



## **DESCRIÇÃO HACKATON – SOFTWARE ARCHITECTURE**

**Luiz Zenha, Erick Muller, José Rubens**

**Versão 1**

## Descrição do Hackaton

A Hackaton Company SA, uma empresa de grande porte com mais de 100.000 colaboradores e que atende diversas áreas, inclusive tecnologia, está em um ponto de transição crucial. Atualmente, está utilizando um sistema de ponto eletrônico terceirizado, mas devido a questões de qualidade e custos mensais significativos por usuário, a diretoria decidiu por construir o próprio sistema de ponto eletrônico interno. E, para que atenda as diversas filiais e pontos de trabalho dos colaboradores, este novo sistema será baseado em nuvem.

A decisão de migrar para um sistema interno e em nuvem é motivada por várias razões. A primeira é relacionada à qualidade do serviço fornecido pelo sistema terceirizado, que tem sido insatisfatória, com problemas frequentes de desempenho e disponibilidade. Somando-se a isso, os custos mensais por usuário estão se tornando proibitivos, especialmente considerando a quantidade de colaboradores. E outro ponto importante é a possibilidade de controlar a evolução do software, implementando as melhorias necessárias sempre que desejado.

Ao construir o próprio sistema em nuvem, espera-se não apenas economizar custos no longo prazo, mas também melhorar a qualidade e a confiabilidade do registro de ponto para os colaboradores. A nuvem oferece benefícios significativos em termos de escalabilidade, disponibilidade e segurança, tornando-a a escolha ideal para o novo sistema.

Este projeto envolverá a criação de um sistema de ponto eletrônico robusto, escalável e seguro que possa suportar a carga de nossa grande força de trabalho. O novo sistema irá implementar as funcionalidades do sistema atual, e terá uma série de recursos adaptados à realidade da nossa empresa, incluindo a autenticação de usuário, registro de ponto, visualização e edição de registros, geração de relatórios, notificações e integração com outros sistemas.

A transição para a nuvem é um passo importante para o início de grandes modernizações dentro da organização.

### Requisitos funcionais MVP

1. **Autenticação de Usuário:** O sistema deve permitir que os usuários se autenticuem usando um nome de usuário ou matrícula e senha.
2. **Registro de Ponto:** O sistema deve permitir que os usuários registrem o horário de entrada, intervalos e saída do trabalho. Isso deve incluir a data e a hora exatas do registro. O usuário apenas registra o evento, e o sistema obtém o horário do momento do registro.
3. **Visualização de Registros:** O sistema deve permitir que os usuários visualizem seus registros de ponto. Isso deve incluir todos os detalhes, como data, hora de entrada, intervalos e saída, e total de horas trabalhadas no dia.
4. **Relatórios:** O sistema deve ser capaz de gerar o espelho de ponto mensal com base nos registros de ponto do mês fechado (anterior) e

enviar esse relatório por e-mail ao solicitante. (Listagem das datas, batimentos de ponto e total de horas trabalhadas)

5. **Segurança:** O sistema deve garantir que os dados dos usuários sejam armazenados de forma segura e que a privacidade seja mantida.

6. **Disponibilidade:** O sistema deve estar disponível 24/7 para permitir que os usuários registrem seu ponto a qualquer momento, e o tempo de resposta dos serviços de marcação de ponto deve ser de até 5 segundos.

1. Lembrando que para os requisitos acima, somente serão necessários o desenvolvimento das APIs que realizarão a autenticação, o registro de ponto e a geração dos relatórios. Não será necessário o desenvolvimento de nenhuma interface.

## Requisitos funcionais versão evolutiva (fase 2)

1. **Edição de Registros:** Em caso de erros, o sistema deve permitir que os usuários editem seus registros de ponto. No entanto, essa funcionalidade deve ser restrita e possivelmente necessitar de aprovação de um supervisor.

2. **Notificações:** O sistema deve ser capaz de enviar notificações para lembrar os usuários de registrar seu ponto.

3. **Administração:** Deve haver uma interface de administração para gerenciar usuários, aprovar edições de registros e visualizar relatórios.

4. **Relatórios:** O sistema deve ser capaz de gerar os relatórios com visão administrativa, para uso dos gestores

## Requisitos não funcionais

1. **Desempenho:** O sistema deve ser capaz de suportar um grande número de usuários simultâneos sem degradação do desempenho. Deve ser capaz de processar rapidamente as solicitações de registro de ponto de todos os 100.000 colaboradores. Horários de pico 9h, 12h as 14h, 18h chegando a picos de quase 1.000 TPS

2. **Escalabilidade:** O sistema deve ser escalável para acomodar o crescimento da empresa. Deve ser capaz de lidar com um aumento no número de usuários sem a necessidade de alterações significativas na arquitetura do sistema, e com nenhuma ação manual.

3. **Disponibilidade:** O sistema deve ter alta disponibilidade, garantindo que esteja operacional 24/7 com um nível de desempenho adequado. Deve haver redundância suficiente para garantir que o sistema continue funcionando mesmo em caso de falha de um ou mais componentes.

4. **Segurança:** O sistema deve seguir as melhores práticas de segurança para proteger os dados dos usuários. Isso inclui criptografia de dados, autenticação segura e proteção contra-ataques comuns.

5. **Integridade dos Dados:** O sistema deve garantir a integridade dos dados, garantindo que os registros de ponto não sejam perdidos, corrompidos ou alterados indevidamente.
6. **Manutenibilidade:** O sistema deve ser fácil de manter, com código bem documentado e seguindo as melhores práticas de engenharia de software.
7. **Resiliência:** O sistema deve ser capaz de se recuperar rapidamente de falhas, minimizando o tempo de inatividade.
8. **Conformidade:** O sistema deve estar em conformidade com todas as leis e regulamentações relevantes relacionadas à privacidade e proteção de dados.

## Entregáveis mínimos

Como entregáveis o grupo deverá gravar um vídeo explicando a solução desenvolvida com as seguintes etapas:

- **Desenho da solução MVP** que atenda aos requisitos funcionais, e justificando escolhas técnicas, abordando também os requisitos não funcionais solicitados.
- **Desenho da solução evolutiva (fase 2)**, esse novo desenho deve ser a visão completa do sistema, com a evolução para atender aos novos requisitos funcionais, mantendo-se aderente aos requisitos não funcionais.
- **Demonstração da infra na cloud** com as peças construídas para atender ao MVP, e com o funcionamento da aplicação usando esta infra, para demonstrar a implementação da solução e o uso da solução (por exemplo, mas não se restringindo a ele: efetuar uma chamada de API e mostrar o log dessa chamada dentro do componente que atende a esta chamada).
- **Demonstração da esteira de CI/CD** para deploy da aplicação (ou das aplicações) criadas para atender os requisitos funcionais do MVP.
- **Demonstração do MVP**, aplicação executando em cloud mostrando todas as funcionalidades de acordo com os requisitos solicitados para o MVP.

A solução evolutiva (fase 2) não deverá ser construídas apenas esperamos o desenho de solução para analisar a solução propostas.

### Atenção:

Itens adicionais aprendidos ao longo do curso serão o diferencial para a classificação e pontuação dos grupos, por exemplo:

- Aplicar os conhecimentos e melhores práticas de qualidade e arquitetura de software na construção da solução

- Aplicar os conhecimentos e melhores práticas de qualidade de solução de infraestrutura.
- Demonstrar e aplicar os conhecimentos de desenvolvimento seguro
- Documentações abrangentes de todos os componentes e níveis da solução, e adequação/utilidade da documentação aos públicos que a usa.
- Automatização de todo o processo de gerência e alteração de Infraestrutura
- Justificativa de abordagem de micros-serviços ou monolito, e outras documentações que mostrem como as decisões de arquitetura de software ou de solução foram tomadas.