

---

## INTRODUCCION.

Supongo que algunas cosas merecen ser explicadas. Bien, este es mi tercer hackmeeting. El primero al que asistí fué al **Madhack 2002** en el célebre Laboratorio de Lavapies, en Madrid. Quede impresionado al ver a toda aquella gente utilizando **Gnu/Linux** con absoluta normalidad. Recuerdo un ordenador funcionando sumergido en una pecera de aceite.

La charla de **Stalman** sobre las patentes de software, la charla sobre criptografía, la colección de viejos ordenadores personales, en la segunda planta. Esta era entonces una vieja imprenta anterior a la época de la informática. Viejas mesas de madera con amplias cajoneras, cuadrículas para guardar inexistentes tipos de plomo. Lámparas en forma de cono suspendidas de un techo de doble altura. En fin, todo aquello que conformaba un departamento tipográfico de los años cincuenta.

El año pasado (2004), el hackmeeting fué en la plaza del Pumarejo, en Sevilla. Esta vez, en lugar de realizarse en un gran espacio ocupado, tuvo lugar en una serie de espacios más pequeños pero bien comunicados. La fiesta tecnopagana fué espectacular. La charla phreakerhack, sobre la génesis del movimiento hacker me gustó mucho. Otra charla titulada: "mentiras sobre la piratería" fué divertidísima. En fin, cada hackmeeting es diferente.

En este hackmeeting de Menorca 2005, estoy escribiendo un diario en donde tomo nota de todo lo que aprendo. En principio es un recurso propio. Un remedio, en cierto modo, a mi falta de memoria. Tengo la costumbre de tomar nota de todo. Soy como el escriba del hackmeeting. Curiosamente, resulta que puedo acordarme de charlas enteras, horas después de haber asistido a ellas. Así que voy tomando notas en cualquier sitio. No me preguntes a que hora me levante ayer, o si he hablado contigo esta mañana. Sinceramente, no puedo recordar lo. Mi memoria me juega estas pasadas.

Finalmente, alguna gente, me ha pedido que les envíe este texto, y eso estoy haciendo. Ha pasado ya una semana del fin del hackmeeting (que pena) y voy recordando todo con claridad.

¿Qué contienen mis diarios exactamente?

Una mezcla de impresiones personales, charla técnica, notas sobre temas concretos (en cualquier lugar), etc. No es un texto ordenado y coherente, sino un conjunto de notas, tutoriales y anexos unidas por una única estructura lógica: el paso del tiempo. Se parece en esto, creo yo al propio hackmeeting.

Si alguien quiere enmendarme la plana, está autorizado para hacerlo. Especialmente interesante me parecería que alguien corrigiese el anexo 5 (**GNU/MUS**). Soy el rey de los novatos jugando al Mus.

## 20-10-05.

Llegada. Toma de contacto. Los habituales problemas con la red. Disponemos de una infraestructura de red para compartir información a la que llamamos **P2P Humano**. Cada cual sube lo que tiene, y descarga lo que le gusta. **Compartir es bueno**. He hecho público gran parte de mis ficheros (anexo2).

El acceso a la red es:

<http://10.0.0.11>  
<ftp://10.0.0.1>

Yo utilizo **gFtp** para las transferencias de ficheros. Por la noche tenemos una asamblea, y cambiamos de sitio los colchones. Es divertido. Más tarde, Marchesi nos propone un par de ejercicios de programación muy interesantes. Se trata de escribir un programa que muestre su código. No vale el comando **cat** o generar un fichero de salida. Otra propuesta consiste en escribir un programa que convierta a números romanos y luego multiplique con ellos. Tampoco vale hacer una función de conversión y luego operar normalmente.

Finalmente, nos mostró un extraño juego de **Emacs**, llamado "*las torres de Hanoi*". Se trata de una construcción en forma de torre formada por piezas de diversos anchos. La pieza superior siempre debe de ser menor que la precedente en el nivel anterior. Si queremos calcular 64 niveles, son necesarios 2 elevado a 64 segundos de calculo. Es decir que aun no se ha completado (partiendo como punto de comienzo el arranque del primer sistema **Unix**, en 1970.)

## 21-10-05.

Programas utilizados para gestionar redes Wireless:

**kismet** \*  
**wawemon**

*\*Mucho mejor.*

Para configurar una red wireless sobre un procesador Centrino. Lo primero que se debe hacer es instalar el Kernel 2.6. Después asegurarse de tener bien instalado (y en una versión adecuada) el driver **iee80211**. Más tarde, se descarga de intel el driver **ipw2100**. Si hay problemas, son debidos generalmente a las dependencias entre las versiones de ambos paquetes.

Para jugar al **freeciv**, en mi ordenador son necesarias algunas operaciones previas. En primer lugar, **freeciv** no se puede ejecutar en modo **root**. Mis usuarios no pueden lanzar

el sistema gráfico, así que es preciso cambiar desde una consola a un usuario normal. Escribimos:

**xhost +**

Autoriza el Display al resto de los usuarios

Y después:

**su antonio**

El nuevo usuario de la consola es: antonio

A continuación, lanzamos en primer lugar el servidor de freeciv:

**civserver**

Finalmente, en otra consola:

**freeciv**

Y ya solo queda escribir en la primera consola:

**star**

Y a jugar!

El escritorio que utilizan algunos hackers es **Ion**.

## **22-10-05.**

**Charla de Marchesi sobre Emacs.** A las 11.00, más o menos. En los ordenadores de los años 50 la entrada se realizaba mediante tarjetas perforadas y la salida mediante impresoras o teletipos. Las tarjetas se perforaban mediante una máquina de escribir especial. La edición se realizaba de modo no interactivo. Por un lado, tenemos el texto que queremos editar y por otro los comandos de edición. Sin embargo, existieron editores interactivos mucho antes de lo que se suele creer. A pesar de ello, se utilizaron mucho más los editores no interactivos. El primer editor interactivo fué **teco**. DEC (conocida normalmente como Digital) hizo evolucionar **teco**, incluyéndolo en sus máquinas a medida que estas se hacían cada vez más potentes. Pero, por motivos desconocidos, en las universidades se trabajaba cada vez más con editores de líneas, lo cual suponía un considerable retroceso.

**Qed**, era un editor muy bueno realizando búsquedas mediante expresiones regulares. Por su parte, el **MIT**, desarrolló un **teco** propio mucho más complejo, que se llamó **ITS teco**. **DEC** respondió llamando a su **teco**: **standar teco**.

**Stalman** se convirtió en el señor “**macro de teco**” del MIT. Se dedicaba a recopilar macros y a distribuirlas en cintas de datos.

Más tarde, **Stalman** viajó a **Stanford** y vió el editor que tenían allí. Era el **ed**, un editor capaz de escribir el carácter **a** cuando se tecleaba una **a**, en tiempo real. Esto le gustó tanto a Richard, que lo implementó en **teco**.

Con el paso del tiempo, el creciente número de macros empezó a producir serios problemas de organización. Por ello, y antes de que la cosa fuera a mayores, se buscó una solución. Esta fue bastante sencilla. Ya que había un archivero (Richard), pues nada, se junta todo en un solo editor y listo. Este editor se llamaría **Emacs**.

La forma de desarrollo de **Emacs** se denominaba la comuna de **Emacs**. Su filosofía acabaría dando lugar a **GNU**. En el libro de **Raymond**: “**la catedral y el bazar**”, la referencia a la catedral, se refería a **GNU**, no al software privativo.

Pasemos a la acción.

Veamos, Cuando arrancamos **Emacs**, se abre un buffer nuevo por defecto, que se llama **scratch**. Podemos escribir lo que queramos en este buffer. Para lanzar un buffer nuevo escribimos: **ctrl x**. Para un primer contacto con **Emacs**, lo mejor es explicar los conceptos que incorpora (especialmente los que son propios y característicos).

Empezar a experimentar con los comandos sin saber lo que es **Emacs** es un error. Empecemos pues, por explicar los conceptos fundamentales.

Estos son seis: **ventana**, **frame**, **buffer**, **puntero**, **marca** y **región**. El **buffer** no hace falta explicarse. Una ventana es un agujero que nos muestra un **buffer**. Es el area blanca donde escribimos. Un **buffer** no tiene porque tener asociado un archivo. Creamos una nueva ventana (**ctrl x 2**) para que sea horizontal y **ctrl x 3** para que sea vertical). Como solo hay un **buffer**, si empezamos a escribir veremos el mismo texto en todas las ventanas.

Podemos editar los **buffers** que queramos. Para crear un **buffer** nuevo escribimos: **ctrl x b**. Para asociarlo a un fichero nuevo escribimos su nombre en el **minibuffer**. El **minibuffer**, es una fila situada en la parte inferior de la pantalla que sirve para comunicarse con **Emacs**. Es desde este lugar, desde el que buscamos o guardamos los ficheros. la línea que se encuentra encima del **minibuffer** se llama **modeline**. Aporta información como el nombre del **buffer** y el **modo** en que se encuentre.

Una **frame**, es lo que en **Xwindows** llamamos una ventana. Podemos abrir todas las **frames** que queramos. Una **marca** es una señal invisible, que se utiliza para seleccionar un espacio entre caracteres. Para que el usuario pueda verlo se muestra a su derecha, una especie de **cursor**, que en **Emacs** se llama **puntero**.

Para seleccionar fragmentos de texto, se hacen dos **marcas**. Los caracteres entre ellas son el texto seleccionado. Pero se da una situación problemática. Las **marcas** no se ven, así que no hay forma de saber qué tenemos seleccionado. Para solucionar esto, hay dos métodos. Uno adecuado y otro bastante poco conveniente. El segundo, es la solución de **Suse**, que consiste en colorear la zona de selección de texto.

Un método realmente original... La solución que, realmente puede denominarse como tal, consiste en resaltar los punteros durante un instante. Esto, es suficiente para ver lo que tenemos seleccionado, de una forma bastante elegante.

Más adelante, Marchesi nos mostró su **Emacs**. Los usuarios de **Emacs** tienden a configurar el programa y adaptarlo a sus gustos y necesidades. Uno de los aspectos más potentes de **Emacs** son los modos. Estos permiten cambiar el aspecto del texto según unos modelos previamente definidos. Estos modos los puede realizar el propio usuario. Por poner un ejemplo, si vamos a utilizar un nuevo lenguaje de programación, para el cual **Emacs** no tenga soporte, creamos un modo nuevo. Si le decimos que el carácter comentario es el carácter `?`, entonces al aplicar este modo a un texto, todos los comentarios aparecerán precedidos por el carácter `?`.

#### **Charla de Marchesi sobre Software Libre. A las 13:00 horas.**

En los años 80 aparecen los ordenadores asequibles para el público general. Así nació un nuevo modelo de negocio: el del software. El modelo se planteó para el lucro de unos cuantos. La inexplicable falta de control del aparato del estado, permitió un negocio donde los usuarios en la práctica carecen de derechos. Este sistema, que puede ser calificado, sin exagerar, de aberración, no tiene precedentes. Si en cualquier otro sector económico se hubiese intentado hacer algo parecido habría habido auténticas revueltas.

- ¿Por qué no hay revueltas en la informática?
- ¿Por qué el estado permitió el modelo privativo en la industria del software?

Pues no son preguntas fáciles de responder. Pero estamos aquí para intentarlo. La opinión de Marchesi al respecto es la siguiente. Existe una brecha entre el conocimiento científico y tecnológico y lo que la población mayoritaria conoce al respecto (incluidos los políticos). Se permitió, porque las autoridades no entendían de estos temas, lo cual propició que todo quedase en manos de unos pocos expertos. La industria solo tenía que "convencerles" para que apoyasen sus propuestas. Por otro lado, también falló la opinión pública, que no hizo nada por evitarlo. El desconocimiento, y la idea ampliamente difundida, de que los temas tecnológicos no eran asunto suyo fueron decisivos.

En mi opinión, también es clave el fracaso del activismo político, que no supo comprender, en un momento clave, la importancia que tenía la reglamentación legal del negocio tecnológico. Como se suele decir, esta nos la colaron...

Cuando se estableció el modelo propietario, algunos hackers empezaron a trabajar para intentar recuperar el modo de hacer las cosas que existía con anterioridad.

El uso de software privativo por parte de la administración impide garantizar el derecho de privacidad de los usuarios. Nadie sabe lo que hace el software propiedad de empresas privadas. Como ejemplo de ello, la célebre polémica sobre el sistema operativo **Windows** de **Microsoft**. Esta empresa, bajo ciertas presiones, muestra las fuentes de **Windows** a algunos gobiernos. Pero el método utilizado para ello es una verdadera tomadura de pelo. Las fuentes de **Windows** mostradas no son compilables. Y no hay forma de saber si las fuentes se corresponden con el binario, que es lo que realmente nos vende esta compañía.

En este contexto nace **GNU/Linux** (¡Y Esta historia es sobradamente conocida!, así que me la ahorro).

El **Hurd** empezó a desarrollarse en 1989. Es el kernel de **GNU**. Se está tardando tanto tiempo en desarrollarse por que es un proyecto muy ambicioso. Seguramente esta ambición fue un error. En aquél momento, necesitábamos un núcleo. No el mejor núcleo del mundo, sino uno que funcionase razonablemente. Como no llegamos a tiempo con **Hurd**, tuvimos que buscar una alternativa. Esta fue el kernel monolítico de **Linux Torvald**.

La licencia **GPL** se realizó para facilitar a los desarrolladores el uso de licencias de distribución libres, capaces de resistir cualquier pleito. En principio, Los creadores de software tienen derecho a escribir sus propias condiciones de distribución. Lo que pasa, es que llega un abogado y da la vuelta a sus palabras. Por ello, se contrató a un equipo de abogados muy competente para implementar la **GPL**.

La revolución **GNU**. El objetivo de **GNU** es muy claro: **"Difundir el modelo libre"**. Es decir, devolver a los usuarios sus derechos. Después de esto. Nos podemos ir a casa.

Supongamos que comparamos la estrategia de desarrollo de **GNU**, con la estructura arquitectónica conocida como arco. Este arco, es un poco especial. Realmente está constituido por otros arcos más pequeños. Cada uno de estos arcos simboliza un área donde GNU necesita capacitarse para tener éxito. El primer arco es la capacidad tecnológica. Cuando empezamos con esto hace ya algunos años, no se sabía si el modelo del software libre sería capaz de producir software de calidad, que además fuera apropiado para su uso por la mayoría de la gente. Hoy en día sabemos que es posible. Así que este pedacito de arco lo tenemos más o menos terminado.

El siguiente arco es la comercialización. Este, lo estamos empezando ahora. Los hackers de esta nueva etapa deberían ser contables o empresarios, en lugar de técnicos como hasta ahora.

**Charla sobre estética ciberpunk.** Esta charla, consistió en una serie de fragmentos de películas del género conocido como ciencia-ficción, relacionados con el movimiento **ciberpunk**. Pudimos ver **Brazil, blade runner, siete monos, nivel 13, matrix, dark city**, etc. El ponente, fué acompañando las escenas con reflexiones acerca de la mismas y su relación con la estética **ciberpunk**, tecnología y política.

Estas películas están ambientadas en un futuro muy cercano, como quien dice, a la vuelta de la esquina. El mundo se presenta como un lugar duro y oscuro. Una especie de edad media contemporánea, mezcla de tecnología y caos. Es decir prácticamente nuestro mundo actual. La relación estética del **ciberpunk** y el mundo **hacker** es bastante clara. Empezó en los años 30, cuando los adolescentes se acostumbraron a vestirse con chaquetas de cuero y gafas de sol negras con cristal reflectante. Estas personas mostraban así su rechazo al mundo convencional.

Refugiados en su indumentaria manifestaban su independencia personal. Muchas veces también su escepticismo e indiferencia hacia todas las cosas. Estos tipos duros pueblan ambientes desoladores, caóticos y tecnológicos. La mayoría de las películas están basadas en obras literarias. **Neal Sthepenson** y **Gipson** son los autores más reconocidos.

En sus obras los protagonistas son, muchas veces mujeres, aunque muy alejadas del estereotipo femenino. Son mujeres duras, sobreviviendo en situaciones extremas. Los personajes generalmente pertenecen a minoría rechazadas. Un buen ejemplo es el protagonista de **Gattaca**. Reducido al nivel social más bajo por razones genéticas, a pesar de poseer un talento y capacidad envidiables. Este carácter marginal hace, que por lo general, no deseemos estar en la piel de nuestros personajes, a pesar de que nos fascinen. Su existencia es demasiado dura para desearla. Como ejemplo, de mujer **ciberpunk** tenemos a **Trinity**, la compañera de **Neo** en **Matrix**.

La mayoría de las películas son de bajo presupuesto. Eso se nota bastante en los resultados estéticos, aunque a mí no me ha importado esto nunca demasiado. La mejor acabada es **Matrix**, que si dispuso de un gran presupuesto. Pese a lo que se suele decir a veces, **Matrix** no es ni mucho menos, la mejor del género. De hecho, pese a su buen acabado formal, es un remake casi continuo de otras películas. **Mátrix**, toma mucho del manga, y algunas escenas están reproducidas literalmente de la obra **Neuromance** de **Gipson**.

La estética es la misma de los muchachos de los años treinta: las gafas negras y la ropa de cuero. El cuadro se completa con las consolas de caracteres verdes sobre fondo también negro, que simbolizan el ciberespacio.

Ya hemos mostrado a los personajes, y los hemos ubicado en su mundo. Ahora nos toca ver el fondo de todo ello. ¿Qué nos quieren decir? En definitiva, cual es el tema de real, que cada libro o película intenta expresar a su manera. Yo creo, que lo que se dice es que el mundo tecnológico en el que vivimos es una farsa, una simulación. Soluciones tecnológicas para hacer frente a problemas generados por la tecnología, que crean a su vez nuevos problemas. Una venda en los ojos que nos impide ver la realidad. Una realidad mucho más terrible de lo que podemos imaginar. Pero también nos muestra la capacidad de algunos individuos para fundirse con la tecnología y organizarse para la lucha.

En definitiva es el intento de volver a poner las cosas en su sitio lógico. De devolver al hombre su espacio de vida real.

En este momento, el ponente relaciona el movimiento **ciberpunk** con el mundo **hacker** actual. Se pregunta, si nuestro conocimiento de la tecnología informática, y nuestro nivel de autorganización pudieran servir para articular un verdadero movimiento tecno-social, capaz de defender los derechos de los ciudadanos frente a los abusos del poder.

## 23-10-05.

### **El último día. Charla sobre GPG. Seguridad informática.**

El objetivo de este taller es proteger nuestros datos personales y nuestro sistema del ataque de hackers. o en cualquier caso minimizar las consecuencias de un ataque.

El ponente ha escrito un texto en una pizarra. Este contiene una lista de direcciones de correo electrónico y también la **finger print** (huella dactilar). Sirve para establecer un primer nivel de confianza. Para verificar que estos datos son correctos, cada usuario se identifica físicamente con su dni, u otro documento similar. Existen varios niveles de seguridad en caso de problemas de identificación.

Este documento se sube a un servidor de claves. El usuario se tiene que descargar la clave del servidor de claves para poder usar el cifrado de documentos, mensajes, etc con **GnuPG** (Gnu Privacy Guard). Para más detalles: **man gnupg**.

Y llega la hora de regresar. Preparamos un cable de red cruzado para intercambio de datos directo entre dos ordenadores. Es el método más rápido. Como pasarela o gateway utilizamos el número ip del otro equipo. Tenemos problemas para intercambiar información mediante ssh, así que utilizamos **scp** (anexo3). La velocidad es excelente.

En el aeropuerto de Menorca hay wireless pero tiene un rendimiento muy bajo.

---

## **ANEXO 1. LIBROS – [sindominio.net/traficantes](http://sindominio.net/traficantes).**

- La pastilla roja. Miguel de Icaza.
- Por una cultura libre. Lawrence Lessig.
- Software libre para una sociedad libre. Richard Stallman.
- El código. Lawrence Lessig.
- Sobrevivir a internet.
- Crítica de la razón informática. Tomás Maldonado.
- El control de internet. Josep Ibañez.
- La fábrica de la infelicidad. Franco Berardi (bifo).  
rekombinant.org

---

## **ANEXO 2. Relación documentos públicos Antoniux.**

### **1-Copyleft.**

#### **1.1-3D.**

- Ficheros de Blender:
  - 225\_game\_demos..... 24 ficheros de ejemplo.
  - Ar\_deco..... proyecto personal. Contiene 160 elementos entre ficheros de blender, imagenes, texturas y videos.
  - test233..... 36 ejercicios de Blender.
  - Proyectos 3D..... Diversos objetos de mi creación. En varios formatos.
  - MESHERS. Otra galería de objetos 3D propios de libre distribución. 148 objetos. Generalmente naves espaciales.

#### **1.2-TEXTOS.**

- Tutoriales. Tutoriales sobre Gnu/Linux. Iniciación, Shell script, vnc, servidores, sketch, etc.
- Textos fundamentales sobre software libre. En el principio fué la línea de comandos, la catedral y el bazar, etc.

#### **1.3-PELICULAS.**

- Hackandalus-04. 26 ficheros en formato .ogg y .ogm de theora. Son las charlas del hackmeeting el año 2004, que tuvo lugar en Sevilla.

-Jornadas telemáticas 2004. 9 ficheros en diversos formatos, con las charlas de las jornadas telemáticas que tuvieron lugar en el Kaslab de Madrid el año 2004.

-Jornadas telemáticas 2005. 12 ficheros en formato .ogg y .ogm de Theora. Idem, pero en el año 2005. Ánimo, que están todavía calentitas.

-3D Desktop. 5 ficheros en formato .ogg de Theora. Muestran

ciertos efectos de escritorio en 3D sobre Linux que se están investigando.

-Personal. 2 ficheros de animación 3D propios.

## 1.4-IMAGENES.

-Temas. Diversos temas para enlightenment, windowmaker, y metacity.

-Fondos de escritorio. Capturas de pantalla del escritorio. 45 elementos.

-Hacking. 2 imágenes de carteles del madhack 02.

## 1.5-JUEGOS.

-Acorn, freecraft, phoenix y gltron.

## 2-Copyright.

### 2.1-MUSICA.

- MP3. Colección de música clásica.

-Suelos. 49 ficheros. Purcell, Prokofiev, Rachmaninov, Tchaikovsky, Bach, etc. 49 ficheros.

-Henry Purcell. 6 ficheros.

-Johan Sebastian Bach. 2 ficheros.

-Rachmaninov. 6 ficheros.

-Segei Prokofiev. 4 ficheros.

CONTENIDOS ADICIONALES. Por autores. 18 GB.

App

Rammstein

Bach	Supertramp
Borodin	Tangerine dream
Claude Debussy	Tomita
Dimitri Shostakovitch	Toru Takemitsu Orchestral Works
Erik Satie	Vangelis
Giacomo Puccini	W.carlos
Henry Purcell	bso
Himekami	cccp
Mussorgsky	joseba
Maurice Andreau	kachaturian
Michael Nyman	prokofiev
Miles Davis	vivaldi

## 2.2-PELICULAS.

-Startreck. Capítulo 37 de la serie de televisión original. Titulo: El suplantador. En Castellano. Formato Divx.

## 2.3-JUEGOS.

-Wolfestein 0.7.16-1 para Gnu/Linux. Demo.

-Unreal 2003 para Gnu/Linux. Demo.

## 2.4-SISTEMAS OPERATIVOS.

- Solaris 10. 4 Cds en formato .iso.

---

## ANEXO 3. PROTOCOLO DE INTERCAMBIO DE DATOS SCP

Sintaxis:

```
scp -r archivo antoniux@codigoabierto.info:/home/antoniux
```

Donde:

**scp** es el protocolo.

**-r** : recursivo. indica que se va a copiar más de un fichero.  
**archivo**: nombre de fichero en ordenador del cliente.  
**antoniux@codigoabierto.info**: cuenta de usuario.  
**/home/antoniux** : ruta de destino en el servidor para el fichero que estamos copiando. Si se desea hacer lo contrario, se invierte la sintaxis. Si se quiere copiar un solo fichero no se pone **-r**.

---

## ANEXO 4. COMANDOS BASICOS DE FREECIV.

<b>g</b>	mover unidad.
<b>x</b>	autoexplorar.
<b>May D</b>	desarticular unidad.
<b>s</b>	vigilancia
<b>b</b>	construir ciudad.
<b>f</b>	atrincherar.

---

## ANEXO 5. GNU/MUS.

El mus es un juego de cartas tradicional muy interesante. He aprendido a jugar en este Hackmeeting (quien lo habría imaginado). Realmente, solo he aprendido a jugar un poco. El mus es un juego complejo. Se parece algo al poker en el sentido de que también se hacen apuestas. Pero el mus es un juego mucho más completo. Se reparten cuatro cartas. El que está a la derecha del que reparte (postre) se llama mano. Es importante ser mano, porque gana siempre en caso de empate.

Es también el que de algún modo empieza el juego. Este tiene cuatro partes: grande, chica, dobles y juego. La grande es obtener la mayor puntuación. La chica la menor. Dobles, tener una o dos parejas, o mejor aún tríos. Dobles parejas se llama dúplex.

El juego consiste en tener 31 puntos. La segunda mejor jugada es 40. 39, 38, etc son peores, pero no puedo precisar bien en que medida. El valor de las cartas es el siguiente: Las figuras valen todas 10 puntos. Todas las demás cartas tienen su valor excepto el dos que vale un punto y el tres que vale lo mismo que un rey. Este último es llamado "cerdo"

Las parejas más valiosas son las de valor más alto. Los reyes son las cartas más altas en este sentido. Pero dobles parejas tiene mayor valor que la pareja de "cerdos". Hay más combinaciones, pero no es momento de complicarse.

No es un juego mecánico. En todo momento se habla, incluso de cualquier otra cosa. Se juega por parejas, y se hacen ciertas señas características, para que tu pareja sepa que cartas llevas. Si tus cartas no te gustan pides mus. Si todo el mundo está de acuerdo te descartas y punto.

Al final se cuenta todo (la grande, la chica, etc), y se reparten las piedras, que se llaman amarracos. Hay piedras de uno y cinco puntos. En todo momento se pueden establecer ciertas apuestas sobre lo que el contrario dice tener. Es un juego de inteligencia y frialdad. Tu oponente afirma tener grandes. Esto puede ser cierto o no (se cuenta al final). Si no te lo crees inicias una primera apuesta con él. Si el

quiere aumentará la apuesta o pasará. Las apuestas tienen un valor que se toma en el recuento final. Si han visto tu envite (apuesta inicial), y quieres insistir, porque estas seguro de tus cartas, o por que piensas que vas a engañar a

tus contrarios proclamas a los cuatro vientos: ¡Ordago!

El envite es la apuesta normal. El ordago es el envite máximo. Hay que estar muy seguro para hacerlo.

Esto es suficiente para sentarse en una mesa e intentar jugar al mus. El resto, con un poco de práctica...

---

Antonio Becerro Martinez

[becerrodelinux@yahoo.es](mailto:becerrodelinux@yahoo.es)

Menorca - 2005