

El plato perfecto para comer de *manera saludable*, o **Plato de Harvard**, ha sido creado por expertos en nutrición de la Escuela de Salud Pública de Harvard. Es una guía para crear comidas saludables y equilibradas.

La **mitad del plato** deber ser de vegetales o de fruta. Ésta, cuanto más variada mejor. Y los vegetales igual. Las papas, en cualquiera de sus formas, no se consideran como verdura.

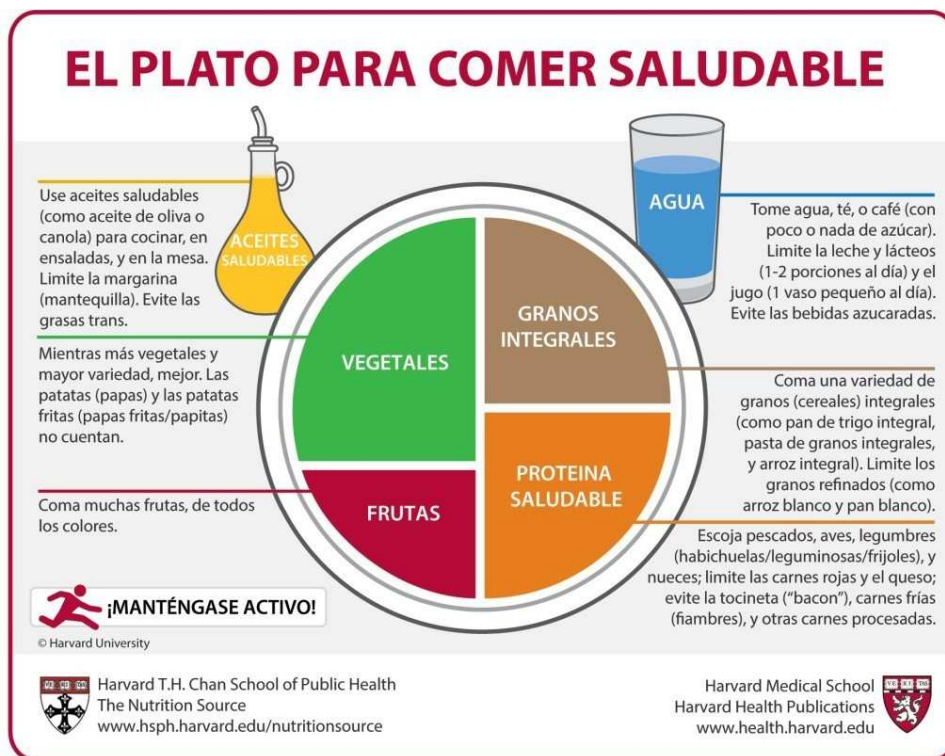
Un **cuarto del plato** debe ser de productos integrales, ya sea arroz, pasta, pan. En la medida de lo posible hay que limitar el pan blanco, el arroz blanco.

Otro **cuarto del plato** lo deben formar proteínas, pollo, pescado, pavo, nueces, legumbres, etc. Hay que limitar las carnes rojas, el cerdo; también los embutidos y las carnes procesadas.

Para beber, agua, café o té. Hay que evitar las bebidas azucaradas. Los zumos, en vaso pequeño, y la leche o lácteos, no más de dos porciones al día.

El aceite que utilicemos, o los diferentes condimentos, cuanto más saludables mejor. Un buen aceite de oliva virgen extra, si quieres una salsa mejor casera. La calidad de los productos empleados le dará el aporte de garantía nutricional al plato.

El uso de la mantequilla debe quedar desterrado.



1. Diseñar e implementar un Lenguaje de Dominio Específico *Domain Specific Language - DSL*, siguiendo la filosofía de Ruby que permita la definición de **Platos Harvard**.

Los DSL son herramientas útiles que permiten expresar fácilmente la lógica específica de un problema particular (dominio) que de otro modo sería difícil o farragoso de escribir en otro idioma. Por lo general, se trata de definir una gramática que se asemeja más al léxico utilizado por el dominio de destino. Por ejemplo, un matemático que trabaja con matrices no piensa en bucles, iteradores o arrays, sino que piensa en términos de vectores, productos y transformaciones.

El uso de un lenguaje de propósito general, como Ruby, con sólo arrays e iteradores requeriría que el matemático practicara gimnasia mental para traducir mentalmente entre el dominio de su problema y el del lenguaje con el que escribe el código (Ruby). El uso de un DSL diseñado para las operaciones que le interesan eliminaría esta traducción mental y proporcionaría un código más conciso.

Los DSL tienen dos formas: externos e internos. Los DSL externos existen independientemente de cualquier otro lenguaje. Los DSL internos están alojados dentro de otro lenguaje de programación - por ejemplo, Rails es un DSL interno que se aloja en el lenguaje de programación Ruby.

El DSL que diseñe ha de permitir la definición de platos de una forma natural. Por ejemplo:

```
lentejas_arroz = Plato.new("Lentejas con arroz, salsa de tomate, huevo y plátano a la plancha") do
  vegetal      "Tomate",
               :porcion => "2 piezas pequeñas"
  fruta        "Plátano",
               :gramos => 20
  cereal        "Arroz",
               :porcion => "1 taza"
  proteina      "Lentejas",
               :porcion => "1/2 cucharón"
  proteina      "Huevo",
               :porcion => "1 pieza"
  aceite        "Aceite de oliva",
               :porcion => "1/2 cucharada"
end
```

2. La salida del ejemplo anterior podría ser como la siguiente:

```
Lentejas con arroz, salsa de tomate, huevo y plátano a la plancha
=====
Composición nutricional:
```

	glúcidos	proteínas	lípidos	valor energético
Tomate	1.0	3.5	0.2	56.4
Plátano	1.2	21.4	0.2	182.4
Arroz	6.8	77.7	0.6	1021.2
Lentejas	23.5	52.0	1.4	1538.0
Huevo	3.3	4.8	3.2	72.32
Aceite de oliva	0.0	0.2	99.6	39.92
Valor energético total				2910.24

3. Crear una cuenta en rubygems.org. El ‘handle’ que se solicita es el nombre de usuario.

4. Almacenar las credenciales de rubygems.org en la máquina local.

```
curl -u qrush https://rubygems.org/api/v1/api_key.yaml > ~/.gem/credentials; chmod 0600 ~/.gem/credentials
```

5. Puesto que ya se ha terminado de escribir código, se está listo para construir y publicar la gema.

- Para *construir* la gema, desde el directorio raíz creado con Bundler ejecutar: **rake build**
- Para crear la *version 0.1.0* de la gema ejecutar: **rake release**
- Para *instalar* gema ejecutar: **rake install**
- Para comprobar que ha ido bien la instalación, ejecutar: **gem menu**

6. Escribir la dirección HTTP de la gema que ha creado en la tarea del campus virtual.