Este documento resume la estructura y los flujos principales del backend de SkillSwap para que nuevos contribuidores puedan entender rˆ¡pidamente cˆ‡mo funciona la API.

PILA TECNOLˆ GICA Y CONFIGURACIˆ N BASE
â¢ **Framework**: Django 5.2.8 con Django REST Framework para exponer APIs.
â¢ **Autenticaciˆ‡n**: Tokens (DRF) y sesiones. Se usa ‘dj-rest-auth‘ + ‘django-allauth‘ para registro e inicio de sesiˆ‡n, con un modelo de usuario personalizado que usa el email como identificador.
â¢ **Internacionalizaciˆ‡n**: Configuraciˆ‡n regional ‘es-cl‘ y zona horaria ‘America/Santiago‘.
â¢ **CORS/CSRF**: Permite orˆ›genes especˆ›ficos (localhost:5173, dominios de produccˆ‡n) y habilita credenciales.
â¢ **Archivos estˆ¡ticos y media**: MEDIA_ROOT apunta a ‘media/‘ y se sirven en desarrollo con ‘DEBUG=True‘.

APPS Y RESPONSABILIDADES
â¢ **core**: ‘healthcheck‘ simple en la raˆ›z (‘/‘) para verificaciones de disponibilidad.
â¢ **usuarios**: Gestiˆ‡n de personas, habilidades, coincidencias (match), valoraciones y notificaciones.
â¢ **chat**: Conversaciones y mensajes entre usuarios que ya hicieron match, con streaming SSE.

MODELOS CLAVE
â¢ **Usuario** (‘usuarios.models.Usuario‘): Extiende ‘AbstractUser‘ eliminando username; campos de nombre, segundo nombre, apellido, email ˆ”nico, telˆ'fono validado, imagen de perfil y enlaces de WhatsApp generados. Mantiene relaciones ManyToMany con habilidades que sabe y quiere aprender, ademˆ¡s de matches entre usuarios.
â¢ **Habilidad / TipoHabilidad**: Catˆ¡logo de habilidades agrupadas por tipo.
â¢ **ValoracionUsuario**: Relaciona evaluador y evaluado, con puntuaciˆ‡n de 1-5, comentario opcional y restricciones para evitar auto-valoraciones y duplicados.
â¢ **SolicitudMatch**: Solicitudes de conexiˆ‡n entre usuarios con estados ‘indefinido‘, ‘aceptado‘ o ‘rechazado‘, controlando que no se envˆ›en a uno mismo. Aceptar agrega el match y oculta notificaciones relacionadas.
â¢ **Notificacion**: Guarda eventos (por ahora ‘solicitud_match‘) con flags de lectura y visibilidad, ordenados del mˆ¡s reciente al mˆ¡s antiguo.
â¢ **Conversacion** (chat): Agrupa participantes y actualiza ‘fecha_actualizacion‘ al enviar mensajes.
â¢ **Mensaje** (chat): Contenido, remitente, marca de lectura y timestamp por conversaciˆ‡n.

ENDPOINTS PRINCIPALES
â¢ **Autenticaciˆ‡n** (‘/api/auth/‘ y ‘/api/auth/registration/‘): Manejado por ‘dj-rest-auth‘; usa serializadores personalizados para login/registro con email.
â¢ **Usuarios** (‘/api/usuarios/‘): CRUD completo. Acciones adicionales:
- ‘coincidencias/‘: devuelve usuarios compatibles segˆ”n habilidades que sabe/quiere aprender el solicitante, paginado.
- ‘buscar/?q=...‘: filtro por nombre/apellidos o nombres de habilidades.
â¢ **Habilidades y tipos**: ‘/api/habilidades/‘ y ‘/api/tipos-habilidad/‘ con CRUD protegido.
â¢ **Valoraciones**: ‘/api/valoraciones/‘ permite crear/consultar valoraciones; el evaluador se fija al usuario autenticado.
â¢ **Solicitudes de match**: ‘/api/solicitudes-match/‘ (GET/POST/DELETE) y acciones ‘aceptar/‘ y ‘rechazar/‘ para el destinatario.

actualizaciones parciales (ej. marcar leˆ›do/mostrar).
â¢ **Chat** (‘/api/chat/conversaciones/‘):
- Listado/creaciˆ‡n de conversaciones solo con participantes que ya tienen match.
- ‘mensajes/‘: lista mensajes, acepta filtro ‘?since=‘ en ISO para obtener solo los nuevos.
- ‘enviar/‘: crea mensajes dentro de una conversaciˆ‡n donde el usuario participa y mantiene match.
- ‘conversaciones/<id>/stream/‘: endpoint SSE (autenticaciˆ‡n por Token) que emite mensajes en tiempo real.
â¢ **Healthcheck**: ‘/‘ responde ‘{ "status": "ok" }‘.

FLUJOS RECOMENDADOS
1. Registrar usuario vˆ›a ‘/api/auth/registration/‘ y completar perfil (habilidades que sabe y desea aprender).
2. Buscar coincidencias o usuarios por nombre/habilidad; enviar ‘solicitudes-match‘ a quienes interesen.
3. Cuando el destinatario acepta, ambos quedan en ‘matches‘ y pueden abrir una conversaciˆ‡n de chat.
4. Enviar y recibir mensajes; el cliente puede suscribirse por SSE para actualizaciones en vivo.
5. Tras colaborar, crear ‘valoraciones‘ para retroalimentaciˆ‡n entre usuarios.

ARCHIVOS DE REFERENCIA
â¢ Configuraciˆ‡n y seguridad: ‘skillswap/skillswap/settings.py‘.
â¢ Modelos y serializers: app ‘usuarios‘ y ‘chat‘ (modelos, serializers y vistas en el mismo directorio).
â¢ Rutas API: ‘skillswap/skillswap/urls.py‘, ‘usuarios/urls.py‘, ‘chat/urls.py‘, ‘core/urls.py‘.

GENERACIˆ N DEL PDF
Ejecuta ‘python docs/generar_pdf_resumen.py‘ para producir ‘docs/backend_resumen.pdf‘ a partir de este resumen (no necesita dependencias externas).