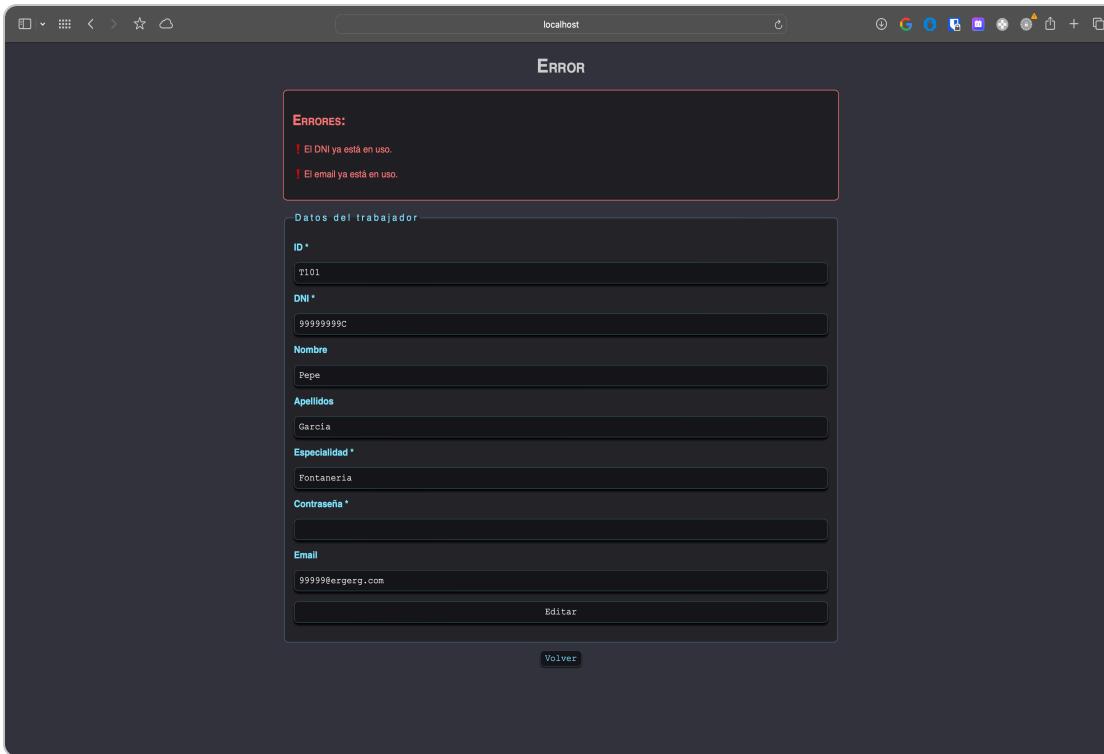
- ⚠️ Updates the information of a worker in the database. If there is no error, redirects to the 'ready' view with a success message. If there is an error, redirects to the 'trabajadores/editar' view with an error message.

Successfully worker edited



Edit success screenshot

Error editing a worker

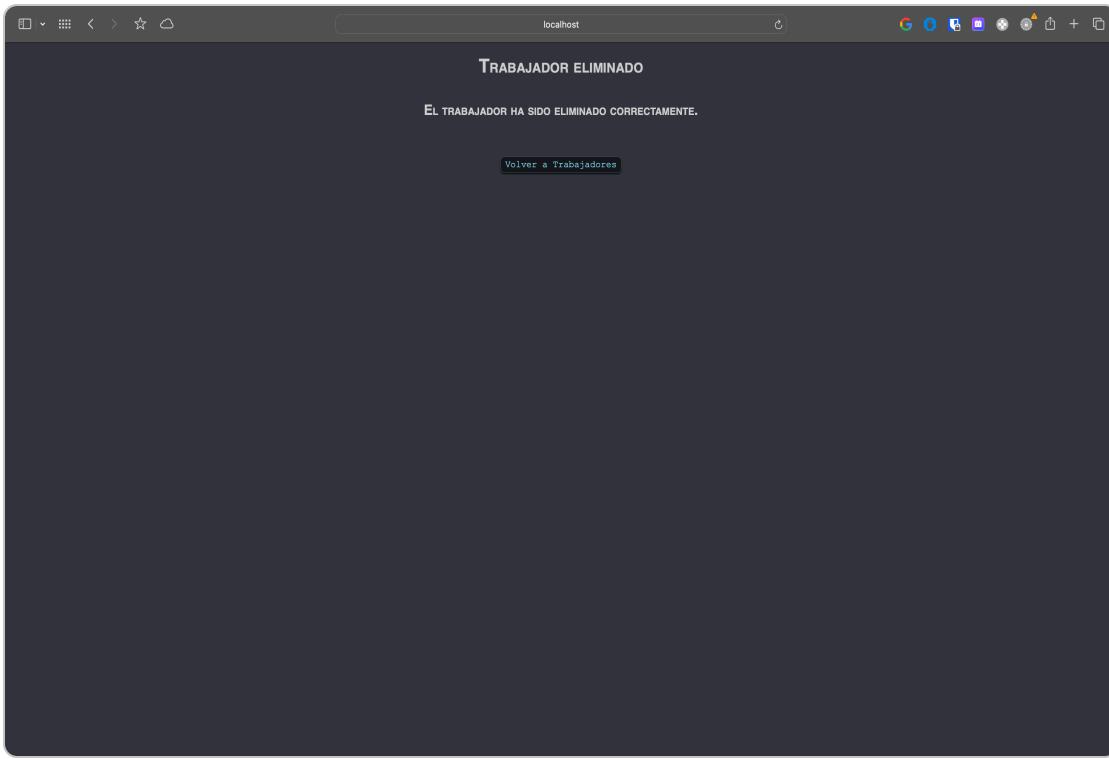


Edit error screenshot

DELETE /trabajadores/delete

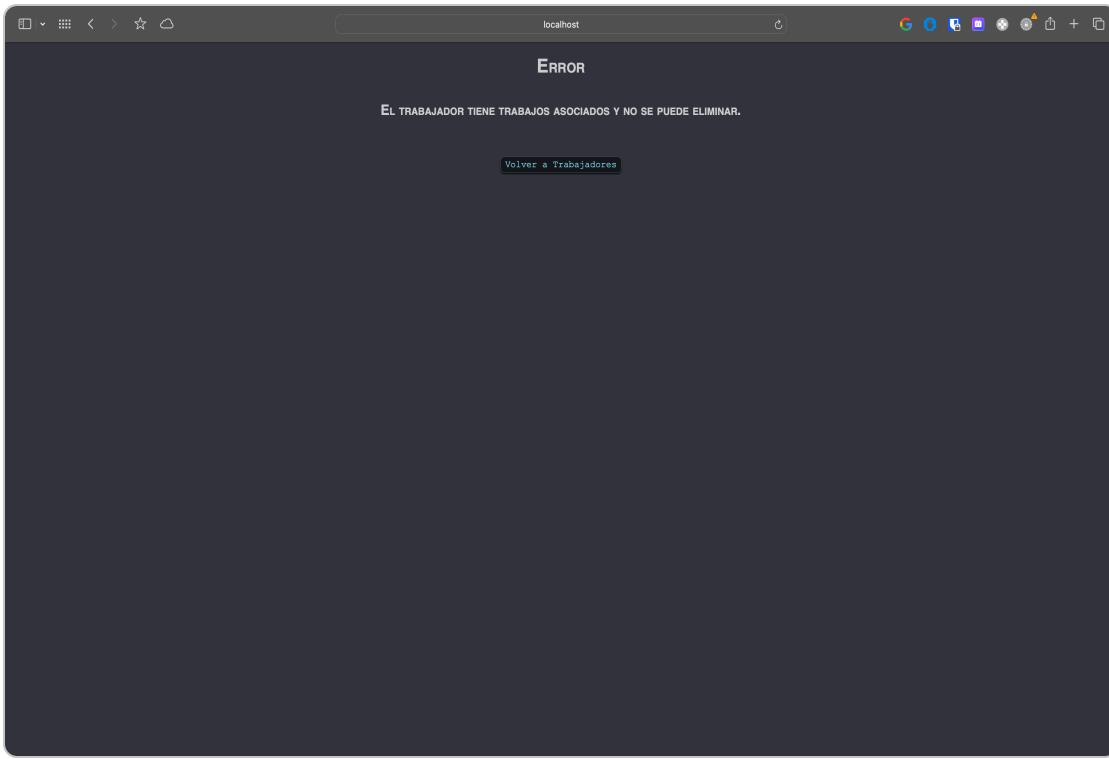
- ⚠** Deletes a worker from the database
Redirects to the 'ready' view with a success message
Workers with tasks assigned cannot be deleted

Successfully worker deleted



Delete success screenshot

Error deleting a worker



Delete error screenshot

Tutorial

How to enter the administration page

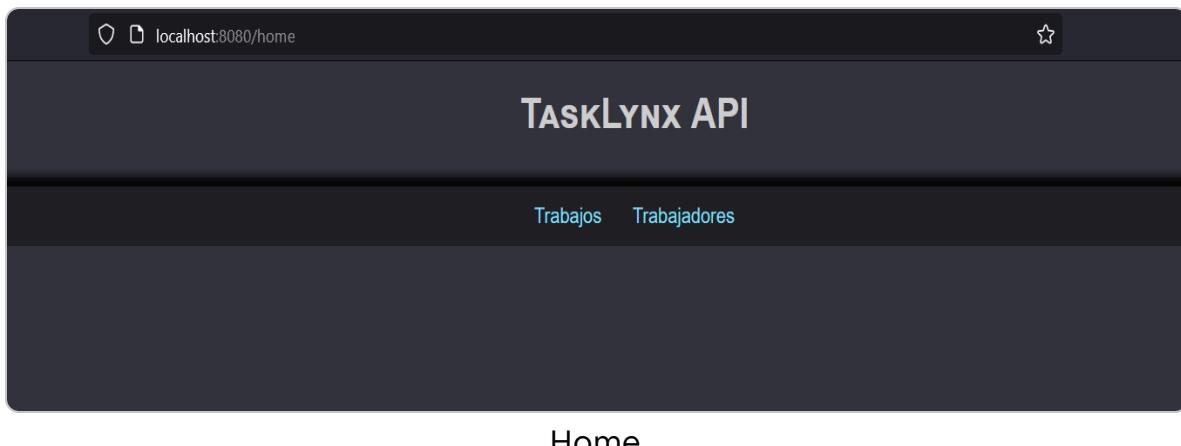
To enter the TaskLynx API administration page, you must access to the URL <http://localhost:8080> (<http://localhost:8080>).

It is also possible to access from:

- <http://localhost:8080/home> (<http://localhost:8080/home>)
- <http://localhost:8080/index> (<http://localhost:8080/index>)

No username or password is required.

The home page is the following:



Here you can navigate to the tasks and workers page.

Tasks

Index

The jobs page lists all the jobs that have been created in the application.

TRABAJOS

Añadir 1

ID	Categoría	Descripción	Fecha de inicio	Fecha de fin	Tiempo	Prioridad	Trabajador	Opciones
J001	Mecánica	Reparación de tuberías	2024-01-01	-	-	2	-	EDITAR 2 ELIMINAR
J002	Fontanería	Reparación de tuberías	2022-01-02	-	5.0 h	2	Ana	EDITAR 3 ELIMINAR
J003	Electricidad	Instalación de luces	2022-01-03	-	8.0 h	3	Carlos	EDITAR ELIMINAR
J004	Electricidad	Prueba de electricidad	2024-05-07	-	-	1	Carlos	EDITAR ELIMINAR
J005	Limpieza	Limpiar la habitación 25	2024-05-07	-	-	3	Carlos	EDITAR ELIMINAR
J006	Carpintería	Pintar puerta 204	2024-05-08	-	-	4	Pepe	EDITAR 5 ELIMINAR
J015	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	Ana	EDITAR ELIMINAR
J016	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	Pepe	EDITAR ELIMINAR
J017	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	-	EDITAR ELIMINAR

Volver 6

Tasks Index

From here, we can:

1. Create a task.
2. Edit a task.
3. Delete a task.
4. Go to the detailed view of a task by clicking its task code.
5. Go to a worker's detailed view by clicking on their name.
6. Go back to Home Page

Add

In this page, you can add a new task to the application.

The screenshot shows a web application window titled "Nuevo Trabajo". The main content area is a form titled "Datos del trabajo". The form contains the following fields:

- Código del trabajo *:** An input field.
- Categoría *:** An input field.
- Descripción *:** An input field.
- Fecha de inicio *:** A date input field with a placeholder "dd / mm / aaaa" and a calendar icon.
- Trabajador asignado:** An input field with a placeholder "-- No asignado --".
- Prioridad *:** An input field containing the value "1".

At the bottom of the form is a large blue button labeled "Crear trabajo". Below the form, there is a small "Volver" button.

Task Add

You must add the information marked as *, the other information is optional. This is an example of a new task.

NUEVO TRABAJO

Datos del trabajo

Código del trabajo *:

J007

Categoría *:

Limpieza

Descripción *:

Limpiar la habitación 120

Fecha de inicio *:

13 / 05 / 2024

Trabajador asignado:

Pepe García

Prioridad *:

2

Crear trabajo

Volver

Task Add with Data

If the information is correct, you will see a page with a confirmation.

TRABAJO AÑADIDO

EL TRABAJO HA SIDO AÑADIDO CORRECTAMENTE.

Volver a Trabajos

Task Add Correct

If not, you will see the errors in the add page.

The screenshot shows a web application interface for creating a new task. The title "NUEVO TRABAJO" is at the top. A red-bordered box labeled "ERRORES:" contains two error messages: "La categoría no puede estar vacía" and "La fecha de inicio no puede estar vacía". The main form fields include:

- Código del trabajo *:** J007
- Categoría *:** (empty input field)
- Descripción *:** Limpiar la habitación 120
- Fecha de inicio *:** (empty input field) with a calendar icon
- Trabajador asignado:** Pepe Garcia
- Prioridad *:** 2

A "Crear trabajo" button is at the bottom right. A "Volver" button is at the bottom center.

Task Add Errors

Detail

This page show the details of a specific task. You can access this page by clicking a task code on the Tasks index page.



Task Detail

From here, we can:

1. Go to a worker's detailed view by clicking on their name.
2. Edit a task.
3. Delete a task.
4. Go back to Tasks Index

Edit

On this page, you can edit the details of a task. You can access this page by clicking the "EDITAR" button on Tasks index page.

The screenshot shows a web application window with a dark background. At the top, the URL is `localhost:8080/trabajos/editar?codTrabajo=J004`. The main title is **EDITAR TRABAJO - J004**. Below it is a section titled **Datos del trabajo**. The form fields are as follows:

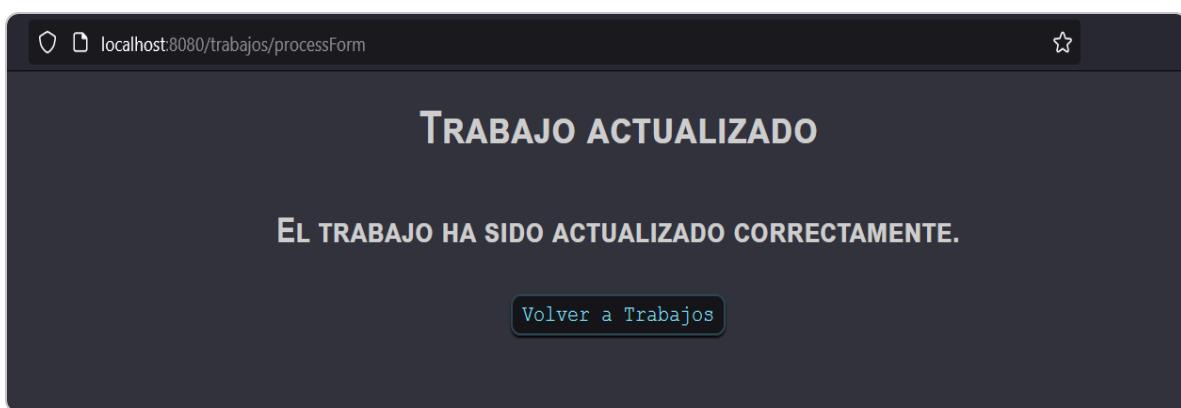
- Código del trabajo :** J004
- Categoría :** Electricidad
- Descripción :** Prueba de electricidad
- Fecha de inicio: Fecha actual 2024-05-07**
- Fecha de fin:** dd / mm / aaaa
- Tiempo en horas:** (empty input field)
- Trabajador asignado:** Carlos Martinez
- Prioridad :** 1

At the bottom right of the form is a button labeled **Editar trabajo**. At the very bottom center is a **Volver** button.

Task Edit

When you modify the information that you need, you can confirm the changes with the button "Editar trabajo".

If the information is correct, you will see a page with a confirmation.



Task Edit Correct

If not, you will see the errors in the edit page.

The screenshot shows a web browser window with the URL `localhost:8080/trabajos/processForm`. The title bar says "EDITAR TRABAJO - J004". Below the title, there is a red box labeled "ERRORES:" containing two error messages: "La categoría no puede estar vacía" and "La fecha de inicio no puede estar vacía". The main form fields are as follows:

- Datos del trabajo**
- Código del trabajo *:** J004
- Categoría *:** (empty field)
- Descripción *:** Prueba de electricidad
- Fecha de inicio:** Fecha actual 2024-05-13
- Fecha de fin:** dd / mm / aaaa
- Tiempo en horas:** (empty field)
- Trabajador asignado:** Carlos Martinez
- Prioridad *:** 1

At the bottom of the form are two buttons: "Editar trabajo" and "Volver".

Task Edit Errors

Delete

On this page, you can delete a task from the application. You can access this page by clicking the "ELIMINAR" button on Tasks index page.

You must confirm a window to process with task deletion.

The screenshot shows a dark-themed web application interface for managing tasks. At the top, the URL is `localhost:8080/trabajos`. Below the header, the title "TRABAJOS" is centered. A button labeled "Añadir" is located above a table. The table has columns: ID, Categoría, Descripción, Fecha de inicio, Fecha de fin, Tiempo, Prioridad, Trabajador, and Opciones. The rows represent tasks J001 through J017. Task J004 is currently selected. A modal dialog box is displayed over the table, containing the text "localhost:8080" and "¿Estás seguro de que quieres eliminar este trabajo? Esta acción no se podrá deshacer". It includes "Aceptar" and "Cancelar" buttons.

ID	Categoría	Descripción	Fecha de inicio	Fecha de fin	Tiempo	Prioridad	Trabajador	Opciones
J001	Mecánica	Reparación de tuberías	2024-01-01	-	-	2	-	<button>EDITAR</button> <button>ELIMINAR</button>
J002	Fontanería	Reparación de tuberías	2022-01-02	-	5.0 h	2	Ana	<button>EDITAR</button> <button>ELIMINAR</button>
J003	Electricidad	Instalación de luces	2022-01-03	-	8.0 h	3	Carlos	<button>EDITAR</button> <button>ELIMINAR</button>
J004	Electricidad	Prueba	localhost:8080	-	-	1	Carlos	<button>EDITAR</button> <button>ELIMINAR</button>
J005	Limpieza	Limpiar	-	-	-	3	Carlos	<button>EDITAR</button> <button>ELIMINAR</button>
J006	Carpintería	Pintar	-	-	-	4	Pepe	<button>EDITAR</button> <button>ELIMINAR</button>
J007	Limpieza	Limpiar la habitación 120	2024-05-13	-	-	2	Pepe	<button>EDITAR</button> <button>ELIMINAR</button>
J015	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	Ana	<button>EDITAR</button> <button>ELIMINAR</button>
J016	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	Pepe	<button>EDITAR</button> <button>ELIMINAR</button>
J017	Fontanería	Reparación de tuberías	2024-01-01	-	-	4	-	<button>EDITAR</button> <button>ELIMINAR</button>

[Volver](#)

Task Delete Confirm

Then, you will see a page with a confirmation.

The screenshot shows a dark-themed web application interface. The title "TRABAJO ELIMINADO" is centered at the top. Below it, the message "EL TRABAJO HA SIDO ELIMINADO CORRECTAMENTE." is displayed. A button labeled "Volver a Trabajos" is located at the bottom.

Task Delete Result

Workers

En la vista de trabajadores se listan todos los trabajadores presentes en la aplicación. Es una vista muy similar a la de trabajos, en la que podemos hacer las siguientes acciones:

1. Crear un trabajador.
2. Editar un trabajador.
3. Eliminar un trabajador.
4. Ir a la vista detalle de un trabajador haciendo click en su ID.
5. Ir a la vista detalle de un trabajo haciendo click en su código de trabajo.

ID	Dni	Nombre	Apellidos	Especialidad	Email	Trabajos asociados	Opciones
T003	34567890C	Carlos	Martinez	Electricidad	carlos.martinez@example.com	J003	<button>EDITAR</button> <button>ELIMINAR</button>
T001	12345678A	Pepe	Gomez	Fontaneria	pepe@gomez.com	J001 J002	<button>EDITAR</button> <button>ELIMINAR</button>
T002	12341234R	Ana	Gómez	Carpintería	ana@gomez.com	-	<button>EDITAR</button> <button>ELIMINAR</button>

trabajadores-table.png

Business – Index



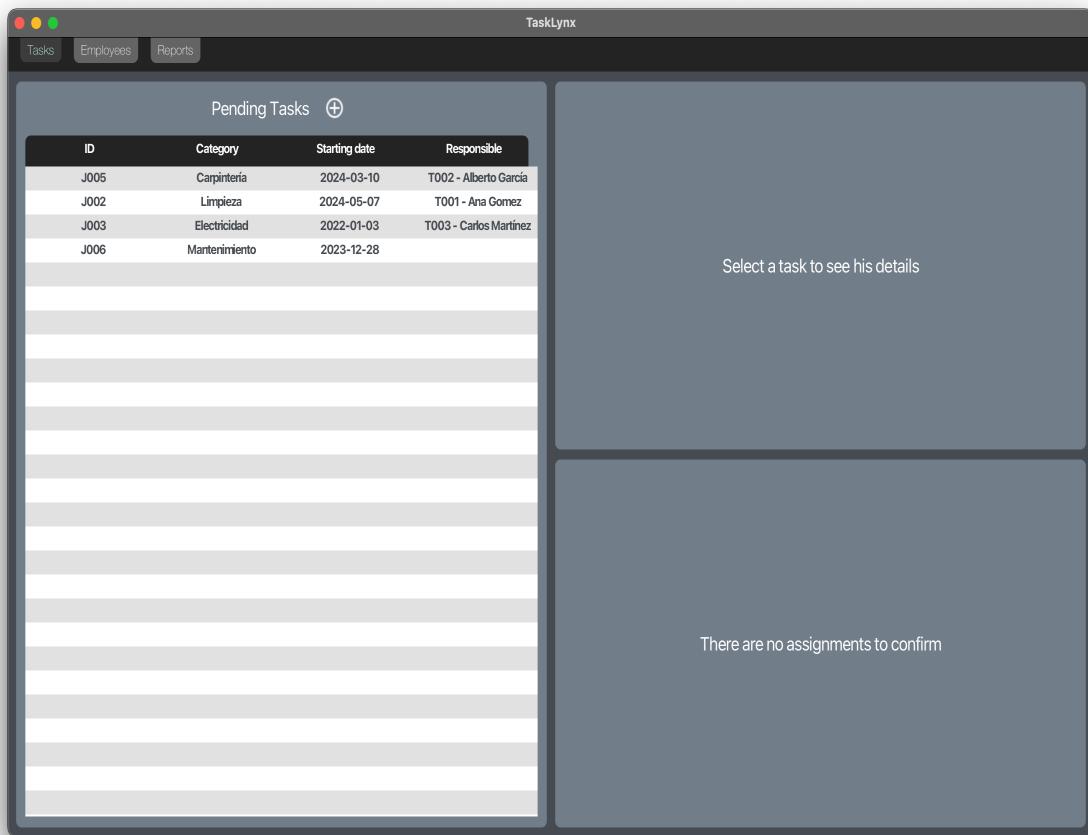
TaskLynx-logo-business-oval.svg

Overview

TaskLynx Business is the comprehensive desktop application for the TaskLynx system, designed to efficiently manage hotel tasks. This application allows administrators to handle various system functions centrally, improving the organization and productivity of the establishment.

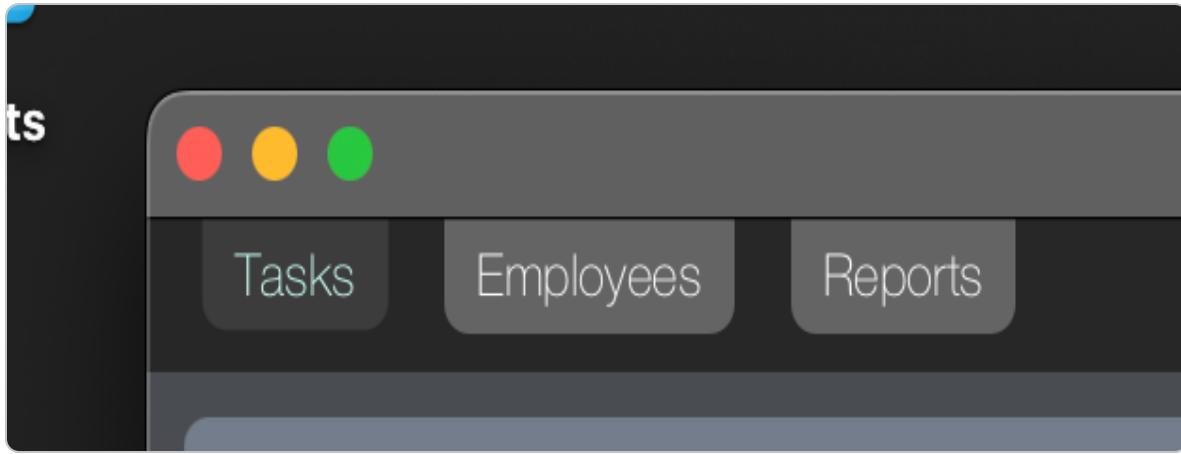
Task Management

At the first time into TaskLynx Business, users are presented with a list of pending tasks.



01-Tareas_ventana_ppal.png

Navigation through different tabs is facilitated by the top menu.



02-Navegacion_pestañas.png

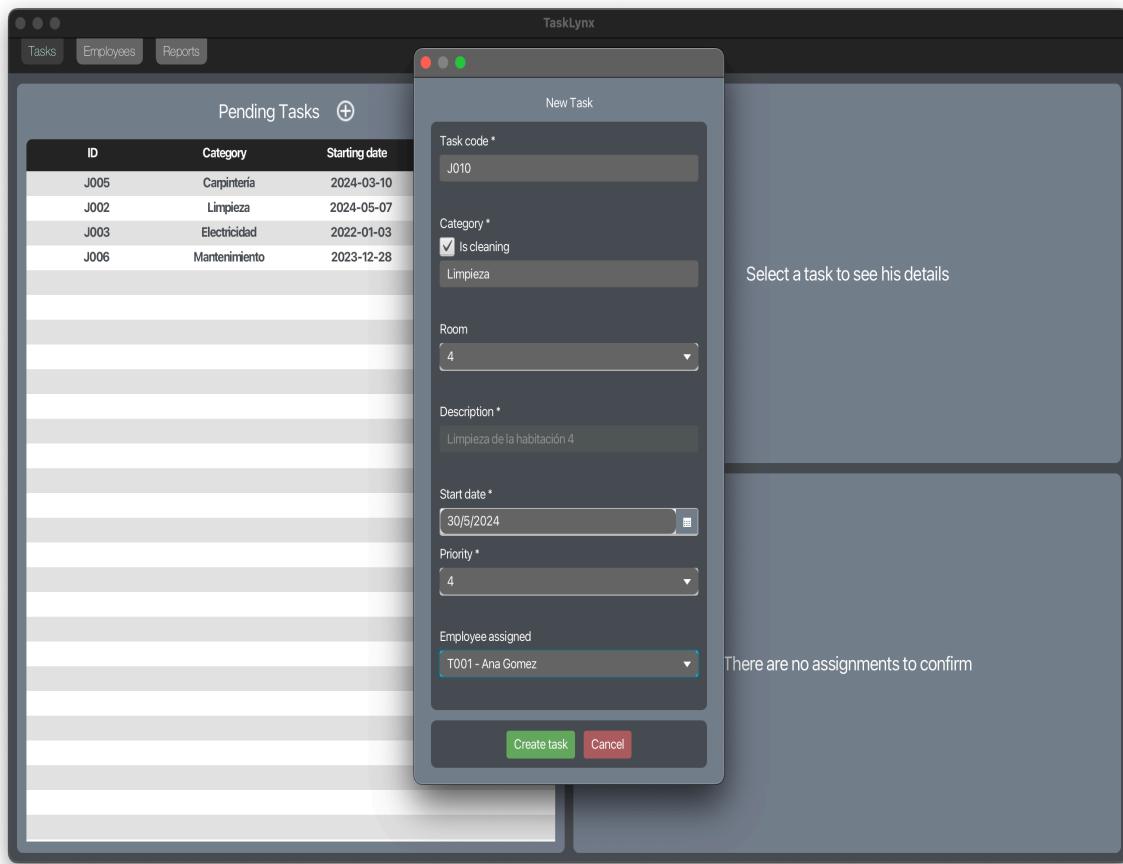
The task list displays all pending tasks, and clicking on a task reveals its details along with various possible actions.

A screenshot of the TaskLynx application interface. On the left, a sidebar shows the navigation tabs: "Tasks", "Employees", and "Reports". The main area is titled "Pending Tasks" and contains a table with six rows of task data. The sixth row, which corresponds to task J006, is highlighted with a blue background. To the right of the table, a detailed view for task J006 is displayed. The title of this view is "Mantenimiento - Detail". It includes buttons for "Edit task", "Delete task", and "Assign employee". Below the buttons, the task's description is listed as "Reparación de aire acondicionado", and its starting date is "2023-12-28". The responsible person is listed as "T003 - Carlos Martinez". The priority is listed as "Priority: 4". At the bottom of the detail view, a message states "There are no assignments to confirm".

ID	Category	Starting date	Responsible
J010	Limpieza	2024-05-30	T001 - Ana Gomez
J005	Carpintería	2024-03-10	T002 - Alberto García
J002	Limpieza	2024-05-07	T001 - Ana Gomez
J003	Electricidad	2022-01-03	T003 - Carlos Martinez
J006	Mantenimiento	2023-12-28	

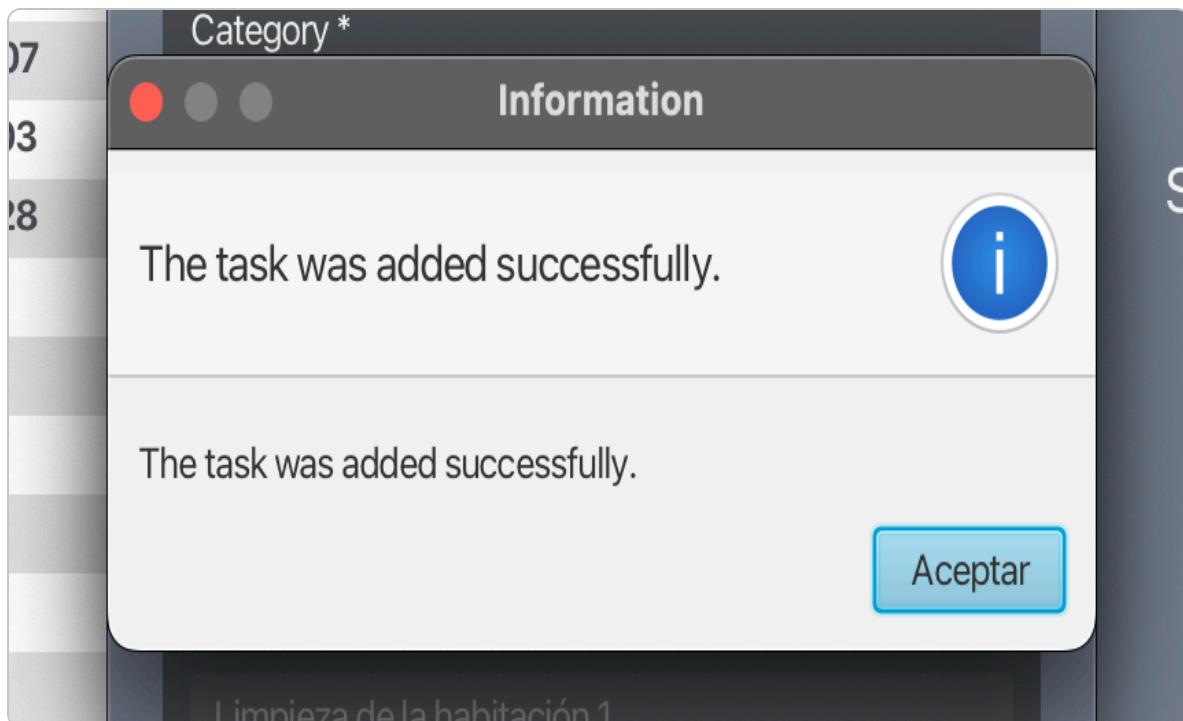
05-Detalles_tarea_sin_asignar.png

Add Task: A "+" button above the table allows adding a new task to the system. Clicking this opens a modal window requesting the task details.



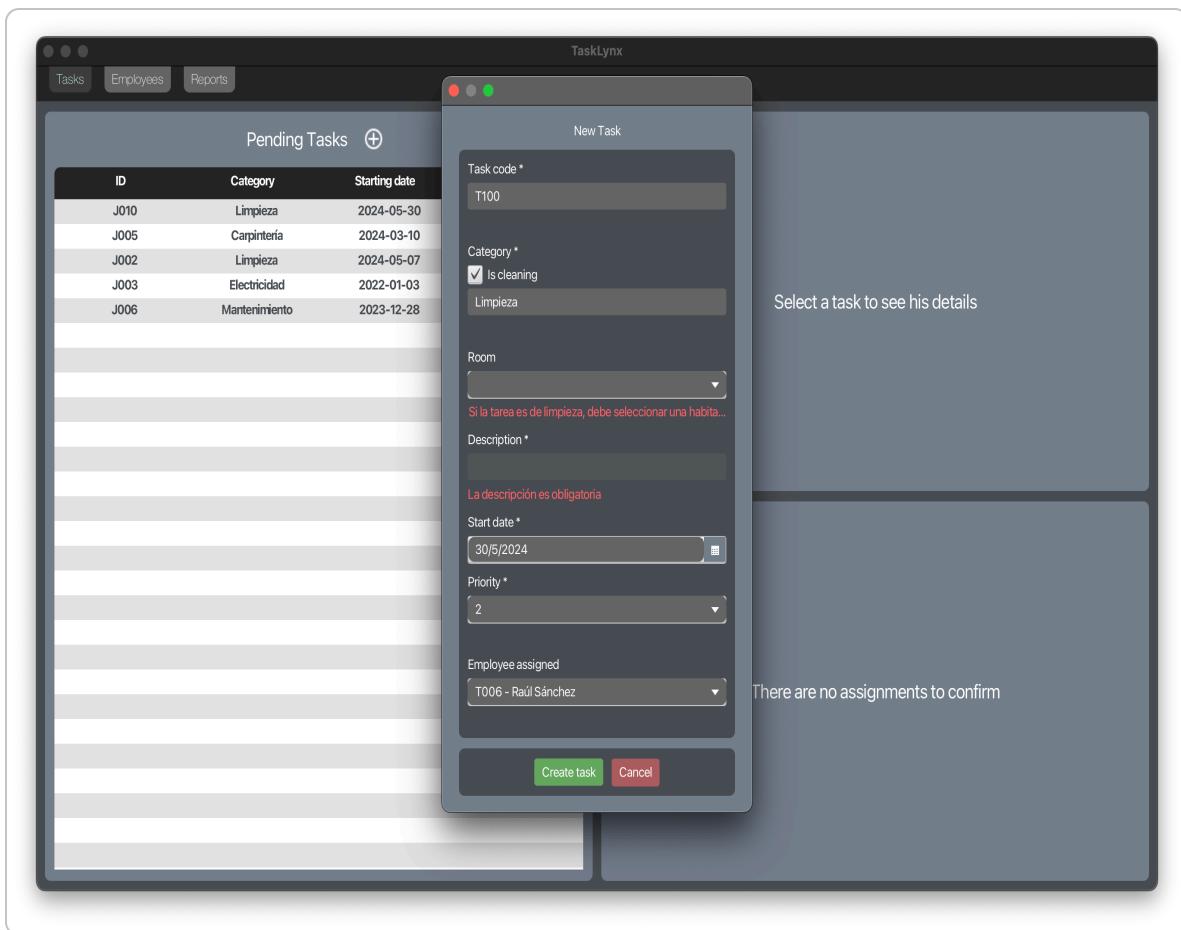
03-Añadir_tarea.png

Once correctly entered, a window confirms the successful creation of the task.



03_3-Añadir_tarea_modal.png

If there is an error in any field, a corresponding error message is displayed.



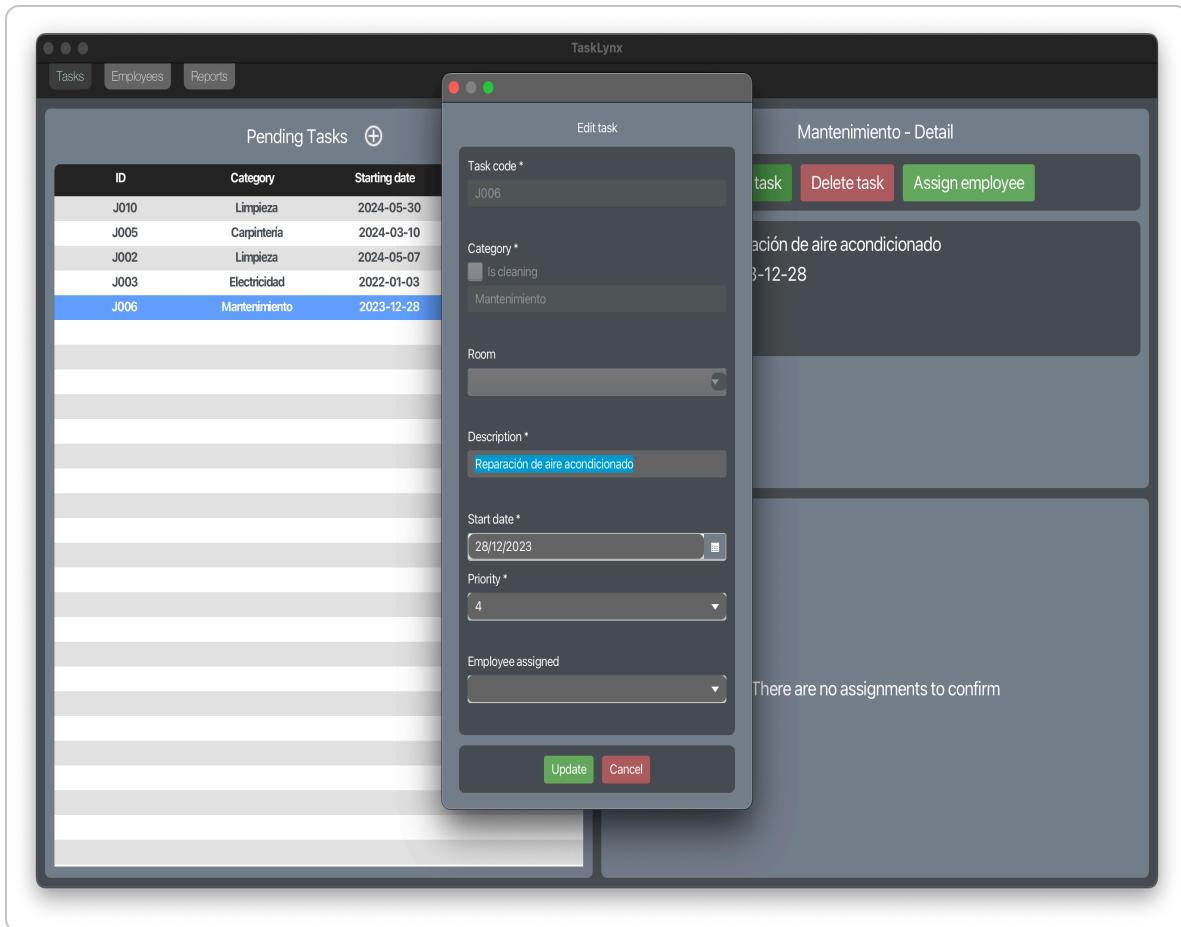
03_2-Añadir_tarea_errores.png

The new task is immediately reflected in the table.

ID	Category	Starting date	Responsible
J010	Limpieza	2024-05-30	T001 - Ana Gomez
J005	Carpintería	2024-03-10	T002 - Alberto García

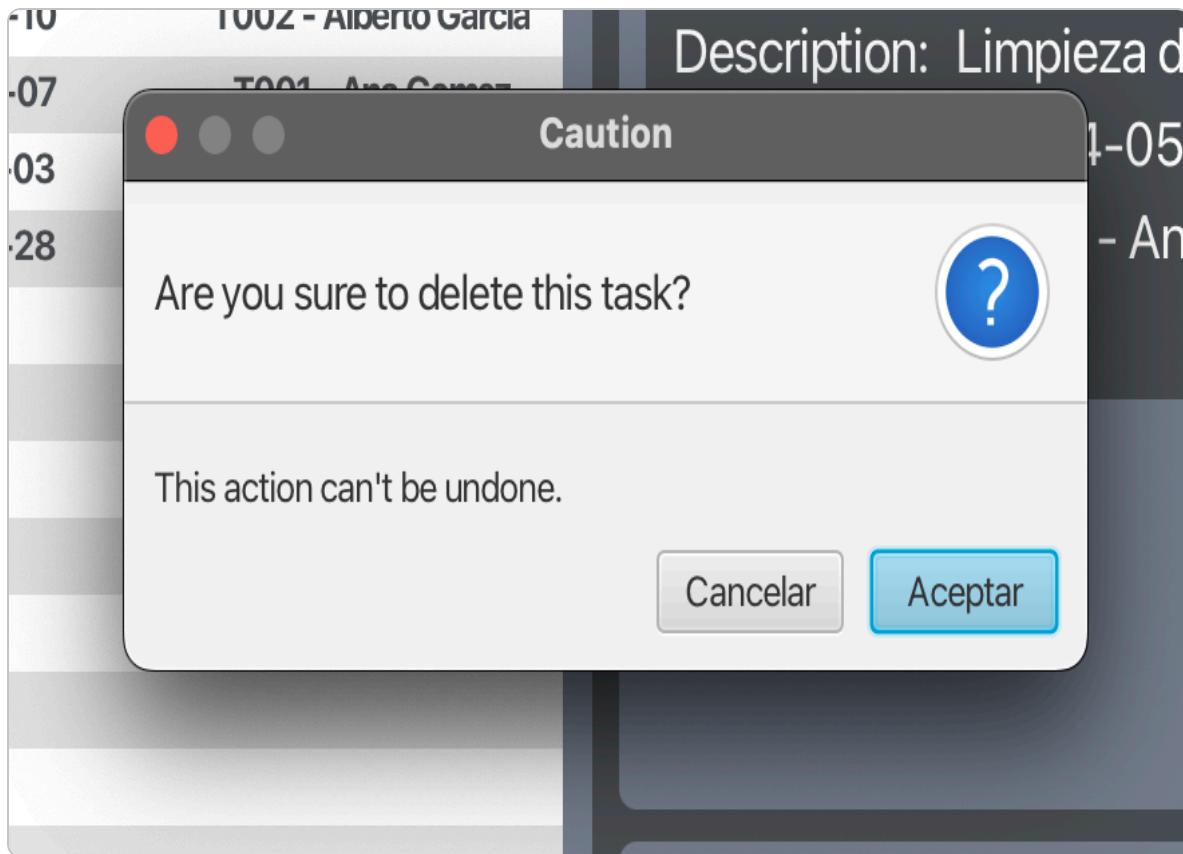
04-Tarea_añadida.png

Edit and Delete Tasks: Existing tasks can be edited via a modal window that allows updating the desired details.



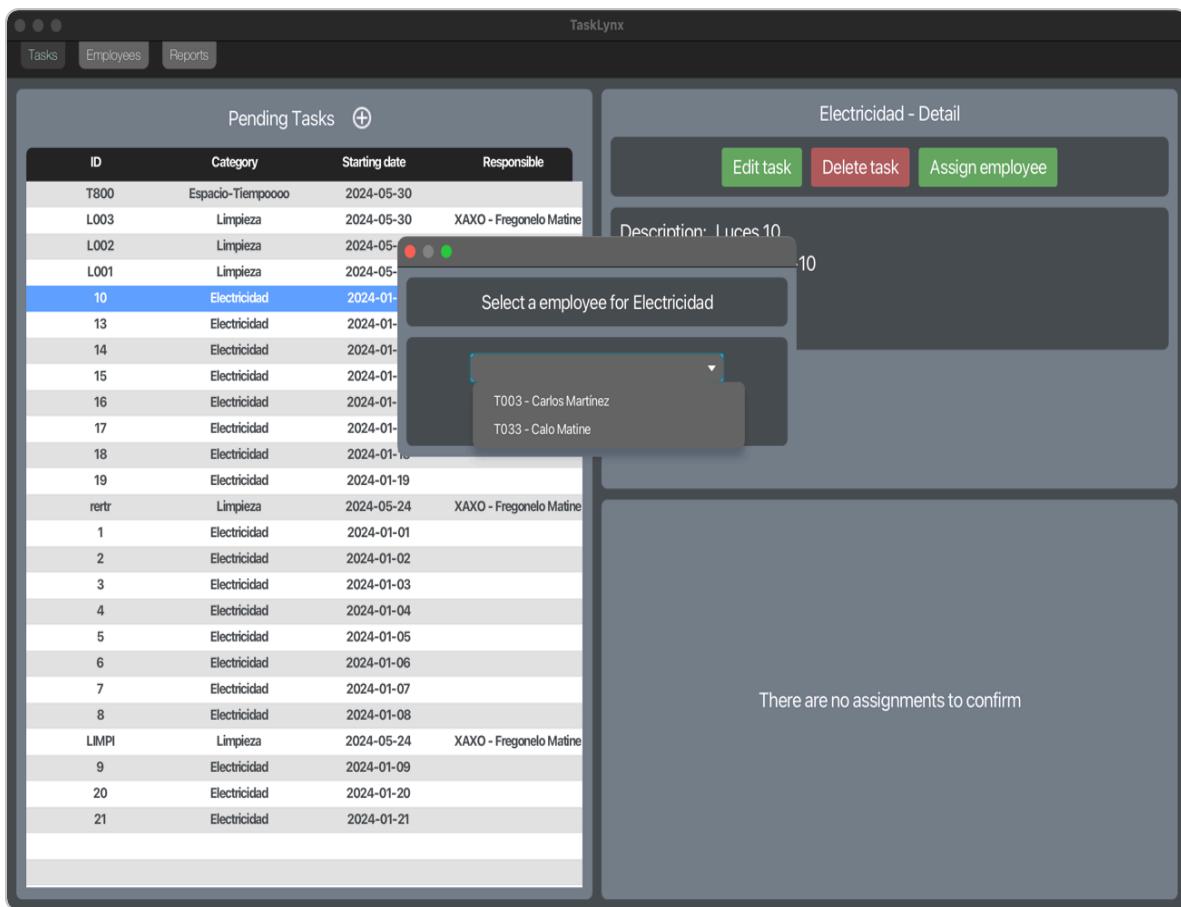
06-Editar_tarea.png

Tasks can also be deleted, following confirmation through a system message.



07-Eliminar_tarea.png

Employee Assignment: Each task can be assigned to a specific employee, and these assignments can be managed and accepted later. This can be done by selecting a task and pressing the Assign employee or by double-clicking on it.



08_2-Asignar_empleado.png

If there is a provisional employee assigned to a task, and we double-click on it again, it will be unassigned.

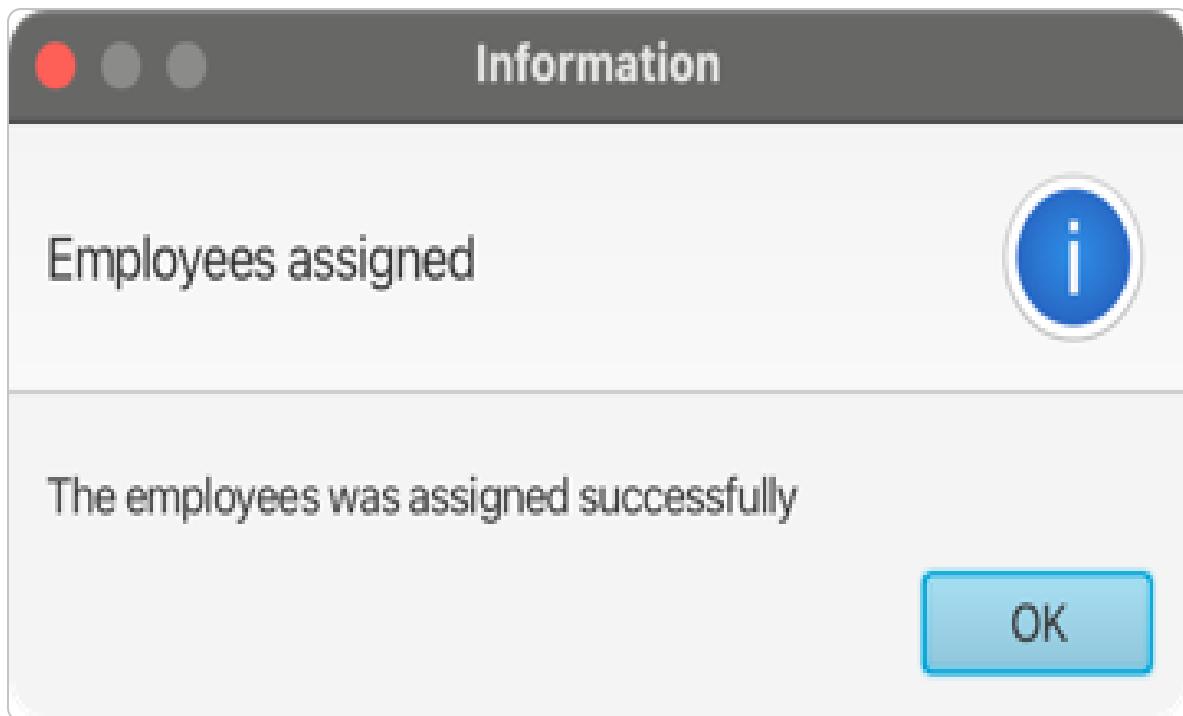
The screenshot shows the TaskLynx application interface. On the left, the 'Pending Tasks' pane displays a list of tasks with columns: ID, Category, Starting date, and Responsible. A task with ID 14 and Category Electricidad is selected and highlighted in blue. On the right, the 'Electricidad - Detail' pane shows details for this selected task: Description: Luces 14, Starting date: 2024-01-14, Responsible: T003 - Carlos Martinez, and Priority: 4. Below these panes, the 'Assigned Tasks' pane lists tasks assigned to the responsible employee, with a 'Confirm tasks' button at the top.

ID	Category	Starting date	Responsible
T800	Espacio-Tiempo000	2024-05-30	XAXO - Fregonelo Matine
L003	Limpieza	2024-05-30	XAXO - Fregonelo Matine
L002	Limpieza	2024-05-30	XAXO - Fregonelo Matine
L001	Limpieza	2024-05-24	XAXO - Fregonelo Matine
10	Electricidad	2024-01-10	T003 - Carlos Martinez
13	Electricidad	2024-01-13	T033 - Calo Matine
14	Electricidad	2024-01-14	T033 - Calo Matine
15	Electricidad	2024-01-15	
16	Electricidad	2024-01-16	
17	Electricidad	2024-01-17	
18	Electricidad	2024-01-18	
19	Electricidad	2024-01-19	
retr	Limpieza	2024-05-24	XAXO - Fregonelo Matine
1	Electricidad	2024-01-01	
2	Electricidad	2024-01-02	
3	Electricidad	2024-01-03	
4	Electricidad	2024-01-04	
5	Electricidad	2024-01-05	
6	Electricidad	2024-01-06	
7	Electricidad	2024-01-07	
8	Electricidad	2024-01-08	
LIMPI	Limpieza	2024-05-24	XAXO - Fregonelo Matine
9	Electricidad	2024-01-09	
20	Electricidad	2024-01-20	
21	Electricidad	2024-01-21	

ID	Category	Starting date	Responsible
10	Electricidad	2024-01-10	T003 - Carlos Martinez
14	Electricidad	2024-01-14	T033 - Calo Matine

08_03-Asignar_empleado.png

By pressing **Confirm tasks**, if the operation is successful, a confirmation message is displayed and every employee will receive an email with a list of their assigned tasks sorted by date.



08_4-Asignar_empleado.png

Hello, Calo!

You have new upcoming tasks!

- Task ID: 14
 - Description: Luces 14
 - Starting Date: 2024-01-14
 - Priority: 4

Be a Lynx, don't miss a task!

© 2024 TaskLynx.

08_5-Asignar_empleado.png

Employee Management

The employee tab functions similarly to the task tab, offering a table with employees and their characteristics.

The screenshot shows the TaskLynx application interface. At the top, there is a dark header bar with three colored window control buttons (red, yellow, green) on the left and the title "TaskLynx" on the right. Below the header is a navigation bar with three tabs: "Tasks" (highlighted in blue), "Employees" (highlighted in red), and "Reports". The main content area is divided into two sections. On the left, there is a table titled "Employees" with a plus sign icon. The table has four columns: "ID", "Name", "Surname", and "Specialty". The data in the table is as follows:

ID	Name	Surname	Specialty
T001	Ana	Gomez	Limpieza
T002	Alberto	García	Carpintería
T003	Carlos	Martinez	Electricidad
T004	Sonia	Pérez	Electricidad
T005	Juan	Soler	Mantenimiento
T006	Raúl	Sánchez	Limpieza

The right section of the main content area is currently empty and displays the placeholder text "Select an employee to see his details".

01-Empleados_Main.png

Clicking on an employee shows their specific details.

The screenshot shows the TaskLynx application interface. On the left, there is a sidebar with three tabs: 'Tasks' (selected), 'Employees', and 'Reports'. Below the tabs is a search bar with placeholder text 'Search tasks or employees'. The main area is divided into two sections: 'Employees' on the left and 'Pending Tasks' on the right.

Employees Section:

- Table Headers:** ID, Name, Surname, Speciality
- Data Rows:**

T001	Ana	Gomez	Limpieza
T002	Alberto	Garcia	Carpinteria
T003	Carlos	Martinez	Electricidad
T004	Sonia	Perez	Electricidad
T005	Juan	Soler	Mantenimiento
T006	Raul	Sanchez	Limpieza
T007	Jose Luis	Estevez	Mantenimiento

Pending Tasks Section:

- Table Headers:** ID, Category, Starting date, Priority
- Data Rows:**

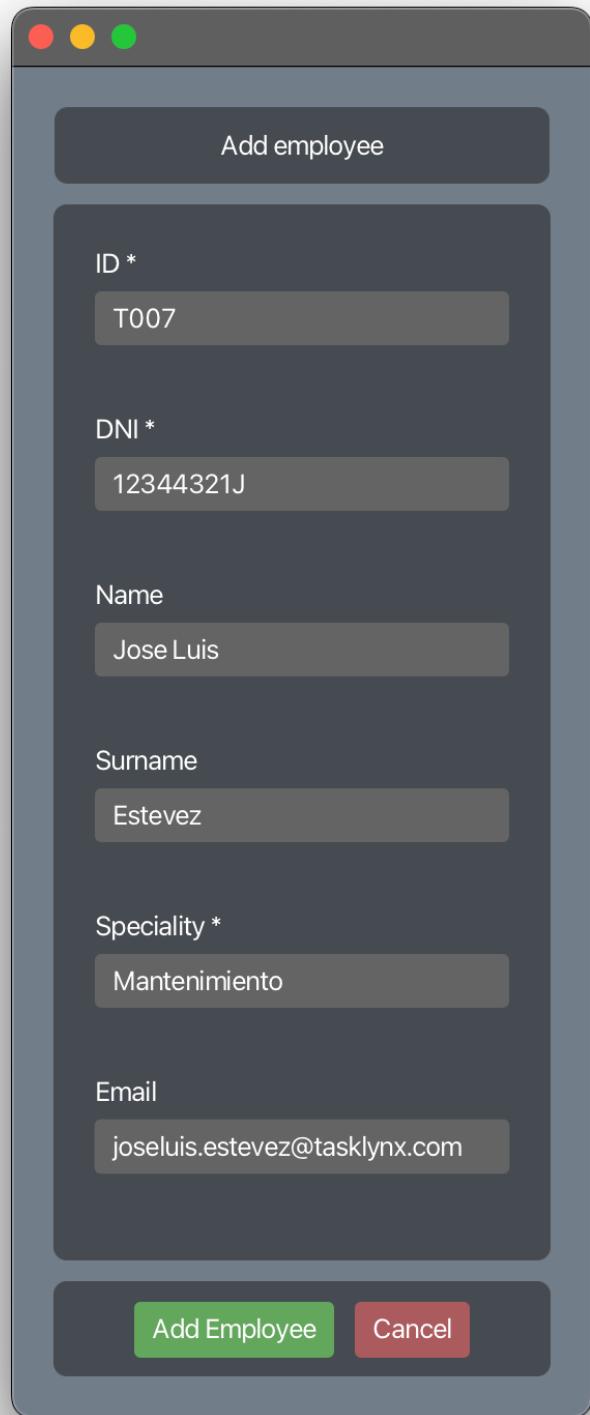
J002	Limpieza	2024-05-07	2
J010	Limpieza	2024-05-30	4

Employee Detail View (Ana Gomez - Detail):

- Buttons:** Edit employee (green), Delete employee (red)
- Information:**
 - Name: Ana
 - Surname: Gomez
 - DNI: 12345678B
 - Speciality: Limpieza
 - Email: ana.gomez@tasklynx.com

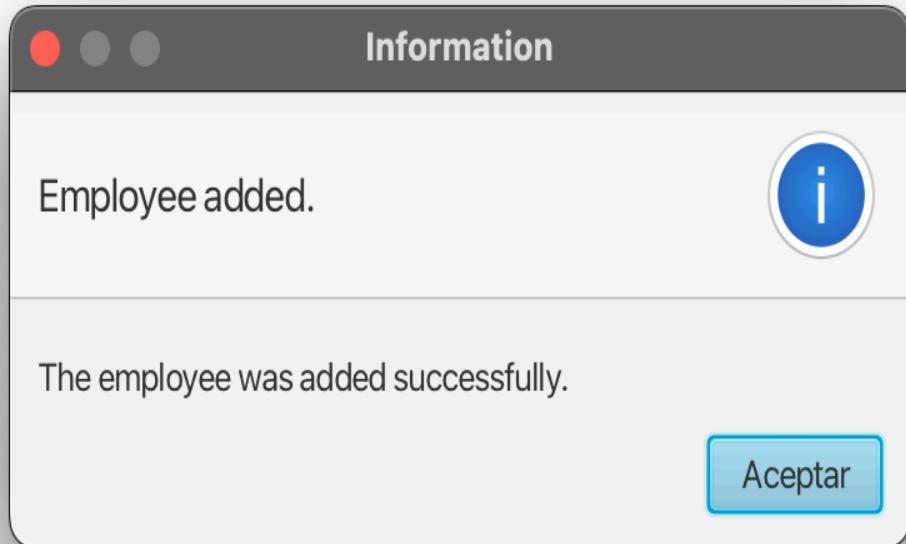
01_2-Employados_General_con_tareas_pendientes.png

Add Employee: The "+" button allows adding a new employee, opening a modal window to enter the necessary details.



02-Añadir_empleados.png

Successful insertion is confirmed



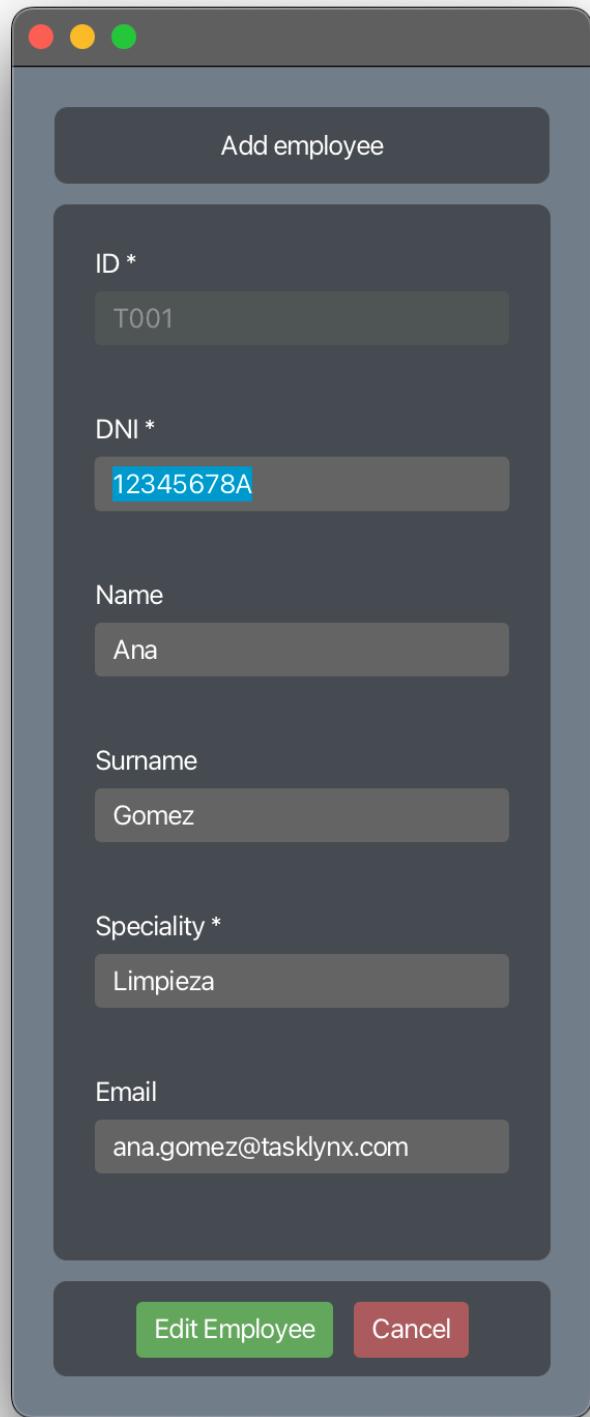
03-Modal_añadir_empleado.png

and the new employee appears in the table.

T007	Jose Luis	Estevez	Mantenimiento
------	-----------	---------	---------------

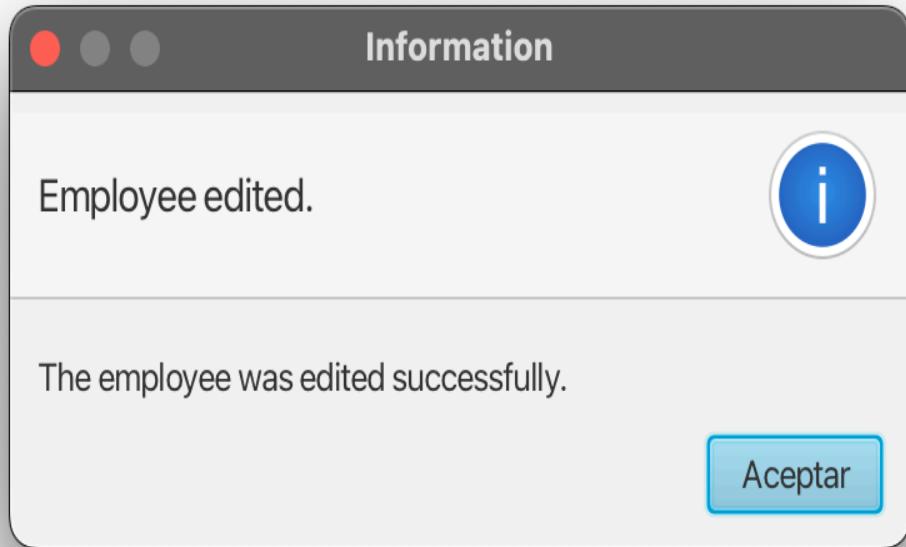
04-Empleado_añadido.png

Edit and Delete Employees: Employee details can be edited via a modal window



05-Editar_empleado.png

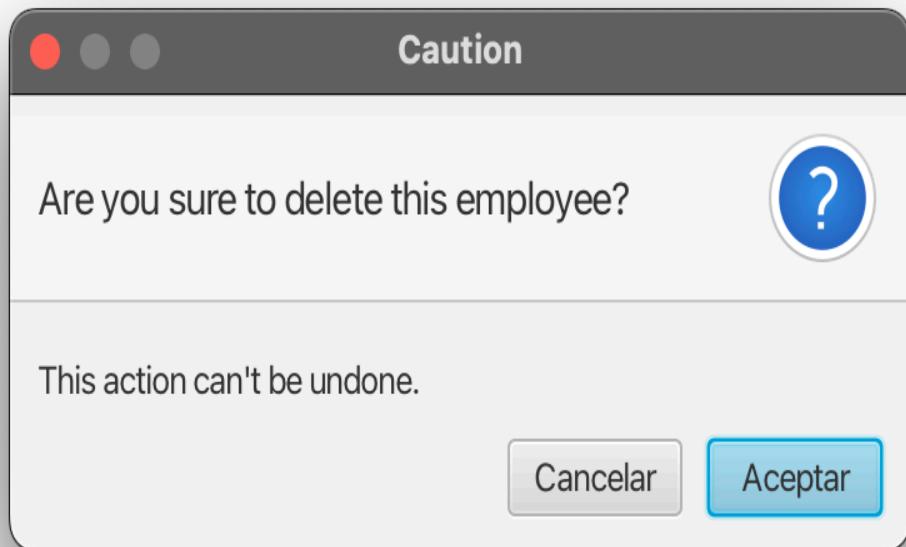
and any successful change is notified by the system



06-Modal_empleado_editado.png

⚠ It will not be possible to delete an employee if they have pending tasks.

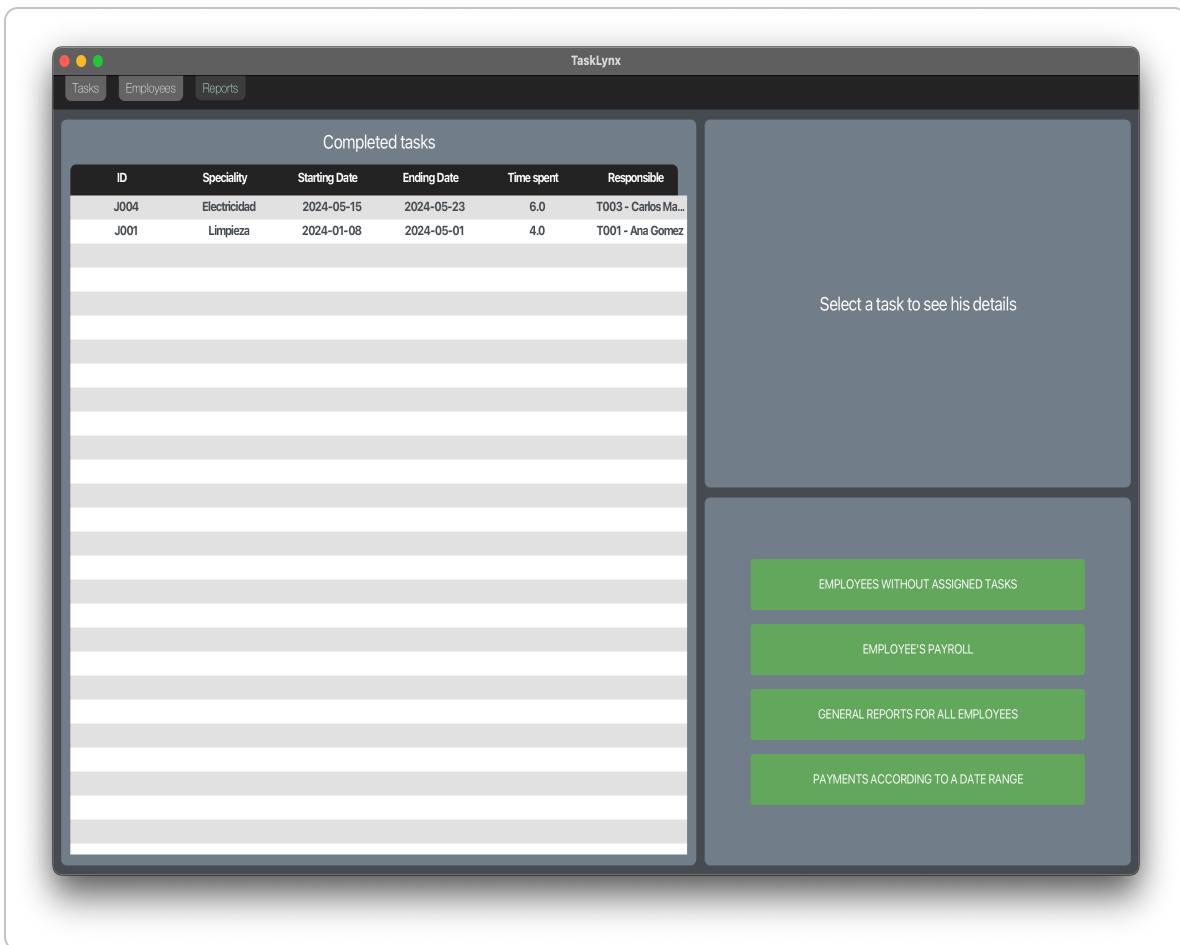
Employees can also be deleted after system confirmation.



07-Modal_eliminar_empleado.png

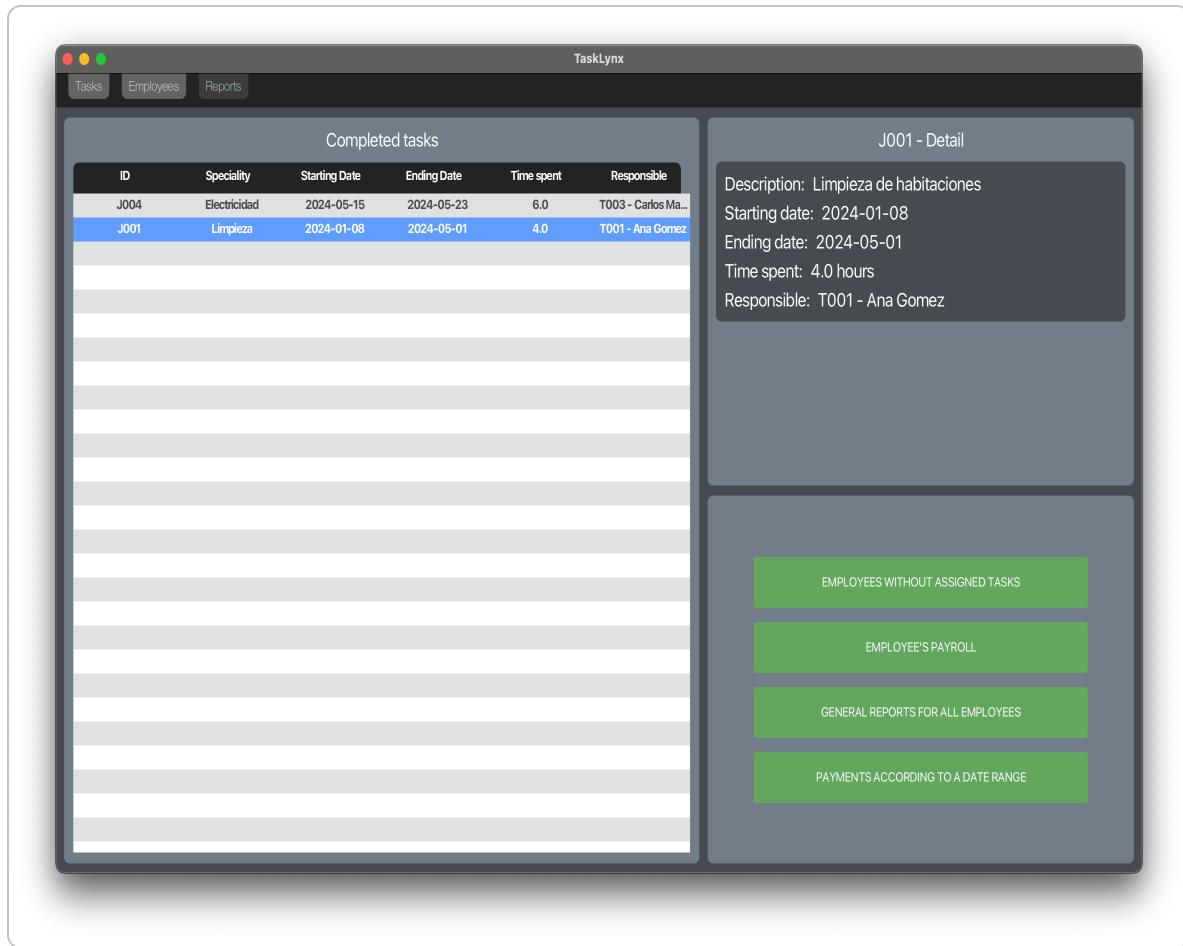
Report Generation

The report tab displays a table with a list of completed tasks.



01-Informes_main.png

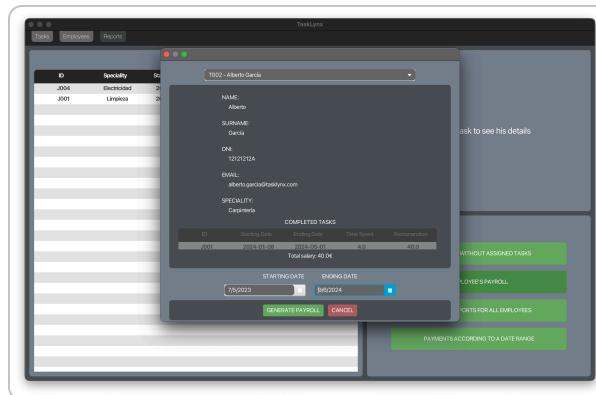
Clicking on a task reveals its details, and various reports can be generated using the buttons located to the right of the window.



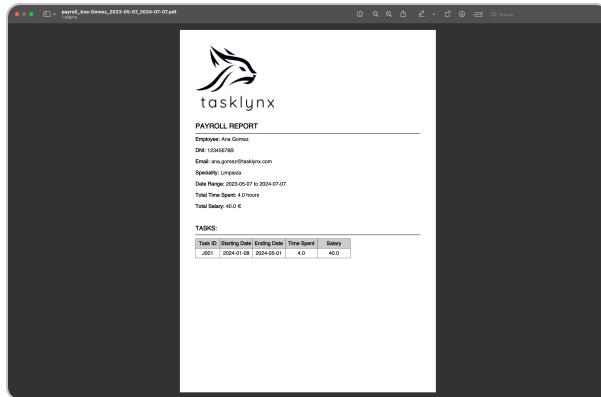
01_2-Informes_con_detalle.png

Available Reports

- Payroll reports can be generated by selecting an employee and a specific date range.

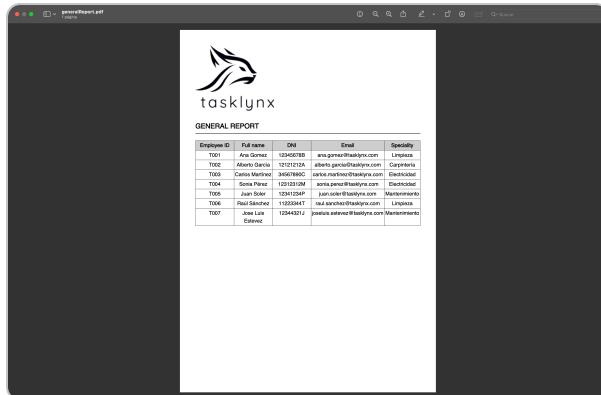


03-Modal_generar_nomina.png



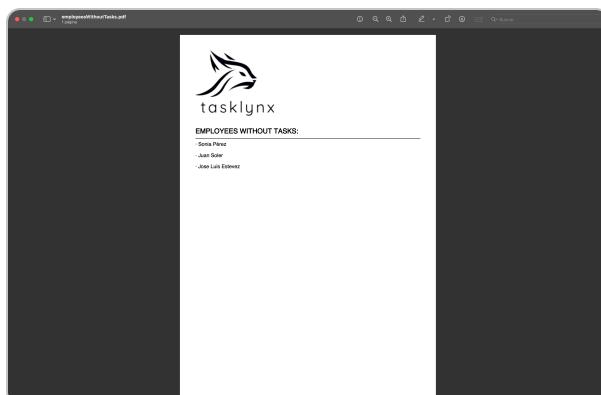
06-Nomina.png

- General reports for all employees



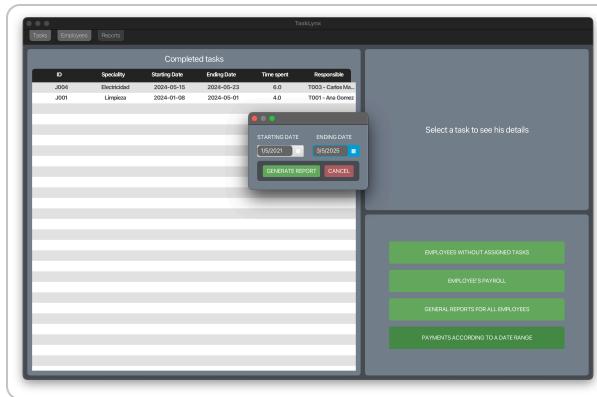
07-Informe_general.png

- Lists of employees without assigned tasks

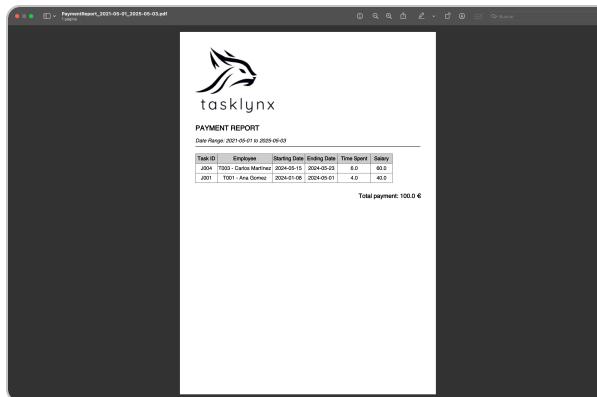


08-Empleados_sin_tareas.png

- Payment reports for workers between two specific dates can also be generated.



04-Modal_pago_por_fechas.png



05-Informe_de_pagos.png

Conclusion

TaskLynx Business provides a robust and efficient solution for managing tasks and employees in hotels. With an intuitive interface and comprehensive functions, it facilitates organization and improves productivity, ensuring smoother and more effective hotel operations.

Mobile – Index



TskLynx-mobile-oval-logo.svg

Nest Hotel – Index



Nest Hotel logo

Introduction

En este documento se detalla el despliegue de los proyectos Nest Hotel (<https://github.com/DanielAlmazan/hotel-nest>) mediante Docker y Gestión Hotelera mediante Apache2 + Passenger.

Este despliegue se ha realizado en un servidor VPS con Debian 11.

Despliegue de Gestión Hotelera



Apache 2 logo



Passenger logo

En nuestro VPS con Debian 11, instalamos Apache2

```
sudo apt-get install apache2
```

Configuración básica

Lo primero, por seguridad, nos hacemos una copia del fichero `apache2.conf`, aunque en principio no lo vamos a modificar.

```
sudo cp /etc/apache2/apache2.conf  
/etc/apache2/apache2.conf.original
```

```
sudo nano /etc/apache2/ports.conf
```

```
# If you just change the port or add more ports here, you will  
likely also  
# have to change the VirtualHost statement in  
# /etc/apache2/sites-enabled/000-default.conf
```

```
Listen 80  
Listen 8000  
Listen 8080
```

```
Listen 8088
Listen 8888

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Instalación de Passenger

Para instalar Passenger en el servidor (Debian, en este caso), ejecutamos los siguientes comandos:

```
sudo apt-get install -y dirmngr gnupg

sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80
--recv-keys 561F9B9CAC40B2F7

sudo apt-get install -y apt-transport-https ca-certificates

sudo sh -c 'echo deb https://oss-binaries.phusionpassenger.com/
apt/passenger bullseye main >
/etc/apt/sources.list.d/passenger.list'

sudo apt-get update

sudo apt-get install -y libapache2-mod-passenger
```

Habilitamos el módulo de Passenger

```
sudo a2enmod passenger
```

Seguidamente, reiniciamos Apache2

```
sudo systemctl restart apache2
```

Vamos a comprobar que el módulo de Passenger se ha instalado correctamente

```
sudo /usr/bin/passenger-config validate-install
```

```
[sudo] password for {user}:
What would you like to validate?
Use <space> to select.
If the menu doesn't display correctly, press '!'
-----
```

- Passenger itself
 - ● Apache
- ```

```

```
Checking whether there are multiple Apache installations...
Only a single installation detected. This is good.
```

```

```

```
* Checking whether this Passenger install is in PATH... ✓
* Checking whether there are no other Passenger installations...
✓
* Checking whether Apache is installed... ✓
* Checking whether the Passenger module is correctly configured
in Apache... ✓
```

Everything looks good. :-)

- i** Es posible que, dependiendo de la configuración de vuestra consola, no se muestre correctamente el menú de opciones. En ese caso, pulsad ! para que se

muestre correctamente.

Creamos el fichero de configuración de nuestro sitio web

```
sudo nano /etc/apache2/sites-available/003-gestion-hotelera.conf
```

```
<VirtualHost *:8080>
 ServerAdmin {email}
 ServerName amorenoborra.es
 DocumentRoot "/home/despliegue/Gestion-Hotelera"
 PassengerAppRoot "/home/despliegue/Gestion-Hotelera"
 PassengerAppType node
 PassengerStartupFile index.js
 <Directory "/home/despliegue/Gestion-Hotelera">
 Allow from all
 Options -MultiViews
 Require all granted
 </Directory>
</VirtualHost>
```

Indicamos que escuche en el puerto 8080, que es el que hemos configurado en ports.conf, que el directorio raíz de nuestro sitio web es /home/despliegue/Gestion-Hotelera, que el tipo de aplicación es node, que el fichero de inicio es index.js y que el directorio /home/despliegue/Gestion-Hotelera tiene permisos de lectura para todos los usuarios.

## Habilitar el sitio

Creamos un enlace simbólico en sites-enabled para habilitar el sitio

```
sudo ln -s /etc/apache2/sites-available/003-gestion-hotelera.conf
\
/etc/apache2/sites-enabled/003-gestion-hotelera.conf
```

En caso de querer deshabilitar el sitio, bastaría con borrar el enlace simbólico.

## Recargar Apache2

```
sudo systemctl reload apache2
```

## Instalación de NodeJS

Ahora vamos a instalar NodeJS en nuestro servidor Debian 11

```
sudo apt-get install curl
curl -sL https://deb.nodesource.com/setup_20.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Comprobamos la versión de NodeJS y de NPM

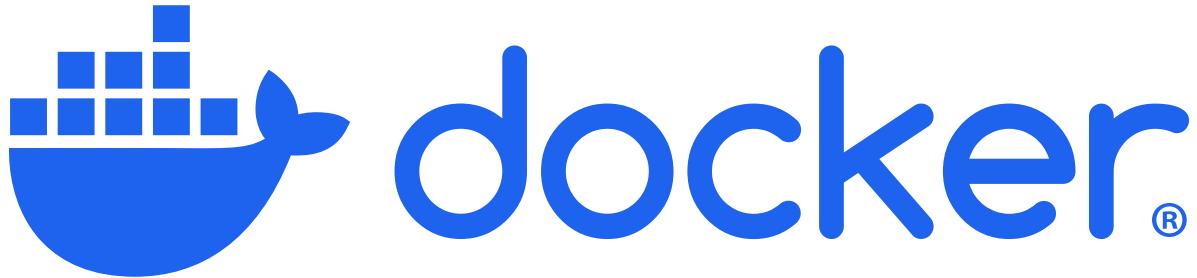
```
node -v
```

```
v20.0.0
```

```
npm -v
```

```
10.3.0
```

# Despliegue de Hotel Nest



Docker logo

## Instalación de Docker

Para instalar Docker en el servidor (Debian, en este caso), ejecutamos los siguientes comandos:

```
for pkg in docker.io docker-doc docker-compose podman-docker \
containerd runc; do sudo apt-get remove $pkg; done
```

## Crear red de Docker

Para crear una red de Docker, ejecutamos el siguiente comando:

```
sudo docker network create node_mongo
```

Comprobamos que se ha creado correctamente con el siguiente comando:

```
sudo docker network ls
```

| NETWORK ID   | NAME       | DRIVER | SCOPE |
|--------------|------------|--------|-------|
| d167b3cb6ed3 | bridge     | bridge | local |
| d407dbcf94cc | host       | host   | local |
| 885425ece5f9 | node_mongo | bridge | local |
| fa4b3a84c89b | none       | null   | local |

## Crear contenedor de MongoDB

```
sudo docker run -d --name hotelDB --network node_mongo --restart \
unless-stopped -p 127.0.0.1:27017:27017 -v ~/data:/data/db mongo
```

Con este comando, creamos un contenedor de MongoDB llamado `hotelDB` que se conecta a la red `node_mongo`, se reinicia automáticamente si se detiene y se mapea el puerto 27017 del contenedor al puerto 27017 del servidor. Además, se monta el directorio `~/data` del servidor en el directorio `/data/db` del contenedor.

## Crear el fichero Dockerfile

```
nano Dockerfile
```

```
Imagen base
FROM node:latest

Crear directorio de la aplicación
RUN mkdir -p /usr/src/app

Establecer directorio de trabajo
WORKDIR /usr/src/app

Copiar los ficheros package.json y package-lock.json
COPY package*json ./
Instalar dependencias
RUN npm install

Copiar el resto de ficheros
```

```
COPY . .

Exponer el puerto 3000
EXPOSE 3000

Comando de inicio
CMD ["npm", "start"]
```

## Crear el fichero .dockerignore

```
nano .dockerignore
```

```
node_modules
```

## Preparar la aplicación para Docker

```
cat {path-to-project}/hotel-nest/src/app.module.ts
```

```
import { Module } from '@nestjs/common';
import { AppController } from './app.controller';
import { AppService } from './app.service';
import { LimpiezaModule } from './limpieza/limpieza.module';
import { AuthModule } from './auth/auth.module';
import { UsuarioModule } from './usuario/usuario.module';
import { MongooseModule } from '@nestjs/mongoose';
import { config } from 'dotenv';

config();

const dbHost = process.env.DEV_DB_HOST || 'hotelDB';
const dbPort = process.env.DEV_DB_HOST_PORT || '27017';
const devDbName = process.env.DB_NAME || 'hotel';

@Module({
 imports: [
```

```

 LimpiezaModule,
 MongooseModule.forRoot(`mongodb://${dbHost}:${dbPort}/${devDbName}`)
 `),
 UsuarioModule,
 AuthModule,
 UsuarioModule,
],
controllers: [AppController],
providers: [AppService],
)
export class AppModule {}

```

**⚠** En este caso, se han añadido variables de entorno para poder hacer configuraciones distintas en función del entorno en el que se ejecute la aplicación.

## Construir la imagen de Docker

```
sudo docker build -t hotel-nest .
```

Comprobamos que se ha creado correctamente con el siguiente comando:

```
docker images
```

| REPOSITORY | TAG    | IMAGE        | ID | CREATED   | SIZE   |
|------------|--------|--------------|----|-----------|--------|
| hotel_nest | latest | 76ccfe099c89 | 2  | hours ago | 1.33GB |
| mongo      | Latest | ff65a94ec485 | 2  | weeks ago | 795MB  |

Ejecutamos el contenedor con el siguiente comando:

```
sudo docker run -d --name HotelNest --network node_mongo --restart
\
unless-stopped -p 3000:3000 hotel_nest
```

Comprobemos que se ha creado correctamente con el siguiente comando:

docker ps

| CONTAINER ID | IMAGE                                      | COMMAND                   | CREATED     |
|--------------|--------------------------------------------|---------------------------|-------------|
| STATUS       | PORTS                                      |                           | NAMES       |
| 3bb36a602b9c | hotel_nest                                 | "docker-entrypoint. S..." | 2 hours ago |
| Up 2 hours   | 0.0.0.0:3000->3000/tcp, ::: 3000->3000/tcp |                           |             |
| HotelNest    |                                            |                           |             |
| 187d7e16ecbf | mongo                                      | "docker-entrypoint. S..." | 2 hours ago |
| Up 2 hours   | 127.0.0.1:27017->27017/tcp                 |                           |             |
| hotelDB      |                                            |                           |             |

## Comprobar que la aplicación funciona

- ⚠** Para comprobar que la aplicación funciona correctamente podemos probar en un navegador:

  - <http://amorenoiborra.es:3000/limpieza/sucias>
    - Este endpoint nos devuelve las habitaciones que no han sido limpiadas el día de hoy.

```
 "descripcion": "No funciona el aire acondicionado",
 "inicio": "2023-09-19T18:12:54.000Z",
 "_id": "10011a1a1a1a1a1a1a1a1a1a1a1a1a1a"
 },
 {
 "descripcion": "No funciona el interruptor del aseo",
 "inicio": "2023-09-20T10:15:06.000Z",
 "_id": "10021a1a1a1a1a1a1a1a1a1a1a1a1a1a"
 }
],
"__v": 0
},
{
 "_id": "2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b2b",
 "numero": 2,
 "tipo": "familiar",
 "descripcion": "Habitación familiar, cama XL y literas, aseo con bañera",
 "ultimaLimpieza": "2023-08-02T10:35:15.000Z",
 "precio": 65.45,
 "incidencias": [],
 "__v": 0
},
{
 "_id": "3c3c3c3c3c3c3c3c3c3c3c3c3c3c3c",
 "numero": 3,
 "tipo": "familiar",
 "descripcion": "Habitación familiar, cama XL y sofá cama, cocina con nevera",
 "precio": 69.15,
 "ultimaLimpieza": "2024-05-18T08:57:42.500Z",
 "incidencias": [],
 "__v": 0
},
{
 "_id": "4d4d4d4d4d4d4d4d4d4d4d4d4d4d",
 "numero": 4,
 "tipo": "suite",
```

```

 "descripcion": "Habitación con dos camas XL, terraza y
vistas al mar",
 "ultimaLimpieza": "2023-10-10T12:05:10.000Z",
 "precio": 110.2,
 "incidencias": [
 {
 "descripcion": "No funciona el jacuzzi",
 "inicio": "2023-10-08T19:24:43.000Z",
 "_id": "10014d4d4d4d4d4d4d4d4d4d4d"
 }
],
 "__v": 0
}
]
}

```



- <http://amorenoiborra.es:3000/limpieza/limpias>
  - Este endpoint nos devuelve las habitaciones que han sido limpiadas el día de hoy.

```
{
 "habitaciones": [
 {
 "_id": "5e5e5e5e5e5e5e5e5e5e5e5e",
 "numero": 5,
 "tipo": "individual",
 "descripcion": "Habitación simple, cama 150",
 "precio": 34.65,
 "ultimaLimpieza": "2024-05-28T08:57:42.501Z",
 "incidencias": [],
 "__v": 0
 }
]
}
```

]

}

# Resumen Tests

En este proyecto se han realizado una serie de tests

(<https://github.com/DanielAlmazan/hotel-nest/blob/main/test/axios/axiosTests.mjs>).

Para ello, se ha utilizado axios.

1. Obtener limpiezas de una habitación (getRoomCleanings):

- Verifica que se puedan obtener las limpiezas de una habitación específica y que la respuesta tenga el estado HTTP correcto (200) y una cantidad mínima de registros.

2. Verificar si una habitación está limpia (isRoomClean):

- Comprueba si una habitación específica está limpia, esperando un estado HTTP 200 y un valor booleano ok que indique la limpieza.

3. Obtener la lista de habitaciones limpiadas hoy (getCleanedRooms):

- Verifica que se pueda obtener una lista de habitaciones limpiadas hoy, esperando un estado HTTP 200 y al menos un registro.

4. Login incorrecto (incorrectLogin):

- Intenta realizar un login con credenciales incorrectas y espera recibir un error 401.

5. Login correcto (correctLogin):

- Realiza un login con credenciales correctas y espera un estado HTTP 201.

6. Insertar limpieza sin login (insertCleaningNoLogin):

- Intenta insertar una limpieza sin haber iniciado sesión y espera un error 401.

7. Insertar limpieza con login correcto (insertCleaning):

- Inserta una limpieza después de haber iniciado sesión correctamente y espera un estado HTTP 201.

8. Actualizar limpieza sin login (updateCleaningNoLogin):

- Intenta actualizar una limpieza sin haber iniciado sesión y espera un error 401.

9. Actualizar limpieza con login correcto (updateCleaning):

- Actualiza una limpieza después de haber iniciado sesión correctamente y espera un estado HTTP 200.

10. Ejecutar todos los tests (executeTests):

- Ejecuta todos los tests anteriores en secuencia y cuenta cuántos tests fueron exitosos y cuántos fallaron, mostrando el resultado final.

```
node test/axios/axiosTests.
```

```
OK - Inserción de limpieza sin login
OK - Actualización de limpieza sin login
OK - Login incorrecto
OK - Login correcto
OK - Inserción de limpieza con login
OK - Actualización de limpieza con login
OK - Habitaciones limpiadas hoy
OK - Limpiezas de habitación
OK - Habitación limpia
OK - Habitación sucia
Tests finalizados
Tests exitosos: 10
Tests fallidos: 0
```

Para más detalles, visitar el Hotel Nest (<https://github.com/DanielAlmazan/hotel-nest/tree/main>) en GitHub.