

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE – UFRN
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA - DIMAp
DIM0531 - TÓPICOS ESPECIAIS EM ENGENHARIA DE SOFTWARE I

DOCENTE:
THAIS VASCONCELOS BATISTA

DISCENTES:
RUBEM KALEBE SANTOS
STEFANO MOMO LOSS

ESTACIONAMENTO INTELIGENTE

NATAL/RN
Setembro de 2016

Proposta

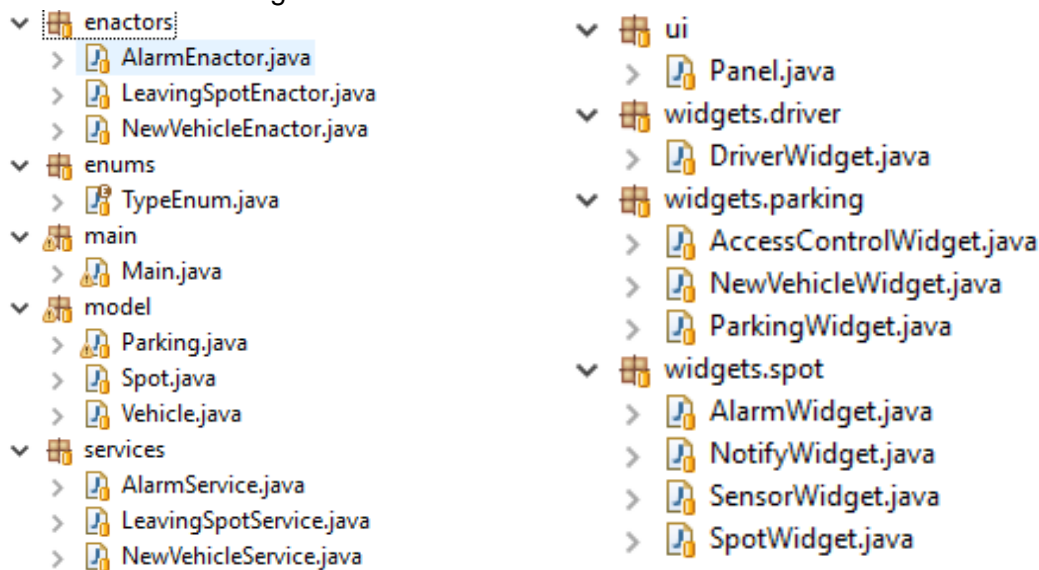
Propõe-se um sistema de estacionamento inteligente capaz de atender tanto a área privada quanto a pública. Além da automatização local o sistema deve permitir certa interação entre os usuários e o próprio estacionamento, através de um aplicativo para dispositivos móveis que exiba o estado das vagas em tempo real para os usuários.

Elementos monitorados

- **Presença de carros:** deve-se detectar se uma vaga está ocupada ou não, e retornar esse estado para os usuários do aplicativo, além de registrar as mudanças de estado em um banco de dados;
- **Motoristas em vagas reservadas:** caso um motorista estacione em uma vaga reservada, o sistema requisita a autenticação do motorista, constrangendo-o através de sinais sonoro e luminoso a fazer a autenticação;
- **Entrada de carros no estacionamento:** Verifica a disponibilidade de vagas e a permissão do veículo para entrar no estacionamento.
- **Saída dos veículos das vagas:** o sistema ajuda na segurança do veículo, notificando o dono quando o seu veículo sair da vaga.

Implementação

O sistema foi implementado totalmente em Java utilizando a biblioteca do CONTEXT TOOLKIT 2.1 estruturado da seguinte maneira:



Nos pacotes *widgets* estão que modelados os sensores dos elementos que são monitorados, já no pacote *enactors* estão as classes onde está a logica, que de acordo com

os valores dos sensores, executa determinados serviços, esses serviços estão descritos nas classes presentes no pacote *services*.

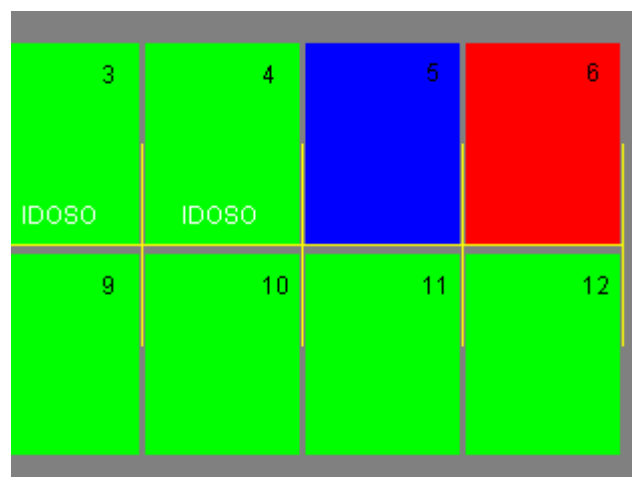
No pacote *model* temos as classes *Spot* e *Driver* que instancia todos os *widgets* relacionados com a vaga e motoristas respectivamente. A classe *Parking* instancia seus *widgets* relacionados e as vagas conforme descrito na seção da interface gráfica.

Interface Gráfica

A interface gráfica foi implementada utilizando JAVA SWING onde foi representado um estacionamento com doze vagas sendo as duas primeiras reservadas para pessoas com deficiência, e as duas vagas seguintes (terceira e quarta) reservadas para pessoas idosas.



As cores das vagas representam o seu status. A cor verde igual vaga livre; azul: vaga ocupada sem autenticação do motorista; vermelha: vaga com carro autenticado.



Simulação

A simulação dos sensores é dada a partir da interação com o terminal da aplicação através da escolha das opções do menu:

```
-----MENU-----  
1 - NOVO VEICULO  
2 - SOLICITAR ENTRADA NO ESTACIONAMENTO  
3 - ESTACIONAR VEICULO  
4 - REGISTRAR VAGA  
5 - SAIR DA VAGA  
0 - SAIR
```

Para executar a simulação é necessário criar novos veículos escolhendo o tipo de permissão (normal, deficiente, idoso) depois, solicitar entrada no estacionamento de cada veículo, estacionar o veículo em alguma vaga (isso altera o sensor de presença dessa vaga) para depois registrar o veículo na vaga. Caso algum veículo estacionar em lugar não permitido haverá uma notificação no terminal: ALARME!!! VAGA NN.