
Diseño y despliegue de un clúster de bajo presupuesto para el desarrollo de las prácticas de PSD



TRABAJO FIN DE GRADO
Ingeniería de Computadores

Dirigido por

Alberto Núñez Covarrubias

Codirigido por

Luis Llana Díaz

Daniel Quiñones Sánchez
Miguel Romero Martínez

Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid

Junio 2018

Diseño y despliegue de un clúster de bajo presupuesto para el desarrollo de las prácticas de PSD

Memoria de Trabajo de Fin de Grado

**Daniel Quiñones Sánchez
Miguel Romero Martínez**

Dirigido por

Alberto Núñez Covarrubias

**Departamento de Sistemas Informáticos y Computación
Facultad de Informática
Universidad Complutense de Madrid**

Junio 2018

*A mi compañero y amigo Rome, por aguantarme en todo momento y
realizar la mayor parte del trabajo.*

*A mi compañero y amigo Daniel, por aguantarme en todo momento y
realizar la mayor parte del trabajo.*

Agradecimientos

Resumen

Palabras clave

Siglas

Abstract

Keywords

Índice

Agradecimientos	IX
Resumen	X
Palabras clave	XI
Siglas	XII
Abstract	XIII
Keywords	XIV
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura	1
1.4. Motivation	1
1.5. Goals	1
	XV

1.6. Structure	1
2. Trabajo relacionado	2
2.1. Nuestra aportación	2
3. Arquitectura del Cluster	3
3.1. Componentes	4
3.2. Diseño del prototipo A	4
3.2.1. Modelado 3D	4
3.2.2. Resultados (fotos)	4
3.3. Diseño del prototipo B	4
3.3.1. Modelado 3D	4
3.3.2. Resultados (fotos)	4
3.4. Diseño del prototipo C	4
3.4.1. Modelado 3D	4
3.4.2. Resultados (fotos)	4
3.5. Evaluación de prototipos A,B,C	4
3.5.1. Rendimiento prototipos A,B,C	4
3.5.2. Temperatura prototipos A,B,C	4
3.6. Problemas	4
3.6.1. Falta de energía	4
4. Configuración del cluster	5

4.1. Virtualización del Sistema	5
4.2. Montaje de servidores	8
4.3. Dynamic Host Configuration Protocol (DHCP)	9
4.3.1. Guía paso a paso de instalación de DHCP	10
4.4. Network File System (NFS)	11
4.4.1. Guía paso a paso de instalación de NFS	12
4.4.2. Creación y lanzamiento de servidor NFS como un dae- mon del sistema	13
4.4.3. SSH	14
4.5. Instalación de SIMCAN	14
4.5.1. Guía paso a paso de instalación en arquitectura ARM	15
4.6. Optimización y rendimiento	15
4.6.1. Modificación del GRUB	15
4.6.2. Paralelización del arranque en servidor	15
4.7. Seguridad	15
4.7.1. Eliminación de usuarios y permisos	15
4.8. Inicialización del sistema mediante scripts	15
 5. Desarrollo del software	 16
 6. Pruebas	 17
6.1. Comparación con otros procesadores de mayor potencia	17

6.2. Estudio a partir de qué nivel de uso del cluster merece la pena ante un procesador normal	17
6.3. Pruebas contra el cluster de Amazon(???)	17
7. Trabajo individual	18
7.1. Miguel Romero Martínez:	18
7.2. Daniel Quiñones Sánchez:	18
8. Conclusiones y trabajo futuro	19
8.1. Conclusiones generales	19
8.2. Trabajo futuro	19
8.3. General conclusions:	19
8.4. Future work:	19
Bibliografía	21

Índice de figuras

4.1. Estructura en árbol	8
4.2. Montaje de servidores	8
4.3. Estructura DHCP	9
5.1. Raspberry, te das cuen	16

Índice de Tablas

Capítulo 1

Introducción

"long long long is too long for GCC"
some programmer from GCC project

1.1. Motivación

1.2. Objetivos

1.3. Estructura

1.4. Motivation

1.5. Goals

1.6. Structure

Capítulo 2

Trabajo relacionado

*Me gusta y me fascina el trabajo. Podría estar sentado
horas y horas mirando a otros cómo trabajan.*

Jerome K. Jerome, Humorista inglés.

2.1. Nuestra aportación

Capítulo 3

Arquitectura del Cluster

3.1. Componentes

3.2. Diseño del prototipo A

3.2.1. Modelado 3D

3.2.2. Resultados (fotos)

3.3. Diseño del prototipo B

3.3.1. Modelado 3D

3.3.2. Resultados (fotos)

3.4. Diseño del prototipo C

3.4.1. Modelado 3D

3.4.2. Resultados (fotos)

3.5. Evaluación de prototipos A,B,C

Capítulo 4

Configuración del cluster

4.1. Virtualización del Sistema

Debido al volumen del material necesario para el desarrollo del proyecto, la necesidad de realizar el montaje y desmontaje de forma manual de éste y la dificultad en el acceso y configuración de cada uno de los nodos que lo componen decidimos invertir tiempo en realizar una virtualización del sistema para minimizar los problemas antes descritos. Así, al disponer de un entorno virtual, que replica el cluster real, se minimiza el tiempo necesario para la configuración de los distintos servicios y servidores del sistema operativo y sirve como banco de pruebas para el desarrollo software.

```
1 sudo apt-get install pollas
2 sudo apt-get update
```

```
3 sudo apt-get update
4 sudo apt-get install isc-dhcp-server
```

SERVIDOR: editar fichero isc-dchp-server

```
5 sudo nano /etc/default/isc-dhcp-server
INTERFACES = eth0
```

SERVIDOR: editar fichero interfaces.d

```
nano /etc/network/interfaces
source-directory /etc/network/interfaces.d

auto eth0
iface eth0 inet static
address 172.16.111.106
netmask 255.255.255.0
network 192.168.100.0
gateway 172.16.111.105

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

SERVIDOR: Editar fichero de configuración /etc/dhcp/dhcpd.conf

Incluir la siguiente configuración al final del fichero:

```
subnet 172.16.111.0 netmask 255.255.255.0{
    range 172.16.111.106 172.16.111.255;
    option domain-name-servers 8.8.8.8, 4.4.4.4;
    default-lease-time 7776000;
    option routers 172.16.111.105;
}

host Rick{
    hardware ethernet <dir MAC del servidor>
    fixed-address 172.16.111.110;
}
```

SERVIDOR: Activación de la red eth0

```
sudo ifconfig eth0 up
```

SERVIDOR: reiniciar

```
sudo reboot now
```

SERVIDOR: Arranque del servicio dhcp

```
sudo /usr/sbin/dhcpd
```

SERVIDOR: Comprobación del correcto funcionamiento del sistema

```
ps -ef | grep dhcpd
```

SERVIDOR: Muestra de las maquinas conectadas al servicio

```
cat /var/lib/dhcp/dhcp.leases
```

Para ello hemos utilizamos VMware workstation 12 como plataforma de software de virtualización y una ISO de Raspbian basada en Debian Stretch disponible en la página oficial de Raspberry Pi. Aunque el sistema del cluster parte de una versión diferente de debian el sistema de carpetas y sobre todo la instalación de SIMCAN es similar al del entorno real.

En el repositorio en github existe una réplica del sistema de carpetas tanto para el servidor como para los nodos slave. Cada una de las carpetas y ficheros modificados en el sistema durante la configuración del sistema quedan reflejados en el repositorio, disponiendo así de un listado en forma de árbol de todas las configuraciones necesarias para el correcto funcionamiento de este.

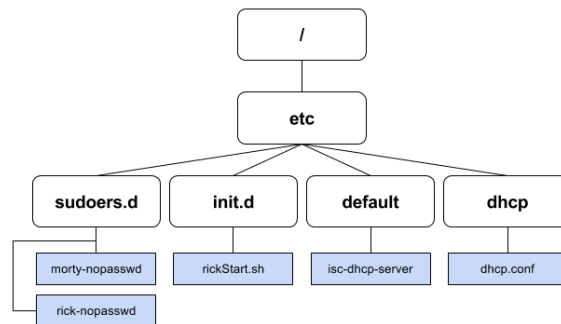


Figura 4.1: Estructura en árbol

4.2. Montaje de servidores

El cluster está dividido en dos partes bien diferenciadas, por un lado, el front-end, donde se dispone de un único servidor, el cual es el único nodo del cluster que tiene instalado el software de SIMCAN además de ofrecer los servicios de NFS, DHCP y SSH entre otros. Éste además es el punto de comunicación con el exterior, al disponer de una tarjeta de red adicional.

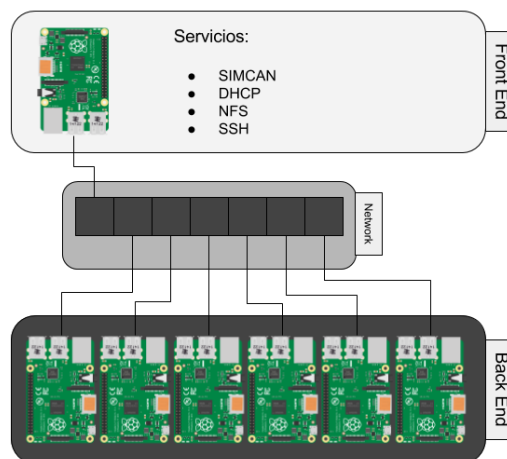


Figura 4.2: Montaje de servidores

Por otro lado, en el back-end, disponemos de varios nodos slave que disponen únicamente del sistema Debian Jessie y unas pocas librerías necesarias para realizar las operaciones de computo.

4.3. Dynamic Host Configuration Protocol (DHCP)

El nodo master hará de servidor repartiendo las direcciones de forma dinámica al resto de clientes. La red para del back-end es la 172.16.111.0/24, el nodo master tiene asignada de forma manual la dirección 172.16.111.1/24 y hace un reparto con el resto de direcciones del rango.

El montaje de DHCP evita tener que asignar manualmente direcciones a todos los nodos, sin embargo es más útil en la práctica que cada uno de los nodos disponga de la misma dirección el máximo tiempo posible, para ello se han modificado el parámetro `default-lease-time`, que determina el tiempo de concesión de una IP cuando el cliente en su solicitud no lo especifica, del fichero `/etc/dhcp/dhcp.conf` a sus máximos valores posibles. Estableciendo un valor de 7776000 segundos en dicho campo obtenemos un tiempo de concesión de noventa días.

Los servicios ofrecidos por el nodo master de cara al front-end están disponibles a través de su segunda tarjeta de red, configurada para obtener una dirección IP desde un ISP.

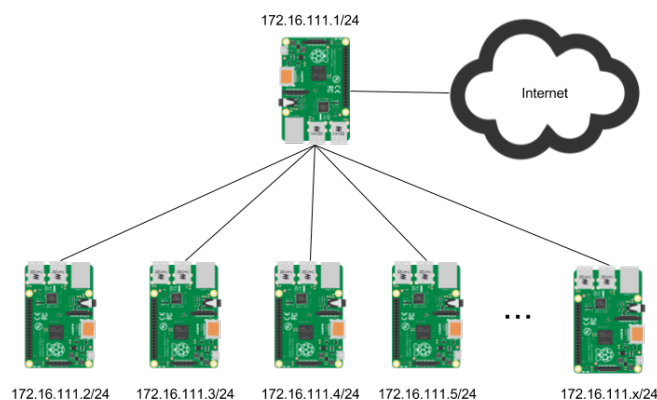


Figura 4.3: Estructura DHCP

4.3.1. Guía paso a paso de instalación de DHCP

SERVIDOR: instalación de paquetes

```
sudo apt-get update
sudo apt-get install isc-dhcp-server
```

SERVIDOR: editar fichero `isc-dhcp-server`

```
sudo nano /etc/default/isc-dhcp-server
INTERFACES = eth0
```

SERVIDOR: editar fichero `interfaces.d`

```
nano /etc/network/interfaces
source-directory /etc/network/interfaces.d
```

```
auto eth0
iface eth0 inet static
address 172.16.111.106
netmask 255.255.255.0
network 192.168.100.0
gateway 172.16.111.105
```

```
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

```
allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

SERVIDOR: Editar fichero de configuración `/etc/dhcp/dhcpd.conf`

Incluir la siguiente configuración al final del fichero:

```
subnet 172.16.111.0 netmask 255.255.255.0{
    range 172.16.111.106 172.16.111.255;
```

```
option domain-name-servers 8.8.8.8, 4.4.4.4;  
default-lease-time 7776000;  
option routers 172.16.111.105;  
}
```

```
host Rick{  
    hardware ethernet <dir MAC del servidor>  
    fixed-address 172.16.111.110;  
}
```

SERVIDOR: Activación de la red eth0

```
sudo ifconfig eth0 up
```

SERVIDOR: reiniciar

```
sudo reboot now
```

SERVIDOR: Arranque del servicio dhcp

```
sudo /usr/sbin/dhcpd
```

SERVIDOR: Comprobación del correcto funcionamiento del sistema

```
ps -ef | grep dhcpd
```

SERVIDOR: Muestra de las maquinas conectadas al servicio

```
cat /var/lib/dhcp/dhcp.leases
```

4.4. Network File System (NFS)

Como se destacaba anteriormente, el nodo servidor es el único que dispone de una versión de SIMCAN instalada, esta configuración ofrece la posibilidad de que la labor de los nodos slave sea únicamente la de realizar el procesamiento de datos. Mediante NFS, el nodo servidor comparte su carpeta home durante el arranque del sistema, de esta forma, el resto de nodos slave realizan el montaje de este home compartido en red en su propio direc-

torio home, creando así un único punto de acceso compartido en red del que se pueden extraer los ejecutables sin la necesidad de disponer de SIMCAN instalado. El master se encarga de realizar la compilación los ficheros .ned y pone a disposición del resto de nodos los ejecutables.

Es necesario que todos los nodos de la red tengan un mismo usuario común para conseguir una correcta sincronización, de igual manera hay que mantener un estricto control de los permisos de cada uno de los nodos slave tanto a nivel interno como de cara al servidor.

4.4.1. Guía paso a paso de instalación de NFS

SERVIDOR: Instalar paquetes

```
sudo apt-get update
sudo apt-get install nfs-kernel-server
```

SERVIDOR: Editar fichero de configuración `"/etc/exports"`

Incluir al final del fichero el directorio a compartir:
Ruta de carpeta
Dirección IP de máquina destino / permisos

Ejemplo:

```
/home/morty 172.16.111.0/24(rw,no_subtree_check)
```

CLIENTE: Instalación de paquetes en el cliente

Instalados por defecto en raspBian
sudo apt-get update
sudo apt-get install build-essential gcc g++ bison flex perl
tcl-dev tk-dev libxml2-dev zlib1g-dev **default-jre** doxygen graphviz libwebkitgtk-

SERVIDOR: Ejecutar cambios realizados

```
sudo exportfs -a
sudo mount 172.16.111.x:/home/morty /home/morty
172.16.111.x es la dirección ip de servidor
```


SERVIDOR: Reiniciar servicios

```
sudo /etc/init.d/rpcbind restart
sudo /etc/init.d/nfs-kernel-server restart
```

PROBLEMAS:

Permisos desde la maquina cliente:

El usuario de la máquina cliente ha de ser sudoer

La carpeta compartida ha de tener todos los permisos `chmod -R 0777 escenario`

A fin de evitar tener que realizar el arranque del servidor de forma manual es recomendable crear un daemon y incluirlo en el directorio `/etc/init.d` para que se ejecute al arranque del sistema de forma automática.

4.4.2. Creación y lanzamiento de servidor NFS como un daemon del sistema

Creación del script:

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:      M.Romero && D. Quinones
# Required-start: $syslog
# Required-stop:  $syslog
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Inicialización de servicios nfs
# Description:
### END INIT INFO

sudo exportfs -a
sudo /etc/init.d/rpcbind restart
sudo /etc/init.d/nfs-kernel-server restart
```

El orden de estos últimos comandos es esencial

Ejemplo de creación de un daemon

```
chmod +x rickStart.sh
cp rickStart.sh /etc/init.d/
cd /etc/init.d
update-rc.d rickStart.sh defaults
```

4.4.3. SSH

4.5. Instalación de SIMCAN

Antes de poder instalar el software SIMCAN es necesario realizar la instalación previa del simulador modular de eventos discretos de redes Omnet++ en su versión 4.6. Además de la suite Inet, que implementa modelos de código abierto OMNeT++ para redes cableadas, inalámbricas y móviles.

Debido a la baja potencia de la Raspberry esta no es capaz de lanzar la aplicación de forma gráfica, esto supone un problema a la hora de realizar la instalación del software. Es por esto que todas las instalaciones han de realizarse de forma manual a través del terminal.

Esto afecta principalmente a la instalación de Inet, ya que las principales guías de instalación disponibles en las webs oficiales parten siempre del entorno gráfico de Omnet++.

Durante el desarrollo del proyecto hemos generado unas guías de instalación y configuración paso a paso que se desglosarán en el siguiente apartado.

4.5.1. Guía paso a paso de instalación en arquitectura ARM

- Descargar los tar.gz de Omnet 4.6, Inet, simcan.tar.
- Esta última (simcan) incluye las bibliotecas que se necesitan para la compilación.

- Copia los archivos .tar de Omnet e Inet en /pi y se descomprimen
- Desde el directorio /pi ejecuta los siguientes comandos

4.6. Optimización y rendimiento

4.6.1. Modificación del GRUB

4.6.2. Paralelización del arranque en servidor

4.7. Seguridad

4.7.1. Eliminación de usuarios y permisos

4.8. Inicialización del sistema mediante scripts

Capítulo 5

Desarrollo del software

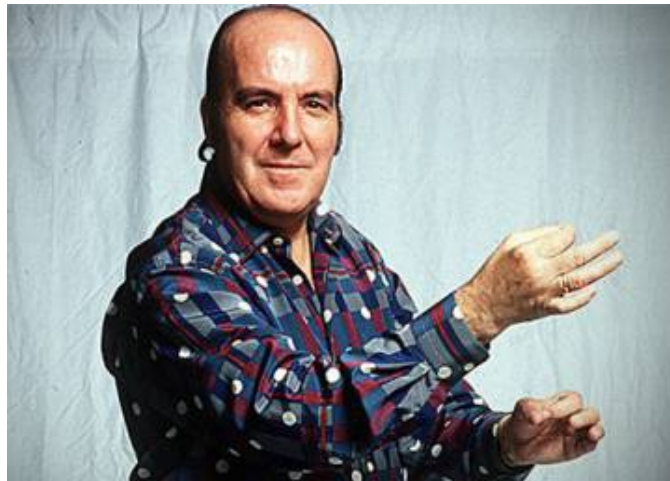


Figura 5.1: Raspberry, te das cuen

Capítulo 6

Pruebas

- 6.1. Comparación con otros procesadores de mayor potencia
- 6.2. Estudio a partir de qué nivel de uso del cluster merece la pena ante un procesador normal
- 6.3. Pruebas contra el cluster de Amazon(???)

Capítulo 7

Trabajo individual

7.1. Miguel Romero Martínez:

7.2. Daniel Quiñones Sánchez:

Capítulo 8

Conclusiones y trabajo futuro

8.1. Conclusiones generales

8.2. Trabajo futuro

8.3. General conclusions:

8.4. Future work:

Herramientas utilizadas

Github

Para realizar el trabajo de forma colaborativa, utilizamos un repositorio privado en la plataforma GitHub, lo que nos permite llevar un control de versiones del proyecto, dividirlo en ramas en función de la fase del desarrollo y mantenernos informados de los cambios que ha introducido cada miembro del grupo.

Latex

L^AT_EX es un sistema de preparación de documentos. Está orientado a la presentación de escritos que requieran de calidad profesional. Se compone de una serie de macros que ayudan a usar el lenguaje T_EX (Wikipedia, TeX). Permite, a su vez, separar el contenido del formato del documento. En este trabajo, hemos utilizado L^AT_EX para la maquetación de la memoria.

Bibliografía

- AB, M. Minecraft. 2011. Disponible en <https://minecraft.net/> (último acceso, Mayo, 2017).
- ALCALÁ, J. Inteligencia artificial en videojuegos. *Laboratorio de Investigación y Desarrollo en Inteligencia Artificial, Departamento de Ciencias e Ingeniería de la Computación, ????*
- ALGORITHMS y MORE. árbol de expansión mínima: Algoritmo de kruskal. 2012. Disponible en <https://jariasf.wordpress.com/2012/04/19/arbol-de-expansion-minima-algoritmo-de-kruskal/> (último acceso, Mayo, 2017).
- ARAUJO, L. y CERVIGÓN, C. *Algoritmos evolutivos: un enfoque práctico*. RA-MA S.A, 2009.
- BULLEN, T. y KATCHABAW, M. Using genetic algorithms to evolve character behaviours in modern video games problem encoding population initialization evaluation selection evolution population replacement. 2004.
- CAPCOM. Street fighter. 1987. Disponible en <http://www.streetfighter.com/> (último acceso, Mayo, 2017).
- CORPORATION, T. Space invaders. 1978. Disponible en <http://www.spaceinvaders.net/> (último acceso, Mayo, 2017).
- ECURED, P. Lisp. 2016. Disponible en <https://www.ecured.cu/Lisp> (último acceso, Mayo, 2017).
- MATEMÁTICA APLICADA Y ESTADÍSTICA, U. D. D. Test de turing. 2004. Disponible en <http://matap.dmae.upm.es/cienciaficcio/DIVULGACION/3/TestTuring.htm> (último acceso, Mayo, 2017).
- FONT, J. M., IZQUIERDO, R., MANRIQUE, D. y TOGELIUS, J. Constrained level generation through grammar-based evolutionary algorithms. 2016.

- GAMASUTRA. Procedural dungeon generation algorithm. 2014. Disponible en http://www.gamasutra.com/blogs/AAdonaac/20150903/252889/Procedural_Dungeon_Generation_Algorithm.php (último acceso, Mayo, 2017).
- GAMES, E. Unreal tournament. 1999. Disponible en <https://www.epicgames.com/unrealtournament/> (último acceso, Mayo, 2017).
- GAMES, H. No man's sky. 2016. Disponible en <https://www.nomanssky.com/> (último acceso, Mayo, 2017).
- GARCÍA-ORTEGA, R. y GARCÍA-SANCHEZ, P. My life as a sim: evolving unique and engaging life stories using virtual worlds. *ALIFE 14*, 2014.
- GUILLÉN TORRES, B. El verdadero padre de la inteligencia artificial. 2016. Disponible en <https://www.bbvaopenmind.com/el-verdadero-padre-de-la-inteligencia-artificial/> (último acceso, Mayo, 2017).
- INTRIAGO, J. Algoritmo a estrella. 2014. Disponible en <https://advanceintelligence.wordpress.com/2014/10/07/algoritmo-a-estrella/> (último acceso, Mayo, 2017).
- JACKSON, D. Evolving defence strategies by genetic programming. *EuroGP*, 2005.
- KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- DEPARTAMENTO DE MATEMÁTICA APLICADA, U. P. D. M. Triangulación de delaunay. 2015. Disponible en http://www.dma.fi.upm.es/recursos/aplicaciones/geometria_computacional_y_grafos/web/triangulaciones/delaunay.html (último acceso, Mayo, 2017).
- MUÑOZ, M. Juegos roguelike: Historia y actualidad. 2014. Disponible en <http://www.fsgamer.com/juegos-roguelike-historia-y-actualidad-20140414.html> (último acceso, Mayo, 2017).
- NAMCO. Pacman. 1980. Disponible en <http://pacman.com/> (último acceso, Mayo, 2017).
- UNIVERSIDAD DE OVIEDO, C. D. I. A. Problema del coloreamiento de un grafo. 1997. Disponible en <http://www.aic.uniovi.es/ssii/Tutorial/Grafos.html> (último acceso, Mayo, 2017).
- PÉREZ, D., TOGELIUS, J., SAMOTHRAKIS, S., ROHLFSHAGEN, P. y LUCAS, S. Automated map generation for the physical travelling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 2013.

- SOFTWARE, G. Borderlands. 2009. Disponible en <https://borderlandsthegame.com/> (último acceso, Mayo, 2017).
- TOGELIUS, J., PREUSS, M., BEUME, N., WESSING, S., HAGELBÄCK, J., YANNAKAKIS, G. N. y GRAPPIOLO, C. Controllable procedural map generation via multiobjective evolution. *Genetic Programming and Evolvable Machines*, 2013.
- TOY, M., WICHMAN, G. y ARNOLD, K. Rogue. 1980.
- UNITY. Angry bots. 2011.
- WIKIPEDIA (TeX). Entrada: “TeX”. Disponible en <https://es.wikipedia.org/wiki/TeX> (último acceso, Mayo, 2017).