Universidad Técnica Nacional

Desarrollo de Software (ISW)

Jean Carlo Vega Bejarano.

Kenet Acuña Lago.

Programming 2

Professor

Dennis Valverde Pacheco

Name of the project

Decision tree

2017Index

# Introduction:

In this project we are going to work with data structures and specially we are going to focus on double linked lists, circular list and decision trees with n quantity of nodes. The idea of this project is learning by doing and finally know the importance of data structures behind of software and the big projects.

The project is going to be create with Java programing language in NetBeans but with JavaFx tool, for this reason we are going to create the structure of project like MVC, models, controllers and views.

Each model is going to be save in the decision package, every controllers is going to be in the decision.controllers package and same way the views is going to have its package called like decision means.

# Problem Description:

The idea of this project consists in that the trees are structures that allow to obtain a response from different scenarios that can influence it, these structures provide a set of rules that are applied to new examples to decide which classification is most appropriate to their attributes.

This project will allow the creation and elimination of users who will build trees and / or execute them to obtain answers to their questions. For each user it is important to know the full name, ID, age, gender and password.

Each user can have a list of developed trees, each decision tree has a numerical identifier, the first node of the tree and a description of the topic that it intends to solve. The list of users will be a double linked and that of trees a double circular.

A user can create decision trees or delete them, for either of the two alternatives must be properly registered and with an active session.

Each of the questions can be inserted in the different levels of the tree, the application must be able to validate that each of the branches has a leaf node that concludes with one of the two final responses of the generated tree "Yes" or "No".

# Solution, the image represents the structure of project.

Imagen que contiene texto, mapa

Descripción generada con confianza muy alta

# User class:

Basically, in this part we can create the abstraction according to user information, for example we are going to save the attributes that has each user: full name, dni, age, sex and the password.

Also, the user can to has a list of his or her tree of decision, for this reason is other attribute that there is in the class, in this list the user can to add new trees, modify the existing trees and can to remove it.

The user is going to save in the double linked list, for this reason is going to has a previous user or a next user, this case is going to depend in the section that user is going to be, for example:

The first user can´t has a previous user for the reason that is going to be the first in the list, in this case this use is root in the list and then of root is going to display the other user,

If the user is in the end of the list only can to has a previous user but can´t has a next user, this is the opposite of root.

Then there are users that they are between of root and end of list, each of these users can has the both cases, previous user and next user, for the reason that they are in the middle of list.

# User Operations class:

In this section we are going to take the control of each operation that user can do while is inside of system.

For example trough of this way the user can to make login and sign up, because this class let us to review information of every user and the same way create of new users according of position they are in the list. Also in this class is the way that user can be registered on system, removed and they can close their sessions.

Basically, this section is a type of intermediary between user class and user controller that get the instruction by user in the view or UI, for this reason the user can does large quantity of things in the system.

# List of tree class:

List of tree class let us to take the control that main function that users can make with list of tree.

The first thing that they can make here is the creation of new trees, but for to impletion of the new trees we are going to work with circular list, that means that the first tree is going to linked with the last tree, and others tree nodes are going to be double linked, due to of this way we can show of search node by node.

Then in this section the users are able to remove some tree of their list and the system is going to able to remake of each connection of every tree node, also is very important to highlight that this class let us show the list of tree nodes from other user, like a catalog, but only the owner of tree can make modifications.

# Tree class:

This class is composed with a description, id and question, it´s the way that the user can save information of new tree, because this class is going to be the father of tree node.

When the user is going register a new tree on the system, the system is able to detect what is the description, what is id and what is the question that the user is writing in the UI and then the system can create the new tree.

Especially this class is going be pillar of the tree node class, because tree class inheritance his attribute.

# Tree Node:

In this section we can get the attribute of tree class, for this reason we are going to have description, id and question thanks of heritage in the tree node class, now we only need the type that is the new node in the tree that we are working.

Currently in the system exist tree types of nodes, we have: leaf, root and decision, that each one of these nodes are sons of tree node class, for this reason they are going to have the same attributes of tree.

# Root:

In this part, we are going to see the way we are using for to identify the main node in the tree, in summary words the root of tree. When the user created the tree, he or she must assign the first question, and this part the system can detect that this node is going to be the root.

# Leaf:

The leaf class consist in the answer according to the question of main node, for this reason this class is going to be the end node of each branch, and the unique in the list of nodes before and this because each decision node needs to has a some other decision nodes or a only leaf.

# Decision:

In the decision class we can said that is going to be intermediary, because this node is going to has, the answer of the node decision previous and the next question.

# Technology Stack:

In this project was used java language that is let us to develop the software and also was used the Git Hub repository for to save the code of application. These technologies focus in pillars of the project.

Then we used differents aspects that have each mentioned technology for example: in java we worked with OOP, Ui thanks by java swing, and in Git Hub with work with Git.

## What is Java?

“Java is a programming language and a computer platform commercialized for the first time in 1995 by Sun Microsystems. There are many applications and websites that will not work unless you have Java installed and more are created every day. Java is fast, secure and reliable. From laptops to data centers, from game consoles to super computers, from mobile phones to the Internet, Java is everywhere.”(Java).

## What is Git Hub?

“GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project. “(TCCruncOrg)

## What is OOP?

On the other hand is very important to say “OOP is a design philosophy. It stands for Object Oriented Programming. Object-Oriented Programming (OOP) uses a different set of programming languages than old procedural programming languages (C, Pascal, etc.). Everything in OOP is grouped as self sustainable "objects". Hence, you gain reusability by means of four main object-oriented programming concepts”(Code Project).

## What is JavaFx?

JavaFX is a library that let us have a good experience when we work with UI of our application and also is a good option because we can split the project in sections like MVC.

"The look and feel of JavaFX applications can be customized. Cascading Style Sheets (CSS) separate appearance and style from implementation so that developers can concentrate on coding. Graphic designers can easily customize the appearance and style of the application through the CSS. If you have a web design background, or if you would like to separate the user interface (UI) and the back-end logic, then you can develop the presentation aspects of the UI in the FXML scripting language and use Java code for the application logic. If you prefer to design UIs without writing code, then use JavaFX Scene Builder. As you design the UI, Scene Builder creates FXML markup that can be ported to an Integrated Development Environment (IDE) so that developers can add the business logic"(OracleJavaFx).

## What is MVC?

* "Model - The model represents data and the rules that govern access to and updates of this data. In enterprise software, a model often serves as a software approximation of a real-world process"(Java SE).
* "View - The view renders the contents of a model. It specifies exactly how the model data should be presented. If the model data changes, the view must update its presentation as needed. This can be achieved by using a push model, in which the view registers itself with the model for change notifications, or a pull model, in which the view is responsible for calling the model when it needs to retrieve the most current data."(JavaSE)
* "Controller - The controller translates the user's interactions with the view into actions that the model will perform. In a stand-alone GUI client, user interactions could be button clicks or menu selections, whereas in an enterprise web application, they appear as GET and POST HTTP requests. Depending on the context, a controller may also select a new view -- for example, a web page of results -- to present back to the user."(Java SE)

# Bibliography:

1. Docs.oracle.com. (2017). What Is JavaFX? | JavaFX 2 Tutorials and Documentation. [online] Available at: https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm [Accessed 6 Nov. 2017].
2. nield, d., Lawrence, C., thomas, o., lee, a., schorr, c. and Lawrence, C. (2017). GitHub For Beginners: Don't Get Scared, Get Started - ReadWrite. [online] ReadWrite. Available at: https://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/ [Accessed 21 Jul. 2017].
3. Java.com. (2017). What is Java and why do I need it?. [online] Available at: https://www.java.com/en/download/faq/whatis\_java.xml [Accessed 17 Jul. 2017].
4. Codeproject.com. (2017). Introduction to Object Oriented Programming Concepts (OOP) and More - CodeProject. [online] Available at: https://www.codeproject.com/Articles/22769/Introduction-to-Object-Oriented-Programming-Concep#OOP [Accessed 17 Jul. 2017].
5. Oracle.com. (2017). Java SE Application Design With MVC. [online] Available at: http://www.oracle.com/technetwork/articles/javase/index-142890.html [Accessed 6 Nov. 2017].