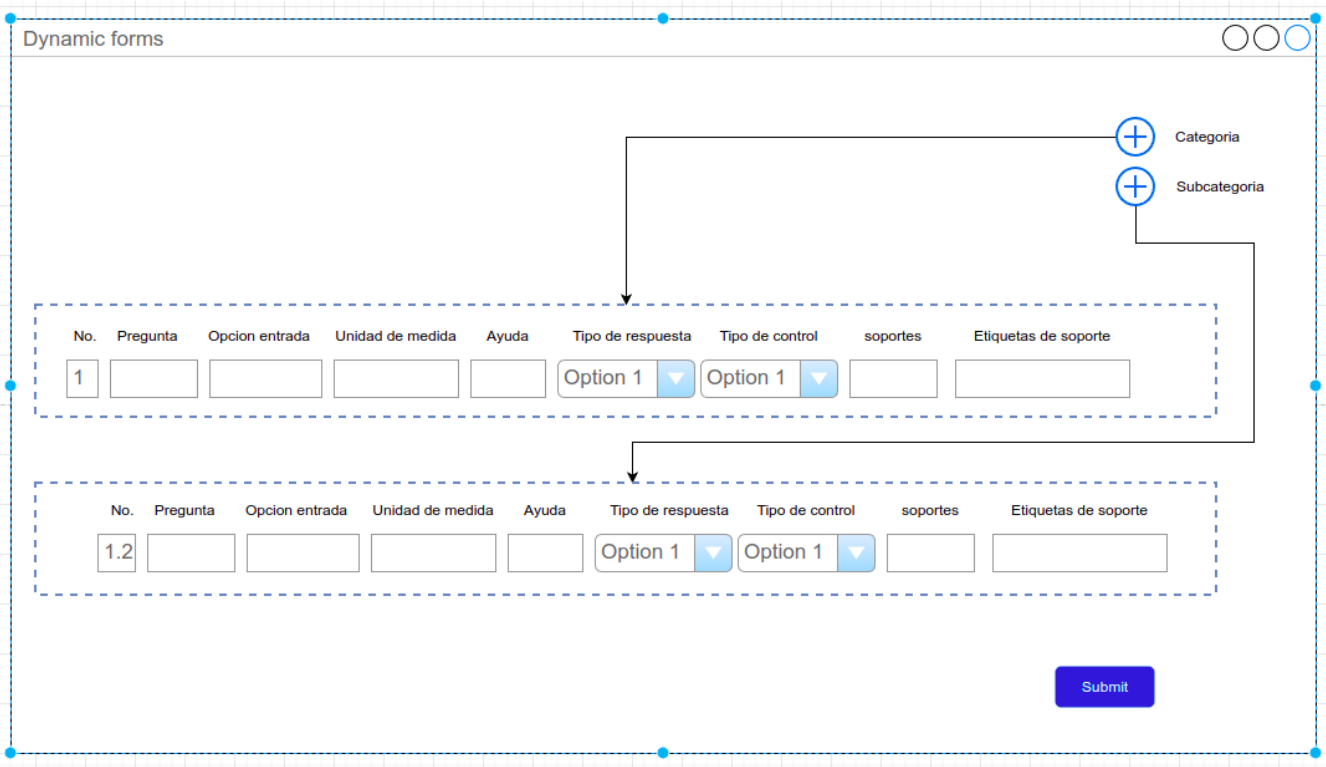
1.0 Objective

Gathering data is one the most difficult tasks in nowadays system. Although, graphical user interfaces have abandon the old rigid style, the browsers still display rigid data that does not yield a good final user experience. The presented assignment is aimed to provide insights about forms flexibilization.

2.0 Adding new entry lines to a form dynamically with java script or similar

2.1 First challenge



The first challagence consist in making a form that grows with the two bottom on top. When clicking the boton to add category, the form will add a literal integer like 1, 2, 3, 4 … and, when the subcat is press, the literal n.1, n.2, n.3 … n.n is added.

Naturally, the botton “add subcat” will be only active id a category exists. The resulting form must be usable, that means that controls like <input text name = entry#> should be diferential, meaning that # must be different in each control.

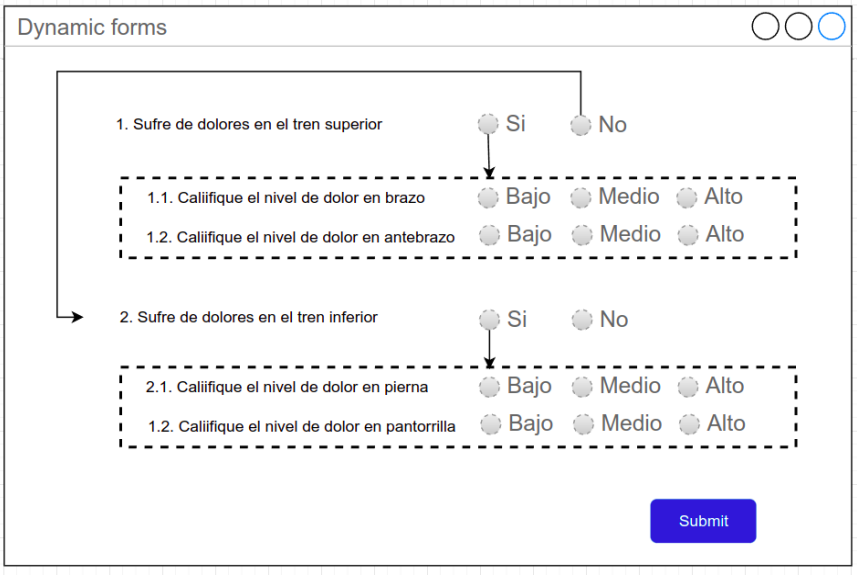
How this will be evaluated?

The user must use the add category and add subcategory in unlimited manner. Then, the enviar button will be press and the students should present the fields introduced by the user in the receving page.

Remember that you can use arrays to set the names of your form elements. For example data[$line][$field].

To recover the file in the landing page, yo can use the comand print_r($_POST[]);

2.1 Second challenge



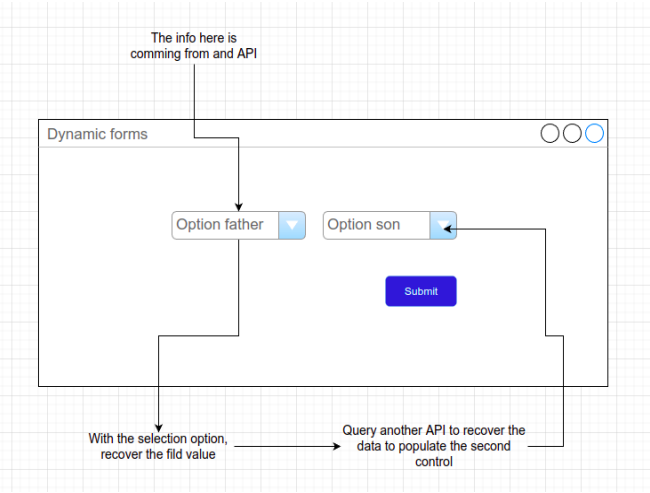
This is a form with some possible answers and, depending on the user answers the form will visualized hidden parts of the questionnaire. If the answer to question 1 is “Si” the questions

1.1 and 1.2 must be displayed. If the response to question 1 is “No” you must keep the question 1.1 and 1.2 hidden so the user focuses in question 2. The whole form must obey to the same behavior.

How this will be evaluated?

The users will fill the form in a random manner and press the buttom “Sumbit”. The receiving page will show the results of print_r($_POST[]);

2.1 Third challenge



As explained in the figure, The option father receives information from and API. You will receive and id and a string so you can populate the select control in the name and value fields. When the user selects and option in “option father” you should grab the selected value and query another API to receive the sons options of that father option. The controls must support a multi select feature.

How this will be evaluated?

The user will chose one or several options in the father control, then will go an select one or several values in the soon control and press send. The receiving page must show the results of print_r($_POST[]);