

## Specifiche Progetto

## Documento di Analisi

Bacheca Annunci

Bacheca Annunci - Compra e Vendi Nuovo o Usato

Nome Utente:

Data - Ora	Utente	Tipologia	Categoria	Messaggio
2016-02-19 18:11:28.0	email1@abc.it	Cerco Usato	Film	Il Signore degli Anelli. Coffanetto completo. Che sia in buono stato.
2016-02-19 18:10:33.0	email1@abc.it	Cerco Nuovo	Musica	Discografia dei Queen. Completa.
2016-02-19 18:09:36.0	email1@abc.it	Vendo Nuovo	Computer	MacBook Pro del 2015.
2016-02-19 18:03:39.0	email1@abc.it	Vendo Usato	Cellulari	Samsung Galaxy S6. Bianco 32 GB di memoria interna. In ottime condizioni e con 2 mesi di vita.
2016-02-19 18:02:33.0	email1@abc.it	Vendo Nuovo	Sport	Zaino per le escursioni. Materiale durevole e molto robusto di colore nero.

**Scegli se PUBBLICARE o effettuare una RICERCA**

-> **Pubblica un post in Bacheca:**

Scrivi messaggio annuncio..

Tipologia acquisto

Categoria

-> **Ricerca parola chiave per tipologia acquisto e categoria:**

Inserisci parola chiave da cercare..

Tipologia acquisto

Categoria

**Annunci Bacheca - Le 5 Categorie più presenti:**

Cellulari  
Sport  
Computer  
Film  
Musica

## Scenario:

- 1 → L'utente avvia l'applicazione e il sistema aggiorna le fette del diagramma a torta (percentuali riguardanti le N = 5 categorie, nell'esempio in figura, dei post più presenti nel DataBase) e aggiorna i post presenti nella Bacheca.
- 2 → L'Utente inserisce il **Nome Utente** (questo dato corrisponde alla propria mail e verrà utilizzato per essere contattati ed identificati in modo univoco rispetto a tutti gli altri Utenti).
  - 2.1 → L'Utente preme sul bottone **SINCRONIZZA** per aggiornare i post in Bacheca. In tutti e due i casi ordinati per Data-Ora. Viene aggiornato anche il diagramma a torta con nuove percentuali.
- 3 → **IF** L'Utente sceglie di pubblicare un nuovo Post nell'area **Scegli se PUBBLICARE o effettuare una RICERCA**.
  - 3.1 → L'Utente inserisce il **messaggio annuncio** indicando i dettagli specifici del bene (diretto, breve, con senso e attinente al bene che lo riguarda).
  - 3.2 → L'utente sceglie la **TIPOLOGIA ACQUISTO** comprese tra le seguenti:
    - **cerco nuovo**
    - **cerco usato**
    - **vendo nuovo**
    - **vendo usato**
  - 3.3 → L'Utente sceglie la **CATEGORIA** comprese tra le seguenti:
    - **Automobili**
    - **Immobili**
    - **Cellulari**
    - **Computer**
    - **Libri**
    - **Film**
    - **Elettronica**

- **Musica**
- **VideoGiochi**
- **Abbigliamento**
- **Sport**

3.4 → L'Utente preme sul bottone **PUBBLICA**. Il sistema archivia nel DataBase il post, lo visualizza nella tabella in base al valore Data-Ora. **NB: Se il post non contiene tutti i parametri richiesti, non viene pubblicato.**

4 → **IF** L'Utente sceglie di effettuare una **ricerca per Parola chiave** tra i Post nell'area **Scegli se PUBBLICARE o effettuare una RICERCA**.

4.1 → **IF** L'Utente inserisce la **parola chiave** ( bene o particolare che intende cercare tra i post presenti in Bacheca ).

4.2 → **IF** L'Utente sceglie la **TIPOLOGIA ACQUISTO** comprese tra le seguenti:

- **cerco nuovo**
- **cerco usato**
- **vendo nuovo**
- **vendo usato**

4.3 → **IF** L'Utente sceglie la **CATEGORIA** comprese tra le seguenti:

- **Automobili**
- **Immobili**
- **Cellulari**
- **Computer**
- **Libri**
- **Film**
- **Elettronica**
- **Musica**
- **VideoGiochi**
- **Abbigliamento**
- **Sport**

4.4 → L'Utente preme sul bottone **CERCA**. Il sistema visualizza tutti i post in base ai parametri scelti per la ricerca. Ordinati per Data-Ora. **NB: la ricerca funziona anche con uno solo dei tre parametri di ricerca.**

### **File di configurazione locale in XML**

All'avvio il sistema legge da file di configurazione i seguenti dati:

- l'indirizzo IP e la Porta del server di log
- numero di categorie da visualizzare nel diagramma a torta ( in base ai post più presenti nel DataBase)
- font della tabella dove vengono visualizzati i post.
- durata ( in giorni ) dei post, oltre i quali i messaggi verranno cancellati

### **Cache locale degli input**

Alla chiusura, se si sono inseriti dei dati nelle aree di testo:

- Nome Utente → Inserisci la tua mail..
- Pubblica un post in Bacheca → Scrivi messaggio annuncio..
- Ricerca parola chiave per tipologia acquisto e categoria → Inserisci parola chiave da cercare..

Se selezionati Tipologie di acquisto e Categorie, il sistema salva tutti i dati se non è stato ancora premuto il bottone riguardante la categoria specifica ( **SINCRONIZZA** o **PUBBLICA** o **CERCA** ).

### **Base di dati**

Alla pressione del tasto **PUBBLICA**, il sistema archivia su DataBase:

- Nome Utente ( mail )
- Tipologia Acquisto, Categoria e Messaggio Annuncio

- Viene incrementata la percentuale della Tipologia Acquisto specifica ( diagramma a torta )
- Identificativo del post in Bacheca
- Data – Ora di pubblicazione del post

### File di log remoto

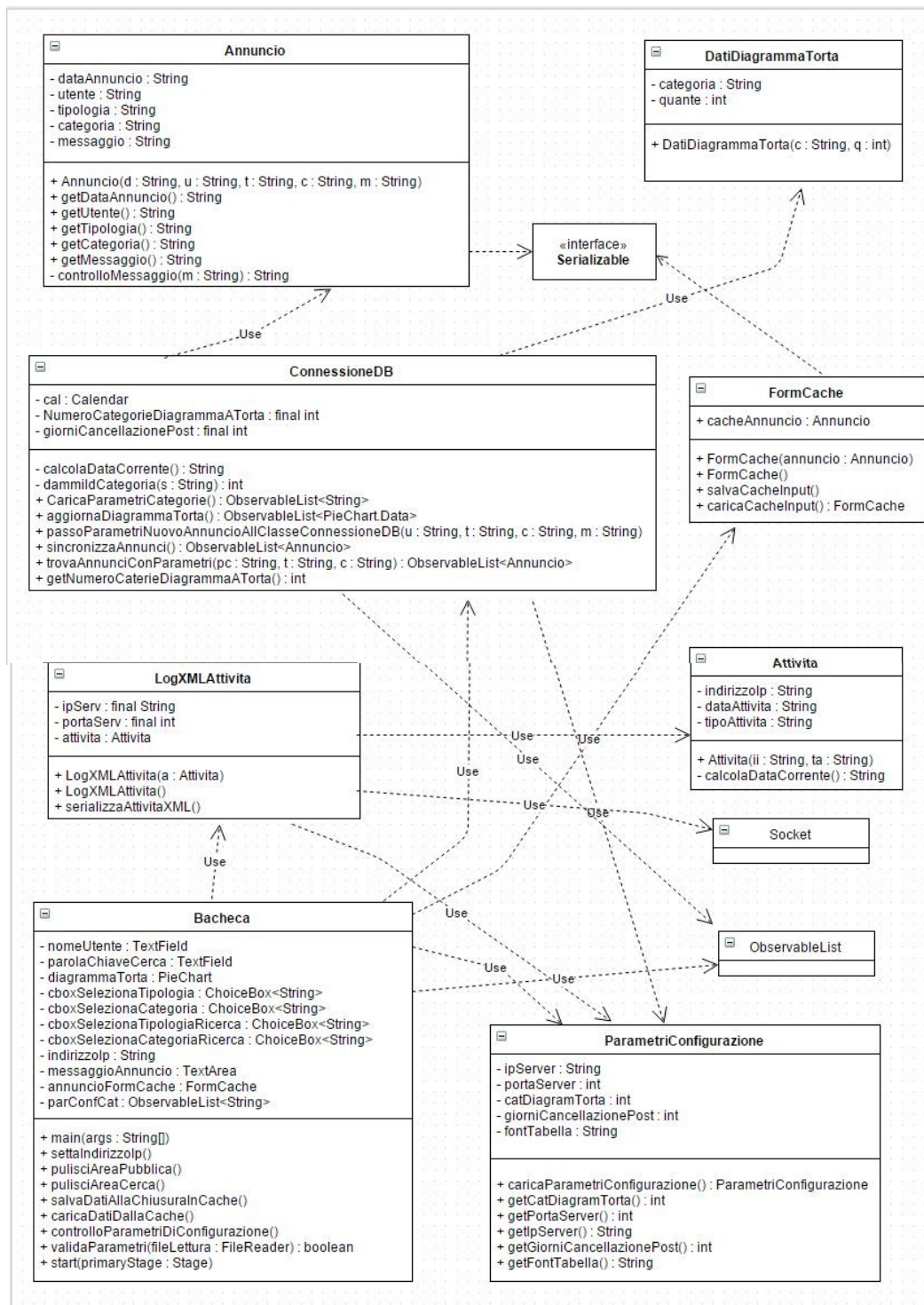
Il sistema invia una riga di log ad ogni evento di seguito elencato:

- avvio dell'applicazione
- pressione bottone SINCRONIZZA, PUBBLICA e CERCA e dei due bottoni PULIZIA CAMPI
- termine dell'applicazione

La riga di log contiene l'indirizzo IP del client, la Data-Ora corrente ed il nome dell'evento.

## Documento di Progetto

### Diagramma UML



## **Responsabilità di ogni classe**

### **Classe FormCache**

- Implementa Serializable
- Contiene i dati di input su file binario (salva e legge da file binario)

### **Classe Attivita**

- Contiene la struttura che andrà a comporre il tipo di attività da registrare.

### **Classe LogXMLAttivita**

Classe che invia l'attività in formato xml alla classe **ServerXMLValida** per la validazione della attività.

### **Classe ConfigurazioneXMLParametri**

- Contiene i parametri di configurazione della applicazione
- legge da file di configurazione alcune caratteristiche dell applicazione

### **Classe Annuncio**

- Contiene i dati che compongono l'annuncio che sarà salvato nel DataBase
- implementa serializable

### **Classe Bacheca**

- Contiene la funzione start() che inizializza lo Stage principale della applicazione
- Gestisce tutti gli eventi che presenta l'applicazione e contiene tutti gli oggetti dell'interfaccia grafica. Contiene bottoni, checkbox, etichette, aree di testo e una tabella dove mostra i post presenti in Bacheca. Contiene metodi che permettono l'aggiornamento degli elementi grafici, elementi che permettono l'invio al Server di log alla pressione dei vari bottoni e all'avvio e alla chiusura dell'applicazione

### **Classe ConnessioneDB**

- Contiene tutte le funzioni che gestiscono le interrogazioni al DataBase grazie alla libreria DriverManager.
- Interagisce con la classe Bacheca in modo da aggiornare l'interfaccia, compreso il diagramma a torta (numero di categorie e percentuali)

### **Classe DatiDiagrammaTorta**

- Contiene i dati diagramma a torta scaricati dal database. Viene utilizzata dalla classe ConnessioneDB per salvare i vari dati che permetteranno la costruzione o aggiornamento del diagramma a torta.

### **Classe ServerXMLValida**

- Implementa un server sempre in ascolto ad una determinata porta e un determinato ip (configurabili da file xml).
- Essa riceve stringhe xml e ne valida il contenuto. Se la validazione è andata a buon fine salva, dentro un file txt, la stringa validata correttamente (file di log con tutte le attività mandate dalla classe LogXMLAttività).

## **DOCUMENTAZIONE**

Questa applicazione potrebbe essere utile alle persone che intendono cercare o pubblicare annunci in una Bacheca. Essa permette sia di cercare beni usati o nuovi oppure la messa vendita di beni usati o nuovi. Questa applicazione è stata pensata ad un vasto pubblico, visto la quantità categorie messe a disposizione per l'utente. Gli utenti interagiranno tramite la loro mail, che sarà proprio il loro nome utente.

### **Scenario tipico**

Uno scenario tipico consiste nell'avvio della applicazione da parte dell'utente. Dopo di che l'utente si troverà davanti una tabella dove verranno visualizzati tutti i post presenti nella tabella, pubblicati fino a N giorni prima dalla data corrente. L'utente si troverà davanti la possibilità di pubblicare un post oppure di cercarlo in quelli già pubblicati in base a dei criteri di ricerca messi a disposizione per l'utente. L'applicazione presenta anche un diagramma a torta che mostra le percentuali delle categorie più presenti nel DataBase della Bacheca.

### **Più precisamente**

- Dopo l'avvio dell'applicazione l'utente inserisce il suo nome utente se non lo ha mai fatto in precedenza.
- Può pubblicare gli annunci, ricordando che il messaggio del bene che vuole mettere in vendita è a sua completa discrezione, in quanto è il diretto interessato a far sì che il suo bene diventi interessante agli occhi di un eventuale compratore. Per quanto riguarda il metodo di pubblicazione dei post, esistono 4 tipologie diverse. VENDO NUOVO : consiste nel mettere un annuncio dove si vuole vendere un bene nuovo. VENDO USATO : consiste nel mettere un annuncio dove si vuole vendere un bene usato. CERCO NUOVO : consiste nel mettere un annuncio dove si esplicita il bene nuovo che si intende comprare, cioè che l'utente cerca, in modo che eventuali utenti possessori del bene possono contattare il proprietario dell'annuncio. CERCO USATO : consiste nel mettere un annuncio dove si esplicita il bene usato che si intende comprare, cioè che l'utente cerca, in modo che eventuali utenti possessori del bene possono contattare il proprietario dell'annuncio.
- Il metodo di ricerca dei post permette di cercare qualsiasi cosa, nello specifico, all'interno della Bacheca Annunci. E' possibile lanciare una ricerca senza selezionare nessun parametro, selezionandone 2, selezionandone N. La ricerca funziona in ogni caso. Naturalmente più parametri si selezionano e più la ricerca sarà specifica. La parola chiave non è Case Sensitive, quindi i post vengono trovati tutti anche scritti in maniera diversa ( si intende lettere maiuscole o minuscole ).
- Quando l'utente effettua la ricerca o la pubblicazione di un annuncio, tutti i campi di testo vengono svuotati e vengono visualizzati subito sulla tabella. Ordinati per data pubblicazione.
- L'applicazione presenta anche un diagramma a torta che si aggiorna in base alla pubblicazione dei post di una certa categoria. Il diagramma visualizzerà i post con le N categoria più presenti nella Bacheca.

## **Caratteristiche della applicazione**

### **Interfaccia grafica implementata in Javafx**

La struttura è esattamente come nell'immagine riportata all'inizio del documento di analisi.

Contiene degli elementi grafici come : Bottoni, Etichette, Aree di testo, Elementi riguardanti il layout, Tabella e Diagramma a torta.

Questa interfaccia ha l'incarico di interagire direttamente con l'utente, tramite la pressione di bottoni e l'inserimento di messaggi di testo. Contiene funzioni coerenti con l'aggiornamento degli elementi grafici ed invia dei Log al server in base a quali bottoni vengono premuti ( eventi ) durante tutto il periodo di tempo che l'applicazione è aperta. Oltre che per la pressione di bottoni anche alla apertura e chiusura della applicazione. La classe che implementa l'interfaccia grafica è Bacheca.

### **Cache locale degli input**

La cache locale degli input è stata implementata tramite la serializzazione di oggetti che permette di realizzare Java. Viene implementata tramite l'interfaccia Serializable e permette la lettura e scrittura di un file binario. La classe è esattamente FormCache assieme alla classe Annuncio che sono tutte e due serializzate. FormCache contiene le funzioni per scrivere o leggere oggetti Annuncio da File, in questo caso con estensione .bin (File binario).

### **DataBase o Base di Dati**

La struttura del DataBase presenta due TABELLE ed una VIEW.

La prima tabella contiene l'intero annuncio pubblicato. Ha la stessa struttura della classe Annuncio, ma con l'attributo idAnnuncio in più che è la chiave primaria.

La seconda tabella presenta tutte le categorie presenti e che è possibile scegliere/selezionare durante la ricerca o pubblicazione di un annuncio.

La view è stata implementata semplicemente per avere una lista, ordinata per categoria, dei post più presenti nel DataBase. La view permette di determinare una minore complessità della query stessa.

### **File di Log remoto XML**

Viene inviato dalla classe LogXMLAttività al server il cui indirizzo e porta sono specificati nei parametri di configurazione locale. La riga di Log viene realizzata dalla classe stessa e si serializza tramite Xstream e poi si invia al server che la valida tramite lo schema presente nei file della

applicazione con nome SchemaAttivita.xsd.

Il Tag tipoAttività rappresenta un attributo. Per le regole di buona progettazione XML infatti gli eventi, che riguardano le attività in questo caso, appartengono ad un set limitato di possibilità e dunque devono essere impostati come attributi e non come elementi.

#### **File di configurazione locale XML**

Viene gestito dalla classe ParametriConfigurazione. Il file XML a cui fa riferimento la classe è denominato ParametriConfigurazione.xml.

Viene serializza o deserializzato dalla classe Xstream e successivamente validato tramite il file XMLSchema SchemaParametriConfigurazione.xsd.

Si interfaccia con le classi ConnessioneDB, Bacheca, LogXMLAttivita. Queste ultime ne estrapolano determinati parametri dopo essere deserializzati dalla classe XStream.