Memoria: Examen Parcial 2

Ingeniería Web

Miguel Ángel Dorado Maldonado

Tecnologías

Las tecnologías utilizadas en el parcial(lenguajes, bibliotecas, frameworks, base de datos, etc.). Para la base de datos vamos a utilizar MongoDB Atlas. Para construir la API usaremos el framework Flask de Python. Usaremos pymongo para conectarnos a la base de datos. Como ejemplos de pruebas ejecutables se ha realizado un conjunto de pruebas en Postman.
En la base de datos se validarán los atributos necesarios para no permitir la entrada de valores erróneos.

Entidades

Vamos a disponer de una entidad Tareas la cual almacenará los atributos responsable, descripcion, habilidades (lista de strings) y segmentos
Se dispondrá tambien de una entidad Colaboradores que tendrá los atributos email, nombre y habilidades
Se creará una coleccion Participantes que relacionará las el id de las tareas con el id de los colaboradores que la realizan.

Quedando este esquema:

Tareas:
- responsable
- descripcion
- habilidades
- segmentos

Colaboradores:
- email
- nombre
- habilidades

Participantes:
- idTarea
- idColaborador
- nombreColaborador

Funciones adicionales

Tareas que requieren una determinada habilidad: Se realiza como searchParam en la petición GET sobre /tareas?habilidad=xxx

Tareas asignadas a un determinado colaborador.
Se ha creado la colección participantes que guarda una relación entre el id de las tareas y el id de los colaboradores
Mediante {base_url}/colaboradores/<id>/tareas se obtiene la lista de tareas asociadas

Asignar un colaborador a una tarea
Se añade comprobando las habilidades requeridas mediante un POST al endpoint {base_url}/colaboradores/<id>/tareas pasando como body {idTarea:xxx}

Buscar candidatos a colaborar en una tarea
Para obtener los emails de los posibles candidatos, hay que solicitar una petición GET a {base_url}/tareas/<id>/candidatos

Tareas completamente asignadas
Para mostrar las tareas que estén completas o sin completar se ha modificado la request GET de {base_url}/tareas para aceptar un searchParams. Con el param “completa” podemos buscar por tareas completas si igualamos a true o tareas incompletas si igualamos a false

- {base_url}/tareas?completa=true (devuelve tareas completas)
- {base_url}/tareas?completa=false (devuelve tareas incompletas)
- {base_url}/tareas (devuelve todas las tareas)

Buscar colaboradores de un determinado usuario
Para mostrar la lista de correos que han trabajado alguna vez para un responsable haremos una peticion GET a {base_url}/colaboradores/<id>/relaciones

Configuración

La url de la conexión a la base de datos de mongo atlas está definida en el .env.
Solo se deberá montar el contenedor de docker. Toda la configuración está ya lista para funcionar sin necesidad de modificar nada.

Para ello, siga los siguiente pasos:

- Descomprimir el archivo zip de la entrega.
- Ejecutar en la ruta de la carpeta “docker compose up –build” o “docker-compose up –build” dependiendo de la versión de docker que disponga. Automáticamente se iniciará el servidor en el puerto definido en el .env
- Tener en cuenta que el desarrollo de pruebas en postman están definidos bajo el puerto 5000.

Los endpoints están definidos arriba y las peticiones se podrán importar a postman a partir del archivo generado en la entrega en formato .json.