Tarea 1: Evolucion y arquitectura de las aplicaciones web   Albrand Aguirre Marc

Arpanet:
In the beginning, the network counted with 4 computers distibuited between different universities of the country. Two years later, It counted with 40 computers connected each one with the other one. So much was the growth of the network that thei comunication sistem became obsolete. To fix this two investigators created the protocol TCP/IP, wich it turned into the standard of comunication inside the informatic networks

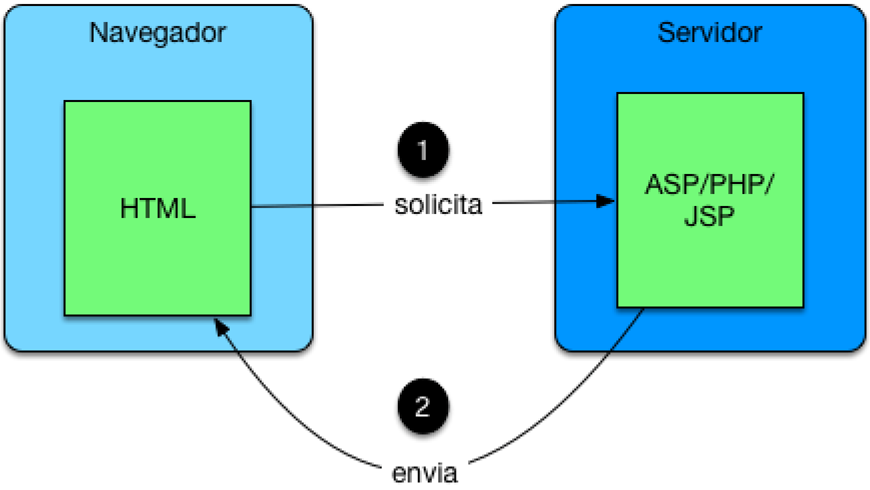
Arpanet continued growing and opening itseflf to the world, and every person with academic purposes or of investigation could have acces to the network.

With the time, military functions were detached from Arpanet and went working to MILNET, a new hnetwork created by the United States of America.

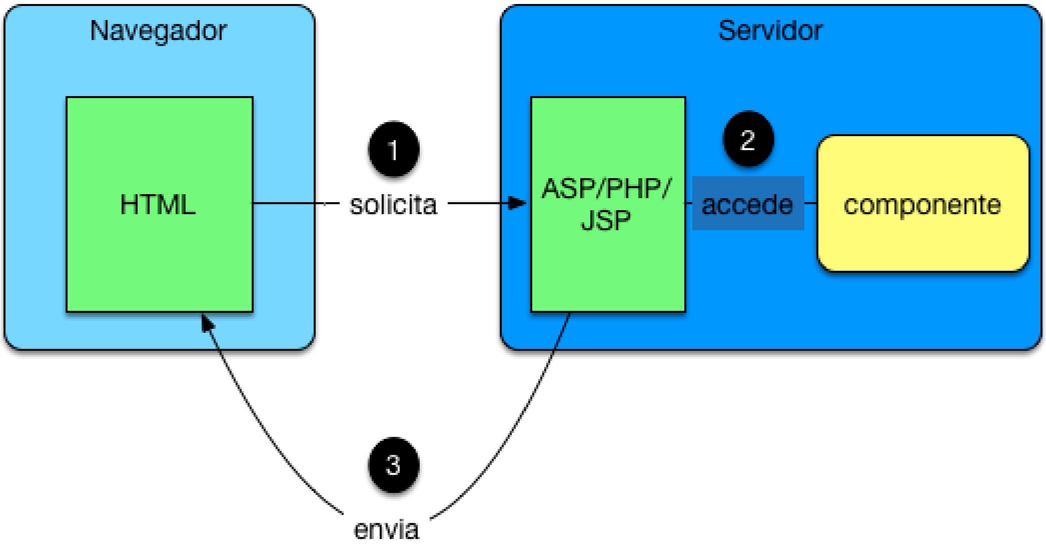
The National Science Fundation created it's own informatic network named NSFNET, that later absorbed Arpanet, creating a big network with scientific and academic purposes.

INTERNET
The development of the networks was huge, and they were created new networks of free acces that later they will unify with aNSFNET, forming the embryo of what today we know as Internet.

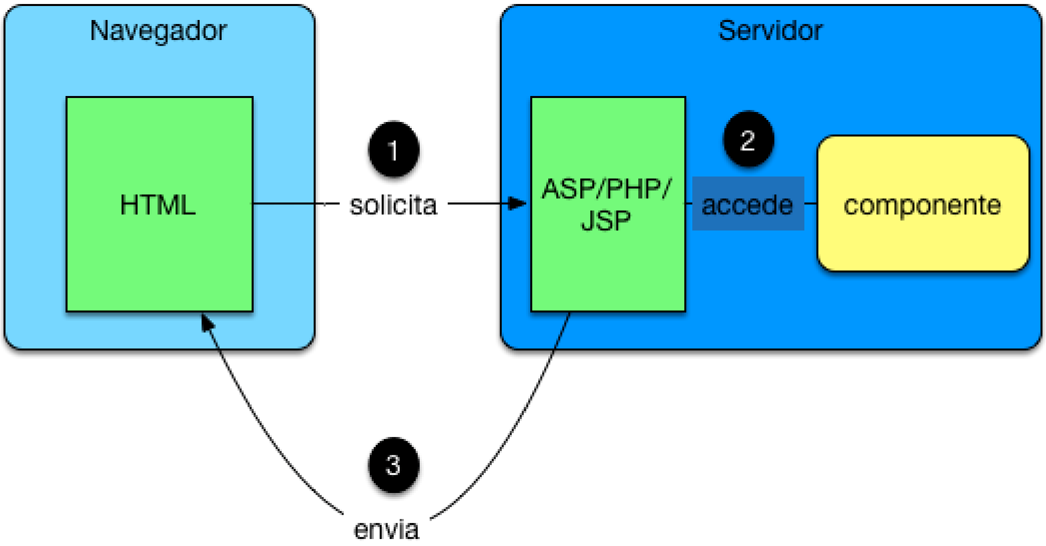
Cero model and spaguetti code:
This is a way to call the initial solutions of client/server in with we have a JSP/ASP/PHP page that conects to a database and generates a new html content. “Spaguetti code”.
An advantage of this is it's simplicity seeing it from an architecure level, and as a disadvantage it's poor flexibility and null capacity of reutilization.



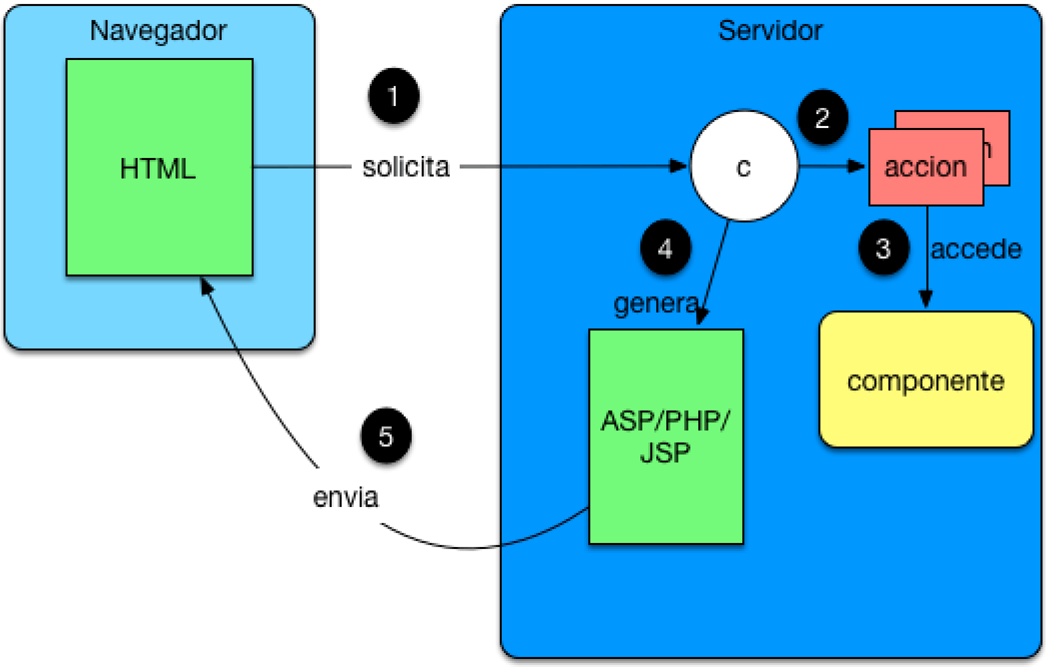
Model 1:
Promotes the modularizacion and the use of compnents through oriented objects programming. “Huge step”.
The fundamental key was to generate components in the backend cape and raise the reutilizacion in that part.



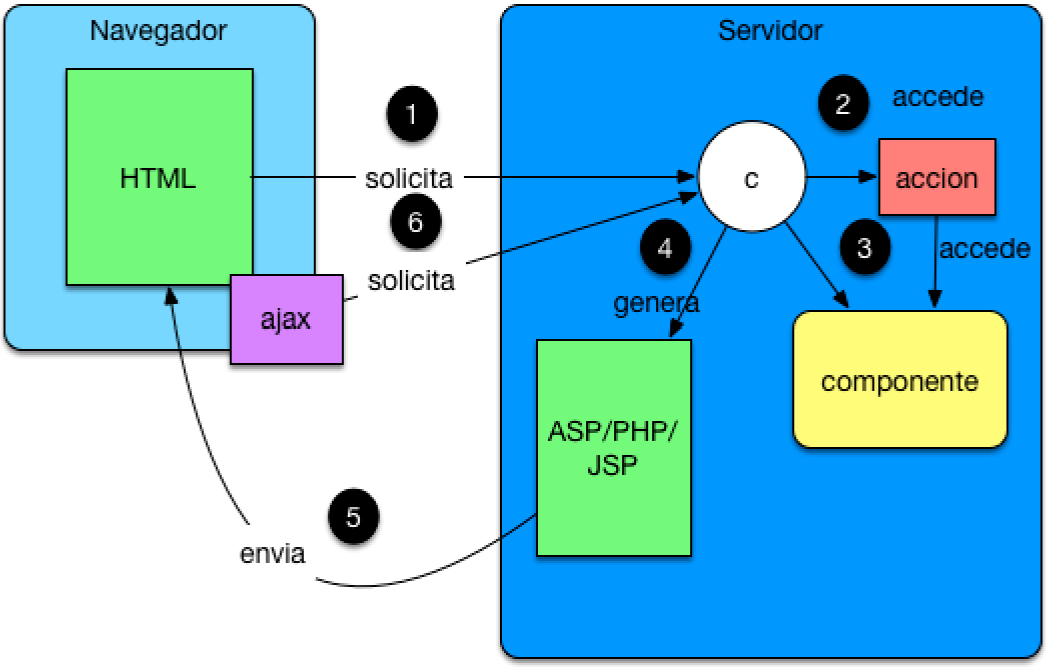
Model 2 (MVC):
Maybe the most important model, it is between the separation of responsabilities from View, Controller and Model. Practically all web frameworks had implemented this focus a way or another.



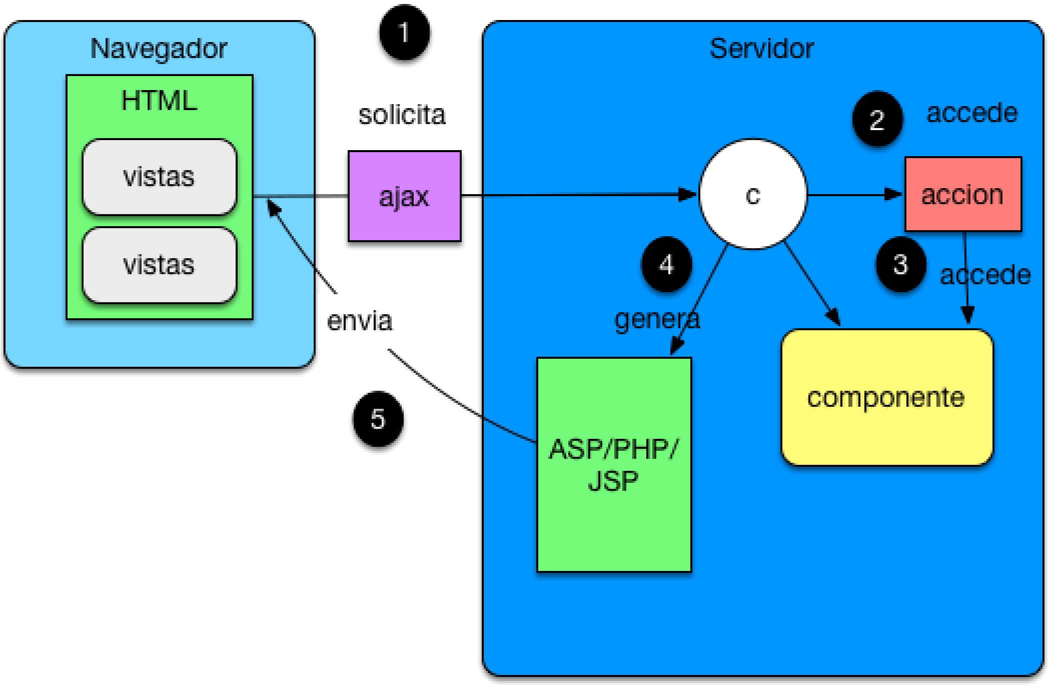
Model MVC 2 FrontController/Router:
It goes for an architecture in wich there's only a principal controller and it manages all through accions.



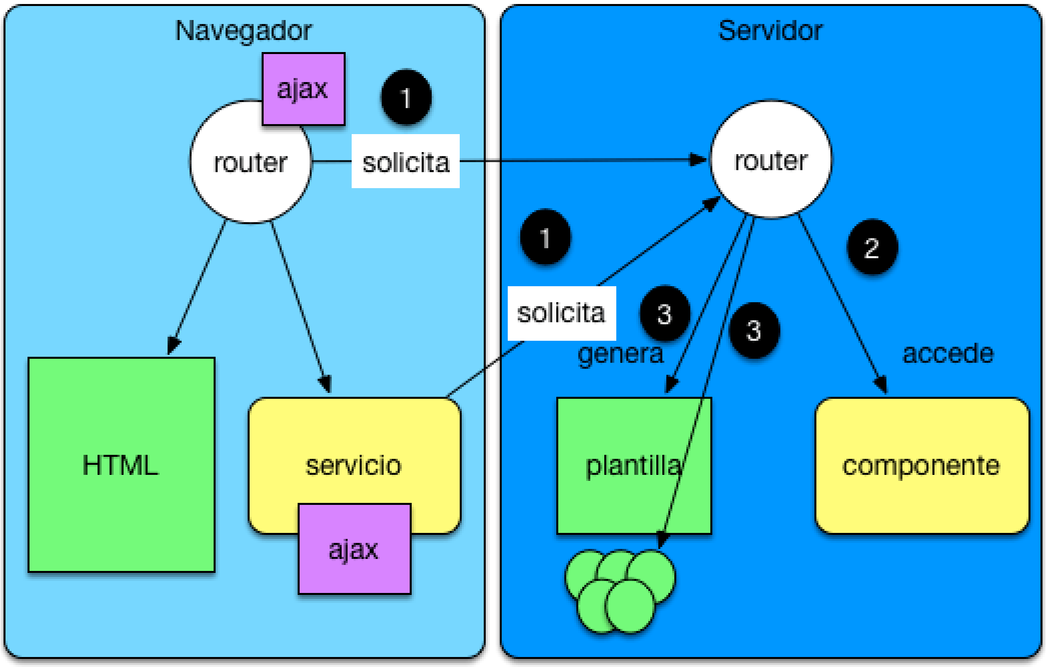
WEB and Ajax Architectures:
Until this moemnt all the evolutions were produced at the side of the server. The client's side had few news. In this situacion is when Ajax comes on as a technology to make better the performance with client and server. This supposed a true way to program.



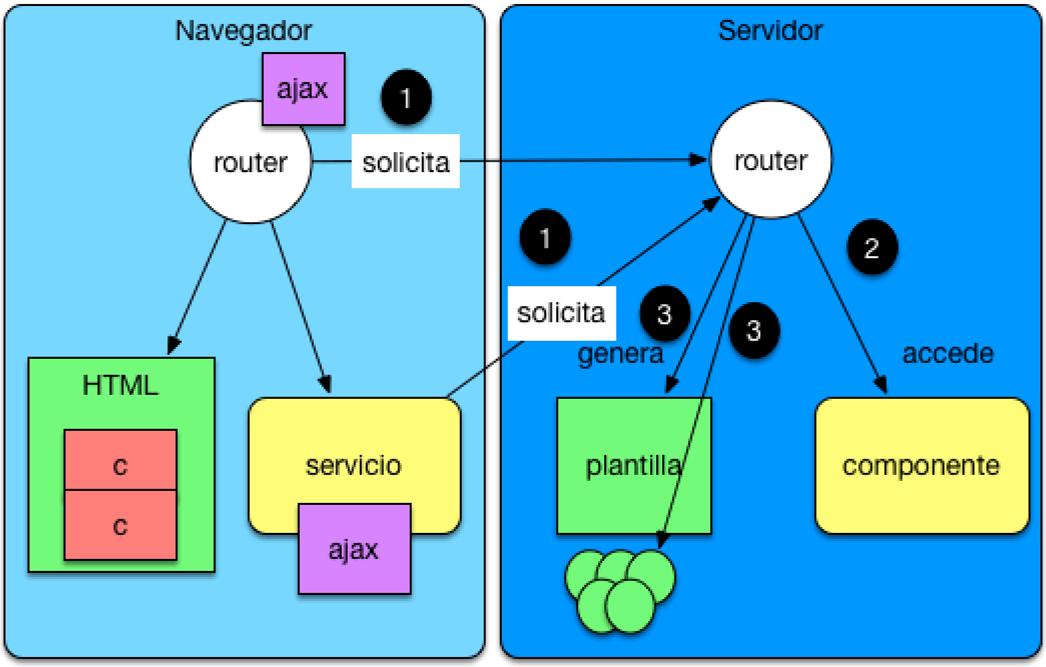
Architecture WEB and SPA:
With the arrive of the movil world and the need of having the web applications even more deconected, arise the architecture SPA (Simple Page Application). Their principal proposal is to give a mayor responsabilitie to the navigator for it to get in charge of charging the views and data using Ajax



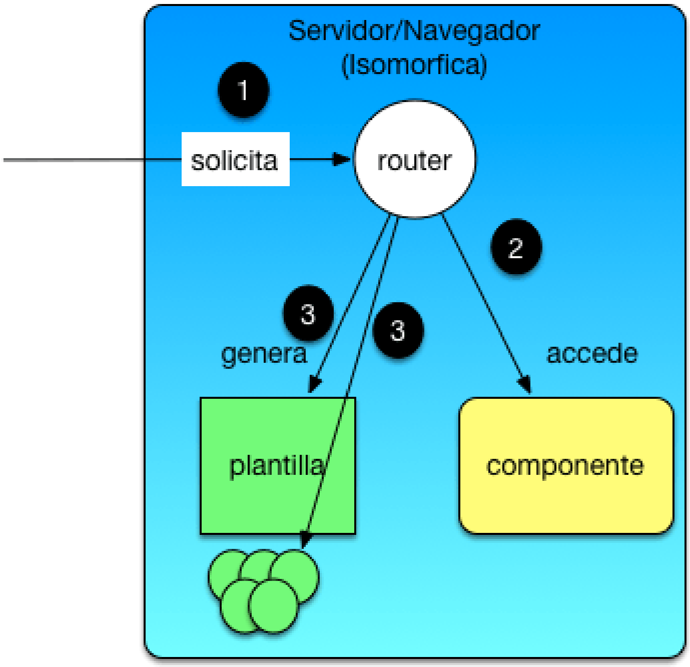
Architectures SPA MVC:
Little by little the client beggins to have more weight in the development's and needs to organize better the javascript code. The first MVC frameworks for clients appear like Backbone.js that allows to divide the responsabilities the same way as in the server.



Architectures SPA MVC and use of components:
This architectures beggin to mature quickly and technologies like Angular.js appear, they promote the use of the MVC model and the utilization of components in labels of presentation. Complementary libraries like React appear wich they center in this last things.



Isomorphic Web Architectures:
Right now we are entering in another phase, where Isomorfic JavaScript beggins to arrive. If we take a look in the last diagram the part of client and the part of server they are very similar. What would happen if in the case that both parts where implemented in JavaScript?. Well that probably a lot of code could be shared and according it would execute the application on client or server the behavior should vary.


Servidor/Navegador
(Isomorfica)
1
solicita
router
2
3
3
genera
accede
plantilla
componente