



Magic Scraper

Magic Scraper

Ricerca carte di Magic
a partire da fonti
reperate in rete

Il progetto

- Fornire un motore di ricerca carte
- Sperimentazione metodologie agile per lo sviluppo in team
- Hosting online

Strumenti utilizzati

- Server:
 - PHP
 - Sqlite
- Client:
 - HTLM/CSS con Bootstrap
 - Javascript con JQuery
- Utility
 - Java

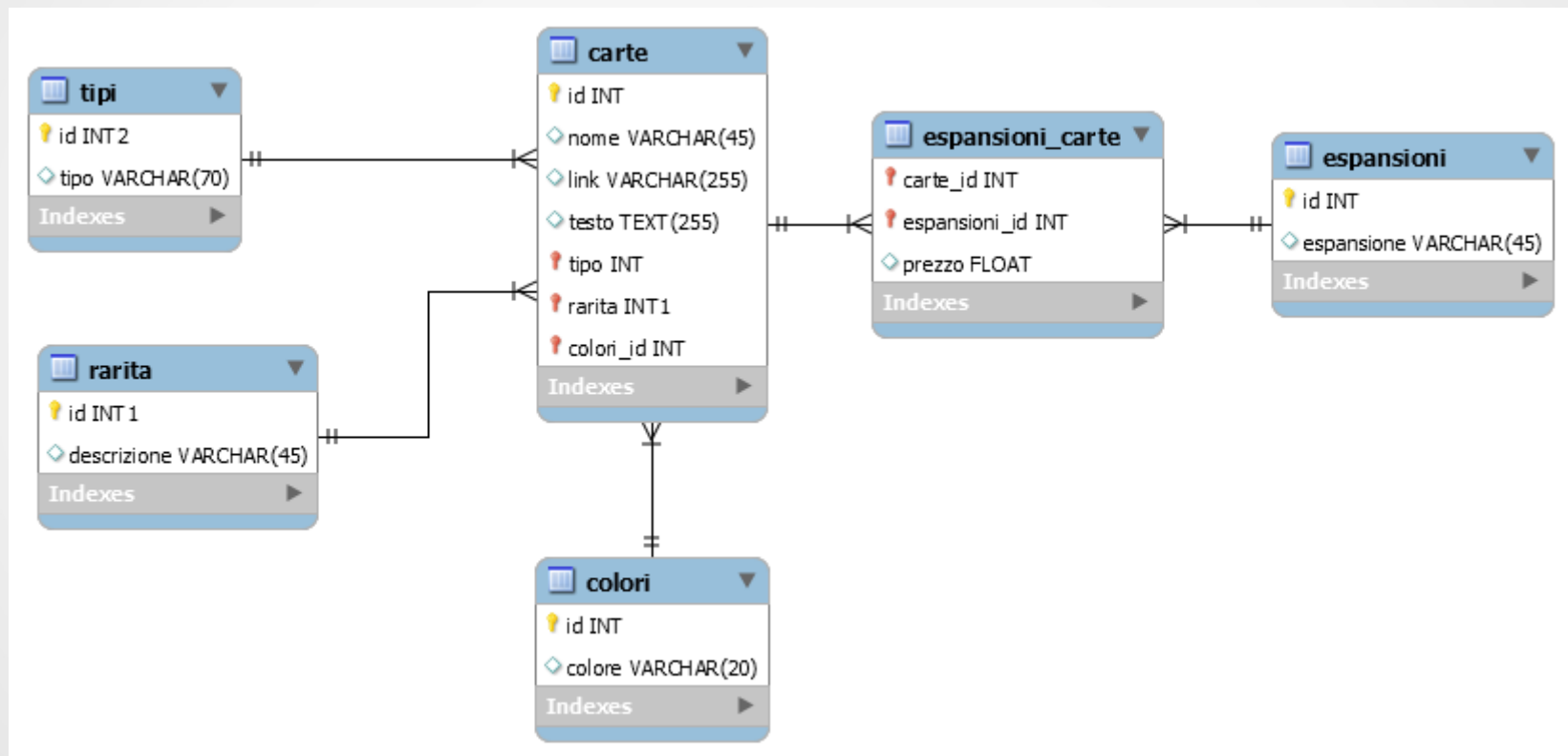
PHP

- Interfacciamento con il database mediante PDO
- Aggiornamento DB
 - Scaricare le carte
 - Controllo di versione tramite l'uso di espressioni regolari
 - Decomprimere i dati scaricati in formato zip

DB: SQLite

- Supporto MySQL limitato in free hosting
- Poter usufruire dello spazio residuo messo a disposizione per il codice
- È embedded

Database



Caricamento db

- L'inizializzazione del db viene gestito da un'utility Java

```
/** Carica una tabella del db composta da un attributo
il contenuto di un file csv ricevuto in input
@param tabella - la tabella da caricare
@param file - il file da cui leggere
*/
public void carica(String tabella, String file) throws SQLException, ClassNotFoundException, IOException {
    PreparedStatement stmSql = null;
    int i = 0;
    String s = null;

    /*Lettura del file contenente i dati da caricare*/
    fr = new FileReader(file);
    leggi = new BufferedReader(fr);

    /*Preparazione della query, successivamente faccio la bind dei parametri.
'conn' è un attributo della classe*/
    stmSql = conn.prepareStatement("INSERT INTO " + tabella + " VALUES (?, ?)");
    /*Fino a quando il file non termina carico il contenuto di ogni riga*/
    while((s=leggi.readLine()) != null){
        /*Il primo attributo è un auto-increment, pertanto lo valorizzo a null*/
        stmSql.setNull(1, Types.INTEGER);
        stmSql.setString(2, s);

        stmSql.executeUpdate();
        System.out.println(i++);
    }

    fr.close();
}
```


Query con PDO: connessione

- Il primo parametro è la stringa di connessione

```
try{
    $conn = new PDO(
        'sqlite:carte.db',
        null,
        null,
        array(PDO::ATTR_PERSISTENT => true)
    );

    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}catch(PDOException $ex){
    echo("Connessione fallita.".$ex->getMessage());
}
```

Query “dinamiche”

- Gestione modulare delle query
- Le query vengono assemblate in base ai filtri che l'utente vuole applicare alla ricerca



Sito Web

- Responsive design
- Flat design
- Utilizzo di bootstrap in alcune sue parti

Sito Web

- Sito Web suddiviso in 3 pagine principali (nav-page).
- È possibile navigarle con il menù (pg-load)

```
<body>

  <div id = "nav-main">
    <div id = "pg-load">

      <div id = "nav-page-1" class = "nav-page current-page">
      <div id = "nav-page-2" class = "nav-page">
      <div id = "nav-page-3" class = "nav-page">
    </div>
  </body>
```

Ricerca

- Gestione filtro di ricerca
- Mediante un menù è possibile selezionare il parametro di ricerca(\$filter)

```
$filter = "nome";

$(document).ready( function() {
    /* changing option menu header and variable filter */
    $('.dropdown-menu li a').on('click', function() {
        $("#"+$filter).removeClass(SELECTEDITEM);
        $(this).addClass(SELECTEDITEM);
        $filter = $(this).attr('id');
    });
});
```

- È possibile effettuare ricerche per colore, nome e tipo

Indirizzi utili

- Progetto su github:
<https://github.com/PeiPei97/ProgettoMagic>
- Hosting online:
<http://magicscrap.co.nf/>

Sviluppatori del progetto

Lerda Alberto
Delmastro Alessandro
Peirone Matteo
Bergerone Alberto
Cesana Davide
Albrile Erik
Sharma Vishesh