

Sistemas y Tecnologías Web (2013-2014)

Casiano Rodriguez Leon

11 de septiembre de 2013

1. Horarios

Tutorías:

- Localización: Despacho #96, 4ta Planta Edificio Física/Matemáticas.
- Reserva: Tfno. 922 31 81 87 Fax. 922 31 81 70
- e-mail: `casiano@ull.es`

Horario:

1. Teoría y Problemas los tenemos en el Aula 1.2 los Jueves de 10:30 a 12:20.
2. Grupo L2 los Miércoles de 10:30 a 12:20 en la sala 1.2.
3. Grupo L3 los Viernes de 8:30 a 10:20 en la sala 2.3.
4. Grupo L1 los Viernes de 10:30 a 12:20 en la sala 2.3.
5. Tutorías de Lunes a Viernes de 14:48 a 16:00 en el despacho 96 de la 4ª planta del edificio de Física/Matemáticas (`crguezl@ull.edu.es`, 922 31 81 87)

Para la tutoría virtual contacta al profesor en google plus usando una "hangout". Mi identificador g+ de ull.edu.es es 101973567008975662199. También puede ir a la página <http://gplus.to/crguezl>.

Véase también el calendario oficial de la ULL en http://www.ull.es/view/institucional/ull/Calendario_vigente/es.

”

2. Programa de Teoría

1. Diseño, desarrollo e implementación Web
2. Medios y librerías digitales
3. Arquitecturas orientadas a servicios
4. Fundamentos, sistemas, servicios y aplicaciones basados en contenidos multimedia
5. Software social y colaborativo
6. Comercio electrónico

3. Presentaciones

Cada alumno realiza una presentación de 15 minutos. El alumno realiza tres de estas exposiciones por curso. En la mayoría de los casos sobre el mismo tema en diferentes niveles de profundidad: básico, intermedio y avanzado. No obstante, es posible cambiar de tema.

Sigue una lista de temas. Pueden proponer cualquier tema que encaje en el perfil de la asignatura. He evitado poner software propietario pero si alguien quiere proponerlo porque tiene acceso a el mismo, es bienvenido.

1. HTML
 - a) HTML 5
 - b) CSS
 - c) HAML <http://haml-lang.com/>
 - d) Ruby's ERB templating system <http://rrn.dk/rubys-erb-templating-system>
2. Frameworks. Si tienes un framework web en el que estés interesado y no figure entre los de la lista, no dudes en proponerlo.
 - a) PHP
 - 1) CakePHP
 - 2) CodeIgniter
 - 3) Slim
 - b) Ruby
 - 1) Rails
 - 2) Sinatra
 - 3) Padrino

c) Python

- 1) Django
- 2) Flask
- 3) Bottle
- 4) Zope

d) Java

- 1) Grails
- 2) Spring
- 3) WaveMaker
- 4) Spark

e) Perl

- 1) Catalyst
- 2) Dancer
- 3) Mojolicious

f) ASP.NET

- 1) ASP.NET MVC Framework (ASP.NET MVC)
- 2) MonoRail
- 3) DotNetNuke
- 4) Nancy (Sinatra like)

g) ECMAScript

- 1) Flex
- 2) MonoRail
- 3) DotNetNuke

h) C++

- 1) Platinum
- 2) Tntnet

3. Google

- a)* Google App Engine
- b)* Google Apps Script
- c)* Google Web Toolkit
- d)* Chrome Developer tools

4. Javascript

- a)* Javascript
- b)* Coffescript
- c)* Ajax

- d) Node.js
 - e) JQuery <http://docs.jquery.com/>
- 5. Testing Aplicaciones Web
 - a) Selenium Grid
 - b) Cucumber
- 6. Deployment
 - a) Capistrano
 - b) Heroku <http://www.heroku.com/>
- 7. Autentificación. No dudes en proponer cualquier servicio de autentificación en el que estés interesado
 - a) SSL
 - b) CAS Central Authentication Service
 - c) OAuth Open standard for Authorization.
 - d) Internet Authentication Service (IAS)
- 8. Redes Sociales. No dudes en proponer tu propio tema aqui.
 - a) Programación de Aplicaciones Facebook
 - b) Programación de Aplicaciones G+
 - c) Redes Sociales Distribuidas
 - d) Community Manager
- 9. Comercio Electrónico
 - a) Shopping cart software
 - b) Online Banking
 - c) Dinero Electrónico
- 10. Administración y Servicios Electrónicos, Gobierno Electrónico (cosas como SARA, red europea de Administración Electrónica TESTA, @Firma, TS@, DNI electrónico, Plataforma de intermediación de datos, Red 060, Notificaciones Telemáticas Seguras, Portal de Administración Electrónica)
- 11. Medios Digitales
 - a) Publicaciones electrónicas (e-books, epub, podcasts, blogs, etc.)
 - b) Bibliotecas digitales

- c) Modelos de Negocio en las Publicaciones Electrónicas
- d) Tecnologías en las Publicaciones Electrónicas (Adobe, Apple, InformIT, etc.)

12. Programación Multimedia

- a) PIL Python Imaging Library
- b) Multimedia en iOS
- c) Rubygame
- d) SFML

13. Software Social y Colaborativo. Esto es, software diseñado para ayudar a la gente a cooperar en la realización de una tarea y a alcanzar unos objetivos que son compartidos por los participantes.

- a) Mensajería Instantánea (Como microsoft messenger)
- b) Text Chat (IRC)
- c) Software Colaborativo (groupware): Moodle, Sharepoint, Google, etc.
- d) Foros, Wikis, Blogs, Redes sociales, Sistemas de Control de Versiones, etc.

4. Proyectos

Plantaremos proyectos en los que trabajarán en grupos de no mas de 3 personas. Se pueden usar las tecnologías que el grupo decida (PHP, Java, Python, Ruby, etc.)

El proyecto usará un sistema de control de versiones (git, hg, svn) y residirá en un repositorio público como GitHub, Google-Code o Bitbucket.

Cada semana habrá una exposición por el grupo de los avances logrados en el proyecto.

Para la primer semana tendremos que hacer una propuesta inicial de grupos y proyectos y ver que tecnologías necesitamos para poder instalarlas, si es posible en el Centro de Cálculo. En caso contrario podremos trabajar con nuestros portátiles.

5. Charlas

Tendremos charlas de ex-alumnos y expertos en sistemas y Tecnologías Web a lo largo del curso que serán convenientemente anunciadas. Si conoces de un ponente interesante no dudes en proponerlo.

También usaremos cursos y charlas en la web (Coursera, Udacity, etc.). En particular:

1. Coursera. Software Engineering for SaaS por Armando Fox y David Patterson
2. Udacity. Web Development (CS253). How to Build a Blog por Steve Huffman

6. Curso de Sinatra

A lo largo del curso iremos desarrollando un curso de uso de Rack y Sinatra.

7. Programa de Prácticas

Se trata de prácticas tutorizadas y prácticas de entregar que se realizan durante la clase. Además está el proyecto final presentado por el grupo.

8. Evaluación

- La calificación se divide en dos partes: prácticas por un lado y presentaciones por otro.
- Las presentaciones e informes son el 30 % de la nota. Las prácticas y los informes de prácticas - evaluadas individualmente - son el 70 % de la nota.
- Haremos también tres micro-exámenes, aproximadamente en las semanas quinta, décima y quince para co-evaluar la parte práctica. Los microexámenes son el 30 % de la nota práctica.
- Es necesario aprobar las dos partes y cada uno de los microexámenes para superar la asignatura.

Referencias

- [1] Fox, Armando; Patterson, David. *“Engineering Long-Lasting Software. An Agile Approach Using SaaS & Cloud Computing”*. Strawberry Canyon LLC. 2012.
- [2] Fowler, Chad. *“Rails Recipes”*. The pragmatic Bookshelf. 2011.
- [3] Marshall, Kevin; Pytel, Chad; Yurek, Jon. *“Pro Active Record: Databases with Ruby and Rails.”*. Apress. 2007
- [4] Griffiths, David, *“Head First Rails: A Learner’s Companion To Ruby On Rails”*. O’Reilly. 2009.

- [5] Al Barazi, Rida; Carneiro Jr., Cloves; Carneiro, Cloves. *“Beginning Rails 3”*. Apress. 2010.
- [6] Hartl, Michael; Prochazka, Aurelius. *“RailsSpace: Building a Social Networking Website with Ruby on Rails”*. Addison-Wesley. 2007.
- [7] Bell, Gavin; *“Building Social Web Applications”*. O’Reilly. 2009.
- [8] Follansbee, Joe. *“Hands-On Guide to Streaming Media”*. Elsevier. 2006.
- [9] Christian; Laine, Jarkko, *“Beginning Ruby on Rails E-Commerce: From Novice to Professional”*. Apress. 2006.
- [10] Dix, Paul, *Service-Oriented Design With Ruby And Rails*. Addison-Wesley. 2010.
- [11] Richardson, Leonard; Ruby, Sam, *“RESTful Web Services”*. O’Reilly.
- [12] Rodríguez C. *Apuntes de la Asignatura Lenguajes y Paradigmas de Programación*
HTML en <http://nereida.deioc.ull.es/~lpp/perlexamples/perlexamples.html>. 2003.
- [13] Vogel, Lars. *Git Tutorial* <http://www.vogella.de/articles/Git/article.html> 2012.
- [14] The Git Community. *Git Community Book* <http://book.git-scm.com/> 2012.
- [15] Scott Chacon. *Pro Git* <http://progit.org/book/> 2010.