My Web Radio



Una Web Radio personale
Per ascoltare quel che vuoi
Quando vuoi

Che cos’è una Web Radio

- Una **Web Radio** non è altro che un **server** che
  - **Accetta** connessioni dai client
  - Per ogni client **trasmette** un flusso audio, normalmente in TCP
- Di norma i **flussi** per i **vari client** sono **identici**
- Normalmente le **WebRadio** si comportano in maniera **molto simile alle radio private**
- Posseggono un **palinsesto**
  - Trasmettono **programmi registrati** o **in diretta**
  - Ecc.
- Normalmente hanno un **IP pubblico**, sono **connesse** ad Internet **senza vincoli** e ci si può **collegare da ogni postazione**

Come ci si connette ad una Web Radio

- Per **connettersi** ad una **Web Radio** si deve avere un **dispositivo connesso ad Internet**
- Possono essere
  - Un **PC**
  - Uno **smartphone**
  - Un **hardware dedicato** (un ricevitore Web Radio)
- Lo **smartphone** è la soluzione probabilmente **più diffusa**
- Esistono un’enorme quantità di **app** dedicate sia **per smartphone** che **per PC**, ad esempio **VLC**

I ricevitori hardware

- Il **mercato** offre una certa scelta di **ricevitori WebRadio**
- Se ne trovano sia
  - di '**puri**' (che ricevono **solo le WebRadio**)
  - di **associati** a **sintonizzatori radio** o impianti **WiFi**
- In linea di massima questi **apparecchi** sono decisamente **poco economici**
- I prezzi vanno dalla **sessantina** di **Euro** a diverse **centinaia**







Un’alternativa, il Uebbi



- Qualche anno fa **TIM** ha messo sul mercato il suo **ricevitore WebRadio**
- Si chiama **Uebbi** (con la U e la I finale, non W e Y)
- Oltre alla **funzione di radio** funziona come **sveglia** e consente anche di **consultare alcuni podcast** su vari argomenti
- Se ne **trovano** ancora diversi in **Internet** ad un prezzo **tra i 30 e i 50 Euro**

Una simpatica sveglia

- Se lo si vuole **utilizzare** come **sveglia** o **ninna-nanna** per addormentarsi un utile accessorio è l'**altoparlante da cuscino** o '***pillow speaker***'
- Anche questo è **facilmente reperibile** in Internet a **poco prezzo**
- È **estremamente utile** se non si dorme da soli e **non si vuole disturbare** chi dorme insieme a noi
- Lo si **infila sotto il cuscino**, nella federa
- Si **appoggia l'orecchio** sul cuscino **in prossimità** dell'altoparlante ed i **nostri vicini** di letto **non sentiranno nulla**.



Ma che cosa ascolto?

- Il **problema** con le **WebRadio**, usate in questo modo, è che **non possiamo scegliere il programma** all'orario che ci fa comodo, come con le radio private
- **Non possiamo** ad esempio **addormentarci** con il **programma di musica pop** perché viene **trasmesso alle 10 di mattina**, ad esempio
- I **programmi** hanno un **orario** che spesso **non è** quello che a noi **farebbe comodo**

I podcast

- La **maggior parte** delle stazioni radio **salvano i programmi** trasmessi (ed a volte anche altri presenti solo sul sito) in **podcast**
- Sicuramente lo fanno le stazioni **RAI** e **Radio 24** del Sole 24 ore
- I **podcast** sono costituiti da:
  - Dei **file audio** (**MP3**) salvati sul sito della radio e liberamente scaricabili
  - Un **file indice** con una riga che descrive ognuno dei file MP3
- **Non** sono disponibili **immediatamente** dopo il termine del programma ma di solito **qualche ora dopo**.
- Se ad **esempio** mi interessa il **programma** di musica classica **delle 10** di mattina non lo trovo nel podcast appena è terminato ma **la sera** quando vado a dormire sicuramente **lo trovo**.

Ma e i ‘player’ di WebRadio?

- Abbiamo visto che possiamo trovare dei **riproduttori di Web Radio**
- Questi si **collegano all’indirizzo** da cui la web radio che ci interessa trasmette lo **stream** e **lo riproducono**
- I siti dei **podcast** però **non fanno questo**, non eseguono lo streaming
- Per **riprodurre il Podcast** devo **collegarmi** al sito e **richiedere lo specifico** file che mi interessa, dopo averlo individuato nell’indice
- Per farlo posso ad esempio collegarmi con **programmi come VLC**
- Posso anche **inserire l’indirizzo del podcast** nel **browser** ed aprirlo: il **browser** lo **riprodurrà**
- Oppure posso **scaricarlo** sul mio PC per **riprodurlo da li**.
- Ma sicuramente **non posso** riprodurlo come **WebRadio**



Cerchiamo una soluzione

- Una **soluzione** potrebbe essere quella di **creare un programma** che si **comporti** come una **web radio**:
  - **Accetti connessioni** dai client
  - Alla ricezione di una richiesta di connessione da un client:
    - **Individui** l'**ultimo podcast trasmesso** al client
    - Cominci a **trasmetterlo** a partire **da dove si è interrotto** all'**ultima connessione**
    - Quando **termina**, ne **scelga un altro** e cominci a riprodurlo
- Dato che ho **più ricevitori** per la casa, ma **sono sempre io** ad utilizzarli, questo meccanismo non è applicato separatamente ad ogni singolo player ma **la scelta è unica per tutti**.

Termine della connessione

- La **connessione** del client al server può **terminare** in **due casi**
  - Se l’**utente termina la riproduzione**, quindi il client si sconnette
  - Se il **file** in riproduzione **termina** e quindi il server chiude la connessione
- In questo **secondo caso** il client **apre** automaticamente una **nuova connessione**

Inizio della connessione

- In ogni caso, **all’inizio** di una **nuova connessione** il **server** deve **scegliere il file** da riprodurre
- Devo a questo punto precisare che **io seguo** una **serie di podcast diversi**
- Questi podcast hanno **frequenza diversa** (giornalieri, settimanali, altro) e **diversa priorità** d’ascolto
- Per **ogni podcast** io mantengo una **serie di puntate**
- Mantengo anche **traccia dell’ultima** puntata **riprodotta** / in riproduzione e del **punto a cui è arrivata** la riproduzione.

Lista dei podcast

- Nel **sistema** quindi esiste una **lista ordinata di podcast**.
- Per **ognuno** di essi mantengo:
  - **Percorso** del podcast
  - **File** correntemente **in riproduzione**
- Per ogni **file** mantengo
  - **Nome** del file
  - **Podcast** a cui appartiene
  - **Posizione** corrente di riproduzione
  - **Indicatore** di file **terminato**

Persistenza

- **Tutti i dati** necessari sono mantenuti un **database sqlite3**
- In particolare abbiamo **tre tabelle**:
- **Podcasts** che contiene
  - il **nome** del podcast
  - l **percorso** della directroy che lo contiene,
  - un **numero d'ordine** nella lista di podcast
  - il riferimento al **file corrente**

…. Continua

Ancora Persistenza

- **Files** che contiene i **dati di ogni file**:
  - il **nome**
  - il **riferimento** al **podcast** a cui appartiene,
  - la **lunghezza** in byte del file,
  - il numero di **byte già letti** (la posizione),
  - la **data** di **ultimo aggiornamento**,
  - i **byte per secondo** dell'audio
  - **indicatore** di file **terminato**
- **States** che contiene di fatto **un solo record**, che indica
  - Lo **stato** (in *riproduzione*, *messaggio* …)
  - Il **podcast corrente**