src/services/LocationService.js

```
// Dados dos locais (simulando uma API)
const locaisData = [
    { id: 1, nome: "Extensão Centro", x: 28, y: 60, andar: "terreo" },
    { id: 2, nome: "Bloco A Didático/Administrativo", x: 44, y: 46, andar: "terreo" },
    { id: 3, nome: "Bloco B Laboratórios/Administrativo", x: 49, y: 46, andar: "terreo" },
    { id: 4, nome: "Galpão de laboratórios", x: 49, y: 29, andar: "terreo" },
    { id: 5, nome: "Bloco C", x: 45, y: 49, andar: "primeiro" }, // Exemplo 1º andar
    { id: 6, nome: "Bloco D", x: 49, y: 49, andar: "primeiro" }, // Exemplo 1º andar
    { id: 7, nome: "Restaurante Universitário (RU)", x: 60, y: 66, andar: "terreo" },
    { id: 8, nome: "Prefeitura do Campus", x: 37, y: 48, andar: "terreo" },
    { id: 9, nome: "Parque Tecnológico", x: 72, y: 82, andar: "terreo" },
    // Adicione outros locais conforme necessário
];

// Dados dos waypoints (simulando uma API)
const campusWaypointsData = [
    { id: "w1", x: 40, y: 60, connections: ["w2", "w3"], andar: "terreo" }, // Entrada principal
    { id: "w2", x: 42, y: 52, connections: ["w1", "w4", "w5", "e1"], andar: "terreo" }, // Bifurcação principal (Adicionada conexão com escada e1)
    { id: "w3", x: 28, y: 60, connections: ["w1"], andar: "terreo" }, // Extensão Centro
    { id: "w4", x: 37, y: 48, connections: ["w2", "w6"], andar: "terreo" }, // Prefeitura do Campus
    { id: "w5", x: 44, y: 46, connections: ["w2", "w7"], andar: "terreo" }, // Bloco A
    { id: "w6", x: 37, y: 42, connections: ["w4", "w7"], andar: "terreo" }, // Corredor oeste
    { id: "w7", x: 49, y: 46, connections: ["w5", "w6", "w8", "w9"], andar: "terreo" }, // Bloco B
    { id: "w8", x: 49, y: 29, connections: ["w7"], andar: "terreo" }, // Galpão de laboratórios
    { id: "w9", x: 54, y: 52, connections: ["w7", "w10"], andar: "terreo" }, // Corredor leste
    { id: "w10", x: 60, y: 66, connections: ["w9", "w11"], andar: "terreo" }, // Restaurante Universitário (ajustado w11)
    { id: "w11", x: 72, y: 82, connections: ["w10"], andar: "terreo" }, // Parque Tecnológico

    // Waypoints do primeiro andar
    { id: "w101", x: 45, y: 49, connections: ["w102", "w103"], andar: "primeiro" }, // Bloco C
    { id: "w102", x: 49, y: 49, connections: ["w101"], andar: "primeiro" }, // Bloco D
    { id: "w103", x: 45, y: 55, connections: ["w101", "e1"], andar: "primeiro" }, // Escada (Adicionada conexão com escada e1)

```

```
    // Conexões entre andares (exemplo de escada)
    // 'conectaAndar' aponta para o ID do waypoint no *outro* andar
    // A conexão normal em 'connections' é para o waypoint *neste* andar que leva à escada
    { id: "e1", x: 45, y: 55, connections: ["w2"], andar: "terreo", conectaAndar: "w103" }, // Escada no térreo, conectada a w2, leva para w103 (1º andar)
    // Nota: O waypoint w103 no primeiro andar também se conecta a 'e1' implicitamente pela regra de conexão entre andares.
    // Ou pode-se adicionar explicitamente em w103: connections: ["w101", "e1"], andar: "primeiro", conectaAndar: "e1" (do terreo) - depende da implementação do algoritmo
];

// Dados dos Andares
const floorsData = [
  {
    id: "terreo",
    name: "Térreo",
    // Substitua pela URL real ou importe a imagem se estiver local
    image: "https://mmkbapkywiapbirvfuez.supabase.co/storage/v1/object/sign/imagem/campusRussas.jpg?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InN0b3JhZ2UtdXJsLXNpZ25pbmcta2V5X2JjYWQzOGJILTA1ZjEtNDBh-NC1hNGQxLTNiN2VlYWM1ZWI5YyJ9.eyJ1cmwiOiJpbWF nZW0vY2FtcHVzUnVzc2FzLmpwZyIsImlhdCI6MTc0NDYyNzY5MiwiZXhwIjoxNzQ1MjMyNDkyfQ.hNC_m0h1cEJ9Kw63Zpq-tvCn5yXweB-hxkdAHCOn7q0",
  },
  {
    id: "primeiro",
    name: "1º Andar",
     // Substitua pela URL real ou importe a imagem se estiver local
    image: "https://mmkbapkywiapbirvfuez.supabase.co/storage/v1/object/sign/imagem/campusRussas.jpg?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6InN0b3JhZ2UtdXJsLXNpZ25pbmcta2V5X2JjYWQzOGJILTA1ZjEtNDBh-NC1hNGQxLTNiN2VlYWM1ZWI5YyJ9.eyJ1cmwiOiJpbWF nZW0vY2FtcHVzUnVzc2FzLmpwZyIsImlhdCI6MTc0NDYyNzY5MiwiZXhwIjoxNzQ1MjMyNDkyfQ.hNC_m0h1cEJ9Kw63Zpq-tvCn5yXweB-hxkdAHCOn7q0", // Usando a mesma imagem como exemplo
  },
];



/**
 * Busca os dados dos locais.
 * @returns {Promise<Array>} Uma promessa que resolve com a lista de locais.
 */
export const fetchLocais = async () => {
```

```
  // Simula um delay de API
  await new Promise(resolve => setTimeout(resolve, 50));
  console.log("Locais carregados:", locaisData);
  return locaisData;
};

/**
 * Retorna os dados dos waypoints.
 * @returns {Array} A lista de waypoints.
 */
export const getWaypoints = () => {
  console.log("Waypoints carregados:", campusWaypointsData);
  return campusWaypointsData;
};

/**
 * Retorna os dados dos andares.
 * @returns {Array} A lista de andares.
 */
export const getFloors = () => {
  return floorsData;
}
```