# Tas-kode

DESCRIPCIÓN BREVE  
Plataforma web que conecta programadores para tareas puntuales (“micro-contratos”). Un usuario publica una vacante, otro envía una propuesta y, al aceptarla, el sistema retiene el pago en garantía (escrow). El contratado sube archivos; el contratante revisa y libera (o disputa) el pago.

CLASES PRINCIPALES

Usuario  
Id: clave primaria.  
Nombre, Email, PasswordHash.  
Rol: Admin o Common.  
AvatarUrl: ruta al avatar.  
Perfil: texto en el que el usuario describe habilidades o necesidades.  
Relaciones: publica Vacantes, envía Propuestas.

Vacante  
Id: clave primaria.  
Título: breve descripción de la tarea.  
Descripción: detalle de la tarea puntual.  
Monto: cantidad ofrecida.  
FechaCreacion: marca cuándo se publicó.  
FechaExpiracion: fecha límite para recibir propuestas.  
IsAbierta: booleano que permanece true hasta que se acepte una propuesta, luego pasa a false.  
Relaciones: pertenece a un Usuario; contiene varias Propuestas.

Propuesta  
Id: clave primaria.  
Mensaje: texto del proponente explicando por qué debe tomar la tarea.  
FechaEnvio: fecha en la que se envió la propuesta.  
IsAceptada: booleano que se marca true cuando el contratante acepta esa propuesta.  
IsRechazada: booleano que se marca true si el contratante la rechaza explícitamente.  
Relaciones: pertenece a un Usuario (quien la envía) y a una Vacante. Si se acepta, genera un Contrato.

Contrato  
Id: clave primaria.  
FechaInicio: se registra al aceptar la propuesta (momento de retención del pago).  
FechaFin: se registra cuando el contrato finaliza (pago liberado o reembolsado).  
IsDelivered: booleano que indica si el contratado ya subió los archivos.  
IsClosed: booleano que marca que el contrato está finalizado.  
IsDisputed: booleano que se activa si el contratante solicita disputa.  
Relaciones: 1-a-1 con una Propuesta; 1-a-1 con un Pago; 1-a-muchos con Archivos.

Pago  
Id: clave primaria.  
Monto: cantidad retenida al aceptar la propuesta.  
FechaRetencion: marca cuándo se bloquean los fondos.  
FechaLiberacion: marca cuándo se libera el pago al contratado.  
FechaReembolso: marca cuándo se devuelve el pago al contratante en caso de disputa.  
IsReleased: booleano que indica si el pago ya fue liberado.  
IsRefunded: booleano que indica si el pago fue devuelto.  
Métodos principales:

Archivo  
Id: clave primaria.  
Nombre: nombre original del archivo.  
Ruta: ruta física o URL donde se almacena.  
FechaSubida: fecha de la carga.  
Relación: pertenece a un Contrato (entregables del contratado).

FLUJO RESUMIDO

1. Registro y Login  
   • El usuario se registra como Common ; sube avatar y completa Perfil.  
   • Identity maneja autenticación y autorización.
2. Publicar Vacante  
   • Cualquier usuario Common puede crear una Vacante con título, descripción, monto, fecha de expiración. Por defecto IsAbierta = true.  
   • La vista principal usa Vue.js + Axios para llamar GET   
   • Los resultados se muestran paginados en la columna central.
3. Enviar Propuesta  
   • El usuario Common navega el listado y hace clic en una tarjeta de vacante.  
   • El detalle aparece en la columna derecha, con botón “Enviar propuesta”  
   • En crear propuesta, el usuario escribe su Mensaje y envía.  
   • Se crea un registro en Propuestas con IsAceptada = false e IsRechazada = false.
4. Aceptar Propuesta → Contrato + Pago  
   • El contratante ve sus propuestas   
   • Al aceptar, se actualiza Propuesta.IsAceptada = true, Vacante.IsAbierta = false.  
   • Se crea un Contrato con FechaInicio = ahora, IsDelivered = false, IsClosed = false, IsDisputed = false.  
   • Se crea un Pago asociado con Retener(), guardando FechaRetencion y bloqueando el monto.
5. Entrega de Archivos y Revisión  
   • El contratado, desde “Mis contratos”, ve el contrato con IsDelivered = false.  
   • Hace clic en “Entregar trabajo”, sube uno o varios Archivos  
   • Al subir, el backend guarda los archivos, genera registros en Archivo con ContratoId, y marca Contrato.IsDelivered = true;   
   • Se inicia un temporizador de 5 días para revisión.
6. Liberación Automática o Aprobación Manual  
   • El contratante revisa los archivos dentro de esos 5 días. Si aprueba, ejecuta Pago.Liberar(), marca IsReleased = true y FechaLiberacion = ahora; luego Contrato.IsClosed = true y FechaFin = ahora.  
   • Si no responde en 5 días, un job automático comprueba Contratos con IsDelivered = true, !IsClosed, !IsDisputed y tiempo > 5 días, y ejecuta Pago.Liberar() y cierra el Contrato.
7. Disputa y Reembolso  
   • Si el contratante rechaza la entrega, hace POST /api/contratos/{id}/disputa, se marca Contrato.IsDisputed = true.  
   • Un Admin accede a “Contratos en disputa”, revisa archivos y decide:  
   – Liberar pago: Pago.Liberar() + Contrato.IsClosed = true.  
   – Reembolsar: Pago.Reembolsar() + Contrato.IsClosed = true.