Contrato de API REST - FITTIA

## 1.1 Registro de usuario

***POST /api/v1/auth/register***

Descripción: Crea un nuevo cliente o entrenador.

Body (JSON):

* ● firstName (string, obligatorio)
* ● lastName (string, obligatorio)
* ● email (string, obligatorio, único, formato válido)
* ● password (string, obligatorio, ≥8 caracteres, 1 mayúscula, 1 número, 1 símbolo)
* ● birthDate (string, obligatorio, ISO8601)
* ● role (string, obligatorio, valores: client | trainer)
* Ejemplo de Body:

{  
 "firstName": "Tomas",  
 "lastName": "Barisone",  
 "email": "tomas@example.com",  
 "password": "Password123!",  
 "birthDate": "1999-04-16",  
 "role": "client"  
}

Respuestas:

* ● 201 Created: Devuelve el objeto User completo (id, firstName, lastName, email, role, birthDate, createdAt).
* ● 400 Bad Request: Datos faltantes o no cumplen validaciones.
* ● 409 Conflict: Email ya registrado.
* Ejemplo de Respuesta:

{  
 "id": "abc123",  
 "firstName": "Tomas",  
 "lastName": "Barisone",  
 "email": "tomas@example.com",  
 "role": "client",  
 "birthDate": "1999-04-16",  
 "createdAt": "2025-04-30T14:32:00Z"  
}

## 1.2 Iniciar sesión

***POST /api/v1/auth/login***

Descripción: Autentica a un usuario y devuelve un token JWT.

Body (JSON):

* ● email (string, obligatorio, formato válido)
* ● password (string, obligatorio)
* Ejemplo de Body:

{  
 "email": "tomas@example.com",  
 "password": "Password123!"  
}

Respuestas:

* ● 200 OK: Devuelve accessToken y datos básicos del usuario.
* ● 400 Bad Request: Faltan campos o son inválidos.
* ● 401 Unauthorized: Credenciales incorrectas.
* Ejemplo de Respuesta:

{  
 "accessToken": "jwt\_token\_aqui",  
 "user": {  
 "id": "abc123",  
 "firstName": "Tomas",  
 "lastName": "Barisone",  
 "email": "tomas@example.com",  
 "role": "client"  
 }  
}

## 1.3 Solicitar restablecimiento de contraseña

***POST /api/v1/auth/password-reset/request***

Descripción: Envía un código al correo electrónico para iniciar el proceso de recuperación de contraseña.

Body (JSON):

* ● email (string, obligatorio, formato válido)
* Ejemplo de Body:

{  
 "email": "tomas@example.com"  
}

Respuestas:

* ● 200 OK: Código enviado correctamente.
* ● 404 Not Found: No existe un usuario con ese email.
* ● 400 Bad Request: Email malformado o faltante.
* Ejemplo de Respuesta:

{  
 "message": "Código de recuperación enviado al correo electrónico."  
}

## 1.4 Validar código de restablecimiento

***POST /api/v1/auth/password-reset/validate***

Descripción: Valida que el código enviado al correo sea correcto y no haya expirado.

Body (JSON):

* ● email (string, obligatorio)
* ● code (string, obligatorio)
* Ejemplo de Body:

{  
 "email": "tomas@example.com",  
 "code": "123456"  
}

Respuestas:

* ● 200 OK: Código válido.
* ● 400 Bad Request: Código inválido, expirado o faltante.
* Ejemplo de Respuesta:

{  
 "message": "Código válido"  
}

## 1.5 Establecer nueva contraseña

***POST /api/v1/auth/password-reset/confirm***

Descripción: Permite establecer una nueva contraseña luego de validar el código.

Body (JSON):

* ● email (string, obligatorio)
* ● newPassword (string, obligatorio, mínimo 8 caracteres, 1 mayúscula, 1 número, 1 símbolo)
* Ejemplo de Body:

{  
 "email": "tomas@example.com",  
 "newPassword": "NuevaClave123!"  
}

Respuestas:

* ● 200 OK: Contraseña restablecida exitosamente.
* ● 400 Bad Request: Contraseña inválida.
* ● 404 Not Found: Usuario no encontrado o flujo inválido.
* Ejemplo de Respuesta:

{  
 "message": "Contraseña actualizada con éxito."  
}

## 2.1 Obtener todos los servicios disponibles

***GET /api/v1/services***

Descripción: Devuelve una lista de todos los servicios disponibles para que un cliente los explore o contrate.

Respuestas:

* ● 200 OK: Lista devuelta correctamente.
* ● 204 No Content: No hay servicios disponibles.
* Ejemplo de Respuesta:

[  
 {  
 "id": "serv123",  
 "title": "Running",  
 "description": "Entrenamiento de running orientado a desarrollar potencia...",  
 "price": 10999,  
 "imageUrl": "https://url.com/running.png",  
 "rating": 4,  
 "trainer": {  
 "id": "user123",  
 "name": "Roberto Carlos"  
 }  
 }  
]

## 2.2 Buscar servicios con filtros

***GET /api/v1/services?category=running&minPrice=5000&maxPrice=15000&location=Palermo&rating=4&mode=virtual***

Descripción: Devuelve una lista de servicios filtrados según las preferencias del cliente.

Body (JSON):

* ● category (string, opcional)
* ● minPrice / maxPrice (int, opcional)
* ● duration (int, opcional)
* ● location (string, opcional)
* ● language (string, opcional)
* ● rating (int, opcional)
* ● mode (string: presencial | virtual, opcional)

Ejemplo de uso:

GET /api/v1/services?category=running&minPrice=5000&maxPrice=15000&location=Palermo&rating=4&mode=virtual?category=running&minPrice=5000&maxPrice=15000&location=Palermo&rating=4&mode=virtual

Respuestas:

* ● 200 OK: Servicios filtrados encontrados.
* ● 204 No Content: No hay coincidencias.
* ● 400 Bad Request: Filtros inválidos.
* Ejemplo de Respuesta:

[  
 {  
 "id": "serv124",  
 "title": "Pilates",  
 "description": "Entrenamiento de pilates enfocado en el core...",  
 "price": 8999,  
 "imageUrl": "https://url.com/pilates.png",  
 "rating": 5,  
 "trainer": {  
 "id": "user124",  
 "name": "Camila Fernandez"  
 }  
 }  
]

## 2.3 Obtener servicios contratados por el cliente

***GET /api/v1/users/{userId}/my-services***

Descripción: Devuelve una lista de los servicios contratados por el cliente autenticado.

Respuestas:

* ● 200 OK: Lista devuelta correctamente.
* ● 204 No Content: El cliente no contrató ningún servicio.
* ● 401 Unauthorized: Token inválido o ausente.
* Ejemplo de Respuesta:

[  
 {  
 "serviceId": "serv123",  
 "title": "Running",  
 "description": "Entrenamiento de running orientado a desarrollar potencia...",  
 "imageUrl": "https://url.com/running.png",  
 "trainer": {  
 "id": "trainer001",  
 "name": "Roberto Carlos"  
 },  
 "rating": 4,  
 "status": "aceptado"  
 }  
]

## 3.1 Obtener servicios creados por un entrenador

***GET /api/v1/trainers/{trainerId}/services***

Descripción: Devuelve la lista de servicios que el entrenador ha creado.

Respuestas:

* ● 200 OK: Lista devuelta correctamente.
* ● 204 No Content: No tiene servicios creados.
* ● 401 Unauthorized: Token inválido o ausente.
* Ejemplo de Respuesta:

[  
 {  
 "id": "001",  
 "title": "Running",  
 "description": "Entrenamiento de running orientado a desarrollar potencia...",  
 "status": "publicado"  
 }  
]

## 3.2 Crear nuevo servicio

***POST /api/v1/services***

Descripción: Crea un nuevo servicio que podrá ser publicado o guardado como borrador.

Body (JSON):

* ● title (string, obligatorio)
* ● description (string, obligatorio)
* ● price (number, obligatorio)
* ● duration (int, obligatorio, en horas)
* ● location (string, obligatorio)
* ● language (string, obligatorio)
* ● mode (string: presencial | virtual, obligatorio)
* ● category (string, obligatorio)
* ● sessionsPerWeek (int, obligatorio)
* ● intensity (string, obligatorio)
* ● status (string: publicado | borrador, opcional)
* Ejemplo de Body:

{  
 "title": "Pilates",  
 "description": "Entrenamiento de pilates para core, equilibrio y fuerza.",  
 "price": 8999,  
 "duration": 2,  
 "location": "Palermo",  
 "language": "Español",  
 "mode": "presencial",  
 "category": "pilates",  
 "sessionsPerWeek": 2,  
 "intensity": "Alta",  
 "status": "borrador"  
}

Respuestas:

* ● 201 Created: Servicio creado con éxito.
* ● 400 Bad Request: Datos faltantes o inválidos.
* ● 401 Unauthorized: Usuario no autorizado.
* Ejemplo de Respuesta:

{  
 "id": "001",  
 "title": "Pilates",  
 "status": "borrador"  
}

## 3.3 Publicar o despublicar servicio

***PATCH /api/v1/services/{serviceId}***

Descripción: Cambia el estado del servicio entre 'publicado' y 'borrador'.

Body (JSON):

* ● status (string, obligatorio: publicado | borrador)
* Ejemplo de Body:

{  
 "status": "publicado"  
}

Respuestas:

* ● 200 OK: Estado actualizado correctamente.
* ● 400 Bad Request: Estado inválido.
* ● 403 Forbidden: No tiene permiso sobre el servicio.
* ● 404 Not Found: Servicio no encontrado.
* Ejemplo de Respuesta:

{  
 "message": "Servicio actualizado a estado publicado."  
}

## 3.4 Eliminar servicio

***DELETE /api/v1/services/{serviceId}***

Descripción: Elimina un servicio creado por el entrenador.

Respuestas:

* ● 204 No Content: Servicio eliminado con éxito.
* ● 403 Forbidden: El usuario no tiene permiso sobre el servicio.
* ● 404 Not Found: Servicio no encontrado.
* Ejemplo de Respuesta:

Sin cuerpo. Código 204 No Content.

## 4.1 Obtener estadísticas del perfil del entrenador

***GET /api/v1/trainers/{trainerId}/stats***

Descripción: Devuelve visualizaciones, promedio de contrataciones, cantidad y promedio de calificaciones, y distribución de estrellas.

Respuestas:

* ● 200 OK: Estadísticas devueltas.
* ● 404 Not Found: Entrenador no encontrado.
* Ejemplo de Respuesta:

{  
 "views": 123,  
 "avgContractRate": 77.2,  
 "ratings": {  
 "count": 90,  
 "average": 4.6,  
 "distribution": {  
 "1": 2,  
 "2": 1,  
 "3": 4,  
 "4": 35,  
 "5": 48  
 }  
 }  
}

## 4.2 Obtener comentarios recibidos

***GET /api/v1/trainers/{trainerId}/comments***

Descripción: Devuelve los comentarios dejados por los clientes al entrenador.

Respuestas:

* ● 200 OK: Lista de comentarios devuelta.
* ● 404 Not Found: Entrenador no encontrado.
* Ejemplo de Respuesta:

[  
 {  
 "id": "cmt001",  
 "clientName": "Santiago Lopez",  
 "date": "2025-04-28T19:04:00",  
 "text": "Muy buen entrenador...",  
 "reply": "¡Gracias!"  
 }  
]

## 4.3 Responder a un comentario

***POST /api/v1/comments***

Descripción: Permite al entrenador responder a un comentario recibido.

Body (JSON):

* ● reply (string, obligatorio)
* Ejemplo de Body:

{  
 "reply": "Gracias por tu comentario :)"  
}

Respuestas:

* ● 200 OK: Respuesta registrada.
* ● 400 Bad Request: Texto vacío o muy largo.
* ● 403 Forbidden: No tiene permiso para responder ese comentario.
* ● 404 Not Found: Comentario no encontrado.
* Ejemplo de Respuesta:

{  
 "message": "Respuesta enviada correctamente."  
}

## 5.1 Ver perfil público del entrenador

***GET /api/v1/trainers/{trainerId}/profile***

Descripción: Devuelve información pública del entrenador, sus servicios publicados, promedio de calificación y comentarios recibidos.

Respuestas:

* ● 200 OK: Perfil público devuelto.
* ● 404 Not Found: Entrenador no encontrado.
* Ejemplo de Respuesta:

{  
 "trainer": {  
 "id": "trainer001",  
 "name": "Roberto Carlos",  
 "profileImage": "https://url.com/profile.jpg",  
 "averageRating": 4.6  
 },  
 "services": [  
 {  
 "id": "001",  
 "title": "Running",  
 "description": "Entrenamiento de running...",  
 "price": 10999  
 }  
 ],  
 "comments": [  
 {  
 "id": "cmt001",  
 "clientName": "Santiago Lopez",  
 "text": "Muy buen entrenador...",  
 "date": "2025-04-28T19:04:00",  
 "reply": "¡Gracias!"  
 }  
 ]  
}

## 5.2 Calificar al entrenador

***POST /api/v1/reviews***

Descripción: Permite a un cliente que haya contratado un servicio con el entrenador calificarlo con una puntuación entre 1 y 5.

Body (JSON):

* ● rating (int, obligatorio, entre 1 y 5)
* Ejemplo de Body:

{  
 "rating": 5  
}

Respuestas:

* ● 200 OK: Calificación registrada o actualizada.
* ● 400 Bad Request: Valor inválido.
* ● 403 Forbidden: El usuario no contrató al entrenador.
* ● 404 Not Found: Entrenador no encontrado.
* Ejemplo de Respuesta:

{  
 "message": "Calificación registrada correctamente."  
}

## 5.3 Comentar en el perfil del entrenador

***POST /api/v1/trainers/{trainerId}/comments***

Descripción: Permite a un cliente que haya contratado un servicio dejar un comentario público en el perfil del entrenador.

Body (JSON):

* ● text (string, obligatorio, máximo 500 caracteres)
* Ejemplo de Body:

{  
 "text": "Gran profesor, muy dedicado."  
}

Respuestas:

* ● 201 Created: Comentario agregado.
* ● 400 Bad Request: Comentario inválido.
* ● 403 Forbidden: No contratado.
* ● 404 Not Found: Entrenador no encontrado.
* Ejemplo de Respuesta:

{  
 "commentId": "cmt999",  
 "message": "Comentario publicado correctamente."  
}

## 6.1 Contratar un servicio

***POST /api/v1/contracts***

Descripción: Permite a un cliente contratar un servicio y enviar los horarios deseados para su aprobación.

Body (JSON):

* ● serviceId (string, obligatorio)
* ● selectedSchedule (array de objetos { day, time }, obligatorio)
* Ejemplo de Body:

{  
 "serviceId": "001",  
 "selectedSchedule": [  
 { "day": "martes", "time": "15:00" },  
 { "day": "jueves", "time": "17:00" }  
 ]  
}

Respuestas:

* ● 201 Created: Solicitud enviada y estado inicial 'pendiente'.
* ● 400 Bad Request: Horarios inválidos o cantidad incorrecta.
* ● 403 Forbidden: Contrato ya existente.
* ● 404 Not Found: Servicio no encontrado.
* Ejemplo de Respuesta:

{  
 "contractId": "c001",  
 "status": "pendiente"  
}

## 6.2 Ver detalle de contratación

***GET /api/v1/contracts/{contractId}***

Descripción: Devuelve los detalles de una contratación específica.

Respuestas:

* ● 200 OK: Contrato encontrado y devuelto.
* ● 403 Forbidden: Acceso no autorizado.
* ● 404 Not Found: Contrato no encontrado.
* Ejemplo de Respuesta:

{  
 "contractId": "c001",  
 "service": {  
 "id": "001",  
 "title": "Running",  
 "price": 10999  
 },  
 "trainer": {  
 "id": "t001",  
 "name": "Roberto Carlos"  
 },  
 "schedule": [  
 { "day": "martes", "time": "15:00" },  
 { "day": "jueves", "time": "17:00" }  
 ],  
 "status": "pendiente",  
 "remainingClasses": 8  
}

## 6.3 Cancelar contratación

***PATCH /api/v1/contracts/{contractId} (usando PATCH con status='cancelado')***

Descripción: Permite al cliente cancelar un servicio contratado.

Respuestas:

* ● 200 OK: Servicio cancelado.
* ● 403 Forbidden: No tiene permiso para cancelar este contrato.
* ● 404 Not Found: Contrato no encontrado o ya cancelado.
* Ejemplo de Respuesta:

{  
 "message": "Contrato cancelado correctamente."  
}

## 6.4 Recontratar servicio completado

***POST /api/v1/contracts (nuevo POST con referencia al contrato anterior)***

Descripción: Permite al cliente volver a contratar un servicio que ya fue completado.

Respuestas:

* ● 201 Created: Nueva contratación iniciada con estado 'pendiente'.
* ● 404 Not Found: Contrato original no encontrado.
* Ejemplo de Respuesta:

{  
 "newContractId": "c010",  
 "status": "pendiente"  
}

## 7.1 Modificar horarios de una contratación activa

***PUT /api/v1/contracts/{contractId} (usando PATCH para newSchedule)***

Descripción: Permite al cliente cambiar los horarios de clases de un contrato ya aceptado.

Body (JSON):

* ● newSchedule (array de objetos { day, time }, obligatorio)
* Ejemplo de Body:

{  
 "newSchedule": [  
 { "day": "martes", "time": "15:00" },  
 { "day": "jueves", "time": "17:00" }  
 ]  
}

Respuestas:

* ● 200 OK: Horarios actualizados.
* ● 400 Bad Request: Horarios inválidos.
* ● 403 Forbidden: No autorizado.
* ● 409 Conflict: No se puede modificar un contrato no aceptado.
* Ejemplo de Respuesta:

{  
 "message": "Horarios actualizados correctamente."  
}

## 8.2 Aceptar una solicitud

***PATCH /api/v1/contracts/{contractId} (usando PATCH con status='aceptado')***

Descripción: Permite al entrenador aceptar una contratación pendiente.

Respuestas:

* ● 200 OK: Solicitud aceptada con éxito.
* ● 403 Forbidden: El contrato no pertenece al entrenador.
* ● 404 Not Found: Contrato inexistente.
* Ejemplo de Respuesta:

{  
 "status": "aceptado"  
}

## 8.3 Rechazar una solicitud

***PATCH /api/v1/contracts/{contractId} (usando PATCH con status='rechazado')***

Descripción: Permite al entrenador rechazar una contratación pendiente.

Respuestas:

* ● 200 OK: Solicitud rechazada.
* ● 403 Forbidden: No autorizado.
* ● 404 Not Found: Contrato no encontrado.
* Ejemplo de Respuesta:

{  
 "status": "rechazado"  
}

## 9.1 Enviar material a un cliente

***POST /api/v1/contracts/{contractId}/materials***

Descripción: Permite al entrenador subir uno o más archivos (PDFs) asociados a un contrato aceptado con un cliente.

Body (JSON):

* ● file (PDF, obligatorio, multipart/form-data)
* ● fileName (string, obligatorio)

Respuestas:

* ● 201 Created: Material subido correctamente.
* ● 400 Bad Request: Archivo no válido.
* ● 403 Forbidden: No autorizado.
* ● 404 Not Found: Contrato no encontrado o no aceptado.
* Ejemplo de Respuesta:

{  
 "materialId": "mat001",  
 "message": "Material agregado correctamente."  
}

## 9.2 Obtener materiales de un contrato

***GET /api/v1/contracts/{contractId}/materials***

Descripción: Permite a cliente o entrenador ver la lista de materiales adjuntos al contrato.

Respuestas:

* ● 200 OK: Lista de materiales devuelta.
* ● 403 Forbidden: No autorizado.
* ● 404 Not Found: Contrato no válido.
* Ejemplo de Respuesta:

[  
 {  
 "materialId": "mat001",  
 "fileName": "Rutina de ejercicios",  
 "downloadUrl": "https://fittia.com/materiales/descargar/mat001.pdf"  
 }  
]

## 9.3 Descargar un material

***GET /api/v1/materials/{materialId}***

Descripción: Permite descargar directamente un archivo PDF de material educativo.

Respuestas:

* ● 200 OK: Devuelve el archivo PDF.
* ● 403 Forbidden: Acceso no autorizado.
* ● 404 Not Found: Material no encontrado.
* Ejemplo de Respuesta:

Retorna archivo binario PDF. Código 200 OK.

## 10.1 Obtener notificaciones del usuario

***GET /api/v1/notifications***

Descripción: Devuelve todas las notificaciones del usuario autenticado (cliente o entrenador).

Body (JSON):

* ● read=false (query param, opcional)
* ● limit=20 (query param, opcional)

Respuestas:

* ● 200 OK: Lista de notificaciones devuelta.
* ● 401 Unauthorized: Token inválido.
* Ejemplo de Respuesta:

[  
 {  
 "id": "notif001",  
 "message": "Recibiste una calificación de 4 estrellas por parte de Luciano Perez.",  
 "timestamp": "2025-04-28T21:13:00",  
 "read": false,  
 "redirectUrl": "/mis-estadisticas"  
 }  
]

## 10.2 Marcar notificación como leída

***PATCH /api/v1/notifications/{notificationId}/read***

Descripción: Marca una notificación como leída.

Respuestas:

* ● 200 OK: Notificación marcada como leída.
* ● 403 Forbidden: No sos el dueño de la notificación.
* ● 404 Not Found: No existe la notificación.
* Ejemplo de Respuesta:

{  
 "message": "Notificación actualizada correctamente."  
}

## [REMOVIDO - Uso interno, no se expone en contrato de API]

***POST /api/v1/notifications***

Descripción: Crea una notificación para un usuario. Usado internamente por el backend.

Body (JSON):

* ● recipientUserId (string, obligatorio)
* ● message (string, obligatorio)
* ● redirectUrl (string, opcional)
* Ejemplo de Body:

{  
 "recipientUserId": "user123",  
 "message": "Has contratado el servicio de Running (#001), queda pendiente de aprobación.",  
 "redirectUrl": "/solicitudes-pendientes"  
}

Respuestas:

* ● 201 Created: Notificación registrada.
* ● 400 Bad Request: Campos inválidos.
* ● 404 Not Found: Usuario no encontrado.
* Ejemplo de Respuesta:

{  
 "notificationId": "notif999",  
 "message": "Notificación creada correctamente."  
}