**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**

**INSTITUTO METRÓPOLE DIGITAL**

**IMD0040 - Linguagem de Programação II - 2019.2**

**Especificação do Trabalho Final**

PROJETO 3ª UNIDADE

BOT PATRIMÔNIO TELEGRAM



**Objetivo**

Você deverá implementar um sistema de patrimônio que tem a função de cadastrar os bens de uma determinada empresa e mantê-los atualizados quanto a sua descrição, localização e número de registro (tombamento).

**Introdução**

Diferentes sistemas web, desktop e mobile existem no mercado com soluções patrimoniais, onde as empresas utilizam seus serviços para manter em ordem os seus bens. Diante disso um novo sistema patrimonial simples e prático que funcione através de um bot do Telegram pode ser o ideal para a utilização em diversos meios acadêmicos e profissionais. O usuário, através de um Bot para o Telegram, poderá fazer fazer o cadastro, consulta e atualização de bens da empresa.

**Bot**

Bot, diminutivo de robot, também conhecido como Internet bot ou web robot, é uma aplicação de software concebido para simular ações humanas repetidas vezes de maneira padrão, da mesma forma como faria um robô. No contexto dos programas de computador, pode ser um utilitário que desempenha tarefas rotineiras ou, num jogo de computador, um adversário com recurso a inteligência artificial.  
fonte: <https://pt.wikipedia.org/wiki/Bot>

**Telegram Bot**

Bots are simply Telegram accounts operated by software – not people – and they'll often have AI features. They can do anything – teach, play, search, broadcast, remind, connect, integrate with other services, or even pass commands to the Internet of Things.  
fonte: <https://telegram.org/blog/bot-revolution>

Para implementar o bot em Java pode ser seguido tutorial disponível em <https://medium.com/@michelpf/criando-bots-do-telegram-com-api-java-64e921762b5f>. Essa é apenas uma sugestão para implementar o bot em telegram com Java. Mas não é **obrigatório seguir essa documentação.**

**Requisitos funcionais**

O sistema de patrimônio desenvolvido deverá implementar as operações abaixo relacioandas:

1. cadastrar localização (ex.: sala, laboratório, auditório, etc.)
   1. Realiza o cadastro de uma localização. Os dados deverão ser salvos em alguma memória não volátil, por exemplo, um arquivo. Uma localização deverá conter no mínimo os atributos nome e descrição.
2. cadastrar categoria de bem (ex.: móvel, eletrônico, material de limpeza, etc.)
   1. Realiza o cadastro da categoria de um bem. Os dados deverão ser salvos em alguma memória não volátil, por exemplo, um arquivo. Uma categoria deverá conter no mínimo os atributos codigo, nome e descricao
3. cadastrar bem (cadeira, mesa, computador, sabão em pó, etc.)
   1. Realiza o cadastro de um bem. Os dados deverão ser salvos em alguma memória não volátil, por exemplo, um arquivo. Um bem deverá conter no mínimo os atributos codig, nome, descricao, localização e categoria
4. listar localizações
   1. Deverá exibir na tela todas as localizações cadastradas no sistema
5. listar categorias
   1. Deverá exibir na tela todas as categorias cadastradas no sistemas
6. listar bens de uma localização
   1. Deverá exibir na tela todos os bens de uma determinada localização informada como parâmetro
7. buscar bem por código
   1. Deverá exibir na tela todas as informações de um bem que possua o código informado como parâmetro
8. buscar bem por nome
   1. Deverá exibir na tela todas as informações de um bem que possua o nome informado como parâmetro
9. buscar bem por descricao
   1. Deverá exibir na tela todas as informações de um bem que possua a descrição informada como parâmetro
10. movimentar bem
    1. Deverá realizar a movimentação de um bem de uma localização para outra localização informada como parâmetro
11. gerar relatório
    1. Gera um relatório listando todos os bens agrupados por localização e depois por categoria e depois por nome (agrupa primeiro por localização, depois por categoria e depois por nome) ordenados alfabeticamente
12. gerar relatório
    1. Gera um relatório em arquivo listando todos os bens agrupados por localização e depois por categoria e depois por nome (agrupa primeiro por localização, depois por categoria e depois por nome) ordenados alfabeticamente. O arquivo poderá ser em formato texto simples ou csv;
13. Carregar dados
    1. Deverá ser informado um local no computador ( diretório) onde o sistema deverá ler todas as informações armazenadas em execuções anteriores
14. Apagar localização
    1. Deverá apagar o registro de localização do sistema. Só deverá ser possível apagar uma localização se não houver nenhum bem com essa localização
15. Apagar categoria
    1. Deverá apagar o registro de categoria do bem do sistema. Só deverá ser possível apagar uma categoria do bem se não houver nenhum bem com essa categoria
16. Apagar bem
    1. Deverá apagar o registro de bem do sistema.

**Implementação**

Seu sistema deverá ser modelado de acordo com as melhores práticas de orientação a objetos, e ser implementado com a linguagem de programação Java. Lembre-se de agrupar classes com funcionalidades semelhantes em pacotes. Além disso, é esperado que o aluno utilize os seguintes conceitos:

1. Modularização e padronização
   1. Nomeação de classes e métodos segundo padrões;
   2. Separação das classes em pacotes
   3. Design de Classes (responsabilidade e coesão)
2. Encapsulamento
   1. Utilização de padronização de acesso e modificação com gets e sets
3. Utilização de bibliotecas externas;
4. Documentação JavaDoc;
5. Herança;
6. Polimorfismo;
7. Classe Abstrata;
8. Interfaces;
9. Tratamento de Exceções
   1. Utilizando a definição de novas classes de Exceção
10. Interface gráfica
    1. Utilização de JavaFx para interface Gráfica

**Entrega**

Seu grupo deverá submeter, via google classroom, um arquivo compactado contendo:

* Código fonte do sistema desenvolvido, incluindo um documento README.TXT contendo instruções de como se pode compilar o código fonte.
* Relatório Técnico, contendo pelo menos as seguintes seções:
  + Introdução: contendo uma breve descrição do problema abordado e do que será apresentado em seu relato;
  + Descrição da abordagem de solução do problema: Descrevendo o projeto OO (diagramas de classes), e destacando as decisões de projetos tomada;
  + Descrição geral das estruturas de dados utilizadas;
  + Descrição de como foram implementados os requisitos funcionais;
  + Conclusão;
  + Referências;

O relatório deverá ser feito seguindo o template da Sociedade Brasileira de Computação (SBC) que pode ser encontrado no seguinte endereço:

<http://www.sbc.org.br/documentos-da-sbc/summary/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>

**Apresentação**

Cada grupo terá 20 minutos para fazer sua apresentação. Nesta apresentação, os grupos deverão demonstrar o funcionamento do programa desenvolvido. Além disso, deverão estar aptos a responder questões sobre o desenvolvimento do projeto.

**Importante**: não será dada uma única nota ao grupo. Cada componente do grupo receberá uma nota de acordo com seu desempenho durante a apresentação.

O programa será avaliado como um todo, ou seja, os requisitos não receberão pontuação individualmente. Dessa forma, a falta de um ou mais requisitos acarretará na perda de pontos, que poderá ser compensada (não totalmente, claro) através de outros componentes bem desenvolvidos. Componentes adicionais serão muito bem vistos, desde que implementados de maneira racional.

**Datas**

* 31/10/2019
  + Bot e comandos de cadastro em memória
* 07/11/2019
  + Bot e comandos de cadastro salvando em arquivo
* 14/11/2019
  + Bot e comandos de remoção
  + Bot e relatórios
* 19/11/2019, 26/11/2019 e 28/11/2019
  + Apresentação dos trabalhos