TITLE: E-Administration: Enterprise Administration applications for the “Síndic de Greuges de Catalunya” using RIA with COM+ technologies

MASTER: Master in Science in Telecommunication Engineering & Management (MASTEAM)

AUTHOR: Jordi Clemente Pascual

DIRECTOR: Roc Meseguer Pallares

SUPERVISOR: Joaquim Lahoz Valls

DATE: 20th of June 2009
Overview

With this project, I have created a set of tools for the Síndic de Greuges de Catalunya taking advantage of the existing infrastructure with the aim of improving communication among workers. My objective has been facilitating the administrative paperwork the workers have to carry out, as well as allowing for a dynamic communication by means of which everybody feels involved, meeting the needs of workers, the administration and the management. Amongst others, the technologies that have been used are:

- HTML, It's the marking-up language web pages are written with.
- CSS, They are cascading style sheets, which describe us how a document is going to appear on the screen or how it’s going to be printed.
- JAVASCRIPT, It's mainly used to improve the management between the client/server interface.
- COM Objects for the transactional management of the applications core from the C++ language.
- XML/XSL Technology to convey the search results and the application construction.
- SQL Structured search language, for the access to relational databases.
- ASP It's mainly used to make dynamic HTML pages in an easy way.
- AJAX as a RIA technology which brings closer the application to the desktop with a much quicker application.
- IIS Server Software, COM+ Server, SQL Server, ...

The objectives of the implementation have been met, thus guaranteeing:

- Taking advantage of the administration infrastructure without having to acquire any type of device.
- That the application is very close to the use of the desktop (RIA) in order to ease the learning of the application by the workers.
- Offering from the applications a virtual attention to workers and the possibility of carrying out online procedures which are related to each worker.
- Centralizing and making more accessible the information available at the administration.
- Automating the incident management, holidays, material, courses, library work, and facilitating the fact that the "user is always informed about the state of his operations".
- Accumulating the knowledge derived from the process management with the aim of improving the information on an ongoing basis.
- Reduce the use of paper in the process.
Dedication

This project, as a final symbol of my telecommunications studies, is dedicated to those persons who have been by my side these latest years in spite of the little time I have been able to devote to them: my parents.

I’d like to mention that it has been a great sacrifice for me combining work and study, but all the suffering, stress and pressure have been worth it because I think I have been trained not only on the theoretical knowledge, which is a good base for the future, but also on facing difficulties, achieving self-confidence, and the results of efforts and work. Everything I think should define an engineer.

I would also like to thank the professors I have had during the degree that have been the final authors of what I am now as an engineer.

This dedication, which I did at the beginning of the project, ended up saying: “Above all this is the best end for my studies, not for the training received, but for my family being by my side with health”.

Unfortunately this is not so anymore because I have lost a part of my life, because without you dad a piece of my heart has gone for good. Everything is thanks to you, for without you I would not be here and much less finishing the project. I love you Dad. Thank you for all you have given me without asking for anything in return.

It’s not much but this work is dedicated to you. You are the best father I could have had.

Thank you Dad.
INDEX OF FIGURES

Fig. 1.1 HP Proliant DL360 Server ......................................................................................... 5
Fig. 1.2 HP Compaq dx 6100 Client's computer ..................................................................... 6
Fig. 2.3 Screenshot of the Education department of Catalonia intranet .................................. 9
Fig. 3.4 Screenshot of the Component Servicies ................................................................. 20
Fig. 3.5 Screenshot of the Component Servicies Properties ............................................... 21
Fig. 3.6 Ajax working diagram ............................................................................................. 28
Fig. 3.7 Screenshot of a step by step Ajax engine-client-server communication ................... 28
Fig. 3.8 Screenshot of Sql Server management Studio view ............................................... 30
Fig. 3.9 Screenshot of the Com+ viewer .............................................................................. 31
Fig. 3.10 Screenshot of the SAP search application .............................................................. 32
Fig. 3.11 MsExcel Medics revision ......................................................................................... 32
Fig. 3.12 MsAcces library application .................................................................................... 33
Fig. 3.13 Ultraedit view with color markup .......................................................................... 34
Fig. 3.14 Screenshot of ultraedit’s selection tool ................................................................. 34
Fig. 3.15 View of the Visual Studio C++ 2005 ..................................................................... 35
Fig. 4.16 Home screen of the Internet Information Server ................................................ 37
Fig. 4.17 Security configuration of the IIS (Internet Information Server) ............................. 38
Fig. 4.18 Computer of the HDX-9450ES production environment .................................. 38
Fig. 4.19 First chart for the application building ................................................................. 39
Fig. 4.20 Second chart of the application building ............................................................. 39
Fig. 4.21 Administrator’s screen of the incidence application ............................................. 40
Fig. 4.22 Administrator’s screen of the incidence application ............................................. 40
Fig. 4.23 Screen of obtained records filtering ....................................................................... 41
Fig. 4.24 Incidence Tables, Estats_inc, Personal ................................................................. 41
Fig. 4.25 Screenshot of the intranet homepage header ....................................................... 42
Fig. 4.26 Screenshoot of the calling to annex files ............................................................... 42
Fig. 4.27 Screenshoot of the intranet checking each of the containers ................................. 45
Fig. 4.28 Quick search engine of the persons directory ..................................................... 45
Fig. 4.29 Advanced search engine of the persons directory ................................................. 45
Fig. 4.30 Advanced search engine of the persons directory for the application administrators 46
INDEX OF TABLES

Table 1.1 Different sections of the: Síndic de Greuges de Catalunya. ..................................................4
Table 1.2 Hardware summary of the organization. ..................................................................................4
Table 1.3 Hardware summary of the HP trademark Server ....................................................................4
Table 1.4 Features summary of the HP trademark Server. ....................................................................5
Table 1.5 Features summary of the HP dx6100. ....................................................................................6
INTRODUCTION

Motivation

Why this project and not another? Well, because I find myself working at the company and I have realised that I like very much the business tool programming for I think the future of business applications is under the web domain. I have many reasons, but these are the main ones: that everyone has already the web compiler, the load and response time is very reduced, people have already a good training, the amount of existing technologies on this domain, an a great many others.

In summary, my main motivation is that this is a way of opening up to a field which during my degree has not had all the specific weight I would have expected it to have.

On the other hand, the fact of studying the subject in more depth, helps me to gain a new insight and at the same time it helps me enlarge the knowledge I have already acquired.

Main goals

From a given infrastructure being able of reducing and improving the response time to the minimum possible.
The union of two technologies quite separated regarding the time, but with a logical unification. These are the COM (Component Object Model) technology by Microsoft 1993 with RIA (AJAX in our case) 2002.

Planning of the project

My proposal of the project planning starts from 5 phases:

Phase 1:

Meetings will be called with the officers and the persons responsible for the sections and at the same time a research will be carried out on the needs and the software used on the bulk of clerk and organizational tasks based on Sharepoint, MS Acces, MS Excel, with the disadvantage of only one person visualizing in a single time space documents of equal nature.

The research of the company's technical infrastructure consists of the setting up of the servers, the services configuration, the performance of the developed applications and their behaviour within the company's structure.

Phase 2:

The learning of the company's current programs structure is established. Both its programming internally as the load and circuits of the programs from the
moment of the client’s demand going through the server stages until the response arrives to the client himself.
The decision-taking of the applications regarding both the subject of the integration of the validation with the server in an automatic way as the introduction of the RIA technology into the applications of the DLL system. Study and decision of a better image so the application looks kinder both for administrator and client users, differentiating each one of the profiles. Personalization of the applications for the user, etc.

Study of difficulties to overcome them. Such as the case of the union of the RIA technology with COM.

Phase 3:
The study of the AJAX technology is carried out as well as its integration in the company’s setting, and next key examples are created in order to facilitate the different sections of the applications.

Phase 4:
The environment is prepared in order to carry out the environment tests. Development of the set of applications for the company, by using all the examples and knowledge acquired on previous phases.

Phase 5:
Taking into account that everything has been programmed by means of testing, we will give way to the application which has to be integrated in the company’s servers structure. Starting with the Databases, etc. The conclusions of the integration and the project could have likely been better and with more recommended options had I not had to start from a particular infrastructure that has restricted me.

**Methodology of the work**

During the 6 months of length of the process I have carried out a series of tasks depending on the situation I found myself in, and some of them have been more appropriate than others. I have to point out, that before this 6-month period, I had dedicated some time to the study of the programming languages used in the information search when finding those programs that let us carry out our work more efficiently, etc. And this period of time which is not counted in here, is equally or more important than the later 6 months, since without great part of the knowledge acquired, it would have been very difficult for me to undertake the tasks I have carried out.

I can differentiate the different phases or periods within the 6 months of the process, which we’ll find on the chapter dedicated to the temporal division.
Temporal division

Calendar of the project.

- Meetings. Study of the needs and Infrastructure.
- Study of the technologies of the company, characteristics for the planning of the project.
- Study of the new tools that I will need for the development of the project.
- Development of the project.
- Restep, Completion I make up of the memory in Catalan and English.
- Preparation of the written as well as oral Presentation.

<table>
<thead>
<tr>
<th>March</th>
<th>February</th>
<th>April</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3</td>
<td>1 2</td>
</tr>
<tr>
<td>4 5 6 7 8 9 10</td>
<td>3 4 5 6 7 8 9</td>
<td>12 13</td>
</tr>
<tr>
<td>11 12 13 14 15 16 17</td>
<td>10 11 12 13 14 15 16</td>
<td>14 15 16 17 18 19 20</td>
</tr>
<tr>
<td>18 19 20 21 22 23 24</td>
<td>17 18 19 20 21 22 23</td>
<td>21 22 23 24 25 26 27</td>
</tr>
<tr>
<td>25 26 27 28 29</td>
<td>24 25 26 27 28 29 30</td>
<td>28 29 30 31</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>May</th>
<th>June</th>
<th>July</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2</td>
<td>3 4</td>
<td>1 1</td>
</tr>
<tr>
<td>5 6 7 8 9 10 11</td>
<td>2 3 4 5 6 7 8</td>
<td>7 8 9 10 11 12 13</td>
</tr>
<tr>
<td>12 13 14 15 16 17 18</td>
<td>9 10 11 12 13 14 15</td>
<td>14 15 16 17 18 19 20</td>
</tr>
<tr>
<td>19 20 21 22 23 24 25</td>
<td>16 17 18 19 20 21 22</td>
<td>21 22 23 24 25 26 27</td>
</tr>
<tr>
<td>26 27 28 29 30</td>
<td>23 24 25 26 27 28 29</td>
<td>28 29 30 31</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>30</td>
</tr>
</tbody>
</table>
CHAPTER 1. PROJECT PRESENTATION

1.1. Company

1.1.1. Introduction of the Company

The Síndic has the function of addressing the complaints of all the people who are unprotected before the lack of action of the administrations. It sees to the good performance of the Administration of la Generalitat and the establishments in Catalonia, such as townhalls and provincial and regional councils. Therefore, it acts as a supervisor and collaborator of the Catalan Administration, with the aim of helping improve its performance.

Elected by the majority vote of the Catalan Parliament, it is politically independent, for it does not depend on any government and it acts with objectivity, freedom of opinion and independence.

Table 1.1 Different sections of the: Síndic de Greuges de Catalunya.

<table>
<thead>
<tr>
<th>SECTIONS</th>
<th>Immigration</th>
<th>Citizen participation</th>
<th>Public safety and Justice</th>
<th>Town planning and Housing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Immigration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Citizen participation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Public safety and Justice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Town planning and Housing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consumption; Culture</td>
<td>Childcare</td>
<td>Labour relations</td>
<td>Social Services</td>
<td></td>
</tr>
<tr>
<td>Education</td>
<td>Environment</td>
<td>Health care</td>
<td>Taxes</td>
<td></td>
</tr>
</tbody>
</table>

On table 1.2 we find a small summary of all the hardware to have an idea of the software infrastructure we carry with ourselves and which will have to support my solution.

Table 1.2 Summary of the organization's hardware.

<table>
<thead>
<tr>
<th>SUMMARY</th>
<th>NUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVER PC</td>
<td>10</td>
</tr>
<tr>
<td>DESKTOP PC</td>
<td>86</td>
</tr>
<tr>
<td>LABTOP PC</td>
<td>21</td>
</tr>
<tr>
<td>RECORDERS</td>
<td>4</td>
</tr>
<tr>
<td>LABELLERS</td>
<td>3</td>
</tr>
<tr>
<td>SCANNERS</td>
<td>5</td>
</tr>
<tr>
<td>PRINTERS</td>
<td>40</td>
</tr>
<tr>
<td>PHOTOCOPIERS</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1.3 Summary of the hardware for the HP trademark Server.

<table>
<thead>
<tr>
<th>NAME</th>
<th>TRAKEYMARK</th>
<th>MODEL</th>
<th>FEATURES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of a typical home server:

![HP Proliant DL360 Server](image)

**Fig. 1.1** HP Proliant DL360 Server.

On the following table we point out the main features.

**Table 1.4** Summary of the features of the HP trademark Server.

| Processor, operating system and memory Type of processor
| Intel® Xeon® X5260 Dual Core a 3,33 GHz Processor |
| Processor speed |
| 3,33 GHz |
| Number of processors |
| 1 processor |
| Available processor core |
| Dual |
| Internal cache memory |
| 6 MB (1 x 6 MB) of level 2 cache |
| Chipset |
| Intel® 5000P Chipset |
| Type of memory |
| DIMMs PC2-5300 with complete intermediate memory |
| Default memory |
| 4 GB (2 x 1 GB) of standard memory |

An example of a typical client PC of the organization would the following figure.
On the following table we can see the main features of the typical client computer.

**Table 1.5** Summary of the features of the HP dx6100.

<table>
<thead>
<tr>
<th>Feature Description</th>
<th>Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium 4 and Intel Celeron processors</td>
<td></td>
</tr>
<tr>
<td>CPU Bus Speed Supported - 800 MHz or 533 MHz FSB, depending on processor</td>
<td></td>
</tr>
<tr>
<td>Chipset</td>
<td>Intel 915G <strong>Chassis</strong> Standard uATX <strong>Cache</strong> 256 or 1024 KB <strong>Memory</strong></td>
</tr>
<tr>
<td>Non-ECC DDR System Dram (ECC not supported by Chipset)</td>
<td></td>
</tr>
<tr>
<td>Dual Channel Memory with 4 available colour coded DIMM slots</td>
<td></td>
</tr>
<tr>
<td>Maximum Memory 4 GB</td>
<td></td>
</tr>
<tr>
<td>Network</td>
<td>Integrated Broadcom NetXtreme Gigabit ethernet for HP</td>
</tr>
<tr>
<td>Slots</td>
<td>PCI slots - 2 full length, full height</td>
</tr>
<tr>
<td>PCI Express x16 slot - 1 full height (Normal Pinout)</td>
<td></td>
</tr>
<tr>
<td>PCI Express x1 - 1 full-height (4.2 inch), length (13.4 inches)</td>
<td></td>
</tr>
<tr>
<td>AGP slot - None</td>
<td></td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Microsoft Windows XP Professional with service pack 1a</td>
</tr>
</tbody>
</table>

The technologies that are used at the organization are based on the sharepoint, office and asp platforms. We will explain them in more detail later on.
CHAPTER 2. THEORETICAL KNOWLEDGE

2.1. RIA TECHNOLOGY

With this technology the costs implied in the installation and updating are reduced, as well as the time that combines the advantages of the client/server approach with the new client free of HTML type.

The use of the web as a platform for increasingly complex applications and the prevailing trend towards development standards have given way to the creation of new client RIA applications, with rich and interactive graphic interfaces that can function within the browser or within another application on host, and in general, they are downloaded on request each time they are executed.

The examples of the applications constructed over RIA clients include systems of reservation, online purchases or auction sites, games, multimedia applications, calendars, maps, chat applications, interactive design applications, systems of entering and presentation of data, amongst others.

The RIA technology guarantees a dynamic, simple and intuitive response, allowing for the visualization in real time and the support of the use off-line, separating the GUI (graphic user interface) server performance, for they both exchange only data and nothing more than data.

2.2. INTRANET

Before explaining our intranet, its characteristics etc. it is necessary to know what an intranet is and what it is used for. Here is the explanation:

2.2.1 What is an Intranet?

It is a private network of educational or business use, which uses Internet TCP/IP protocols for their basic transport. Those students and employees who are registered in an Intranet network can access a great variety of Internet resources, which without that registration access could not be possible, for they are configurated with a restricted access.

An Intranet is usually made up of a number of different networks within a university, company, etc, which communicates with another by means of the TCP/IP. These separated networks are known as subnetworks. The programs they are made up of allow people to: communicate with each other by means of the e-mail and notice boards of public messages, collaborate in the production by using the working groups programs, which is among the most powerful Intranet programs.
The applications that allow the different departments of universities to send information, and allow the students to fill in forms and use the university’s personal and general information in a private way, are very popular. In general we could state that it is a local area network which provides Internet tools directed to a business or educational group, which allows for all the group members to communicate with each other, and which furthermore does not allow any person from the Internet to access this private network called Intranet.

2.2.2 What is it useful for?

An intranet network facilitates the publication of information by means of the HTML language for all the people who have access to the intranet. Besides facilitating the publication of information and important data related to the business or educational group it has been created for, this kind of networks provide security to these groups, because the information is only available on a local network, and it does not appear on the Internet, which is where everyone has access and many people could access data that could be confidential. That’s why they provide confidentiality among its users, preventing any unregistered member to gain access. However, from a private network, Internet can be equally accessed, for initial multimedia pages can be created, which in turn are made up of text, graphic and multimedia contents whose hypertext links allow for the moving around in the web, which means sites can be searched for within an Intranet or outside the Internet from an home page.

2.2.3 Advantages and Inconvenients

As we have already seen, an intranet provides great advantages: It allows for the communication among its users, as well as security on the data gathering, for everything is within a private LAN, which could in turn be an inconvenient, for the fact that the existence of private information is the object of many hackers. However, what is clear is that nowadays everything is computerized, and one of the safest ways of having information available accessible for a company, school, university, etc, is by means of an intranet network.

2.2.4 Examples of an intranet

We can see examples of intranets in the majority of companies. A good example could be the intranet of the Education Department of Catalonia.
Fig. 2.3 Screenshoot of the Education Department of Catalonia intranet.

On the former figure we can see the Home Page of the Education Department of Catalonia intranet. All the employees of this public administration have access to it, by means of which the users are informed about all the news, changes, promotions, and modifications that take place. Another service offered by these networks, is the research of personal data of any administrator that is a member of the entity.

2.3. **COM Objects and management of Transactions**

Component Object Model (COM) is a Microsoft platform for software components launched by this company in 1993. This platform is used to allow for the communication between processes and the dynamic creation of objects, in any programming language that supports this technology. The term COM is often used in the world of software development as a term that includes OLE, OLE Automation, ActiveX, COM+ and DCOM technologies. Even though COM was first launched in 1993, Microsoft did not emphasize the name COM until 1997.

COM also competes with CORBA and Java Beans menjo software components system.

Its main use is being able to work with Windows API even making webservice, for it provides us a very high degree of depth at the time of programming. It has many implementations, such as generating webservice, activeX, etc.

The application’s main function is being able to look up databases and being able to apply the treatment to the query result depending on the desired conditions with loops, etc. Finally generating an XML code with the data we want, for the ajax engine that constructs the application. And taking advantage of all this being comprised in a DLL already combined with the increase of speed that represents as well as its security. I would like to add that this way, together with a database engine that admits transactions, such as the SQL Server, we can control all the transactions process. But all this is described in fuller detail on the application explanation.
What I mean by transactions process is basically controlling the requests to the databases. For example at the time of making an insertion on a table it cannot be inserted at the same time, for it could happen that two records had the same ID, causing wrong information and problems on the applications. Managing the actions over the databases by means of the transactions we made sure this could not happen. All the requests go through the DLL, which has already the transaction defined. Once the transaction is started another one cannot be started until the former is finished.
CHAPTER 3. TECHNOLOGIES

3.1. Languages

3.1.1 HTML

HTML is the marking-up language by means of which webpages are written. Thanks to the Internet and the web browsers, it has become one of the most popular existing formats for the drawing up of documents. It consists of a structured hypertext language, that allows to have a reference of other text documents by means of links called hyperlinks. This text is composed of tags which indicate us where the document elements start and where they end.

A hypertext document is not only made up of text, it can also include images, sound, etc. Therefore, its result can be considered as a multimedia document. Thus, HTML, gives the authors of the pages that have been created, the necessary tools to:

• Publish documents online with headers, texts, tables, pictures...
• Obtain information online by means of hypertext links.
• Design forms that allow us to register, make reservations, etc.

3.1.1.1 HTML History

HTML was originally developed by Tim Berners-Lee. In the mid nineties, the enlarging of HTML started to achieve the desired presentation, but always from the different developers' perspective, who ended up with different non-standard solutions for different browsers. HTML 2.0 was developed to encode what was common practice at the end of 1994. In 1993, HTML+ and HTML 3.0 in 1995, proposed more enriched versions of HTML.

Even though an agreement has not been reached regarding the discussions over the standards, these drafts led to the adoption of a number of new features. The efforts of the HTML Working Group of the World Wide Web Consortium to encode the common practice, culminated in HTML 3.2 which was approved in January 2007. This new standard included enhancements provided by the Internet Explorer and Netscape Navigator browsers which had already carried out some extensions of the HTML 2.0 standard. Each of the HTML versions tried to reflect an increasingly greater consensus among the industry’s spokesmen, so the investments the contents providers had made were not depreciated and so the documents stopped being readable in the short term.

In December 1997 the HTML 4.0 standard was approved, which developed the HTML language with mechanisms by the style sheets, scripts executions, frames, as well as improved support for the text from right to left and mixed
directions, richer tables and better forms. Besides it provided accessibility improvements for disabled people.
Finally in September 2001 the 4.01 standard was approved after a revision of the HTML 4.0 which amended errors and at the same time included some changes of the former revision.

3.1.1.2 Core principles

Browsers have the function of interpreting the HTML code of the documents and show the users the resulting webpages of the interpreted code.

Core tags.

Every page of HTML must have some beginning and ending tags, which define a webpage and which without them a Web on HTML does not make any sense.

```html
<html> </html>
<head> </head>
<body> </body>
```

The rest of information and tags we have used in our intranet, will be explained later on another section. **CORE TAGS OF A WEBPAGE**

3.1.2 CSS

3.1.2.1 What is CSS?

They are cascading style sheets, which show us how a document is going to appear on the screen or how it is going to be printed. These sheets allow the webpages creators to have a total control over their documents style and format.

They are a practical and efficient system to control the appearance of the webpages. They let us control everything regarding the representation on the
screen, or even on other supports, such as printed means, voice readers... Therefore, CSS is an easy language for the application of styles to a HTML element.
The main function of the style sheets is allowing the designer to separate the contents from the presentation. The basic idea is that HTML takes care of checking the text contents (paragraphs, headers, lists...), and everything related to the document presentation (colour, font, aligning...) is in the hands of the CSS.

3.1.2.2 What is its aim?
The HTML language is limited at the time of giving shape to a document, that is, providing the wallpaper, etc., with colour, different text styles...
This is so because HTML was conceived for scientific use, different to the current use and a lot wider, where the presentation details were not taken into account.
So with the aim of solving this problem, designers used techniques such as the use of tables, images, other tags different from the HTML’s and similar techniques. But all these techniques gave a lot of trouble at the time of visualizing the pages on different platforms.
That is why its designers searched for a more efficient solution and this is how the CSS style sheets came into being.

3.1.2.3 What is it used for? Advantages
The performance of the CSS consists of defining, by means of a special syntax, the shape, colour, etc, we shall apply to:

- A whole web, in such a way that its shape can be defined all at once.
- A HTML document or page, for the shape can be defined in a small piece of code on the header of the whole page.
- A part of the document, by applying visible styles in a piece of the sheet.
- A particular tag, being even able to define different styles in one single tag. This is important for it provides great power to our programming. For example, we can define different types of paragraphs: some in red, some others in blue, with or without margins...

The technology’s power leaps into view, it is not only limited to this, but furthermore, besides this CSS syntax, it allows to apply to the document, a format in a much more precise way.
If some time ago HTML fell short with the pages and we had to use tricks to achieve our effects, nowadays we count on a lot more tools to define this shape:

- We can define the distance between the document’s lines.
- We can place elements on the page with greater precision, and without any room for errors.
• And much more, such as defining the elements’ visibility, margins, underlinings, linings...

And it has more advantages, for if with HTML we could only define attributes on the pages with pixels and percentages, now we can define them by using a lot more units such as:

• Pixels and percentage (%), as before.
• Inches (in)
• Points (pt)
• Centimetres (cm)

Therefore, as a summary of the advantages the CSS brings forward, we can state they are:

• Centralized control of a website presentation where its updating is sped up to a great extent.

• The browser allows the users to specify their own local style sheets which will be applied to a remote website, increasing the accessibility to a considerable extent. For instance, people with sight problems can configure their own style sheet to increase the text size or highlight links.

• A page can have different style sheets depending on the device that shows it or even the user’s election. For example, it can be printed, showed on a mobile device, or be "read" by a voice sintetizer.

• The HTML document itself is easier to understand and it is possible to reduce its size to a great extent.

• Separation of contents from presentation.

• The code has multiple uses. The same style sheet can be applied to different HTML documents: we write it only once and we apply it to as many pages as we wish.

• Easy to modify. If for instance we want to change the appearance of all paragraphs, we only have to modify the corresponding definition on the style sheet.

• Greater control over the document’s presentation. The use of CSS allows to control many aspects related to the presentation of the traditional HTML.

3.1.2.4 How does it work?
The creation of a style sheet is done by means of statements whereby we can define the properties the text will have, to which they are referred. Each statement is made up of a selector and a declaration block.

Therefore:

**Statement = selector + declaration block**

Selector: identifies the HTML elements the statement will have an effect on. Declaration block: it tells the browser how to present the selected elements, that is, which will be the properties of these elements. The declaration block is closed between curly brackets `{ i }`.

Out of each declaration we must define the text property and value. In such a way that by means of this statement we will modify the font size, which will be of 10 pixels. The colour, which will be black, and the font family, which will be Verdana, for instance.

In order to associate a style sheet, we count on two basic methods, which are **linking** and **embedding**.

Furthermore there is yet another commonly used possibility which is called **inline style sheets**.

Before going into the explanation of each method we have to bear in mind that:

- HTML pages are extension files `.html`
- The style sheets are files with extension `.css`
- Within these files the aforementioned statements are embedded.
- Each of the statements has a name. If this name coincides with any of the names of the HTML tags, the style will automatically warn us when we are within this tag.

```html
P {
  font-size: 10px;
  colour: #000000;
  font-family: Verdana }
```

3.1.2.5 **Linking**

It consists of using the LINK element on the HEAD section of the HTML document to indicate which `.css` file has to be used on different HTML pages. Besides it's clear that on the one side there is the contents (**file .html**) and on the other the presentation (**file .css**). It's possible to associate more than one `.css` file by using a LINK element for each one of them.
The syntax to link a style sheet is:
The URL is the address where the .css file is located. It can be a total or a local address. Besides, in the case where the HTML file and the CSS file are on the same directory, we will only have to type the name of the .css file.
What we do next, is by means of the HTML LINK tag, we call an address where there is a style sheet within which the styles we want to use are defined.

### 3.1.2.6 Embedding

The other option is to “embed” the same HTML document (also on the HEAD section) on the style sheet, that is, type the code directly. In this case, there will be no .css file, both the style sheet and the HTML will be on the same file.
This has several disadvantages, the main one being that we will be likely to copy the style sheets on each and every document that uses them. Any later change will be more time-consuming and it will be more likely to have more errors than if we only had to modify one file. However, deciding on one possibility or the other will depend on each case’s particular factors.
The syntax to “embed” a style sheet is:

```html
<HTML>
<HEAD>
  <LINK rel="stylesheet" type="text/css" href=uri>
</HEAD>
</HTML>
```

Therefore, in this case there will be no .css file, but the statements are directly written on the .html file.

#### Inline style sheets

A last alternative is to include the declaration block in the HTML individual elements. Strictly speaking, we do not assign any style sheet to the document, but we define the style of a particular element directly on the HTML code. To do it we will use the style attribute, which can be applied to (nearly) any HTML element, and as a value we will write the declaration block without the curly brackets `{ i }`.

This method does not allow the reuse of the style (if we want to apply it to all paragraphs we will have to copy it on each of them: it’s a useful resource in certain situations).

```html
<HTML>
  <HEAD>
    <STYLE type="text/css">
      ...
      all the statements are located here
    </STYLE>
  </HEAD>
  <P style="color:red;font-size:20px">
```

A paragraph of red colour and size 20px.<p>

3.1.3  JAVASCRIPT

3.1.3.1  What is Javascript?

JavaScript is a script language based on the prototype concept, originally implemented by Netscape Communications Corporation, which resulted in the ECMAScript standard. It’s mainly known by its use on webpages, but it’s also used on other applications.
In spite of its name, JavaScript does not come from the Java programming language, but both of them share a similar syntax inspired by the C language.
The name “JavaScript” is a trademark by Sun Microsystems.
Javascript, like Java or VRML, is one of the multiple ways that have come up to extend the capacities of the HTML language.

3.1.3.2  History

More or less in the same period of time, Sun Microsystems launched the JAVA programming language (which was originally called Oak), which quickly gained popularity, and for merely commercial reasons its name was changed into Javascript.
The 1.0 version appeared with the 2.0 of the browser of the same company and it was later on incorporated in the iExplorer 3.0 by Microsoft.
A short time later there also appears VBScript (Visual Basic Script) by Microsoft, a very good competitor.

3.1.3.3  What is it used for?

In order to understand what Javascript is used for, first we have to know what a script is. That’s why we are going to describe it next:

**What is a script?**

*It is a sequence of commands, in a particular language, which can be executed by a Web client from its browser and so he can be able to visualize the document which is contained in it. Currently the two more widely used script languages on Webpages are Javascript and VBScript. The inclusion of script on HTML documents makes this latter more intelligent. The contents is generated in a dynamic way, while the values introduced in the forms can be checked locally, without the need of having a server and wasting a certain amount of time on this. In spite of the name, Javascript, this language has little to do with JAVA.*

Javascript is mainly used to improve the client/server interface management. A Javascript script inserted on a HTML document allows for the recognition and local treatment of the client and the events generated by the user. These events can be the HTML document’s path or a form process.
3.1.3.4 Features

Javascript is a language interpreted on the client by the browser when the page is loaded, it’s multiplatform, directed to events with object handling, which code is directly included in the same HTML document.

• It’s simple, for it isn’t necessary to have knowledge on programming to be able to do a program on Javascript.

• It handles objects within our Webpage and on this object we can define different actions. These objects facilitate the programming of interactive pages, and at the same time prevent the execution of commands that can be dangerous for the user's computer, such as unit formats, file modifications, etc.

• It’s dynamic, it responds to actions in real time. Actions such as pressing a button, dragging the mouse pointer on a particular text or simply load a page or expire a time. With this we can completely change the page’s appearance according to the user’s taste, avoiding having a page for each taste on the server, make calculations over the variables base whose value is determined by the user, etc.

3.1.3.5 Their main applications are:

• Respond to local actions within the page, such as pressing a button.

• The calculation-making in real time.

• The form validation within a page.

• The personalization of the page for the user, which will allow him to have a tailor-made webpage.

• The inclusion of data of the system itself, such as the time and date.

3.1.3.6 How to use it?

The Javascript code is directly inserted on the HTML page. There are four ways to do it:
1) On the document body:
That is the <BODY> and </BODY> controls and using the <SCRIPT> control.

For example:
This code is immediately executed when the page is loaded and produces a text over the page. That’s why the “write” method is used, which represents the
current document.

2) On a separate file:
   In this case the script code is located on another file and this file is called for.
The call here is included within the body.

   <HTML>
   <HEAD><TITLE>Title</TITLE></HEAD>
   <BODY>
   <SCRIPT LANGUAGE=JavaScript>
   document.write("Hello ");
   </SCRIPT>
   </BODY>
   </HTML>

3) Using action handlers:
The Javascript controls are immediately evaluated when the page is loaded.
The functions are stored, but not executed, until a certain action is called.

4) Calling a function:
   Within the header, after the title. That is, between the <TITLE> and <HEAD>
   controls.

   Then the call to the functions will be done from the body.

   <HTML>
   <HEAD><TITLE>Title</TITLE><HEAD>
   <BODY>
   <SCRIPT SRC=.js code>
   </SCRIPT>
   </BODY>
   </HTML>

   <HTML>
   <HEAD><TITLE>Title</TITLE></HEAD>
   <BODY>
   <A HREF="" action=method or call to an internal function>TEST</A>
   </BODY>
   </HTML>

   <HTML>
   <HEAD><TITLE>Title</TITLE>
   <SCRIPT LANGUAGE=JavaScript>
   <!--
   function See() {
      alert("Hello hello ");
   }
   // -->
   </SCRIPT>
   </HEAD>
3.1.4 COM with C++ (Creation of DLL’s)

The language I use when generating DLL’s is C++, because it is on a lower level and it allows more depth and control at the time of programming. I could have also used Java, but it’s on a higher level and I could not make such a good code.

The mechanism that is quite higher would be: I generate the code on C++ after compiling it. This includes the dll already compiled. Then I add the container I have created on the component service.

![Screenshot of the Component Services.](image)

And I define the properties of the new document. Such as the transactions, issues on licenses and security, etc.
3.1.5 XML

XML is a standard system of information encoding. The programs that use the XML format can easily exchange their data, since they respond to the same internal logics.

The XML documents are text files that contain information which is organized in the shape of a tree: each branch can have its own attributes and be used as a base for other branches.

The application I will give it will be like an information container where the information will appear from. I will also apply input parameters to the DLL, for it's a structured and descriptive language.

An example of a brief XML document could be:

```xml
<Root>
    <name>jordi</name>
    <name>Pepe</name>
</Root>
```

3.1.6 SQL

3.1.6.1 What is SQL?

SQL (Structured Query Language), is a language derived from an IBM research project to access relational databases. It has currently become a database language standard, and most database systems support it, from systems for personal computers, to big computers.
As its name suggests, SQL allows us to make queries to the database. But the name falls short for SQL further carries out database definition, control and management functions. SQL statements are classified according to their purpose, resulting in three "languages" or better still, "sublanguages":

- DDL (Data Description Language), it includes commands to define, modify or erase the tables where the data are stored and the relations between data. (Is the language that varies most from one system to the other).

- DCL (Data Control Language), it contains useful elements to work on a multiuser environment, where data protection is very important, as well as the tables security and the establishment of access restrictions, besides elements to coordinate data sharing by the attending users, making sure they do not interfere in each others' actions.

- DML (Data Manipulation Language), it allows us to retrieve data stored in the database and it also includes commands to allow the user to update the database by adding new data, erasing old data or modifying data already stored.

3.1.6.2 General features

SQL is a language of access to databases which exploits the flexibility and power of relational systems resulting in a great variety of operations over them. It’s characterized by being of high level or non-procedure, which thanks to its strong theoretical base and its focus towards the handling of sets of records, and not towards individual records, it allows for a high encoding productivity.

3.1.6.3 How to use it?

An SQL statement is like a sentence (written in English) by means of which we say what we want to get and where we want to get it from. All the statements start with a verb (word that indicates the action to be carried out), followed by the rest of clauses, some compulsory, and some optional, which complete the sentence. All the statements follow a syntax so they can be correctly executed. In order to describe this syntax we will use a syntax diagram as the one shown below. Although before continuing with the SQL statements structure, it’s important to bear in mind the working method of the databases these statements or queries act on. As we know a database consists of tables where we store listed records according to different fields (features).

An aspect we have to take into account is the nature of the values we enter in these fields. Given that a database works with all kind of informations, it’s important to specify which type of value we are entering, for on the one hand this facilitates the research later, and on the other we optimize the memory resources.
Each database enters a type of field values which are not necessarily present in others, however, there is a set of types which are represented on all the databases. These common types are the following:

- **Alphanumeric** They contain figures and letters. They have a limited length (255 characters).
- **Numeric** There are different types, mainly whole numbers (without decimals) and real numbers (with decimals).
- **Booleans** There are two types: True and false (Yes or No)
- **Dates** They store data facilitating their later exploitation. Storing data this way allows to order the records on a date-basis or calculate the days between one date and another...
- **Memos** They are alphanumeric fields of unlimited length.
- **Self-increment** They are fields of whole numbers which increment in a unit its value per each incorporated record. Their usefulness is more than obvious: they act as IDs for they are exclusive of a record.

### 3.1.6.4 Structure of an SQL statement

There are many actions, conditions, etc. Next we will explain the most important ones and further on we will define the ones we have used.

Basic statements and queries

### 3.1.6.5 Actions of an SQL statement

**SELECT**

The SELECT statement, which allows to retrieve data from one or several tables. The SELECT statement is the more complex and powerful of the SQL statements. With SELECT, we can retrieve data from a database, but it’s also important to be able to insert new rows, erase rows or change the contents of a table's rows. These operations modify the data stored on the tables but not so their structure nor their definition. We do this by means of the statements: INSERT INTO, DELETE and UPDATE respectively.

**INSERT INTO**

It’s important to be able to insert new data in a table, where for example there is an open listing of people, where more people have to be inserted. For such purpose we have the statement **INSERT INTO.**
The syntax to be used is:

The words in red do not change, the words in orange are the table where the information will be inserted, as well as the values to be inserted respectively.

**DELETE**

The DELETE statement erases rows from a table.
Its syntax is the following:

Just like before, we have the fixed words in red and the variable words in orange. In this case, the first one in orange will contain what we have to erase and the second one the place where we will erase it from.

**UPDATE**

The UPDATE statement modifies the values of one or more columns on the selected rows of one or several tables.
Its syntax is the following:

The name of the table, refers to the table on which we will make the modifications, the name of the column, refers to the column which will be modified and finally the expression is the new value that the modified column will contain.

There are many statements to create tables, such as CREATE TABLE, etc. But these are not so important and we will only name a few.

The CREATE TABLE statement is used to create the structure of a table, not to fill it in with data, and it allows us to define the columns it has and the restrictions these columns have to meet.
The ALTER TABLE statement is used to modify the structure of an existing table. By means of this command we can add new columns, and erase others.
The DROP TABLE statement is used to erase a table, although if it’s open we will not be able to erase it. It will neither be possible if we have erased it. It infringes the *referential integrity rules (if it is involved as a father table in a relation and has related records).
The DROP INDEX statement is used to erase a table’s index.
The index is erased but the columns that comprise it are not.
The TRANSFORM statement is used to define a cross-reference query.

3.1.6.6 Elements of which the action falls

When we put an asterisk (*) as an element on which the action falls, we are indicating all the fields of the table: nothing in particular is selected but the table as a whole.
On the other hand, we can write only the elements of a table on which we want the action to fall separated between commas.

Element 1, element 2
FROM
THE SOURCE TABLE FROM
With the FROM statement we indicate on which table we have to look for the information.

FROM table specifications.

A table specification can be the name of a saved query, or the name of a table. In summary, with the FROM statement what we do is indicate where the information we need has to be searched.

Other important statements which are not compulsory, but which have to be taken into account are:
WHERE and HAVING statements

Subqueries are usually used on the WHERE or HAVING statements when the data we want to view are on a table but in order to select the rows of this table we need a piece of data which is on another table.

ORDER BY
To order the rows of a query result, we have the ORDER statement.

With this statement the visualization order of the table rows is altered, but not so the order of the rows within the table. The table is not modified.

The DISTINCT / ALL statements

When including the DISTINCT statement in the SELECT statement, the repetition of rows is erased from the result. If on the contrary we want the included and duplicated rows to appear, we can include the ALL statement or nothing at all, for ALL is the value that SQL assumes by default.

The TOP statement

The TOP statement allows to extract the first rows from the source table. It does not choose among equal values: if I ask for the first 25 values but the value no. 26 has the same value as 25, then it will return 26 records instead of 25 (or however many there are). It is always guided by the ordering column, which appears on the ORDER BY statement or otherwise the table's main key.

The WHERE statement

The WHERE statement selects only the rows that meet the condition of the specified selection.

On the query there will only appear the rows whose value is true (TRUE). The null (NULL) values are not included in the result's rows.
The selection condition can be any valid condition or combination of conditions by using the NOT, AND and OR operators.

The GROUP BY statement
With the GROUP BY statement subtotals can be obtained. A query with the GROUP BY statement is called grouped query for it groups the data from the source table and it produces a single summary row per each group formed. The columns indicated by the GROUP BY are called grouping columns.

**The HAVING statement**

The HAVING statement allows us to select rows from the resulting table of a summary query.

For the selection condition the same comparison tests described on the WHERE statement can be used. Compound conditions (jointly with the OR, AND, NOT operators) can be written, but there is a restriction.

On the selection condition there can only appear:

- constant values
- column functions
- grouping columns (columns that appear on the GROUP BY statement)
- any expression based on the former statements.

**NAME OF THE TABLE**

As its name suggests, here we will put the name of the table on which we want the action to fall.

### 3.1.7 ASP

ASP (Active Server Pages) is an environment to create and execute dynamic and interactive applications on the Web. HTML pages, control statements and Active X can be combined to create interactive Web pages and applications.

The ASP pages start to be executed when a user requests an .asp file to the Web server through the browser. The web server calls the ASP, which reads the requested file, it executes the commands sequences it finds and it sends the results to the client's browser. Since the command sequences are executed on the server, and not on the client, the server is the one who does all the necessary work to generate the pages which are sent to the browser. The command sequences are hidden from the users, these latter only receive the execution result in HTML format.

Therefore the problem of whether the client can or cannot execute command sequences disappears, the Web server only sends the results on HTML code which is standard and readable for any browser. The ASP page mixes on the same file command sequences with standard HTML code. The asp command sequences are distinguished from the rest of the file text by means of delimiters (a delimiter is a character or character sequence which marks up the beginning or end of a unit).
Some ASP integrated objects are:

Request Object: the Request object is used to have access to the information that goes through the HTTP requests. This information includes the parameters that are passed on from the HTML forms by means of the POST method or the GET method, cookies and client's certificates.

Response Object: the Response object is used to control the information sent to the user. This includes the sending of information directly to the browser, the redirection of the browser to another URL address or the establishment of the cookies values.

Server Object: the Server object provides access to the methods and properties of the server. The most commonly used method is the one that creates an instance of an ActiveX component (Server.CreateObject).

Session Object: the Session object allows the storage of the necessary information for a particular user’s session. The variables stored in the Session object are not discarded when the user goes from one page to another within the application, but these variables persist during the time the user has access to the application pages. He can also use the Session methods to explicitly finish a session and establish the downtime period of the sessions.

In the case of HTML, such delimiters are the "" symbols that frame the Html tags. ASP uses the < % i > delimiters to frame the orders.

An example:
<HTML>
<BODY>
The date is : <%=Now( )%>
</BODY>
</HTML>

The VBScript NOW() function returns the current date and time.

3.1.8 AJAX

According to Wikipedia, AJAX (Asynchronous Javascript And XML) is a web development technique to create interactive applications by means of the combinations of three already existing technologies we will learn on this manual. So you can understand it better, it is a way of creating an application that responds to the user's actions without refreshing the page against the server.

Its operation would be...The user accesses to the application which is sent by the server in HTML, Javascript and CSS formats. Then the application's Javascript code requests the server the data it wants to show, and the server executes a server code on its side which sends the data in XML format to the navigator.
Each time the user carries out an action that means showing a set of data, the javascript layer, repeats the former action in an invisible way for the user and it shows the desired data.

The main problem of the vast majority of applications is the low compatibility between browsers, for the Javascript layer is very complex and often, because of lack of experience on the language, or due to lack of time, the common practice is to program only for Internet Explorer. A future improvement will be making the AJAX application compatible with all navigators, including navigators without javascript.

As an example I will take GMAIL. The key class on Ajax that makes everything possible is: XMLHttpRequest. As I have already pointed out is a javascript class with the attributes: readyState, responseText, responseXML, status, statusText.
Let's choose: readyState and responseText. Let's see readyState: It defines the life cycle a request goes through.

0 initiated, 1 loading, 2 loaded, 3 interactive, 4 Completed and it returns to 0 …

Thanks to these values, available on readyState, we can take a decision taking into account this attribute's value.

Now let’s see responseText. When we make a request to a particular URL, this will go through the former states. When readyState = 4, (request completed), the responseText's attribute will contain the data existing on this URL. I

Under the javascript we would create a function that would be called when clicking on an email to be read and which would create a new XMLHttpRequest object:

```
myrequest = new XMLHttpRequest();
```

After this, I have to make a reading request of our URL, let's place, for example, a .html file on our server:

```
myrequest.open('GET','myfile.html',true);
```

'GET' is the request method, 'myfile.html' is the file to be loaded, and true establishes the how-to in an asynchronous way.

Finally, and after this, it would be enough to wait for the state to be completed. The behaviour on each state change is defined by the "onreadystatechange" attribute, and it is usually assigned to a function that checks the state our request is in. Thus, while we are on 1 (loading), we can show the typical "Loading..." tag. In other words:

```
myrequest.onreadystatechange = function(){
  if (myrequest.readyState == 1) { we show the tag }
  else{ if (myrequest.readyState == 4) {we load the page obtained on the div container }
```

This latter part would be carried out by means of the typical assignment (or in a similar way):

```
document.getElementById('container').innerHTML=myrequest.responseText;
```

This would be a quite simple example of ajax. Instead of using a HTML file what we can request is an xml with data and treat them with the javascript engine according to the user's events over the ajax application engine.

### 3.2. Company’s Infrastructure and System

The environment is made up of two servers:

- SQL Server 2005 Database Server
- Application and Component Container (DLL’s) with the Complus and the IIS (Internet Information Server)

The database server has quite a standard configuration. The encoding used is Latin1. On the figure we can see its structure.

![Database Structure](image)

**Fig. 3.8** Screenshot of Sql Server management Studio view.

As I have already mentioned, on the other server we will have an IIS which will be the one compiling the ASP code and will resolve the HTTP requests of the html or ASP types.

We will also have the COM+ component Server, where we will have the components, as I have already explained.
3.3. Former applications

Now we will recap on what the former applications are like and we will comment briefly which their weak points are.

3.4.1 ASP Type

The former ASP applications is caracteristic of the fact that per each event the whole page is loaded over and over on the same asp page or another time from a form. This makes both the client and the server compile and transmit each time a great amount of code which is the same over and over.

I mean if we have a menu and a table with the report processing and we want to order some column each time we load the information on the whole page again with the menu, etc, but with the new order. Actually it’s the same information but with another order. But it has already involved some ASP and HTML code compilations. And so on and so forth for each event. It does not have a real engine (application) as is the case of an application with AJAX technology.

One application would be the Service of Attention to People (SAP).
3.4.2 MsExcel Type

It has many disadvantages, but even though the folder that contains the file can be shared so everyone can access to it, everybody writes over the same document. Everyone can see what the others have written without any confidentiality whatsoever. Besides when a person has the document open, the rest cannot write until the document is closed by the person who opened it in the first place.

The chosen example of the excel application is that of updating the checkup, because I think the problem is quite obvious.

3.4.3 MsAcces Type

All the problems mentioned in the previous section regarding the MsExcel type applications are the same with MsAcces, but adding the issue that because many people have access to the file containing the access data, the file gets easily corrupted.

Another great disadvantage which is more obvious than in Excel, which usually contains less information, is that in the case of the MsAcces file, you are dragging the whole database with the file. And in the example of the library
materials the amount of information is considerable, for the file reaches the value of 200Mb, and the user has to save this file in order to interact with it.

![Image](image.png)

**Fig. 3.12** MsAcces library application.

The bigger application that works with MsAcces is the library materials one.

### 3.4. Tools

The tools I use are quite basic at first sight, but they have small features which make my programming task a lot easier. When I use a text editor it looks like I go more slowly than with a web editor such as Dreamweaver, but for me it’s a lot better for Dreamweaver is not very practical for me.

The only disadvantage with the ASP language is that I cannot see the compiled code, but as I have a production environment I work on the container folder and I simply call the page I have programmed which is solved by IIS. And this way I can see the result compiled on the navigator.

In the case of the C++, to do the DLL’s I use Visual Studio C++, which allows me to do an application debug, but not visualizing it depends on the component Service. So the strategy is similar to the ASP one, only that on this one I pack it on a DLL, place it on the Com+ and create and object from the component. In case of error I only have to visualize the system’s event viewer.

#### 3.5.1 Ultraedit

This editor cannot be considered an editor along the lines of notepad, for it has a series of extras that make it quite different. I would like to point out that depending on the programming language it shows colour markups, so it’s easy to see whether we have typed the wrong key or whether the syntax is not correct.
It also carries a converter to hexadecimal, it allows to select text columns, which for instance is very handy at the time of generating inserts for the database.

I would also like to remark that it also supports multiple files at the same time, and of unlimited size, “dragging and releasing”, hexadecimal edition, colour configuration depending on the syntax for programmers (it already has HTML, Java, Javascript, C/C++, VB by default), column edition, a customizable tool bar, installation on the task bar tray.

It offers UNICODE support, text completed-act, enhancements on the syntax highlighting (with font styles), support to save and extract multiple elements.
from the Clipboard, keywords correction-act, possibility of erasing bookmarks and jump to the previous one, IME integrated support, and much more.

### 3.5.2 Visual Studio C++

Visual C++ is a development integrated environment which allows the programming oriented to objects (POO) together with the SDK development system (or so-called API) by Windows. Because it is an integrated environment, Visual C++ includes, amongst others, the following development tools: Compiler, Linker and Purifier text editor, Data and dependence viewer (Browser). But if Windows is a friendly system from the user’s point of view, from the developer’s point of view it is not.

Windows SDK is nothing but a complex set of functions that it furthermore adds numerous definitions of new data types for any C/C++ programmer.

On the .Cpp Validation example I have chosen, the structure is clear: a column treesystem-like that shows the classes and functions. On the right we can see the chosen function. It also has colour code to see if have made a mistake, then the tool menu to be able to do a Debug and compile on DLL.

![Fig. 3.15 View of Visual Studio C++ 2005.](image)

### 3.5. Technologies difficulties

- The use of ajax to make requests and receive information by means of http_request. As we have to do a task, two http_request could be gathered at the same time, causing the information to be cross-checked and being incorrect.
• Use of the object to be able to attach an xml document with the xsl by means of asp. However in order to have more control over it I use directly xml-ajax without using the xsl sheets. We modify the content directly with CSS.

• The handicap of making use of the component (com) service and the dll's programming, which I had not done before for what it implies. This includes the transaction management, etc.

• Several tasks on javascript such as functions to do searches, paths, both matrix and variable, which contain information of the XML type.

• The use and modification of graphic libraries on javascript and provide a more friendly environment along the lines of the desktop, which is one of the things the RIA (in our case AJAX) technology allows.

• Generating a validation that was integrated on the active directory and validated the user to the applications from the first moment of validating to enter the computer (communication by means of Idap with the active directory server), thus achieving that a new user arrives to the organization when the Active directory generates a user, and so not having to type the user name and password each time. It also allows the management of licenses on the applications starting from the com+ licenses on the active directory.

• utf8 encoding problem when doing the http_request, which request is done with utf-8. Then as our web is on Latin1, at the time of using a form the information to be sent is on Latin1, but as it is being sent with the http_request request on utf-8 the information is encoded. So we have to decod the utf-8 to latin1 so the information encoding coincides with the database.
CHAPTER 4. APPLICATION DEVELOPMENT

4.1. Application requirements

In principle being two technologies, one AJAX, based on an object that exists on the Windows operating system like COM+, no requirement was necessary. But it is necessary a configuration series on the Web system regarding the security when logging in and validating.

On the following figure we can see clearly where I have unchecked the anonymous access, so somebody who is not registered on the active directory cannot access the intranet. Besides checking the integrated Windows Authentication, with this configuration everything would work even with the VPN from outside the company, because if you enter without being validated within the domain you are asked for authentication by means of the Ldap protocol.
4.2. Environment of test

For the same reason as on the previous paragraph, as these technologies are compatible with Microsoft’s operating system, I can install everything on a desktop computer like a production environment. The chosen computer is HP HDX-9450ES.

Fig. 4.17 IIS (Internet Information Server) Security configuration.

Fig. 4.18 Production environment computer HDX-9450ES.

HDX-9450ES Technical Data

Trademark: HP Functionality: Multimedia Processor: Intel Core 2 Duo T8100 Processor Speed: 2.1 GHz
Cache Memory: 3 MB of level 2 cache.
RAM Memory: 4096 (2 x 2048 MB) MB.
RAM Memory Type: DDR2
Hard Disk: 500 (2 x 250 GB) GB.
Hard Disk Controller: SATA a 5,400 rpm
Optical Unit: Blu-Ray ROM with DVD±R/RW Double layer SuperMulti Graphic Card: NVIDIA GeForce 8800M GTS, until a total of 1791 MB of available memory for graphics with 512 MB of exclusive dedication
4.3. **Basic application structure**

All the technologies are used on a set of steps trying to take the maximum advantage of their strong points. This is summed up in two similar charts but with slight differences which I will mention later on on the final conclusions.

The first chart:

![First chart of the application construction](image1)

**Fig. 4.19** First chart of the application construction.

The user makes the request to the ASP homepage which already contains the HTML code and the Ajax engine. When the user makes an event managed by the Ajax engine of the application, it fills in a series of variables which are sent by means of the post method towards an asp which builds an input XML and sends it to the corresponding function of the DLL generated. Then the function carries out its tasks, among which are the making and gathering of requests according to the parameters transported to the database. The DLL function gathers this information, building the output XML, which in turn is gathered by the ASP, which is treated if it is necessary, and arrives to the Ajax engine, modifying the content the user has requested by means of the event without the need of reloading the whole web, only loading the information requested.

The second chart:

![Second chart of the application construction](image2)

**Fig. 4.20** Second chart of the application construction.
The second chart is exactly the same as the first one, but only a little more complex. When I arrived to the organization, webs used to be constructed with xml and xsl documents, for this way the xsl was modified, and without touching the xml all the structure and image of the ASP web was able to be changed. The difference is that when the page is loaded the ASP homepage is not directly loaded with Ajax, but the asp creates an object that fusions the XML data with an XSL template, which will be the one the Ajax engine will contain. Unlike the first chart, where the HTML code was on the ASP homepage.

My intention was keeping on using this methodology, but it was a little redundant as the task I was carrying out with xsl could also be carried out from Ajax. Instead of modifying the XSL I will now modify the graphic part of the Ajax engine. This is the reason why I have opted for the first chart which I think is the most ideal one at this moment, searching for optimum and quick results.

I would like to remark that I could do without using ASP, but for security and time reasons I prefer this option for the future.

4.4. Incidence Application

A good way to see what the applications are like is showing you the first one I developed. It is the application of incidents which was born out of a need at the entity for the management both of maintenance and computer incidents.

4.4.1 Description

The application is made up of several parts: one for the client and another for the administrator, both for maintenance and computer incidents.

On the following figure we can see the client application, where it shows the validated user the computer requests and the state they are in, the state date, etc.

![Fig. 4.21 Client screen of the incident application.](image)

The administration part is seen on the following figure’s screenshot, where the user that has made the request can be filtered as well as all the users.
4.4.2 Databases

The system of tables is quite simple. It is mainly a table that saves all the incidents, be them material or computer, with their respective ID. Also the staff table for the requester data. Table of the incident state type.

The relation between the tables is done:

Relation1: dbo.INCIDENTS.ID_STATE with dbo.STATES_INC .ID_STATE
Relation2: dbo.INCIDENTS.REQUESTING with dbo.STAFF.ID_PERSON
We carry out the queries, updates, inserts and eliminations with the SQL language which I have previously mentioned, by means of the DLL's.

### 4.4.3 Structure

The folder structure is based on three folders:

**CSS:** They are the style sheets.

**IMG:** Image folders and subfolders.

**JS:** Files with the ajax engine and functions.

The pages are of the asp and xsl type. The asp ones are used to treat the validation and call objects, as well as to generate xml type files and attach them with the xsl to generate the code which is compiled for the user on the navigator.

The file that constructs the page is the XSL file, which we will study next in order to understand its whole performance.

This file has a header where the encoding, xsl file version, etc, are determined. As we can see on the following figure.

![Fig. 4.25 Screenshot of the intranet homepage header.](image)

On the following figure we can see how the js files which contain the javascript function are annexed. Also the annex of the css files where all the component styles are defined.

![Fig. 4.26 Screenshot of the call to annexed files.](image)

Within the document we find three containers where all the information will move around.

- **Results:** Information about the movements.
- **Main Table:** All the applications.
- **Footer:** Menu, validation and page footer.
In order to explain the movement of everything on the page I will divide it into two parts, a home and a work one.

Home:

The call is made to the asp document that constructs the page from the xsl document, AJAX and DLL’s functions.

Work:

Depending on the events on the AJAX engine http requests are made to other asp pages which will call DLL functions returning the result on xml data format, and the AJAX engine will treat this data accordingly. The syntax would be: the user generates an event which causes a function to be executed with a series of variables which arrive to a DLL from and asp and this latter returns it as we had already stated before.

On the client and administrator part applications we have the following functions:

**validation.js**

function maxLength(field,maxChars)
function maxLengthPaste(field,maxChars)
function empty(field)
function characters(fieldid)
function validate(form)
function validate_ant(form)
function other validations
function gather(name)
function data_cat()
function users(type)

**incidencies.js**

makeRequest(url)
limitfil(xchain)
alertContents()
function comment(io,iden)
function print()
function maintenance()
function queryi()
function quierm()
function comptingi()
function comptingm()
function compting()
function query()
function maintenance()
function staff()
incidents_insert.js

function creaAjax()
function FAjax (url,layer,values,method)

a incidents.js

function makeRequest(URL, type)
function alertContents()
function changestate(newstate,record)
function comment(io,iden)

As it would be too long to comment on them, I will mention the most descriptive one:
function FAjax (url, layers, values, method)

This function creates the object which we we call objectajax. With the url, the values for the url and the method used. What is done on the url, is that the values are passed on by means of the desired method. An example.

FAjax(“insert.asp”, “Result”, “incident=488”, “POST”)

What this would do is generating a HTTP request of the insert.asp page passing on the incident value=488, with the POST method. What the HTTP request returns goes to the Contents container.

Then “insert.asp” in this case it only gathers the input variables and converts them into XML format and the ones of the insert functions, such as the parameters of the Database name, table to be inserted. It calls the object function of the COM+ DLL, which in this case is “incidents.Cincidents”. Resulting in: “IncidentsInsert(istrXmlIn)”. This function would receive the xml, and it would do an “insert” on the corresponding database responding with an xml document and giving the ok. And all this within a transaction, since it is a no select insert.

The asp document would receive the ok, which in this case would not do the treatment of the xml file and would send it to the ajax engine which would give the insert ok on the result container.

4.5. Other applications and Future applications

The intranet contents management is done by directly reading from an Asp of contents reading and constructing the read document within the intranet’s corresponding container.
On the previous figure we can see the contents container, one of procedures and another of links. When a container is selected from the menu, what we do is, without having to download the whole page, reading the contents, procedures and links related to the selected menu option, and the contents are filled in with the information returned by the Ajax engine from the Asp. Another application is the persons directory's where we can visualize all the workers and their information in a very dynamic way.

We have a reduced client part, an advanced client part and an administration part.

---

Fig. 4.27 Screenshot of intranet checking each of the containers.

Fig. 4.28 Quick search engine of the persons directory.

Fig. 4.29 Advanced search engine of the persons directory.
Fig. 4.30 Advanced search engine of the persons directory for the application administrators.

The great advantage is that when the intranet is loaded all the persons and their information are downloaded. So the searches are done directly on the user, and this way there is not a request surplus on the database and the server, for the application generates a search event each time we type a letter with the keyboard on the search input.

Other applications are the management of holidays, material, training courses, library, etc. The idea is to adapt all the sections of the old intranet to the new technology and benefit from all its advantages.
CHAPTER 5. CONCLUSIONS

5.1. Summary of results and goals

I have achieved an intranet based on a series of technologies that take load out of the servers, minimize the requests throughput and also minimize the process time thanks to Ajax.

I have achieved the goals I had set out of benefitting from the current technologies to improve and dynamize the organization's intranet.

5.2. Difficulties of the project

The main difficulty is working with the RIA technology in depth, for during the degree, in order to make the maximum amount of contents possible, I had only been able to see the possibilities of RIA without studying it in depth, although I have been able to do it on this final project. And within this technology I have avoided many light difficulties, such as the subject regarding the Ajax request which is encoded on UTF-8 when we have all the information on latin1, including the database. Also issues with Internet Explorer and the Cache management, which have been solved by requesting the asp contents not directly from the file. This way the cache does not intervene. Also the issue that small HTTP are created, which was solved by controlling the HTTP object. I empty the variable when the request is completed, and a request does not start until the first one is finished.

The second difficulty was working again with C++ and program COM+ DLL with the transaction management.

5.3. Personal conclusions

Points in favour: all those related to Ajax, because even though it is not as fast as a compiled code, it has the portability and flexibility of the interpreted codes. But if the user, regarding the web applications, has an extra point in his favour, that is the process power, for the process that needs an Ajax engine is very small and perfectly bearable for the user.

Use of the transaction Management, having more control over the requests and guaranteeing the non-loss of information or corrupted information on the databases.

Working with compiled DLL’s, which give a lot of process speed. A lot more than on the interpreted languages.

5.4. Technology

The following technological step is to use the RPC protocol to be able to do an instant client-server communication. When a change is done on the database it has to be reported to all clients. It is a kind of Webservice.
As a future programming technology we have the Ruby and Python languages, which are the next languages to be disassociated from the former ones.

Ruby for the possibility of calling directly the operating system. Very powerful for the handling of chains and regular expressions. It is not necessary to declare variables. The syntax is simple and sound. Difference between uppercase and lowercase letters. Management of automatic memory. Everything is an object. Free format: programs can be written from any line and column.

Python has as a main feature the advantage of compiled languages, which is a faster execution. However, the interpreted languages are more flexible and portable. In spite of this, Python has many of the features of the compiled languages, so we could state it’s a semi-interpreted language. On Python, like on Java and many other languages, the source code is translated into an intermediate computer pseudocode.

5.5. Future

My intention was keeping on using the methodology already implemented at the organization, but it was a bit redundant with the integration of the RIA technology. Because the work I was carrying out with the xsl could also be done directly from Ajax. Instead of modifying the XSL I will modify now the graphic part of the Ajax engine. This is the reason why I have opted for the first chart which I think is the best one at this moment, searching for optimum and quick results. What I would have liked to do is to directly call on the DLL object on Ajax, saving many steps. Solving the security issue directly with Ajax, for anybody could edit information from the application engine. On the following figure we can see what this final chart would look like.

And the most important thing which is quite clearly seen is that the Web server now has to solve only HTTP requests, it does not have to compile the asp code with each request to return the HTML code. The most important load falls on the user with the engine he has downloaded on the application and the Database server.

The next step is done thanks to working with DLL and API (Application Programming Interface) and TAPI (Telephony API) by Windows.
Being able to do implementations on the applications such as by means of Twain, in order to scan directly on the application without the need for applications to third ones.

The big thing would be working with TAPI (Telephony API), making for example that when we search for a person on the intranet directory, when clicking we call that person thanks to the TAPI communication of the IP Switchboard we have at the organization, regarding both internal and external calls. Do Rate-setting tasks. Extrapolate this other applications of the organization. For instance when we deal with a dossier, being able to call the interested person in a dynamic and fast way.
BIBLIOGRAPHY


