Get Your Macronutrients for a Healthy Diet Program

Manuel Flores Quiñones

20 November 2018

Contents

1. Context of the problem

a. Interest in Nutrition
b. Lack of free tools
c. Complexity of the logic
d. Complexity in other solutions

The first problem was the lack of interest in nutrition, Mexico is the second country with the highest rate of obesity (34.4% of the population is obese) and The US is the first country (with 38.2%). Even though, there are multiple government programs in Mexico that wants to deal with this issue, the culture is remarkably influenced by their neighbors (US).

One of the main reasons for not dealing with the issue is that there is a lack of interest and, since it's a delicate issue, people don't contact nutritionist in order to get a diet. In Mexico, the word of voice is primarily the first driving force for people to find a "good" nutritionist that accommodates their needs. A big problem is that, since they need to refactor their way of living by (in some cases) dramatically changing their eating habits, it's hard for a lot of people to continue with the diet.

There are diets that make you lose weight dramatically in a short period of time, but in the long run the person gains back that weight. The best diet, is the one that slowly but surely makes you lose weight or in the desired case, makes you gain it. The process of losing weight consists of a balanced diet that may consists of any type of food you like, but the trick is to always stay inside the range of the nutritional macros and always exercise.

When a person wants to lose weight or entire a diet to gain some muscle, he will encounter multiple problems. If you are interested in having a balanced diet, you might get away from paying a nutritionist, or in some cases you might be lacking the funding to do so. In this situation, the subject might want to search for some tool online. Mobile apps, or web sites might provide (via subscription or free of charge) a calculator for you to calculate your nutritional macros. This application might work well at first, but, as time goes by, become irrelevant or outdated and the user will be left as he started: With the desire for a balanced diet but doesn't know where to get it from.

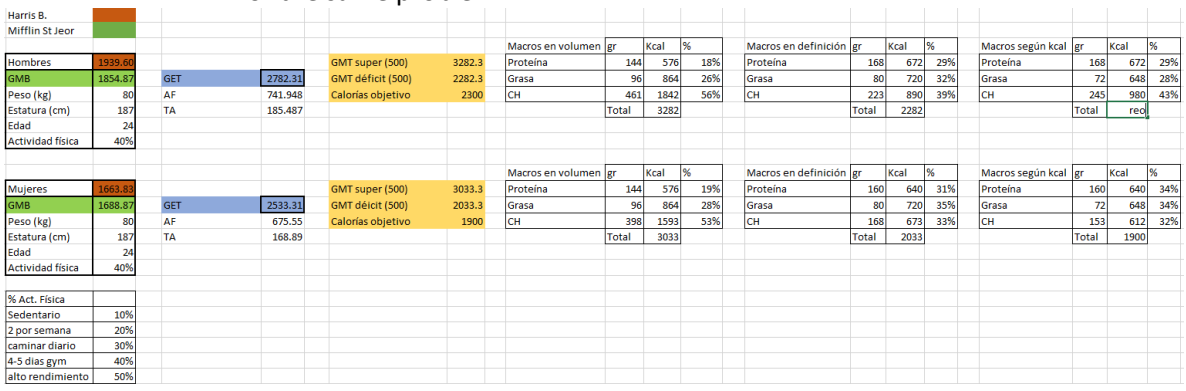
This proof of concept for an app exactly wants to deal with that problem. As a person becomes more interested in his diet and the logic behind the nutrition and the nutritional macronutrients, he will start to search more on the topic. The information about the relation between the amount of exercise and your weight loss factor is information that constantly is updated, or you simply find one value you found on an article much more accurate to your situation. The proof of concept for a prolog application that calculates macros is not the nutritional macronutrients by themselves, is the fact that any user can modify (and is encouraged to do so) as they get more knowledge.

Other sites might provide an extensive excel sheet filled with rows and columns of numbers and formulas that in the best of all cases will have some input for your information. These tools are effective as a base guide, but when the user is more experienced, they become very hard to manipulate or in the worst of cases they cannot be modified. This prolog application has that in mind for the user. It provides an application that has the basic formulas that will work for the general public, but, if a user wants to change some parameters, he will be able to do so. Thanks to the logical paradigm, this perspective of the problem becomes much more user friendly.

2. Solution

The solution to the problems and scenarios presented before is this proof of concept as a prolog application. The app itself will give the nutritional macronutrients needed in order to keep the desired weight (user input). All the problems will be attacked one by one in order to have an optimal solution for each one of them.

a. Interest in nutrition.
    i. Thanks to the accessibility, and the cost of this app (free), the user will no longer have impediments for him to start his healthy life. Still, the interest in nutrition can be achieved with much more than a user-friendly app such as this.
b. Lack of free tools.
    i. This app will become the stepping stone for any person interested in starting or continuing a healthy life. Its free, with the ability of improve upon it, expand it or change the preset parameters.
c. Complexity of the logic.
    i. The prolog application is simple for any person to read and get a grasp of wats going on behind all the logic. More than the application, the code is well documented and separated in chunks for better understanding. This had in mind the user that might want to change either parameters or formulas inside the code, and since its open source, a better version might be around the corner.
d. Complexity in other solutions
    i. Other solutions consist in detailed but confusing excels that are not user friendly at all. Here we have an example of a poorly implemented solution for the same problem.



Harris B.	
Mifflin St Jeor	

Hombres	1939.60
GMB	1854.87
Peso (kg)	80
Estatura (cm)	187
Edad	24
Actividad física	40%

GET	2782.31
AF	741.948
TA	185.487

GMT super (500)	3282.3
GMT déficit (500)	2282.3
Calorías objetivo	2300

Macros en volumen	gr	Kcal	%
Proteína	144	576	18%
Grasa	96	864	26%
CH	461	1842	56%
Total		3282	

Macros en definición	gr	Kcal	%
Proteína	168	672	29%
Grasa	80	720	32%
CH	223	890	39%
Total		2282	

Macros según kcal	gr	Kcal	%
Proteína	168	672	29%
Grasa	72	648	28%
CH	245	980	43%
Total		reo	

Mujeres	1663.85
GMB	1688.87
Peso (kg)	80
Estatura (cm)	187
Edad	24
Actividad física	40%

GET	2533.31
AF	675.55
TA	168.89

GMT super (500)	3033.3
GMT déicit (500)	2033.3
Calorías objetivo	1900

Macros en volumen	gr	Kcal	%
Proteína	144	576	19%
Grasa	96	864	28%
CH	398	1593	53%
Total		3033	

Macros en definición	gr	Kcal	%
Proteína	160	640	31%
Grasa	80	720	35%
CH	168	673	33%
Total		2033	

Macros según kcal	gr	Kcal	%
Proteína	160	640	34%
Grasa	72	648	34%
CH	153	612	32%
Total		1900	

% Act. Física	
Sedentario	10%
2 por semana	20%
caminar diario	30%
4-5 dias gym	40%
alto rendimiento	50%

    ii. The image presents a nutritional macronutrients calculator but lacks the proper documentation, references for the formulas and logic, and overall, for a normal user will become a nightmare to understand. The prolog solution provides a much cleaner implementation of the same logic, but it is much easier to understand, modify and expand than the picture shown.

3. Results

This app had three concepts in mid.

User friendly

The usability must be as easy as it could be, without filling cells in excel or calculating numbers outside of it. It's simple, write your gender, desired weight, height, age, physical activity, **GrProteins, GrFat and GrCarbohydrates**). The last three parameters are in bold since they are the variables that the program will give in return. You can see more about it in the next section.

```
?- getMacros(female, 80, 187, 24, 0.4, GrProteins, GrFat, GrCarbohydrates).
GrProteins = 160,
GrFat = 72.0,
GrCarbohydrates = 186.32625000000007.
```

Example of the program running.

Expandable

Nutrition is a complex and with area, this app only focused in the macronutrients needed to begin a diet to gain, lose or maintain the weight. The principal intent of this application is for it to grow and start to attack much more complex problems or solve some nuances that building a diet with a gym routine. Thanks to the GNU license, any person, any company is legally able to use this code and make it something much bigger and that helps more areas than the introduction for a healthy life.

Modifiable

Other solutions (like the excel example) typically use strictly unmodifiable formulas out of the reach for the user, this limitation is a problem because it sticks to one way of solving it. Nutrition is very ambiguous when it comes to what numbers works with a person, one might find a research and the relation between kilogram to one gram of protein to be 1 to 7 but others find it much closer to their proportions in another way. Thanks to the logical paradigm it is very easy for a person to change those basic relations with a simple tweak. Or, there might be cases where you can't to add more to the relations, lets say you want to make your fat cat a more agile and fit one, there is no way you would be able to use the same excel that you use for a human to have a cat´s diet. Prolog is very kind in these situations, just add some lines to de knowledgebase and you are all set.

```
% -----KNOWLEDGE BASE-----
%This is the relation of one kilogram to grams of Proteins
kb_grProteins(male, 2.1).
kb_grProteins(female, 2).
%This is the relation of one Kilogram to grams of Fat
kb_grFat(male, 0.9).
kb_grFat(female, 0.9).
```

Example of program running.

Now, let's say you want to add your cat and you already made the research needed to get the relations for proteins and for the fat. The only thing you'll need to do is add a simple line beneath the "kb_grProteins(female, 2)." And add the same premise but with the "cat" as first parameter followed by a comma and the number you want to add. Take in mind, this relation is based of 1 kilogram of weight to 1 gram of protein in this case. So, if you find another equivalence you must make the conversion.

```
% -----KNOWLEDGE BASE-----
%This is the relation of one kilogram to grams of Proteins
kb_grProteins(male, 2.1).
kb_grProteins(female, 2).
kb_grProteins(cat, 1.4).
%This is the relation of one Kilogram to grams of Fat
kb_grFat(male, 0.9).
kb_grFat(female, 0.9).
kb_grFat(cat, 0.5).
```

Example of the addition of the cat relation.

By combining these 3 rules, the program is user friendly, expandable and modifiable. The result is a prolog code that has the goal of introducing more people to the world of nutrition and to set the bases of what a diet means.

At this point, there is only one more thing to talk about besides the conclusions and the improvements opportunities: Why Prolog?

4. WHY PROLOG?

Getting the nutritional macronutrients consists in a series of formulas applied to a series of relations between one kilogram and one gram of fat, proteins and carbohydrates. During the design state of the development cycle there were a lot of different languages and paradigms analyzed to see which one suited best this implementation of the solution. At the end, there were two contenders, scheme with the functional paradigm and prolog with logical. The result was made in prolog since it was the best choice for the solution to be user friendly, expandable and modifiable.

a. Prolog to be user friendly

The user experience in prolog completely depends on how you develop, name and make the functions and variables. Since it's a language that has a lot of freedom you could make a program that only you could understand, in the other hand, compared to other languages that needs basic knowledge of syntaxis and how declaration works, prolog works more naturally. Prolog (since its logical) operates via premises and conclusions, a basic principle that any person may understand. An example is the relation between one kilogram and one gram of protein, the relation is 1 to 2.1 for a male. Other languages would only include this logic using variables and multiplications/divisions to calculate the result. In this program, the relation is much simpler and instead of leaving it as a relation of 1 to 2.1 it went one step further. Dealing with strings comparisons and relations with integers in other languages might get confusing but in prolog it is simpler.

```
kb_grProteins(male, 2.1).
kb_grProteins(female, 2).
```

In this example we can see that there are two "logical relations" with the same name but deferent parameters, one means that for a man, one kilogram is 2.1 grams of proteins and the second one is that same relation but fur a female. This is called a "knowledgebase" and is the base for all the logical relations in the program, the stepping stone in the calculations for the macronutrients.

b. Prolog to be expandable.

This program is the beginning of a bigger system that has the capability of grow as big as you want. Prolog and its knowledgebase make this process much easier by allowing anybody to add more relations and complex methods on top of the ones we have, and since prolog calculates every single possible answer, who knows? Maybe you would develop an addition module that can give you a complete diet based on the micronutrients using the relations of a food product with the contents of proteins, carbohydrates and fat it contains.

The only limit for the program's future is the size of the knowledgebase!

c. Prolog to be Modifiable.

Since prolog is made around the idea of premises and conclusions, then, it is very easy to modify, tweak and improve the functions and the knowledgebase. The logical paradigm helps stablish the ground for simple parameter modification by changing the premises in the knowledgebase of the program. Thanks to this approach it is very easy for a person to change relations, add new ones or even add small chunks of code for example that lets you introduce libras instead of kilos. A simple addition to the knowledgebase and some minuscular changes will make this change possible and with little to no time in development.

5. Conclusions

Developing this app help me realize how hard is for people that is interested in getting fit but lacks either more motivation or the funds necessary to hire someone. During the first research I encountered a lot of sites that claimed to have “the magic diet” but it was nothing but a starvation way of living, I asked numerous professionals like the nutritionist at the Tec de Monterrey Campus Queretaro and she told me that the best diet is the one that makes you lose or gain weight at a slower pace. This made me think about all those diets that made you lose weight like crazy in one week, but you gained even more the week afterwards. This claims even became much more apparent when I asked a family member that just got his personal trainer certificate and official ID, he showed me his way of calculating the basic diet and I realized the potential this app has. Scrolling through a nested excel is not the best experience and hiring an expert for a basic diet is not optimal either. So, I said “Why not developing an easy tool, one input, and one diet as output, simple yet complex for people that could improve it and make it better?” so I did. I built the foundation of what could become a very sophisticated peace of software that could help the scholars of nutrition and the new comers that only one a straight answer.

6. Areas of improvement

Of course, am no genius and I having no unlimited time to develop this foundation, so problems might appear or some areas of improvement.

The first version only worked with men’s calculations but adding the female section was no hard task, I would say, it was much more time writing it down than thinking about it, during that process I realized one of the weaknesses of the language itself. Since prolog works with immutable objects (objects that cannot be modified) you always need new ones for the results and there is no such thing as “passing a parameter or passing a previous calculated variable” the code must be crafted delicately in a way to calculate the variables in a sequential way and declaring them inn the right order. And how can I forget about the long parameters that the main functions have. See, when I said that prolog uses immutable objects, it was not a compliment. Thinking around that limitation is hard, and it might be a push over for new comers for the language, having that limitation means that for an object that adds A and B you need A, B and C as parameters, but, if A is calculated with X and Y the function for C would need A, B, C, X, Y because that’s how it works.

At the end of the day, I tried to make a simple program that saves a lot of time and effort while preserving the pro of expanding it. Right now, the program can calculate a diet for any kind of living thing on the planet (if you have the data for the KB of course), but there are a lot of areas of improvement, like adding support for other measurement systems such as using ft instead of cm or lbs. instead of kg. I think, my original research made this easier because I cannot imagine implementing this in scheme (parenthesis everywhere).

Another thing I would like to add is a UI (User Interface) since, yes, I get it, writing this line of code is not optimal.

```
getMacros(female, 75, 156, 21, 0.3, GrProteins, GrFat, GrCarbohydrates).
```

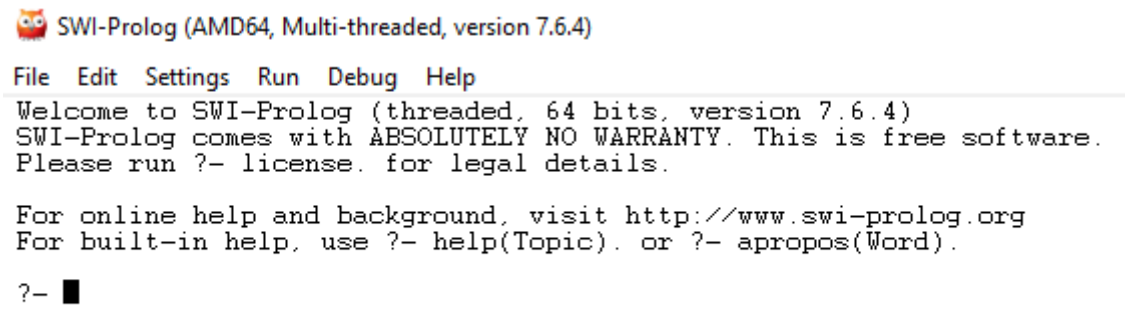
But using the prolog terminal requires you to add the parameters that you want to get, other wise you will get a pretty but insufficient **True.**

The UI might mitigate this issue by in the back end calling the function that way but the user only enters the inputs and press “calculate”.

7. Set Up

This program works with prolog and is only runnable with the prolog compiler. The steps in order tu set up the compiler, run the code and use it are as follows:

1. Enter to this link: http://www.swi-prolog.org/download/stable and select your corresponding package. This is needed since it’s the prolog compilator needed in order to run the program.
2. Once you have prolog installed you must open it. A white screen will appear with little to no words, like this one:



3. To open the file, you must press File>Consult and then you select the project. If you want to modify it or expand it, simply select edit and open the project.
4. Once you’ve opened the project, the screen will be the same but this time you will be able to do a simple command to get your macronutrients. This is a simple example.

```
?- getMacros(female, 75, 156, 21, 0.3, GrProteins, GrFat, GrCarbohydrates).
```

5. The command *getMacros* will need to be invoked with 8 parameters, no less, no more. The first parameter is your gender stated as “male” or “female” in case you added more types to get macronutrients like for dogs or cats, you must invoke it with the name you wrote. The second parameter is your desired weight, this is a positive number of the weight you want to maintain or your goal. The third parameter is your height in cm, this is a positive number in centimeters. The fourth parameter is your age, this must me a positive integer.
6. The next parameter is your physical activity, you must follow the next table in order to fill this part of the input. The number must be like the example, the percentage MUST be divided by 100 and with a cero before de point. Example: if you have 30% it will be 0.3, don’t forget the cero before the “.3”

Physical Activity	
Sedentary	10% (0.1)
Two times a week	20% (0.2)
Daily walking	30% (0.3)
4 – 5 going to the Gym	40% (0.4)
High Performance Athlete	50% (0.5)

7. The next three parameters will be the result diet, the order is as follows
    a. Grams of Proteins
    b. Grams of Fat
    c. Grams of Carbohydrates
8. Its recommended to use the names as stated in the example (GrProteins, GrFat, GrCarbohydrates) since they are very easy to understand, but in case you want to writte something different just take in mind the next considerations:
    a. The order will be the same no matter what you write, if you write grProteins in the slot of Carbohydrates you will have the value of the protein printed as if it was the carbohydrates, so be careful.
    b. The name MUST be in Upper case, if not you will get a sad “False” as a return.
    c. The name of the variable must NOT be separated with a space or any other symbol besides “_”.
    d. DO NOT write lower case words or numbers, otherwise you will get a *False*.
9. Special considerations.
    a. This program operates under the GNU license, for more information consult the page: https://www.gnu.org/licenses/gpl-3.0.en.html.

10. References

Arija, V. and Ortega, R. (2018). Valores de referencia de ingesta dietética y de antropometría en estudios poblacionales. [online] Renc.es. Available at: http://www.renc.es/imagenes/auxiliar/files/RENC2015supl1VREF.pdf [Accessed 20 Nov. 2018].

Bachus, T. (2018). How to Determine Macronutrient Needs Based on Goals. [online] Acefitness.org. Available at: https://www.acefitness.org/education-and-resources/professional/expert-articles/5904/how-to-determine-macronutrient-needs-based-on-goals [Accessed 20 Nov. 2018].

MBMK.com. (2018). Macronutrients: Calculating Your Proteins, Fats & Carbs. [online] Available at: https://mybodymykitchen.com/calculate-your-macronutrients-protein-fats-carbs/ [Accessed 20 Nov. 2018].

National Institute of Diabetes and Digestive and Kidney Diseases. (2018). La nutrición y la pérdida de peso: mitos y verdades | NIDDK. [online] Available at: https://www.niddk.nih.gov/health-information/informacion-de-la-salud/control-de-peso/nutricion-perdida-peso-mitos-verdades [Accessed 20 Nov. 2018].

OCDE. Obesity Update. 2017. Disponible desde: http://oment.uanl.mx/descarga/obesity-update-2017_ocde.pdf

OMENT. (2017). México ocupa el 2º lugar en obesidad en adultos según la OCDE | OMENT. [online] Available at: http://oment.uanl.mx/mexico-ocupa-el-2o-lugar-en-obesidad-en-adultos-segun-la-ocde/ [Accessed 20 Nov. 2018].

Overvoedingengezondheid.nl. (2018). Sport nutrition: A review of the latest guidelines for exercise and sport nutrition from the American College of Sport Nutrition, the International Olympic Committee and the International Society for Sports Nutrition. [online] Available at: https://www.overvoedingengezondheid.nl/wp-content/uploads/2015/03/88379-219821-1-PB.pdf [Accessed 20 Nov. 2018].

Seom.org. (2018). Tablas de Recomendaciones. [online] Available at: https://seom.org/seomcms/images/stories/recursos/infopublico/publicaciones/soporteNutricional/pdf/anexo_05.pdf?fbclid=IwAR3MK8YTNCAPxZw5QEGNhHmrsvM344rCE5PHUIlns3AQdtodlcaygfft9Fw [Accessed 20 Nov. 2018].