

# **An Open Source Solution for Education Management - EduXes**

**V Master on Free Software Projects Development and  
Management 2011-2012**

— Author —

— José Antonio Salgueiro Aquino <[info@joseantonio.org](mailto:info@joseantonio.org)> —

—Tutor—

Manuel Rego Casasnovas

December 23, 2012



# Copyright

Copyright ©2012 José Antonio Salgueiro Aquino. Some rights reserved. This work is free and is licensed under the conditions of Creative Commons Attribution - Share Alike 3.0 Unsupported license. You can use, distribute and reuse this work if the same license is applied and the author is quoted Full text of the license can be read on <http://creativecommons.org/licenses/by-sa/3.0/deed.en>





# Acknowledgments

Acknowledgments to my mentor, Manuel Rego. To Ramón Castro Pérez who sent me a patch to allow Siestta work. To José Manuel Ciges Regueiro, which memory I used as a template for my own work.



# Description of the practicum

The main objective of this Master Thesis consist in the development of an mobile application to be used int highschoools by teachers. It could allows teachers to carry on control students attendance, their behavior. Also it permits quick assessment by activity. Teachers would read students reports: weekly and daily assessment, by activity assessments and total marks.

- Name : José Antonio Salgueiro Aquino.
- Birth date: august 5th 1970
- Education: B.Sc. in Fundamental Physics, University of Santiago de Compostela University 1988-1993.
- Address: Marin (Pontevedra). Spain.
- Current position: Secondary School teacher in Technology.

Working times (planned) : 300 hours. From 6th August, to 30 September, on an eight hours day basis.

Technologies involved:

- Java <sup>TM</sup>language.
- Android <sup>TM</sup>. The operatig skeleton application.
- PhoneGap <sup>TM</sup>(alias Cordova ) framework to develop multi-platform applications.
- JQuery and JQueryMobile to development of mobile oriented applications.
- JavaScript with Web Databases.
- Git for version control system.
- L<sup>A</sup>T<sub>E</sub>X for documentation.

Meetings:

- Technologies to be used were stated, work methodologies, first application windows (pages), Android version to be used (2.3.3) because is the most popular.
- Several emails and gtalk conversations about organization, general problems were written.

Teleworking is carried on

Materials and special equipment used:

- Hardware: Intel Quad, 6GiB RAM, 500GiB HD.
- Software: Debian GNU/Linux Wheezy (testing), Eclipse Juno, JQuery 1.8.1, jQueryMobile 1.1.1, and PhoneGap-Cordova 1.8.1, Android Virtual Manager 2.3.3, Git 1.7.10.4-1.
- Real testing: Sony-Ericsson Xperia V mobile phone, with USB cable and wifi.



# Contents

Copyright	i
Acknowledgments	iii
Description of the practicum	v
<b>1 Introduction</b>	<b>1</b>
<b>2 Working plan</b>	<b>3</b>
2.1 Description and objectives . . . . .	3
2.2 Motivation . . . . .	6
2.3 Methodology . . . . .	7
2.3.1 HTML . . . . .	7
2.3.2 Interface code . . . . .	9
2.3.3 Database code . . . . .	11
2.3.4 Tools . . . . .	12
2.4 Work plan . . . . .	14
<b>3 State-of-art solutions</b>	<b>15</b>
3.1 Siestta . . . . .	15
3.2 Descripción del problema . . . . .	16
3.3 Descripción de los trabajos anteriores que se han dedicado a resolverlo . . .	17
<b>4 Objetivo (5%)</b>	<b>19</b>
4.1 Descripción, en un objetivo general, de la finalidad del proyecto. . . . .	19
4.2 Descripción de objetivos parciales que se necesitan cubrir para llegar al objetivo final . . . . .	19
4.3 Descripción de alto nivel de las etapas que sigues en el desarrollo . . . . .	19
<b>5 Descripción Informática (20-35%)</b>	<b>21</b>
5.1 La base de datos coleccionada (si tiene sentido). . . . .	21

5.2	Los algoritmos para el desarrollo de la solución . . . . .	21
5.3	qué quieres resolver . . . . .	21
5.4	cómo lo vas a hacer . . . . .	21
5.5	herramientas conceptuales necesarias . . . . .	21
5.6	herramientas utilizadas . . . . .	21
<b>6</b>	<b>Results</b>	<b>23</b>
6.1	Objectives completed . . . . .	23
6.2	Technical details . . . . .	23
6.3	Further objectives . . . . .	23
6.4	Resultados en forma de tablas, gráficas e imágenes donde se describa cuan- titativa y cualitativamente el funcionamiento de la aplicación . . . . .	24
6.5	Análisis crítico de los resultados con el objetivo de decidir si el sistema implementado es válido . . . . .	24
<b>7</b>	<b>Conclusiones y trabajos futuros (5%)</b>	<b>25</b>
7.1	Implementación . . . . .	25
7.2	Resultados . . . . .	25
	<b>Bibliography</b>	<b>27</b>

# List of Figures



# Chapter 1

## Introduction

In a high school, there are classes which attendance, assessment and group dynamics is difficult to control. This is especially relevant for the technology workshop, this could be a noisy, annoying, even dangerous place, which requires teacher's supervision. This workshop requires a lightweight, reliable and complete software tool. A simple web solution is not enough, and even use a tablet can be heavy, it requires a new approach.

The development of a multiplatform tool, open source, for smartphones, including tablets, for attendance monitoring and evaluation of students is suggested. To achieve this goal a client-side application will be developed using a multiplatform framework: Phonegap (Cordova) [Pho12b]. Phonegap allows you to develop utilizing well-known languages as Javascript and HTML. It permit us to deploy an application for both Android, WebOS, iOS and others.

For data management, a built-in database, SQLite, will be employed.

The tool that enables rapid development of the application and integration of performance tests is Eclipse. It uses the Android virtual machine (AVM).



# Chapter 2

## Working plan

### 2.1 Description and objectives

An open source multi-platform management application for high school teachers is developed. It can be run on a smartphone or tablet. The actual objectives of the application are students management:

- Attendance and punctuality control.
- Misbehaviour control.
- Activities assessment. Each student will have an activity mark.

Application should also include these features:

- Data visualization. As table-like format. Attendance and misbehaviour
- Server synchronization with a custom application or Xade [dEXdG12].

The final goal is to develop an application to make teacher's work easier and comfortable. This application is focused on user (teacher) experience. Also an objective is to write extensible, easy to read code, which allows external developers to take part into application development.

Task to be done:

- Study state-of-art solutions:

Find out other solutions: PDAs and smart-phone or tablet related and web-based applications.

Download to study and reuse graphical user interfaces, code or/and database structure.

- Develop database structure: tables (field names and type of data), and relationships among tables.
- Preparation for development:

Build development and staging environment:

Install Eclipse [Fou01], Android Virtual Machine [Inc01], Aptana Plugin for Eclipse, JQuery, JQueryMobile [Mob12a] and Phonegap [Pho12b] from their respective websites [Pho12a].

Choose application name and folder's policy.

Make a simple application: only a blank page.

Configure a git repository and upload the application: [Aqu12].

Development will follow these stages:

- Populate database with sample data. Firstly, data will be hard-coded into source code to tagging. Several tables will be created: groups, students, sessions, teacher schedule, students attendance, activities, student activities, and activities per group. Secondly the appropriate windows to manage these tables will be built.
- Groups: Several groups (four) of students will be hard-coded into javascript source code, with three or four students each other.

Make list of groups window. This will list the four groups.

Groups management window. Another group could be added, or removed.

- Students information:

Make list of students window per group and complete list of students.

Students management window: to insert and update data students: name, surname, birthday, address, e-mail, tutor name, landline and cell phone numbers and nationality.

- Sessions. Each lecture has a description (as 'first hour', or 'recreation'), starting and ending time. These sessions will be hard-coded on first version.
- Teacher schedule. For the current teacher, it contains weekly and daily schedule: name of group, session and day of the week. This information will be also hard-coded.



- Main window (or *page* in PhoneGap notation) will be created. Teacher can set current data and go to *daily page*, manage activities, students and groups or manage reports: assessment and attendance. Next page will be *daily page*.
- Timetable for current date: daily schedule, list of groups for each day ordered by session.
- Attendance page will be next to be build. It contains a list of students by group. Teacher can assign an attendance code to each student (attendance, misbehaviour, unpunctuality or excused).
- Assessment page is next to previous one. It contains one upper list of activities and a list of students similar to *attendance page* . Teacher could set activity and assign adequate mark to each student.
- Activities. This page manage activities (add and disable activities <sup>1</sup>): name and description of assessable exercises.

Activities group will be set in activities page, and activities student will be related with assessment.

- Reports pages.

List activities.

List groups.

List students by group.

List students attendance by week. User can select previous or next week, or set another date.

List students marks and final mark.

- Error handling. While developing each SQL query, any error will be catched and an error window will appears.
- Eventually, test into real hardware: an Android 2.3.3 mobile phone, tablets, and so on.

Next steps will include :

- Load data from an external file. This is well-documented [].
- Load images from disk (SD-Card).

---

<sup>1</sup>still not coded

- Save or download data from database to disk.
- Xade web interface.
  - a) Study Xade web interface.
  - b) Develop an ad-hoc application for retrieve Xade's data. With javascript or a native one.
- 8. Develop an ad-hoc application for store data.
- 9. Synchronization with a custom server or with Xade.
- Test units. To ensure previous step were implemented.
- User documentation. Manual with images.
- Developer's documentation.
- Find out a website to host a forum, a bug report system, documentation and application download.

In the following table a broad estimation of time spent in each task are shown.

Task	Time (hours)
State-of-art solutions	10
Develop database	8
Preparation for development	40
Development.	
Populate database with sample data	20
Groups. List and management	50
Students. List and management	30
Timetable for actual date	80
Add attendance, behaviour	50
Add error handling	2
Retrieve and insert data from and to database	30
List of attendance, misbehaviour incidents.	12
Add activities grades for each student.	12
List students marks and final mark.	14
Activities management window	14
Management of student notes.	20
List of student notes.	2
Test into real hardware	20
Total	384

## 2.2 Motivation

As a Technologies teacher, in my daily work I have to evaluate students work such as working with tools, cooperative work with other classmates etc., besides usual activities

as written exercises. A long data sheet could be used, or an awkward long spreadsheet, but a portable device with a custom application should be desirable.

This application tries to increase teacher's productivity because teacher only has to write attendance, or unpunctuality two times (on official report and on application's window), and classroom notes and activity grades on very easy way.

The most important feature is to be as easy, fast and intuitive as possible. It could be desirable to be platform independent (Android, iOS, Windows RT), but Android is preferred because it is open source and has a higher market share.

On the other hand, development of this application improves my computer science skills in mobile-phone applications development: JQuery, jQueryMobile, PhoneGap/Cordova, SQLite, Android, git repository management.

## 2.3 Methodology

This work was carried on building little blocks, also called pages, and make it up into final application. Database structure was separated from interface, and interface was also separated into dynamic and static contents. Each new functionality was written, tested, and polished. Each new function was written from previous one, and so on.

### 2.3.1 HTML

There are only two html files: **index.html** and **remove.html**. The most important: **index.html** file is build by blocks called pages, gathered together [Mob12b]. Each page is its own **div** with custom properties (*propieraty* properties in Eclipse jargon) : **data-role="page"**. Below as show an example: list of several reports, user could choose one and a function (*onReportListAttendance()* or *onReportListAssessment()*) is executed .

Inside each page there are several identificatives *id=name* which are used by application to fill with data (e.g. within *daily\_schedule* page *current\_day* id is used to set date to current date).



## Reports Page

```

<div data-role="page" id="list\_reports" data-add-back-btn="false">
  <div data-role="header" data-back-btn-text="previous"
    data-add-back-btn="false" >
    <a href="#daily\_work" data-icon="arrow-l" data-theme="a"
      data-role="button">Back</a>

    <h1>Reports</h1>
  </div>
  <div data-role="content">

    <ul data-role="listview" id="list\_reports\_ul"
      data-split-icon="gear" data-split-theme="a"
      data-filter="false"
      data-inset="true" data-theme="a" >
      <li><a href="#" onClick="onReportListAttendance();"
        >Attendance</a></li>
      <li><a href="#" onClick="onReportListAssessment();"
        >Assessment</a></li>
      <li><a href="#" onClick="" >File</a></li>
    </ul>
  </div>
  <div data-role="footer" class="footer-docs" data-rel="back"
    data-theme="a">
  </div>
</div>

```

A complete list of pages is shown below:



## List of Pages

intro	Introductiony page
daily_work	daily work
list_reports	List reports
list_settings	List settings
file	Exit application
daily_schedule	List of groups per day
list_students_attendance	
list_students_assessment	
list_students_attendance_by_group	
general_list_attendance	
list_groups_attendance	
list_groups_assessment	
list_students_assessment_reports	
list_groupsto_edit	
edit_new_group	
edit_groups	
list_groups	
list_students_by_group	
list_students	
edit_student	
edit_new_student	
list_all_activities	
update_activity	

### 2.3.2 Interface code

There is only one file which deals with interface interactions (events from **html** files): *interface.js*. If the function is called directly from *html* code, function name contains **on** prefix. There are several groups of functions:

- *Students* and *Groups* functions, which manage and list general students and groups information.
- *Activities* functions, which update, list and add new activities.
- *Assessment* functions, list and update students marks.
- *Attendance* functions, list when student attend classes and update their values.

Whether these functions need access to data (every function but *onGeneral\** ) they call their counterpart function in **database.js** file.



## List of Functions: interface.js

Task	Name	Group
Firstly loaded, create and populate database, load initial page	<i>onDeviceReady()</i>	Initial
Initial	<i>init()</i>	
Load default date	<i>initialize_data()</i>	Initial
Daily work page: list of groups	<i>open_daily_page()</i>	Schedule
Students Attendance page	<i>listStudentsAttendance()</i>	
New Group	<i>requestNewGroup()</i>	Groups
Edit Group	<i>onAddNewGroup()</i>	Groups
Update Group	<i>onAddNewGroup()</i>	Groups
Delete Group	<i>onDeleteGroup()</i>	Groups
Save Group	<i>onSaveNewGroup()</i>	Groups
List All Groups	<i>onListAllGroups()</i>	Groups
Edit Student	<i>EditStudent()</i>	Students
Delete Student	<i>onDeleteStudent()</i>	Students
Add new Student	<i>onAddNewStudent()</i>	Students
Save Student	<i>onSaveStudent()</i>	Students
Save New Student	<i>onSaveNewStudent()</i>	Students
Full list of students	<i>listStudents()</i>	Students
List Students by group	<i>listStudentsByGroup()</i>	Students
Full list of students	<i>onListAllStudents()</i>	Students
List All Activities	<i>onListAllActivities()</i>	Activities
Activity update	<i>onUpdateActivity()</i>	Activities
Add new Activity	<i>onAddNewActivity()</i>	Activities
Save New Activity	<i>onSaveNewActivity()</i>	Activities
Open Students Assessment	<i>onOpenStudentsAssessment()</i>	Assessment
Reload Group Assessment	<i>onRefreshGroupAssessment()</i>	Assessment
Open Student Attendance	<i>onOpenStudentsAttendance()</i>	Attendance
List Attendance	<i>onReportListAttendance()</i>	Attendance Reports
List Students by Group Attendance	<i>listStudentsByGroupAttendance()</i>	Attendance Reports
List Students by Group Previous week	<i>studentsAttendanceListPrevious()</i>	Attendance Reports
List Students by Group Next week	<i>studentsAttendanceListNext()</i>	Attendance Reports
List Groups for Assessment	<i>onReportListAssessment()</i>	Assessment Reports
List Students for Assessment	<i>onListStudentsAssessment()</i>	Assessment Reports
Student State	<i>studentState()</i>	Student
Student Attendance	<i>Attendance()</i>	Student Attendance
Exit	<i>onGeneralFile()</i>	Exit
List Reports	<i>onGeneralListReports()</i>	Reports
List Settings	<i>onGeneralListSettings()</i>	Settings

### 2.3.3 Database code

The file which contains functions which carry on queries and data manipulation (on data base (SQLite)) is : *database.js*

There are global variables on top of this file. These variables are used by functions because there are not obvious ways to pass values from **html** code through functions ( e.g. as day week, database, current date, ... ):

Variable	Meaning
global_id	General and unique identification of any table as students or groups <sup>a</sup>
table_global	students, groups table
global_id_group	Identification of a group
global_id_student	Identification of a student
global_id_activity	Identification of an activity
global_max_activities	Number of activities
global_no_groups	Number of groups
global_week_day	Number of week day (0-6)
global_db	Database pointer
global_session	Selected session
global_actual_date	Current date
global_reports_date	Date for reports
global_is_new	Whether activity is new
global_exist	if exist current record
STATE_NONE	Default Student state
STATE_ABSENCE	Truancy
STATE_UNPUNCTUAL	Unpunctuality
STATE_EXCUSED	Excused unattendance
STATE_BEHAVIOR	Bad behaviour

<sup>a</sup>will be deprecated

Several approaches have been considered, a lot of tiny, one function equals one simple task, but code were growing in complexity, became very difficult to read, maintain and pass variables to them. Other perspective was to write only several complete functions which carry all (or almost all) the work. From this point of view functions become longer, (up to a hundred of lines of code) but easier to test and follow.



#### List of Functions: database.js

Task	Name	Group
Firstly loaded, create and	<i>onDeviceReady()</i>	Initial

### 2.3.4 Tools

Tools involved were Eclipse IDE (with plug-ins) and Android Virtual Manager (AVM) on Debian GNU/Linux Wheezy. When a new functionality was developed, application was tested on AVM, if it worked, source code was polished, applicable was tested again, if it was satisfactory a new change was committed into git repository. From time to time application was downloaded from git repository into a real hardware and tested.





## Main Page

```

<div data-role="page" id="daily\_work">
  <div data-role="header" data-add-back-btn="true" data-theme="a">
    <h1>Day</h1>
  </div>
  <div data-role="content" data-theme="a">
    <div class="ui-grid-b">
      <div class="ui-block-a">
        <div data-role="fieldcontain">
          <label for="date" >Date:</label>
        </div>
      </div>
      <div class="ui-block-b">
        <input id="daily\_date\_scroller"
          name="daily\_date\_scroller" />
      </div>
      <div class="ui-block-c">
        <a href="#" data-role='button'
          data-icon='info' data-iconpos='notext'
          style='float: right;'
          onClick="help('date');" >Help</a>
        <a href="#" data-role="button"
          data-icon="arrow-r" data-iconpos="notext"
          onclick="open\_daily\_page() " >Go</a>
      </div>
    </div>
  </div>
  <div data-role="navbar">
    <ul>
      <li>
        <a href="#" onClick="onGeneralFile()"
          data-role="button" data-icon="star"
          data-theme="a" > File</a>
      </li>
      <li>
        <a href="#" onClick="onGeneralListReports();"
          data-role="button" data-icon="grid"
          data-theme="a" >Reports</a>
      </li>
      <li>
        <a href="#" onClick="onGeneralListSettings();"
          data-role="button" data-icon="gear"
          data-theme="a" >Settings</a>
      </li>
    </ul>
  </div><!-- /navbar -->
  <div data-role="footer" class="footer-docs" data-theme="a">
    <p style="text-align:center;" id="teachers\_name"></p>
  </div>
</div>

```

## 2.4 Work plan

Several problems were faced: Eclipse environment: A stable, reliable and up-to-date IDE, with several plug-ins is needed. Download vanilla Eclipse Juno from its web-site is chosen because it is more stable, reliable, compatible with newer versions. Aptana Javascript plugin was chosen because Aptana allows source code auto-completion in JQuery. PhoneGap and Android incompatibilities. Android 2.3.3 requires JQuery-1.8.1 and does not work on higher versions. Error handlers. I have had several problems with `tx.executeSql(...)` function, it confused me with `db.transaction(...)`: `tx.executeSql(sql, [parameters], successHandler, errorHandler)` and `db.transaction(queryFunction, errorHandler, successHandler)` have up to four and three parameters respectively, only first one is mandatory. I rather use success and error handlers for `tx.executeSql` function, atomic error control could be better choice. Passing variables to functions: Only if another solution is not known or feasible, global variables are used: named after `global_`, and in block capitals. Deadline. Development was delayed because I have no enough spare time and above problems were time costly.

# Chapter 3

## State-of-art solutions

Only an open source application was found for study, Siestta , nevertheless there are a lot of educational software (Sixa2, Unisoft3) but they are privative, Microsoft Windows freeware or both (SAS académico4). Siestta was evaluated. Technically it is an GPL'ed old style PHP-based web application with Ajax, an interactive editor, fckeditor and fpdf to generate reports. From user point-of-view there are online documentation5. This application includes management of students, attendance, marks, tasks, incidents, general queries, letters to parents, interviews with parents, messages, appointments, exams and more. Several screen-shots were taken and will be reused in current application:

### 3.1 Siestta

This application (Siestta) are also available for PDAs, it could be a valid solution but it is server-side with outdated technologies. Data structure from Siestta is standard and fully functional, and it could be partially reused by EduXes. Source code are also shown: calendario.php. It shows us a PHP application which uses sessions variables and is not Model-View-Controller oriented.

All parts of this prealgebra textbook are copyrighted © 2009 in the name Department of Mathematics, College of the Redwoods. They are not in the public domain. However, they are being made available free for use in educational institutions. This offer does not extend to any application that is made for profit. Users who have such applications in mind should contact David Arnold or Bruce Wagner at [hidden email] or [hidden email].

text text

```
<?php
session_start();
require('config.php');
require('idioma/' . $idioma . '');
```

```

include('funciones_calendario.php');
$docente = $_SESSION['usuario_sesion'];
//recogemos variables
$mes_actual = $_POST['mes'];
$anyo_actual = $_POST['anyo'];
if($mes_actual || $anyo_actual) {
include('funciones.php');
conecta();
}
//si es la primera vez que entramos, cargamos la fecha actual
if(!isset($mes_actual)) $mes_actual = date('m');
if(!isset($anyo_actual)) $anyo_actual = date('Y');
//presentamos ahora el calendario del mes actual o cargado
//tabla con nombre mes y año y las flechas para navegar
echo '
<br />
<table class="tablacentrada_i">
<tr>
<td>
<a href="#" onclick="navegaMes(\'\'.$mes_actual.\'\' ,\'\'\'.$anyo_actual.
\'\' ,\'\'menos\')" title="\'\'.$id_anterior.\'\'"></a>
\'';
$nombre_mes = numero_mes_a_nombre($mes_actual);

```

Develop database structure: tables and relationships. Data base structure looks like  
Illustration 5: EduXes Database structure

Basándose en artículos, libros, etc. que se os haya facilitado y de otros que estiméis oportuno, se hablará de:

## 3.2 Descripción del problema

...

### **3.3 Descripción de los trabajos anteriores que se han dedicado a resolverlo**

...



## Chapter 4

### Objetivo (5%)

4.1 Descripción, en un objetivo general, de la finalidad del proyecto.

...

4.2 Descripción de objetivos parciales que se necesitan cubrir para llegar al objetivo final

...

4.3 Descripción de alto nivel de las etapas que sigues en el desarrollo

...





# Chapter 5

## Descripción Informática (20-35%)

Para ello describirás:

**5.1 La base de datos coleccionada (si tiene sentido).**

...

**5.2 Los algoritmos para el desarrollo de la solución**

...

**5.3 qué quieres resolver**

...

**5.4 cómo lo vas a hacer**

...

**5.5 herramientas conceptuales necesarias**

...

**5.6 herramientas utilizadas**

...

ASD
-----

FFG

OVAL BOX

OVAL BOX shadowblock



**Contenido**

| , contenido ...

# Chapter 6

## Results

Application is evolving from list, edit students and groups, to its final goals. These objectives were fulfilled:

### 6.1 Objectives completed

- Access to any workday, any group and student.
- Management of attendance and misbehaviour of each student. The students information is still hard-coded into source files.
- 

### 6.2 Technical details

### 6.3 Further objectives

- Links to student and group management. These pages were done but links are not missing into main application window.
- There are several objectives not fulfilled yet, but I am on the way to get those done, those are, in priority order:
- Data visualization. Student attendance and misbehaviour have to be shown in table-like window.
- Test into real hardware. EduXes.apk has to be copied into mobile phone.
- Activities evaluation per student. A window to display activities marks and final mark.

- Timetable management. A window to manage groups timetable. When a group has class with this teacher.
- Server synchronization with a custom application or

These objectives were not fulfilled because time and skills lack.

En este apartado deberán quedar reflejados los experimentos realizados. Para ello se mostrarán:

## 6.4 Resultados en forma de tablas, gráficas e imágenes donde se describa cuantitativa y cualitativamente el funcionamiento de la aplicación

...

## 6.5 Análisis crítico de los resultados con el objetivo de decidir si el sistema implementado es válido

...

```
ohcount -i assets/www/js/interface.js assets/www/js/database.js assets/www/js/create.js
Examining 5 file(s)
```

Ohloh Line Count						
Language	Code	Comment	Comment %	Blank	Total	File
-----	-----	-----	-----	-----	-----	-----
javascript	393	93	19.1%	54	540	interface.js
javascript	1462	187	11.3%	226	1875	database.js
javascript	402	57	12.4%	46	505	create\_popu
html	546	99	15.3%	117	762	index.html
javascript	1	0	0.0%	0	1	index.html
html	28	1	3.4%	8	37	remove.html

# Chapter 7

## Conclusiones y trabajos futuros (5%)

Resumen de los logros principales conseguidos, destacando:

### 7.1 Implementación

...

### 7.2 Resultados

...

En futuros trabajos, a partir de una crítica constructiva del trabajo realizado, plantear mejoras y extensiones del mismo.



# Bibliography

- [Aqu12] José Antonio Salgueiro Aquino. Classroom management for high school teachers. <https://github.com/joseantoniosa/EduXes>, 2012.
- [dEXdG12] Consellería de Educación. Xunta de Galicia. Xade web is a web interface for educational data management. <http://edu.xunta.es/xade/>, 2012.
- [Fou01] Eclipse Foundation. Eclipse interface development environment. <http://www.eclipse.org/>, 2001.
- [Inc01] Android Inc. Android simple development kit. <http://developer.android.com/sdk/index.html>, 2001.
- [Mob12a] JQuery Mobile. JQuery mobile. <http://jquerymobile.com/demos/1.1.1/>, 2012.
- [Mob12b] JQuery Mobile. The jquery mobile "page" structure. <http://jquerymobile.com/demos/1.1.1/docs/pages/page-anatomy.html>, 2012.
- [Pho12a] PhoneGap. Phonegap getting started with android. [http://docs.phonegap.com/en/1.8.1/guide\\_getting-started\\_android\\_index.md.html#Getting%20Started%20with%20Android](http://docs.phonegap.com/en/1.8.1/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android), 2012.
- [Pho12b] PhoneGap. Phonegap is a free and open source framework that allows you to create mobile apps using standardized web apis for the platforms you care about. <http://www.phonegap.com>, 2012.

Lo relativo al anexo 1 ...

Ejemplos de tratamiento de texto:

Probamos una cita [?]

Citamos un libro[?]

Probamos a poner una nota al pié<sup>1</sup>

Probando: *cursiva* **negrita** subrayada enfatizar

*Tools used: Sqlfairy. Tranforms SQL language into a png image. LibreOffice 3.5.4.2 to write this document. 1 Gimp 2.8.2 to get screen-shots. GNU/ Debian Wheeze October 2012*

---

<sup>1</sup>Mi primera nota al pié