CareMI-API

* a) (até 5 Pontos) Apresentar o cronograma de desenvolvimento e respeitar os prazos. Crie um o documento dizendo quem fará o que e quando a atividade deverá ser ou foi realizada; - concluído
* b) (até 10 Pontos) Imagens explicativas da arquitetura, definição das classes de domínio da aplicação e o respectivo Diagrama de Classes de Entidade. O Diagrama de Entidade e Relacionamento (DER) e o Diagrama de Classe das Entidades deverão ser coerentes. Se necessário faça uma breve explicação sobre os relacionamentos e as constraints envolvidas; - concluído
* c) (até 40 Pontos) Implementação das classes de Entidade necessárias para solução do problema que se propõe resolver. Atenção ao correto encapsulamento, à tipagem dos atributos e ao Mapeamento Objeto Relacional com JPA e Hibernate (A equipe deverá demonstrar a cada Sprint que houve evolução em relação à sprint anterior);- concluído
* d) (até 15 Pontos) A aplicação respeita os conceitos fundamentais do REST criados por Roy Fielding (RESTful) e API está de acordo com o modelo de maturidade nível 2 proposto por Leonard Richardson. - concluído
* e) (até 10 Pontos) A Gestão de Configuração dos Artefatos de Software foram demonstrados pela equipe. Ou seja, todos os artefatos produzidos estão no Github e os professores têm acesso ao repositório. - concluído
* f) (até 10 pontos) Envio do link para o projeto público compartilhado no github contendo entre outros: - concluído
  + 1) Toda documentação; - concluído
  + 2) Código-fonte; - concluído
  + 3) Arquivo README.md com texto contendo: - concluído
  + 3.1) Nome da aplicação; - concluído
  + 3.2) Nome completo e breve apresentação dos integrantes do Grupo (Atividade da qual ficou responsável no projeto); - concluído
  + 3.3) instruções de como rodar a aplicação; - concluído
  + 3.4) imagem dos diagramas; - concluído
  + 3.5) link para vídeo apresentando a Proposta Tecnológica, o público-alvo da aplicação e os problemas que a aplicação se propõe a solucionar; - concluído
  + 3.6) Listagem de todos os endpoints (Documentação da API); - concluído
* g) (até 10 pontos) Se a equipe demonstrou preocupação em testar a aplicação e provou com documentos. Disponibilizou arquivos para que o professor seja capaz de realizar teste dos endpoints e houve, após os testes, a perfeita persistência e recuperação dos dados. (exporte do Postman ou do Insomnia as requisições aos endpoints). Inclua na pasta documentos do projeto. - concluído

Cronograma de desenvolvimento

* Iniciar projeto e criar repositório - 08/04/24 -(Miguel Fernandes).- concluído
* Criar Diagrama de entidade e relacionamento (DER) - (Matheus Rodrigues) - concluído
* Criação do script do vídeo com a proposta tecnologica, o público-alvo da aplicação e os problemas que a aplicação se propõe a solucionar - 12/04/24 - (Alberto Seiji) - concluído
* Produção e publicação do vídeo com a proposta tecnologica, o público-alvo da aplicação e os problemas que a aplicação se propõe a solucionar - 15/04/24 - (Patrick Jaguski) - concluído
* Testar a aplicação e documentar - 20/05/24 - (Nicolly de Almeida). - concluído
* Documentar cronograma de desenvolvimento - 20/05/24 - (Miguel Fernandes). - concluído
* Mudar Diagrama de classes e Modelagem de dados - 20/05/2024 - (Nicolly de Almeida). - concluído
* Mapeamento dos relacionamentos e criação dos endpoints- 06/05/24 a 20/05/24 - (Nicolly de Almeida). - concluído
* Criação das queries - 20/05/2024 - (Nicolly de Almeida) - concluído
* Envio do link para o projeto público compartilhado no github - 20/05/24 - (Matheus Rodrigues). - concluído
* Mudanças na implementação da aplicação- 06/05/25 a 20/05/24 - (Nicolly de Almeida). - concluído
* Gestão de Configuração dos Artefatos de Software – 06/05/24 a 20/05/24 - (Nicolly de Almeida). - concluído