

Laboratório 2.4 – Relato de Modelagem Projeto Simples II

Objetivo

Relatar um processo de modelagem, relacionado as decisões com os padrões GRASP.

Relato

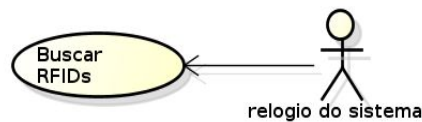
Ver “Laboratório 02.04 – Projeto Simples” para instruções e requisitos.

Passo ZERO, estreite o relacionamento com seu sistema, dê um nome a ele ou ela. Eu vou chamar o sistema de “**Saudações Ao Visitantes do Stand – SAVS**”.

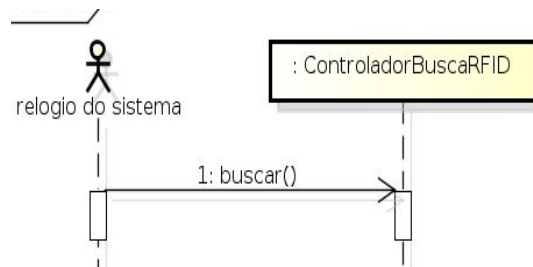
Passo 1 – Prepare um diagrama de sequência para ilustrar o caso de uso “Buscar RFIDs”

Ao modelar a realização de um caso de uso em DS, é necessário entender o escopo de comportamento do caso de uso e quais informações são usadas para que esse comportamento seja suportado.

Modelos de casos de uso possuem duas informações importantes: Atores e Objetivos (nomes dos casos de uso). Os atores interagem com o SAVS, e os objetivos são o que o SAVS realiza a partir da interação do ator. Temos então:



Através de uma interface de chamada de procedimentos, o relógio do **sistema operacional** vai disparar o caso de uso. Isso pode ser implementado de várias formas (threads, scheduled jobs, etc), mas para fins da nossa modelagem podemos apenas considerar que uma chamada será recebida em um componente que represente o limite entre a interface de entrada de dados e o núcleo do SAVS. Ou seja, podemos fazer uma direta analogia com a separação UI e Modelo de Domínio: quem fica no meio do caminho? O Controlador. Vejamos:

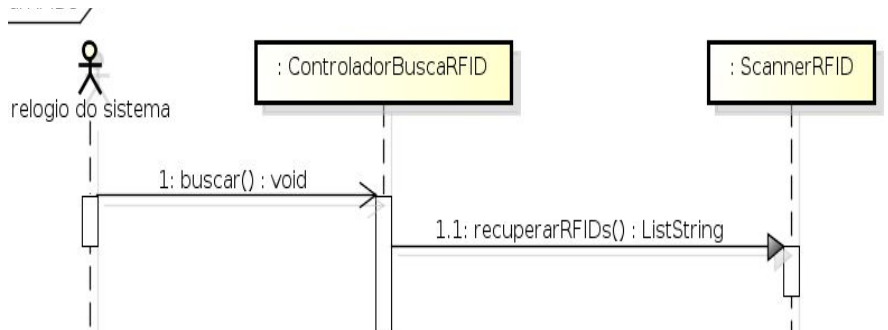


Referência: Padrão GRASP Controlador, página 318 (LARMAN), na discussão sobre a **Solução**.

Pronto, temos o início do caso de uso devidamente modelado. O que é realizado por esse caso de uso? O requisito nos diz que os RFIDs contêm o número de inscrição dos participantes, que podem ser checado no sistema do evento para recuperar detalhes. Então, o SAVS precisa ir no dispositivo Scanner ler os RFIDs, e depois consultar o sistema externo.

Após isso, precisa enviar as mensagens de saudações através das redes sociais.

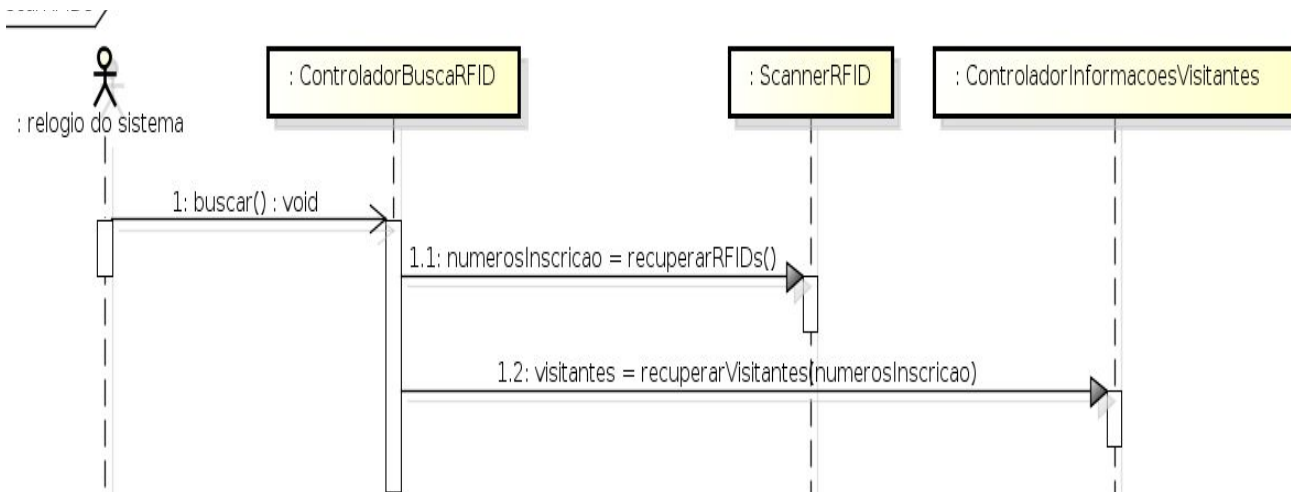
Vamos refletir primeiro sobre a interação com o Scanner. Não conheço esse dispositivo, mas como qualquer dispositivo esse possui uma forma de comunicação (serial, rede, wireless) e um protocolo de troca de mensagens. Não faz muito sentido os detalhes da comunicação com o Scanner estarem implementados diretamente no Controlador, isso quebraria a coesão do Controlador. Então, seria bom “inventar” um componente apenas para lidar com a comunicação com o Scanner:



Esse novo componente inventado tem o nome “ScannerRFID”, e vai fornecer um serviço simples: sempre que for consultado através do método “recuperarRFIDs()”, irá retornar uma lista dos últimos RFIDs detectados no stand do evento. Como o requisitos informar que esses rótulos RFIDs contém os números de inscrição, então nesse ponto do fluxo, os números de inscrição (em formato de String) já estão disponíveis no Controlador.

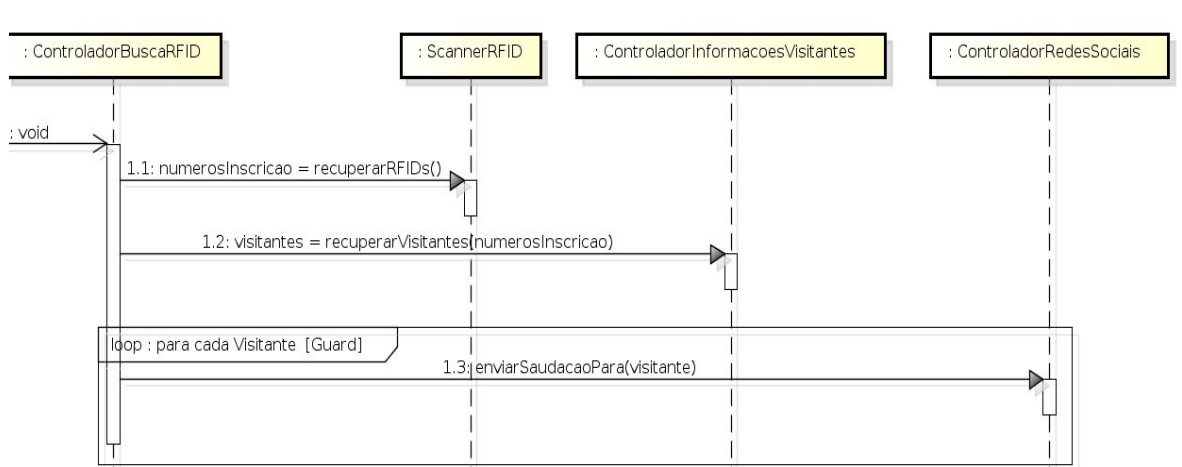
É necessário agora verificar o sistema do evento para recuperar mais informações dos visitantes, usando o número de inscrição como chave de consulta. Vários números podem ter sido retornados pelo Scanner, então vários serão verificados no sistema através do webservice disponibilizado.

No levantamento de requisitos, já foi idealizado um caso de uso separado – talvez pensando em reusar essa checagem das informações em alguma outra situação. Então, aqui aplicamos novamente o padrão Controlador, para realizar o caso de uso “Obter Informações Visitantes”.



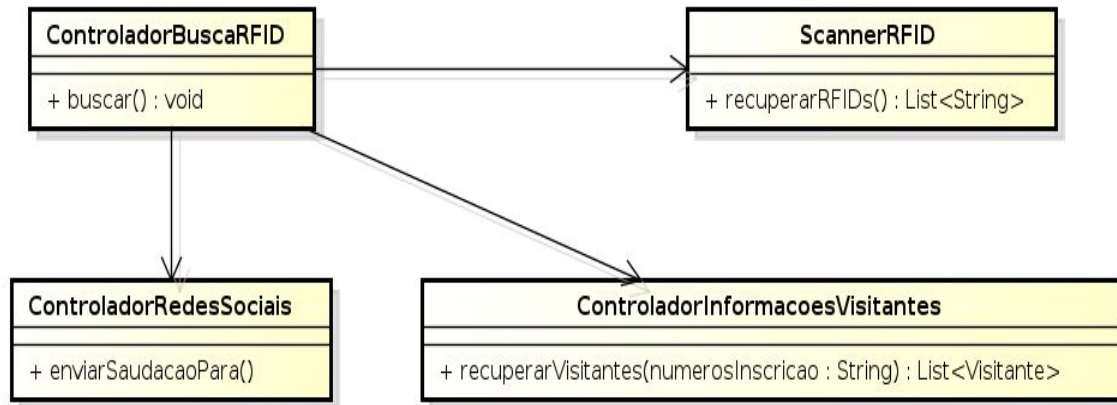
Referência: Padrão Controlador.

Por fim, o caso de uso “Buscar RFIDs” deve acionar outro caso de uso de extensão, a fim de que o SAVS realize a postagem da saudação nos perfis de redes sociais da empresa, citando/marcando os visitantes identificados. Para cada Visitante, acionar o caso de uso de extensão:



Referência: Padrão Controlador.

Para quem estiver curioso em saber como estaria a VCP nesse momento, uma visualização rápida:



Passo 2 – Prepare diagramas separados para os casos de uso “Realizar Postagem” e “Obter Informações Participantes”

Passo 3 – Prepare um diagrama de classes participantes

Passo 4 – Prepare um diagrama de sequência para ilustrar a inicialização do sistema