



# Crie usando o documento de um pedido de construção de produto

Segue um PRD estruturado com base no documento anexado do desafio, cobrindo problema, escopo funcional, critérios de sucesso e considerações de entrega para a API de cadastro de agências e consulta de distâncias em Spring Boot.<sup>[1]</sup>

## Problem Statement

É necessário disponibilizar uma API REST em Spring Boot para cadastrar agências com coordenadas em um plano 2D e retornar as distâncias entre uma posição informada pelo usuário e as agências cadastradas.<sup>[1]</sup>

O sistema deve permitir o cadastro de agências via POST e calcular, ordenar e retornar as distâncias via GET em formato JSON, priorizando organização do código, boas práticas e aderência ao desafio.<sup>[1]</sup>

## Goals and Objectives

Entregar uma API REST funcional com endpoints de cadastro de agências e consulta de distâncias, com respostas ordenadas da menor para a maior distância.<sup>[1]</sup>

Garantir código legível, limpo e bem organizado, adotando boas práticas de programação, testes unitários e foco em prazo, com performance como adicional valorizado.<sup>[1]</sup>

## User Personas

Usuário final que informa uma posição  $X, Y$  e precisa identificar rapidamente as agências mais próximas para tomada de decisão geográfica simples.<sup>[1]</sup>

Avaliador técnico que valida legibilidade, boas práticas, cobertura de testes, aderência ao escopo e eventuais inovações além do mínimo do desafio.<sup>[1]</sup>

## Use Cases

Cadastrar uma agência informando coordenadas  $posX, posY$  para persistência em armazenamento em memória e disponibilidade em consultas posteriores.<sup>[1]</sup>

Consultar as distâncias entre uma posição  $posX, posY$  do usuário e todas as agências cadastradas, recebendo um JSON ordenado por proximidade.<sup>[1]</sup>

## Key Features

Endpoint POST `"/desafio/cadastrar"` para registrar uma agência com campos `posX` e `posY` recebidos em JSON, persistindo em armazenamento em memória.<sup>[1]</sup>

Endpoint GET `"/desafio/distancia"` recebendo `posX` e `posY` como parâmetros, retornando JSON com distâncias calculadas e ordenadas da menor para a maior.<sup>[1]</sup>

Retorno de exemplo com chaves de agências e valores de distância decimal, demonstrando ordenação e formatação esperada do payload.<sup>[1]</sup>

## Success Metrics

Aderência funcional: retorno ordenado corretamente, cálculo de distâncias para todas as agências cadastradas e conformidade com os formatos exemplificados.<sup>[1]</sup>

Qualidade de engenharia: legibilidade, organização, boas práticas e cobertura de testes unitários nas classes como critério explícito de avaliação.<sup>[1]</sup>

Operacionais: cumprimento de prazo como critério de avaliação e, adicionalmente, desempenho adequado considerando volumetria de um banco de grande porte.<sup>[1]</sup>

## Assumptions

Persistência em memória é aceitável e recomendada para facilitar a implementação, embora outras abordagens possam ser utilizadas conforme preferência.<sup>[1]</sup>

A especificação não define fórmula de distância nem esquema de identificação de agências no payload, apenas exemplifica chaves `"AGENCIA_n"` e distâncias decimais.<sup>[1]</sup>

## Timeline

Fase v1: implementar POST de cadastro em memória, GET de distâncias ordenadas e payloads conforme exemplos, garantindo legibilidade e boas práticas.<sup>[1]</sup>

Fase v2: ampliar testes unitários, explorar performance como adicional valorizado e considerar melhorias e inovações além do escopo mínimo.<sup>[1]</sup>

## Stakeholders

Time técnico responsável pela implementação da API em Spring Boot e pela qualidade do código entregue.<sup>[1]</sup>

Avaliadores do desafio Santander responsáveis por validar critérios funcionais, qualidade, prazo e diferenciais de performance e inovação.<sup>[1]</sup>

## Known Constraints or Dependencies

Uso de Spring Boot para construção da API REST e necessidade de um armazenamento em memória funcional para resgatar posições cadastradas.<sup>[1]</sup>

Cálculo e ordenação das distâncias devem ocorrer em cada consulta, com resposta em JSON e formatação compatível com os exemplos.<sup>[1]</sup>

## Open Questions

Qual fórmula exata de distância deve ser adotada para o plano  $X, Y$  e qual a precisão decimal esperada no retorno.<sup>[1]</sup>

Como deve ser definido o identificador das agências no JSON de resposta e se há requisitos adicionais de validação de entrada.<sup>[1]</sup>

## Risks

Risco de escolhas arbitrárias na fórmula de distância e identificação de agências impactarem a avaliação de aderência ao desafio, caso destoem da expectativa implícita dos exemplos.<sup>[1]</sup>

Risco de degradação de desempenho ao crescer o volume de agências se a solução em memória não considerar eficiência no cálculo e ordenação, embora performance seja um adicional.<sup>[1]</sup>

✱

1. Desafio-Back-N8-2-1-\_251004\_064537.pdf