

## Modelo de Worknest

### Usuario

- Pertenece a varios workspace
- Tener distintos roles según el workspace
- Ser Team Leader en uno y colaborador en otro

**El rol NO vive en el usuario, vive en su relación con el workspace**

### Workspace

- Representa:
  - Un FreeLancer solo
  - Un equipo pequeño
  - Una miniempresa
- Tiene:
  - Muchos usuarios
  - Muchos proyectos
  - Invitaciones pendientes

**El creador del workspace es Team Leader por defecto**

### WorkspaceMember

- Relación a usuario
- Relación a workspace
- Rol en el workspace (Team Leader, Colaborator, Client)
- Un usuario puede tener roles distintos
- Los clientes existen sin poder editar nada

## Proyecto

- Tiene:
  - o Nombre
  - o Descripcion
  - o Estado
  - o Team Leaders asignados
  - o Sub-equipos
- Solo los Team Leaders pueden:
  - o Crear tareas
  - o Modificar tareas
  - o Asignar tareas

## Sub-equipo (Team)

- Un proyecto puede tener:
  - o Varios sub-equipos
  - o Cada sub-equipo con su propio Team Leader
  - o Varios colaboradores
- Esto sirve para:
  - o Backend
  - o Frontend
  - o Diseño
  - o QA

## Tarea (Task)

- Pertenece a un proyecto
- Puede pertenecer a un sub-equipo
- Tiene:
  - Estado
  - Prioridad
  - Fechas
  - Usuarios asignados

**Si no estas asignado NO puedes cambiarla**

## Asignación de Tareas (TaskAssignment)

- Relaciona:
  - Usuarios
  - Tareas
- Sirve para:
  - Validar permisos
  - Saber quién puede marcar como hecha
  - Evitar trampas del frontend

## Invitaciones

- Email
- Workspace
- Rol al que se invita
- Token
- Estado (pendiente/aceptada)

**Cuando el usuario acepta se crea un WorkspaceMember**

## Relaciones importantes

- Usuario ↔ Workspace → WorkspaceMember
- Workspace → Proyectos
- Proyecto → Sub-equipos
- Proyecto → Tareas
- Tarea → Usuario (TaskAssignment)

## Reglas

- Clientes → solo lectura
- Colaborador no asignado → no cambia estado de tarea
- Team Leader puede delegar
- Sub-equipos tienen jerarquía
- Seguridad vive en backend

## Endpoints

Convención general:

- URL Base: /api/
- Regresa JSON
- Autenticación con token (JWT)

Autenticación:

- POST /api/auth/register/ → registrar una cuenta
- POST /api/auth/login/ → iniciar session
- POST /api/auth/logout/ → cerrar session
- GET /api/auth/me/ → saber quién soy

Workspaces

- GET /api/workspaces/ → regresar mis workspaces
- POST /api/workspaces/ → crear workspace
- GET /api/workspaces/{id} → regresar workspace específico

WorkspacesMembers

- GET /api/workspaces/{id}/members/ → regresar los miembros de un workspace
- POST /api/workspaces/{id}/invite/ → enviar correo de invitacion
- PATCH /api/workspaces/{id}/members/{member\_id}/role/ → asignar rol al miembro

## Invitaciones

- POST /api/invitations/accept/ → aceptar la invitacion
- GET /api/invitations/{token}/ → regresar la invitacion

## Proyectos

- GET /api/workspaces/{id}/projects/ → regresar los proyectos del workspace
- POST /api/workspaces/{id}/projects/ → crear un proyecto en el workspace
- GET /api/projects/{id}/ → regresar el Proyecto
- PATCH /api/projects/{id}/ → editar datos de un proyecto

## Sub-Equipos

- GET /api/projects/{id}/teams/ → regresar los equipos del proyecto
- POST /api/projects/{id}/teams/ → crear un equipo para el proyecto
- PATCH /api/teams/{id} → editar un equipo

## Tareas

- GET /api/projects/{id}/tasks/ → regresar las tareas del Proyecto
- POST /api/projects/{id}/tasks/ → crear una tarea para el proyecto
- GET /api/tasks/{id} → regresar una tarea especifica
- PATCH /api/tasks/{id} → cambiar estado o editar la tarea

## Asignacion de Tarea

- POST /api/tasks/{id}/assign/ → asignar un usuario a la tarea
- DELETE /api/tasks/{id}/unassign/{user\_id}/ → desasignar un usuario de la tarea

