

# Especificação de Requisitos da API – Fundação Cristiano Varella (FCV)

---

## 1. Objetivo Geral

---

A API tem como objetivo principal **orientar pacientes, acompanhantes e visitantes dentro do hospital**, oferecendo acesso rápido e contextualizado a informações essenciais com base na localização. O foco é **reduzir a ansiedade e o tempo de deslocamento** em um ambiente sensível como um hospital oncológico, promovendo uma experiência mais **simples, segura e humanizada**.

---

## 2. Requisitos Funcionais

---

### 2.0 Cadastro Admin

- A API deve permitir cadastrar contas para **admins** que serão usadas para atualizar, apagar e criar novas localidades, eventos, vagas e notícias.

### 2.1. Localização

- A API deve permitir consultar a localização de **exames**, **setores** ou **bloco**s por meio de um único endpoint ( `GET /localizacao` ).
- A requisição deve aceitar pelo menos um dos seguintes parâmetros: `exame`, `setor` ou `bloco` (por nome ou ID).
- A resposta deve incluir:
  - Nome do item buscado
  - Setor e bloco associados
  - Andar
  - Coordenada geográfica (interna)
  - Referências visuais e textuais (imagens e descrições) para orientação

### 2.2. Eventos

- A API deve retornar eventos futuros, em andamento ou encerrados ( `GET /eventos` ).
- Deve permitir filtragem por:

- `status` ( `futuros`, `em_andamento`, `encerrado` )
- `dias` (eventos nos próximos N dias)
- Deve permitir consultar um evento específico por ID ( `GET /eventos/:id` ).
- **(ADMIN)**: Permitir criar, atualizar e excluir eventos via `POST`, `PATCH` e `DELETE`.

## 2.3. Notícias

- A API deve retornar notícias publicadas nos últimos 30 dias ( `GET /noticias` ).
- Deve permitir filtragem por:
  - `recentes` (últimos N dias)
  - `bloco`, `setor` ou `exame` relacionado
- Deve permitir consultar uma notícia específica por ID ( `GET /noticias/:id` ).
- **(ADMIN)**: Permitir criar, atualizar e excluir notícias.

## 2.4. Informações Institucionais

- A API deve retornar informações institucionais (missão, visão, valores, compliance) via `GET /sobre`.
- **(ADMIN)**: Permitir atualizar essas informações via `PATCH /sobre`.

## 2.5. Vagas de Emprego

- A API deve retornar vagas ativas por padrão ( `GET /vagas` ).
- Deve permitir filtragem por:
  - `encerradas` (booleano)
  - `cargo`
  - `contrato` ( `CLT`, `Estagio`, `PJ` )
  - `data_publicacao`
- Deve permitir consultar uma vaga específica por ID ( `GET /vagas/:id` ).
- **(ADMIN)**: Permitir criar, atualizar e excluir vagas.

---

## 3. Requisitos Não Funcionais

### 3.1. Desempenho

- A API deve responder a requisições de leitura ( `GET` ) em **menos de 1 segundo** em condições normais de carga.
- Deve suportar múltiplas requisições simultâneas (múltiplos usuários do hospital)

Deve suportar **múltiplos usuários simultâneos** (público geral do hospital).

### 3.2. Confiabilidade

- Todos os dados devem ser consistentes e atualizados em tempo hábil.
- A API deve retornar códigos de status HTTP apropriados (ex: `400` se nenhum parâmetro for fornecido em `/localizacao`).

### 3.3. Manutenibilidade

- A estrutura dos endpoints e respostas deve ser clara e documentada, facilitando futuras atualizações.

### 3.4. Compatibilidade

- A API deve usar **formato ISO 8601** para todas as datas.
- Parâmetros de busca (ex: `bloco`, `setor`) devem ser **case-insensitive**.

---

## 4. Requisitos de Segurança

- Endpoints de escrita ( `POST` , `PATCH` , `DELETE` ) são **restritos a administradores**.
- Requisições administrativas devem incluir **autenticação válida** (não detalhada no documento, mas exigida).
- Dados sensíveis (como coordenadas internas) não devem expor informações de segurança física do hospital.
- **Imagens utilizadas como referência** (por exemplo, em localizações, notícias ou setores) **não devem conter pessoas identificáveis**. Caso haja pessoas nas fotos, **seus rostos devem ser previamente borrados ou anonimizados** antes da publicação.

---

## 5. Requisitos de Usabilidade

- As respostas devem incluir **referências visuais e textuais** (imagens e descrições) para facilitar a navegação de usuários leigos.
- A API deve ser intuitiva para integração com aplicativos móveis ou quiosques de autoatendimento.
- "É necessário fornecer pelo menos um dos parâmetros..."

## 6. Observações Gerais

---

- Todos os endpoints de **leitura** ( **GET** ) são **públicos**.
- Endpoints de **escrita** exigem **permissão de administrador**.
- A API prioriza **contextualização por localização**, integrando informações de **eventos**, **notícias**, **vagas** e **serviços** ao espaço físico do hospital.