



php | **VIM**, el **IDE** cómo programar más rápido

Victor Guardiola

Responsable Técnico producto Hoteles @ ATRAPALO.COM

Manuel Aguilar

Responsable Técnico productos Common & Internacional @ ATRAPALO.COM



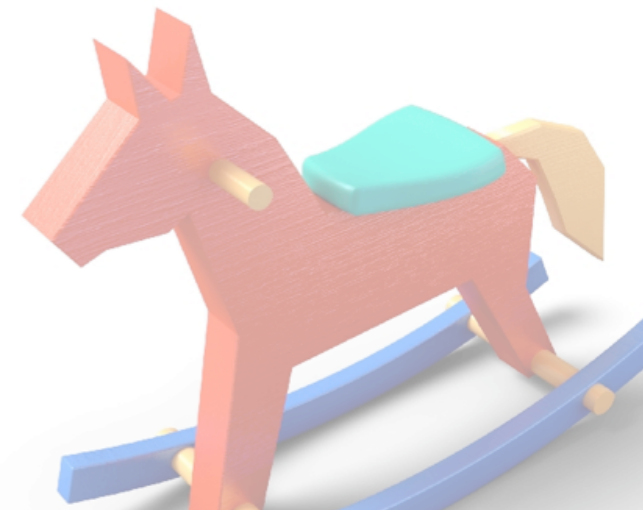
Qué vamos a tratar

1 ¿Qué es VIM?

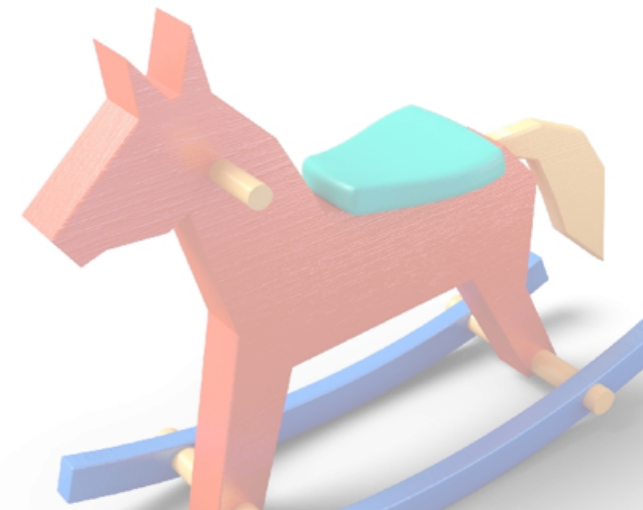
2 Modos

3 Edición avanzada

4 Extensiones

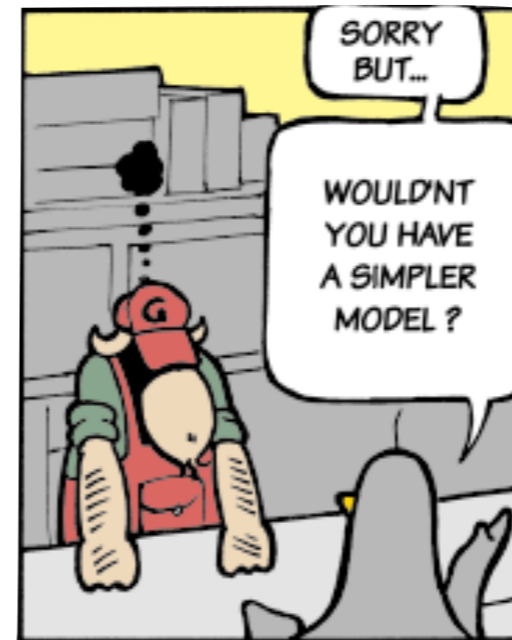
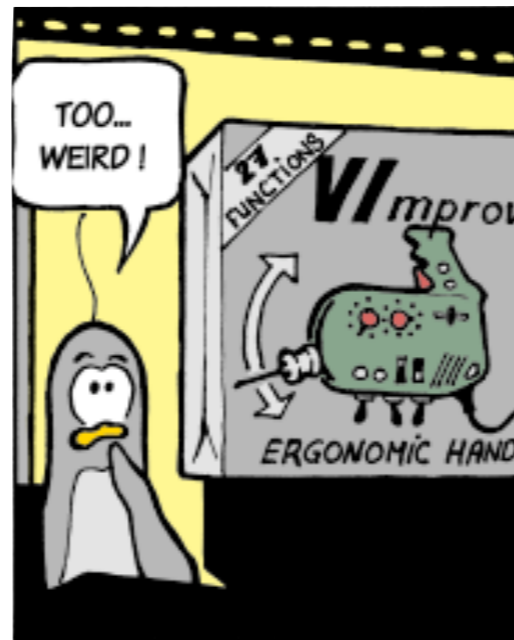
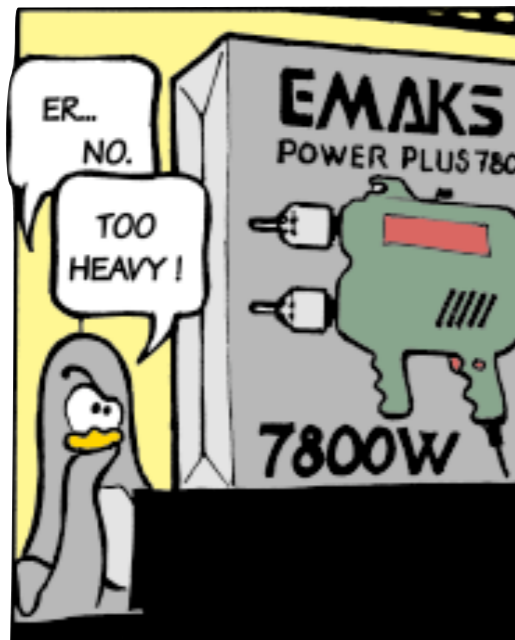


¿Qué es VIM?



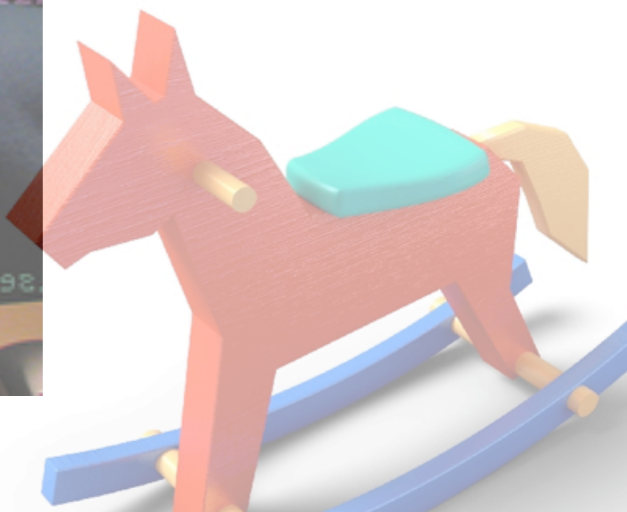
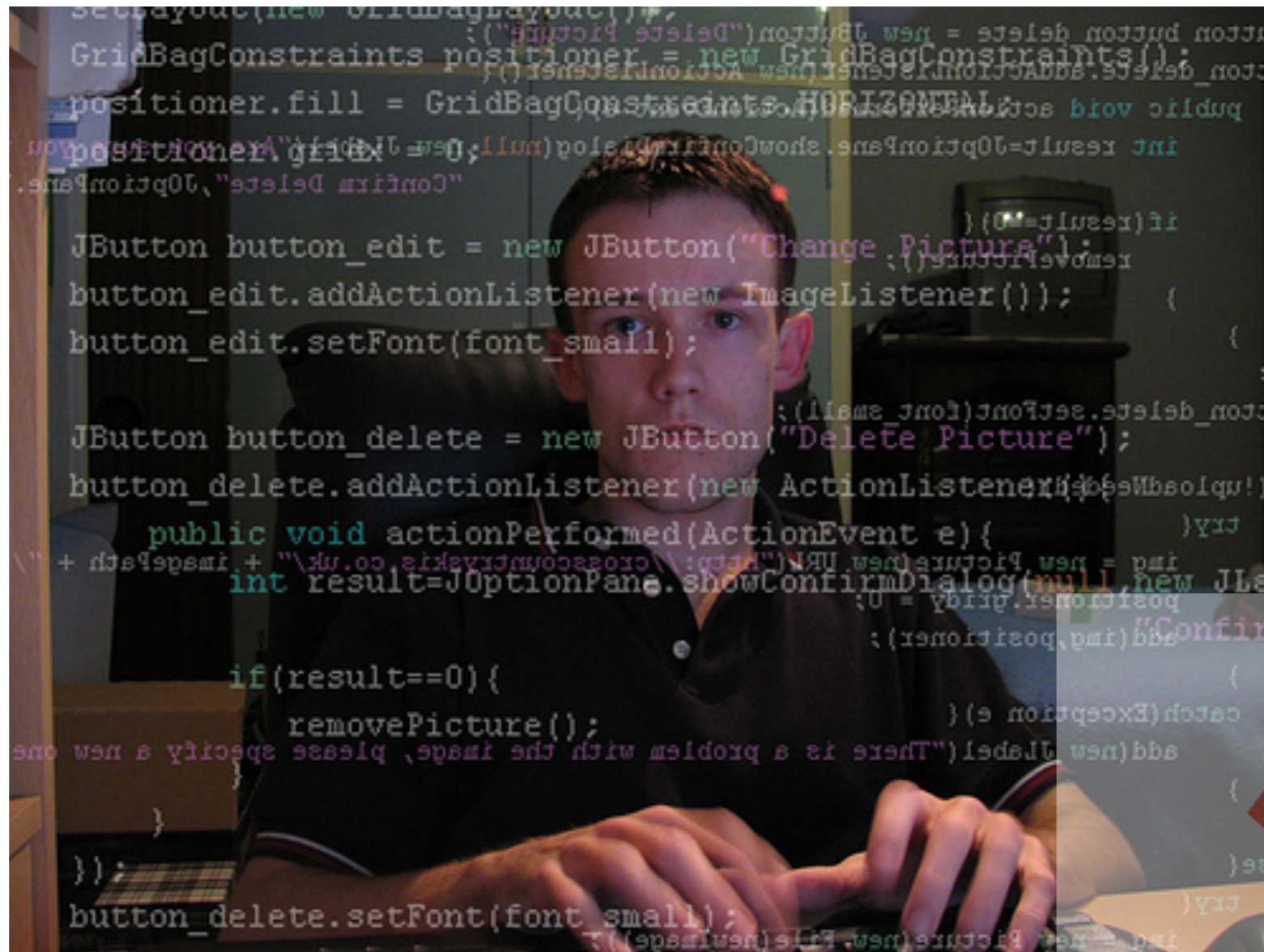
¿Qué es VIM?

Wikipedia: **VIM** (del inglés *Vi IMproved*) es una versión mejorada del editor de texto **vi**, presente en todos los sistemas UNIX. Su autor, Bram Moolenaar, presentó la primera versión en 1991, fecha desde la que ha experimentado muchas mejoras. La principal característica tanto de Vim como de Vi consiste en que disponen de diferentes modos entre los que se alterna para realizar ciertas operaciones, lo que les diferencia de la mayoría de editores comunes, que tienen un sólo modo en el que se introducen los comandos mediante combinaciones de teclas o interfaces gráficas.



¿Por qué VIM?

Porque en nuestra época como programadores, **pasamos la mayor parte de nuestra vida delante de una pantalla, delante de un editor.**



¿Por qué VIM?

- ▶ Por tanto, invertir tiempo en encontrar nuestro editor de código ideal es muy importante
- ▶ Idealmente **debemos usar el mejor editor de código** que encontremos:
 - nos ha de ayudar a introducir código lo más rápido posible (el tiempo es oro)
 - nos ha de ayudar a tener código libre de errores
 - idealmente ha de integrarse con otros procesos para no obligarnos a estar cambiando continuamente de aplicación
- ▶ Pero, básicamente, el editor que escojamos ha de cumplir siempre este objetivo: **no interponerse entre nosotros y el código**, pasar desapercibido y ser lo más discreto posible
 - nunca hemos de utilizar un editor cuyas características **nos condicionen** en nuestra programación



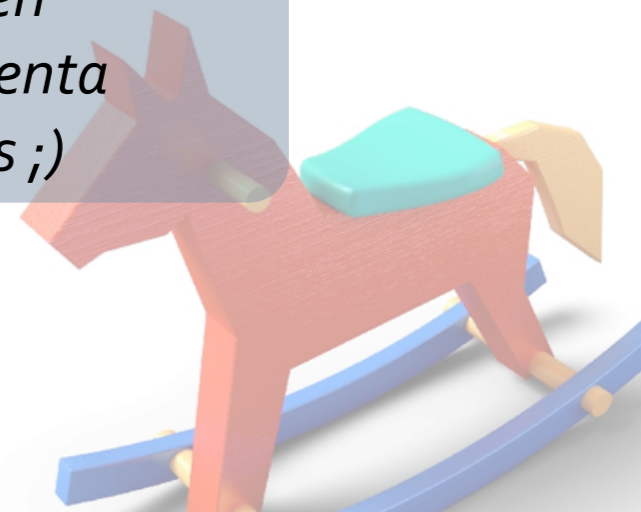
¿Por qué VIM?

-Porque lo dicen en los foros de phpbarcelona.org :-p

*Para gustos, colores... pero si ya dominas lo básico de vim, yo te recomiendo que le eches un ojo como editor PHP. **Puedes llegar a ser realmente muy productivo.** Yo lo uso y estoy muy muy satisfecho.*

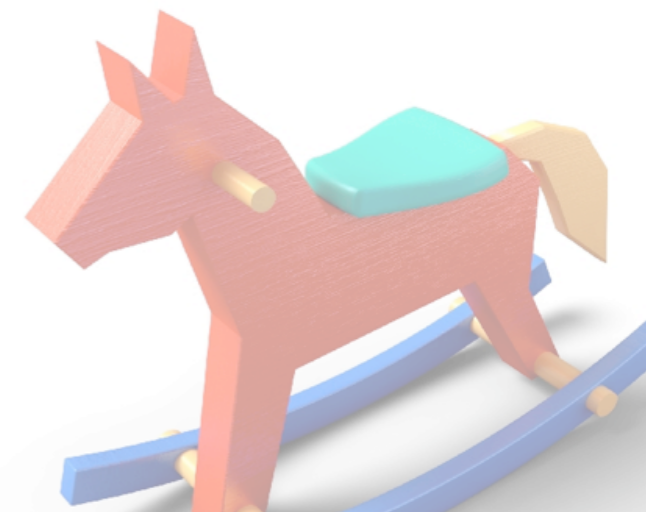
***Si cambias de lenguaje, no tienes que cambiar de IDE :).** Yo por ejemplo empecé a usar vim para PHP/smarty, y ahora soy un feliz usuario de vim para programar en python*

***Ligero.** Puedes tener sin problemas 7 u 8 ventanas abiertas de vim en proyectos diferentes, tareas diferentes, lenguajes diferentes, Intenta hacer eso en un IDE integrado (del estilo Eclipse) y luego me cuentas ;)*



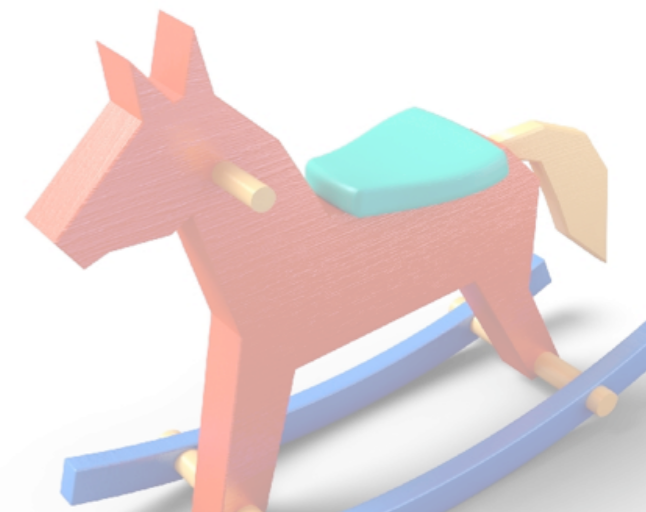
Algunas características destacadas de VIM

- ▶ ¡**RÁPIDO**, rápido, rápidoooooooo!
- Al contrario que los IDEs java al uso (Eclipse, Zend Studio)
- No **consume apenas recursos**, funcional en los entornos más modestos
- ▶ Totalmente **operable vía teclado** (no se necesita ratón en ningún caso)
- ▶ **Centenares de atajos de teclado**, no hay dos usuarios que lo manejen igual
- ▶ **+2000 extensiones** (plugins) a nuestra disposición (<http://www.vim.org>)
- ▶ Pero has de aprender a hablar en su lenguaje (**gran curva de aprendizaje**)
- ▶ Proyectas a los demás la imagen de que controlas muuuuucho :-)
- ▶ VIM es vírico: te encontrarás pulsando la tecla ESC instintivamente en el resto de aplicaciones cuando quieras salir de una orden :-p



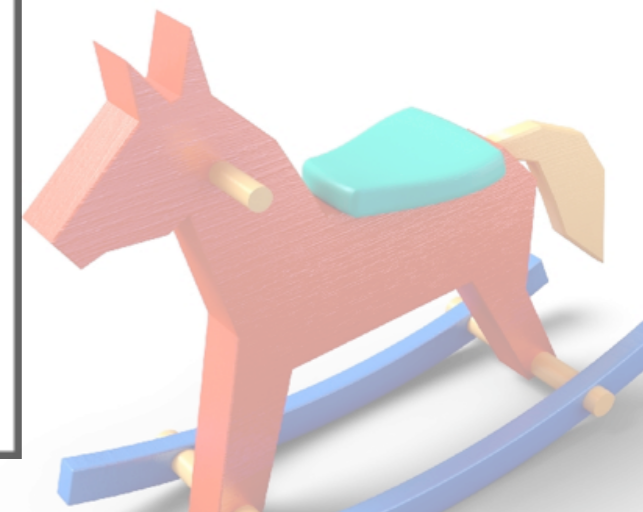
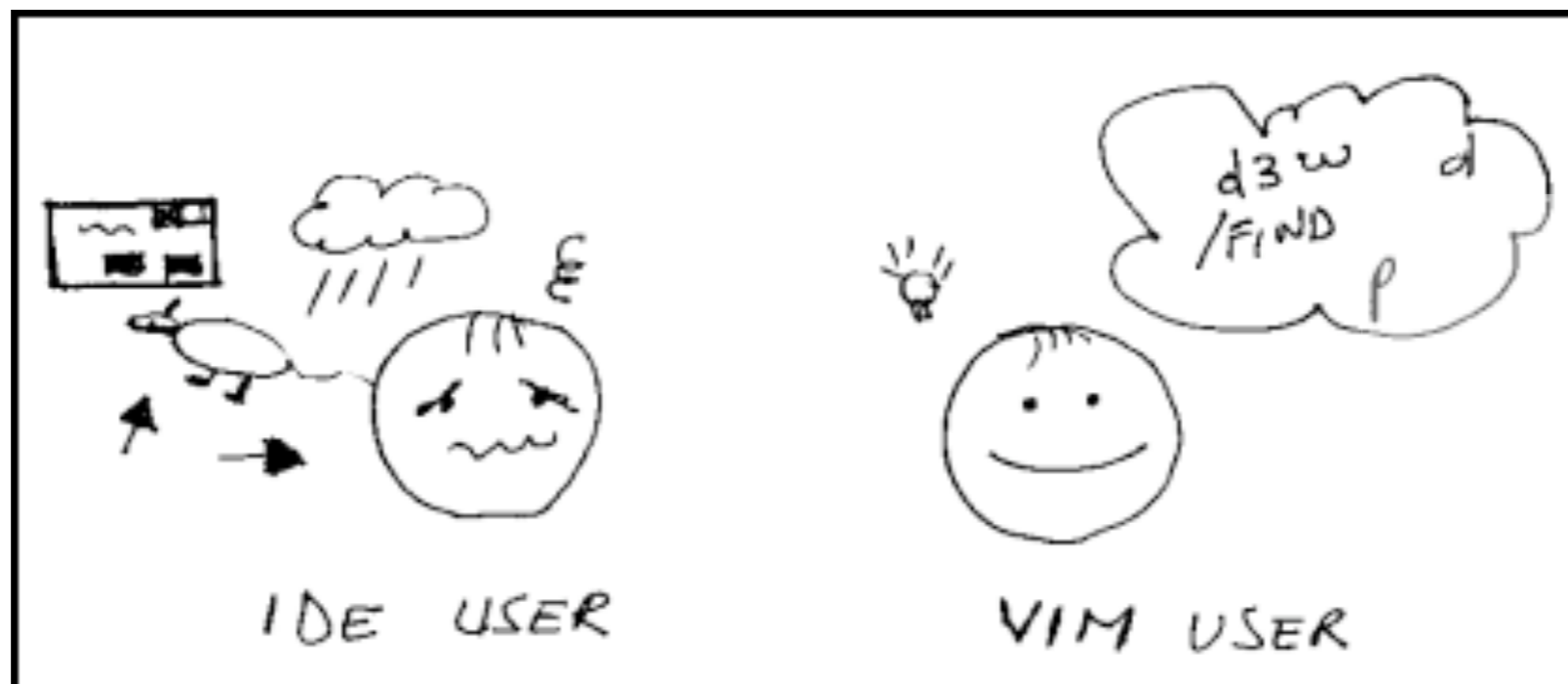
Funcionalidades ¿exclusivas? de VIM

- ▶ Modos
 - Inserción
 - Visual
 - Línea de comando, etc.
- ▶ Bloques
 - Operaciones de edición en bloque (también en Textmate, ¿otros?)
- ▶ Múltiples clipboards (en VIM, registros)
- ▶ Macros
 - Potente grabador de operaciones
- ▶ Etc.



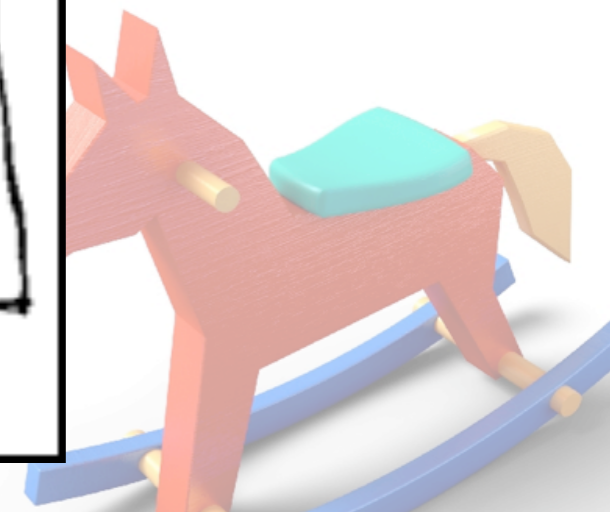
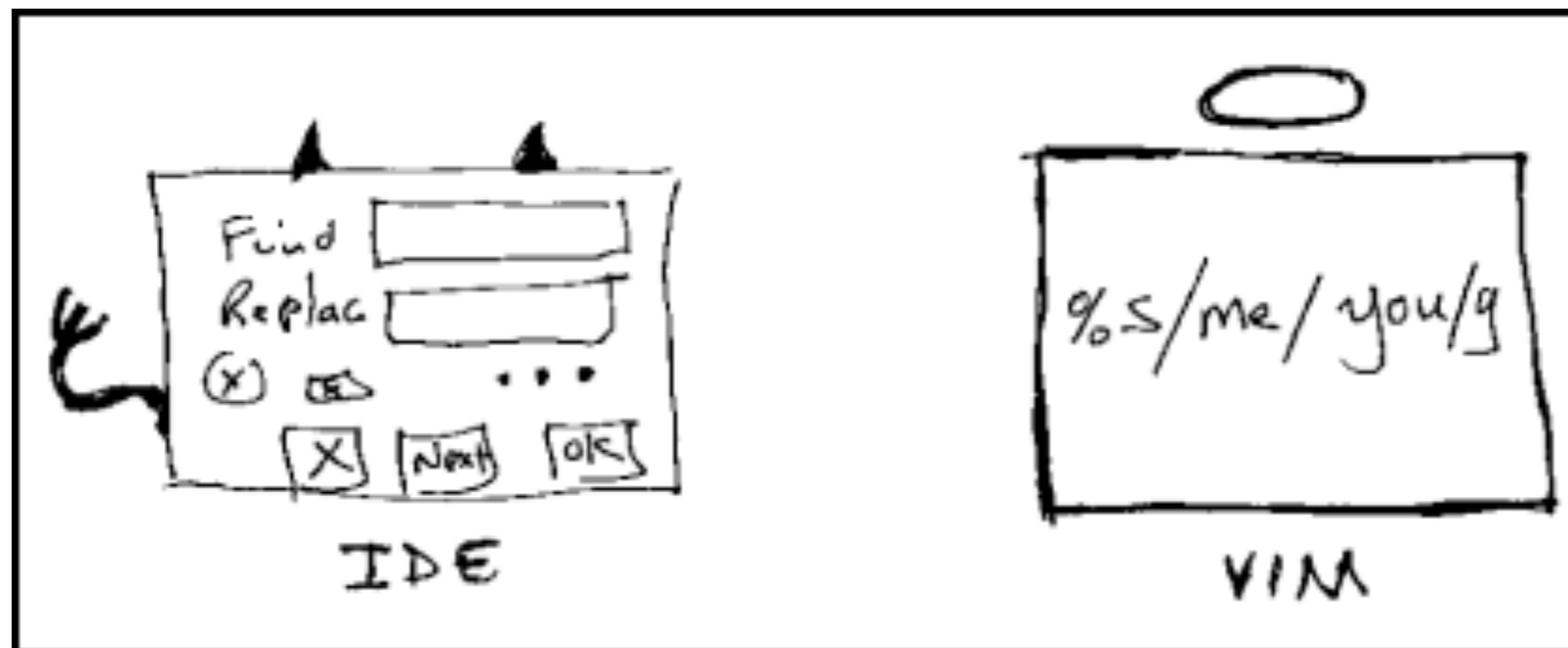
Ventajas con respecto a otros IDEs

- ▶ VIM entiende la **sintaxis de decenas de lenguajes de programación**
 - No tendremos que cambiar de IDE con cada cambio de lenguaje
- ▶ **No** necesitamos usar continuamente el **ratón** para operar con VIM
 - Por tanto, +velocidad tanto en edición como en navegación de código
- ▶ Con VIM podemos **controlar** la edición hasta el más mínimo detalle
 - Ve a la línea 4, busca la primera comilla, elimina el contenido, vuelve al principio del documento: `4Gf"vf"hdgg`
- ▶ El límite está en cuántos comandos somos **capaces de recordar**, no en VIM



Ventajas con respecto a otros IDEs

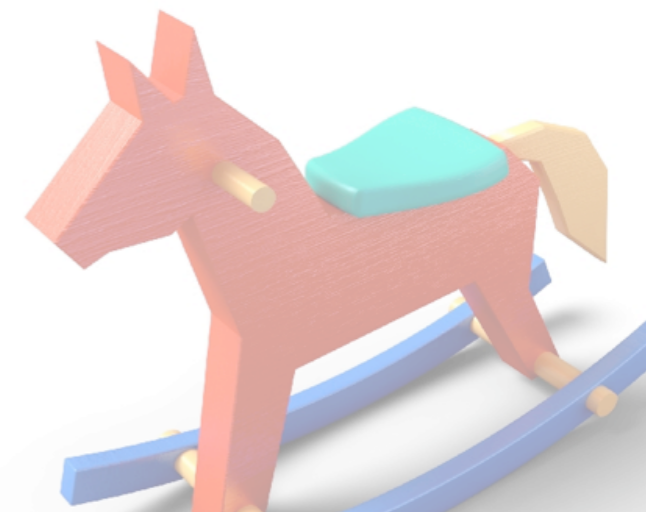
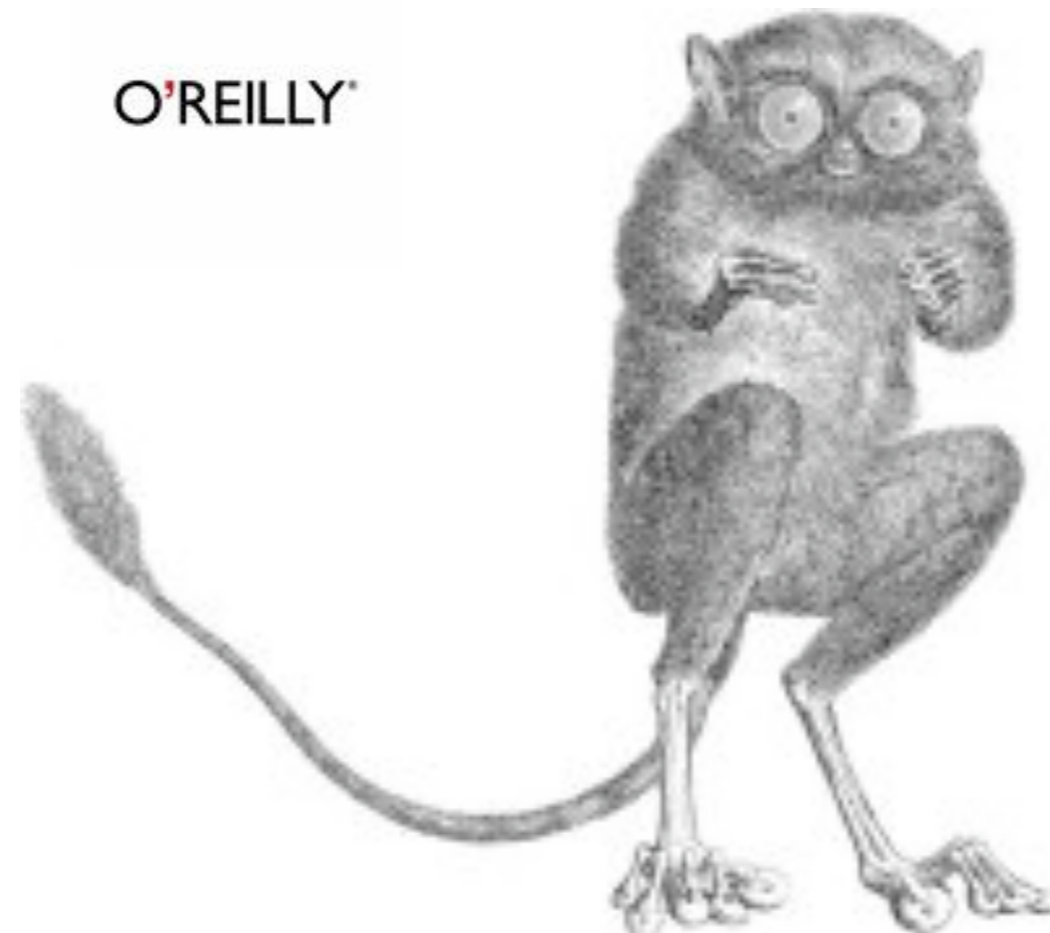
- ▶ Altamente **configurable**:
 - podemos configurar TODOS los atajos de teclado y crear nuevos
 - podemos crear funciones propias que operen con el código (sí, Texmate y otros IDEs también)
- ▶ Podemos **extender** VIM con **plugins** en múltiples lenguajes de programación
 - Escogemos por tanto el más familiar para nosotros
- ▶ En cualquier máquina *nix viene **instalado por defecto**
 - por tanto, **UBICUIDAD**
- ▶ Potente sistema de **búsqueda/reemplazado**



Inconvenientes con respecto a otros IDEs

► Un único inconveniente, **pero muy gordo, muy gordo:**

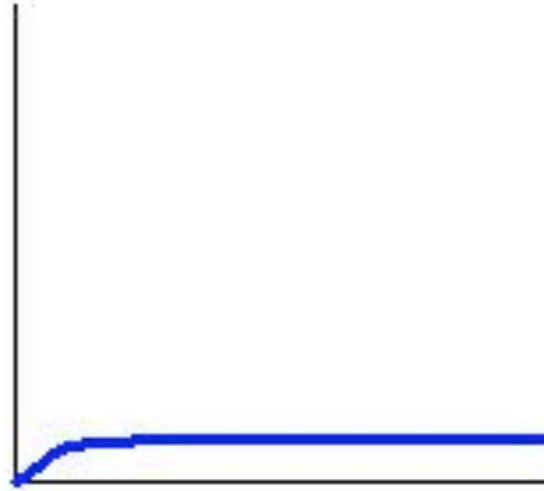
GRAN CURVA DE APRENDIZAJE



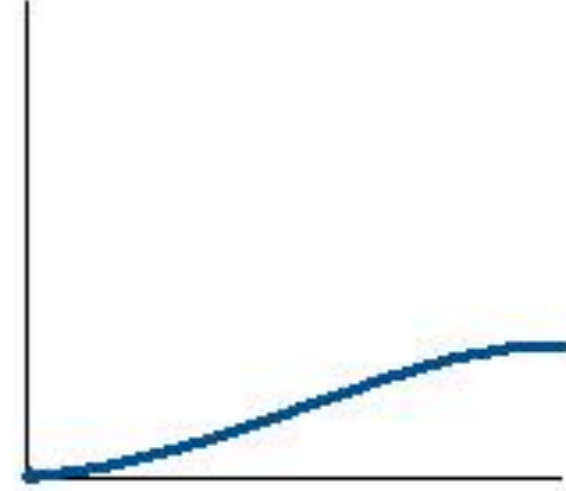
php | VIM, el IDE

Classical learning curves for some common editors

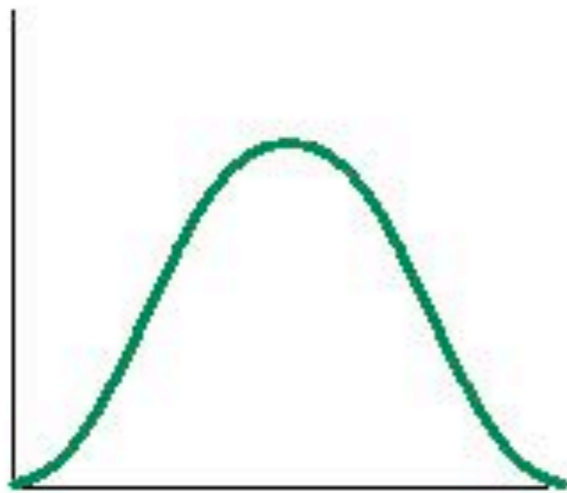
Notepad



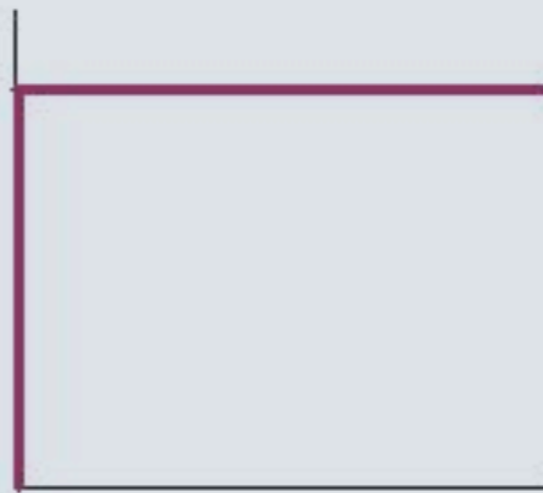
Pico



Visual Studio



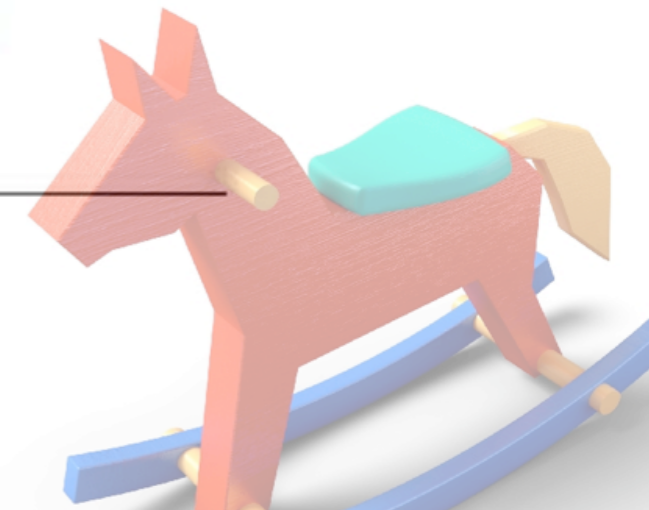
vi



emacs



⌘
11-17-01

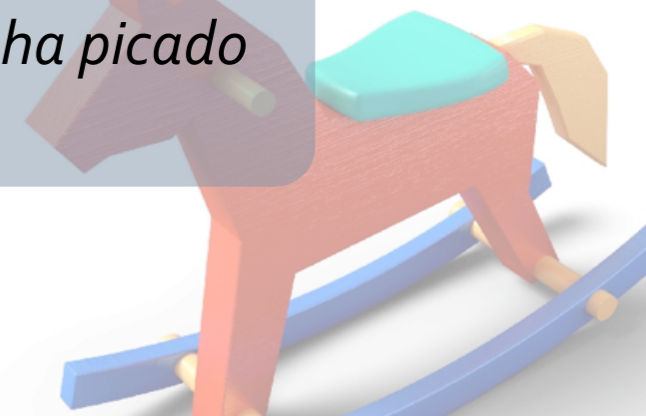




nuestra primera sesión en VIM

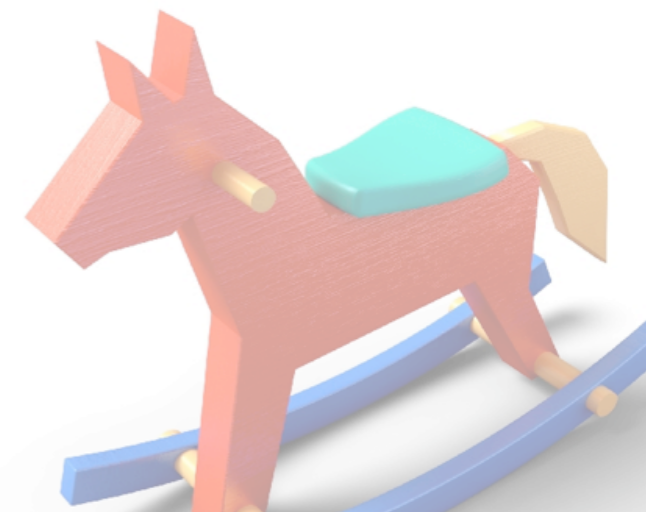
```
^C^C^X^X^X^XquitqQ!qjodeer[esc]qwertyuiopasdfghjkl;:me-voy-al-mouse;:nunca-mais-vim
```

yo uso eclipse, y me he mirado el Vim... y por de pronto se me ha caído el alma a los pies!!! buf... la verdad, no dudo que sea bueno (por vuestros comentarios), pero realmente se necesita un master!!! aún así, me ha picado la curiosidad y a partir de los posts probaré que tal...



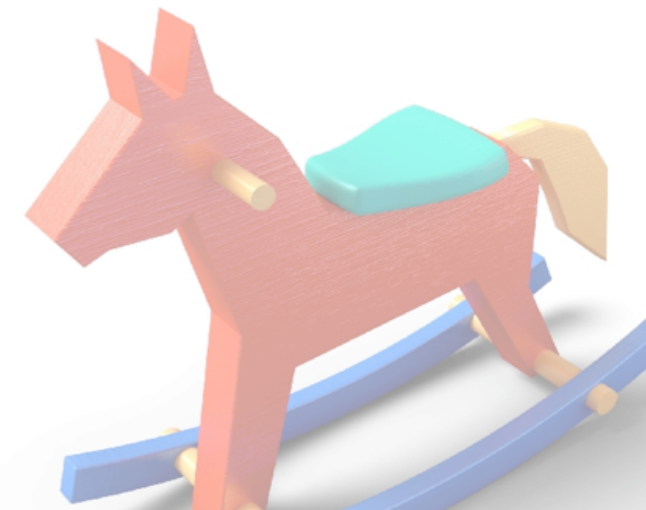
VIM y PHP

- ▶ Algunas funcionalidades vienen por defecto, otras las obtenemos instalando extensiones:
 - coloreado de sintaxis (por defecto)
 - autocompletado (exuberant ctags)
 - navegación por clases. *gf*, o *ctrl+click* te lleva a la definición de la función debajo del cursor
 - check de errores al guardar (script)
 - xdebug (potente debugger para PHP)
 - etc.



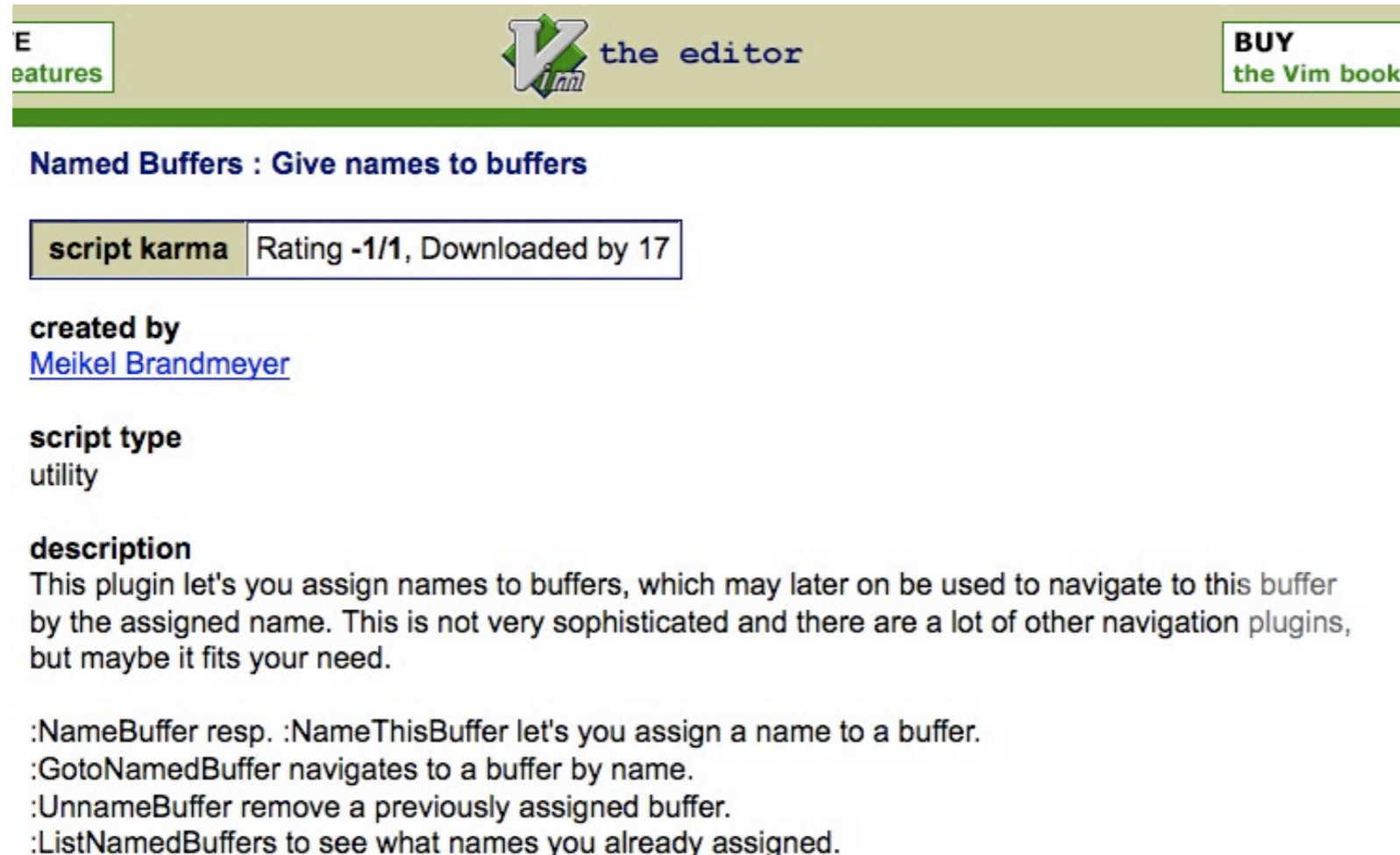
VIM se integra con otros procesos


- ▶ Vim se integra con un montón de herramientas externas:
 - exuberant ctags
 - control de versiones (CVS, subversion, etc.)
 - bases de datos (MySQL, PostgreSQL, etc.)
 - blogs (gestión de posts de wordpress)
 - etc.



VIM es ampliable

- ▶ Existen **más de 2000 plugins** para VIM, pero puede ser que necesites algo muy concreto
- ▶ **Extiende VIM.**
 - Reprograma todas las teclas, macros, a tus necesidades
 - Crea tus propios plugins, varios lenguajes de programación



Eatures  the editor **BUY** the Vim book

Named Buffers : Give names to buffers

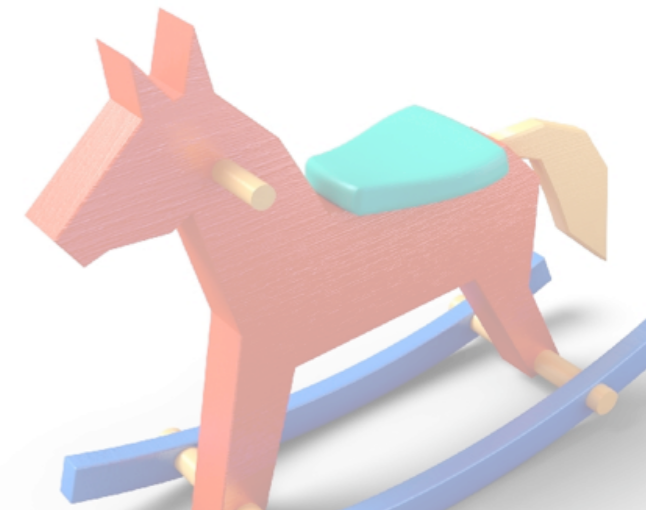
script karma Rating -1/1, Downloaded by 17

created by
[Meikel Brandmeyer](#)

script type
utility

description
This plugin let's you assign names to buffers, which may later on be used to navigate to this buffer by the assigned name. This is not very sophisticated and there are a lot of other navigation plugins, but maybe it fits your need.

:NameBuffer resp. :NameThisBuffer let's you assign a name to a buffer.
:GotoNamedBuffer navigates to a buffer by name.
:UnnameBuffer remove a previously assigned buffer.
:ListNamedBuffers to see what names you already assigned.

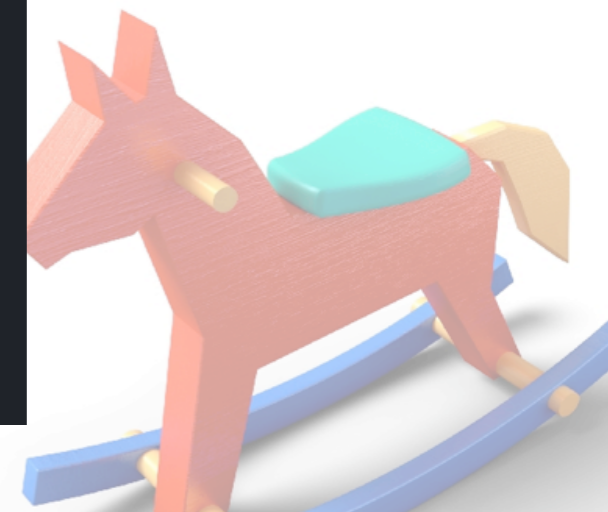


VIM es ampliable

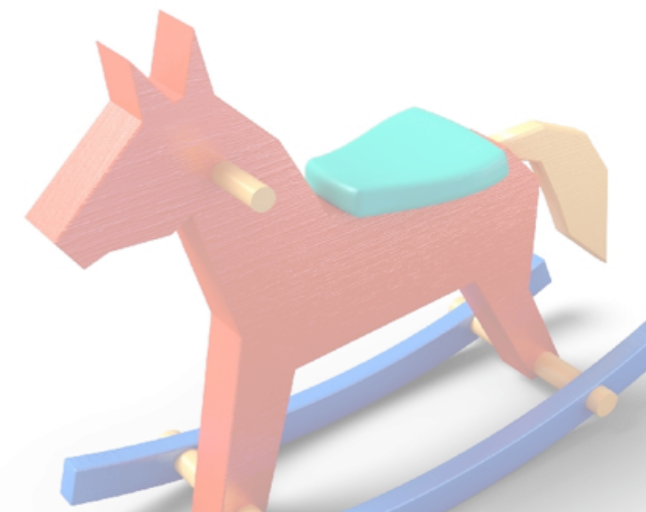
- Ejemplo, hemos adaptado un plugin a la idiosincrasia de nuestro trabajo: el engendro se llama *FindMate*
 - Lo invocas (*,,*), tecleas partes de un nombre de fichero y FindMate te muestra las correspondencias de entre los ficheros de tu proyecto.
 - Eliges uno/varios del listado de ficheros, y los abre todos en varias ventanas horizontales
 - No te pases el día parseando con la vista directorios en busca del fichero que necesitas: **deja que el ordenador lo haga por ti**

:Findmate edit|php

```
1 ./app/engine/controllers/dir/edit.php
2 ./app/engine/controllers/file/edit.php
3 ./app/engine/controllers/organization/edit.php
4 ./app/engine/controllers/resource/edit_entry.php
5 ./app/engine/controllers/space/edit.php
6 ./app/engine/controllers/task/deliverable-edit.php
7 ./app/engine/controllers/task/edit.php
8 ./app/engine/controllers/user/edit.php
Which ? (<enter>=nothing)
```



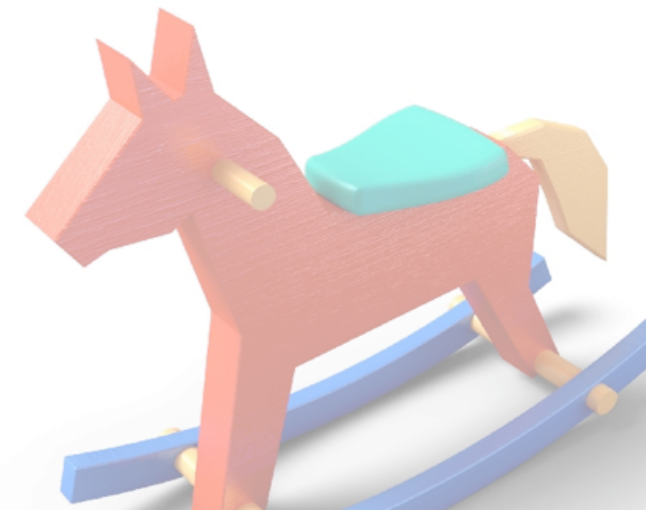
Pasemos a ver VIM en detalle...



Modos

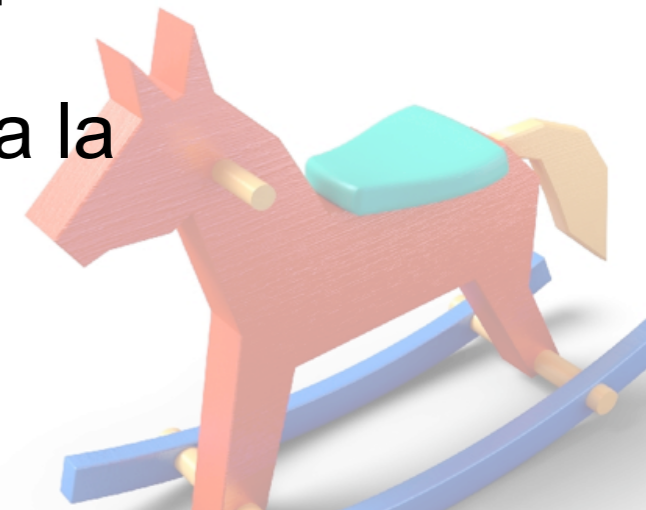
VIM dispone de varios modos de trabajo:

- 1. Comandos:** Es el modo donde se pueden emplear combinaciones de teclas para, por ejemplo, copiar líneas y trabajar en el formato del texto.
- 2. Inserción:** En modo inserción cuando se pulsan las teclas se edita el texto como en otros editores.
- 3. Línea de comandos:** A este modo se accede pulsando la tecla dos puntos :. Las búsquedas se pueden realizar con el comando / (hacia adelante) y ? (hacia atrás). También se pueden filtrar líneas mediante !.



Modos

- 1. Visual:** Mediante unas ciertas combinaciones de teclas en combinación con las teclas de movimiento del cursor, se puede marcar un área de texto, ya sea un grupo de líneas o un bloque. Una vez se tiene el texto marcado se pueden usar comandos del modo comandos para manipularlo.
- 2. Ex:** Este modo se asemeja al modo línea de comandos, con la diferencia de que tras la ejecución de un comando no se vuelve al modo comandos. Se entra en este modo pulsando **Q** y se termina con **vi**. En este modo Vim imita al editor de [UNIX](#) ex, que manipulaba el texto línea a línea debido a las limitaciones de la época, en lugar de editar toda la página.

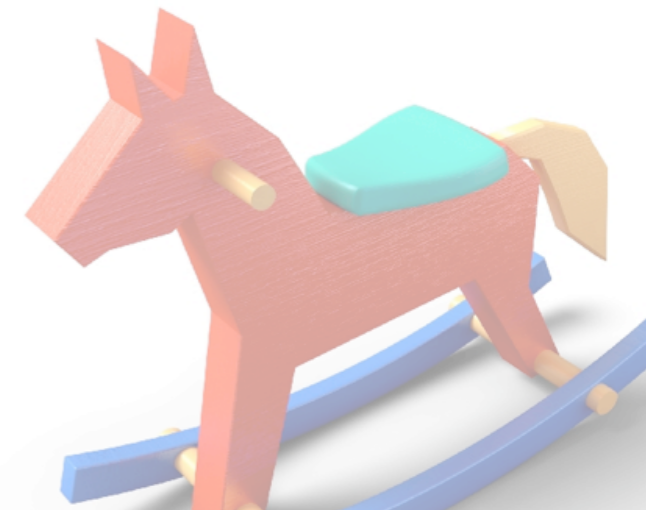


.vimrc

Como casi todas las aplicaciones *nix VIM tiene su archivo de configuración, .vimrc

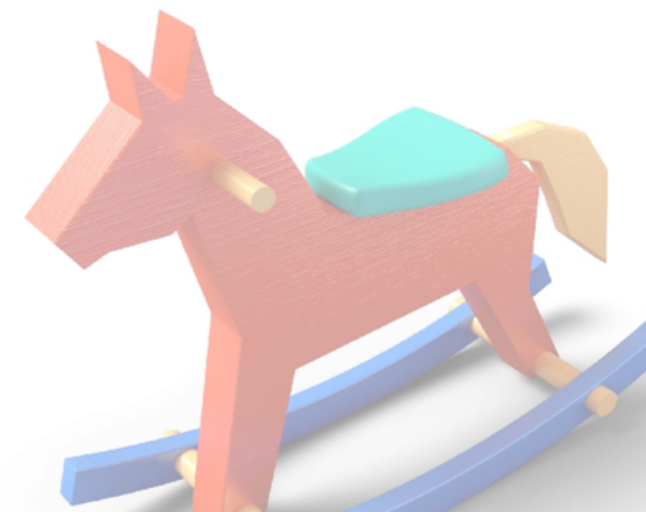
En el podemos configurar aquellos parametros que deseamos cambiar de Vim para trabajar mejor, como la tabulación, el resaltado de búsquedas, la tipografía a usar, el tema de colores.

También, como veremos en los plugins, se pone todas aquellas variables que se necesiten para un plugin.



Edición avanzada

- ▶ Navegación
- ▶ Edición
- ▶ Bloques
- ▶ Macros
- ▶ Buscar / Reemplazar



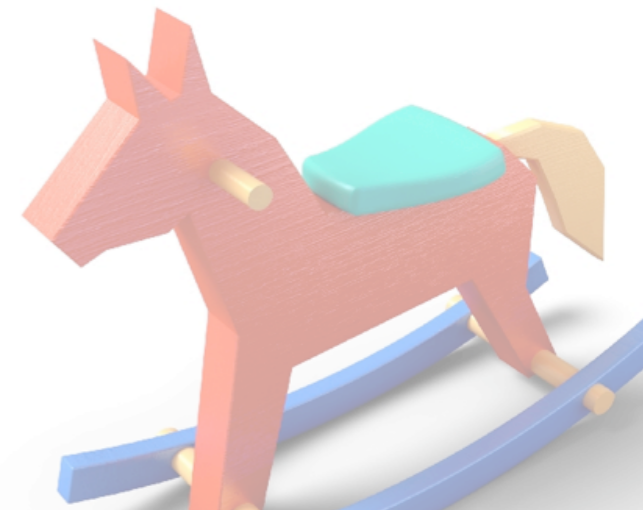
Navegación

Los principales movimientos se pueden hacer con las teclas:

- **k** para ir arriba
- **j** para ir abajo
- **h** para ir izquierda
- **l** para ir derecha

También podemos saltar directamente a una línea, hay varias formas pero la más útil es:

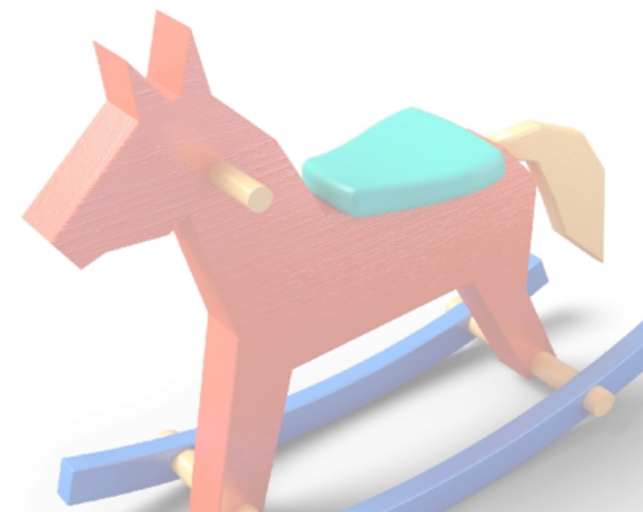
- **<ESC>:81<CR>**



Navegación

Otros comandos:

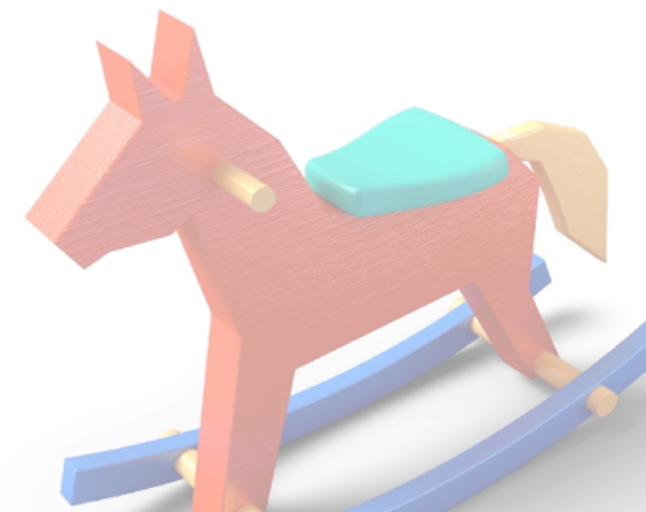
- **E** o **e** Mover al final de la palabra
- **B** o **b** Mover al principio de una palabra
- **O** Inicio de línea
- **^** Mover al primer carácter no-blanco de la línea
- **\$** Mover al final de la línea
- **H** Mover a la primera línea en pantalla
- **M** Mover a la línea del medio de la pantalla
- **L** Mover a la última línea de la pantalla



Edición

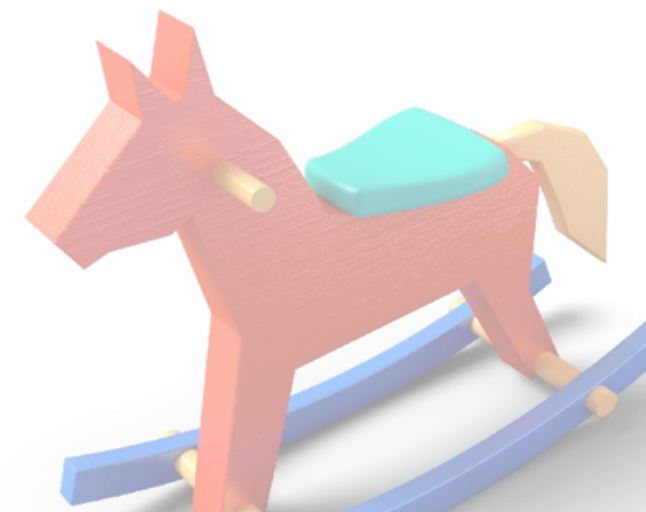
Para empezar la edición tenemos varias letras:

- **i** o **I** para insertar, si es mayúscula insertará al inicio
- **a** o **A** para añadir, si es mayúscula lo hará al final
- **o** o **O** para añadir un salto de línea y empezar a escribir, si es mayúscula el salto de línea lo hará por arriba
- **y** o **Y** para copiar, mayúscula una línea entera
- **p** o **P** para pegar, mayúscula pega
- **dd** borra la línea
- **df** + un carácter borra desde el cursor hasta el carácter
- **dw** borra la palabra



Edición

- **D** borra desde la posición hasta el final
- **u** deshacer
- **^R** re-hacer
- **x** o **X** para borrar el carácter, mayúscula para borrar el de atrás
- **r** o **R** remplazar un carácter, mayúscula varios caracteres
- **<esc> :r [name]** inserta el contenido del archivo
- **<esc> :r !{cmd}** inserta el retorno del comando



Bloques

Para entrar en el modo visual hay que apretar la **v** en el modo comando. Y después nos movemos para seleccionar.

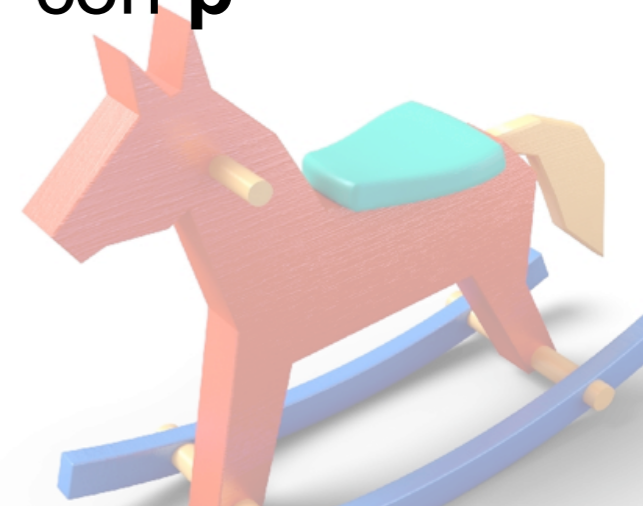
Pero si hacemos **<CTRL> v** entraremos en el modo visual pero con selección por bloque.

Una vez seleccionado el bloque podemos.

x borra el bloque

i insertar al principio de cada línea del bloque

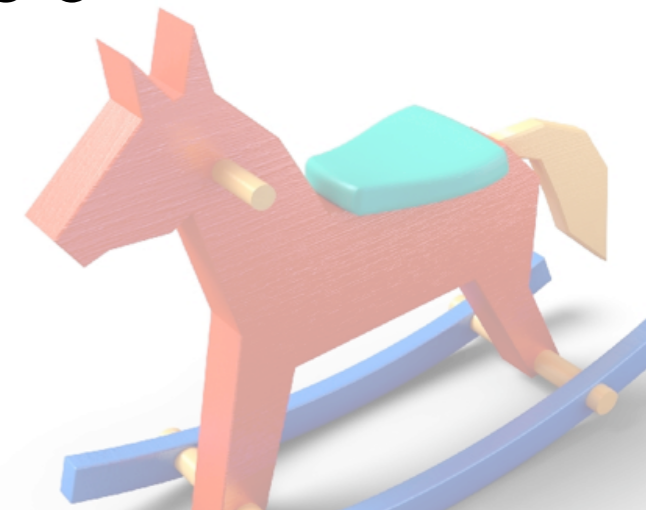
y copiamos el bloque que después podremos pegar con **p**



Macros

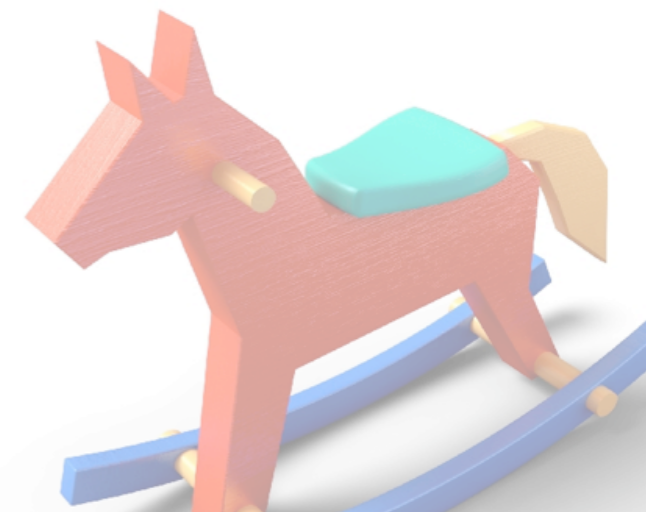
Los macros son series de actuaciones que las has de repetir varias veces, para crearlas sólo hay que seguir este patrón:

1. Aprietas la **q** y seguido de la letra que tomara como nombre.
2. Realizas todas las acciones que quieres grabar para después repetir.
3. Una vez finalizado sales al modo comando con **<ESC>** y después la **q** para para parar la grabación.
4. para ejecutar la macro sólo debes teclear en modo comando **@+la letra** que le pusimos de nombre o **#@la letra**



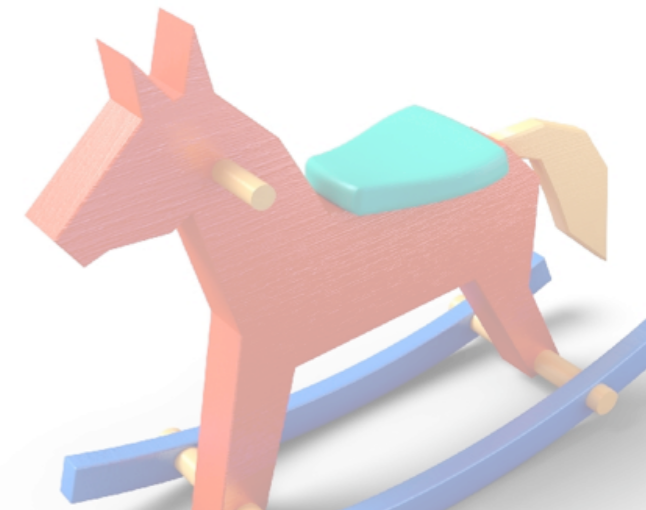
Buscar/reemplazar

- **f** más un carácter busca la siguiente aparición de dicho carácter en la misma línea
- **/** más el texto a buscar, acepta expresiones regulares
- **:s/old/new/g** busca old y lo reemplaza por new en la línea
- **:%s/old/new/g** busca old y lo reemplaza por new en el documento
- **:#,#s/old/new/g** busca old y lo reemplaza por new en el rango de líneas entre # y #



Extensiones (plugins, scripts)

- ▶ FindMate
- ▶ VCS (control de versiones)
- ▶ SQLUtil (formato queries)
- ▶ DBExt (operaciones bbdd)
- ▶ NerdCommenter
- ▶ NerdTree (directorios)
- ▶ PHP Check Syntax
- ▶ XDebug
- ▶ Tags

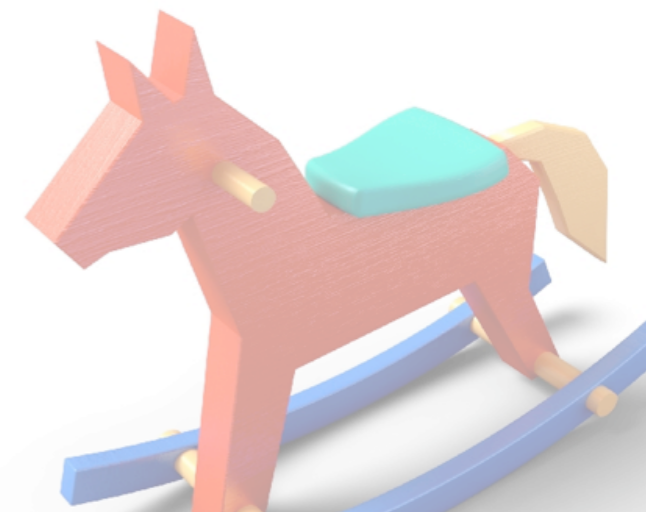


plugins: Findmate

FindMate es un buscador de archivos dentro del árbol de directorio.

- *:FindMate nombre*
- *:FindMate nombre+*
- *:FindMate nombre|texto*

Nosotros tenemos mapeadas la combinación de teclas ,, para hacer la invocación de FindMate



plugins: VCS

http://www.vim.org/scripts/script.php?script_id=90

Es un plugin para gestionar las versiones de nuestro código, con este plugin podemos conectarnos a CVS,SVN,GIT y SVK

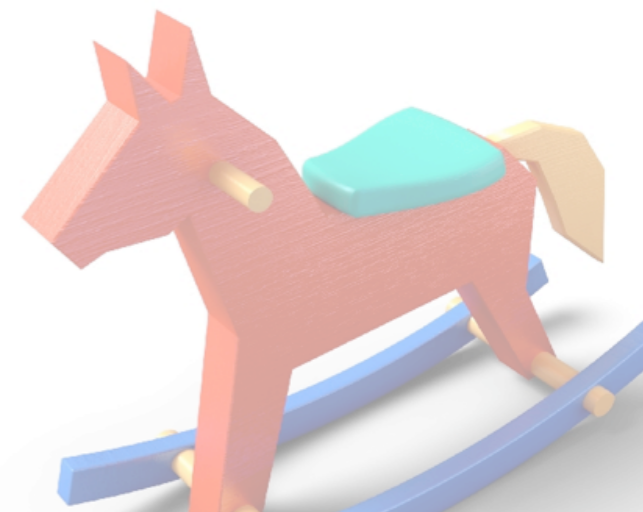
Sólo nos hará falta aprendernos los comandos de un plugin y gestionamos varios repositorios.

Algunos de ellos:

:VCSAdd

:VCSCommit

:VCSDiff



plugins: SQLUtil

http://www.vim.org/scripts/script.php?script_id=492

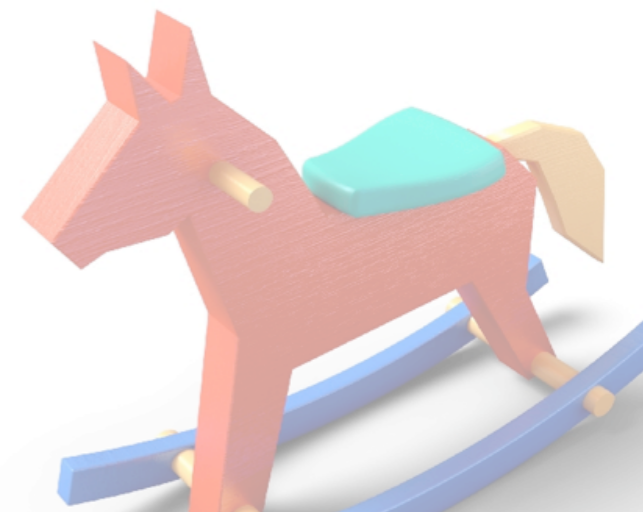
http://www.vim.org/script.php?script_id=294

Principalmente nos sirve para limpiarle la cara a nuestras queries.

También puede generar la lista de columnas de una tabla si en cualquiera de los buffers tenemos el CREATE TABLE.

Para ejecutarlo seleccionaremos la query y después:

:SQLUFormatter



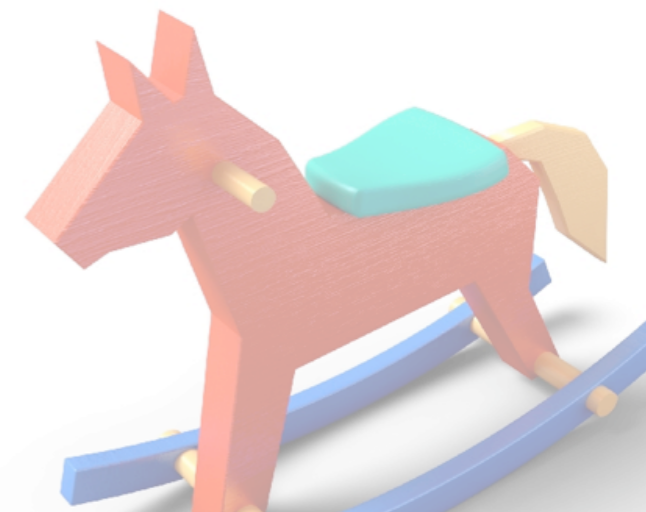
plugins: DBExt

http://vim.sourceforge.net/script.php?script_id=356

Con este plugin podremos conectarnos a varias BBDD y hacer consultas.

La configuración de la conexión se puede hacer al momento o tener un pool de conexiones en nuestro .vimrc

```
let g:dbext_default_profile_local =  
'type=MYSQL:user=mysql:passwd=:dbname=mysql:extra=-t'
```

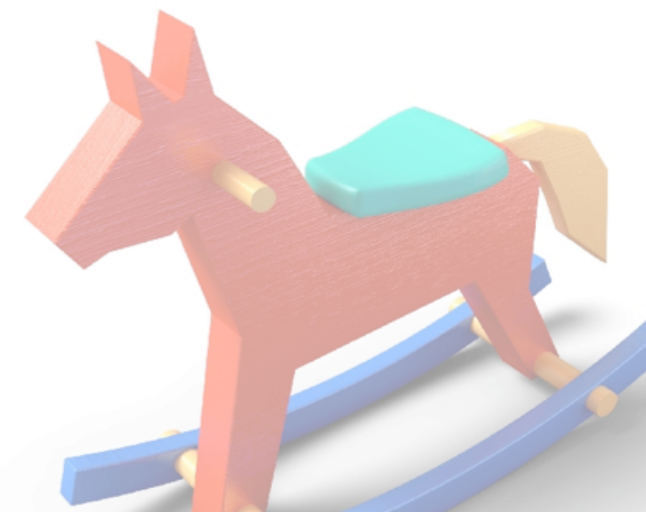


plugins: NerdCommenter

http://www.vim.org/scripts/script.php?script_id=1218

Con NerdCommenter podremos comentar líneas o bloques de código con pocas teclas, pondrá los comentarios según el tipo de archivo que estemos editando.

Si seleccionas un bloque de texto y tecleas **,cc** te comentará cada línea por separado, para cambiar el estado de la línea con **,c<space>** y para comentar un bloque de forma más sexy **,cs**



plugins: NerdTree

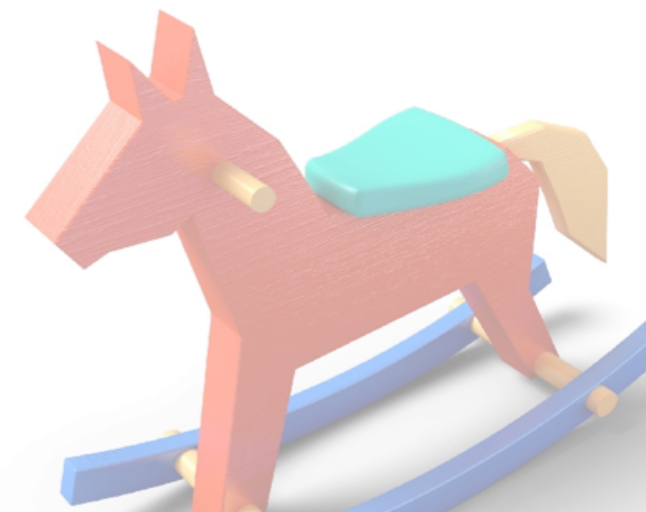
http://vim.sourceforge.net/scripts/script.php?script_id=1658

NerdTree es un navegador de archivos en formato de árbol.

- `:NERDTree` abre el árbol
- `:NERDTreeClose` cierra el árbol
- `:NERDTreeToggle` abre/cierra el árbol

Aunque lo mejor es mapear una tecla para que ejecute **NERDTreeToggle** y esto lo podemos hacer añadiendo la siguiente línea en nuestro `.vimrc`

```
map <F2> :NERDTreeToggle<cr>
```



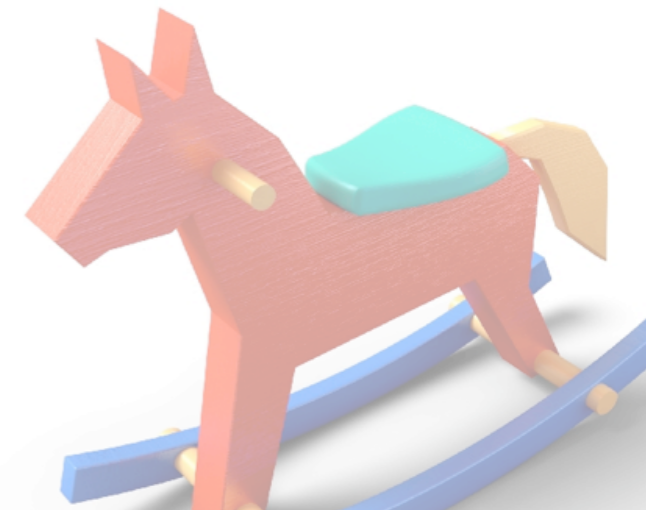
plugins: Check Syntax

http://vim.sourceforge.net/scripts/script.php?script_id=1272

Cada vez que guardemos un archivo php o cliquemos la tecla **F5** se ejecutará la comprobación de la sintaxis PHP, indicándonos los errores si los hubiera.



¡¡Altamente recomendable!!



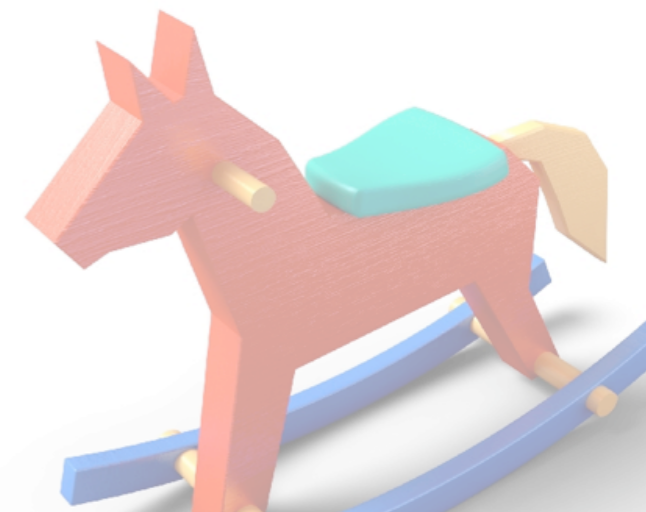
plugins: XDebug

http://www.vim.org/scripts/script.php?script_id=2218

http://www.vim.org/scripts/script.php?script_id=1152

Este plugin nos permitirá conectarnos al modulo [XDebug](#) de [PHP](#) , podremos poner Breakpoints, ver el contenido de las variables en un momento dado.

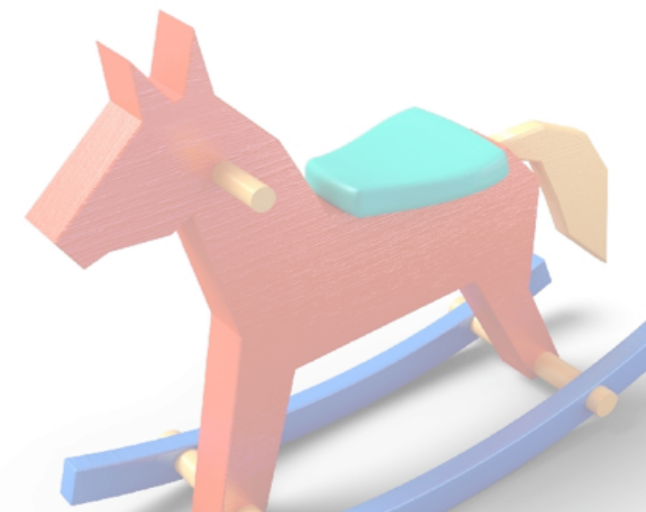
Configuración en el php: `xdebug.remote_host=127.0.0.1`



plugins: Tags

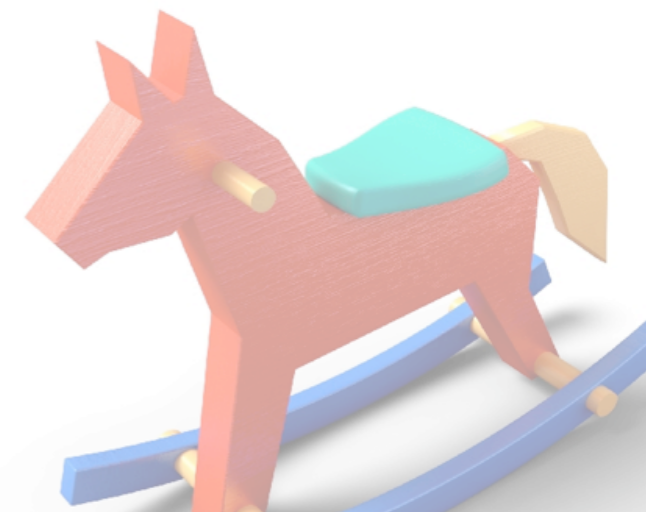
http://www.vim.org/scripts/script.php?script_id=273

El plugin de tags unido con el programa **exuberant-ctags** nos permitirán saltar al código de un método o clase con sólo clickar sobre él mientras apretamos el **<CTRL>**



Algunas URLs de interés sobre VIM

- <http://www.vim.org/>
- <http://vim.wikia.com/>
- <http://gpl.internetconnection.net/vi/>
- http://h1.ripway.com/chube/Manual_Vi/Indice_vi.htm
- <http://www.moolenaar.net/vim.html>
- <http://vim.sourceforge.net/>
- <http://www.viemu.com/a-why-vi-vim.html>
- <http://xhtml-css.com/blog/seven-reasons-every-programmer-should-use-vim>



¿Quieres ser un...



Very Improved Man

¡a practicar!... pero recuerda

“Un gran poder requiere de una gran responsabilidad”

Spiderman



¡Gracias!

Esta presentación y más cosas las tienes en:

<http://www.phpbarcelona.org>

