

Projeto Integrador I – Documentação Técnica

Responsável: Emanuel Filipe da Silva –  [GitHub](#) -  [LinkedIn](#)

Nome do sistema: BeyondApp

Versão: 0.01

Apresentação

Meu sistema visa unir a precisão, organização e usabilidade do Google Maps com a experiência visual e técnicas de marketing dos influencers digitais no Youtube, Instagram e TikTok.

O Problema

Para o consumidor: Quando usamos o Google Maps para procurarmos locais, o sistema realiza uma geolocalização perfeita, porém entrega apenas notas, opiniões frias, e fotos muitas vezes desatualizadas. O Instagram, por outro lado, pode possuir fotos, menções recentes de outros consumidores e Reels feitas como “publi” local, mas geolocalização péssima (isso vale para o TikTok e Youtube Shorts também).

Para o comerciante: É exaustivo e caro ter que atualizar o cardápio, manual do hóspede, ou guia virtual em vários aplicativos diferentes ao mesmo tempo.

A Solução

Criar um "**Mapa Vivo**" e social. Um Mini aplicativo onde o usuário explora a própria cidade através de vídeos curtos, fotos, e comentários (Reels/TikTok/Shorts) utilizando o Maps como distribuição e organização dos posts.

Você abre o mapa e, em vez de ver apenas fotos desatualizadas e estrelas, você assiste à yitrine real e atualizada daquele comércio a 5, 10, 100km de distância!

O Grande Diferencial

Automação zero-esforço. O dono do estabelecimento não precisa postar nada de novo no nosso app. Ele conecta as redes sociais dele uma única vez, e nosso

sistema puxa automaticamente os vídeos e fotos desejados para o mapa geolocalizado.

O cliente, também usuário da extensão, utiliza o Maps para encontrar seu estabelecimento, mas com adicionais das fotos e vídeos curtos, junto a outra função do Maps "Como chegar" e/ou "Rotas", que guia o cliente até a porta da loja.

Segurança e qualidade. Para garantir que a extensão seja um ambiente limpo, seguro e focado em comércio, utilizamos I.A. e, se necessário, verificações manuais, para moderar as imagens/vídeos em tempo real, bloqueando spam ou conteúdos indevidos antes mesmo de irem ao ar.

1. Descrição Técnica do Sistema

O sistema é uma plataforma de **Enriquecimento de Localização Geoespacial**. Ele atua como um agregador de conteúdo visual (vídeos curtos e fotos), vinculando mídias extraídas via API (Instagram Graph, TikTok e Google Places) a coordenadas geográficas específicas (Latitude/Longitude e *Place_ID*).

O sistema possui uma arquitetura multiplataforma:

- **Mobile/PC App:** Um *Progressive Web App* (PWA) ou aplicativo nativo (React Native/Flutter) focado no consumo de mídia por geolocalização e rotas.
- **Extensão Web (Browser):** Um *plug-in* que atua em segundo plano no navegador do usuário. Quando ele acessa o site do Google Maps no PC, a extensão injeta (via manipulação de DOM) os vídeos e fotos do nosso banco de dados diretamente na barra lateral do local pesquisado.

2. Perfis de Usuários (Atores do Sistema)

O sistema possui três níveis distintos de permissão de acesso:

1. **Consumidor (B2C):** Usuário final que busca locais, assiste aos vídeos e faz avaliações.
2. **Comerciante (B2B):** Dono do estabelecimento que gerencia a sincronização de suas redes sociais com o mapa.

3. **Administrador (SysAdmin):** Equipe interna da plataforma responsável por manter a ordem, moderar conteúdos sensíveis e gerenciar denúncias.

3. Funcionalidades por Tela e Permissões

Abaixo está o mapeamento do que cada perfil pode ver e fazer dentro das aplicações:

Módulo do Consumidor (App Mobile/PC e Extensão)

Permissões: Leitura de dados públicos, gravação do próprio conteúdo, permissão de localização (GPS).

- **Tela Inicial (Mapa / Discovery):**
 - Visualizar o mapa com raio ajustável (ex: 1km a 50km).
 - Filtrar locais por categoria (Alimentação, Vestuário, Postos, etc.).
 - Navegar no formato "Feed/Swipe" (arrastar para cima/lados) para ver os vídeos curtos dos locais próximos.
- **Tela de Detalhes do Local:**
 - Visualizar o conteúdo sincronizado do comerciante.
 - Acessar botão "Como Chegar" (integração com Google Maps).
- **Tela de Avaliação (Review):**
 - Fazer upload de fotos e vídeos próprios no local.
 - *Feature:* Marcar o checkbox "Postar também no meu Instagram" (requer autenticação via OAuth 2.0).

Módulo do Comerciante (Dashboard Web/App)

Permissões: Gerenciamento do próprio local (Place_ID), leitura de métricas, integração de APIs restritas.

- **Tela de Onboarding / Interações:**
 - Reivindicar a propriedade do local (via Google Business API).
 - Conectar contas do Instagram Profissional e TikTok via botões de autorização OAuth.
- **Painel de Mídia (Media Manager):**

- Visualizar tudo o que o sistema puxou automaticamente das redes.
 - Ocultar mídias específicas que ele não quer que apareçam no mapa (ex: posts de promoções antigas).
- **Tela de Insights (Estatísticas):**
 - Visualizar métricas: "Quantas pessoas clicaram em 'Como Chegar'", visualizações de vídeos via mapa.

Módulo do Administrador (Painel Web Interno)

Permissões: Acesso total (Super Admin), alteração de status de conteúdo, banimento de usuários.

- **Tela de Moderação (Fila de Revisão):**
 - Visualizar mídias retidas pela Inteligência Artificial.
 - Aprovar ou rejeitar permanentemente conteúdos sinalizados como SPAM ou impróprios (Violência, Nudez, etc.).
- **Tela de Gestão de Usuários e Locais:**
 - Suspender contas que violem os Termos de Uso.
 - Remover locais duplicados ou fraudulentos.

4. Condições e Regras de Negócio (Back-end)

Para que o sistema funcione corretamente, ele obedece a estas lógicas de bastidores:

- **Condição de Sincronização:** O sistema não puxa os dados das redes sociais toda vez que alguém abre o app (isso esgotaria a cota das APIs). Um CRON Job (rotina agendada) no servidor roda de 12 em 12 horas para buscar postagens novas dos comerciantes e salvar no banco de dados.
- **Condição de IA (Segurança):** Todo upload feito diretamente no app (seja pelo consumidor ou comerciante) passa por um *Webhook* de IA (Ex: AWS Rekognition). Se o índice de "Conteúdo Inseguro" for maior que 85%, a imagem é bloqueada automaticamente. Se for entre 50% e 84%, vai para a fila de revisão manual do Administrador.

Requisitos funcionais

Módulo 1: Consumidor (App Mobile/PC e Extensão Web)

Ações focadas na descoberta geolocalizada e na interação do usuário final.

- **RF-01 | Detecção de Localização:** O sistema deve capturar a geolocalização atual do usuário (latitude e longitude) mediante sua permissão.
- **RF-02 | Busca e Filtros Radiais:** O sistema deve permitir que o usuário filtre estabelecimentos por categorias (ex: Restaurantes, Postos, Vestuário) e defina um raio de busca (ex: 1km a 15km).
- **RF-03 | Visualização de Feed (Discovery):** O sistema deve exibir um feed vertical (estilo *swipe*) ou em grade contendo vídeos curtos e fotos associados aos estabelecimentos próximos.
- **RF-04 | Detalhes do Local:** O sistema deve exibir as informações do estabelecimento (nome, endereço, horário de funcionamento) extraídas da base do Google Places, junto à galeria de mídias sincronizadas.
- **RF-05 | Integração de Rotas:** O sistema deve fornecer um botão "Como chegar", que redireciona o usuário com a rota traçada para o aplicativo de mapas nativo do dispositivo.
- **RF-06 | Publicação de Avaliações:** O sistema deve permitir que o usuário faça o upload de fotos e vídeos próprios, inserindo uma avaliação (nota e texto) em um estabelecimento específico.
- **RF-07 | Compartilhamento Externo (Cross-posting):** O sistema deve oferecer a opção para o usuário compartilhar sua avaliação com foto/vídeo automaticamente no seu próprio feed/stories do Instagram, exigindo autenticação OAuth prévia.

Módulo 2: Comerciante (Dashboard B2B)

Ações focadas na automação e gestão de conteúdo pelo dono do estabelecimento.

- **RF-08 | Reivindicação de Propriedade:** O sistema deve permitir que o comerciante se cadastre e valide a propriedade do seu estabelecimento utilizando a autenticação da *Google Business Profile API*.
- **RF-09 | Conexão de Redes Sociais:** O sistema deve permitir que o comerciante vincule suas contas do *Instagram Professional* e *TikTok* via autenticação OAuth 2.0.

- **RF-10 | Sincronização Automática (CRON):** O sistema deve buscar e importar automaticamente novas postagens (vídeos e fotos) das redes sociais vinculadas do comerciante em intervalos de tempo pré-definidos.
- **RF-11 | Gestão de Mídia:** O sistema deve permitir que o comerciante ative, oculte ou exclua mídias específicas do seu catálogo dentro da plataforma, sem afetar a postagem original na rede social.
- **RF-12 | Painel de Métricas (Insights):** O sistema deve exibir ao comerciante estatísticas de visualizações das suas mídias na plataforma e a quantidade de cliques no botão "Como chegar".

Módulo 3: Motor do Sistema e Inteligência Artificial

Ações que ocorrem em segundo plano (Backend) para garantir a segurança e o funcionamento.

- **RF-13 | Análise de IA em Tempo Real:** O sistema deve enviar todo upload de mídia feito de forma direta (por consumidores) para uma API de Visão Computacional antes da publicação.
- **RF-14 | Bloqueio Automático:** O sistema deve rejeitar automaticamente e alertar o usuário caso a IA identifique a mídia com alta probabilidade de conteúdo explícito, violência ou SPAM.
- **RF-15 | Injeção de Conteúdo (Extensão Web):** A extensão do navegador deve identificar quando o usuário acessa uma URL do Google Maps e injetar dinamicamente as mídias do banco de dados na interface da página, baseando-se no *Place_ID* em tela.

Módulo 4: Administrador (Painel Interno)

Ações focadas na governança da plataforma.

- **RF-16 | Fila de Moderação:** O sistema deve listar em um painel administrativo todas as mídias que a IA marcou como "suspeitas" (probabilidade intermediária de violação de regras).
- **RF-17 | Revisão Manual:** O sistema deve permitir que o Administrador aprove ou rejeite permanentemente as mídias retidas na fila de moderação.
- **RF-18 | Gestão de Contas:** O sistema deve permitir que o Administrador suspenda ou bane contas de Consumidores ou Comerciantes por violação dos Termos de Uso.

Requisitos não funcionais

Categoria 1: Desempenho e Eficiência (Performance)

Focam no tempo de resposta e no consumo de recursos, cruciais já que lidamos com vídeos e mapas.

- **RNF-01 | Tempo de Carregamento de Mídia:** O tempo máximo para iniciar a reprodução de um vídeo curto (Reels/TikTok) no feed geolocalizado não deve ultrapassar 2 segundos em conexões 4G/5G padrão.
- **RNF-02 | Paginação de Dados (Lazy Loading):** O sistema deve carregar os locais e as mídias sob demanda (ex: blocos de 10 em 10) conforme o usuário rola a tela (scroll) para evitar sobrecarga de memória no dispositivo mobile.
- **RNF-03 | Otimização de Imagens:** Todas as fotos armazenadas em cache ou no servidor devem ser automaticamente comprimidas para o formato WebP para economizar banda.

Categoria 2: Segurança e Privacidade (Security)

Focam na proteção dos dados dos usuários e das credenciais das APIs, essenciais para estar em conformidade com a LGPD.

- **RNF-04 | Criptografia de Trânsito:** Toda a comunicação entre o aplicativo (Front-end), a extensão e o servidor (Back-end) deve ser feita exclusivamente via protocolo HTTPS (TLS 1.2 ou superior).
- **RNF-05 | Proteção de Chaves de API:** As credenciais das APIs do Google Places, Meta Graph e IA de moderação devem ser armazenadas em variáveis de ambiente (.env) no servidor, jamais expostas no código fonte ou na aplicação cliente.
- **RNF-06 | Autenticação Padronizada:** O login de comerciantes e o vínculo com redes sociais devem utilizar estritamente o protocolo OAuth 2.0, garantindo que o sistema não armazene senhas do Instagram ou TikTok no banco de dados.

Categoria 3: Usabilidade e Acessibilidade (Usability)

Focam na experiência do usuário e na facilidade de uso em diferentes condições.

- **RNF-07 | Design Responsivo (Mobile-First):** A interface do aplicativo web/mobile deve se adaptar fluidamente a diferentes tamanhos de tela (smartphones, tablets e monitores desktop), priorizando a experiência em dispositivos móveis.
- **RNF-08 | Acessibilidade (WCAG):** A interface deve possuir contraste adequado de cores e suporte a leitores de tela nativos (como VoiceOver e TalkBack) nos botões de navegação, atingindo no mínimo o nível AA das diretrizes WCAG.

Categoria 4: Disponibilidade e Confiabilidade (Reliability)

Focam em manter o sistema no ar e lidar com falhas de serviços externos.

- **RNF-09 | Tolerância a Falhas de API (Fallback):** Caso as APIs do Instagram ou TikTok fiquem fora do ar temporariamente, o sistema não deve "quebrar" (crash), mas sim exibir as informações básicas do local (do Google Places) com uma mensagem amigável de que a mídia está indisponível no momento.
- **RNF-10 | Limite de Requisições (Rate Limiting):** O Back-end deve implementar um controle de taxa de requisições para respeitar as cotas diárias gratuitas/pagas da Google Places API, evitando bloqueios do serviço.

Categoria 5: Manutenibilidade e Portabilidade (Maintainability)

Focam na facilidade de dar manutenção no código e evoluir o projeto no futuro.

- **RNF-11 | Arquitetura de API:** O Back-end deve ser desenvolvido seguindo os princípios de uma API RESTful, retornando respostas padronizadas no formato JSON.
- **RNF-12 | Versionamento de Código:** Todo o histórico de desenvolvimento deve ser mantido no Git, utilizando branches separadas para novas funcionalidades (ex: feature/mapa, feature/oauth) e documentado no arquivo README do repositório.

Observações técnicas

1. Linguagens e Frameworks

Para garantir um desenvolvimento ágil e criar um projeto forte para o seu portfólio no GitHub, a stack deve ser moderna e amplamente adotada pelo mercado:

- **Back-end (API Central):** Node.js com TypeScript (usando o framework Express ou NestJS). O Node.js é excelente para lidar com múltiplas requisições assíncronas (como consumir dados do Google e do Instagram ao mesmo tempo).
- **Front-end (Mobile/PWA):** React Native (com Expo) ou Flutter. Permitem escrever o código uma única vez e compilar para Android e iOS, acelerando a entrega do MVP.
- **Front-end (Extensão Web):** JavaScript Vanilla ou React embutido, seguindo o padrão **Manifest V3** dos navegadores baseados em Chromium (Chrome, Edge) para injetar os componentes visuais na página do Google Maps.

2. Banco de Dados e Armazenamento (Storage)

- **Banco de Dados Relacional:** PostgreSQL.
 - *Observação Crítica:* É obrigatório o uso da extensão **PostGIS** acoplada ao PostgreSQL. Ela é o padrão da indústria para dados geoespaciais, permitindo realizar cálculos complexos de raio e distância (ex: *ST_Distance*) direto no banco, com alta performance.
- **Armazenamento em Nuvem (Object Storage):** As imagens de perfil e mídias de upload direto não devem ser salvas no banco de dados. O sistema utilizará AWS S3 ou Cloudinary para hospedar os arquivos, salvando apenas a URL (link) no PostgreSQL.

3. Ecossistema de APIs de Terceiros

O sistema atua como um orquestrador de serviços externos. O consumo das seguintes APIs será necessário:

- **Geolocalização:** Google Places API (para dados e Place_ID) e Google Maps JavaScript API.
- **Redes Sociais:** Instagram Graph API (para leitura de mídias via OAuth) e TikTok for Developers API.
- **Segurança e Moderação:** Google Cloud Vision API ou AWS Rekognition para análise de imagens em tempo real usando Inteligência Artificial.

4. Estratégia de Hospedagem e Backup

- **Cloud Provider:** Para o MVP, a hospedagem do Back-end pode ser feita em plataformas PaaS (Platform as a Service) como Render ou Railway, que facilitam o deploy contínuo integrado ao GitHub.
- **Rotina de Backup:** O servidor de banco de dados deverá ter uma rotina automatizada (CRON Job) que realiza um *dump* (cópia) diário de toda a base de dados na madrugada, enviando o arquivo criptografado para um *Bucket S3* isolado, garantindo a recuperação de desastres (Disaster Recovery).

5. Regras de Negócio e Restrições Arquiteturais Específicas

- **Single Source of Truth (Fonte Única da Verdade):** Para evitar que um mesmo restaurante seja cadastrado duas vezes, o identificador único universal (UUID) de todos os locais no nosso banco de dados deve ser obrigatoriamente o Place_ID fornecido pelo Google.
- **Economia de Storage (Embed vs. Download):** Para mídias puxadas do Instagram e TikTok, o sistema deve priorizar salvar apenas a *URL de Embed* no banco de dados, renderizando o player nativo das redes no Front-end. Isso economiza custos massivos de servidor e evita problemas de direitos autorais, já que o vídeo não é "baixado" e republicado.
- **LGPD e Revogação de Acesso:** O lojista deve ter um botão claro de "Desconectar Instagram" no painel. Ao clicar, o sistema deve não apenas excluir o token de acesso do banco de dados, mas também deletar (via *Soft Delete*) todos os links de mídia associados àquela conta que estavam em cache.

Referências Bibliográficas e Tecnológicas

1. Normas, Leis e Diretrizes de Qualidade

- **ISO/IEC 25010:** Padrão internacional para modelagem e avaliação de qualidade de software (utilizado como base para a categorização dos Requisitos Não Funcionais).
- **WCAG (Web Content Accessibility Guidelines):** Diretrizes de Acessibilidade para Conteúdo Web, mantidas pelo W3C (aplicado aos requisitos de interface e usabilidade).
- **LGPD (Lei Geral de Proteção de Dados Pessoais - Lei nº 13.709/2018):** Base legal para as decisões arquiteturais de privacidade, proteção de dados e revogação de acesso (Soft Delete).
- **Scrum / Kanban:** Metodologias Ágeis de desenvolvimento de software utilizadas para o planejamento e divisão de épicas.

2. Documentações de APIs Oficiais

- **Google Maps Platform:** Documentação técnica da *Google Places API* e *Maps JavaScript API* (para geolocalização, rotas e Place_ID).
- **Meta for Developers:** Documentação da *Instagram Graph API* (para autenticação, permissões e extração de mídia do Instagram Professional).
- **TikTok for Developers:** Documentação oficial para integração e login social.
- **Google Business Profile API:** Documentação para a validação de propriedade de estabelecimentos comerciais.

3. Serviços em Nuvem e Inteligência Artificial

- **AWS (Amazon Web Services):** Manuais do AWS S3 (Object Storage para imagens) e AWS *Rekognition* (Visão computacional para moderação de conteúdo).
- **Google Cloud Platform (GCP):** Documentação técnica da *Cloud Vision API*.

4. Arquitetura, Protocolos e Bancos de Dados

- **OAuth 2.0:** Protocolo padrão da indústria para autorização segura, utilizado nas integrações de login social.
- **Manifest V3:** Arquitetura oficial e documentação do Google Chrome para o desenvolvimento da extensão de navegador.
- **PostgreSQL & PostGIS:** Documentação oficial do banco de dados relacional e de sua extensão espacial (essencial para cálculos de raio de distância e coordenadas nativas).
- **REST (Representational State Transfer):** Padrão de arquitetura de software utilizado para modelar as requisições e respostas do Back-end.

Referências Bibliográficas e Tecnológicas

1. Normas, Leis e Diretrizes de Qualidade

- **ISO/IEC 25010:** Padrão internacional para modelagem e avaliação de qualidade de software (utilizado como base para a categorização dos Requisitos Não Funcionais).
- **WCAG (Web Content Accessibility Guidelines):** Diretrizes de Acessibilidade para Conteúdo Web, mantidas pelo W3C (aplicado aos requisitos de interface e usabilidade).
- **LGPD (Lei Geral de Proteção de Dados Pessoais - Lei nº 13.709/2018):** Base legal para as decisões arquiteturais de privacidade, proteção de dados e revogação de acesso (Soft Delete).
- **Scrum / Kanban:** Metodologias Ágeis de desenvolvimento de software utilizadas para o planejamento e divisão de épicas.

2. Documentações de APIs Oficiais

- **Google Maps Platform:** Documentação técnica da *Google Places API* e *Maps JavaScript API* (para geolocalização, rotas e Place_ID).
- **Meta for Developers:** Documentação da *Instagram Graph API* (para autenticação, permissões e extração de mídia do Instagram Professional).
- **TikTok for Developers:** Documentação oficial para integração e login social.
- **Google Business Profile API:** Documentação para a validação de propriedade de estabelecimentos comerciais.

3. Serviços em Nuvem e Inteligência Artificial

- **AWS (Amazon Web Services):** Manuais do AWS S3 (Object Storage para imagens) e AWS *Rekognition* (Visão computacional para moderação de conteúdo).
- **Google Cloud Platform (GCP):** Documentação técnica da *Cloud Vision API*.

4. Arquitetura, Protocolos e Bancos de Dados

- **OAuth 2.0:** Protocolo padrão da indústria para autorização segura, utilizado nas integrações de login social.
- **Manifest V3:** Arquitetura oficial e documentação do Google Chrome para o desenvolvimento da extensão de navegador.
- **PostgreSQL & PostGIS:** Documentação oficial do banco de dados relacional e de sua extensão espacial (essencial para cálculos de raio de distância e coordenadas nativas).
- **REST (Representational State Transfer):** Padrão de arquitetura de software utilizado para modelar as requisições e respostas do Back-end.

 Aviso de Licença

Este projeto é de autoria própria. O uso, reprodução ou inspiração para projetos derivados deve obrigatoriamente citar o autor original. A cópia não autorizada do código ou da documentação sem os devidos créditos é considerada plágio conforme a legislação vigente.