TITLE: Research evaluation and selection of a CMS (Content Management System) platform and a web applications platform for the Gavà’s city council.

DEGREE: Diploma in telematics (First cycle).

AUTHOR: Rubén Marín Garrote.

DIRECTOR: Manuel Antonio Marín Vivas.

CO-DIRECTOR: Oscar Caro.

SUPERVISOR: Dolores Royo Vallés.

Overview

This project is the first step of a modernization project in the technological field that the Gava’s city council is carrying out.

The project consists in the research, evaluation and selection of a CMS platform and a web application platform and finally, the execution of a portal using the solution chosen.

The CMS platform must have the following features:

- The platform has to be based in J2EE
- It has to be possible to run over a LAMP (Linux, Apache, MySQL and PHP) server.
- It must be an open source solution.
- The solution must support the features in the current portals apart from adding new ones.

The web application platform must be compatible with the CMS solution chosen and with the following features:

- It has to support applications web based on JSF and Struts.
- It also has to provide different tools for manage and keep updated the data base.
- It must support portlets and web-services technologies.

At the end of the project we have achieved the main aim improving the initial expectations. The success of the solution chosen has permitted the possibility of creating not only one but two portals www.gavatv.cat and www.elbruguers.cat
Acknowledgements

I would like to thank to the I.T department of Gavà’s city council the possibility of carrying out such an interesting project, especially to Manel Marin and Oscar Caro who guided, taught and helped me.

I would also thank to the rest of the team who treat me as a member of the group.

I know that without them, it couldn’t have been possible.

Thank you.
INDEX

CHAPTER 1. INTRODUCTION .................................................................................. 7
  1.1 Current situation in the Gavà’s city council ................................................. 7
  1.2 Conclusions ................................................................................................. 8

CHAPTER 2. Evaluation of the products ............................................................... 9
  2.1 Marked options ............................................................................................ 9
  2.2 Why java? .................................................................................................. 10
  2.3 Final choice ................................................................................................ 10
    2.3.1 JBoss Portal ......................................................................................... 11
    2.3.2 Exo Platform ....................................................................................... 12
    2.3.3 LifeRay ............................................................................................... 13
  2.4 Conclusion .................................................................................................. 17

CHAPTER 3. Liferay ............................................................................................ 18
  3.1 Introduction ................................................................................................ 18
  3.2 History ....................................................................................................... 18
  3.3 The product ................................................................................................ 18

CHAPTER 4. New Gavà Televisió site ................................................................. 19
  4.1 Introduction ................................................................................................ 19
  4.2 Frontend Pages .......................................................................................... 21
    4.2.1 Inici page ............................................................................................ 21
    4.2.2 Televisió pages ................................................................................... 22
    4.2.2.2 Objectius page ............................................................................... 22
    4.2.3 Programació pages ............................................................................ 24
    4.2.4 Serveis page ....................................................................................... 27
    4.2.5 Videoteca page ................................................................................... 28
    4.2.6 Other pages ......................................................................................... 28
  4.3 Backend pages .......................................................................................... 29
    4.3.1 Master page ....................................................................................... 29
    4.3.2 Admin page ......................................................................................... 30
    4.3.3 Tutorials page ..................................................................................... 30
  4.4 Own portlets .............................................................................................. 31
    4.4.1 Portlet videoteca ................................................................................ 31
    4.4.2 Portlet videoteca youtube .................................................................. 36

CHAPTER 5. New El Bruguers site ................................................................. 42
  5.1 Introduction ................................................................................................ 42
  5.2 Frontend pages .......................................................................................... 43
    5.2.1 Portada page ....................................................................................... 43
    5.2.2 Hemeroteca page ............................................................................... 45
    5.2.3 Esports page ....................................................................................... 46
    5.2.4 Opinió política page ........................................................................... 46
    5.2.5 Edició impresa page .......................................................................... 47
    5.2.6 L’enquesta page ................................................................................ 47
    5.2.7 Contacta page .................................................................................... 47
    5.2.8 Subscripcions al diari and Subscripcions a l’agenda pages ................. 47
    5.2.9 Farmacies and telèfons pages ............................................................. 48
    5.2.10 7 dies pages ....................................................................................... 48
  5.3 Backend pages .......................................................................................... 49
    5.3.1 Portades Programades pages .............................................................. 49
    5.3.2 Administració page ............................................................................ 50
  5.4 Own portlets .............................................................................................. 50
    5.4.1 Portlet búsqueda impresa .................................................................. 50

CHAPTER 6. CONCLUSIONS .............................................................................. 55
  6.1 Conclusions ................................................................................................ 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP4.2</td>
<td>Installation and configuration files</td>
</tr>
<tr>
<td>AP4.3</td>
<td>Deployment</td>
</tr>
<tr>
<td>AP4.4</td>
<td>What is a Theme plugin?</td>
</tr>
<tr>
<td>AP4.4.1</td>
<td>How to create a theme using plugin SDK</td>
</tr>
<tr>
<td>AP4.5</td>
<td>What is a Portlet plugin?</td>
</tr>
<tr>
<td>AP4.5.1</td>
<td>How to create a portlet using plugin SDK</td>
</tr>
<tr>
<td>AP4.6</td>
<td>What is a Layout template plugin?</td>
</tr>
<tr>
<td>AP4.6.1</td>
<td>How to create a layout template using plugin SDK</td>
</tr>
<tr>
<td>APENDIX 5</td>
<td>Getting started</td>
</tr>
<tr>
<td>AP5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>AP5.2</td>
<td>Communities</td>
</tr>
<tr>
<td>AP5.2.1</td>
<td>Create a Community</td>
</tr>
<tr>
<td>AP5.2.2</td>
<td>Assign a Community Virtual Hosting</td>
</tr>
<tr>
<td>AP5.3</td>
<td>Layouts</td>
</tr>
<tr>
<td>AP5.3.1</td>
<td>Adding a new layout</td>
</tr>
<tr>
<td>AP5.3.2</td>
<td>Configuration of a new Layout</td>
</tr>
<tr>
<td>AP5.4</td>
<td>Creating new content</td>
</tr>
<tr>
<td>AP5.4.1</td>
<td>Structures</td>
</tr>
<tr>
<td>AP5.4.2</td>
<td>Templates</td>
</tr>
<tr>
<td>AP5.4.3</td>
<td>Articles</td>
</tr>
<tr>
<td>AP5.5</td>
<td>Roles</td>
</tr>
<tr>
<td>AP5.5.1</td>
<td>Default community roles</td>
</tr>
<tr>
<td>AP5.5.2</td>
<td>Creation and definition of community roles</td>
</tr>
<tr>
<td>AP5.5.3</td>
<td>Adding a new user</td>
</tr>
<tr>
<td>AP5.5.4</td>
<td>Assignment of community roles</td>
</tr>
<tr>
<td>APENDIX 6</td>
<td>Letter of acknowledgment</td>
</tr>
</tbody>
</table>
CHAPTER 1. INTRODUCTION

1.1 Current situation in the Gavà’s city council

In 2001, the city council started a modernization project in the technological field which included the creation of the 1st version of the city web portal. After some time, the portal has suffered a lot of improvements and during its evolution there have been new versions, the current one is the 3rd version.

At the same time, apart from the city website some others have also been developed to complete the city internet services; the digital newspaper, the museum of Gavà, the companies support centre and the local television are some examples.

All those portals are being developed in ASP 3.0 + COM. Active Server Pages (ASP) is Microsoft's first server-side script engine for dynamically-generated web pages.

It was initially marketed as an add-on to Internet Information Services (IIS) via the Windows NT 4.0 Option Pack, but has been included as a free component of Windows Server since the initial release of Windows 2000 Server.

Programming ASP websites is made easier by various built-in objects. Each object corresponds to a group of frequently-used functions useful for creating dynamic web pages.

What ASP technology tries to do is to program in the same way as you do with Visual Basic but of course with a lot of limitations as it is a platform that has not been developed as expected.

An interesting thing about this model is the possibility of using various components already developed like some ActiveX controls (COM).

But on the other hand, a big disadvantage is the lack of useful tutorials to understand this complex technology making virtually impossible to get the most of its possibilities.

ASP has gone through three major releases:

- ASP version 1.0 (distributed with IIS 3.0) in December 1996
- ASP version 2.0 (distributed with IIS 4.0) in September 1997
- ASP version 3.0 (distributed with IIS 5.0) in November 2000

So at the moment the portal is running under an 8 years-old technology which obviously has been highly overcome by others. That makes so very difficult to find support in the developers’ communities.
1.2 Conclusions

The development of the current portal is constrained by the base platform. The new trends focus the web applications towards the end user, providing them some control over the look and also over the contents. That makes easier the collaboration among communities known as social networks.

This is the philosophy in Web 2.0 which adopts different technologies as Ajax [Appendix 1.11], Portlets, RSS, XML, XHTML, Web Services [Appendix 1.13], etc.

Some of these technologies are being used as RSS or XHTML. However, the rest of them are extremely difficult to use them under ASP or even impossible.

These technical limitations prevent the programmers from carrying out new projects as improving the back office tools, customizing the portal home, changing the intranet, etc. Therefore a new base technology for the portal is required in order to keep up with the new possibilities.

Therefore, the objective of this project is evaluating new solutions changing the base technology with the following features:

- It must run over a Tomcat server.
- It must use MySQL as database technology.
- It must be an open source solution.
- It must support all the functionalities and characteristics of the current portal.
- It will have to add new features and make easier the content creation and interaction with the end user (commenting, voting and rating the articles).
- It must have the possibility of adding videos, customize the content, using tags for the news and incorporate wikis, forums and so on.
- The platform needs to support applications based on Java Server Faces and Struts.
- It also has to provide content creation assistants (What You See Is What You Get editors).
- It must have the possibility of add CRUD [Appendix 1.12] applications in order to create read update and delete information from the database tables.
- It must be 100% JSR 168 compliant (Must support portlets).
- It will have to support Web Services technology.

In addition, in a 1st stage, it will be developed two new portals; the Gavà television and the digital newspaper sites, in order to check if the chosen solution satisfies the features required before extending the solutions to the rest of the portals.
CHAPTER 2. Evaluation of the products

2.1 Marked options

There is a wide range of products designed to develop portals and content managers. There are dozens of choices in different programming languages. As for example, there are a large number in PHP whose characteristics and stability are excellent. There are also others in Java (J2EE) which is the most common platform used among the web servers. And in a smaller number, there are applications in Perl or Asp.

2.2 Why java?

Content management systems [Appendix 1.2] are a handy way of building dynamic sites instead of starting from scratch. PHP seems to provide what everyone needs, including a healthy competition among its foremost projects.

Because in most cases CMS’s are deployed on a dedicated server, Java should not be at a disadvantage in this category.

Java uses the Content Repository API (JCR) specification for accessing content repositories in a uniform manner. The content repositories are used in content management systems to keep the content data and also the meta-data used in CMS such as versioning meta-data.

For now, the JCR API does not seem enough however. The PHP CMS projects are continuously gaining ground, probably because PHP hosting is cheap and easy to set up, and because the projects themselves are highly usable.

System integration this is where Java has a big advantage over PHP. Although some PHP managers have support for web services, they have been specially developed for specific applications which do not follow neither the interoperability standards WS-I nor the second generation web services standards like WS-Security or WS-Addressing.

Java is an object oriented programming language whereas PHP is a scripting language. Scripting languages is fast and easy to write which make them very productive for the web. The problem is that there is no separation between the presentation and logic layers.

For example: it is very easy to create a PHP page which connect to a database, execute a select and list all the results in a table but if you do it in the page, multiplied per hundreds of pages your portal may have, Wouldn’t it be hard to find an error or change something?. 
Java, though JSF, forces the developer to split those layers making easier to reuse and maintain the code separating the model view and controller layers.

All the applications could be design to run over a J2EE server to be used through the browser. In addition, using the JSF/Ajax standard, they could be as cool as the classic server/client applications.

In addition, it would be possible to create a UDDI [Appendix 1.10] register and list all the services provided.

The other and perhaps the most decisive advantage is that Java platforms offers better possibilities of integration with other applications using the web services standards and content display in websites (Portlets, WSRP [Appendix 1.14]).

Furthermore, any application developed on J2EE in the department could be done using the same tools and environment. This situation favours the help between colleagues.

2.3 Final choice

After choosing java as a base platform, we start the research of the best software for that project.

Consulting some reviews of important websites like www.InfoWorld.com, www.EcontentMag.com, www.Manageability.org, www.TheServerSide.com and Sun.SystemNews.com. It is a bit difficult to compare different portal framework as each of them addresses different requirements and technologies. We tried our best to be objective and compare different ones with a broad range of criteria to accommodate the speciality of each Portal Framework and maximum consideration of user requirements.

Portals are gaining attention among programmers due to their ease in development, richness in functionality, customization of interface and pluggable architecture. With this popularity today there are many open source Portal Frameworks available and list of these is all the time increasing. It is not possible to evaluate all these Portal Frameworks in an effective manner and we have to select a small number based on their popularity and our experience of using them for development work. This doesn’t mean that other frameworks are below standard or have limitations. We have selected the following ones for evaluation:

- JBoss portal
- Exo Platform
- LifeRay
2.3.1 JBoss Portal

JBoss has also created its own portal with JBoss Portal, a part of the company’s JEMS (JBoss Enterprise Middleware System) Java application development stack. This is a JSR 168-compliant portlet container that offers customization through JavaServer Faces, MyFaces, and Spring, plus additional content management and administration.

The installation of that application is very easy and smooth; it doesn’t take more that ten minutes. You only have to install a copy of JBoss Application Server, unzip the JBoss Portal into the deploy directory, set up a database and start the JBoss AS. It is not necessary to execute any SQL script because the application does it the first time it starts.

Unfortunately with this release there are not a lot of portlets bundled only around fifteen and most of them are for the portal configuration and some other like a (weather portlet which I find useless) and a RSS portlet.

In addition, the UI is not so intuitive, there isn’t any quick help section and adding portlets to pages is at first quite complicated to understand because it is done through a dashboard configuration page rather than the easy drag and drop process.

The last thing worth mentioning is that, JBoss Portal relies on and only works with JBoss AS. This can be a big turn off for applications that only need a servlet
container and if the portal is tied to a unique server application, the solution is so limited in case of incompatibilities with other applications.

I find that this lack of portlets and its dependency to Jboss AS are a big step back comparing it with the other possible choices.

2.3.2 Exo Platform

**eXo platform.** Still in its production phase, this project is designed as a customizable enterprise portal and CMS. Its portlet container is actually certified compliant with JSR 168 and has been enhanced with caching, connection pooling, and shared-session features. The portal is based on JavaServer Faces (JSR 127) technology and offers a Model View Controller architecture.

The installation of the application is as simple as download and unzip the bundle and run a *bat* file.

At first sight, the UI is far cooler and dynamic than jBoss portal UI. In the default installation, it comes with some pages showing some portlet uses and four registered users which you can use to check the variety of possible roles.

When I log in and try to add new applications, the system throws an exception which prevents adding new ones.
Although there are several features that seem not to work properly (perhaps because it is not a final release), the appearance as well as the description of the product make me to consider it and relay on the developers for future version of the product.

2.3.3 LifeRay

*Liferay Portal* has a similar feature set to eXo, including JSR 168 and WSRP compliance. Instead of JavaServer Faces, however, Liferay uses the more mature Struts and Tiles technologies. It runs atop a wide variety of application servers and databases and provides hooks for SSO (single sign-on) capabilities. As with eXo, professional support is available from the Liferay developers themselves, although the support costs can quickly add up.

Installing Liferay is like the other two solutions; download and unzip the bundle into the proper folder. But it is also necessary to configure some files.
Before starting it, it is necessary to run a SQL script which creates the database tables and fills it with some example content. Then, you are ready to run the starting script. The bundle comes with an astonish number of portlets. Even there is one which permit download and install more portlets, templates and themes from the community servers.

The appearance is fabulous there are almost a hundred themes to download and install automatically to change the look and feel. But if you still don’t like any one, it is possible to create new themes easily using a downloadable plugin.

It is also remarkable how easy is to add applications to new layers, just drag and dropping the desired one from the list of portlets onto the desired place in the layout.

After a while working with liferay, there are some handled exceptions in the log files (it is common that in a project of this size not everything is perfect). Despite of that, the normal use of the portal is not interrupted. So it is shown that although is not a 100% error free, it achieves to control most of the critical situations.

It seems to be that eXo Platform and Liferay are quite similar in features but if you search through their insides you will see that that is not true. It is easy to prove it; we have a look to www.cmsmatrix.org comparing Exo and Liferay.

Summary of the comparison:

<table>
<thead>
<tr>
<th>System Requirements</th>
<th>eXo Platform</th>
<th>Liferay Portal / CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Tomcat, JOnAS or any J2EE server</td>
<td>Tested compatibility with Borland ES 6.5, JBoss 4.0.2, JOnAS 4.4.3, JRun 4 Updater 3, OracleAS 10.1.2, Orion 2.0.6, Pramati 4.1, RexIP 2.5, Sun JSAS 8.01, WebLogic 8.1 SP4, WebSphere 5.1</td>
</tr>
<tr>
<td>Approximate Cost</td>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Database</td>
<td>Any (depends only on JDBC drivers)</td>
<td>DB2, Firebird, Hypersonic, InterBase, JDataStore, MySQL, Oracle, PSQL</td>
</tr>
<tr>
<td>License</td>
<td>GNU GPL</td>
<td>MIT Open Source non-restrictive business friendly license</td>
</tr>
<tr>
<td>Operating System</td>
<td>Any</td>
<td>Windows, Mac OS X, BSD, Linux, Solaris</td>
</tr>
<tr>
<td>Programming Language</td>
<td>Java</td>
<td>Java 1.4. +</td>
</tr>
<tr>
<td>Security</td>
<td>eXo Platform</td>
<td>Liferay Portal / CMS</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Captcha</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Captcha: A challenge-response system designed to defeat bots from being able to use user-only features of a system.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Content Approval</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Email Verification</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Email Verification: Does the system send an activation key to users to make sure they've entered a valid email address?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Granular Privileges</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Granular Privileges: Does the system allow read and write privileges per page or per content item basis, as well as separate privileges for other system functions?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDAP Authentication</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Login History</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SSL Compatible</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SSL Logins</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SSL Logins: Can this system be configured to switch to SSL mode (HTTPS) for logins, and then back to normal HTTP after the login?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Versioning Content</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support</th>
<th>eXo Platform</th>
<th>Liferay Portal / CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial Support</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Commercial Support: Can support be purchased from a commercial organization with trained staff members?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer Community</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Developer Community: Is there a free online developer community specifically for this product?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Public Forum</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Users Conferences</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>eXo Platform</th>
<th>Liferay Portal / CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load Balancing</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Page Caching</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Static Content Export</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>eXo Platform</td>
<td>Liferay Portal / CMS</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Drag-N-Drop Content</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Image Resizing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Is the system capable of allowing users to resize uploaded images so they need not mess around with an external image editor?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mass Upload</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Subscriptions</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Can the user subscribe to various sections of the site and receive notifications on new/updated content?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WYSIWYG Editor</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Zip Archives</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Does the system allow a user to upload a zip full of static content, which is then published to the site?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management</th>
<th>eXo Platform</th>
<th>Liferay Portal / CMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertising Management</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Does the CMS have a banner or other management system?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Asset Management</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Is there a central repository for uploading images and other files so they can be reused throughout the site?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Content Scheduling</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Does the system allow for content to be automatically added or removed from a site based upon date?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Content Staging</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Can content be created on one server and easily “pushed” to another server?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Inline Administration</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Is content edited directly in the page that it will be placed?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Online Administration</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Can the system be completely managed through a web browser?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
It is not difficult to see that Liferay far exceeds the features offered in comparison with Exo Platform:

- LR can run over multiple servers, databases and operating systems.
- Allows working with LDAP (used in the city council to manage all the users).
- It is possible to have a backstage to create content and check the composition before publishing.
- It has more security features such as Captchas to prevent bot attacks or SSL loggings in order to encrypt personal details.
- In addition, the ease of use and manageability are far too easy making Liferay the most tempting option.

2.4 Conclusion

At first sight all the portal solutions seemed to be suitable for this project, but when looking for enterprise strength, Liferay Portal stands apart. To start, it is easy to customize and handles large organizational structures, such as giving each business unit a unique look and security permissions.

For users, the GUI offers conveniences such as drag and drop portlet positioning. And Liferay comes with more than 60 JSR-168 compliant portlets covering everything from administration and content management to community features and personal tool. It is not tied into a particular app server, it also has support for organization and community structures, solid content management, and the list goes on and on.

Moreover, another thing to emphasize is the large number of awards obtained and the thousands of big organizations which have chosen Liferay as their solution. That proves its maturity and confidence level [Appendix 2].

The solution which seem to be more appropriate for this project is: Liferay
CHAPTER 3. Liferay

3.1 Introduction

Liferay, Inc. is a professional open-source company that provides free documentation and paid professional service to users of its software. Mainly focused on enterprise portal technology, the company was founded in 2000 by Brian Chan and currently has its headquarters in Los Angeles, CA.

3.2 History

Liferay was created in 2000 by Chief Software Architect Brian Chan to provide an enterprise portal solution for non-profit organizations. In 2006, the company was incorporated under the name Liferay, Inc., formalized its Germany subsidiary Liferay GmbH and named as its new CEO Bryan Cheung, the company's former Director of Business Development.

The company's enterprise portal product has been acknowledged by several notable organizations [Appendix 2]. It was recognized by EContent magazine in its "EContent 100" list of industry leaders and in 2007, InfoWorld named it a "Technology of the Year." In July 2007, they announced a partnership with ICEsoft Technologies for developing AJAX technology for their enterprise portal software.

3.3 The product

Liferay is a framework for integrating information, people and processes across organizational boundaries. It provides a secure unified access point, often in the form of a web-based user interface, and is designed to aggregate and personalize information through application-specific portlets.

All business logic is concentrated inside POJO [Appendix 1.4] (Plain Old Java Object) implementations that are looked up and instantiated by Spring [Appendix 1.9].

All data is persisted using Hibernate [Appendix 1.7] and is called through the POJO implementations. Liferay used to rely on CMP technology to achieve persistence, but switched over to Hibernate because of its speed and flexibility.

Liferay is database agnostic and can run on a variety of popular databases.
CHAPTER 4. New Gavà Televisió site

4.1 Introduction

The history of Gavà television started at the end of 1996 when the city council requested a broadcasting licence to the Generalitat de Catalunya. Around the middle of 1997, it started the construction of the studios and the some equipment was bought.

The testing broadcasting started 7th March 1998 under the direction of Joan Vila. And finally, the inauguration was 8th November 1998

After 5 year, in October 2003 Gavà television started a new stage with the new director Cristina Vicente. All the contents and sceneries are improved, five new TV shows started and the channel entered in the Net with a website.

It began a new path towards quality, professionalism and proximity.

At the end of 2004, there is an investment on the GTV notícies program and the channel becomes a member of the Xarxa de Televisions Locals. This allows Gavà Television to share contents with other 10 local televisions.

For the first time, it is shown a film once per week and some programs for the youth.

In July 2007, there is a change in the management that leads David Achell to the top of the organization. And again, this is the beginning of a renewing project which included the improvement of the corporative image, the creation of new contents and the redevelopment of the website.
This is the tree view of the new website:

This site is divided into two parts; the *frontend* and the *backend*. The frontend pages are those which are public to all users and contain all the information published by the developers.

While the backend are those which are used only by the creator to generate, update or delete content and configure the site.
4.2 Frontend Pages

These are the public pages that the users can visit and where all the information is shown.

4.2.1 Inici page

This is the main page of the site.

This page contains a videoteca youtube portlet at the top of the page which has been configured to display the last six videos uploaded by Gavà Televisió.

At the left margin, there are three RSS portlets getting the news from:

1. The Bruguers rss for local news.
2. Meteoclimatic rss for the local weather.
3. La Malla rss headlines for general news.

At the right margin, there are three images which link to other pages. In addition, there is a music player reproducing the main channel melody.

While in the middle there are two Asset Publisher portlets;

The first portlet has been configured to show the last news published in the Novetats page (last news tagged as Novetat).

The second portlet shows the own programs that are going to be on TV this day. To achieve that, we have created seven articles, one per each day of the week and they have been tagged as avui and the day of the week. For example: the article containing the programs for Monday is tagged as avui and dilluns.

Then, the Asset Publisher portlet is configured to show the articles tagged as avui and the day of the week. To do that, we had to modify the Asset Publisher portlet in order to calculate the current day of the week and add the day name to the tag list. So the daily programs are automatically updated.
4.2.2 Televisió pages

These pages tell a bit about the Gavà Television, like its history and path as well as which are their objectives and who makes up the GTV team. There is also a description of all the hi-tech equips and the location of the studios.

4.2.2.1 Història page

This page has a unique Journal Content portlet which contains some videos with text explanations about the history of Gavà Televisió. The videos are stored in the Jack Rabbit database through the Document Library portlet. And reproduced using the flash plugin JW_FLV_Media_Player.

The CSS layer has been composed using blocs of layers of two columns, the left column of 400px for the video and the right of 440px for the text.

4.2.2.2 Objectius page

This page has a unique Journal Content portlet which explains the company objectives.

The CSS layer has been composed using a structure of two columns of 420px.
4.2.2.3 Trajectòria page

This page has a unique Journal Content portlet which explains the path of the company displaying two videos and an explicative text.

The CSS layer has been composed in the same way as the història page; using a structure of two columns of 400px and 440px.

4.2.2.4 Compromís page

This page has a unique Journal Content portlet which explains the commitment that the company has over the society with some videos and texts.

The CSS layer has been composed in the same way as the història and Trajectòria pages; using a structure of two columns of 400px and 440px.

4.2.2.5 Equip humà page

This page is a composition of Journal Content portlets inside a nested portlet containing all the members of the company and another Journal Content portlet with a group picture.

All the pictures can be maximized with the light box 2 effect and have a brief description of each member:
4.2.6.6 Equipament tècnic page

This page has only one Journal Content portlet which contains a list of the company equipment.

The CSS layer has been composed using 2 columns of 420px each one:

4.2.7 On som page

This page contains two Journal Content portlets that explain where the studios are placed. Thanks to Google Maps we have included a street map in order to make it easier.

4.2.3 Programació pages

These pages contain all the information of the TV shows as well as the daily and monthly time tables.
4.2.3.1 Graella d’avui page

This page contains an Asset Publisher portlet which displays one of the seven articles (one per day of the week) that contain the daily TV programs of that week.

The uploading of the content of this page is done automatically thanks to the new portlet feature explained in the Inici page.

4.2.3.2 Graella setmanal page

This page shows the weekly TV programs through an Asset Publisher. The content of this page is published each month due to periodic changes.
4.2.3.3 Programes page

This page shows though two Journal Content portlets all the TV programs that are on Gavà televisió channel. The Programació externa section is a list of links of those programs that are not produced by Gavà Televisió studios.

This section created using a CSS grid of a five columns.

While the Programació Pròpia section is a mosaic of logos of their own programs created with a CSS grid of three columns.

Each logo is a link to the program site with its own look and feel. For example:

Each site has four pages which have a Navigation portlet to access to them:

- **The main page**: this page has a Journal Content portlet which displays a brief summary about the program and a short introductory video.

- **The team page**: this page is dedicated to the program team

- **The contact page**: this page has one or two web form portlets to give the possibility of asking or requesting whatever the user may want to know to the editorial team.

- **The video search page**: This page has two videoteca youtube portlets:
  - One configured in the reduced view to show the last videos of that program.
  - The other one is configured in the expanded view to search videos of that program.
4.2.4 Serveis page

These are the pages which manage all the information about all the services offered; for example: how to get in contact with the GTV staff, to contract publicity spaces, read some news or take part in some TV shows.

4.2.4.1 Atenció a l’espectador/a page

This page shows though a Journal Content portlet how the users can get in contact with the Gavà Televisió staff.

4.2.4.2 Publicitat page

This page contains a brief summary about advertisements and a link to download the fares in a pdf file from the document library.

4.2.4.3 Petició de còpies page

This page has a Journal Content portlet where it is explained how and under what conditions it is possible to request a copy of a filmed program. In addition, there is a web form portlet that permit the user to request for a copy.

4.2.4.4 Suggeriments page

This page contains two web form portlets whose aim is to give the possibility of sending suggestions not only to the editorial team but to the webmaster.

The reason of this page is having some feedback information from the end users.

4.2.4.5 Novetats page

This page is where all the Journal Articles tagged as Novetat are displayed using an Asset Publisher portlet.

The aim of this page is to update the last news about events, TV programs and so on.

4.2.4.6 La ciutat en directe page

This page embeds an ASP script from the Gavà Ciutat site. This script displays a web cam which can be moved by the user to focus on different parts of the town.
4.2.4.7 Enllaços page

This page uses a Journal Content portlet to display six logos that are links to other sites, for example, the Bruguers site or the Radio Sellarés site.

4.2.4.8 Vull participar page

This page gives the possibility to the end user to take part in some programs that require the audience participation.

There are two web form but it might change depending on the programs.

4.2.5 Videoteca page

This page is dedicated exclusively to the fragment of programs uploaded to youtube. It is a composition of four videoteca youtube portlets that allows the user to search and view all the videos;

- Three in the reduced view showing:
  - The most voted.
  - The most viewed.
  - The last uploaded.

- One in the expanded view to search over all the videos uploaded by the user gavatv.

4.2.6 Other pages

Finally, these are some other pages about general information.

4.2.6.1 Contacta page

This page has a grid with all the departments and a link to the E-mail a * pages. These pages are basically a web form portlet to get in contact with each department.
4.2.6.2 **Avís legal page**

In this page it has been written a legal advice. This is necessary because all the media content has broadcasting rights which need to be preserved.

4.2.6.3 **Com veure-ns page**

Here it is explained how to tune in the *Gavà Televisió* channel and also how to enjoy *Gavà Televisió* though the Net.

4.2.6.4 **E-mail a * pages**

All these pages contain one or various *webform* portlets which the end user can send an e-mail in order to get in contact with the different departments of *Gavà Televisió*.

4.3 **Backend pages**

These pages are only used by the developers in order to create and manage all the information and control all the system features.

4.3.1 **Master page**

This page is hidden to all users except to the administrator user. This is done modifying the page permissions and selecting only the specified user to see the page.

This page contains all the administration portlets apart from the gallery image portlet and document library portlet. The aim of the portlet is to configure all the portal features:

- The users and its roles.
- The communities, organizations.
- The pages, templates and structures.
- All the pictures and documents uploaded.
4.3.2 Admin page

This page is hidden to all users except to the administrator and the gavatelevisio users. The .admin page is used by the gavatelevisio to add new structures, templates and create, modify, expire or remove articles.

This is done thanks to the Journal portlet.

In addition, it has also been added the Document Library portlet in order to give to this user the possibility of upload documents to the server.

4.3.3 Tutorials page

Like the .Admin page, this one can only be seen by the Administrator and the gavatelevisio users.

It is common that when users start to work with new tools, they forget some details about how to use it. Then, they have to call the IT department and interrupt their work to ask how it was and so on.

To try to avoid this situation, we have made the most of the Camtasia Software creating video tutorials that emulate the process of the tasks that they have to do.

Right now, there are only two video tutorials because the Gavà televisió team decided to have only those tasks until they get used to it.

- One video shows how to create and remove an article and tagged it as Novetat.
- The other video shows how to modify the daily TV program articles and how to increase the article version to have different daily TV program articles.
4.4 Own portlets

These are the portlets which were necessary to implement in order to satisfy the features demanded.

4.4.1 Portlet videoteca

This portlet allows searching Liferay articles tagged as “youtube”. These articles contain youtube-videos URLs. From this URL, the video id is obtained (http://uk.youtube.com/watch?v=O5Tjya5TTgY). Not only will it be necessary to get the video but the video title, video description, video rating and so on.
Process diagram of the *videoteca* portlet:

Portlet form

- **Date fromDate**
- **Date toDate**
- **String entryName**
- **String notEntryNames**

**Title, Summary, picture, rate...**

**obtainVideoFeed()**

**Youtube query(s)**

**&**

**buscar()**

**List of articles**

**obtainVideoId()**

**TagAssetLocalServiceUtil.getAssets()**

**getEntryIds()**

**For each element**

**Max 9 ids / query**

**obtainVideoId()**
This is the package view:

Firstly the method `processAction (ActionRequest req, ActionResponse res)` receives the interval dates, the tags and the keywords for the research. Using the method `getEntryIds (long companyId, String[] entryNames)` from the `TagsEntryLocalServiceUtil` class, the portlet fills a long class vector with all the tag ids and calls the method `buscar (long[], long[], boolean, int, Date, Date, ActionRequest)`.

This method obtains all the articles containing the tags given by using the method `.getAssets (long groupId, long[] classNameIds, long[] entryIds, long[] notEntryIds, boolean andOperator, String orderByCol1, String orderByCol2, String orderByType1, String orderByType2, boolean excludeZeroViewCount, Date publishDate, Date expirationDate, int begin, int end)` from the class `TagsAssetLocalServiceUtil`. 

Portlet main class

TagsAssetYoutube class
Then, for each element is called the method `obtainVideoId (TagsAsset asset, long groupId, Date fromDate, Date toDate)` which parses the article in order to obtain the video Id and returns it only if the `article-modified-date` is between dates given. All these video ids are stored into an array.

Immediately after, it is created a String object which concat all the video ids previously obtained, in that way: `videoid OR videoid OR videoid` in order to call the method `obtainVideoFeed (String queryString)`. The maximum number of video ids per query is nine, if there were more there would be another query.

Then, by using the method `obtainVideoFeed (String queryString)`, the portlet sends the query/queries to the youtube database so this method returns a video feed which contains the video entries.

Each entry contains all the video information needed: the video title and the video description, therefore, for each video entry, it is created a `TagsAssetYoutube` object which contains the title, the description and the video id and is stored into a list object.

Finally, the jsp gets from the portlet session the `TagsAssetYoutube` list and represents each element:

- Videoid: `TagsAssetYoutube.getVideoidTitle ();`
- Title: `TagsAssetYoutube.getTitle ();`
- Description: `TagsAssetYoutube.getDesc () ;`
- Video: `http://uk.youtube.com/watch?v=videoid`
- Picture: `http://i.ytimg.com/vi/videoid/default.jpg`
As the same way that the other portlets, the view.jsp is linked with the results and the video.jsp page with the video player is done by using the portlet tag `<portlet:renderURL>:

```xml
<portlet:renderURL var="aURL">
  <portlet:param name="goto" value="video" />
  <portlet:param name="videoId" value="<%=videoId%>" />
</portlet:renderURL>

<a href="<%=aURL.toString()%>"> título </a>
```

The tag `<portlet:renderURL>` triggers the method `doView (RenderRequest req, RenderResponse res)` from de JSPPortlet class.

When this method receives the parameter `goto` with the value set to `video` dispatches the video.jsp page.

Then, the video.jsp page gets the `<String, TagAssetYouube> HashMap with the results from the portlet session.

```java
PortletSession session = ((PortletRequest) request)
.getAttribute("javax.portlet.request").getPortletSession();

HashMap<String, TagAssetYouube> lista = new HashMap<String, TagsAssetYouubes> ();

lista = {HashMap<String, TagAssetYouube> session
.getAttribute("lista", PortletSession.PORTLET_SCOPE);
```

Using the videoId parameter gets the TagAssetYouube class element from the hash map and prints its title using the `getTitle()` method.

Once obtained the title and the video Id, the video.jsp page shows the YouTube video player playing the video.

```html
String path = 'http://www.youtube.com/'} + videoId + "?enablejsapi=1&playerapiid=ytplayer";

<script type='text/javascript'>
  var params = {allowScriptAccess: "always"};
  var atts = {id: "myytplayer"};
  var obj = new YouTube({'<%=path %>', 'ytapiplayer', '425', '356', '0', null, null, params, atts);
</script>
```
When you click in the link, it will appear the youtube player:

![YouTube Player](image)

**4.4.2 Portlet videoteca youtube**

This portlet is similar to the videoteca portlet, but with this one you are able to look for all videos from a specific author rather than only those which had been previously added in the liferay database.

![Portlet Display Options](image)
This portlet gives the administrator the following possibilities by the configuration page:

- Choose different views (extended or reduced).
- Choose the author of the videos.
- Choose up to six videos to show in the reduced view.
- Choose either vertical or horizontal display using the reduced view.
- Include keywords into the user search (used for filtering).
- Sort the results by date added, view count, relevance or rating.

Portlet main class

TagsAssetYoutube Class
Process diagram of the videoteca youtubeportlet:

For each element
Title, Summary, picture, rate...

Generates a youtube query with the portlet preferences
Firstly, the method `processAction(ActionRequest req, ActionResponse res)` gets the portlet preferences (the portlet configuration parameters apart from others) and the portlet session.

The portlet preferences set by the administrator are:

- Autor.
- Ordenar [Visualización, valoración, relevancia, fecha de publicación].
- Vista [Reducida, Expandida].
- Número [1 - 6] for the reduced view.
- Disposición [Horizontal, vertical].
- Predef [Administrator’s included words]

Then, if the action is rendering the portlet, it is called the method `doView (RenderRequest req, RenderResponse res)`.

If the preference `vista` is set to `reducida` is called directly the method `buscar (String keywords, PortletPreferences prefs)` with a “*” as a keyword, in order to look for all videos. Then it is dispatched the page `viewsmall.jsp` which shows the search results.

Nevertheless, if the preference `vista` is set to `expandida`, the portlet only dispatches the page `view.js`. This page contains a form to let the user search videos.

While if the action is searching (only available from the extended view), the `processAction(ActionRequest req, ActionResponse res)` gets the keywords and calls the method `buscar(String keywords, PortletPreferences prefs)`.

What the method `buscar(String keywords, PortletPreferences pref)` basically does is to build a query using the youtube API with the keywords and the display preferences.

```java
YouTubeQuery query = new YouTubeQuery (new URL ("http://gdapi.youtube.com/feeds/api/videos");
query.setAuthor (prefs.getValue ("autor", ""));
query.setOrderBy (prefs.getValue ("ordenar", ""));
query.setTime (YouTubeQuery.Time.ALL_TIME);
if (reduced view)
  query.setMaxResults (prefs.getValue ("numero", ""));
else
  query.setMaxResults (50); // max permitted

query.setVideoQuery (keywords + " " + prefs.getValue ("predef", ""));
query.setInfluencer (false);
```
Once executed the service, it returns a VideoFeed class object with all the video results. Then a list of TagsAssetYoutube class objects is filled by the setter methods

```java
TagsAssetYoutube assetYoutube = null;
for (each entry)
    assetYoutube = new TagsAssetYoutube();
    assetYoutube.setTitle(entry.getTitle().getPlainText());
    assetYoutube.setDesc(entry.getMediaGroup().getDescription().getPlainTextContent());
    assetYoutube.setVideoId(entry.getVideoId()); // previously parsed
    assetYoutube.setViewCount(entry.getStatistics().getViewCount());
    assetYoutube.setPublished(entry.getPublished());
    list.add(assetYoutube);
```

Once the list has already been filled, it is stored into the portlet session to be able to get it from the jsp page.

```java
PortletSession session = req.getPortletSession();
session.setAttribute('listaVideos', list);
```

Results of a search
The method `doView (RenderRequest req, RenderResponse res)` from de JSPPortlet class is responsible for rendering the portlet by dispatching the corresponding jsp page.

Depending on the preference `vista` this method dispatches either the `view_small.jsp` or the `view.jsp`.

The way is linked the `view.jsp` page with the results and the `video.jsp` page with the video player is done by using the portlet tag `<portlet:renderURL>`:

```html
<portlet:renderURL var="aURL">
    <portlet:param name="goto" value="video" />
    <portlet:param name='videoId' value='"><%=videoId%>' />
<portal:renderURL>

<a href='"<%=aURL.toString()%>"'> titulo </a>
```

The tag `<portlet:renderURL>` triggers the method `doView (RenderRequest req, RenderResponse res)` from de JSPPortlet class.

If this method receives the parameter `goto` with the value set to `video` dispatches the `video.jsp` page. Then, the `video.jsp` page gets the `<String, TagAssetYoutbe>` HashMap with the results from the portlet session.

```java
PortletSession session = ((PortletRequest) request)
    .getAttribute("javax.portlet.request").getPortletSession();

HashMap <String, TagAssetYoutbe> lista =
new HashMap <String, TagsAssetYoutube> ();

lista = (HashMap <String, TagAssetYoutube>) session
    .getAttribute("lista", PortletSession.PORTLET_SCOPE);
```

Using the videoId parameter gets the TagAssetYoutbe class element from the hash map and prints its title using the `getTitle ()` method.

In addition, the `video.jsp` page shows the YouTube video player playing the video.

```html
String path = "http://www.youtube.com/w + videoid + "&enablejsapi=1&playerapid=ytplayer";

<script type='text/javascript'>
        var params = {allowScriptAccess: "always"};
        var atts = {id: "myytplayer"};
        swfobject.embedSWF
            ("" + path + ", 'ytapiplayer', '125', '356', '8', null, null, params, atts);
</script>
```
CHAPTER 5. New El Bruguers site

5.1 Introduction

The el bruguers is the local newspaper. With more than 40 years of history, it is published monthly.

In 2003 with the Internet revolution, the city council took part into a modernization project which included the creation of a digital version of the el bruguers: www.elbruguers.cat had born.

This is the tree view of the new El bruguers site:
This site is divided, like the Gavà television site, into two parts: the frontend and the backend. The frontend pages are those which are public to all users and contain all the information published by the developers.

While the backend are those which are used only by the creator to generate, update or delete content and configure the site.

### 5.2 Frontend pages

These are the public pages that the users can visit and where all the information is shown.

#### 5.2.1 Portada page

This is the main page of the digital newspaper. The editorial staff is in charge of composing the front page after having written the articles through the administration page.

This page varies depending on the number and the large of the news of each day.

The front page is created using the portadas programadas pages and the scheduler portlet which copies the corresponding page when it is the time.

The editorial staff can only add Journal Content portlets to the centre column. The rest of portlets (which contain the banners and menus) are only administrable by the administrators.

The editorial team has the possibility of choosing between two possible templates; a headline template or a normal template. These templates change the size of the text to emphasize the importance of them.

In addition, there is also the possibility of tagging each article (it is necessary to tag the article with at least one tag).

These tags are taken by calling the corresponding service in the velocity template and are printed over the article to categorize them:
How to obtain the tags calling a service in velocity

Because the articles are usually large texts, it is only displayed the headline and the lead-in with a small picture and when you want to read the whole article, you only have to click in the headline and it will be expanded.
This is done checking the portlet window state;

- If it is **normal**, the template displays a small view of the article.

- But when you click in the headline, it is called the URL which will render the portlet in a **maximized** state. Then, the template displays the whole article.

```java
// Inicialización de variables
// maximized window state URL
#set ($url-max=$request.get('render-url-maximized'))
// normal window state URL
#set ($url-nor=$request.get('render-url-normal'))
// current window state
#set ($window-state=$request.get('window-state'))

#if ($window-state=="normal")
...
<a class="link" title="Ampliar" href="${url-max}vista=ampliada">HEADLINE</a>
</a>

Journal Template - How to maximize and minimize the articles

### Hemeroteca page

The aim of this page is to permit the user to search old news stored in the database.

Basically, this page only contains a **search** portlet. But this portlet has suffered a little modification, instead of searching either for title or description, now the keyword will be searched into the whole article (title, description and content).

```java
// Modificación para permitir pasar palabras claves desde los templates XSLT
String keyword = DAOParamUtil.getLike(req, "keyword");
if (keyword!=null){
    content = keyword;
    andOperator = true; //AND entre las condiciones de title description, content
    ...
}
```

Modifications in the class: GetArticlesAction.java
5.2.3 Esports page

This page has exactly the same structure then the front page, but unlike this one, the Esports page only contains articles related to sport.

5.2.4 Opinió política page

This section of the site is reserved for the members of all political parties of Gavà. In essence, it is composed by six articles where each of them can write their thoughts, purposes and express their opinions.

5.2.5 Edició impresa page

Apart from the digital newspaper, there is also one monthly printed version. This printed version is stored in the database in a PDF format file with a brief summary.

Thanks to the búsqueda impresa portlet, it is possible to search by number edition or keyword and download a copy (see own portlet section to see how it works).
5.2.6 L’enquesta page

This page contains two poll portlets that allow the editorial team to suggest questions about local events, government decisions or just things to ponder.

With this page the users can send feedback information to the team who will focus the news field depending on the demand.

5.2.7 Contacta page

You have also the chance of getting in contact with the staff (the edition team and the designer team) in order to contribute to the site, express your opinions, complain, request more information about some news, or even alert if something works improperly.

The user only needs to fill the web form to send an e-mail to the administrators.

5.2.8 Subscripcions al diari and Subscripcions a l’agenda pages

This pair of pages contains an iframe portlet which embed the subscription services to the agenda and to the news. This service is given by the old portal.

It hasn’t been implemented with the new technology because those subscriptions affect to almost the rest of the portals, so we have decided to wait until having more portals in Liferay to transfer this service.
5.2.9 Farmacies and telèfons pages

The content of these pages is also displayed using iframe portlets because it is managed through the agenda application. Like the subscription pages, they will be migrated in the future.

5.2.10 7 dies pages

These group of seven pages (one per each day of the week) store the front pages of the previous week. In this way, the user can visit the last seven front pages.

(The process of overriding the corresponding pages is explained in the portades programades section).
5.3 Backend pages

These pages are only used by the developers in order to create and manage all the information and control all the system features.

5.3.1 Portades Programades pages

The aim of these pages is to create future front pages for weekends, bank holidays, etc and publish them.

The team creates the exact front pages desired in one of the six layouts offered. Then, using the scheduler portlet indicates the date and time when they want to publish each front page. When the time arrives the page is published automatically. To obtain this effect there has been some modifications in the LayoutLocalServiceImpl.java class:

When a layout is updated, it stores as an attribute the date when is supposed to be published.

The pages that will substitute the front pages are identified by the friendly URL, they are /portadaprograma*.

When the stored publishing date is the current date, the method which imports the layouts:

1. calculates the day of the week that the layout was published

2. Once calculated, it is stored into the layout with that name in the 7 dies section; (if it is the front page of Monday, it will override the layout /monday → /dilluns).

3. Overrides the front page with the newest (based in the stored publishing date).
5.3.2 Administració page

This is the page from where the editorial team creates all the articles. It contains:

- A Document Library portlet to upload all the PDF’s and attached documents.
- An Image Gallery portlet to upload all images.
- A Journal portlet to add and remove articles.
- A Scheduler portlet for the future front pages.

5.4 Own portlets

This is the portlet which was necessary to implement in order to satisfy the features demanded.

5.4.1 Portlet búsqueda impresa

This portlet allows searching articles which contain the printed version of the monthly newspaper *El Briguers* in pdf.
Process diagram of the *búsqueda impresa* portlet:

This is the package view:

**Portlet main class**

```java
import declarations

/* JSPPortlet 1.3 (ASCII-kw) */

class PdfArticles {
    private String pdfPath;
    private String imagePath;
    private String index;
    private String title;
    private String articleId;
    private String getArticleId();
    private String getPDFPath();
    private String getImagePath();
    private String getTitle();
    private String getIndex();
    private String setArticleId(String);
    private String setPDFPath(String);
    private String setImagePath(String);
    private String setTitle(String);
    private String setIndex(String);
    private String init();
    private String doDispatch(RenderRequest, RenderResponse);
    private String doDR(RenderRequest, RenderResponse);
    private String doHelp(RenderRequest, RenderResponse);
    private String doView(RenderRequest, RenderResponse);
    private String processAction(ActionRequest, ActionResponse);
    private String getPDFElements(JournalArticle, long, String, String);
    private String getArticles(long, long, Date, Date, String, String, String);
    private String include(String, RenderRequest, RenderResponse);
    private String editJSP : String;
    private String helpJSP : String;
    private String viewJSP : String;
    private String _log : Log;
}
```

**PdfArticles class**

```java
import declarations

package com.example.jsp.portlet;

public class PdfArticles {
    private String pdfPath;
    private String imagePath;
    private String index;
    private String title;
    private String articleId;
    private String getArticleId();
    private String getPDFPath();
    private String getImagePath();
    private String getTitle();
    private String getIndex();
    private String setArticleId(String);
    private String setPDFPath(String);
    private String setImagePath(String);
    private String setTitle(String);
    private String setIndex(String);
    private String init();
    private String doDispatch(RenderRequest, RenderResponse);
    private String doDR(RenderRequest, RenderResponse);
    private String doHelp(RenderRequest, RenderResponse);
    private String doView(RenderRequest, RenderResponse);
    private String processAction(ActionRequest, ActionResponse);
    private String getPDFElements(JournalArticle, long, String, String);
    private String getArticles(long, long, Date, Date, String, String, String);
    private String include(String, RenderRequest, RenderResponse);
    private String editJSP : String;
    private String helpJSP : String;
    private String viewJSP : String;
    private String _log : Log;
}
```
Firstly, the `processAction(ActionRequest req, ActionResponse res)` method gets the period of time, the keywords and/or the edition number for searching.

Then, it is called the method `getArticles(long companyId, long groupId, Date displayDateGT, Date displayDateLT, String keywords, int numBruguers, String structureId)`.

- **CompanyId**: is the id of the company
- **GroupId**: is the community’s id.
- **DisplayDateGT**: from date.
- **DisplayDateLT**: to date.
- **Keywords**: user keywords for searching.
- **NumBruguers**: it is also possible to search a specific edition number.
- **structureId**: “EDICION-IMPRESA” is the structure id for those articles.

That method returns either a list of articles containing the keyword and whose modified date is between the dates given or the article which contains the specified number of the printed newspaper (if exists) by using the method `JournalArticleLocalServiceUtil.search (long companyId, long groupId, String articleId, double version, String title, String description, String content, String type, String structureId, String templateId, Date displayDateGT, Date displayDateLT, boolean approved, boolean expired, Date reviewDate, boolean andOperator, int begin, int end, OrderByComparator obc)`.

Then, for each element of the list, is called the method `getPdfElements(JournalArticle article, long groupId, String imagePath, String mainPath)`;

- **Article**: each article found.
- **GroupId**: is the community’s id.
- **imagePath**: the path of the image gallery (usually /image)
- **mainPath**: the path of the document library (usually /c/document_library/)

The aim of this method is to parse the article content in order to obtain each element of the article.

The `getPdfElements(JournalArticle article, long groupId, String imagePath, String mainPath)`; method returns a string array with the image path of the picture, the pdf path and the edition summary.

Each element of the String array is stored in a list as a PdfArticle class object and finally the list with all the results is stored into the portlet session to be able to recover it from the jsp.
Once you have clicked in the *buscar* button, it will appear the results obtained:

```
if (pdfArticlesList != null)
for (each elemento)
{
  <div id="ingreso">
    <portlet:renderURL var="contentURL">
      <portlet:param name="articleId" value="<%= pdfArticlesList.get(elemento).getArticleId()%>" />
      <portlet:param name="goto" value="1"/>
    </portlet:renderURL>
    <div id="ingreso-titulo">
      <%= pdfArticlesList.get(elemento).getTitulo()%>
    </div>
    <div id="ingreso-images">
      <a href="<%= contentURL %>">
        <img src="<%= pdfArticlesList.get(elemento).getImagePath()%>" width="160px"/>
      </a>
    </div>
  </div>
//end for
//end if
```

When the method `doView (RenderRequest req, RenderResponse res)` receives the parameter `goto` set to a value different that `null` (in that case `1`) it dispatches the `view_article_content.jsp` page.
The `view_article_content.jsp` is responsible for getting the article content (the title, the summary, the image and the PDF file to download)

```jsp
<%= request.getAttribute(WebKeys.JOURNAL_ARTICLE_CONTENT) %>
```

Clicking in the image, it will be shown the detailed view:

---

Detail of a result – `view_article_content.jsp`
CHAPTER 6. CONCLUSIONS

6.1 Conclusions

With this project we have achieved to search, evaluate and select a CMS platform and a web application platform and the execution of two portals using Liferay.

At the end of the project we have achieved the main objective exceeding the initial expectations. The success of the solution chosen has permitted the possibility of creating not only one but two portals www.gavatv.cat and www.elbruguers.cat.

With this project I have learned to get a necessity and search the best solution for it by looking for information, testing different ones, reading comments and opinions of other users, reading specialist websites and so on.

Open source projects:

Working with open source projects offers the possibility of developing huge software maintained by the community. This option has loads of advantages; you can adapt the project to your specific necessities. In addition, the whole community takes part into the project so it is constantly being extended, revised and improved. And of course it is free.

The main problem I have found was to find the proper information about the product. Usually, the community works hard into the project but forgets to document their work. Therefore, sometimes it is very tough to understand how it works. But when you start to get used to working and understand its ins and outs, you begin to enjoy it and be able to “create”.

Personally, I have experienced a situation which can be described with the learning process graphic. At the beginning I was so lost and I had to read a lot, post a lot in the community forums and almost everything took a long time. But then I saw the light and began to fit all the pieces; I was ready to create.
**Working on CSS:**

Although I haven’t talked a lot about CSS’s, it is simply a world apart.

Each browser interprets most of the properties in a different way. Even some are ignored (like `height` property). This is because not all of them follow the standard, so it is necessary to find patterns which work with all of them or even create different properties depending on the browser. Therefore, we decided to buy a CSS patterns book [3] which guided us through this complicated world.

**General opinion:**

This project has helped me to understand that open source solutions are a very interesting option to consider.

Moreover, because I had done it inside a company I have learned how to work side by side with a professional team. I have also learned how to deal with other departments that not always had the same point of view and whose opinions and ideas contributed to improve the product.

This project not only consists in this bunch of pages and a pair of websites but a set of new experiences, new techniques and procedures which I couldn’t have learnt in any book.

**6.2 Future lines**

This project is only the first step of a large modernization project of the city council. Until now, we have only chosen the framework and developed two portals to test its features.

With this project, we have confirmed that the provided solution is suitable to the necessities. So now is the time to redevelop the rest of the municipal portals (still in ASP) with Liferay.

There is also another pair of things that need to be redeveloped, the local intranet and the electronic processing.

Because this project has been developed under an open source platform, it is easy joint it with other open source projects to improve the features a lot more. For example with:

- OpenXava
- Ice faces
**OpenXava** is a web application framework for developing business applications in an effective way. It allows rapid and easy developing of CRUD modules and report generation, but also it’s flexible enough to develop complex real life business applications as, customer relationship, invoicing, warehouse management, etc.

**ICE faces** is an open source Ajax framework that enables Java application developers to easily create and deploy server-based rich Internet application (RIA) in Java.

### 6.3 Environmental review

Until now, the administration wastes loads of paper in all the bureaucratic proceedings, dossiers and photocopies. The fact that this project provides electronic tools to administer all these proceedings will make possible to save money, space and time.
BIBLIOGRAPHY


[w5] http://findarticles.com/p/articles/mi_m0EIN/is_2005_June_20/ai_n13821536 (Software Security Analysis website)


[w9] content.liferay.com/4.3/doc/devel/liferay_4_portlet_development_guide/onepage/ (Portlet development guide)


[w16] http://www.icefaces.org/JForum/posts/list/9187.page (Forum thread about liferay vs jboss vs exo)

[w17] http://saloon.javaranch.com/cgi-bin/ubb/ultimatebb.cgi?ubb=get_topic&f=85&t=000322 (Forum thread about liferay vs jboss)

[w18] http://www.experts-exchange.com/Programming/Languages/Java/Q_21211663.html (Forum thread about Exo vs Liferay)

[w19] http://www.jboss.org (JBOSS website)


APENDIXS

TITLE: Research evaluation and selection of a CMS (Content Management System) platform and a web applications platform for the Gavà's city council.

DEGREE: Diploma in telematics (First cycle).

AUTHOR: Rubén Marín Garrote.

DIRECTOR: Manuel Antonio Marín Vivas.

CO-DIRECTOR: Oscar Caro.

SUPERVISOR: Dolores Royo Vallés.

APENDIX 0. Main concepts

AP0.1 What is a Portal?

A portal is web-based application for integrating information, people and processes across organizational boundaries. It provides a secure unified access point, often in the form of a web-based user interface, and is designed to aggregate and personalize information through application-specific portlets. One outstanding feature of portals is the de-centralized content contribution and content management, which keeps the information always updated.

The fundamental features are:

- Single Point of Entry: portals can provide single sign-on capabilities between their users and various other systems. This requires a user to authenticate only once. Access control lists manage the mapping between portal content and services over the portal user base.

- Integration: the connection of functions and data from multiple systems into new components.

- Federation: the integration of content provided by other portals, typically through the use of Web Services for Remote Portlets (WSRP) or similar technologies.

- Personalization: Users can customize the look and feel of their environment. Even master users can edit and design their own web sites. They can also choose the specific content and services they prefer.

- Permission: the ability for portal administrators to limit specific types of content and services access. For example, a company’s proprietary information can be entitled for only company employee access.
AP0.2 What is a Portlet container?

A portlet container runs portlets and provides them with the required runtime environment. A portlet container contains portlets and manages their life cycles. It also provides persistent storage mechanisms for the portlet preferences. A portlet container receives requests from the portal to execute requests on the portlets hosted by it. A portlet container is not responsible for aggregating the content produced by the portlets; the portal itself handles aggregation.

A portal and a portlet container can be built together as a single component of an application or as two separate components of a portal.

The portlet container must ignore all references to custom portlet modes that are not supported by the portal implementation, or that have no mapping to portlet modes supported by the portal.

Liferay's core is a portlet container, and this container is only responsible for aggregating the set of portlets that are to appear on any particular page. This means that all of the features and functionality of your portal application must be in its portlets.

AP0.3 What is a Portlet?

Portlets are small web applications that run in a portion of a web page. Portlets produce fragments of code that are aggregated into a portal page. The heart of any portal implementation is its portlets, because portlets are where the functionality of any portal resides.

Portlet applications, like servlet applications, have become a Java standard which various portal server vendors have implemented. The JSR-168 [1] standard defines the portlet specification. A JSR-168 standard portlet should be deployable on any JSR-168 portlet container.
JSR 168's goals are:

- Define the runtime environment, or the portlet container, for portlets.
- Define the API between portlet container and portlets.
- Provide mechanisms to store persistent data for portlets.
- Provide a mechanism that allows portlets to include servlets and JSP (JavaServer Pages).
- Define a packaging of portlets to allow easy deployment.
- Run JSR 168 portlets remotely using the protocol WSRP [2].

Web clients interact with portlets via a request/response paradigm implemented by the portal. Usually, users interact with content produced by portlets by, for example, following links or submitting forms, resulting in portlet actions being received by the portal, which then forward to the portlets targeted by the user’s interactions.

**AP0.3.1 Portlet modes**

A portlet mode indicates the function a portlet is performing. Normally, portlets perform different tasks and create different content depending on the function they are currently performing. A portlet mode advises the portlet what task it should perform and what content it should generate. When invoking a portlet, the portlet container provides the current portlet mode to the portlet. Portlets can programmatically change their portlet mode when processing an action request.

The Portlet Specification defines three portlet modes, VIEW, EDIT, and HELP. The PortletMode class defines constants for these portlet modes.

The availability of the portlet modes, for a portlet, may be restricted to specific user roles by the portal. For example, anonymous users could be allowed to use the VIEW and HELP portlet modes but only authenticated users could use the EDIT portlet mode.
AP0.3.2 Defining portlet modes support

Portlets must describe within their definition, in the deployment descriptor, the portlet modes they can handle for each markup type they support. As all portlets must support the VIEW portlet mode, VIEW does not have to be indicated. The portlet must not be invoked in a portlet mode that has not been declared as supported for a given markup type.

The following example shows a snippet of the portlet modes a portlet defines as supporting in its deployment descriptor definition:

```xml
... 
<supports>
  <mime-type>text/html</mime-type>
  <portlet-mode>edit</portlet-mode>
  <portlet-mode>help</portlet-mode>
</supports>

<supports>
  <mime-type>text/vnd.wap.wml</mime-type>
  <portlet-mode>help</portlet-mode>
</supports>
...
```

For HTML markup, this portlet supports the EDIT and HELP portlet modes in addition to the required VIEW portlet mode. For WML markup, it supports the VIEW and HELP portlet modes.
APENDIX 1. Glosary

AP1.1 Apache Velocity

Velocity is a Java-based template engine that provides a simple yet powerful template language to reference objects defined in Java code. Its aim is to ensure clean separation between the presentation tier and business tiers in a Web application.

AP1.2 Content Management System

A CMS is computer software used to create, edit, manage, and publish content in a consistently organized fashion. CMSs are frequently used for storing, controlling, versioning, and publishing industry-specific documentation such as news articles, operators’ manuals, technical manuals, sales guides, and marketing brochures. The content managed may include computer files, image media, audio files, video files, electronic documents, and Web content.

AP1.3 Enterprise Java Beans (EJB)

EJB is a managed, server-side component architecture for modular construction of enterprise applications.

The EJB specification is one of the several Java APIs in the Java Platform, Enterprise Edition. EJB is a server-side component that encapsulates the business logic of an application.

The EJB specification intends to provide a standard way to implement the back-end 'business' code typically found in enterprise applications (as opposed to 'front-end' user-interface code). Such code was frequently found to reproduce the same types of problems, and it was found that solutions to these problems are often repeatedly re-implemented by programmers.

Enterprise Java Beans were intended to handle such common concerns as persistence, transactional integrity, and security in a standard way, leaving programmers free to concentrate on the particular problem at hand.
Accordingly, the EJB specification details how an application server provides:

- **Persistence**
  
  *persistence* refers to the characteristic of data that outlives the execution of the program that created it. Without this capability, data only exists in RAM, and will be lost when the memory loses power, such as on computer shutdown.

- **Transaction processing**
  
  *transaction processing* is information processing that is divided into individual, indivisible operations, called *transactions*. Each transaction must succeed or fail as a complete unit; it cannot remain in an intermediate state.

- **Concurrency control**
  
  *concurrency control* ensures that correct results for concurrent operations are generated, while getting those results as quickly as possible.

- **Events using Java Message Service**
  
  The *Java Message Service* (JMS) API is a Java Message Oriented Middleware (MOM) API for sending messages between two or more clients. JMS is a part of the Java Platform, Enterprise Edition, and is defined by a specification developed under the Java Community Process as JSR 914.

- **Naming and directory services (JNDI)**
  
  The *Java Naming and Directory Interface* (JNDI) is an API for directory service that allows clients to discover and lookup data and objects via a name. Like all Java APIs that interface with host systems, JNDI is independent of the underlying implementation.

- **Security (Java Cryptography Extension (JCE) and JAAS)**
  
  *Java Authentication and Authorization Service* or JAAS is a Java security framework for user-centric security to augment the Java code-based security.

- **Deployment of software components in an application server**

- **Remote procedure calls using RMI-IIOP**
  
  *Remote procedure call* (RPC) is a technology that allows a computer program to cause a subroutine or procedure to execute in another address space (commonly on another computer on a shared network) without the programmer explicitly coding the details for this remote interaction.

  **Java Remote Method Invocation** ("Java RMI") technology run over Internet *Inter-Orb Protocol* ("RMI-IIOP") delivers Common Object Request Broker Architecture (CORBA) distributed computing capabilities to the Java 2 platform. Java RMI over IIOP was developed by Sun and IBM. The joint work by Sun and IBM to implement Object Management Group (OMG) standards demonstrates the spirit of collaboration that continually moves the Java platform forward.

- **Exposing business methods as Web Services.**
  
  A *Web service* (also Web Service) is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network." Web services are frequently just Web APIs that can be
accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

Additionally, the Enterprise JavaBean specification defines the roles played by the EJB container and the EJBs as well as how to deploy the EJBs in a container.

**AP1.4 POJO applications**

This is an acronym for **Plain Old Java Object**, and is favoured by advocates of the idea that the simpler the design, the better. The name is used to emphasize that the object in question is an ordinary Java Object, not a special object, and in particular not an Enterprise Java Bean.

**AP1.5 Object-relational mapping**

**ORM**, **O/RM**, and **O/R mapping** is a programming technique for converting data between incompatible type systems in relational databases and object-oriented programming languages. This creates, in effect, a "virtual object database" which can be used from within the programming language.

**AP1.6 Software framework**

In broad terms is a code library that is designed to help software development. The idea is that all of the tedious, low-level details of creating an application are already in a reusable package. For a developer, this means you can spend your time worrying about specific problems related to your application, and not the actual building of the code behind it. For example, a developer can quickly and easily access a database using a framework, rather than having to write all of the code required to accomplish this simple task.

**AP1.7 Hibernate**

Hibernate is an object –relational mapping (ORM) library for the Lava language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves Object-Relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

Hibernate is a free, open source Java package that makes it easy to work with relational databases. Hibernate makes it seem as if your database contains plain Java objects like you use every day, without having to worry about how to get them out of (or back into) mysterious database tables. It liberates you to focus on the objects and features of your application, without having to worry about how to store them or find them later.
Hibernate's primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate also provides data query and retrieval facilities. Hibernate generates the SQL calls and relieves the developer from manual result set handling and object conversion, keeping the application portable to all SQL databases.

Hibernate provides transparent persistence for Plain Old Java Objects (POJOs).

Hibernate provides a dirty checking feature that avoids unnecessary database write actions by performing SQL updates only on the modified fields of persistent objects.

Hibernate can be used both in standalone Java applications and in Java EE applications using servlets or EJB session beans.

**AP1.8 Apache JackRabbit**

Jackrabbit is a fully conforming implementation of the Content Repository for Java Technology API (JCR). A content repository is a hierarchical content store with support for structured and unstructured content, full text search, versioning, transactions, observation, and so on. Typical applications that use content repositories include content management, document management, and records management systems.

Version 1.0 of the JCR API was specified by the Java Specification Request 170 (JSR 170)

**AP1.9 Spring Framework**

This is an open source framework created to address the complexity of enterprise application development. One of the chief advantages of the Spring framework is its layered architecture, which allows you to be selective about which of its components you use while also providing a cohesive framework for J2EE application development.

Spring is essentially a technology dedicated to enabling you to build applications using POJOs. This desirable goal requires a sophisticated framework, which conceals much complexity from the developer.

Spring framework functionality can be used in any J2EE server and most of it also is adaptable to non-managed environments. A central focus of Spring is to allow for reusable business and data-access objects that are not tied to specific J2EE services. Such objects can be reused across J2EE environments (Web or EJB), standalone applications, test environments, and so on, without any hassle.
AP1.10 Universal Description, Discovery and Integration (UDDI)

UDDI is a platform-independent, XML-based registry for businesses worldwide to list them on the Internet. UDDI enables to publish service listings and discover each other and define how the services or software applications interact over the Internet.

AP1.11 AJAX

Ajax (asynchronous JavaScript and XML), or AJAX, is a group of interrelated web development techniques used for creating interactive web applications or rich Internet applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page.

AP1.12 CRUD operations

Create, Read, Update and Delete are the four basic functions of persistent storage.

AP1.13 Web services

A Web service is defined by the W3C as a software system designed to support interoperable machine-to-machine interaction over a network. Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

AP1.14 WSRP Standard

Web Services for Remote Portlets is a network protocol standard designed for communications with remote portlets. The WSRP specification defines a web-service interface for interacting with presentation-oriented web services.

The WSRP specification does not make any statements as to implementation. Java's portlet specification, JSR 168, and WSRP are not competing technologies. JSR 168 may be used to define a portlet, and WSRP may be used to define a portlet's operations to remote containers. JSR 168 portlets and WSRP may be used together to define a portlet and to provide remote operations.
Scenarios that motivate WSRP functionality include:

- Content hosts, such as portal servers, providing portlets as presentation-oriented web services that can be used by aggregation engines;

- Content aggregators, such as portal servers, consuming presentation-oriented web services provided by portal or non-portal content providers and integrating them into a portal framework.
APENDIX 2. Awards and web examples

AP2.1 LifeRay Awards

Liferay Portal Wins InfoWorld’s Best Open Source Portal Award for 2008

There are a number of well-known open source portal solutions. But when looking for enterprise strength, Liferay Portal stands apart.

EContent 100 Award (November 2007)

Liferay 4.2 was again named to EContent magazine’s list of the 100 most influential and innovative companies in the digital content industry in the Intranets & Portals category.

InfoWorld BOSSIE Award 2007

"Putting aside other standards, which our four portal finalists all follow, the compelling argument for winner Liferay Portal can be summed up by mentioning usability, architecture, security, integration, and portlets. Liferay's intuitive user experience, featuring drag-and-drop portlet arrangement and management, is tops."

InfoWorld 2007 Technology of the Year Awards: Best Open Source Portal!

"Liferay Portal 4.0 rivals many commercial portal platforms in both functionality and technology..."
**EContent 100 Award (December 2006)**

Liferay 4.0 was named to EContent magazine's list of the 100 most influential and innovative companies in the digital content industry in the Intranets & Portals category. Unlike many other industry accolades, companies may not nominate themselves for inclusion in the EContent 100. EContent magazine's professional staff members generate their own lists, which are then debated and ultimately assembled into the one hundred finalists.

**eWeek Benchmark (July 2006)**

eWeek benchmarking test, a J2EE stack built on CentOS Linux, Apache, and Liferay (labeled “Linux J2EE in comparison charts) was measured as delivering the highest average hits per second, the second highest average throughput per second, and the third highest average document download time among eight different portal solutions, including JBoss Portal and Microsoft SharePoint.

**CIO 100 Award (2006)**

Goodwill Industries won a 2006 CIO 100 award for the Web-based knowledge portal Liferay and Goodwill built using Liferay 3.6.1. The system provides an online community for employees that allows for collaboration to solve business problems. It also offers access to training materials.

“Participation in the online community has increased steadily, and user registration is higher than any system Goodwill has ever had. Monthly document downloads have grown more than tenfold since the release.”

**eWeek Review (October 2005)**

eWeek reviewed Liferay 3.6.1.

“It’s long been commonplace to find open-source portals that rival or better their commercial counterparts in features and capabilities. And for a few years now, open-source portals have featured intuitive and flexible browser-based administration interfaces
All of the above is true for the open-source Liferay Portal, but its browser-based administration and content creation interface goes much further, providing in-depth interactivity and capabilities that one usually finds only in desktop applications—providing this breadth of capabilities across all current-generation Web browsers.

**AP2.2 Some websites running on Liferay**

**National Center for Supercomputing Applications (NCSA)**

One of the five original centers in the National Science Foundation’s Supercomputer Centers Program (www.ncsa.uiuc.edu)

“We have been very happy with Liferay portal since 2004 and hope to see it extend to other areas of our organization.”

-- Jim Myers, Associate Director for Cyberenvironments, NCSA

**The Consortium for the Commercial Promotion of Catalonia (COPCA)**

Catalan government’s tool to promote the internationalization of Catalan companies (http://www.anella.cat/)

“Thanks to Liferay, COPCA developed a first class technological platform while minimizing costs. Because Liferay is an open source solution, we are able to quickly respond to business requirements, easily developing small customizations over standard components. Two years after choosing Liferay Portal, we can still say we made the right selection and continue to work with this platform.”

--José Oliver, COPCA’s Digitalization Manager.

**French Armaments Procurement Agency (DGA)**

”The Espace Partenaire” collaborative Platform (EPNG) for the French defense community (http://www.ixarm.com/Platform-hub)

Liferay is now a serious player in defense space. Many projects are looking at EPNG as a critical reference for an infrastructure portal.

--Geoffray Gruel, Director, Ippon Technologies
EducaMadrid

Madrid school system (Spain) (http://www.educa.madrid.org/)

"Its open source nature has made possible a big deal of changes and adaptation which would have been impossible or very costly with a commercial product."

- Jorge Ferrer, Germinus

Networked Environmental Information System for Global Emissions Inventories (NEISGEI)

System for Global Emissions Inventories initiative (NEISGEI – pronounced ‘nice guy’) with the goal of applying the latest information science and technologies to open web access of disparate emissions data and analysis tools. (www.neisgei.org)

“After reviewing a number of other options, we believe Liferay remains the leading portal solution out there. Its dedication to open standards is a key factor in using the Liferay portal for community driven air quality work.”

– Stefan Falke, Research Professor of Energy, Environmental and Chemical Engineering at Washington University in St. Louis

Christian Science Monitor

International daily newspaper (http://www.csmonitor.com/)

“We strongly believe in leveraging the expertise of our open source partners and community. Liferay has been evolving and innovating in the open source portal market for quite some time and the strength and maturity of its community is reflected in the maturity of the product’s features.”

Russ Danner, Lead Architect, Christian Science Monitor
Goodwill Industries

One of the world’s largest nonprofit providers of education, training, and career services for the underprivileged. (International) (http://www.goodwill.org/)

"Liferay Portal was by far the most cost effective and efficient portal technology we considered."

--Steve Bergman, CIO, Goodwill Industries International

PanTech

2nd largest handset manufacturer in South Korea (Korea) (http://www.pantechusa.com/)

"Liferay Portal has been a key enabler of our marketing efforts in Pantech’s new North American market. Having kept pace with 12 product releases in 2006 alone, the technology is clearly nimble and powerful but also simple enough to maintain with minimal support. We foresee Liferay as an integral part of Pantech’s IT investment in the future"

– Philip Hyun, Project Manager, URi

CAP GmbH

Internal solution for Deutsch Telecom (Germany) (http://www.cap.de/)

"With CAPassist, Liferay Portal and Ancud IT Services have been leveraged to deliver a workflow portal that structures and accelerates the internal processes of CAP GmbH. Based on open source components, the portal empowers CAP GmbH’s various divisions and their external service providers to collaborate effectively"

- Christoph Kalka, Customer Service, CAP GmbH
**HanseMerkur**

3rd oldest Insurance company in Germany (Germany) (http://www.hansemerkur.de/)

“Liferay’s turnaround on our project was exceptional. Thanks to their ideas and input, the end result is an excellent working relationship and a quality site that exceeded HanseMerkur’s expectations.”

- Dr. Horst Karaschewski, Head of Application Development, HanseMerkur

**Smart Fixtures**

Space planning retailer featured on HGTV & ABC (US) (http://www.smartfixtures.com/)

“We had been tracking Liferay Portal for the past 2 or 3 years and knew it was the best choice for our business objectives.”

--Chris Matthews, CIO, Smart Furniture & Smart Fixtures

**Oakwood**

World’s largest rental housing provider (International) (http://www.oakwood.com/)

“Our view is that a portal is fundamentally an application integration tool rather than a solution in and of itself, so we were intent on applying our program costs to rapid customization, not license fees. This turned out to be the right mindset because within a month of initial rollout, our Liferay-based customer portal helped us edge out competitors in several hard-won contracts. Sales people actually started calling HQ to thank the development team.”

- Eric Nathanson, Oakwood

*Read more in [http://www.liferay.com/web/guest/stories](http://www.liferay.com/web/guest/stories)*
APENDIX 3. Installation and configuration

AP3.1 Prerequisites

In order to install a portal bundle you will need the following software installed:

- Java 5 or later
- A web browser such as Firefox or Internet Explorer

AP3.2 Installation in Windows

Liferay Portal is distributed with the following bundle options:

<table>
<thead>
<tr>
<th>Geronimo+Tomcat</th>
<th>JBoss+Tomcat 4.2</th>
<th>Pramati</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glassfish</td>
<td>Jetty</td>
<td>Resin</td>
</tr>
<tr>
<td>JBoss+Jetty 4.0</td>
<td>JOnAS+Jetty</td>
<td>Tomcat 5.5 for JDK 1.4</td>
</tr>
<tr>
<td>JBoss+Tomcat 4.0</td>
<td>JOnAS+Tomcat</td>
<td>Tomcat 5.5 for JDK 5.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tomcat 6.0</td>
</tr>
</tbody>
</table>

1. Download and install javaSE 5.0 and define the JAVA_HOME environment variable pointing to the installation path:

![Modify variable del sistema](Modificador_variable_del_sistema.png)

2. Download the Tomcat 5.5 for JDK 5.0 bundle from www.liferay.com/web/guest/downloads/portal

3. Unzip the bundle file. It's recommended to avoid unzipping the file within a folder which has spaces in the name or that has a parent with spaces in the name. For example, use C:\liferay\tomcat.
4. Download and execute the scripts to generate the database tables from www.liferay.com/web/guest/downloads/additional. Use the create-minimal-mysql.sql to generate the tables and the minimal necessary data.

5. Enter the folder created, and the bin subfolder. Run Liferay by executing the startup.bat script.

   Liferay will startup in 30-60 seconds on modern computers. Wait until you see:

   **INFO: Server startup in 42950 ms**

6. Once the server has already been started and if there were not any error, you will be able to enter to the main page: http://localhost:8080. In case you want to use another host you have the possibility of editing the hosts file which is in C:\WINNT\system32\drivers\etc.

   | 127.0.0.1 | localhost |
   | 127.0.0.1 | www.gavstv.cat |

7. Login as user test@liferay.com and password test. This user has all administration privileges.

**AP3.2.1 Further customize**

Liferay Portal incorporates an embedded database in its default installation. This database is fine for demoing Liferay, but for development a real SQL database server should be used. Liferay supports any database that your application server can provide access to via a JDBC data source.

To install MySQL for a Developer:

1. Download and install MySQL Server and MySQL GUI tools from its website.

2. Execute the script create-minimal-mysql.sql in order to create the necessary tables and populate them with example data. Available in www.liferay.com/web/guest/downloads/additional

3. Next, you will need to copy the MySQL JDBC driver jar file to the proper location (/lib/ext) in the Tomcat server and configure the resource in your application server to point to the recently created database. File: conf\Catalina\localhost\ROOT.xml:
4. You are ready to lunch Liferay. Navigate to the bin directory and run it using the provided script; in windows `startup.bat`. If you want to stop the server you will only have to execute the `shutdown.bat` script.

5. Go to http://localhost:8080 and the default Liferay home page will be displayed.

6. To change the default port in order to use the standard HTML port, it is necessary to edit the file `conf/server.xml`:

```xml
<Connector port="80" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"
    URIEncoding="UTF-8" />
```
**AP3.3 Development**

Necessary if you want to make changes into the source code

**AP3.3.1 Installing the Liferay Extension Environment**

Liferay has provided a means of customization which enables clear separation of customized code from the Liferay core source. This provides several benefits:

- The upgrade path is kept clear, as customizations are kept separate from the Liferay source.

- Organization or environment specific customizations don't need to be contributed back to the Liferay project.

- One, separate project for all customizations can be checked in to an organization's source code repository and applied to multiple instances of Liferay, or the reverse: an organization can have several instances of Liferay with different customizations.

The procedure for installing a Liferay extension environment is as follows:

1. Obtain a bundle.
2. Install Apache Ant 1.7.0 or above.
3. Obtain the Liferay source.
4. Create configuration files.
5. Create the extension environment source.
6. Deploy the extension environment.
1. Obtain a bundle

Download the tomcat bundle from http://www.liferay.com/web/guest/downloads/portal and unzip it into a folder called c:\liferay\tomcat.

2. Install Apache Ant 1.7.0 or above

Download the latest version of Ant from http://ant.apache.org. Uncompress the archive into an appropriate folder (c:\Archivos de programa\Java\apache-ant-1.7.0).

Next, set an environment variable called ANT_HOME which points to the folder where Ant has been installed.

To do this on a Windows platform, go to Start -> Control Panel, and double-click the System icon. Go to Advanced, and then click the Environment Variables button. Under System Variables, select New.

Make the **Variable Name** ANT_HOME and the **Variable Value** the path.

Select New again. This time name the **Variable Name** ANT_OPTS and the **Variable Value** -Xms256M -Xmx512M

Scroll down until you find the PATH environment variable. Select it and select Edit. Add %ANT_HOME%\bin.
Open a command prompt and type `ant` and press Enter. If you get a build file not found error, you have correctly installed Ant.

![Console output with 'Build failed' message]

3. Obtain the Liferay Source

Download the source from [http://www.liferay.com/web/guest/downloads/additional](http://www.liferay.com/web/guest/downloads/additional) and unzip the source code into a folder called `C:\liferay\portal`.

4. Create configuration files

There are two files which have to be customized:

- `release.properties`
- `app.server.properties`

It is necessary to create a new version of these files with the user name inserted in a dot-delimited fashion.

**release.ruben marin.properties:**

```
###
### Extension Environment
###

ip.ext.dir=c:/liferay/ext
```
This property defines where is going to be placed the extension environment.

\textit{app.server.ruben marin.properties:}

```
#
# server configuration
#
app.server.type=Tomcat
app.server.tomcat.dir=c:/liferay/tomcat

clean.log.dir=true
clean.temp.dir=true
clean.work.dir=true
```

These properties define what application server is being used and the bundle placement where the changes will be deployed.

5. Create the extension environment source

You are now ready to create your extension environment. Open up a command window in your operating system and navigate to your portal project folder. This is the folder into which you just put the customized properties files. And type the following commands:

```
C:\liferay\portal>ant clean start build-ext
BuildFile: build.xml
```

Ant will use the build script in this folder (called build.xml) to compile Liferay and create a clean extension environment in the folder you specified in the properties files.

Copy your customized properties files from your portal folder into your new ext folder. The same properties apply to the extension environment apply to the portal source project.

6. Deploy the extension environment

You are now ready to deploy the extension environment to your Tomcat instance. The contents of the portal project will be merged with the contents of the ext project and the result will be deployed to your new Tomcat instance. Type the following command:
Since you don't currently have any customizations in the extension environment, the end result will be that you will have deployed the version of Liferay whose source you downloaded, plus the sample Reports portlet that is in the extension environment by default.

**AP3.4 Install liferay as Service**

From `{tomcat_dir}/bin`, type: `service install [service_name]`

That will install Liferay as the service name, and register it as "Apache Tomcat Liferay" within Windows' Service viewer.

Then to edit, again in `{tomcat_dir}/bin`, simply type: `tomcat5w //ES//[service_name]`

This will bring up a GUI where you can then change the service's settings. You will want to do the following:

**In the Java tab:**

Make sure "Use default" is unchecked. Point your Java Virtual Machine to:

```
C:\Archivos de programa\Java\jdk1.6.0_04\jre\bin\server\jvm.dll
```

Add the arguments to the Java Options:

- `Dfile.encoding=UTF8`
- `Duser.timezone=GMT + 1`
- `Djava.security.auth.login.config=c:\liferay\tomcat\conf\jaas.config`

Finally, change the *Initial memory pool* and *MaxPermSize* to 256MB and Maximum memory pool to 1256MB
To remove the window service, from `{tomcat_dir}/bin`, type: `service remove [service_name]`

```
C:\liferay\tomcat\bin> service remove liferay
The service 'Liferay' has been removed
```

**AP3.5 LDAP**

An LDAP directory tree reflects various political, geographic, and/or organizational boundaries, depending on the model chosen. LDAP deployments today tend to use Domain name system (DNS) names for structuring the topmost levels of the hierarchy. Deeper inside the directory might appear entries representing people, organizational units, printers, documents, groups of people or anything else that represents a given tree entry (or multiple entries).

**AP3.5.1 How to connect LDAP with Liferay**

There are two ways to configure and connect the LDAP server, by the `portal-ext.properties` file or by the *Enterprise Admin portlet*. In our case we are going to use the portlet to do so.
AP3.5.2 Connection

These are the parameters required for connecting to the LDAP server.

- **Base provider url**: This tells the portal where the LDAP server is located.
- **Base DN**: this tells the parameters for the connection.
- **Principal**: username
- **Credential**: password

AP3.5.3 Users

These are filters and parameters needed for importing the users from the LDAP server.

**Authentication Search Filter**: This is the filter which determines the login, in that case will be the screen name; Liferay will search for it in LDAP and determine if there is a match, according to this filter.

**Import Search Filter**: When importing users, Liferay searches all entries that match the following search filter: `objectClass=inetOrgPerson`
AP3.5.3.1 User Mapping

These are the parameters that will be copied into Liferay database from the LDAP server.

It is important to check that there are five required fields in order for the user to be recognized by Liferay as a complete user. There must be a mapping and corresponding values in LDAP for the following fields:

- screenName
- password
- emailAddress
- firstName
- lastName
AP3.5.3.2 Groups

It is also configurable the groups where each member belongs to.

Liferay Portal can be configured to periodically check the LDAP server and add any new user accounts to the portal database. The importer is able to:

- Find new users in LDAP and add them to the portal copying the appropriate information.
- Update the information about existing users.
- Detect membership of users through groups by using a configurable attribute. LDAP groups are mapped to portal "user groups" and are created if they do not exist already.

This configuration will check and import new changes each three hours:

AP3.5.3.3 Before importing the users

It is strongly recommended that these changes are made before importing the user list because each user will have those configuration values in their account.
It is necessary to change the default language and time zone before importing all users; this is done through the same portlet in the general tag.

In addition, it is also necessary to change the default role association in order to change the default power user role for user role.

**AP3.6 Logging**

Liferay uses the *Apache Log4j* library to perform all of its logging operations. This allows for a very configurable setup with different levels of messages priority, from *FATAL* to *DEBUG*. It also allows configuring a different level of messages to be shown for every package or class within Liferay or any of its libraries.

By default, the logs are written to the standard output which means that they'll be outputted to the application server log file. *Tomcat: $TOMCAT/bin/catalina.out*
AP3.6.1 Customizing the Log4j configuration using the Enterprise Admin Portlet

The Admin portlet available to portal administrators offers a UI to dynamically change the log levels for many of the packages of Liferay. To access this UI add the Admin Portlet to a portal page.

By using the dropdown you can select a different log level for each of the paragraphs, from DEBUG to FATAL, or turn them off completely.

It is also possible to add easily your own classes in order to monitor them. This is done by adding your class as a new category and choosing the logging level.
All these changes are stored in memory, which means that they'll be lost when you reboot the server. To preserve the changes it is necessary to create a configuration file.

**AP3.6.2 Creating a custom configuration file: portal-log4j-ext.xml**

The logging configuration of Liferay is stored in a file called `portal-impl/classes/WEB-INF/portal-log4j.xml`. It's possible to override all or certain parts of that configuration by creating another file called `portal-log4.xml` in `ext/ext-impl/classes/WEB-INF` and deploying it.

Example of adding the class *JournalArticleListener* to the logging configuration file to log messages from the **DEBUG** level

```xml
<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/'>
  <category name='com.ext.portlet.journal.model.JournalArticleListener'>
    <priority value="DEBUG" />
  </category>
</log4j:configuration>
```

**AP3.6.3 Storing log files**

Up to now, all the messages are printed in the `catalina.out` file, what makes difficult to search for messages due to the large quantity. It is necessary to print the messages somewhere else.

Log4j allows logging requests to print to multiple destinations. In log4j terminology; an output destination is called an appender. The common ones are:

- **FileAppender**, it appends log events to a file.
- **RollingFileAppender**, it extends **FileAppender** to backup the log files when they reach a certain size.
- **DailyRollingFileAppender** - it extends **FileAppender** so that the underlying file is rolled over at a user chosen frequency.
- **ConsoleAppender** - it appends log events to `System.out` or `System.err` using a layout specified by the user. The default target is `System.out`.

The new appenders must be written in the file `ext/classes/log4j.properties`
This is an example of a *RollingFileAppender* configuration:

- **Type of appender**: `log4j.appender.ROL`
- **Path of the logging file**: `log4j.appender.ROL.File`
- **Maximum file size**: `log4j.appender.ROL.MaxFileSize`
- **Maximum number of files**: `log4j.appender.ROL.MaxBackupIndex`

```
l og4j.rootCategory=INFO, CONSOLE, ROL

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%d{ABSOLUTE} %5p [%c{l};%l] %m%n

log4j.appender.ROL=org.apache.log4j.RollingFileAppender
log4j.appender.ROL.File=/logs/liferay.log
log4j.appender.ROL.MaxFileSize=4000KB
log4j.appender.ROL.MaxBackupIndex=10
log4j.appender.ROL.layout=org.apache.log4j.PatternLayout
log4j.appender.ROL.layout.ConversionPattern=%d{ABSOLUTE} %5p [%c{l};%l] %m%n
```

*log4j.properties*

This appender will create up to 10 files of 4000 KB max, which will store all messages from the *INFO* logging level.
AP3.6.4 Adding log messages

When creating our new applications which need to be logged, it is necessary to include some code:

Firstly, it is created an object witch implements log4j:

```java
private static Log _log = LogFactory.getLog(JCRHook.class);
```

Then, it is used the proper method depending on the logging level preferred:

```java
if (_log.isInfoEnabled())
    _log.info("Log message");
```

AP3.7 SSL

For enabling ssl for Liferay, it is only necessary to follow the steps in enabling ssl for Tomcat.

The only additional web.xml configuration change necessary is the following. This change will take care of redirecting all http requests to https requests.

```xml
<security-constraint>
    <web-resource-collection>
        <web-resource-name> Secret Stuff </web-resource-name>
        <url-pattern>/</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```
AP3.8 Back ups

It is very important to save the changes in data and portal code. This will help to recover old versions in case of something wrong takes place.

It is necessary to back up daily:

1. **Portal source code** and **ext code** stored by the developers in the CVS server.

2. The **JackRabbit** database which contains all the pdf’s, videos and so on. Stored in `C:\Documents and Settings\Default User\liferay\jackrabbit`.

3. The **mySQL database dump file** which contains all the portals. Stored in `E:\backups\mysql_diari\lportal_backup_diari.dump`.

The mySQL dump file is created hourly using a scheduled task.

This task executes a `backup.bat` file which generates the dump file:

```
ECHO OFF
FOR /F "TOKENS=1,2 DELIMS=" %%A IN ('TIME/T') DO |
SET HOUR=%%A

mysql -u root -p -e "--DATABASES lportal" | mysql --user=root --host=localhost --batch --password

E:ackups\mysql_hora\lportal_back_%HOUR%.dump
```

*Backup.bat file*
In case of needing to restore the database dump file, it is only necessary to type this command:

```
mysql -u root -proot 1portal < 1portal.back.dump
```

**AP3.9 Languages in Liferay**

Liferay is designed to handle as many languages as you want to support. This is done by pulling out all language specific text and storing them in `language.properties` files. This way, when a page loads, Liferay will detect the language, pull up the corresponding language file and display the text in the correct language.

This allows:

- Easily support as many (or as few) languages as desire
- Have a central location for multiple languages
- Change the way the words are translated.

**AP3.9.1 Changing existing language entries**

Because of the large number of languages, not all the words are properly translated. If it is necessary to change or even create new translations is easy to do so by changing the properties files.

Example: Changing the text of an existing portlet in Liferay from *Missatges* to *Fòrum*

Liferay currently displays *Missatges* because in the file `portal/portal-ejb/classes/content/Language_ca.properties` is written the property:

```
javax.portlet.title.19=Missatges
```

To change this, modify (or create) your `Language_ca.properties` file and add the entry, create the file `ext/ext-ejb/classes/content/Language_en.properties` and override the property:

```
javax.portlet.title.19=Fòrum
```
AP3.9.2 Removing unwanted language

By default, Liferay supports all the following languages:

ar_SA, ca_AD, ca_ES, zh_CN, zh_TW, cs_CZ, nl_NL, en_US, fi_FI, fr_FR, de_DE, el_GR, hu_HU, it_IT, ja_JP, ko_KR, fa_IR, pt_BR, ru_RU, es_ES, sv_SE, tr_TR, vi_VN

If we only want to support Catalan, Spanish and English, it is only necessary to override that property in the file ext-text-impl\META-INF\portal-ext.properties:

Locales = ca_ES, es_ES, en_US

AP3.10 Google Analytics

Google Analytics is a free service offered by Google that generates detailed statistics about the visitors to a website. Its main highlight is that the product is aimed at marketers as opposed to webmasters and technologists which is where the industry of web analytics traditionally grew from.

This service can track visitors from all referrers, including search engines, display advertising, pay-per-click networks, email marketing and even digital collateral such as links within PDF documents.

It is possible to include easily this service into our portals. After creating an account in Google (gavaciutat@gmail.com) we will obtain a java script code to include into our portal:

```html
<script type="text/javascript"> var gaJsHost = (("https:" == document.location.protocol) ? "https:" : "http:"); document.write("<script src=" + gaJsHost + "google-analytics.js">" + text + "</script>" + document.location.protocol); var pageTracker = _gat._getTracker("UA-4864674-1"); pageTracker._trackPageview();</script>
```

To include this code into our portal, we will have to paste it at the end of the body of the portal_normal.vm template. Then, we create a new website profile for www.gavatv.cat.

A website profile is essentially a set of rules that define the reports that you see. Generally, a website profile corresponds with a domain – There will be one profile per domain, so that it is possible to view reports for each domain separately.
And after 24 hours, the page will start representing statistics:

Google Analytics provides the ability to add any number of users to the account, and to grant varying levels of access to the reports. So we obtain a new Google account (gavatelevisio@gmail.com) and it is added to the list of users providing only the possibility of checking the reports of www.gavatv.cat.
**AP3.11 Mail server**

In our portal there are some web mail forms in order to send emails through the website. Therefore, it is necessary to configure the tomcat server to point to the mail server. It will be also necessary to open the IMAP and SMTP ports in the server firewall.

To make the portal server point to the mail server, we will create a new resource in the file `conf/catalina/localhost/root.xml` specifying the following properties:

```
<Resource
    name="mail/MailSession"
    auth="Container"
    type="javax.mail.Session"
    mail.imap.host='172.16.0.56'
    mail.pop3.host='localhost'
    mail.smtp.host='172.16.0.56'
    mail.store.protocol="imap"
    mail.transport.protocol="smtp"
/>
```

Root.xml

It is also necessary to configure the mail domain into the admin portlet:
AP3.12 Optimizations

Liferay comes with a lot of portlets and filters bundled that are not always all necessary. There is the possibility of slimming the bundle to gain memory, CPU cycles, etc.

AP3.12.1 How to remove unused portlets

Edit the *liferay-portlet.xml* (or *liferay-portlet-ext.xml* if using the ext environment) and add an entry to every portlet that is going to be removed setting the include element to false. For example, to remove the mail portlet:

```
<portlet>
  <portlet-name>mail</portlet-name>
  <struts-path>mail</struts-path>
  <include>false</include>
</portlet>
```

Liferay-portlet-ext.xml

Liferay will completely ignore the portlet, making the load process faster and the memory consumption to decrease.

It is also possible to disable them through the `Enterprise Admin` portlet:
Click on the portlet that you want to disable and uncheck the checkbox and save:

![Administració de empreses](image)

To optimize our portal we have disabled the following portlets:

<table>
<thead>
<tr>
<th>Amazon Rankings</th>
<th>Invitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analogue Clock</td>
<td>Loan Calculator</td>
</tr>
<tr>
<td>Announcements</td>
<td>Maps</td>
</tr>
<tr>
<td>Bible Gateway</td>
<td>My Activities</td>
</tr>
<tr>
<td>Blogs</td>
<td>Network Utilities</td>
</tr>
<tr>
<td>Blogs Aggregator</td>
<td>OpenID Sign In</td>
</tr>
<tr>
<td>Calculator, Calendar</td>
<td>Our Activities</td>
</tr>
<tr>
<td>Chat</td>
<td>Password Generator</td>
</tr>
<tr>
<td>Currency Converter</td>
<td>Pipeline</td>
</tr>
<tr>
<td>Dictionary</td>
<td>Quick Note</td>
</tr>
<tr>
<td>Games</td>
<td>Random Bible Verse</td>
</tr>
<tr>
<td>Global Prayer Digest</td>
<td>Recent Bloggers</td>
</tr>
<tr>
<td>Gospel for Asia</td>
<td>Recent Documents</td>
</tr>
<tr>
<td>Hello Velocity</td>
<td>Reports</td>
</tr>
<tr>
<td>Hello World</td>
<td>SMS Text Messenger</td>
</tr>
<tr>
<td>Instant Messenger</td>
<td>Shopping</td>
</tr>
<tr>
<td></td>
<td>Software Catalogue</td>
</tr>
<tr>
<td></td>
<td>Stocks</td>
</tr>
<tr>
<td></td>
<td>Today in Christian History</td>
</tr>
<tr>
<td></td>
<td>Top Opportunities</td>
</tr>
<tr>
<td></td>
<td>Translator</td>
</tr>
<tr>
<td></td>
<td>Unit Converter</td>
</tr>
<tr>
<td></td>
<td>WSRP Proxy</td>
</tr>
<tr>
<td></td>
<td>Westminster Catechism</td>
</tr>
<tr>
<td></td>
<td>Wiki</td>
</tr>
<tr>
<td></td>
<td>Wiki Display</td>
</tr>
<tr>
<td></td>
<td>Words</td>
</tr>
<tr>
<td></td>
<td>Workflow</td>
</tr>
<tr>
<td></td>
<td>XSL Content</td>
</tr>
</tbody>
</table>
AP3.12.2 How to remove unused filters

There are also some filters which can be disabled to increase the performance. To remove them, it is only necessary to edit the file \texttt{tomcat\webapps\ROOT\WEB-INF\web.xml} commenting the \texttt{filter} and \texttt{filter-mapping} of those filters that are going to be disabled.

For our server we have disabled the following services:

- CAS,
- Compression,
- Ntlm,
- Open SSO
- Strip
APENDIX 4. Liferay plugins guide

AP4.1 Using the Plugins Software Development Kid

Plugins (portlets and themes) are the easiest way to add functionality to Liferay, as they have several benefits over using the extension environment:

- Plugins can be composed of multiple smaller portlet and theme projects. This reduces the complexity of individual projects, allowing developers to more easily divide up project functionality.

- Plugins are completely separate from the Liferay core. Portlet plugins written to the JSR-168 standard are deployable on any portlet container.

- Plugins can be hot deployed (i.e., deployed while the server is running) and are available immediately. This prevents any server downtime for deployments.

There are multiple ways to create portlet and theme plugins for Liferay. Many IDEs on the market today support portlet projects natively, and theme projects are nothing more than standard web modules with style sheets, images, and optional JavaScript and Velocity templates in them. Because of this, there are many tools which can be used to create plugins, from text editors to full blown integrated development environments.

Liferay provides a Plugins Software Development Kit (SDK) which may be used to create both portlet and theme plugins. This SDK may be used with any text editor or IDE to create plugins for Liferay. Though it is not necessary to use this SDK to create plugins, it is the recommended method.

The plugins SDK can be used to house all of your portlet and theme projects enterprise-wide, or you can have separate plugins SDK projects for each of your portal projects.

You could also use the plugins SDK as a simple cross-platform new project generator. You can generate the project using the ant scripts in the plugins SDK and then copy the resulting project from the portlets or themes folder to your IDE of choice. You would need to customize the ant script if you wish to do that, but this allows organizations which have strict standards for their Java projects to adhere to those standards.
AP4.2 Installation and configuration files

Setting up the plugins SDK is as simple as downloading the archive from http://www.liferay.com/web/guest/downloads/additional and unzip it to the location in which you will be doing your work.

It is necessary to edit the file called build.properties. This file contains the settings for where you have Liferay installed and where your deployment folder is going to be. **Do not** customize this file. Instead, create a new file in the same folder called build.username.properties, where username is your user ID on your machine.

AP4.3 Deployment

When a project has been created in the Plugins SDK, an ant script is also created for it. To deploy a plugin, it is necessary to run the deploy ant task in the project. This task will compile the plugin (theme or portlet), store it in a dist folder, and deploy the plugin to the local Liferay installation.

This is done by copying the plugin .war file to the Liferay hot deploy folder. If a local installation of Liferay is running, the plugin will be automatically picked up by the server and deployed.

Watch your Liferay console for messages. When you see in the console:

```
Registering themes for :example-theme
Themes for example-theme registered successfully
```

Your plugin has been deployed to the server and is ready for use.

If the plugin is a portlet, it will be possible to add it to a page by hovering over the Dock and clicking Add Content.

Find the portlet in the category specified in the liferay-display.xml file. If the file hasn’t been customized yet, the portlet will be in the Samples category. Simply click the Add button next to it to add it to the page.
You will likely need to customize the following properties:

```properties
#
# Specify the paths to an unzipped Tomcat bundle.
#
app.server.dir=c:/liferay/tomcat
app.server.classes.portal.dir=${app.server.portal.dir}/WEB-INF/classes
app.server.lib.global.dir=${app.server.dir}/common/lib/dep
app.server.lib.portal.dir=${app.server.portal.dir}/WEB-INF/lib
app.server.portal.dir=${app.server.dir}/webapps/ROOT

### Auto Deploy
####
#
# Plugins will be deployed to this directory. Make sure that Liferay is
# also set to scan this directory.
#
auto.deploy.dir=${user.home}/liferay/deploy
```

**app.server.dir**: This is the folder into which it has been installed the development version of Liferay.

**auto.deploy.dir**: This is the folder into which plugins should be placed in order for them to be hot deployed to Liferay. By default, this folder is in `$HOME/liferay/deploy`.

**app.server.lib.portal.dir**: This folder is where Liferay’s libraries are installed.

**app.server.portal.dir**: This folder is the folder to which Liferay is installed inside of the application server. For the Liferay-Tomcat bundle: `${app.server.dir}/webapps/ROOT`. 
AP4.4 What is a Theme plugin?

Themes are hot deployable plugins which can completely transform the look and feel of the portal.

Theme creators can make themes to provide an interface that is unique to the site that the portal will serve. Themes make it possible to change the user interface so completely that it would be difficult or impossible to tell that the site is running on Liferay.

Liferay provides a well organized, modular structure to its themes. This allows the theme developer to be able to quickly modify everything from the border around a portlet window to every object on the page, because all of the objects are easy to find. Additionally, theme developers do not have to customize every aspect of their themes.

If the plugin SDK is used, themes become only a list of differences from the default theme. This allows themes to be smaller and less cluttered with extraneous data that already exists in the default theme (such as graphics for emoticons for the message board’s portlet).

AP4.4.1 How to create a theme using plugin SDK

Creation of themes is done in a similar manner to the creation of portlets. There is a themes folder inside the plugins SDK where all new themes reside.

To create a new theme, go to the themes folder and type the following command:

```
D:\Projects\Liferay\plugins-sdk\themes>create example "example"
```

This command will create a blank theme in the themes folder to be customized.

Custom themes are based on differences between the custom code and the default Liferay theme, called Classic. There is a folder called _diffs inside of each custom theme folder.
Example of a theme folder

This is where you will place the theme code. It is only necessary to customize the parts of the theme that will differ from what is already displayed in the Classic theme. To do this, it has to be mirrored the directory structure of the Classic theme inside of the _diffs folder, placing only the folders and files you need to customize there.

### AP4.4.1.1 CSS folder

This folder contains all the css files:

- **Main.css**: this includes all the other css files.
- **Custom.css**: this file is where the developer should place all of their css that is different from the other files. All properties written here will override the ones written into the other css files.
- **Base.css**: this file contains all of the base styling that is fairly generic, such as the styling for all elements not directly related to another aspect of the site, such as the forms or navigation or dock.
• **Forms.css:** this file contains all of the styling related to the form elements of the page.

• **Layout.css:** this file contains all of the styling related to the layouts.

• **Navigation.css:** this file contains all of the styling related to the navigation, as well as the dock.

• **Portlet.css:** this file contains all of the styling related to the portlets, including the JSR-168 class names.

• **Tabs.css:** this file contains all of the styling related to the tabs in the portlets.

**AP4.4.1.2 Images folder**

This folder contains all the images of the theme. It is easy to reference the theme images directory from the css. It is only necessary to put the relative path like so:

```css
background-image: url(../images/custom/bg.gif);
```

**AP4.4.1.3 Javascript folder**

Liferay includes the JQuery javascript library, and themes developers can include any plugin that JQuery supports.

Inside of the javascript.js file, you will find three different function calls, like this:
jQuery(document).ready(
    /* This function gets loaded when all the HTML, not including the portlets, is loaded. */
    function() {
    }
);

Liferay.Portlet.ready(
    /*
     * This function gets loaded after each and every portlet on the page.
     * portletId: the current portlet's id
     * jQueryObj: the jQuery wrapped object of the current portlet
     */
    function(portletId, jQueryObj) {
    }
);

jQuery(document).last(
    /* This function gets loaded when everything, including the portlets, is on the page. */
    function() {
    }
);

- **jQuery(document).ready (fn);**

  The function is executed as soon as the HTML in the page has finished loading (minus any portlets loaded via ajax).

- **Liferay.Portlet.ready (fn);**

  The function is executed after each portlet has loaded. The function receives two variables, `portletId` and `jQueryObj`. `PortletId` is the id of the current portlet that has loaded, and `jQueryObj` is the jQuery object of the current portlet element.

- **jQuery(document).last (fn);**

  The function is executed after everything, including AJAX portlets.
**AP4.4.1.4 Templates folder**

This folder contains all the velocity templates including the HTML code and all the elements of the page.

- **Portal_normal.vm**: this file contains the overall site structure, from opening HTML tag to closing. It includes the header, and the footer, and includes the two templates dock.vm and navigation.vm and also includes the system files needed by liferay core.

- **Dock.vm**: this file contains the entire HTML code for the dock.

- **Navigation.vm**: this file contains the entire HTML code for the navigation.

- **Portal_pop_up.vm**: this file contains the entire HTML code for popup windows.

- **Portlet.vm**: this file contains the HTML code that wraps every portlet including the portlet title and portlet icons.

**AP4.5 What is a Portlet plugin?**

Portlets plugins are generic portlet applications which will give all the functionality to the portal.

**AP4.5.1 How to create a portlet using plugin SDK**

Creating portlets with the plugins SDK is a straightforward process. There is a portlets folder inside of the plugins SDK folder. This is where your portlet projects will reside. To create a new portlet, there is a script which generates all the folders and files.

In windows type: `create project-name “project name”`
You should get a **BUILD SUCCESSFUL** message from Ant, and there will now be a new folder inside of the *portlet* folder in the plugins SDK. This folder is your new portlet project. This is where you will be implementing your own functionality.

Once the project has been generated successfully, it is imported into the workspace:

To resolve the dependencies for portlet projects, it is necessary to add some externals jars:

- Common-logging.jar
- Dom4j.jar
- Portal-impl.jar
- Portal-kernel.jar
- Portal-service.jar
- Portal.jar
- Servlet.jar

A portlet project is made up at a minimum of three components:

1. Java Source
2. Configuration files
3. Client-side files (*.jsp, *.css, *.js, graphics, etc.)

These files are stored in a standard directory structure. The default portlet is configured as a standard JSR-168 portlet which uses separate JSPs for its three portlet modes (view, edit, and help).

Only the view.jsp is implemented by default; the code will need to be customized to enable the other modes.
The **Java Source** is stored in the `docroot/WEB-INF/src` folder. It is possible to go in and customize (and rename) the portlet class and add any classes necessary to implement new functionalities.

The **Configuration Files** are stored in the `docroot/WEB-INF` folder. The two standard JSR-168 portlet configuration files, `web.xml` and `portlet.xml` are here, as well as three Liferay-specific configuration files.

These files are completely optional, but are important if your portlets are going to be deployed on a Liferay Portal server:

- *liferay-display.xml*: This file describes for Liferay what category the portlet should appear under in the *Add Content* window.

- *liferay-portlet.xml*: This file describes some optional Liferay-specific enhancements for JSR-168 portlets that are installed on a Liferay Portal server. For example, you can set whether a portlet is *instanceable*, which means that you can place more than one instance on a page, and each portlet will have its own data.

- *liferay-plugin-package.properties*: This file describes the plugin to Liferay's hot deployer. One of the things that can be configured in this file is dependency .jars. If a portlet plugin has dependencies on particular .jar files that already come with Liferay, you can specify them in this file and the hot deployer will modify the .war file on deployment so that those .jars are on the class path.

**Client Side Files** are the .jsp, .css, and JavaScript files written to implement the portlet's user interface. These files should go in the `docroot` folder.
The `build.xml` file is the Ant script which is used to trigger the deploy target. This target allows to compile and deploy the new changes into the deploy folder, usually in `{user.home}/liferay/deploy` (defined in the `build.{username}.properties` file).
AP4.5.1.1 Method Init ()

```java
public void init() throws PortletException {
    editJSP = getInitParameter("edit-jsp");
    helpJSP = getInitParameter("help-jsp");
    viewJSP = getInitParameter("view-jsp");
}
```

This method initializes the jsp path of the configuration page, help page and main page defined in the `portlet.xml`

```
<init-param>
    <name>viewjsp</name>
    <value>/view.jsp</value>
</init-param>
```

AP4.5.1.2 Method include (String, RenderRequest, RenderResponse)

```java
protected void include(String path, RenderRequest req, RenderResponse res) throws IOException, PortletException {
    PortletRequestDispatcher pcd = 
        getPortletContext().getRequestDispatcher(path);

    if (pcd == null) {
        _log.error(path + " is not a valid include");
    }
    else {
        pcd.include(req, res);
    }
}
```

This method includes into the Portlet request the page specified by the variable `path`.
AP4.5.1.3 Method doDispatch (RenderRequest, RenderResponse)

```java
public void doDispatch(RenderRequest req, RenderResponse res)
    throws IOException, PortletException {
    String jspPage = req.getParameter("jspPage");
    if (jspPage != null) {
        include(jspPage, req, res);
    } else {
        super.doDispatch(req, res);
    }
```

The GenericPortlet class implementation of the render method dispatches requests to the doView, doEdit or doHelp method depending on the portlet mode indicated in the request using the doDispatch method.

AP4.5.1.4 Method doEdit (RenderRequest, RenderResponse)

```java
public void doEdit(RenderRequest req, RenderResponse res)
    throws IOException, PortletException {
    if (req.getPreferences() == null) {
        super.doEdit(req, res);
    } else {
        include("editJSP", req, res);
    }
```

Within the EDIT portlet mode, a portlet should provide content and logic that lets a user customize the behaviour of the portlet. The EDIT portlet mode may include one or more screens among which users can navigate to enter their customization data.

Typically, portlets in EDIT portlet mode will set or update portlet preferences.

Portlets are not required to support the EDIT portlet mode.

This method calls the method include (String, RenderRequest, RenderResponse) in order to dispatch the edit page only if it has been previously defined in the portlet.xml file.
When in HELP portlet mode, a portlet should provide help information about the portlet. This help information could be a simple help screen explaining the entire portlet in coherent text or it could be context-sensitive help.

Portlets are not required to support the HELP portlet mode.

This method calls the method `include(String, RenderRequest, RenderResponse)` in order to dispatch the help page defined in the `portlet.xml` file.
AP4.5.1.6 Method doView (RenderRequest, RenderResponse)

```java
public void doView(RenderRequest req, RenderResponse res)
    throws IOException, PortletException {
    include("viewJSP", req, res);
}
```

The expected functionality for a portlet in VIEW portlet mode is to generate markup reflecting the current state of the portlet. For example, the VIEW portlet mode of a portlet may include one or more screens that the user can navigate and interact with, or it may consist of static content that does not require any user interaction.

Portlets must support the VIEW portlet mode. This method calls the method `include (String, RenderRequest, RenderResponse)` in order to dispatch the main page defined in the `portlet.xml` file.

AP4.5.1.7 Method processAction (RenderRequest, RenderResponse)

```java
public void processAction(ActionRequest req, ActionResponse res)
    throws IOException, PortletException {
}
```

Typically, in response to an action request, a portlet updates state based on the information sent in the action request parameters.

The `processAction (RenderRequest, RenderResponse)` method of the Portlet interface receives two parameters, `ActionRequest` and `ActionResponse`.

The ActionRequest object provides access to information such as the parameters of the action request, the window state, the portlet mode, the portal context, the portlet session and the portlet preferences data.

While processing an action request, the portlet may instruct the portal/portlet-container to redirect the user to a specific URL. If the portlet issues a redirection, when the processAction method concludes, the portal/portlet-container must send the redirection back to the user agent and it must finalize the processing of the client request.
The portlet may also set, in the ActionResponse object, render parameters during the processing of an action request.

**AP4.6 What is a Layout template plugin?**

Layout Templates are ways of choosing how the portlets will be arranged on a page. They make up the body of the page, the large area where is possible to drag and drop portlets to create pages. Liferay Portal comes with several built-in layout templates, but it is also possible to create new ones.

Layout Templates are the easiest plugins to create. They comprise only a few files with simple table or CSS-based containers into which portlets can be dropped, as well as a thumbnail image of what the layout looks like.

**AP4.6.1 How to create a layout template using plugin SDK**

Creation of layout templates is done in a similar manner to the creation of portlets and themes. There is a *layouttpl* folder inside the plugins SDK where all new layout templates reside. To create a new layout template, you run a command in this folder similar to the one you used to create a new portlet or theme.

```
C:\liferay\plugins-sdk\layouttpl>create 3-columns "3 Columns"
Buildfile: build.xml
```

This command will create a blank layout template in your *layouttpl* folder.

**AP4.6.1.1 Anatomy of a Layout Template**

Layout Template projects are very simple.

```
3-columns-layouttpl
 WEB-INF
  geronimo-web.xml
  liferay-layout-templates.xml
  liferay-plugin-package.xml
  web.xml
  3-columns.png
  3-columns.tpl
  3-columns-wap.tpl
```
The *WEB-INF* folder contains the *xml* files which define the plug-in components (it is no necessary to edit them). The files which need to be edited are the *3-columns.* files:

**3-columns.png**: this is the thumbnail which will be displayed in the configuration page with all the other possible templates to choose.

**3-columns.tpl**: this file is where the template is defined. In that case, the code would be the following:

```html
<div class="columns-3" id="content-wrapper">
  <table class="lfr-grid" id="layout-grid">
    <tr>
      <td class="lfr-column twenty" id="column-1" valign="top">
        $processor.processColumn('column-1')
      </td>
      <td class="lfr-column sixty" id="column-2" valign="top">
        $processor.processColumn('column-2')
      </td>
      <td class="lfr-column twenty" id="column-3" valign="top">
        $processor.processColumn('column-3')
      </td>
    </tr>
  </table>
</div>
```

Each table cell has a CSS class associated with it, as well as an ID. These may be customized by modifying the theme that is being used to display the layout.
3-columns.wap.tpl: this is the file where is defined the template for mobile devices. For the WAP version, the syntax is simpler:

```html
<table>
  <tr>
    <td>
      \$processor.processColumn('column-1')
    </td>
    <td>
      \$processor.processColumn('column-2')
    </td>
    <td>
      \$processor.processColumn('column-3')
    </td>
  </tr>
</table>
```

WAP doesn't have the benefit of CSS, so we will have to settle for three evenly-spaced columns.

Once finished editing, all that is left is to deploy the layout template, as the various configuration files have been already generated properly by the Ant scripts in the Plugins SDK.
APENDIX 5. Getting started

AP5.1 Introduction

After having installed configured the application is time to start working. So it is necessary to create:

1. A community
2. A set of public or private layouts
3. Some structures and templates
4. Some articles filled with content
5. Roles for the end users

AP5.2 Communities

A community is a key concept of Liferay. By default Liferay provides a community called Guest plus a personal community for the administrator.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Members</th>
<th>Online Now</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>guest</td>
<td>Open</td>
<td>2</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>ing</td>
<td>Open</td>
<td>1</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>pan</td>
<td>Restricted</td>
<td>1</td>
<td>1</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Communities examples

The Communities portlet can be used to create as many communities as desired. Each community is formed by:

- A set of users: which are called members and which can be given different permissions within the community by assigning them Community Roles.

- A set of public and private pages: the private pages will only be seen by members while the public pages are visible to anyone. It’s optional for each community to have public pages, private pages, both or none.

- A set of permissions: that all of its members will inherit.
AP5.2.1 Create a Community

To create a community, it is done through the Communities portlet and click the add Community button to introduce its name, description and type (Open, Restricted or Private).

AP5.2.2 Assign a Community Virtual Hosting

Community Virtual Hosting allows one or more communities in a single portal instance to be identified by separate and unique host names (i.e. DNS entries).

It would appear to the users that they are visiting different websites, while they are in fact being directed to different communities inside a single web server.

Once the community has already been created through the community portlet, we go to the virtual host tab and add a public virtual host name, optionally a private virtual host name and a friendly URL.

If the virtual host name is not a public domain name which points to the web server, it will be necessary to edit the C:\WINDOWS\system32\drivers\etc\hosts file to make your browser point to the proper server:
Then, if you type in your browser www.gavatv.cat, the server Ra will display the main page of the gavatv community. Always remember to remove that entry once the domain name has been publicised.

**AP5.3 Layouts**

A layout is the technical name given to a page within a community. They have a unique internal ID known as the portlet layout Id. They are stored in the Layout table.

A layout is assigned one of four types:

1. **Portlet**: A standard page that is a container for one or more portlets.

2. **Embedded**: Embeds the contents of another URL into the page (i.e. the page contents will be the page contents of the specified URL).

3. **URL**: Selecting this page will redirect the user to the new URL. Unlike Embedded, the browser’s address will become the new URL (vs. having the contents found at the URL embedded in the portal).

4. **Article**: Unknown purpose.

**AP5.3.1 Adding a new layout**

To add a new layout, we will use the manage pages option in the gavatv community of the communities portlet. Go to the Children tab and there you will see a form to write the page name, choose the layout type and decide if this page is going to appear in the menu or is going to be hidden.
Each layout has the possibility of having its own look and feel. By default the layout will inherit the theme chosen for the root node. However, this can be changed through the *Look and Feel* tab.

**AP5.3.2 Configuration of a new Layout**

Once you have created a new layout, it is possible to add a *friendly URL* which will help to remember it.
There is also the possibility of include an icon in the browser bar:

If you don’t write any Friendly URL, it will be shown the Layout ID

<table>
<thead>
<tr>
<th>Without Friendly URL</th>
<th><a href="http://www.gavatv.cat/23">http://www.gavatv.cat/23</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Friendly URL <a href="http://www.gavatv.cat/televisio">http://www.gavatv.cat/televisio</a></td>
<td>Configuration</td>
</tr>
</tbody>
</table>

| With Friendly URL | http://www.gavatv.cat/televisio |

**AP5.4 Creating new content**

Almost all information displayed in a portal, is done through the **Journal portlet** creating a new **journal article**.

The **Journal portlet** provides a user interface for content administrators to add, delete, and update content. It also displays a list of articles that have been approved, including all versions of a given article.
To add new content is necessary to:

- Create and define all the elements of the article in a new structure (if is not already created).

- Create and define the display of the elements in a new a template and associate it to a structure (if is not already created).

- Create an article with a specific structure and concrete template filling all the elements with the proper information.

It is also necessary to understand that:

- One structure can have $n$ structures associated so an article can be represented in $n$ different ways but only one at a time.

- One article can only be associated to one structure while one structure can be associated to $n$ articles.
AP5.4.1 Structures

The structure is mainly the framework of the article; it defines through a XML schema the components and its names.

There is a UI to add new structures through the Journal portlet. Using this UI, you specify the name and description as well as all the elements [text, text box, text area (HTML), image, image gallery, document library, boolean flag, selection list, multiselection list].

After having created and defined the new structure, it will appear in a list from where it is possible to remove it, edit it, add a new structure or add a new article.
AP5.4.2 Templates

The template is responsible of displaying those elements defined in the structure and added through the article.

There is a UI to add new templates through the Journal portlet. Using this UI, you specify the name and description as well as if the content is cacheable and with which structure is associated.

![Journal](image)

**Defining a new template**

Although templates can be written in Velocity or XSL language, Velocity is faster, easier and more used.

Velocity is a Java-based template engine that provides a simple and powerful template language to reference objects. Its aim is to ensure clean separation between the presentation tier and business tiers in a Web application.
The next step is to elaborate the template code clicking over the launch editor button. If you had chosen VM (Velocity) as a language type, this would be an example:

```html
<a href="$link.getData()" target="_blank" align="middle">
  <img src="$imagen1.getData()" title="$texto-alternativo.getData()" alt="$texto-alternativo.getData()" width="$width.getData()" height="$height.getData()">
</a>
```

This template generates an image link where:

- The destination link is given by: `$link.getData()`.
- The image is given by: `$imagen1.getData()`.
- The title and the alternative text are given by: `$texto-alternativo.getData()`.
- And either the width or the height are given by: `$width.getData()` or `$height.getData()`.

After having created and defined the new template, it will appear in a list from where it is possible to remove it, edit it or add a new article.
AP5.4.2.1 Velocity Variables

Liferay exposes a large number of variables to be used by Velocity [Appendix 1.1] for example $user, $company, $layout, and $themeDisplay.

For example inside of the Classic theme’s dock.vm, there is a line of code:

```
<h2 class="user-greeting"><span> $user_greeting </span></h2>
```

Dock.vm

$user_greeting prints the Welcome Rubén sign that appears in the dock bar when I log in. This is because inside the template init.vm its being used the method $user.getGreetings ()

```
#{set ($user_greeting = $user.getGreetings())}
```

Init.vm

The variable $user is one of those variables exposed in Liferay through VelocityVariables.java

```
vo.put('user', themeDisplay.getUser());
```

VelocityVariables.java
vc is an instance of VelocityContext, and what it does is getting the user object (themeDisplay.getUser ()) and assigning it the key of user. This translates to the ability to using $user as the User object.

**AP5.4.2.2 Calling services from Velocity**

In your themes you sometimes it is necessary to invoke Liferay services. Liferay provides a utility to obtain references to services; $serviceLocator. For example:

```velocity
#set($layoutService = $serviceLocator.findService("com.liferay.portal.service.LayoutLocalService"))

#set($sanclayout = $layoutService.getLayout($layout.getAncestorLayoutId(),$layout.getOwnerId()))

#set($sanclayoutlink = $sanclayout.getFriendlyURL())
```

This method invokes the service, so it is possible to have access to its methods through a new variable, in this case layoutService.

To call services from templates is necessary to modify the portal-ext.properties and enable de property journal.template.velocity.restricted.variables. This is necessary because it is disabled by default to prevent abuse, or simply to protect less aware portal admin.
AP5.4.3 Articles

The last step is to add the article. Using the Journal portlet UI we will create an article, select a structure and choose between all the templates associated to this structure.

After having selected the structure, we will see that all the elements previously defined will appear to be filled with the proper information.

Creating a new article

Once finished entering all the information and if we have the required role permission, we will save and approve the article in order to display it (it will no be displayable if it is not approved).
AP5.5 Roles

Community Roles allow portal administrators to delegate permissions effectively in scenarios where lots of community are created and administered independently.

The main difference of Community Roles with Regular Roles is that while the latter are assigned to the whole portal, the former are assigned to a community. That means that a user can be a "Community Owner" of one community, a "Community Administrator" in another, etc.

AP5.5.1 Default community roles

Liferay Portal provides three default community roles that cannot be deleted or modified. They are:

- **Community Owner**: this role is automatically given to the creator of a community and gives him total control over the community management including website configuration and content management.

- **Community Administrator**: users with this role can administer the community but cannot assign new users or edit existing one. They can create new content in the community portlets but they cannot manage the content created by others.

- **Community Member**: It's a role that is automatically given to users when they are assigned to a community. It does not give any special right by default but can be edited by the portal administrator to add privileges that might be desirable in certain situations.

AP5.5.2 Creation and definition of community roles

In addition, portal administrators can also create and define custom community roles. The steps to do this are:

1. Go to the Roles tab in the Enterprise Admin portlet and click on Create Role.

2. Enter the name of the role and select Community in the type field select box. And click Create.
3. To define the roles click the Define Permission icon. And click the Add Portlet Permission button to add new permissions. The scope of all selected permissions will be the community to which the role is assigned.

There are seven explicit permissions for the editor gavatv role:

- Six for the Journal portlet which give the user the entirely control of the portlet.
- One for the Journal Content portlet in order to give the possibility of adding or removing the portlet borders.
AP5.5.3 Adding a new user

To add a new user use the button add user of the Enterprise Admin portlet and fill and the required parameters:

![User Add Form](image)

AP5.5.4 Assignment of community roles

To assign a community role use the communities portlet and follow these steps:

1. Click on the Assign Members icon. If the user is not yet a member associate it to the community using the Available tab.
2. The list of users in the community will be shown providing information about the assigned roles. To change the assignments click the *Assign user roles* icon. Select or deselect the check boxes for each role as desired and click *Update Associations*.

<table>
<thead>
<tr>
<th>Name</th>
<th>Screen Name</th>
<th>Community Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>gavatz_gavaz</td>
<td>gavatz9810</td>
<td>Community Member, editor gavatz</td>
</tr>
</tbody>
</table>
APENDIX 6. Letter of acknowledgment

This is a letter of recognition sent by David Achell, director of Gavà Televisió, in order to express his gratitude about the project and the way it was carried out.

Sr. Manel Marin
Dep. Serveis Informàtiques
i Noves Tecnologies

Benvolgut Manel,

M’adreço a tu per traslladar-te formalment l’agratiment de Gavà Televisió, i el meu personal, per la feina feta pel Departament d’Informàtica i Noves Tecnologies que dirigeixes en tot el procés de renovació del portal web de Gavà Televisió.

No només estem plenament satisfets del产物te final, els seus serveis i el seu nou plantejament de portal informatiu i interactiu, sinó també del procés que s’ha seguit.

Per aquest motiu no només vull felicitar-te per la tasca global del Departament, sinó també per l’equip que ha estat treballant aquests mesos juntament amb el de Gavà Televisió per desenvolupar tot el conjunt d’elements i aspectes de gestió del portal.

Un agratiment que vull personalitzar en l’Òscar Caro i el Rubén Marín, per la seva implicació professional i personal en el projecte. La sintonia amb ells a l’hora de treballar ha estat excel·lent i crec que val la pena reconèixer el seu meritat i esforç.

Moltes gràcies de nou per la feina feta.
T’agrairé que traslladies les meves felicitacions al teu equip.

Una solució, i bon estiu.

DAVID ACHELL
Director

Gavà, 10 de juliol de 2008