



# **GAP per Android**

*Applicazione Android per l'accesso al sistema Gestione Aule Poli dell'Università di Pisa*

Progetto didattico per il corso di:

**Sistemi Mobili e Pervasivi**

Fabio Falzoi



## Descrizione del problema

Si vuole realizzare un'applicazione che consenta, ad un utente dotato di un device Android, di accedere alle informazioni messe a disposizione dal sistema Gestione Aule Poli dell'Università di Pisa.

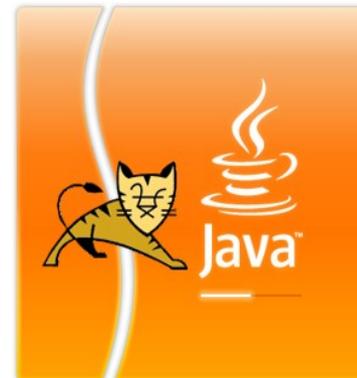
In particolare, si vuole ottenere l'elenco delle aule libere in uno qualsiasi dei poli facenti parte dell'ateneo.



# GAP per Android

GAP per Android è un'applicazione composta da 2 componenti:

- una Servlet JAVA che consente il caching dei dati messi a disposizione dal server del sistema GAP
- una Android app per:
  - l'accesso da device mobile Android alla Servlet
  - la visualizzazione dei dati da essa forniti con una User Interface adeguata ai device mobili





# Sistema GAP - Descrizione

Il server del sistema GAP può essere contattato all'URL:

<http://gap.adm.unipi.it/GAP-SI-NomePolo/rawdata.cgi/DataInizio-DataFine>

Lo script CGI risponde inviando informazioni in formato CSV (Comma Separated Value).

In particolare, viene inviato un elenco degli eventi prenotati su ciascuna aula afferente al polo richiesto.

Ciascuna riga del file CSV descrive un evento prenotato, e contiene, tra le altre, le seguenti informazioni:

- Data
- Giorno
- Aula
- Dalle
- Alle





# GAP Servlet - Caching 1/2

L'estrazione dei dati prima descritti pone un carico non trascurabile sul server, che deve estrarre i dati da un DB non normalizzato: si realizza pertanto un meccanismo di caching.

La Servlet implementa la cache utilizzando una HashMap del Java Collection Framework, dove:

- Le chiavi sono realizzate da oggetti che incapsulano le seguenti informazioni:
  - < Data, Polo >
- I valori sono oggetti che incapsulano le informazioni ritornate dal server GAP, cioè le righe relative a eventi presso quel polo e in quella data.

La cache è salvata in un file sul disco fisso.



## GAP Servlet - Caching 2/2

Strutturando la cache in questo modo, si ha un importante vantaggio.

Supponiamo che un utente faccia richiesta per le aule libere del polo Fibonacci, dal 25/06 al 27/06.

Nell'ipotesi che tali informazioni non siano in cache, la servlet:

- Contatta il server GAP per ottenere l'elenco degli eventi prenotati
- Suddivide le righe CSV in base al polo e alla data
- Aggiorna la cache creando le opportune entries
- Invia le informazioni all'app Android

Qualunque successiva richiesta relativa agli eventi prenotati presso le aule del polo Fibonacci (non necessariamente una richiesta identica in ogni parametro) in uno dei giorni tra 25/06 e 27/06 non provocherà ulteriore traffico verso il server.



# GAP Servlet – Cache Aging e File Lock

A ciascuna entry nella cache è associato un timestamp che permette di scartare informazioni considerate non recenti.

La servlet considera recenti le entry nella cache inserite entro l'ultima ora.

Il file che contiene la cache è protetto da accessi concorrenti mediante:

- Blocchi ***synchronized*** per prevenire race conditions fra più thread
- Oggetti FileLock del package `java.nio.channels` per prevenire modifiche concorrenti al file da parte di processi su altre JVM



# GAP Android – UI 1/4

L'Activity principale della app Android mostra una form che permette agli utenti di conoscere le aule libere.

E' possibile impostare:

- Polo
- Data di inizio
- Data di fine
- Filtro sulla durata

The screenshot shows the main interface of the 'GAP per Android' app. At the top, the status bar displays '3G', signal strength, battery, and the time '5:11'. Below the app title 'GAP per Android', there is a section titled 'Ricerca aule libere'. The interface includes several filter options: 'Polo' with a dropdown menu currently set to 'Fibonacci'; 'Dal' with the date '24/6/2012' and a 'Scegli' button; 'Al' with the date '25/6/2012' and a 'Scegli' button; and 'Libere per almeno' with a dropdown menu set to 'Nessun limite'. At the bottom, there is a large 'Invia Richiesta' button.



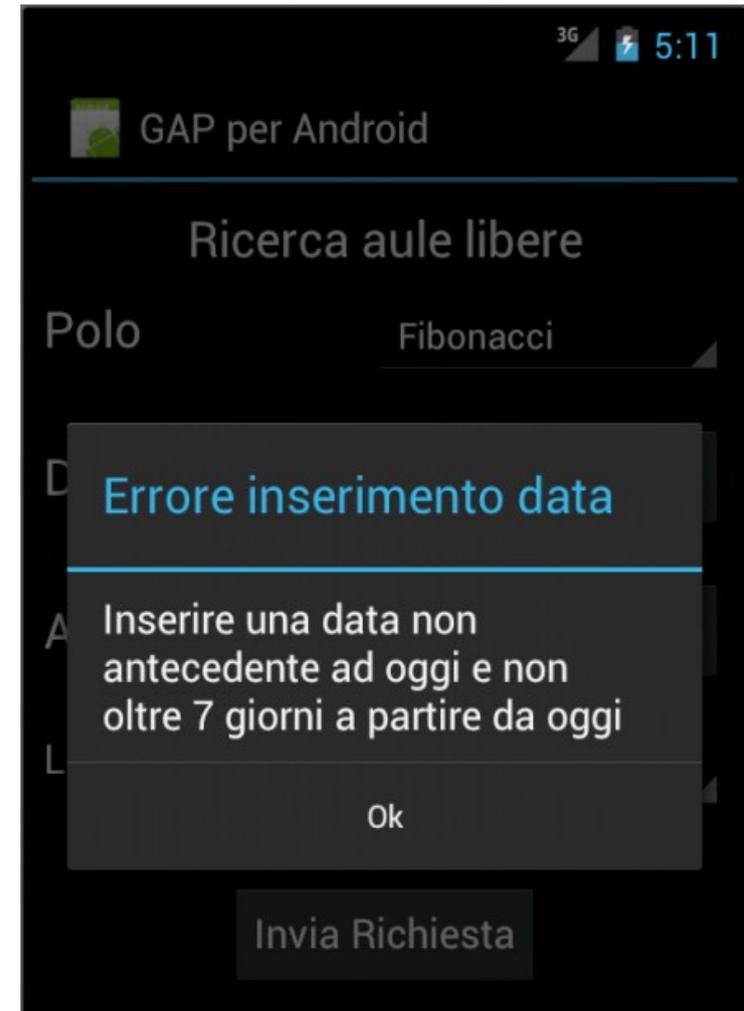
# GAP Android – UI 2/4

I dati inseriti dall'utente sono soggetti a validazione.

Non è consentito:

- l'inserimento di date antecedenti alla data odierna
- l'inserimento di una data di fine precedente alla data di inizio
- l'inserimento di date successive ai 7 giorni a partire dalla data odierna

L'utente conferma la scelta premendo il pulsante Invia Richiesta

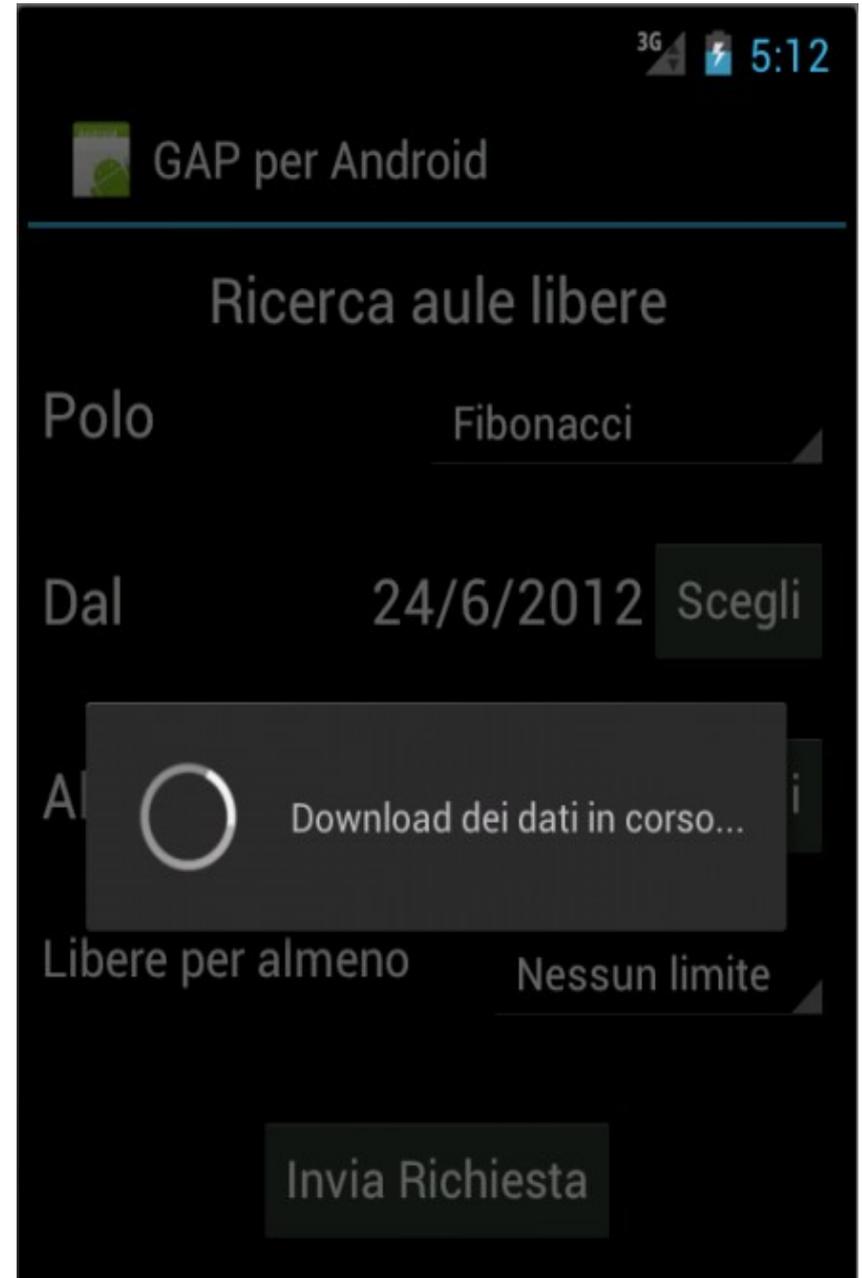




# GAP Android – UI 3/4

Il download dei dati dalla Servlet e la loro elaborazione avviene ad opera di un AsyncTask.

L'utente viene notificato dei progressi nel download e il device resta responsivo ad eventuali azioni intraprese dall'utente.





# GAP Android – UI 4/4

Alla ricezione dei dati, si presenta una seconda Activity dove sono elencate tutte le aule libere afferenti al polo.

Con un tocco su ciascuna di esse, è possibile mostrare in quale orario sono libere.

In alto, l'utente può scorrere con i pulsanti Avanti e Indietro per avere informazioni sulle aule libere nei giorni precedenti e successivi, nei limiti della richiesta fatta.





# GAP Android – Dettagli sul calcolo aule libere

L'elaborazione dei dati consiste nel:

- Parsing del file CSV ricevuto dalla Servlet
  - Si utilizza il package opencsv
- Calcolo degli intervalli di tempo in cui le singole aule risultano libere
- Applicazioni di alcuni filtri:
  - sulla durata degli intervalli (impostato dall'utente)
  - Sui giorni della settimana (es: Domenica)

L'algoritmo implementato per il calcolo degli intervalli in cui le aule risultano libere esegue correttamente in presenza di eventi che:

- Iniziano a qualsiasi orario
- Hanno una qualunque durata