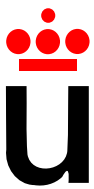
# Redes sociales: ¿todos para uno o todos para todos?

## Arquitecturas centralizadas o federadas

Jesús M. González Barahona

jgb@gsyc.es
http://identi.ca/jgbarah http://twitter.com/jgbarah
GSyC/LibreSoft, Universidad Rey Juan Carlos

Colmux 2012
Colmenar Viejo, 4 de mayo de 2012

Universidad
Rey Juan Carlos

# ¿Qué es una red social?

*"Estructura social compuesta de grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos."*

"Redes sociales", Wikipedia-ES, mayo 2012
`http://es.wikipedia.org/wiki/Red_social`

# En realidad, hablamos de un servicio...

*"Medio de comunicación social que se centra en encontrar gente para relacionarse en línea. Están formadas por personas que comparten alguna relación, principalmente de amistad, mantienen intereses y actividades en común, o están interesados en explorar los intereses y las actividades de otros."*

"Servicios de red social", Wikipedia-ES, mayo 2012
`http://es.wikipedia.org/wiki/Servicios_de_red_social`

# Servicios de red social centralizados...

Normalmente, se montan "como si fueran un sitio":

- Una url base. Por ejemplo:
  `http://facebook.com`, `http://twitter.com`
- Un frontal que proporciona HTML para navegadores
- Una API, que proporciona acceso a aplicaciones
- Una base de datos, con toda la información
- Puntos de control: los tres anteriores
- Mucho interés en mantener ese control
  (base de modelos de negocio)

# Instagram: un ejemplo (con software libre)

- Plataforma: Ubuntu Linux 11.04 sobre EC3
  decenas de máquinas virtuales
- Equilibrio de carga:
  nginx (HTTP, proxy inverso), DNS Round-Robin
- Servidores de aplicaciones:
  - Django (Python web framework)
  - Gnunicorn (WSGI HTTP Server)
  - Auxiliar (control de ejecución en varias instancias): Fabric
- Almacenamiento de datos:
  - PostgreSQL (base de datos)
  - Pgbouncer (conexión con los servidores de aplicaciones)
  - Redis (servidor de datos complejos)
  - XFS (sistema de ficheros sobre RAID)

`http://instagram-engineering.tumblr.com/`

`http://instagram-engineering.tumblr.com/post/13649370142/what-powers-instagram-hundreds-of-instances-dozens-of`

# ...y aislados

- "Si no ves qué se vende, es que el producto eres tú"
- Interés en aumentar la cartera de "productos"
- Necesidad de poner barreras a que los usuarios se vayan, interoperen fuera de esquemas concretos
- Maximización del coste de migración a otro servicio, minimización del coste de migración desde otro servicio
- Los usuarios son agentes activos de crecimiento (invitaciones, presión social, contenidos)

**"Jardines amurallados" en lugar de "web abierto"**

# Cómo el control lleva al negocio

- Interfaz HTML: muestra publicidad, o
  "lo que el proveedor del servicio quiera"
  Ej: Cajas de publicidad en Facebook
- API: acceso limitado y controlado (se puede restringir)
  Ej: Google Maps
- Base de datos: el valor de la información
  Ej: acceso al "firehose" de Twitter
- Modelo fremium: parte gratis, parte de pago
  Ej: Más capacidad (SlideShare), más funcionalidad (Scoop.it)
  normalmente, ambas cosas (Flickr, Vimeo)
- Todo controlado:
  - Legalmente: condiciones de uso del servicio
  - Tecnológicamente: control de acceso, funcionalidad del interfaz

# Todo esto no es nuevo

Algunos episodios del pasado:

- Protocolos de redes, y redes (1970-1995):
  DECnet (Digital), SNA (IBM), Netware (Novell), etc.

  **Internet** como red global

- Servicios a comunidades de usuarios (1985-2000):
  America Online, Compuserve, Prodigy, GEnie, etc.

  **Web** como servicio global

# Un servicio, varias arquitecturas

Correo electrónico:

- Centralizado: Un servidor para dominarlos a todos...
  Hubo sus intentos, pero la escala lo evitó
  ¿O no?: mensajes en Facebook, WhatsApp
- Federado: Cada cual que se organice como quiera
  (pero todos interoperan)
  Aquí estamos ahora (o casi)
- Pares: Cada uno con sus propios recursos
  Aquí podríamos estar

¿Cuál se puede controlar mejor?
¿En cuál se puede innovar mejor?
¿Cuál se puede adaptar mejor?

# ¿Y si metemos nubes por medio?

Hay ventajas:

- Infraestructura de terceros, mantenida de forma transparente
- Mucha escalabilidad, de poco a muchísimo
- Ciertas economías de escala

Pero también problemas:

- ¿Quién tiene acceso a mis datos?
- ¿Quién controla que mi servicio se proporcione?
- ¿Quién obtiene el beneficio de la escala?

Servicios centralizados y "nube" son cosas distintas...
...pero suelen venir juntas

# Ejemplos de redes sociales

- Centralizadas: FaceBook, Twitter
- Federadas: Diaspora, Status.net (Identi.ca)
- Entre pares: por ahora, compartición de ficheros pero hay mucha experimentación

[¿Conocéis Freedom Box?]

`http://diasporaproject.org/`
`http://status.net`
`http://freedomboxfoundation.org/`

# Cómo es una red social federada

- Múltiples instancias:
  - Todas corriendo software similar (Status.net, Diaspora) o no
  - Protocolos para comunicación entre instancias (OSStatus, XMPP)
  - Cada instancia gestionada por una institución
- Los usuarios actúan en la instancia que quieran
- Pueden interactuar con usuarios en otras instancias
- Puede complementarse con respaldo / recuperación en distintas instancias
- Si el software es libre, cualquiera puede montar una instancia

# Redes federadas: características

- No hay puntos especiales de control (salvo instancias enormes)
- Es fácil moverse a otro proveedor
  pero suels haber mecanismos disuasorios, como la dirección
- No hay necesidad de "atraer" contactos
  pueden mantenerse en su proveedor
- Innovación muy repartida...
  ...pero no es fácil generalizar a toda la red, si hay que coordinarse
- Mucha adaptación al usuario en la interfaz...
  ...pero poca en el protocolo
- Distintos modelos de negocio, costes decrecientes con el tiempo

# ¿Cuál es el problema?

El problema no es que una empresa pueda controlar nuesta red social

El problema es que nuestra red social pueda controlarse fácilmente

# La que se nos viene encima

- Dispositivos permanentemente conectados
- Anchos de banda cada vez mayores (móviles y fijos)
- Almacenamiento masivo muy barato
- Dispositivos pequeños, muy poco consumo
- El software, una vez hecho, puede ser muy barato
  (y puede ser mantenido por comunidades)
- El control sobre la propia privacidad más y más importante

¿Qué se puede hacer con un Rasperry Pi, con una FreedomBox?

`http://www.raspberrypi.org/`

# Muchas amenazas de jardines amurallados

- Redes sociales que todo lo engullen
- APIs e infraestructura para aplicaciones móviles
- Ordenadores donde no puedes instalar lo que quieres

¿Seguiremos teniendo un "open web" dentro de 10 años?

# Para leer...

- "Long Live the Web: A Call for Continued Open Standards and Neutrality",
  Tim Berners-Lee, Scientific American
  `http://www.scientificamerican.com/article.cfm?id=long-live-the-web`
- "Walled gardens look rosy for Facebook, Apple and would-be censors",
  Charles Arthur, The Guardian
  `http://www.guardian.co.uk/technology/2012/apr/17/walled-gardens-facebook-apple-censors`
- "Google: Sergey Brin's double standard on web freedom",
  Emma Barnett, The Telegraph
  `http://www.telegraph.co.uk/technology/google/9206897/Google-Sergey-Brins-double-standard-on-web-freedom.html`