Projeto Inside Threat
Averiguação Comportamental de Atividades

Andrécio Costa Bezerra, Dante Alighieri Miranda dos Santos

Instituto Metrópole Digital – Universidade Federal do Rio Grande do Norte (UFRN)

andrecio@ufrn.edu.br, dante.alighierimds@gmail.com

Resumo. *Este artigo descreve o desenvolvimento de software para inspeção de comportamento de atividades dos usuários numa determinada rede, através da análise dos dados destas atividades, previamente levantados e salvos em arquivos do tipo CSV.*

Abstract. *This paper describes the development of software to inspect the behavior users' activities in a given network, by analyzing the data of these activies, previously raised and salved in CSV files.*

1. Abordagem

O programa foi desenvolvido a partir do paradigma de orientação à objetos, usando uma abordagem MVC com expansão, conforme o diagrama de classes abaixo.

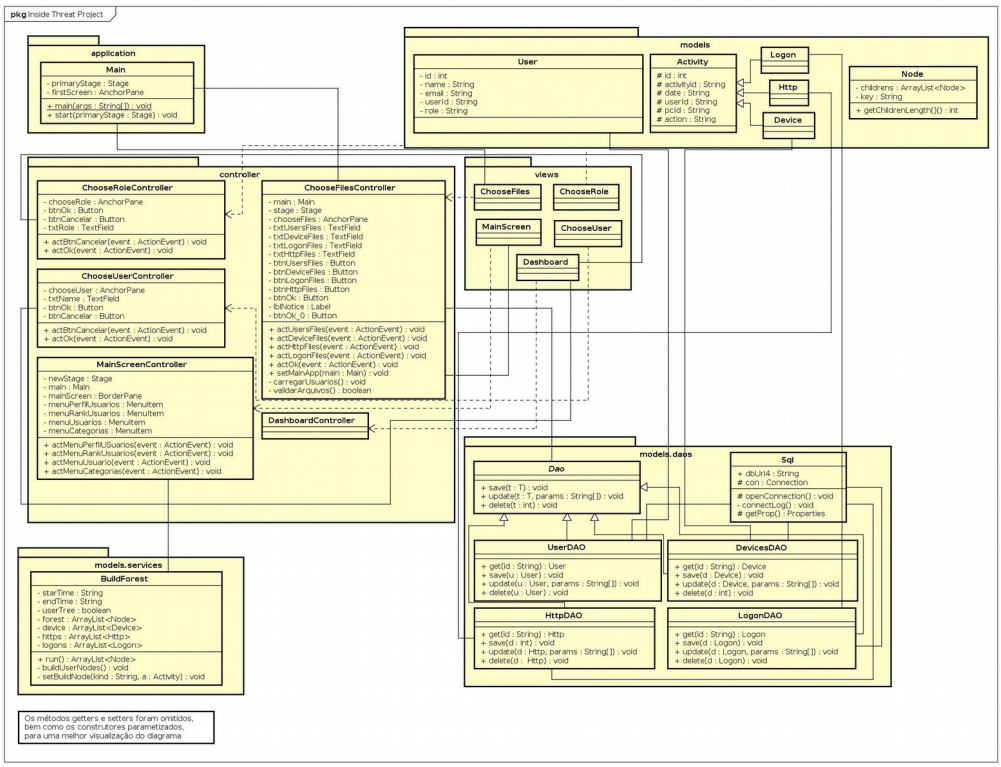

Figure 1. Diagrama de classes

Há o carregamento da informações contidas no arquivos CSV para o banco de dados Sql, de onde as informações serão carregadas na floresta de dados para a criação

histogramas. Após essa fase, o usuário pode decidir carregar o perfil de um usuário ou seu histograma, como também o histograma de uma categoria.

Os referidos dados serão apresentados em gráficos de linhas, apresentando sempre a média em caso, como os demais dados.

3. Estruturas utilizadas

Diversas estruturas foram criadas para execução deste projeto. Dentre as quais foi criada uma floresta de dados a partir do algorítimo abaixo :

```
public class BuildForest {
String startTime;
String endTime;
boolean userTree;
ArrayList<Node> forest;
ArrayList<Device> devices;
ArrayList<Http> https;
ArrayList<Logon> logons;

public BuildForest(String start_at, String end_at, boolean ut) {
        startTime = start_at;
        endTime = end_at;
        userTree = ut;
        forest = new ArrayList<Node>();
}

/**
 * Recupera os objetos de todas as atividades num periodo te tempo
 * @return forest
 */
public ArrayList<Node> run() {

        DeviceDAO dDAO = new DeviceDAO();
        HttpDAO hDAO = new HttpDAO();
        LogonDAO lDAO = new LogonDAO();

        try {
                devices = dDAO.betweenDates(startTime, endTime);
                https = hDAO.betweenDates(startTime, endTime);
                logons = lDAO.betweenDates(startTime, endTime);
        } catch (SQLException e) {
                System.out.println(e.getMessage());
        }

        buildUserNodes();

        return forest;
```

```
	}

	/**
	 * Percorrer cada objeto e procurar no banco o usuário
	 * Adicionar o usuário num nó
	 * No nó deste usuário adicionar a data
	 */
	private void buildUserNodes() {

		for (Device d : devices)
			setBuildNode("Device", d);
		for (Http h : https)
			setBuildNode("Http", h);
		for (Logon l : logons)
			setBuildNode("Logon", l);

	}

	private void setBuildNode(String kind, Activity a) {
		ArrayList<String> branch = new ArrayList<String>();
		branch.add(a.getDate());
		branch.add(a.getPcId());
		branch.add(kind);
		branch.add(a.getActivity());

		for (Node n : forest) {
			if (n.getKey() == a.getUserId()) {
				n.addNodeInChain(branch);

				return;
			}
		}

		Node node = new Node(a.getUserId());
		node.addNodeInChain(branch);
		forest.add(node);
	}
});
```

Outra estrutura utilizada foi a população da floresta, através do código abaixo:

```
	void setBuildForest(ActionEvent event) {
		BuildForest b = new BuildForest(startDate.getText(), endDate.getText(),
				false);
		ArrayList<Node> forest = b.run();
		try {

			FXMLLoader loader = new FXMLLoader();
```

```
loader.setLocation(MainScreenController.class.getResource("../views/ChooseUser.fxml
"));
                        AnchorPane chooseUserScreen = (AnchorPane) loader.load();
                        ChooseUserController controle = new ChooseUserController();
                        controle.carregarUsuarios(forest);

                        mainScreen.setCenter(chooseUserScreen);
                } catch (IOException e) {
                        e.printStackTrace();
                }
}
```

4. Conclusão

Através da população da floresta é feito os histogramas por usuário e por categoria(perfil médio), is quais infelizmente não puderam ser alcançados nessa etapa.

Referências

Gomes, Robson Fernandes (2012) “Utilizar “Property” em java para informação privada”, em: Devmedia em https://www.devmedia.com.br/utilizando-arquivos-de-propriedades-no-java/25546, dezembro.

Redko, Alla (2013) “File Choose”, em: Oracle em https://docs.oracle.com/javafx/2/ui_controls/file-chooser.htm, setembro.