Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación
IIC2513 Tecnologías y Aplicaciones Web

Documentación Proyecto
GoGym

Profesor del curso : Sebastián Vicencio
Ayudante : Jorge Becerra
Grupo : 404
Integrantes : *Trinidad Vargas*
Daniela Villarroel
Juan Zapata
Link aplicación web : https://sleepy-brook-55941.herokuapp.com/

Santiago, 2 de diciembre de 2020

Tabla de Contenidos

1. Introducción

Debido a la pandemia, los gimnasios han debido cerrar sus puertas y cuando las puedan abrir, deberán disminuir su capacidad para adoptar las medidas de seguridad y no convertirse en un foco de infección. Por esto, es fundamental optimizar los turnos y poder agendar horas con anticipación.

2. Descripción

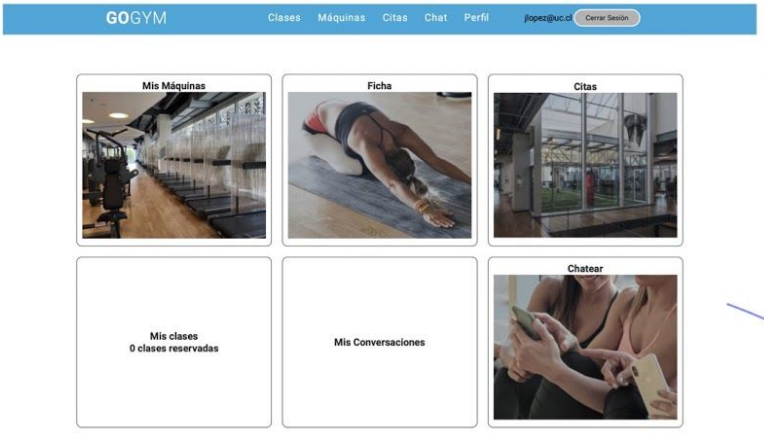
Para asegurar que el gimnasio sea un ambiente seguro, se implementa Gogym. En esta aplicación web podrás reservar las máquinas, clases y citas a las cuales desees asistir. De esta manera se cumplirá el aforo del gimnasio disminuyendo la probabilidad de contagio.
En Gogym podrás acceder a tu perfil, planes mensuales, clases, citas y chatear con tu personal trainer. Además, podrá agendar horas con un nutricionistas para llevar una alimentación saludable. Mediante el chat podrás tener una interacción más directa con tu entrenador y nutricionista. Los entrenadores tendrán acceso a tu perfil con datos de tu progreso.

2.1 Tipos de Usuario

- **Administrador** encargado de manejar la aplicación. Puede crear salas, eventos, reservas de máquinas y horarios disponibles para citas con especialistas.
- **Especialista** acceso más restringido que el administrador. Puede agregar o eliminar horarios en el que estará disponible y los clientes pueden reservar. Luego de la consulta, puede crear una nueva ficha de salud para el cliente, la que contendrá los avances de distintas medidas corporales y un comentario o resumen de la cita.
- **Cliente** Organiza tu tiempo de la mejor manera, reserva máquinas o citas con especialistas desde tu celular. Evita el contacto con otras personas y disminuye el riesgo de contagio. Además tendrás un chat disponible para conversar con los especialistas o administrador en caso de cualquier problema o pregunta.

2.2 Cómo usar la aplicación web

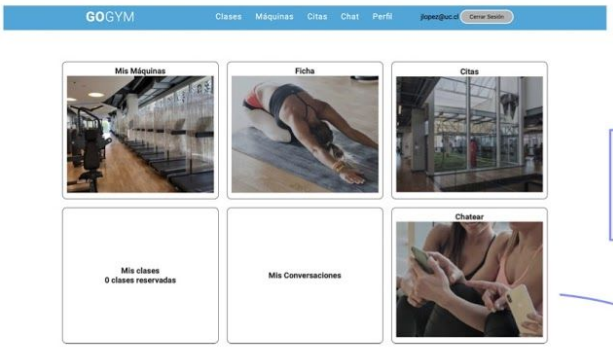
Se presenta el flujo de navegación del cliente, la aplicación web puede encontrarse en el siguiente link: https://sleepy-brook-55941.herokuapp.com



Para visualizar las citas disponibles aprieta citas en navbar o perfil.

Podrás ver la disponibilidad de horas. Haz click en ver más para reservar.

Haz click en agendas hora.



Para chatear puedes hacer click en la navbar o en el perfil en la imagen que dice chatear

En el chat tendrás acceso a tus conversaciones, eliminarlas o crear nuevas

Crear nueva conversación con administrador, entrenador o nutritionistas

Manda mensajes

1.4 Valor agregado

Chat
La aplicación web incluye una sección de chats para comunicarse con los especialistas del gimnasio y el administrador. El chat está construido a partir de una aplicación React la cual consume una api del mismo servidor.

Charts
También incluye Charts.js, lo cual le da un valor agregado para el administrador. De esta manera puede ver sus estadísticas y acceder a la información de forma rápida. Faltó seguir desarrollando esta parte en el proyecto.

Solución a un problema actual
Finalmente, el diseño de esta aplicación está pensado para solucionar un problema que tienen hoy en día los gimnasios y la gente que asiste a ellos. Debido a la pandemia, muchos gimnasios se han visto afectados por el riesgo que implica asistir a un gimnasio en relación al contagio del virus, es por esto que nuestra aplicación facilita la vuelta de los usuarios al gimnasio.

3. Especificación de requerimientos

3.1 Requerimientos funcionales

Sistema de usuarios para Administrador, entrenador, nutricionista y cliente.

- Administrador:
    - Definir capacidad de salas,disponibilidad de citas.
    - Ver usuarios.
    - Ver fichas de salud.
    - Crear salas, máquinas, citas y fichas.
    - Chat con clientes, entrenadores y nutricionistas.
- Cliente:
    - Poder agendar horas.
    - Ver ficha de salud.
    - Ver máquinas y clases disponibles.
    - Reservar máquinas y clases disponibles.
    - Chat con especialistas, nutricionista y administrador.
- Especialista:
    - Ver ficha de salud.
    - Crear fichas de salud.
    - Ver progreso clientes.
    - Reservar máquinas y clases disponibles.
    - Chat con especialistas, nutricionista y administrador.

- Sistema para agendar horas:
    - Mostrar horas disponibles con entrenadores
    - Mostrar horas disponibles con nutricionistas
    - Clientes:
        - Reservar/ cancelar sala de máquinas
        - Agendar/ cancelar hora con especialista.

3.2 Herramientas

La aplicación se desarrolló en Node.js y koa. Para el cliente, se usó usar HTML, CSS y JavaScript.

entornos de desarrollo integrados, plataformas y herramientas empleadas en la implementación del sistema

3.3 Planificación

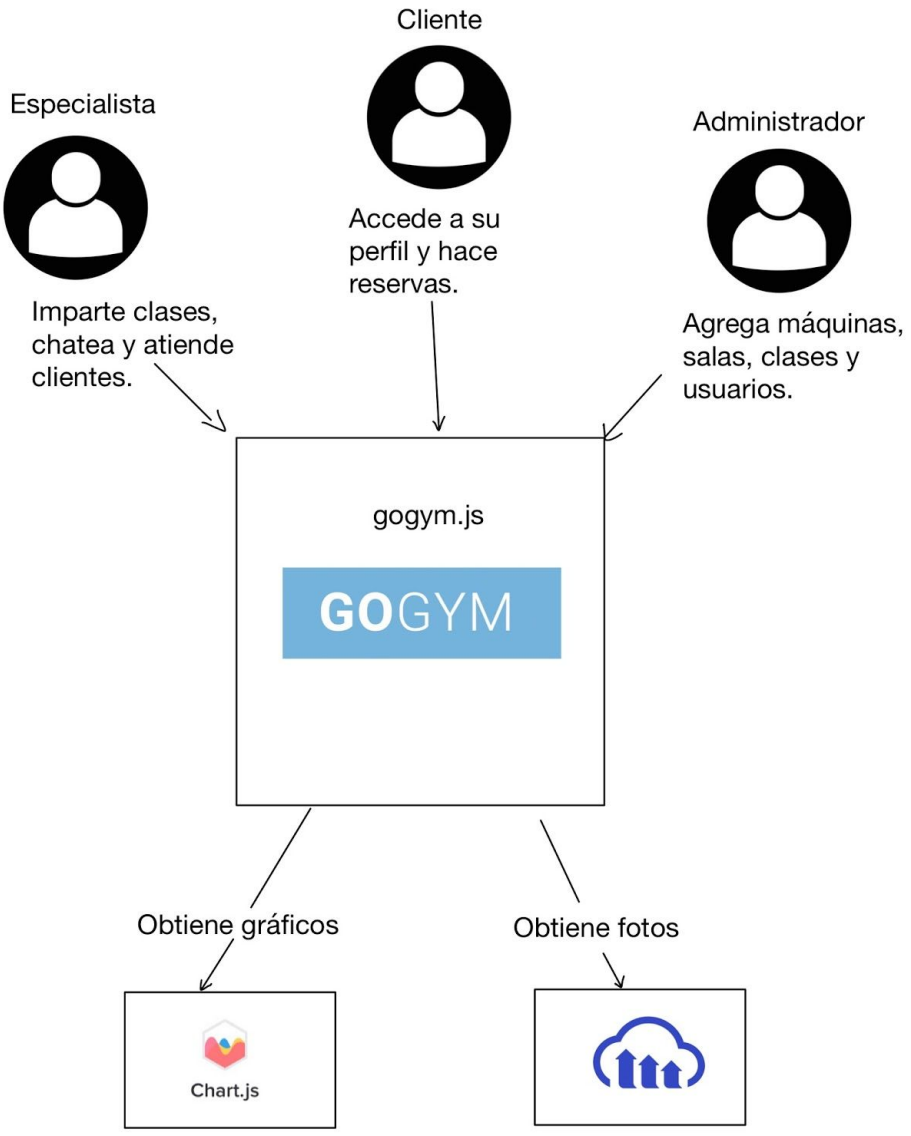
En primer lugar, se diseñó la aplicación, luego se diseñaron mockups y diagramas de datos. Posteriormente, se crearon todos los CRUD y las asociaciones para empezar a crear las funcionalidades. Finalmente, se incorporó React y se hizo una REST API. Para la organización del proyecto se ocupó la herramienta Trello, de esta manera se repartían las tareas de forma equitativa dentro del equipo. El proyecto se dividió en 6 entregas con plazos de dos semanas para cada entrega.

4. Arquitectura del sistema

Para describir la arquitectura del proyecto se ocupará un lenguaje gráfico denominado el modelo C4.

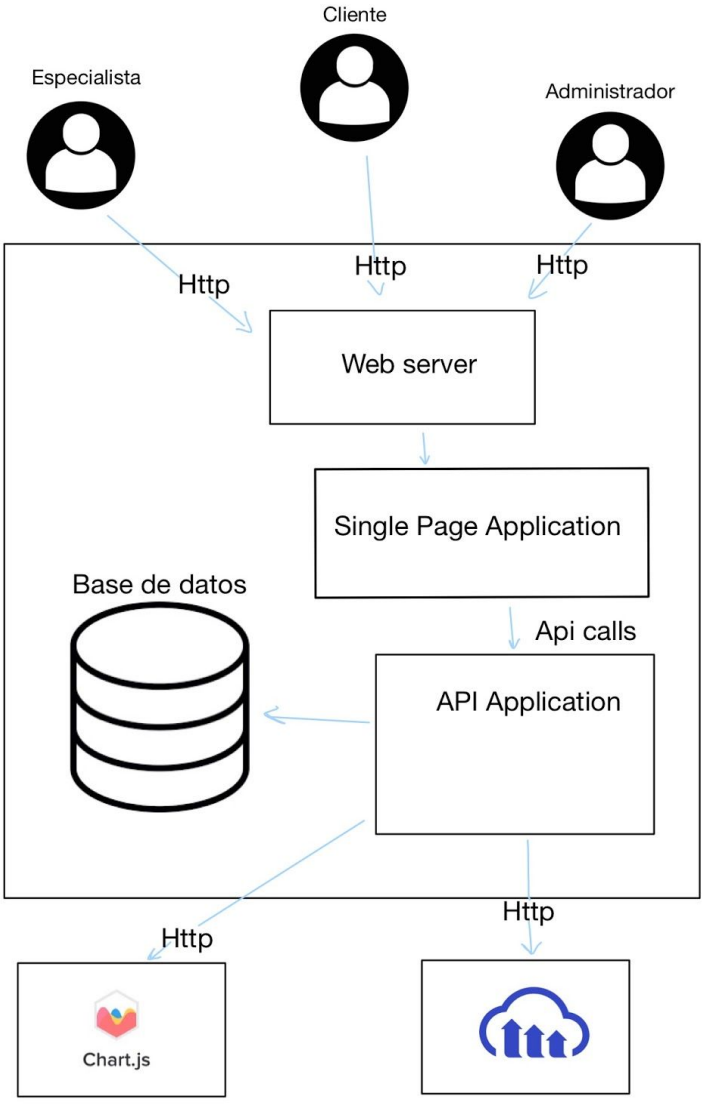
4.1 Nivel de contexto

En el diagrama de contexto podemos apreciar 3 usuarios principales de la página. El cliente , el administrador y el especialista. Además hay 2 subsistemas externos con los cuales interactuar, cloudinary y charts.



4.2 Nivel de contenedores

Los contenedores representan los grandes subsistemas que dan forma al sistema que en este caso sería Gogym. Aquí se pueden ver ya algunas decisiones tecnológicas de alto nivel que tienen que ver con la forma de separar estas grandes unidades y los estándares usados para comunicarlas. Las flechas pueden incluir ahora los protocolos y estándares HTTPS y JSON en el caso del acceso a la API.



4.3 Nivel de componentes

Un componente representa una parte modular de un sistema, que encapsula su contenido. En el caso de nuestro proyecto se crean grandes componentes que

tienen dentro de ellos funcionalidades más pequeñas. Esto consiste en revisar las partes al interior de los contenedores del modelo.

4.4 Nivel de código

Para programar el proyecto se ocupó Node.js y koa, para el cliente Para el cliente, deberán usar HTML, CSS y JavaScript. A nivel de código se explica con mayor profundidad a lo largo de la documentación. Además se incluye un modelo de datos con todas las entidades y sus respectivos atributos.

Los principales usuarios son:

Administrador:

Email: admin@gogym.cl
Clave: admin

Entrenador:

Email: entrenador@gogym.cl
Clave: admin

5. Modelo de Datos

Entidades

Users: Email, contraseña, nombre, apellido, rut, tipo usuario, imagen
Membership: id usuario, tipo, inicio, término
Specialist: id usuario, rol, telefono, descripcion
Appointment: id especialista, id cliente, tipo consulta, fecha, descripción, lugar
HealthProfile: id usuario, fecha nacimiento, nivel, género, altura, peso, porcentaje grasa, telefono de emergencia y descripción
Conversations: id usuario 1, id usuario 2
Messages: id conversacion, id usuario origen, mensaje
Gym: dirección, número telefónico, capacidad y descripción
Room: id gimnasio, tipo, capacidad, imagen
Event: nombre, descripcion, días, hora inicio, hora término, id sala, imagen
EventInscription: id evento, id usuario
Machine: id gimnasio, nombre, descripción, disponible, imagen, tipo
User Machine: id usuario, id máquina fecha, duración, estado

Relaciones entre modelos

Usuario:
- tiene un especialista
- tiene muchos appointments
- tiene muchas fichas de salud
- tiene muchas conversaciones
- tiene muchas reservas de máquinas

Gimnasio
- tiene muchos usuarios
- tiene muchas membresías
- tiene muchas salas
- tiene muchas máquinas
- tiene muchos eventos

Membresía
- pertenece a un usuario
- pertenece a un gimnasio

Especialista
- pertenece a un usuario

Cita con especialista
- pertenece a un especialista
- pertenece a un usuario (puede no estar asignado si está disponible)

Ficha de salud
- pertenece a un usuario

Conversación
- pertenece a 2 usuarios (dos campos distintos con belongsTo)

- tiene muchos mensajes

Mensajes
- pertenece a una conversación
- pertenece a un usuario (usuario de origen)

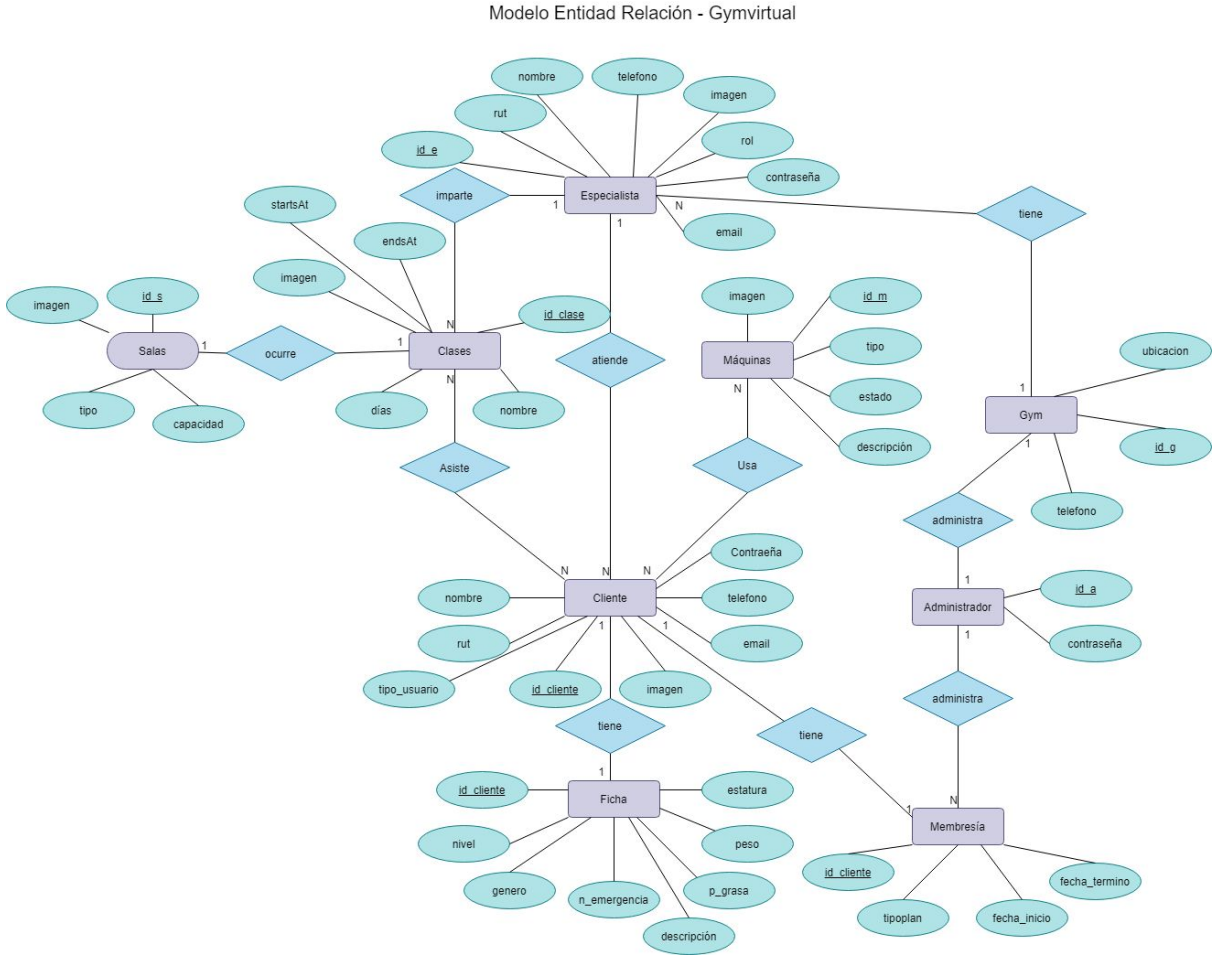
Sala
- pertenece a un gimnasio
- tiene muchos eventos

Eventos
- pertenece a una sala
- pertenece a un usuario
- tiene muchos usuarios (a través de event inscriptions)

Máquina
- pertenece a un gimnasio
- tiene muchas reservas

Reserva de máquinas
- pertenece a una máquina
- pertenece a un usuario (no está asignado si está disponible)

El siguiente diagrama también muestra las relaciones entre las entidades



6. Descripción de procesos y servicios ofrecidos por la aplicación

6.1 A continuación se describen los principales servicios o tareas que ofrece la aplicación.

Usuarios
Cada usuario tiene un mail y una contraseña que permiten iniciar sesión. Además cada usuario tiene un tipo de usuario que puede ser Administrador (0), Cliente (1), Entrenador (2) o nutricionista (3). Para agregar un tipo de usuario distinto solo hay que definir un número y agregar sus nuevas funcionalidades. Además, cada usuario tiene información básica como nombre, apellido, rut y foto de perfil.

Sesión
Las sesiones fueron implementadas con “koa-session”. Estas se crean al iniciar sesión con el mail y contraseña. Al estar logueado cada usuario puede interactuar con los distintos componentes de la aplicación.

Chat
El chat permite mantener conversaciones entre clientes y especialistas o administradores. Los clientes no pueden crear conversaciones entre ellos, pero, los otros tipos de usuarios pueden crear conversaciones con cualquier usuario.

Salas
Las salas son los espacios que tiene disponible el gimnasio para crear distintos eventos. Tienen una capacidad máxima y una descripción.

Eventos
Los eventos son creados por un especialista o administrador en particular. Permite que los clientes sepan qué actividades pueden realizar con un instructor. Pueden ser actividades únicas o recurrentes. Los clientes pueden inscribirse y desuscribirse a los eventos. Una vez inscritos pueden ver todas sus inscripciones en su perfil

Citas con Especialistas
Los clientes pueden pedir citas con especialistas para lograr las metas más rápido y estar en forma. Los clientes pueden agendar y desagendar la cita. Luego pueden ver todas sus consultas en su perfil para llevar registro de ellas. Además le pueden dejar un comentario al especialista al momento de reservar.
Los administradores pueden crear citas para cualquier especialista. En cambio un especialista solo puede crear y eliminar las propias.

Ficha de Salud
Luego de una cita con especialista tu o el especialista pueden crear una nueva ficha con tu información de salud como peso, porcentaje de grasa o músculo y una descripción de los objetivos y cómo realizarlos. Luego el cliente puede ver su progreso mes a mes.

Máquinas

Las máquinas son los implementos más importantes de un gimnasio. Los clientes pueden revisar cuales son las máquinas disponibles y sus características.

Reservas de máquinas
El administrador puede definir los horarios en que se pueden usar las máquinas y la duración que tendrá cada sesión. Luego cada cliente la puede reservar y así asegurarse que tendrá un espacio seguro para entrenar. Al reservar una máquina cambia el campo *userId* y *status*. El cliente que reserva también puede cancelar la reserva, lo que borra la información de *userId* y *status*.

Membresías
Cada cliente tiene una membresía donde está la información de pago y otros. Esta tiene una fecha de inicio y término. El administrador puede consultar esta información y administrar fácilmente el estado de cada cliente.

Gimnasio
Actualmente la aplicación fue creada con un solo gimnasio, pero esto no quiere decir que no pensamos en esto. La arquitectura de la aplicación está modelada para agregar fácilmente varios locales. Solo es necesario agregar los datos de a qué gimnasio pertenecen las salas, máquinas , membresías de los clientes, y citas con especialistas. Los campos de id ya están creados en las tablas de datos.

6.2 React
Se describe los componentes que están implementados en el fronted de la aplicación con React. Esto permite una aplicación más amigable con el usuario.

1. **Validación del formulario**
   El formulario para crear un nuevo usuario es parte del fronted de la aplicación. Este formulario tiene una validación en tiempo real del mail y la contraseña. El mail se valida con regex y la contraseña debe ser de mínimo 6 caracteres. En caso de que esto no se cumple se despliega un mensaje en rojo.
2. **Filtro de máquinas**
   Un gimnasio puede tener cientos de máquinas de distintos tipos. Para el usuario puede ser incómodo revisar cual de todas quiere reservar si solo está buscando una en particular. Con este filtro puede seleccionar por tipo de máquina (ej Cardio, Tren superior o otro). Este filtro está implementado en el fronted. La aplicación tiene una lista de todas las máquinas y muestra solo las pedidas sin necesidad de cargarlas de nuevo.
3. **Chat**
   El chat que disponen los clientes para comunicarse con los especialistas o el administrador, no necesita recargar la página completa cada vez que se envía un mensaje. Si tiene que recargar la sección de los mensajes porque no se implementó con threads.

7. Documentación técnica - Especificación API

Se indica el propósito y breve descripción de cada ruta de la API, con su prototipo indicando argumentos (nombre, tipo, propósito de cada uno en **body**) y respuesta (**response**).
Se indican las rutas para autenticación (Login), Registro de nuevo usuario, eventos, usuarios, fichas de salud y máquinas. Las tres primeras rutas son públicas, el resto requieren autentificación obtenida en Login.
Rutas públicas: Las siguientes rutas no requieren autorización

1. Login

Funcionalidad: Obtener el token de un usuario para acceder a las rutas privadas
Ruta: https://sleepy-brook-55941.herokuapp.com/api/auth
Método: POST
body:

```
{
   "password": "123456",
   "email": "cliente1@gogym.cl"
}
```

response:

```
{
   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOjE2LCJpYX"
}
```

2. Register

Funcionalidad: Crear un nuevo usuario para acceder a la aplicación y API
Ruta: https://sleepy-brook-55941.herokuapp.com/api/register
Método
body:

```
{
   "name": "User",
   "password": "123456",
   "lastname": "New",
   "email": "newuser@gogym.cl",
   "rut": "11111111-1"
}
```

Response:

```
{
   "state": 201
}
```

3. Events

Funcionalidad obtener los eventos disponibles organizadas por el gimnasio
Ruta: https://sleepy-brook-55941.herokuapp.com/api/events
Método: GET
Response:


```
{
    {
        "id": 1,
        "name": "Crossfit",
        "description": "Dirigidas a todas las personas to...",
        "days": "Lunes, Miércoles y Viernes",
        "startsAt": "10:00:00",
        "endsAt": "11:00:00",
        "eventUrl": "http://localhost:3000/api/events/1"
    },
    {
        "id": 3,
        "name": "Ashtanga Yoga",
        "description": "Clases guiadas de la primera serie de asht...",
        "days": "Martes y Jueves",
        "startsAt": "19:00:00",
        "endsAt": "20:00:00",
        "eventUrl": "http://localhost:3000/api/events/3"
    },
}
```


Ruta: https://sleepy-brook-55941.herokuapp.com/api/events/id
Método: GET
Response:


```
{
    "id": 1,
    "name": "Crossfit",
    "description": "Dirigidas a todas las personas todas...",
    "days": "Lunes, Miércoles y Viernes",
    "startsAt": "10:00:00",
    "endsAt": "11:00:00",
    "roomId": 1,
    "image": "gbojhyzqqtj2sea7vnau",
}
```


Rutas privadas: Requieren el token de cada usuario en el header del request

4. Users

Funcionalidad: Obtener una lista de todos los usuarios
Ruta: http://sleepy-brook-55941.herokuapp.com/api/users/all
Método: GET

Response:


```
{
    {
        "id": 2,
        "name": "Entrenador",
        "lastname": "Gogym",
        "email": "entrenador@gogym.cl",
        "rut": "11111111-2",
        "user_type": 2,
        "image": "default-user_nmftx2",
        "createdAt": "2020-12-01T03:24:32.127Z",
        "updatedAt": "2020-12-01T03:24:32.127Z"
    },
    {
        "id": 3,
        "name": "Luis",
        "lastname": "Arancibia",
        "email": "entrenador1@gogym.cl",
        "rut": "11111111-3",
        "user_type": 2,
        "image": "default-user_nmftx2",
        "createdAt": "2020-12-01T03:24:32.127Z",
        "updatedAt": "2020-12-01T03:24:32.127Z"

    }
}
```


Funcionalidad: Obtener información personal de la cuenta
Ruta: https://sleepy-brook-55941.herokuapp.com/api/users/me
Método: GET
Response:


```
{
    "id": 1,
    "name": "Admin",
    "lastname": "Gogym",
    "email": "admin@gogym.cl",
    "rut": "11111111-1",
    "user_type": 0,
    "image": "default-user_nmftx2",
    "createdAt": "2020-12-01T03:24:32.127Z",
    "updatedAt": "2020-12-01T06:32:56.102Z"
}
```


Funcionalidad: Cambiar contraseña u otro de un usuario. Solo puedes cambiar información personal, no puedes cambiar el tipo de usuario. El administrador si puede.
Ruta: http://sleepy-brook-55941.herokuapp.com/api/users/:id
Método: PATCH

Body:


```
{
   "password": "123456",
   "email": "admin@gogym.cl"
}
```


Response:


```
{
   "user": {
        "id": 1,
        "name": "Admin",
        "lastname": "Gogym",
        "email": "admin@gogym.cl",
        "rut": "11111111-1",
        "user_type": 0,
        "image": "default-user_nmftx2"
   }
}
```


5. Rooms

Funcionalidad:
Ruta: http://sleepy-brook-55941.herokuapp.com/api/rooms/
Método: GET
Response:


```
{
  {
        "id": 1,
        "gym_id": 1,
        "type": "Sala Principal",
        "capacity": 20,
        "image": null,
        "roomUrl": "http://localhost:3000/api/rooms/1"
   },
   {
        "id": 2,
        "gym_id": 2,
        "type": "Sala Mediana",
        "capacity": 15,
        "image": null,
        "roomUrl": "http://localhost:3000/api/rooms/2"
   },
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/rooms/:id
Método: GET
Response:

```
{
   "id": 1,
   "gym_id": 1,
   "type": "Sala Principal",
   "capacity": 20,
   "image": null
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/rooms
Método: POST
Body:


```
{
   "type": "Sala Consultas",
   "capacity": "5"
}
```


Response:


```
{
   "content": "sala creada"
}
```


Funcionalidad:
Ruta: http://sleepy-brook-55941.herokuapp.com/api/rooms/:id
Método: PATCH
Body:


```
{
   "capacity": "12"
}
```


Response:


```
{
   "room": {
       "id": 1,
       "gym_id": 1,
       "type": "Sala Principal",
       "capacity": "12",
       "image": null,
       "createdAt": "2020-12-01T03:24:32.137Z",
       "updatedAt": "2020-12-02T17:58:30.285Z"
   }
}
```


Funcionalidad: Eliminar una sala del gimnasio
Ruta: http://sleepy-brook-55941.herokuapp.com/api/rooms/:id
Método: DELETE
Response:


```
{
   "content": "sala eliminada"
}
```

6. Healthprofiles

Ruta: http://sleepy-brook-55941.herokuapp.com/api/healthprofiles
Método: GET
Response:


```
{
  {
       "id": 9,
       "user_id": 1,
       "birth": "2020-12-18T00:00:00.000Z",
       "level": null,
       "gender": 0,
       "height": 165,
       "weight": 55,
       "fat_percentage": 30,
       "emergency_number": 1234567,
       "description": "Buen progreso!",
       "healthprofileUrl": "http://localhost:3000/api/healthprofiles/9"
   },
   {
       "id": 10,
       "user_id": 3,
       "birth": "2020-12-22T00:00:00.000Z",
       "level": null,
       "gender": 1,
       "height": 165,
       "weight": 58,
       "fat_percentage": 30,
       "emergency_number": 1234567,
       "description": "Sigue así, recuerda mantener una dieta equilibrada",
       "healthprofileUrl": "http://localhost:3000/api/healthprofiles/10"
   }
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/healthprofiles/:id
Método: GET
Response:


```
{
   "id": 12,
   "user_id": 3,
   "gender": 1,
   "height": 165,
   "weight": 123,
   "fat_percentage": 50,
   "emergency_number": 1234567,
```

```
    "description": "comentario",
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/healthprofiles
Método: POST
Body:


```
{
    "user_id": 3,
    "birth": "2020-12-24T00:00:00.000Z",
    "level": 5,
    "gender": 1,
    "height": 165,
    "weight": 123,
    "fat_percentage": 50,
    "emergency_number": 1234567,
    "description": "comentario"
}
```


Response:


```
{
    "content": "ficha creada"
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/healthprofiles/:id
Método: PATCH
Body:


```
{
    "emergency_number": 7777777
}
```


Response:


```
{
    "healthprofile": {
        "id": 12,
        "user_id": 3,
        "birth": "2020-12-24T00:00:00.000Z",
        "gender": 1,
        "height": 165,
        "weight": 123,
        "fat_percentage": 50,
        "emergency_number": 7777777,
        "description": "comentario",
    }
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/healthprofiles/:id

Método: DELETE
Response:


```
{
   "content": "ficha eliminada"
}
```


7. Machines

Ruta: http://sleepy-brook-55941.herokuapp.com/api/machines
Método: GET
Response:


```
{
        "id": 1,
        "name": "Trotadora",
        "description": "Maquina de ejercicio cardiovascular, simula trote en
espacio abierto. Múltiples velocidades.",
        "available": true,
        "image": "txmbg8s794toj6chdffi",
        "tipo": "Cardio",
        "machineUrl": "http://localhost:3000/api/machines/1"
    },
    {
        "id": 2,
        "name": "Remadora",
        "description": "Máquina de ejercicio remadora",
        "available": true,
        "image": "rsh1frukkbxa7lap9lwg",
        "tipo": "Cardio",
        "machineUrl": "http://localhost:3000/api/machines/2"
    },
}
```


Ruta: http://sleepy-brook-55941.herokuapp.com/api/machines/:id
Método: GET
Response:


```
{
   "id": 3,
   "name": "Leg Press",
   "description": "Prensa de piernas con carga de discos. Capacidad de 880
lbs con traba manual de seguridad y almacenamiento posterior de discos.",
   "available": true,
   "image": "j8it1de3gae7eotcvxu5",
   "tipo": "Cardio"
}
```

Ruta: http://sleepy-brook-55941.herokuapp.com/api/machines
Método: POST
Body:

```
{
   "gym_id": 1,
   "name": "New",
   "description": "Nueva Máquina que perminte ...",
   "available": false,
   "image": "j8it1de3gae7eotcvxu5",
   "tipo": "Cardio"
}
```

Response:

```
{
   "content": "máquina creada"
}
```

Ruta: http://sleepy-brook-55941.herokuapp.com/api/machines/:id
Método: DELETE
Response:

```
{
   "content": "máquina eliminada"
}
```