

Cognome e Nome		Matricola	
----------------	--	-----------	--

Esercizio 1

Si vuole realizzare un sistema di rete per la memorizzazione e la ricerca di file.

Il sistema è composto da tre gruppi di nodi:

- **n** StorageServer utilizzati per memorizzare i file.
- **1** IndexServer utilizzato per indicizzare i file sulla base dei loro attributi e per supportare la loro ricerca.
- **m** Client che possono memorizzare i file sugli StorageServer o cercarli utilizzando l'IndexServer.

Ogni *File* è caratterizzato dai seguenti attributi di tipo String: *filename* (nome del file scelto dal Client) ed una lista di *keywords* utilizzate per classificare il contenuto del file.

Ogni StorageServer rimane in ascolto sulla porta **TCP 2000** in attesa di connessioni da parte dei Client. Quando un Client si connette, invia allo StorageServer il file da memorizzare (da simulare con un oggetto String) e gli attributi necessari per descriverlo (*filename* e *keywords*).

L'IndexServer utilizza due porte e protocolli distinti per le sue funzioni:

- Porta **UDP 3000**, su cui riceve da un generico StorageServer **x** l'informazione che **x** memorizza un nuovo file. A tale scopo **x** invia mediante datagramma tutti gli attributi del file in modo da permetterne l'indicizzazione sull'IndexServer, il quale a tale scopo utilizza idonee strutture dati.
- Porta **TCP 4000**, su cui riceve da un generico Client **x** la richiesta di cercare tutti i file, indicizzati dall'IndexServer, che soddisfino i valori degli attributi ricevuti tramite il socket. Ad esempio, la ricerca con i parametri *filename*="song123", *keywords*=["classic", "2015"]> restituirà gli indirizzi IP degli StorageServer contenenti i file con nome "song123" e classificati con le keywords "classic" e "2015".

Si assuma che i Client conoscano tutti gli StorageServer e che scelgano di volta in volta e a caso quello su cui memorizzare un file.

Il programma deve comprendere un main in cui un Client memorizza un file su uno StorageServer; questo file viene successivamente cercato da un altro Client rivolgendosi all'IndexServer.

Esercizio 2

Si confrontino le **caratteristiche principali dei seguenti approcci peer-to-peer**: *con server centrale, basato su flooding, basato su DHT*.

Esercizio 3

Si descriva, anche mediante figure opportunamente commentate, come vengono effettuate la **connessione e la disconnessione nel protocollo TCP**.

Esercizio 4

Si realizzi un *Web Service* che permette di ottenere alcune informazioni sui voli di un aeroporto.

In particolare, il servizio espone:

1. un metodo che, dato il nome di una città **C** e una data **D**, restituisce il codice del primo volo in partenza per **C** in data **D**.
2. un metodo che, dato il codice di un volo **V** e una data **D**, restituisce l'orario di partenza di **V** in data **D**.

come specificato nel file WSDL allegato.

Si implementi in Java una classe che implementa il servizio e si descrivano le principali fasi necessarie alla sua messa in opera.

Allegato all'esercizio 4

```
<wsdl:definitions targetNamespace="http://www.examples.com/wsdl/AirportService">

  <wsdl:types>
    <schema targetNamespace="http://DefaultNamespace">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />

      <complexType name="Data">
        <sequence>
          <element name="Giorno" type="xsd:int"/>
          <element name="Mese" type="xsd:int"/>
          <element name="Anno" type="xsd:int"/>
        </sequence>
      </complexType>

      <complexType name="Orario">
        <sequence>
          <element name="Ore" type="xsd:int"/>
          <element name="Minuti" type="xsd:int"/>
        </sequence>
      </complexType>

    </schema>
  </wsdl:types>

  <wsdl:message name="PrimoVoloRequest">
    <wsdl:part name="in0" type="xsd:string"/>
    <wsdl:part name="in1" type="tnsl:Data"/>
  </message>

  <wsdl:message name="PrimoVoloResponse">
    <wsdl:part name="in0" type="xsd:string"/>
  </message>

  <wsdl:message name="OrarioVoloRequest">
    <wsdl:part name="in0" type="xsd:string"/>
    <wsdl:part name="in1" type="tnsl:Data"/>
  </message>

  <wsdl:message name="OrarioVoloResponse">
    <wsdl:part name="in0" type="tnsl:Orario"/>
  </message>

  <wsdl:portType name="AirportService">

    <wsdl:operation name="PrimoVolo" parameterOrder="in0 in1">
      <wsdl:input message="impl:PrimoVoloRequest" name=PrimoVoloRequest/>
      <wsdl:output message="impl:PrimoVoloResponse" name=PrimoVoloResponse/>
    </operation>

    <wsdl:operation name="OrarioVolo" parameterOrder="in0 in1">
      <wsdl:input message="impl:OrarioVoloRequest" name=OrarioVoloRequest/>
      <wsdl:output message="impl:OrarioVoloResponse" name=OrarioVoloResponse/>
    </operation>

  </wsdl:portType>

  <wsdl:binding ...>
    ...
  </wsdl:binding>

  <wsdl:service ...>
    ...
  </wsdl:service>

</wsdl:definitions>
```