Prontos para o Desafio Final?

# Contexto

Chegou a hora de aplicar todos os conhecimentos adquiridos até aqui. Está pronto para o desafio?

Acabou de surgir uma nova pousada no mercado, e adivinha só… Você e sua equipe, por terem se destacado no treinamento da Gama, foram convidados para criar o sistema que vai gerenciar os quartos, clientes, e reservas. Deu um friozinho na barriga só de pensar?! Não se preocupe! Nosso analista de negócio já se reuniu com o cliente e deixou tudo mastigadinho pra vocês.

A dona da pousada quer um sistema em que ela consiga cadastrar os seus quartos e seus clientes, também é importante que consigamos deletar os usuários e quartos cadastrados, vai que alguma coisa dê errado e o cadastro precise ser refeito. Até aqui moleza, né?!

Depois de cadastrar tudo chegou a hora de colocar esse sistema no ar e começar a controlar as reservas. Lembre-se que só deve ser possível realizar a reserva se tanto o cliente quanto o quarto estiverem cadastrados no banco de dados. Além disso, não se pode reservar um quarto que já foi reservado por outro cliente, por isso muito cuidado! Não queremos confusão na recepção, não é mesmo?!

As pessoas não vão ficar nos quartos para sempre, esses quartos devem ser reservados e liberados eventualmente. Por isso, é importante que vocês desenvolvam alguma forma de realizar o check-in e o check-out do quarto. Nem preciso falar que só pode ser feito check-in de quarto reservado, e check-out depois do check-in, certo?!

Para facilitar a operação o sistema deve gerar alguns relatórios. É importante listarmos os quartos que estão vazios, assim saberemos quais podem ser reservados. Também temos que saber quais quartos não podem ser reservados, para quando o sistema devolver erro de quarto ocupado, então coloquem uma função para listar esses também. Pode ser uma listagem só com o status de cada quarto ou uma listagem para cada, vocês quem decidem!

Ah!!! Não podemos nos esquecer do menu. Para facilitar a vida da nossa cliente criem um menu bem bonito e intuitivo, não queremos que ela vá para a concorrência.

Muita coisa? Então vamos resumir:

* Primeiro: o quarto e o cliente já tem que estar previamente cadastrados
* Segundo: deve haver uma opção para listar todos os quartos vazios e todos os quartos ocupados.
* Terceiro: o cliente escolhe um quarto vazio e o sistema realiza a reserva.
* Quarto: quando o cliente chega na recepção é feito o check-in, indicando que o cliente está no quarto.
* Quinto: ao terminar a estadia é realizado o check-out e o quarto deve ser liberado para reserva novamente.
* Sexto: Não temos um sexto, mas sintam-se livres para adicionar funcionalidades que considerem um diferencial. Se der tempo, claro! Afinal, temos apenas uma semana para fazer a entrega.

Se organizar direitinho dá pra fazer. Cinco pontos em cinco dias é tempo de sobra. Principalmente com as super dicas que vamos deixar abaixo:

# Dicas

* Planejem todas as tarefas em uma planilha, estipulem prazos, atribuam responsáveis, criem um cronograma, etc. Depois utilizem essa planilha para cadastrar essas atividade em uma ferramenta de gestão de tarefas, como Trello ou Notion.
* Criem um grupo no slack e realizem dailys assíncronas, ou seja, todo dia escrevam uma mensagem dizendo o que fizeram no dia anterior, o que pretendem fazer no dia atual, e se estão com algum problema ou precisando de ajuda.
* Analisem os pontos fortes e fracos de cada integrante e atribuam tarefas baseados em afinidade e habilidade de cada integrante em cada conteúdo.
* Pararelizem as tarefas, enquanto um cria o board, outro já pode começar o desenvolvimento do código, e um terceiro pode iniciar a criação de um banco de dados, outro pode fazer o repositório no github, etc.
* Lembrem-se, “feito é melhor que perfeito” foquem em terminar as tarefas primeiro e depois se preocupem com melhorias. Um software básico funcionando e bem documentado é melhor do que um super criativo que não funciona.
* Não é necessário fazer um readme muito grande ou com muitas imagens, a ideia é ser simples porém conciso e explicativo. O ideal é que a leitura seja rápida e informativa, sem ser extensa.
* Organize os arquivos do projeto em pastas.

# Entregas Mínimas

1. Metodologias Ágeis:
   1. Board (Kanban) ou planilha criada.
   2. Tarefas organizadas, responsáveis e prazos definidos.
2. Git:
   1. Pelo menos 2 commits por aluno.
   2. Código entregue no Github.
3. Banco de Dados:
   1. MER do banco de dados como imagem no github.
   2. Scripts SQL de criação dos objetos (DDL) do banco de dados (database, schema, tabelas, etc.).
4. Python
   1. Utilização de estrutura de dados (lista, tuplas, dicionários, etc.).
   2. Pelo menos 1 bibliotecas.
   3. Pelo menos 4 funções.
   4. Código documentado.

# Critérios de Avaliação

O projeto foi dividido em tópicos, representados pelas letras: a, b, c, e d. E em assuntos, representados pelos números: 1, 2, 3 e 4. Cada tópico equivale a 1 ponto, totalizando 10 pontos no projeto.

No assunto “Metodologias Ágeis” avaliaremos a organização, trabalho em equipe, gestão de tempo, e divisão de tarefas. Em “Git” serão avaliadas as capacidades de utilização da ferramenta Git e Github para trabalho colaborativo e assíncrono.

Já em “Banco de Dados”, espera-se que os alunos consigam persistir as informações do projeto de maneira organizada, além de trabalhar o planejamento ao utilizar alguma ferramenta para criação do MER, e habilidade com SQL na criação, manipulação e consulta de objetos de banco de dados, como bases, esquemas e tabelas.

Para “Python”, além de um programa funcional e que atenda aos requisitos solicitados, é importante que o código seja documentado e otimizado com a utilização de bibliotecas, funções, e estrutura de dados. Esses detalhes ajudam os alunos a se preocuparem não só com o desenvolvimento do código mas também com a qualidade do mesmo.

Por fim, espera-se que os alunos consigam descrever (de forma resumida) qual era o desafio proposto e o que eles fizeram para solucioná-lo. É muito comum a utilização de “readmes” em repositórios Github, tal documento é o primeiro ponto que um visitante se depara ao entrar em um repositório.

Recomenda-se a utilização de imagens explicativas, no intuito de resumir o trabalho realizado no banco de dados, no código, ou que mostrem quais foram as ferramentas utilizadas no projeto, como o projeto foi dividido entre os integrantes, etc.